Proceedings of the Fifteenth International AAAI Conference on Web and Social Media (ICWSM 2021)

Which Node Attribute Prediction Task Are We Solving? Within-Network, Across-Network, or Across-Layer Tasks

Kristen M. Altenburger, [1] **Johan Ugander** [1]
[1] Stanford University
kristenaltenburger@gmail.com, jugander@stanford.edu

Abstract

Node attribute prediction tasks arise in a wide range of classification tasks on social networks. Examples include detecting spam accounts, identifying compromised accounts, and inferring user demographics for targeted marketing. Despite the prevalence of these types of tasks in machine learning and social science settings, clear problem definitions are lacking. Do all nodes have to be connected in a single network instance? What if there are labels in one network but not another? In this work, we propose a taxonomy that distinguishes between different node attribute prediction tasks; we formalize the existing distinction between within-network and across-network attribute prediction, which have been informally described in prior work, and also introduce a variation we call across-layer attribute prediction. With this framework in place, we observe that methods framed as applicable to across-network tasks have a history of being evaluated on across-layer problem instances. While the methods do well in the across-layer setting, we find that when evaluated in genuine across-network settings, performance can be more limited than previously suggested. We provide a way to analyze and possibly reconcile this predictive performance gap, and highlight why across-network prediction remains an important and open problem domain.

Introduction

Predicting node attributes on networks is a problem with a rich history in graph mining and the social sciences (Sen et al. 2008; Wang, Gong, and Fu 2017; Jia et al. 2017; Chakrabarti et al. 2017). The extensive literature on attribute prediction has generally categorized problems as either within-network (Bhagat, Cormode, and Muthukrishnan 2011; Desrosiers and Karypis 2009; Macskassy and Provost 2007) or across-network tasks (Lu and Getoor 2003; Craven et al. 1998), with within-network prediction receiving the most attention. The within-network setting studies a single fixed network, only partially labeled, and the prediction goal is to infer the missing labels within this single network. The across-network setting instead assumes one wants to infer the attributes of an unlabeled network based on another labeled network. Earlier work has shown promising results with transferring structural information across networks. However, in this work we identify important distinctions between the tasks that can explain predictive performance differences, and show that across-network prediction can be a more challenging task than previously appreciated. Formalizing the distinction between attribute prediction problems requires consideration of how the labeled network and unlabeled network are related to each other, if at all, in order to understand how different methods may or may not be suitable for such tasks.

Overview of attribute prediction taxonomy. Within-network attribute prediction has been widely studied, and effective methods for many domains include relatively simple approaches based on majority vote algorithms among network neighbors (Macskassy and Provost 2007), longer-range network aggregation methods based on collective inference (Neville and Jensen 2000; Jensen, Neville, and Gallagher 2004; Sen et al. 2008), LINK methods (Lu and Getoor 2003; Zheleva and Getoor 2009) that employs rows of the adjacency matrix as feature vectors, methods for semi-supervised learning on graphs (Zhu, Ghahramani, and Lafferty 2003; Zhou et al. 2004; Koutra et al. 2011), as well as methods based on embedding-based representations, both unsupervised (DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), LINE (Tang et al. 2015), and node2vec (Grover and Leskovec 2016)) and semi-supervised (Yang, Cohen, and Salakhudinov 2016). These methods for within-network prediction are fundamentally driven by various structural assumptions about how attributes are distributed within a single network of study, assuming that either homophily (McPherson, Smith-Lovin, and Cook 2001) or structural equivalence (Burt 1987; Altenburger and Ugander 2018; Peel, Delvenne, and Lambiotte 2018) governs some extent of how individuals relate to each other in the network. As such, these methods all base their predictions on similarities (of some variety) along edges or paths, and are explicitly not identifying innate structural features of nodes with given attributes.

Meanwhile, the across-network task assumes one wants to infer the attributes of unlabeled nodes in one network based on labels for nodes in another network (where no edges or paths exist to relate unlabeled to labeled nodes). For example, consider a social network of students at one college accompanied by the self-reported gender labels of these students, which we'll refer to as a *source graph*. In across-network prediction, one seeks to infer the gender labels of students in another social network for a completely different

college, which we'll refer to as a *target graph*. We assume for now that one does not observe edges, if they exist, between the source graph and the target graph. Without any way to relate unlabeled to labeled nodes, the only methods that are admissible for across-network tasks are methods that learn relationships between attribute labels and innate structural features of nodes: how many friends they have, how many triangles they belong to, and so on.

In this work, we observe that previous benchmarks for across-network tasks have been evaluated on a simplified variation of the problem that analyzed a single node set across different edge types, or what we call "across-layer" prediction. The term "across-layer" invokes the language of multilayer networks (Kivelä et al. 2014; Pilosof et al. 2017) and notes that the two networks are most accurately viewed as two different layers of a single multilayer network object. Informally, the distinction between across-network and across-layer prediction is the latter involves predictions between two "networks" that share a common node set, making it possible to bridge information from the source graph into the target graph via the known node correspondence. When previous benchmarks for across-network tasks are actually evaluated on across-network prediction settings and not across-layer settings, we find that predictive performance can suffer.

We highlight the importance of distinguishing between across-network vs. across-layer tasks because across-network prediction requires learning social structure innate to the way the nodes in the source graph are positioned that also holds true in the target graph. In certain domains this might sound like a surmountable task. For example, when seeking a predictive model of gender based on structural features, perhaps male users make systematically more or less phone calls than women, or male call patterns form many more triangles (Blumenstock and Eagle 2010; Psylla et al. 2017). One might plausibly speculate that across-network gender prediction might be a relatively easy problem. However, at least for the social settings we study, across-network prediction appears to be notably more difficult than (a) within-network prediction and (b) more difficult than what previous results on across-network prediction suggest. We will highlight in a simulated network setting when we expect across-network tasks to have more limited performance.

Overview of categorizing features. The feature representations of nodes play a critical part in determining the performance of the different prediction tasks. We categorize the features admissible for each prediction task based on two different properties: dependence on node labels and dependence on node identities. Label-dependence (Gallagher and Eliassi-Rad 2010) captures whether a node depends on attribute labels of other nodes in the graph, e.g., fraction of friends that are male (label-dependent) vs. degree (label-independent). Identity-dependence captures whether a feature depends on the exact identity of any node such as, e.g., being friends with node 17 (identity-dependent) vs. degree (identity-independent). As will be explained in more detail and is also shown in Figure 2, we explain which features are admissible for each prediction task.

Summary of contribution. The paper proceeds as follows. We first formalize a taxonomy of node attribute prediction tasks (within-network, across-network, and across-layer) and a taxonomy for node features that are admissible for these different sets of tasks (label-independent, label-dependent, and identity-dependent). Then we leverage a simulated network setting to directly compare within-network, across-network, and across-layer tasks under two data-generating-processes. Then, we describe three attribute prediction settings where we can make pairwise comparisons between methods. We observe that across-network prediction can be more difficult than previously appreciated and that across-layer problems can be easier than previous approaches may have suggested. Finally, we suggest ways to analyze and possibly close the predictive performance gap between across-network prediction and the other approaches. Amidst recent calls to establish unified benchmarks for comparing different network prediction methods (Hu et al. 2020), this work also highlights the need to clarify the type of node attribute prediction task a particular method is being evaluated on.

