Link Prediction by Exploiting Network Formation Games in Exchangeable Graphs

Liqiang Wang[†], Yafang Wang[†]*, Bin Liu[§], Lirong He[§], Shijun Liu[†]*, Gerard de Melo[‡], Zenglin Xu[§]*
[†]Shandong University, China
[§]Big Data Res. Center, School of Comp. Sci. and Engin., Univ. of Electr. Sci. and Techn. of China, China
[‡]Rutgers University, New Brunswick, USA

***Abstract*—In social network analysis, we often need to predict new links, given some available evidence. This may, for instance, enable us to study user behavior and infer likely new interactions in the near future. Recently, a family of algorithms based on exchangeable graphs has proven effective for link prediction. The network is modeled as an exchangeable array, whose entries can flexibly be traced back to random function priors (e.g., block models, Gaussian Processes). Unfortunately, the burdensome computational complexity of these methods inhibit their application to even just moderate-scale networks. In this paper, we present a novel online training algorithm based on local Gaussian processes on subgraphs, which successfully overcomes this challenge. Moreover, we address the sparsity problem of links in social networks by presenting an improved algorithm based on network formation games. The network formation games we design also shed light on the ambiguity of missing links – not observed vs. non-existing. We evaluate our method against state-of-the-art algorithms on real-world datasets, demonstrating both the effectiveness and the efficiency of our method.**

***Keywords*—link prediction; network formation games; exchangeable graphs; graphical model**

I. Introduction

Motivation. With over a billion active users every single day on Facebook alone, online social networks have become vital platforms for sharing information and interacting with others via postings, messaging, games, and applications. Accordingly, this rich new source of data has also fostered an ecosystem of novel user-centric services and tools.

Realizing the potential of this data often hinges on our ability to model the social network adequately for a given purpose. Oftentimes, we are interested in predicting missing links or predicting potential interactions in the near future. Since only past interactions with the system can be observed, one often relies on latent variable models that explain network data in terms of underlying structures or summaries, such as low-rank approximations of an observed array or an embedding in a Euclidean space (e.g., via matrix factorization [26], [12] or stochastic block models [9], [1]). Many existing approaches [8], [20], [3], [19] can be regarded as a family of approaches based on exchangeable arrays. The notion of exchangeable arrays here reflects the observation that users in a social network have no natural ordering. Recent research has shown that random arrays that satisfy an exchangeability property can be represented in terms of a random function [19], [22]. Therefore, by specifying a prior on the random (measurable) functions, informative priors for exchangeable random graphs can be introduced, giving rise to more powerful Bayesian network modeling approaches. These models are called Random Function Models (RFM), and turn out to be particularly useful for social network analysis.

Unfortunately, despite the powerful modeling abilities of Bayesian network models of the sort mentioned above, there are significant computational and modeling challenges. First, the prohibitive computational complexity of these methods inhibits their use on even just moderately sized social network data. In particular, a Bayesian modeling of the network links usually entails a high computational complexity, although approximation methods may be exploited. For example, the matrix-variate distributions approach [33], [31], [32] requires the computation of the Kronecker product, while the approach of Gaussian processes on network entries [19] involves a training set of n^2 instances (i.e., network links) for n nodes. Second, available network data are generally sparse and most existing methods have not fully accounted for this property. In particular, while we can view existing links in the network as positive examples (1-links), we cannot take for granted that the absence of a connection between two nodes implies that we have a genuine negative example (0-links). Many such cases may simply be missing links or potential future links.

Contributions. In this paper, we improve the exchangeable arrays method for link prediction in several important ways. To enhance its scalability, we derive an online variational inference algorithm to efficiently learn latent Gaussian processes on exchangeable graphs, thereby greatly reducing the computational complexity in comparison with the batch counterpart. The derived inference algorithm also enables parallel computation over network nodes. The training data is sliced into several sub-blocks, and each of these are generated from a local latent variate Gaussian Process. To further improve the efficiency of the training process and balance the training data, we rely on Network Formation Games (NFGs) to select the most valuable negative samples.

In order to evaluate the proposed techniques, we conduct experiments on a set of real-world network datasets. Our experimental findings demonstrate the superiority of our method over existing random function models both in terms of the predictive performance and computational efficiency.

* The three corresponding authors: {yafang.wang, lsj}@sdu.edu.cn, zenglin@gmail.com

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

II. Related Work

Before introducing the details of our model, we first review relevant previous work.

Exchangeable Arrays and LFM. A number of works have explored the use of exchangeable arrays and random function priors to model relational data [19], [22]. Nonparametric models of networks include the infinite relational model (IRM) [10], latent feature relational model (LFRM) [21], and infinite latent attribute model (ILA) [23], among others. Lloyd et al. [19] introduced a novel latent variable model (RFM) between the entities of two arrays with random function priors, which may also be applied in an n-arrays setting. Based on the random priors, the RFM method learns latent parameters with partial observations of 2-arrays and n-arrays. The Eigenmodel [2] defined exchangeability arguments representing the relationships between two nodes as the weighted inner product of node-specific vectors of latent characteristics. GPLVM [11] interprets PCA as a specific Gaussian prior on a mapping from a latent space to the data space. It assumes independence of the rows or the columns of the random array. Compared to the GPLVM, SMGB [33] uses both row- and column-wise covariance functions to represent row- and column-wise interactions. Unfortunately, owing to their high computational complexity, all of the above methods suffer from limited scalability, severely restricting their applicability to large real-world datasets.

Network Formation Models. How to form networks has received significant attention. There are two main kinds of network formation models. The first comprises dynamic ones [25], [30] derived from basic network structure features such as the average path length, degree distribution, and clustering coefficient. The second form consists of agent-based models drawing on game theory. Here, entities are modeled as pursuing an objective of maximizing a benefit, while incurring costs for maintaining relationships with others. Hence, the game process can reflect the formation of a network, and the final network structure emerges as an equilibrium in such games. There are surveys that provide an overview of state-of-the-art formation models [7] as well as analyses of their stability and efficiency [5].

Link Prediction. Link Prediction, a form of one-class recommendation [24], is one of the core problems in social network analytics. Traditional methods are based mostly on the network topology, often relying on similarity measures between two entities [18], [28], [13]. Such measures can be recast as representing the probability of a link formation between them. Typically, the measure is computed based on the network structure and the attributes of the entities. Liben-Nowell and Kleinberg [17] presented a survey of this sort of link prediction. An alternative form of link prediction is to extract features and patterns from the network and rely on regular supervised learning to predict the links [14], [4], [34]. Recently, Zhao et al. [35] incorporated game theory into machine learning models to improve the recommendation accuracy. This model forms a pairwise ranking based on the Bayesian Personalized Ranking framework, aiming at a one-class recommendation per user. Hence, it is not applicable to our link prediction setting.

III. Our Model

Figure 1 illustrates the graphical model used in our framework, while the online link prediction algorithm is given in Algorithm 1. Details are provided in the following subsections.

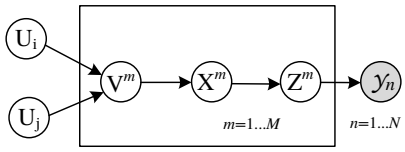


Fig. 1: *The graphical model representation.*

A. Exchangeable Arrays

In order to model social networks as random exchangeable arrays, we first introduce the De Finiti Theorem for random arrays.

Theorem III.1. *[Aldous, Hoover] A random 2-array (Y_{ij}) is exchangeable if and only if there is a random (measurable) function* $\mathrm{F} : [0,1]^3 \to Y$ *such that*

$$Y_{ij} := \mathrm{F}(U_i, U_j, U_{ij}) \tag{1}$$

for every collection $(U_i)_{i \in N}$ *for i.i.d. uniform* $[0,1]$ *random variables, where* $U_{ji} = U_{ij}$ *for all* $j < i \in N$.

For undirected graphs, the representation in (1) can be simplified further: There is a random function $\theta : [0,1]^2 \to [0,1]$, symmetric in its arguments, such that

$$\mathrm{F}(U_i, U_j, U_{ij}) := \begin{cases} 1 & \text{if } U_{ij} < \theta(U_i, U_j) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

satisfies (1). Let the variable U_i denote the vertex i, U_j denote the vertex j, and the variables U_{ij} refer to the edge associated with i and j. Representation (1) is equivalent to the sampling scheme

$$U_1, U_2, \cdots \sim_{iid} \mathrm{Uniform}(0,1) \tag{3}$$

$$Y_{ij} = Y_{ji} \sim \mathrm{Bernouli}(\theta(U_i, U_j)) \tag{4}$$

B. Random Functions with Gaussian Process Priors

In Theorem III.1, the distributions of nodes are sampled from a 2-dimensional uniform distribution, i.e., $U_i \sim \mathrm{Uniform}(0,1)$. However, the uniform distribution may underestimate complex interactions among nodes. Therefore, we follow [19] in using more powerful non-atomic probability measures. In particular, we extend U_i from a 2-dimensional uniform distribution to an r-dimensional multivariate norm distribution. That is,

$$U_i, U_j \sim N(0, I_r), 1 \leq i, j \leq n. \tag{5}$$

For convenience, we introduce an auxiliary variable $\mathbf{v}$ to denote the concatenation between nodes, i.e.,

$$\mathbf{v} \sim N(0, I_{2r}), \mathbf{v}_k = [U_i, U_j], \mathbf{v}_k \in \mathbb{R}^{2r}, 1 \leq k \leq n(n-1)/2$$

Let $\mathbf{y}$ denote a vector that represents the links among users, where $y_i = 1$ indicates that the i-th link is valid, and $y_i = 0$ otherwise. We assume the elements of $\mathbf{y}$ are conditionally independent given a latent continuous variable $\mathbf{x}$. These two matrices are linked together via probit functions:

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{1 \leq i \leq n(n-1)/2} \Phi(x_i)^{y_i}(1 - \Phi(x_i))^{(1-y_i)} \tag{6}$$

Applying the theory of exchangeable arrays to social networks, it is reasonable to model the latent variable $\mathbf{x}$ with the auxiliary variable $\mathbf{v}$. Since our work is in the framework of Gaussian process regression, this is an empirical way of assigning a Gaussian process prior for $\mathbf{x}$, i.e.,

$$p(\mathbf{x} \mid \mathbf{v}) = \mathcal{N}(0, \mathbf{K}), \tag{7}$$

where $\mathbf{K}$ is the covariance matrix defined on $\mathbf{v}$ (i.e., a Gaussian Process with zero mean and covariance matrix $\mathbf{K}$, where the the i, j-th entry of $\mathbf{K}$ is some kernel function evaluated at $\mathbf{v}_i$, $\mathbf{v}_j$). Equivalently, one may assign $\vartheta \sim \mathcal{GP}(0, \mathbf{K})$ in Eq. (2).