Within-Network, Across-Network, or Across-Layer

We define within-network, across-network, and across-layer attributed prediction tasks, as illustrated in Figure 1, based on (a) whether there is one node set of interest, consisting of labeled and unlabeled nodes, or whether there are multiple (disjoint) node sets, and (b) whether there is a single network generating process or multiple network generating processes. Next, we define label-independent and label-dependent feature representations of nodes in graphs and discuss how these representations are then admissible (or not) for each of these different tasks as shown in Figure 2.

Defining Node Attribute Prediction Tasks. The standard prediction task on networks is to begin with a graph $G = (V, E)$ that may be directed or undirected, weighted or unweighted, along with a labeling function $\ell : V \rightarrow L$ that maps nodes to some label space L. In order to connect the discussion of disparate networks (in across-network tasks), we introduce an abstraction that captures the generative process, called the *network generating process (NGP)*, of how edges form on node sets. Examples of *NGP*s include "Facebook friendships", "phone calls during month t", or "spatial proximity during month t'". This abstraction allows us to then talk about how different networks may share (or not) a common NGP. Giving consideration to when two networks share the same generative process – as a separate matter from whether they share nodes or not – gives more precision for distinguishing between types of node attribute prediction tasks.

For a generic node set V, we refer to the set of nodes whose attributes are labeled/unlabeled as V_L (labeled) and V_U (unlabeled) and assume both sets are non-empty. A network generating process $\mathcal{P}$ then maps a node set to a set of

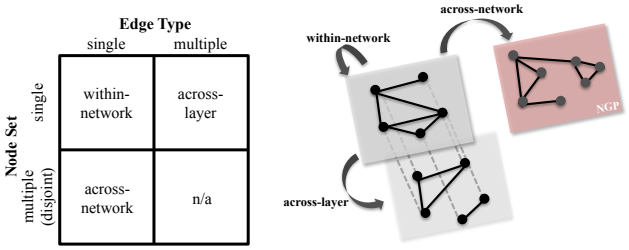

Figure 1: (Left): We illustrate a general way to classify the type of node classification problem. (Right): We visually illustrate the different node attribute prediction tasks.

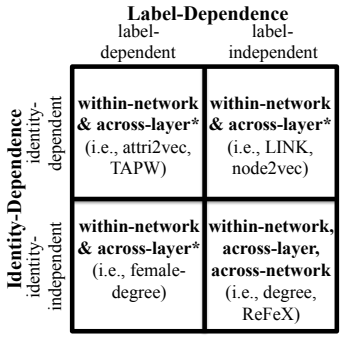

Figure 2: Characterizing which feature representations are admissible for each node prediction task. We observe that label-dependent features are never permissible for across-network tasks. We denote with * that label-dependent features can usually be used for an across-layer task as long as there is partial overlap in the node sets between layers.

edges, $\mathcal{P}(V) = E$. It will be useful to divide the edges into subsets that are between unlabeled nodes (E_{UU}), between labeled and unlabeled nodes (E_{LU}), and between labeled nodes (E_{LL}). Not all these edge sets need be non-empty.

We begin our distinction between different node prediction tasks based on the configuration of a source graph $G_s = (V_s, E_s)$ for learning a predictive model and a target graph $G_t = (V_t, E_t)$ for inferring labels. We are given attribute labels $y_i \in L$ for each node v_i in a labeled set of nodes $V_L \subset V_s$. The labels belong to a space L that can be binary (e.g., fraudulent vs. not fraudulent accounts on a transaction platform), multi-class (e.g., majors in a college network), or continuous (e.g., age). We aim to make predictions $\hat{y}_i$ for each node v_i in an unlabeled set of nodes V_U.

For within-network problems, the source and target network are the same, $V_U = V_s \setminus V_L$. For across-network problems, we assume $V_U \cap V_L = \emptyset$ and therefore $V_U = V_t$. For across-layer problems, we have $V_U \cap V_L \neq \emptyset$. In applied work, one generates a model based on V_L and predicts on V_U, ignoring settings where $V_U \cap V_L \neq \emptyset$. We simplify the node set definitions here for simplicity, though the set-up can generalize to layers which only share a subset of nodes. We refer the interested read to (Kivelä et al. 2014) which documents the history of multilayer networks and alternative set-ups throughout the literature.

Within-Network Prediction: In the setting where the target graph G_t and source graph G_s are the same, then we define a single graph $G = (V_L \cup V_U, E_{LL} \cup E_{LU} \cup E_{UU})$. We will assume that G is connected (in the case of directed graphs, weakly connected), implying that $E_{LU} \neq \emptyset$. When the network is connected, there exists an undirected path from each unlabeled node in V_U to each labeled node in V_L. Predicting attribute labels for these unlabeled nodes thus constitutes a *within-network attribute prediction* task.

In general, within-network prediction may be posed on networks of multiple connected components as long as each component contains labeled nodes (a slightly stronger, component-wise condition). If using only path-based relational methods, a problem on multiple components can then be decomposed and studied as disjoint within-network tasks on each component.

Across-Network Prediction: Next, suppose $G_s \neq G_t$ and that $V_s \cap V_t = \emptyset$, but that G_s and G_t arise separately from a common network generating process $\mathcal{P}$ of interest (i.e. friendship, calls, etc.) with $\mathcal{P}(V_s) = E_s$ and $\mathcal{P}(V_t) = E_t$. The fact that the networks arise separately means the labeled nodes set and the unlabeled node set of interest are not connected in a single network instance, so there are no paths from any node in V_s to any node in V_t. When tasked with predicting the attribute labels for the unlabeled nodes, we call this *across-network attribute prediction*.

For within-network prediction tasks where attribute labels are few and far between, predicting attribute labels for nodes far from the labeled set may mimic an across-network task. For example, in the telephone survey conducted by Blumenstock et al. (Blumenstock, Cadamuro, and On 2015) there were just 856 labeled nodes in a national call network of 1.5 million nodes. As such, very few unlabeled nodes were directly connected to any labeled node, and relatively few nodes were within 2-hops of any labeled node. That said, because of the small-world phenomenon (Travers and Milgram 1967), most nodes are within just a few hops of even a small random sample (Radaelli et al. 2018).

Across-Layer Prediction: Finally, suppose that $G_t \neq G_s$ but the source and target node sets are equivalent or have partial overlap. Further, assume there are different network generating processes $\mathcal{P}_s$ and $\mathcal{P}_t$ that underlie G_s and G_t, e.g., one NGP generates a phone call network while the other generates a text messaging network. When predicting labels for the unlabeled nodes in the target graph G_t, we call this task *across-layer attribute prediction*.

One way to see the distinction of across-layer as distinct from across-network is to consider the identifier attribute or the unique id of each node (Perlich and Provost 2006). Then in an across-layer set-up, the identifiers are preserved for different network generating processes whereas in an across-network set-up the identifiers are distinct between the nodes in the source and target graphs. An example of across-layer attribute prediction is making attribute predictions across time (Henderson et al. 2011, 2012). Consider attribute-labeled network data from one time period

(e.g., January) and the goal is to predict attribute labels in another time period (e.g., February). That is, we'd say that $V = V_{Jan} = V_{Feb}$ with $\mathcal{P}_{Jan}(V) = E_{Jan}$ and $\mathcal{P}_{Feb}(V) = E_{Feb}$. This task is also not a within-network task, as there are distinct network generating processes $\mathcal{P}_{Jan}$ and $\mathcal{P}_{Feb}$. That said, for across-layer tasks it is possible to use relational learning with label-dependent features, as we shall see in the next section, making it altogether similar to a within-network task. It is important to highlight, however, that the success of relational learning across layers relies on the general transferability of insights derived from the network generating process (NGP) of the source graph, $\mathcal{P}_s$, to the NGP of the target graph, $\mathcal{P}_t$. If, to take an extreme example, one NGP encodes friend relations while the other encodes enemy relations, we would not expect a model for predicting attributes based on friend relations to be transferable to predict attributes based on enemy relations.

Label- and Identity-Dependence

The type of node attribute prediction task determines which feature representations are admissible. A *label-dependent* node feature (Gallagher and Eliassi-Rad 2010) depends on the attribute labels of nodes in the graph, whereas a *label-independent* node feature is a feature that does not. In the context of gender prediction, clear examples of label-dependent features of a node v_i include the number of female friends of v_i or the distance from v_i to the nearest male node. Clear examples of label-independent features include the degree of v_i or the number of triangles containing v_i, neither of which depend on any node labels. When only a subset of attribute labels are known, label-dependent features are traditionally restricted to the known attribute labels (e.g., the number of known female friends of v_i).

We introduce another property of feature representations, identity-dependence, based on whether the features depend on the *identity labels* of nodes in the network. Consider LINK features (Zheleva and Getoor 2009), where rows of the graph adjacency matrix are employed as a large sparse feature vector. LINK features coupled with regularized logistic regression have been found to be highly effective when deployed for various attribute prediction tasks (Zheleva and Getoor 2009; Altenburger and Ugander 2018). These LINK features might be considered label-independent in the sense that they do not depend on any node attribute labels. LINK features do, however, depend on the identity labels, $1, \ldots, n$, of the nodes. We will thus consider features that depend on the identity labels of nodes to be *identity-dependent*. Invoking the basis used in graph theory to define a *graph invariant*, we define a feature to be *identity-independent* if it is invariant to arbitrary re-labelings of the node set, and identity-dependent otherwise.

It is clear that LINK features such as "is friends with node 17" are therefore identity-dependent, even if they are label-independent. It is also clear that structural features such as the degree of a node are both label-independent and identity-independent. Beyond LINK, we note that the feature vectors produced by embedding methods such as node2vec (Grover and Leskovec 2016) are identity-dependent, in the sense that the feature vectors of nodes from a given graph are not meaningfully related to the vectors from another graph. More specifically, even if two isomorphic graphs were passed as input into the optimization problem solved by node2vec or DeepWalk, the solutions are only defined up to rotation, making the optimization identity-dependent. Recent work on relational pooling (Murphy et al. 2019; Chen et al. 2019) and graph alignment (Bayati et al. 2009; Kuchaiev et al. 2010; Heimann et al. 2018) (and related work on aligning language embeddings (Chung et al. 2018; Lample et al. 2018)) may provide paths forward on adapting identity-dependent features (including LINK) for use in across-network tasks. This alignment will always lack guarantees as long as it falls short of solving the graph isomorphism problem (Babai 2016), which is also why embedding alignment work has generally focused on comparisons with the Weisfeiler–Leman graph kernel heuristic (Shervashidze et al. 2011; Morris et al. 2019)).

We see that in order for a feature to be useful for an across-network task it must be both label-independent and identity-independent: features such as "number of female friends" or "is friends with node 17" clearly can't be translated from a source graph containing labeled nodes, where they've been computed, to a target graph for predicting unlabeled nodes. The nodes in the target graph have no relationship to the attribute labels or identities in the source graph. Meanwhile, we highlight that identity-dependent methods are admissible for *across-layer* tasks because the node set is the same. Identity-dependent methods are also admissible even for situations where there are only a partial overlap in the node sets. We illustrate in Figure 2 which feature combinations are admissible for different attribute prediction tasks. We also observe that within-network and across-layer tasks have access to more feature sets while the across-network task relies on more limited features.

ReFeX/RolX. A prominent suite of feature representations that are both label-independent and identity-independent are the ReFeX (Recursive Feature eXtraction) (Henderson et al. 2011) and RolX (Role eXtraction) (Henderson et al. 2012) methods, initially developed for "role discovery". ReFeX and RolX both aim to learn functional roles of nodes. In the present work, ReFeX serves as an important baseline against which we seek improved methods suitable for across-network prediction.

The ReFeX method works by taking a label- and identity-independent base representation of a node (typically just in-degree, out-degree, total degree, and clustering coefficient), and recursively expanding the representation though concatenation with features variously aggregated over neighbors. These aggregations are typically limited to just averaging and/or summation, but other aggregations such as variance, entropy, or max/min aggregations are also viable, as long as the aggregations are themselves label- and identity-independent (e.g., an aggregation of the form "gender of the lowest id node" would not be identity-independent). The choice of base representation, aggregation functions, and recursion steps essentially define a ReFeX implementation.

Given ReFeX representations and corresponding known

attribute labels, one can then train a supervised learning model (e.g., a random forest or regression model). RolX extends ReFeX by applying non-negative matrix factorization to the ReFeX feature matrix as a dimensionality reduction technique, producing a possibly more interpretable (but similar performance) representation. Run for several recursive iterations, ReFeX quickly generates a vector of hundreds of features that can then be passed to a (regularized) supervised learning method. The standard implementation of ReFeX contains an internal pruning mechanism to sequentially exclude features that are highly collinear with previous features, which can speed up the feature generation process. It also speeds up the model training process, even if regularization towards sparse models can replicate the same pruning process.

It is clear that the above ReFeX representation is admissible for within-network, across-network, and across-layer attribute prediction: the node feature representation can be extracted from both a source network G_s and target network G_t without any attribute labels needed for G_t. A model trained on one network can clearly be applied to another network or another network layer.

Variations and other methods. The ReFeX idea of extracting node features from a graph recursively has been widely adopted across diverse works, though the sensitivity of such recursive feature extraction processes remains unclear. In Blumenstock et al.'s study predicting poverty and wealth from mobile phone usage in Rwanda, a recursive feature extraction algorithm based on a deterministic finite automaton (DFA) (Rabin and Scott 1959) is used. This method was used to generate thousands of features that were then used to fit regularized linear and logistic regression models, much in the same way as ReFeX. The call data record (CDR) dataset used in that study contained rich metadata (time stamped edges) and also information about cell tower connections and cell tower locations, necessitating the need for a much more general feature extraction routine than ReFeX.