In order to model the relationship between the random function variables and $\mathbf{y}$ within the Bayesian framework, we extend Eq. (6) with the probit model [33]. This is achieved by introducing an auxiliary variable $\mathbf{z}$ with the same dimensionality as $\mathbf{y}$, and then decomposing the probit model as Eq. (8), where $\mathbf{1}(\cdot)$ denotes the indicator function, yielding 1 if the statement argument is true, and 0 otherwise. $\mathcal{N}(\mathbf{z}_i; \mathbf{x}_i, 1)$ is the univariate normal probability density function with mean $\mathbf{x}_i$ and variance 1.

$$\begin{aligned} p(\mathbf{y}_i \mid \mathbf{z}_i, \mathbf{x}_i) &= \{\mathbf{1}(\mathbf{y}_i = 1)\mathbf{1}(\mathbf{z}_i > 0) + \mathbf{1}(\mathbf{y}_i = 0)\mathbf{1}(\mathbf{z}_i \leq 0)\} \\ p(\mathbf{z}_i \mid \mathbf{x}_i) &= \mathcal{N}(\mathbf{z}_i; \mathbf{x}_i, 1)) \end{aligned} \tag{8}$$

According to Eqs. (5), (13), and (15), the joint distribution of our model is

$$\begin{aligned} p(\mathbf{y}, \mathbf{z}, \mathbf{x}, \mathbf{v}) &= p(\mathbf{v})\ p(\mathbf{x} \mid \mathbf{v})\ p(\mathbf{z} \mid \mathbf{x})\ p(\mathbf{y} \mid \mathbf{z}) \\ &= \mathcal{N}(\mathbf{v}; 0, I_{2r})\ \mathcal{N}(\mathbf{x}; 0, \mathbf{K}) \cdot \\ &\quad \prod_{1 \leq i \leq n(n-1)/2} p(\mathbf{y}_i \mid \mathbf{z}_i, \mathbf{x}_i)\ p(\mathbf{z}_i \mid \mathbf{x}_i) \end{aligned} \tag{9}$$

We rely on a divide-and-conquer strategy to partition the whole graph into non-overlapping sub-blocks $\{G_1, G_2, \ldots, G_M\}$ of equal size τ. Given a sub-block G_m, the latent variable $\mathbf{x}$ is generated by a local $\mathcal{GP}(0, \mathbf{K}^m)$. The latent variable $\mathbf{z}^m$ is generated by $p(\mathbf{z}_i^m \mid \mathbf{x}_i^m) = \mathcal{N}(\mathbf{z}_i^m; \mathbf{x}_i^m, 1)$. Thus, the joint probability of our model is

$$\begin{aligned} p(\mathbf{y}^m, \mathbf{z}^m, \mathbf{x}^m, \mathbf{v}^m) &= p(\mathbf{v}^m)p(\mathbf{x}^m \mid \mathbf{v}^m)p(\mathbf{z}^m \mid \mathbf{x}^m)p(\mathbf{y}^m \mid \mathbf{z}^m) \\ &= \mathcal{N}(\mathbf{v}^m; 0, I_{2r})\, \mathcal{N}(\mathbf{x}^m; 0, \sigma) \cdot \\ &\quad \prod_{1 \leq i \leq n(n-1)/2} p(\mathbf{y}_i^m \mid \mathbf{z}_i^m, \mathbf{x}_i^m) p(\mathbf{z}_i^m \mid \mathbf{x}_i^m) \end{aligned} \tag{10}$$

C. Variational Expectation Maximization

Based on this model, we rely on variational expectation maximization, which consists of two steps: the variational E-step and a gradient-based M-step. For each sub-block, we apply variational-EM independently. Given sub-block G_m, the E-step approximates the posterior distribution $p(\mathbf{z}^m, \mathbf{x}^m \mid \mathbf{y}^m, \mathbf{v}^m)$ by a fully factorized distribution

$$q(\mathbf{z}^m, \mathbf{x}^m) = q(\mathbf{z}^m)q(\mathbf{x}^m) \tag{11}$$

Variational inference minimizes the KL divergence between the approximate and the exact posteriors

$$\min_q KL(q(\mathbf{z}^m)q(\mathbf{x}^m) || p(\mathbf{z}^m, \mathbf{x}^m \mid \mathbf{y}^m, \mathbf{v}^m)). \tag{12}$$

We iteratively update $q(\mathbf{z}^m)$ and $q(\mathbf{x}^m)$. The derivation of the updates is similar to [33]. Let $\Sigma_{\mathbf{x}}^m = \mathbf{K}^m(\mathbf{K}^m + \mathbf{I})^{-1}$. Given the current $q(\mathbf{z}^m)$, we update $q(\mathbf{x}^m)$

$$\langle \mathbf{x} \rangle^m = \mathbf{K}^m(\mathbf{K}^m + \mathbf{I})^{-1}\langle \mathbf{z}^m \rangle \tag{13}$$

Here, $\langle \cdot \rangle$ denotes the expectation. The variational distribution $q(\mathbf{z}_i^m)$ is a truncated normal distribution, which is updated as follows

$$q(\mathbf{z}_i^m) \propto N(\langle \mathbf{x}_i^m \rangle, \mathbf{1})\, \mathbf{1}(\mathbf{z}_i^m > 0) \tag{14}$$

The mean of the normal distribution is adjusted by

$$\langle \mathbf{z}^m \rangle = \langle \mathbf{x}^m \rangle + \frac{(2\mathbf{y}^m - 1)\, \mathcal{N}(\langle \mathbf{x}^m \rangle; 0, 1)}{\Phi((2\mathbf{y}^m - 1)\langle \mathbf{x}^m \rangle)} \tag{15}$$

Next, we maximize the expected log-probability of the joint model over $\mathbf{v}^m$ in the M-step.

$$\max_{\mathbf{v}^m} \mathbb{E}_q[\log\, p(\mathbf{y}^m, \mathbf{z}^m, \mathbf{x}^m \mid \mathbf{v}^m)\, p(\mathbf{v}^m)] \tag{16}$$

Eliminating constant terms in the above equation, we obtain the following optimization problem:

$$\begin{aligned} \max_{\mathbf{v}^m} f(\mathbf{v}^m) = &-\frac{n}{2}\log|\mathbf{K}^m| - \frac{1}{2}\mathrm{tr}([\mathbf{K}^m]^{-1}\langle \mathbf{x}^m \rangle \langle \mathbf{x}^m \rangle^\top) \\ &- \frac{1}{2}\mathrm{tr}([\mathbf{K}^m]^{-1}\Sigma_x) - \lambda\|\mathbf{v}^m\|_1 + const. \end{aligned} \tag{17}$$

Omitting the l_1 penalization term ($\lambda\|\mathbf{v}^m\|_1$), the gradient of the first three terms in f *w.r.t.* a scalar $\mathbf{v}_{kr}^m$, the r-th element of $\mathbf{v}_k^m$, is Eq. (18). Then a variant of the L-BFGS method is used to optimize $f(\mathbf{v}^m)$ with the l_1 penalty [27].

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{v}_{kr}^m} = &-\frac{n}{2}\mathbf{tr}\left([\mathbf{K}^m]^{-1}\frac{\partial \mathbf{K}^m}{\partial v_{kr}^m}\right) \\ &+ \frac{1}{2}\mathbf{tr}\left[[\mathbf{K}^m]^{-1}\frac{\partial \mathbf{K}^m}{\partial v_{kr}^m}[\mathbf{K}^m]^{-1}\langle \mathbf{x}^m \rangle \langle \mathbf{x}^m \rangle^\top\right] \\ &+ \frac{1}{2}\mathbf{tr}\left[[\mathbf{K}^m]^{-1}\frac{\partial \mathbf{K}^m}{\partial v_{kr}^m}[\mathbf{K}^m]^{-1}\Sigma_x\right] \end{aligned} \tag{18}$$

D. Prediction

Given the missing value set $\mathcal{P}$ and index $i \in \{i_1, \ldots, i_{|\mathcal{P}|}\}$ as well as the observed data $\mathcal{Y}$, the predictive distribution is

$$\begin{aligned} p(y_i \mid \mathcal{Y}) &\approx \int p(y_i = 1 \mid z_i) p(z_i \mid x_i) p(x_i \mid \mathbf{x}) q(\mathbf{x}) dz_i dx_i d\mathbf{x} \\ &= \int \mathbf{1}(z_i > 0) \mathcal{N}(z_i | \mu_i(1), \nu_i^2(1))\, dz_i = \Phi\left(\frac{\mu_i(1)}{\nu_i^2(1)}\right) \end{aligned} \tag{19}$$

Let $\mathbb{O}$ denote the indices of the observed entries in $\mathcal{Y}$.

$$K_{ij} = kernel(\mathbf{v}_i, \mathbf{v}_j), \mathbf{k} = [K_{ij}]_{j \in \mathbb{O}}^\top, \hat{\mathbf{k}} = [K_{jj}]_{j \in \mathbb{O}}$$
$$\mu_i(\rho) = \mathbf{k}^\top (\hat{\mathbf{k}} + \rho^2 \mathbf{I})^{-1} \vec{(}\mathcal{Y})$$
$$\nu_i^2(\rho) = K_{ii} - \mathbf{k}^\top (\hat{\mathbf{k}} + \rho^2 \mathbf{I})^{-1} \mathbf{k}$$

E. Online Link Prediction Model

Finally, we describe the online link prediction model given by Algorithm 1. The entire graph is modeled by an exchangeable array (see Section III-A). U and $\mathbf{v}_k = [U_i, U_j]$ refer to the nodes and edges of a network graph, respectively. Since the graph may be very large, we partition the whole graph into sub-blocks to allow for incremental processing. Every sub-block maintains a local U' and $\mathbf{v}'$, which are updated by Eqs. (13), (15), and (18) independently. After all the sub-blocks have been updated, the global U and $\mathbf{v}$ are updated by averaging all the local copies of sub-blocks. Then, the global U is used for the prediction.

Algorithm 1 Online Link Prediction.