While most modern graph embedding-based techniques (e.g., DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), LINE (Tang et al. 2015), node2vec (Grover and Leskovec 2016), position-aware graph neural networks (You, Ying, and Leskovec 2019)) for attribute prediction are unsuited for across-network tasks, recent work has begun to develop so-called inductive representations (a term of art from transfer learning), which for the purposes of our work means the representation is both label-independent and identity-independent) suitable for across-network attribute prediction (struc2vec (Ribeiro, Saverese, and Figueiredo 2017), graphSAGE (Hamilton, Ying, and Leskovec 2017), DeepGL (Rossi, Zhou, and Ahmed 2017), CDNE (Shen and Chung 2019), and GraphWave (Donnat et al. 2018)). These methods all employ ReFeX's basic idea of feature extraction through recursion but take added steps to optimize the feature representation itself as an inherent part of the supervised learning process. These methods are thus all much more involved than the representations we consider, training multi-layer neural network models with brittle optimization landscapes. We focus on ReFeX due to its interpretability

Simulated Networks: Predictive Performance of Label- vs. Identity-Based Features

As will be discussed in the next section, no network datasets exist to directly compare each of the different node prediction tasks. We instead rely on a stochastic block model to compare the relative performance of each under different network generating processes. We first consider when performance on an across-layer task is higher than an across-network task. Suppose a network instance is generated from a stochastic block model (SBM) with balanced classes and with strong block structure (i.e., strong homophily) (Holland, Laskey, and Leinhardt 1983). While there is structural information contained in label-dependent and identity-independent features (the relative fraction of friends from each class), this feature is not useful for inferring class labels across different networks if we do not have the labels of any nodes on the target graph we seek to make predictions in. That is, a homophily-based feature can be leveraged for learning in within-network and across-layer settings but would not be useful in across-network as illustrated in the top row of Figure 3.

First, in the top row, we consider a network with a homophily-only structure. More specifically, we generate a strongly homophilous SBM with two blocks of equal size ($n = 2000$ nodes in total, average degree of 84, and for block structure $\lambda = 1.1$). These parameters were selected to mimic some of the structures of the Amherst College network (FB100) which will be described in the next section. For all results we use a regularized Logistic Regression model. We evaluate the within-network performance of LINK and ReFeX features on this network instance across a varying percent of nodes initially labeled, and report mean AUC and variability over 25 random draws of $x\%$ of nodes initially labeled. We observe that LINK performs very well since it can leverage the similarity among *nodes* that serve as useful features. Meanwhile ReFeX is not able to leverage any structural signal because, from the perspective of ReFeX, nodes in the two blocks look identical. We see that ReFeX and related label-independent features are blind to simple, strong, but symmetric structures that do not translate easily to label-independent node features.

Next, in the top-middle row we consider an across-layer task which utilizes the same SBM set-up but varies the block structure. We then train on the fully labeled within-network instance from the within-layer setting, and report average results across multiple instances for each increasing block structure. We observe that LINK is sensitive to the increasing homophily while ReFeX again cannot utilize a homophily-only signal. Finally, we consider in the top-right row an across-network task. Here we modify the test SBM so that there are unequal classes $n_1 = 500$ and $n_2 = 1500$ unlike the training network where there are equal classes. Note that LINK is not able to be used in an across-network setting, and we observe limited predicted performance from ReFeX which is not surprising given its low performance for within-network and across-layer tasks.

In the second row, we consider a network generating process where there is both homophily (label-dependent and

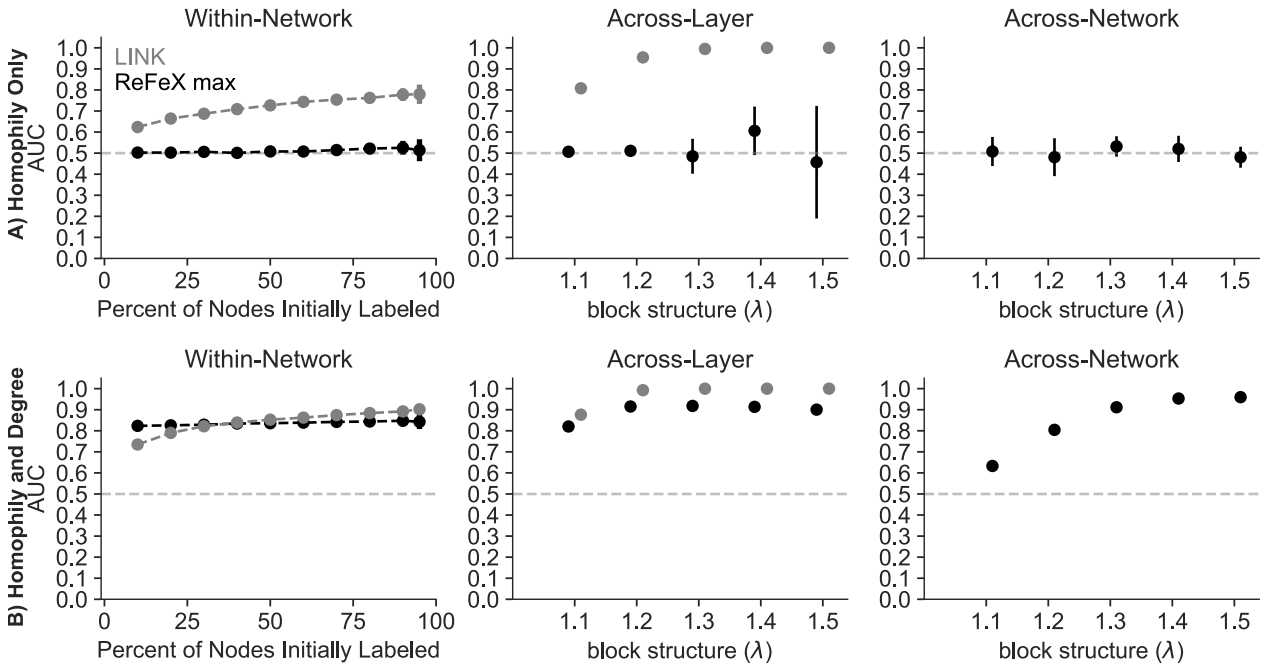

Figure 3: Within-Network vs. Across-Layer vs. Across-Network prediction on a stochastic block model (SBM). In the top row where we have homophily-only structure, we observe LINK has high within-network and across-layer performance while recursive features (ReFeX) are limited due to the lack of a structural signal. In the bottom row where we have both homophily and degree structure, we observe high performance of both methods.

identity-independent) and degree (label-independent and identity-independent) structures to leverage. This setting for the training network instance has n_1=500 and n_2=1500 nodes with an average degree of 84 and slight block structure (λ=1.1). We expect both LINK and ReFeX to perform well in this setting, which we observe in the within-network and across-layer setting. Additionally, unlike the homophily-only setting, for an across-network task, we observe high-performance of ReFeX where the test network has on n_1=800 and n_2=1200 nodes. Note that ReFeX is sensitive to increasing homophily as it's also increasing the difference in average degree between classes. Again, all results are reported in terms of mean AUC across multiple testing instances drawn from the SBM.

This simple example highlights why we might expect a method like ReFeX to be low-performing in the across-network setting when there's label-dependent and identity-independent signal only (i.e., homophily) vs. high-performing when there's label-independent and identity-independent signal (i.e., degree differences by class).

Datasets

We next demonstrate the empirical prediction performance differences for within-network, across-network, and across-layer tasks in the context of student status in the Reality Mining dataset (Eagle and Pentland 2006), gender in the Copenhagen Networks Study (Stopczynski et al. 2014; Sapiezynski et al. 2019), and gender in the Facebook100 (FB100) dataset (Traud et al. 2011; Traud, Mucha, and Porter 2012). We summarize in Table 1 the datasets considered and for which prediction task. Note we rely on these 3 datasets to make pairwise comparisons of the prediction tasks as there's not a single dataset to compare all tasks.

The Reality Mining dataset consists of a study at MIT in 2004-2005 that tracked the cell phone usage of 94 subjects, including students and faculty. The subjects agreed to have their interactions and proximity to one another recorded. One can conceive of different network attribute prediction tasks such as inferring whether a participant is a graduate student in the Media Lab or is a business school student (Henderson et al. 2012). It is possible to view this dataset through either: (a) the lens of across-layer prediction by treating each month as a different network-generating layer, or (b) the lens of within-network prediction by treating the full network of 94 subjects and their interactions over time as the single network of interest. However, the dataset has historically been sliced across time for the purpose of creating a plausible "across-network" task. As we will demonstrate, this framing leads to a very different task compared to other settings where the node population of interest is genuinely different from the source node population.

The Copenhagen Networks Study dataset consists of a survey of 800 students that recorded different types of interactions among them, including Facebook friendships, SMS messages, calls, and bluetooth co-location. We can compare predictive performance in: (a) an across-layer set-up where

	Dataset		
	Reality Mining	**Copenhagen**	**FB100**
within-network	student status prediction within MIT (Sloan/not)	gender prediction within school	gender prediction within Amherst College
across-layer	student status prediction across an academic year	gender prediction across interaction types	n/a
across-network	n/a	n/a	gender prediction across schools

Table 1: The different prediction tasks evaluated for each of the datasets.

layers represent different types of interactions recorded or (b) a within-network task by treating one interaction type as the network of interest.

Finally, the FB100 dataset consists of the online friendship networks from the first 100 colleges that accessed the Facebook platform in 2005, as released by Facebook, and includes gender, class year, and high school attributes. The FB100 dataset lends itself well to: (a) across-network tasks as we have networks across different college settings, and (b) within-network tasks where we can treat a specific school as the population network of interest.

Within-Network Prediction

In this section we compare within-network prediction tasks in three different settings: student status prediction on the Reality Mining dataset, gender prediction on the Copenhagen Facebook network, and gender prediction for a representative FB100 college network. For each problem we vary the percent of nodes, selected uniformly at random, that are initially labeled in the networks. We consider the predictive performance of statistical models using LINK, ReFeX, and node2vec features. All these feature representations are label-independent while ReFeX is identity-independent and LINK and node2vec are identity-dependent.

LINK-based models can leverage friend-of-friend information to achieve high predictive performance in the presence of monophily (Altenburger and Ugander 2018), an analog of homophily that corresponds to structural equivalence. In Figure 4 we observe that LINK-regularized logistic regression achieves high performance. As we've discussed, LINK is a label-independent and identity-dependent feature representation, making it network-specific and only applicable to within-network tasks. As such, LINK cannot be applied in across-network settings. We therefore also investigate how well we can do on this within-network task using label-independent feature representations, specifically ReFeX, which are the only feature representation permissible in an across-network setting.

Using ReFeX features, we train a regularized Logistic Regression model which is flexible in handling correlated features. The original ReFeX study paired the ReFeX features with a LogForest model, an ensemble of logistic regression models (Gallagher et al. 2008). We find approximately comparable performance for Regularized Logistic Regression instead of LogForests (when comparing only on the FB100 dataset) so use regularized Logistic Regression with ReFeX features. We compute ReFeX features in several different ways. Specifically, we vary the recursion depth of the feature generating process, extracting representations based on 1, 2, 3, and a maximum (up to 100) number of recursive iterations. The "max" recursion representations are based on only a handful of recursions, far less than 100, because of the feature pruning mechanism built into the ReFeX implementation we employed. For predicting student status in the Reality Mining data, we observe high performance of ReFeX features. For predicting gender on the Copenhagen Networks dataset, we notice that the models trained on ReFeX features do slightly worse than the models trained on LINK features. However, for gender prediction on Amherst College we observe extremely limited performance based on ReFeX features.

Across-Network vs. Across-Layer Prediction

Compared to across-layer prediction, across-network prediction can be a more difficult problem than previously appreciated as was demonstrated on the simulated network. We now illustrate this point by comparing the across-time (i.e across-layer) prediction for business school student status (Reality Mining), the across-layer prediction task for gender (Copenhagen), and the across-school (i.e. across-network) prediction task for gender (Amherst College from FB100). For the Reality Mining data, the across-layer predictions of student status are based on a time-sliced network with each month corresponding to a different network-generating layer. From month to month we are still making predictions on approximately the same node set, and arguably label-independent and identity-dependent features such as LINK are fully admissible. Similarly for the Copenhagen network, we observe different interactions among the 800 study participants with interactions like SMS messaging, Facebook friends, bluetooth co-location, and phone calls. For the across-layer prediction task, we primarily consider students who participate in a specific interaction type, which means the node sets approximately coincide across interaction types. In contrast, the across-network gender prediction task in the FB100 networks takes place across school networks with disjoint node sets.

We again consider ReFeX feature sets generated by 1, 2, 3, and a maximum number of recursive iterations, allowing us to identify at which recursion step performance gains occur. For generating ReFeX features, we note that the node representations are trained during separate ReFeX iterations, which can result in features being (a) binned differently (we standardize all features to be in $[0, 1]$) and (b) have different