Input: A social network $G = (U, \mathbf{v})$
Output: $\Theta = \{\langle \mathbf{x} \rangle, \langle \mathbf{z} \rangle, \mathbf{v}\}$

1: Initialize all parameters
2: // Start training
3: Break G into sub-blocks $\{G_1, G_2, .., G_M\}$ with divide-and-conquer strategy
4: Initialize local U' and $\mathbf{v}'$ for each sub-block
5: **for** each sub-block $G_i \in \{G_1, G_2, .., G_M\}$ **do**
6: Update local Θ_i according to Eqs. (13), (15), and (18)
7: Update global U and $\mathbf{v}$ by averaging all local U' and $\mathbf{v}'$

IV. Random Function Models with Network Formation Games

In this section, we first introduce Network Formation Games (NFGs) and then describe the details of the unified model.

A. Network Formation Games

NFGs are methods to describe the factors that affect the network formation from the perspective of game theory, motivated by the idea of capturing the trade-off between benefits and costs. People usually benefit from building relationships with others, but maintaining a relationship also costs time and energy. As people's time and energy are limited, they may seek to maximize their benefit with limited resources. Different definitions of benefits and costs lead to different network formation processes. In our experiments, we evaluate our method with two models: the connections model and the co-author model [6].

The Connections Model (Core–Periphery). In this model, users benefit from both direct and indirect connections, but only pay for their direct connections. Thus modeled networks exhibit a core–periphery structure, which categorizes the nodes as belonging to either the core or periphery. Formally,

$$u_i(G) = \sum_{j \neq i; j \in N_i^2(G)} b(d_{ij}(G)) - \sum_{j \in N_i(G)} c_{ij} \tag{20}$$

$$Uti(G) = \sum_{i=1}^{n} u_i(G) \tag{21}$$

The utility of a user i in the network G is represented by $u_i(G)$ and the overall utility of the network is $Uti(G)$. $N_i(G)$ is the neighbor set of user i. $N_i^2(G)$ refers to the users that can be reached from user i within 2 hops. $d_{ij}(G)$ is the distance between users i and j, which is 1 or 2. $b(d_{ij}(G))$ represents the benefit that user i yields from user j, as a function of the distance between them. c_{ij} is the cost of user i maintaining the relationship with user j. Based on this, the benefit and cost functions are defined as:

$$b(d_{ij}(G)) = \delta^{d_{ij}(G)} \tag{22}$$

$$c_{ij} = c_i = \frac{1}{\mid N_i(G) \mid} \tag{23}$$

Assuming that we rank the users by the cost in ascending order, without loss of generality, cost c_{ij} can be renamed to c_n such that $c_1 < c_2 < \cdots < c_n$. We can then find a boundary user k as the first user satisfying the condition given by Eq. (24). The users whose cost is smaller than c_k are considered *core users*, and others are *periphery users*. Since complex networks typically follow a power law distribution, we choose a specific kind of power law effect, the Pareto principle (a.k.a. the 80-20 rule or the law of the vital few), to determine the user k. Based on this, the 20% of most popular users are most likely to be the core users. Taking k as $0.2 \cdot n$, we find the k^{th} user and its cost c_k in the sorted user sequence. Finally, we get a range of δ. By tuning it, we can filter different numbers of unreliable 0-links, subject to

$$\delta - \delta^2 > 0.5(c_{k-1} + c_k) \tag{24}$$

Consider a pair of users $l = (i, j)$ who are not friends. If we add l to the network G, we define Δ_i as the change in the utility for user i. User i obtains additional benefits in two parts: δ from user j and $\delta^2 \mid N_j(G) \mid$ from user j's friends. At the same time, the only new cost incurred is that of maintainining the relationship with user j. Thus

$$\begin{aligned} \Delta_i &= u_i(G \cup \{l\}) - u_i(G) \\ &= \delta + \delta^2 |N_j(G)| - c_i \end{aligned} \tag{25}$$

The Co-Author Model. The co-author model [6] aims at describing collaborations between researchers. Nodes in the network represent researchers and links refer to collaborations amongst them. The utility function of a given researcher i is defined as Eq. (26). Here n_i is the degree of node i.

$$u_i(G) = \sum_{j \in N_i(G)} \left(\frac{1}{n_i} + \frac{1}{n_j} + \frac{1}{n_i n_j}\right) \tag{26}$$

Similar to the connections model, we can compute the difference in utility Δ_i when researcher i tries to build a link l with researcher j as Eq. (27).

$$\begin{aligned}\Delta_i &= u_i(G \cup \{l\}) - u_i(G) \\ &= \frac{1}{n_j} + \frac{1}{n_j(n_i+1)} - \frac{1}{n_i(n_i+1)} \sum_{k \in N_i(G)} \frac{1}{n_k} \end{aligned} \quad (27)$$

For both models, we can compute the difference in utility for user j in the same way. If Δ_i and Δ_j are both less than 0, we consider l a genuine 0-link. If either of them is greater than 0, we regard l as an unreliable 0-link.

B. Unified Online Link Prediction Model with Game Theory

Algorithm 2 Online Link Prediction with Game Theory.