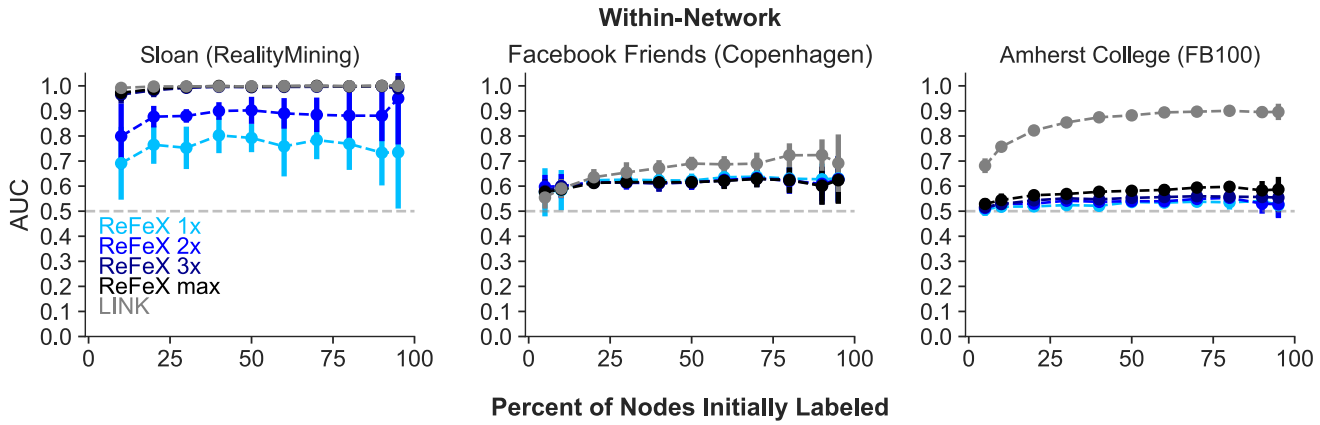

Figure 4: Within-Network Prediction: We contrast the high predictive performance of inferring whether a student is a business school student using label-independent (ReFeX) features (left) versus the more limited predicted performance of inferring gender in the Copenhagen Network study (middle) or in the FB100 dataset (right) using label-independent features.

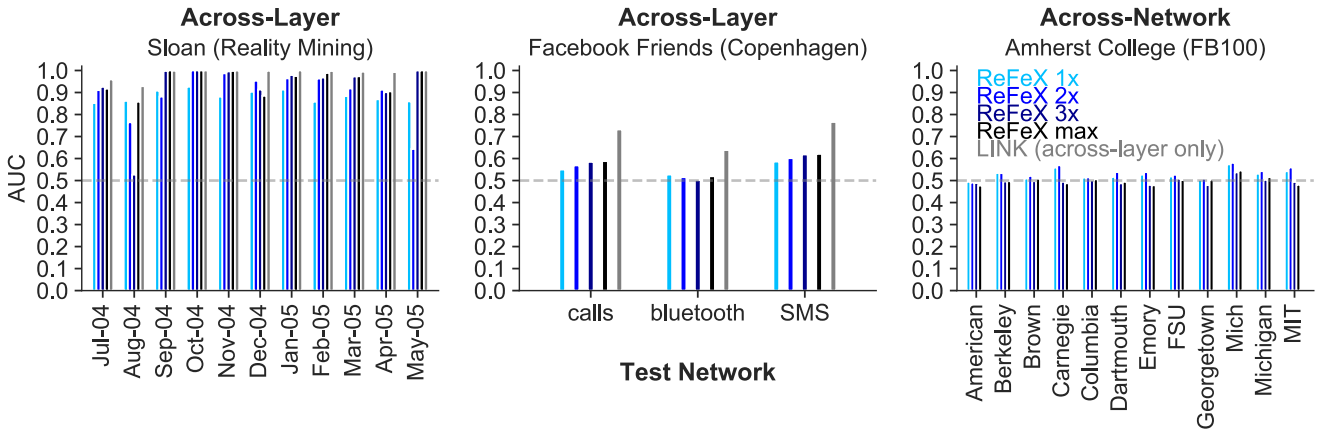

Figure 5: Across-Layer vs. Across-Network Prediction: We demonstrate across-layer prediction with the Reality Mining and Copenhagen datasets and demonstrate across-network prediction with FB100. For the across-layer task for the Reality Mining dataset, we train on the date indicated on the x-axis and report predictions for the subsequent date. For the across-layer task for the Copenhagen dataset (middle), we train a gender inference model based on Facebook friends and report predictive performance across different layers (i.e. SMS, bluetooth, calls). For the across-network task in the FB100 dataset (Amherst on right), we see no significant predictive performance gain at deeper levels of recursion.

feature sets selected by the pruning mechanism. To circumvent the latter challenge we employ a "double pass" routine to find all the features selected by the pruning mechanism across all networks in the collection in a first pass, and then repeat the feature extraction while requiring ReFeX to return the union of all these features (by overriding the pruning mechanism) as the representation.

For predicting student status in the Reality Mining dataset, we follow earlier work by using consecutive months in a paired train/test set-up, observing that this is actually an across-layer task since all training and testing nodes come from the same MIT population. For predicting gender in the Copenhagen dataset, we use the network of Facebook interactions as the training layer and compare performance when using different layers for testing. For predicting gender in the FB100 dataset, we use Amherst College as the training school and compare performance when using different schools for testing, noting this is considered an across-network task since training and testing nodes come from different college populations. Other test schools give comparable results. As illustrated in Figure 5 for the across-layer task, we observe slight performance improvement at the maximum number of recursions for ReFeX when predicting business student status in the Reality Mining dataset, though the main performance improvement (over the baseline) comes from the ReFeX base representation (ReFeX

1x) before any aggregation functions have been applied. For gender prediction in the across-network task shown in Figure 5, we observe consistently low-performing predictions across all recursive depths. Compared to the across-layer Reality Mining problem shown in the left of Figure 5 which is historically posed as an across-network task, we observe that across-network tasks can be more limited as shown on the right. For the across-layer tasks, we also report results from LINK which shows higher predictive performance, though is not useful in the across-network setting.

These simple examples demonstrate instances when within-network prediction can be high (e.g, LINK in Amherst College) but where across-network prediction performance is more limited (e.g. ReFeX across other FB100 colleges). That is, high predictive performance for within-network tasks using label-dependent features do not imply high performance for the across-network task using label-independent features. We notice in the earlier Figure 4 that the ReFeX feature set had very limited performance for within-network prediction on Amherst. Therefore, it is not surprising that there's limited performance from ReFeX features in the across-network setting as well. Meanwhile, for the across-layer setting, it is also not surprising that ReFeX performs well in this task given the high performance of label-independent features for the within-network task.

"Bridging" and Across-Network Tasks

As was emphasized in the simulated network setting, if the only structural signal present in a network is a label-dependent signal such as homophily, then across-network prediction is constrained, since label-dependent and identity-dependent information cannot transfer. For improving across-network tasks, we borrow from the *ideal points* literature in American politics, which pursues a structural approach to rank judges across time based on their voting behavior (Poole and Rosenthal 2000; Martin and Quinn 2002). This makes it possible to compare judges even when their periods of service have no overlap, addressing statistical issues such as temporal extrapolation (Ho and Quinn 2010). One of the basic insights of that literature is to leverage judges that span multiple courts. These "bridge judges" serve as a reference point for ranking judges that do not sit in the same court.

In a similar spirit, we can try to identify "bridges" in an across-network setting. If such "bridges" exist, then we can utilize LINK-based methods since there would be nodes or edges that span the networks. We explore a "bridge" approach first in a simulated example by identifying *edges* that span networks converting this to a within-network task, and then in an empirical example by identifying *nodes* that likely span both networks, converting the across-network to an across-layer task. For the simulated example, suppose we have Amherst College from the FB100 dataset that we divide into communities, and focus on two communities where we treat community 1 as the train community and community 2 as the test community. We then remove the across-community edges such that community 1 and community 2 are disconnected, creating an across-community (i.e across-network) task where LINK performance would have 0.50 AUC on the test community. By tracking the rise of LINK performance as we randomly add in across-community edges, illustrated in Figure 6, we observe that LINK (a) rises rapidly in performance after only a few edges are added and (b) has notably higher predictive performance when randomly adding back in the empirical across-community edges compared to a null process of adding in random new across-community edges between pairs of nodes. This result suggests there exist useful network structure to leverage in an across-network setting if one can identify even a few edges that connect the networks.

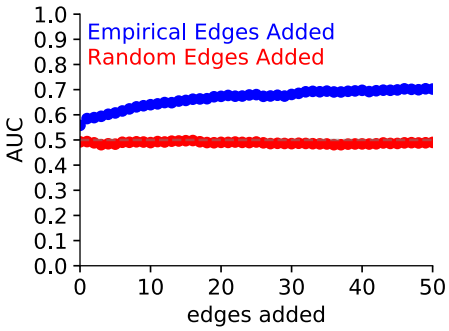

Figure 6: We compare the average predictive performance of LINK for an across-community task on Amherst College when adding back the original edges between communities versus randomly adding edges from all possible cross-community pairs of nodes.

Next, in an empirical example, we try to identify "bridge" nodes as nodes that are structurally similar across two distinct networks. This inquiry follows a larger literature on network alignment approaches (Bayati et al. 2009; Kuchaiev et al. 2010; Heimann et al. 2018), and we explore a simple strategy here as a proof-of-concept for this type of approach. We first use the ReFeX feature representation to identify structurally similar nodes as defined by those with a small Euclidean distance between their ReFeX feature representation. Then we align nodes across the two networks that are structurally similar, train a LINK model on this consolidated network representation, and evaluate results on the test network. We observe limited performance of this approach on the FB100 network. It is possible there might be limited performance because "bridge" nodes don't exist. Alternatively, it is possible there is limited performance due to our current alignment strategy.

Conclusions

This work highlights an important distinction between across-network and across-layer problems. Our investigation into across-network prediction follows other applications studying networks across varied social settings such as comparing heterogeneity across different networks (Jacobs et al. 2015), evaluating link prediction across different networks (Dong et al. 2012), and demonstrating how prediction problems can be more difficult than previously appreciated (Cohen and Ruths 2013).

Despite the impressive performance of LINK for the within-network problem, it is important to note that rela-

tional inference requires training a model specific to the network in question. As an open question, our analysis provides no theoretical basis for limiting the predictive performance of models based on ReFeX features (or any other label-independent features) relative to performance with LINK; it is possible the "right" label-independent features enable across-network predictions with performance on par with the within-network predictions based on LINK features. How should we reconsider how we think about across-network tasks? What does the transferability of a model between pairs of networks say about those networks in a larger population? A first step forward is to define a clear consistent set of across-network tasks, with many challenges to follow.

Code Availability Statement

All replication code is available at https://github.com/kaltenburger/NetworkPredictionICWSM2021

Acknowledgements

We thank Tina Eliassi-Rad and Leman Akoglu for answering questions about the Reality Mining set-up and thank Keith Henderson for clarifying ReFeX. We thank Piotr Sapiezynski for generous help with the Copenhagen Networks Study data. See Stanford IRB protocol #49530. We also thank Daniel E. Ho, Ramesh Johari, Philip Kegelmeyer, and anonymous reviewers for helpful comments.

References

Altenburger, K. M.; and Ugander, J. 2018. Monophily in social networks introduces similarity among friends-of-friends. *Nature Human Behaviour* 2(4): 284.

Babai, L. 2016. Graph isomorphism in quasipolynomial time. In *Theory of Computing*, 684–697. ACM.

Bayati, M.; Gerritsen, M.; Gleich, D. F.; Saberi, A.; and Wang, Y. 2009. Algorithms for large, sparse network alignment problems. In *ICDM*, 705–710. IEEE.

Bhagat, S.; Cormode, G.; and Muthukrishnan, S. 2011. Node classification in social networks. In *Social Network Data Analytics*, 115–148. Springer.

Blumenstock, J.; Cadamuro, G.; and On, R. 2015. Predicting poverty and wealth from mobile phone metadata. *Science* 350(6264): 1073–1076.

Blumenstock, J.; and Eagle, N. 2010. Mobile divides: gender, socioeconomic status, and mobile phone use in Rwanda. In *Conf. on Information and Communication Technologies and Development*, 6. ACM.

Burt, R. S. 1987. Social contagion and innovation: Cohesion versus structural equivalence. *AJS* 92(6): 1287–1335.

Chakrabarti, D.; Funiak, S.; Chang, J.; and Macskassy, S. A. 2017. Joint label inference in networks. *JMLR* 18(1): 1941–1979.

Chen, Z.; Villar, S.; Chen, L.; and Bruna, J. 2019. On the equivalence between graph isomorphism testing and function approximation with GNNs. *arXiv:1905.12560* .

Chung, Y.-A.; Weng, W.-H.; Tong, S.; and Glass, J. 2018. Unsupervised cross-modal alignment of speech and text embedding spaces. In *NeurIPS*, 7354–7364.

Cohen, R.; and Ruths, D. 2013. Classifying political orientation on Twitter: It's not easy! In *ICWSM*.

Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; and Slattery, S. 1998. Learning to extract symbolic knowledge from the World Wide Web. In *Conference on Artificial Intelligence*, 509–516. American Association for Artificial Intelligence.

Desrosiers, C.; and Karypis, G. 2009. Within-network classification using local structure similarity. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 260–275. Springer.

Dong, Y.; Tang, J.; Wu, S.; Tian, J.; Chawla, N. V.; Rao, J.; and Cao, H. 2012. Link prediction and recommendation across heterogeneous social networks. In *ICDM*, 181–190.

Donnat, C.; Zitnik, M.; Hallac, D.; and Leskovec, J. 2018. Learning structural node embeddings via diffusion wavelets. In *ACM SIGKDD*, 1320–1329. ACM.

Eagle, N.; and Pentland, A. 2006. Reality Mining: Sensing Complex Social Systems. *Personal and Ubiquitous Computing* 10(4): 255–268.

Gallagher, B.; and Eliassi-Rad, T. 2010. Leveraging label-independent features for classification in sparsely labeled networks: An empirical study. In *Advances in Social Network Mining and Analysis*, 1–19. Springer.

Gallagher, B.; Tong, H.; Eliassi-Rad, T.; and Faloutsos, C. 2008. Using ghost edges for classification in sparsely labeled networks. In *ACM SIGKDD*, 256–264. ACM.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *ACM SIGKDD*, 855–864. ACM.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *NeurIPS*, 1024–1034.

Heimann, M.; Shen, H.; Safavi, T.; and Koutra, D. 2018. Regal: Representation learning-based graph alignment. In *International Conference on Information and Knowledge Management*, 117–126.