Input: A social network $G = (U, \mathbf{v})$
Output: $\Theta = \{\langle \mathbf{x} \rangle, \langle \mathbf{z} \rangle, \mathbf{v}\}$

```
Initialize all parameters
if large dataset then
    s = sample from v using sampling strategy
else
    s = v
// Start sampling by game theory
for each user i ∈ U do
    v^+(i, *) ∈ s = all existing links of user i
    v^-(i, *) = ∅
    for each link s_j ∈ s do
        s_j = (i, j) is a 0-link of user i and user j
        Δ_i = u_i(G ∪ {s_j}) − u_i(G)
        Δ_j = u_j(G ∪ {s_j}) − u_j(G)
        if Δ_i ≤ 0 ∧ Δ_j ≤ 0 then
            Add s_j into v^-_*(i, *)
Set training dataset ṽ = v^+(U, *) ∪ v^-(U, *)
Set G̃ = (U, ṽ)
// Start training
Given G̃ call Algorithm 1
```

For large-scale datasets, we additionally rely on sampling strategies to sample 0-links, reducing the size of training data. The sampling strategies considered in our experiments include:

- **Uniform sampling** samples a set of 0-links with the same size uniformly for each node.
- **Weighted sampling** samples a set of 0-links according to the weight of each node. The weight of each node is defined by its degree. Thus, nodes with higher degree get more 0-links.
- **Grid sampling**: The entire graph is first partitioned into multiple segments of equal size. 0-links are sampled randomly with the same size for every segment.

The 0-links sampled via the above-mentioned strategies are subsequently filtered by the NFG to retain only genuine 0-links. Links satisfying the NFG utility function are selected for training ($\tilde{\mathbf{v}}$). A new graph $\tilde{G}$ is obtained by discarding unreliable 0-links. Then, we run the online link prediction algorithm (cf. Algorithm 1) with this new graph $\tilde{G}$. The details of the overall sampling algorithm are given by Algorithm 2.

Let τ denote the size of a sub-block. The algorithm then has a $O(\tau^6)$ time complexity and $O(\tau^4)$ space complexity at each iteration. Note that the training time depends on the size of training links. As game theory can reduce the number of links, the reduced numbers of retained genuine 0-links can also improve the computational efficiency.

V. Experiments

In this section, we evaluate our proposed method on several real-world datasets. The experiments evaluate both the efficiency and the predictive accuracy. Our method uses the isotropic kernel function ($kernel(v_i, v_j) = e^{-\gamma \|v_i - v_j\|^2}$) for all the experiments. The hyperparameter γ is determined via cross-validation.

TABLE I: *Datasets.*

Dataset	Vertices	Edges	Density	References	Scale
Highschool	90	269	6.6×10^{-3}	e.g. [2]	Small
NIPS	234	598	2.2×10^{-3}	e.g. [21]	Small
Protein	230	695	2.6×10^{-3}	e.g. [2]	Small
Ciao	2,342	51,789	9.4×10^{-3}	e.g. [29]	Medium
HEP-PH	12,008	118,521	1.6×10^{-3}	e.g. [15]	Medium
Enron	36,692	183,831	1.4×10^{-4}	e.g. [16]	Big
Slashdot	82,168	948,464	1.4×10^{-4}	e.g. [16]	Big

A. Experimental Setup

Datasets. Table I lists the seven real-world datasets used in our experiments. Among them, the following small datasets are widely used in the literature (e.g., [19]):

- The high school dataset is a social network, in which the edges reflect friendship ties among students. We used the same subset (90 vertices) of the dataset as [19].
- The NIPS dataset consists of all papers and authors from NIPS 1-17. We use the same subset as [19], who selected the 234 authors who had published with the most other people and considered their co-authorship information.
- The protein-protein interaction dataset describes the binding relationship between proteins. While it is not a social network, it exhibits a typical core-periphery structure and was also used by [19].

As medium-sized datasets, we consider:

- [29] crawled the trust networks of the product review site Ciao[1]. Users can add other users to their trust network if they find their reviews interesting and helpful.
- HEP-PH (High Energy Physics – Phenomenolog) collaboration network [15] is extracted from the arXiv e-print repository and covers scientific collaborations between authors whose papers were submitted in the High Energy Physics – Phenomenology category.

As large datasets, we use:

- The Enron email communication network dataset [16] covers all the email communication within a dataset of around 0.5 million emails.

[1]http://www.ciao.co.uk

- The Slashdot dataset [16] consists of links between users who tagged each other as friends or foes on the Slashdot website, which features user-submitted and editor-evaluated news that is primarily technology-oriented.

Competing Methods. We compare our approach against four other methods: probabilistic matrix factorization (PMF) [26], GPLVM [12], SMGB [33], and the random function model (RFM) [19]. Additionally, we also compare our method with game theory (named ORFP-gt) to the method without game theory (called ORFP). Specifically, we refer to our method using the connections model (core-periphery) as ORFP-cp, while ORFP-ca uses the co-author model. Among these competing methods, SMGB and RFM fail on medium-sized datasets and GPLVM also fails on the big ones. Thus, for medium-sized datasets, we only compare PMF, GPLVM, and ORFP-cp. The evaluation on big datasets compares ORFP-cp and PMF employing different sampling strategies.

Parameter Settings. Following [19], the latent dimensions were chosen from $\{1, 2, 3\}$ for small datasets and from $\{3, 5, 7\}$ for medium and big-sized datasets, which require more information. The learning rate was chosen from $\{10^{-5}, 10^{-4}, 10^{-3}\}$. The mini-batch block size τ was 10×10 for small and medium datasets, and is applied by partitioning the graph into non-overlapping 10×10 sub-blocks following a divide-and-conquer strategy, while for large datasets, the block size was 100×100. All parameters are determined through cross-validation. We use 5-fold cross-validation for all methods, i.e., links are predicted for a held-out partition, given 4 others for training.

B. Experimental Results

TABLE II: *AUC of different systems on small datasets.*

Dataset	High school			NIPS			Protein		
Latent dimensions	1	2	3	1	2	3	1	2	3
PMF	0.727	0.760	0.776	0.755	0.796	0.853	0.788	0.806	0.822
GPLVM	0.781	0.818	0.737	0.768	0.849	0.853	0.775	0.882	0.885
SMGB	0.801	0.830	0.810	0.783	0.856	0.855	0.820	0.882	0.851
RFM	0.802	0.831	0.811	0.895	0.902	0.870	0.862	0.880	0.882
ORFP	0.795	0.807	0.801	0.782	0.855	0.879	0.813	0.881	0.882
ORFP-ca	0.801	0.829	0.802	0.847	0.879	0.888	0.816	0.889	0.882
ORFP-cp	0.802	0.830	0.809	0.837	0.869	0.885	0.829	0.890	0.897

TABLE III: *AUC of different systems on medium-sized datasets.*

Dataset	Ciao			HEP-PH		
Latent dimensions	3	5	7	3	5	7
PMF	0.908	0.886	0.886	0.881	0.884	0.888
GPLVM	0.911	0.937	0.925	0.879	0.915	0.903
ORFP-cp	0.927	**0.945**	0.944	0.902	**0.915**	0.910

Prediction Performance. Area Under Curve (AUC) values are exploited to measure the prediction performance. Higher AUC values indicate a better performance. We report the average AUC over 5 runs in Tables II, III, and IV. The results demonstrate that ORFP-gt (game theory either by using the connections model or the co-author model) and ORFP are comparable to RFM on the three small datasets. Furthermore, ORFP-gt significantly outperforms all the alternatives on medium and large datasets with respect to predictive accuracy.

TABLE IV: *AUC of different systems on big datasets for different sampling strategies (u: unweighted, w: weighted, g: grid sampling).*

Dataset	Enron			Slashdot		
Latent dimensions	3	5	7	3	5	7
PMF-u	0.823	0.829	0.837	0.634	0.749	0.757
PMF-w	0.826	0.830	0.836	0.766	0.766	0.808
PMF-g	0.831	0.831	0.833	0.656	0.662	0.784
ORFP-cp-u	0.905	0.897	0.900	0.873	0.899	**0.907**
ORFP-cp-w	0.867	0.883	0.891	0.884	0.878	0.893
ORFP-cp-g	0.889	0.915	**0.931**	0.830	0.874	0.879

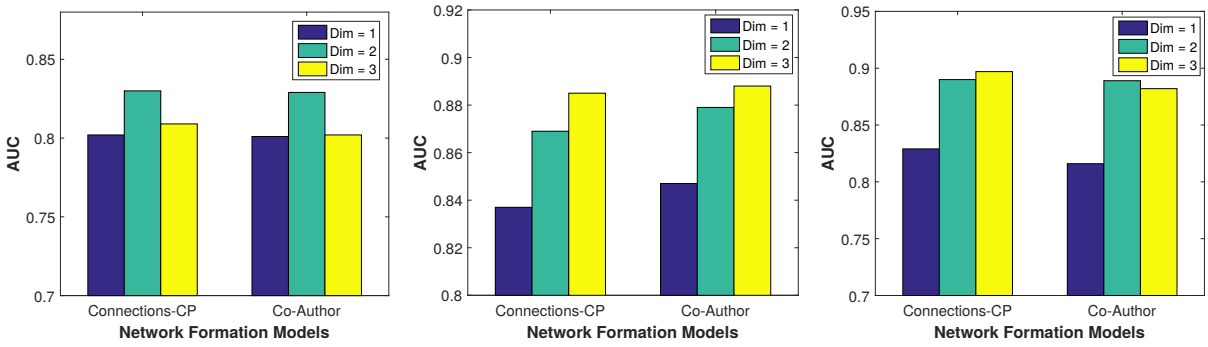
On small datasets, ORFP-gt performs much better than ORFP, though it uses less training data. This shows that game theory can indeed select higher-quality training instances, filtering out unreliable 0-links. This high-quality sampled dataset results in both better accuracy and efficiency. It is notable that on small datasets, most methods perform best when the number of latent dimensions of U is 3. However, for the highschool dataset, the optimal number of latent dimensions of U is 2 for most methods. We conclude that due to the small scale of the highschool dataset, 2 dimensions are sufficient to encode enough information to represent the entire graph.

On medium-sized datasets, ORFP-cp outperforms the other two methods on both medium-sized datasets. It gets the best results on latent dimension 5.

On big datasets, ORFP-cp-g and ORFP-cp-u obtain the best results on the Enron and Slashdot datasets respectively. Additionally, almost every method achieves its best results when the number of latent dimensions of U is set to 7. Hence, for big datasets, the feature space requires a vector of sufficiently high dimensionality to describe the graph.