Henderson, K.; Gallagher, B.; Eliassi-Rad, T.; Tong, H.; Basu, S.; Akoglu, L.; Koutra, D.; Faloutsos, C.; and Li, L. 2012. RolX: structural role extraction & mining in large graphs. In *ACM SIGKDD*, 1231–1239. ACM.

Henderson, K.; Gallagher, B.; Li, L.; Akoglu, L.; Eliassi-Rad, T.; Tong, H.; and Faloutsos, C. 2011. It's who you know: graph mining using recursive structural features. In *ACM SIGKDD*, 663–671. ACM.

Ho, D. E.; and Quinn, K. M. 2010. How not to lie with judicial votes: Misconceptions, measurement, and models. *California Law Review* 98(3): 813–876.

Holland, P. W.; Laskey, K. B.; and Leinhardt, S. 1983. Stochastic blockmodels: First steps. *Social Networks* 5(2): 109–137.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open graph benchmark: Datasets for machine learning on graphs. *arXiv:2005.00687* .

Jacobs, A. Z.; Way, S. F.; Ugander, J.; and Clauset, A. 2015. Assembling thefacebook: Using heterogeneity to understand online social network assembly. In *ACM Web Science*, 18.

Jensen, D.; Neville, J.; and Gallagher, B. 2004. Why collective inference improves relational classification. In *ACM SIGKDD*, 593–598. ACM.

Jia, J.; Wang, B.; Zhang, L.; and Gong, N. Z. 2017. AttriInfer: Inferring user attributes in online social networks using markov random fields. In *WWW*, 1561–1569.

Kivelä, M.; Arenas, A.; Barthelemy, M.; Gleeson, J. P.; Moreno, Y.; and Porter, M. A. 2014. Multilayer networks. *Journal of Complex Networks* 2(3): 203–271.

Koutra, D.; Ke, T.-Y.; Kang, U.; Chau, D. H. P.; Pao, H.-K. K.; and Faloutsos, C. 2011. Unifying guilt-by-association approaches: Theorems and fast algorithms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 245–260. Springer.

Kuchaiev, O.; Milenković, T.; Memišević, V.; Hayes, W.; and Pržulj, N. 2010. Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface* 7(50): 1341–1354.

Lample, G.; Conneau, A.; Denoyer, L.; et al. 2018. Unsupervised Machine Translation Using Monolingual Corpora Only. In *International Conference on Learning Representations*.

Lu, Q.; and Getoor, L. 2003. Link-based classification. In *ICML-03*, 496–503.

Macskassy, S. A.; and Provost, F. 2007. Classification in networked data: A toolkit and a univariate case study. *JMLR* 8: 935–983.

Martin, A. D.; and Quinn, K. M. 2002. Dynamic ideal point estimation via Markov chain Monte Carlo for the US Supreme Court, 1953–1999. *Political Analysis* 10(2): 134–153.

McPherson, M.; Smith-Lovin, L.; and Cook, J. M. 2001. Birds of a feather: Homophily in social networks. *Annual Review of Sociology* 415–444.

Morris, C.; Ritzert, M.; Fey, M.; Hamilton, W. L.; Lenssen, J. E.; Rattan, G.; and Grohe, M. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI Conference on Artificial Intelligence*, volume 33, 4602–4609.

Murphy, R.; Srinivasan, B.; Rao, V.; and Ribeiro, B. 2019. Relational Pooling for Graph Representations. In *ICML*, 4663–4673.

Neville, J.; and Jensen, D. 2000. Iterative classification in relational data. In *Proc. AAAI-2000 Workshop*, 13–20.

Peel, L.; Delvenne, J.-C.; and Lambiotte, R. 2018. Multiscale mixing patterns in networks. *PNAS* 115(16): 4057–4062.

Perlich, C.; and Provost, F. 2006. Distribution-based aggregation for relational learning with identifier attributes. *Machine Learning* 62(1-2): 65–105.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *ACM SIGKDD*, 701–710. ACM.

Pilosof, S.; Porter, M. A.; Pascual, M.; and Kéfi, S. 2017. The multilayer nature of ecological networks. *Nature Ecology & Evolution* 1(4): 0101.

Poole, K. T.; and Rosenthal, H. 2000. *Congress: A political-economic history of roll call voting*. Oxford University Press on Demand.

Psylla, I.; Sapiezynski, P.; Mones, E.; and Lehmann, S. 2017. The role of gender in social network organization. *PloS One* 12(12).

Rabin, M. O.; and Scott, D. 1959. Finite automata and their decision problems. *Journal of Research and Development* 3(2): 114–125.

Radaelli, L.; Sapiezynski, P.; Houssiau, F.; Shmueli, E.; and de Montjoye, Y.-A. 2018. Quantifying Surveillance in the Networked Age: Node-based Intrusions and Group Privacy. *arXiv:1803.09007* .

Ribeiro, L. F.; Saverese, P. H.; and Figueiredo, D. R. 2017. struc2vec: Learning node representations from structural identity. In *ACM SIGKDD*, 385–394. ACM.

Rossi, R. A.; Zhou, R.; and Ahmed, N. K. 2017. Deep feature learning for graphs. *arXiv:1704.08829* .

Sapiezynski, P.; Stopczynski, A.; Lassen, D. D.; and Lehmann, S. 2019. Interaction data from the copenhagen networks study. *Scientific Data* 6(1): 1–10.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective Classification in Network Data. *AI Magazine* 29(3): 93.

Shen, X.; and Chung, F.-l. 2019. Network Embedding for Cross-network Node Classification. *arXiv:1901.07264* .

Shervashidze, N.; Schweitzer, P.; Leeuwen, E. J. v.; Mehlhorn, K.; and Borgwardt, K. M. 2011. Weisfeiler-lehman graph kernels. *JMLR* 12(Sep): 2539–2561.

Stopczynski, A.; Sekara, V.; Sapiezynski, P.; Cuttone, A.; Madsen, M. M.; Larsen, J. E.; and Lehmann, S. 2014. Measuring large-scale social networks with high resolution. *PloS One* 9(4).

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *WWW*, 1067–1077.

Traud, A. L.; Kelsic, E. D.; Mucha, P. J.; and Porter, M. A. 2011. Comparing community structure to characteristics in online collegiate social networks. *SIAM Review* 53(3): 526–543.

Traud, A. L.; Mucha, P. J.; and Porter, M. A. 2012. Social structure of Facebook networks. *Physica A* 391(16): 4165–4180.

Travers, J.; and Milgram, S. 1967. The small world problem. *Psychology Today* 1(1): 61–67.

Wang, B.; Gong, N. Z.; and Fu, H. 2017. GANG: Detecting fraudulent users in online social networks via guilt-by-association on directed graphs. In *ICDM*, 465–474. IEEE.

Yang, Z.; Cohen, W.; and Salakhudinov, R. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *ICML*, 40–48.

You, J.; Ying, R.; and Leskovec, J. 2019. Position-aware Graph Neural Networks. In *ICML*, 7134–7143.

Zheleva, E.; and Getoor, L. 2009. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *WWW*, 531–540.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2004. Learning with local and global consistency. In *NeurIPS*, 321–328.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, 912–919.