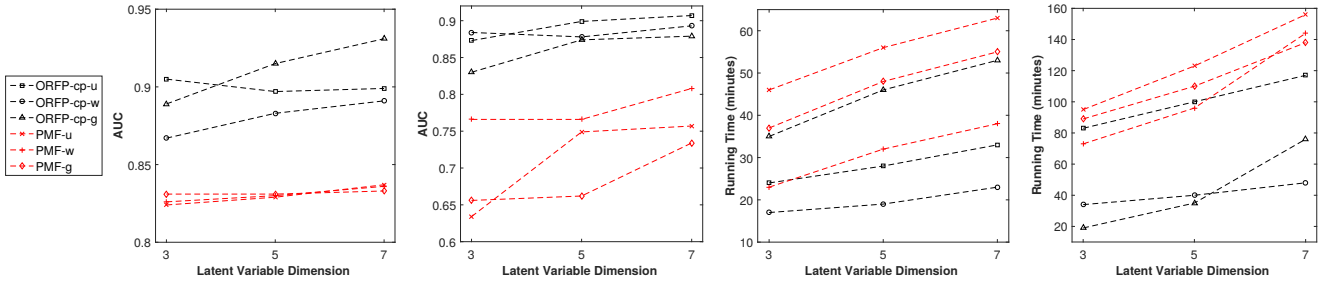
Game Theory Analysis. As we can observe in Table II, ORFP-cp outperforms ORFP-ca and ORFP in most cases. However, the performance of ORFP-ca is better than ORFP-cp on the NIPS dataset. The comparison between ORFP using the co-author and connections models is visualized in Figure 2. NIPS is a collaboration network and ORFP-ca samples using the co-author model, which is designed for modeling collaborations among researchers. Thus, the co-author model is more suitable for NIPS than the core-periphery model. On the other hand, the core-periphery model satisfies the requirements of other datasets, which do not capture collaborations.

Sampling Strategy Analysis. Table IV and Fig. 3 compare PMF with ORFP-cp by using different sampling strategies. PMF-u, PMF-w and PMF-g refer to the results by running PMF with uniform sampling, weighted sampling, and grid sampling respectively. The same naming rule also satisfies ORFP-cp-u, ORFP-cp-w, and ORFP-cp-g. ORFP-cp achieves



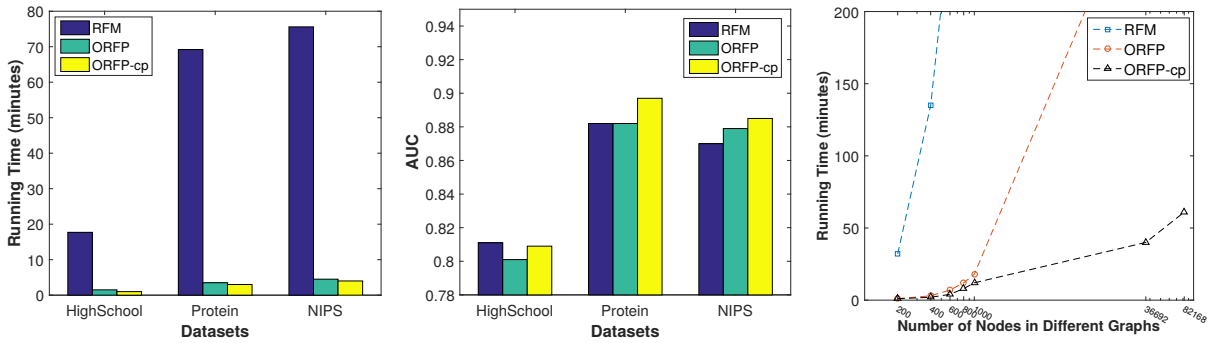
(a) Highschool (b) NIPS (c) Protein

Fig. 2: *Comparison of ORFP using different game theory models.*



(a) Enron (b) Slashdot (c) Enron (d) Slashdot

Fig. 3: *AUC and running time of PMF and ORFP-cp on big datasets.*



(a) Running time (b) AUC (c) Efficiency

Fig. 4: *Runtime and AUC of RFM, ORFP and ORFP-cp on small-scale datasets.*

higher prediction accuracy than PMF. Specifically, ORFP-cp-g fares the best on the Enron dataset and ORFP-cp-u does the best on Slashdot. The weighted sampling strategy performs a little worse than the other two strategies. The reason is that the weighted sampling strategy has opposite effects with NFGs on high degree nodes when sampling 0-links. Weighted sampling retrieves more 0-links for high-degree nodes due to their higher weights. However, NFGs determine that high degree nodes are more likely to build more connections with other nodes, and thus they should have more 1-links but fewer 0-links. On the other hand, low degree nodes get fewer 0-links with weighted sampling. Thus, both high- and low-degree nodes have very few 0-links when employing weighted sampling and NFGs. The imbalanced training data affects the performance of methods using weighted sampling.

Figs. 3(c) and 3(d) show that ORFP-cp runs faster than PMF using the same sampling strategy. With the same amount of training data, PMF takes more time for convergence to a reasonable result. Our model does not need a lot of iterations to make the results converge. Thus it performs better with respect to the consumed time than PMF.

Efficiency Performance. Fig. 4 summarizes the runtime and AUC of RFM, ORFP, and ORFP-cp by setting the latent dimensionality of U as 3. The runtime and AUC of the three

methods on small datasets are given in Figs. 4(a) and 4(b). For further insights, we evaluated the runtime on different subgraphs of the Ciao dataset with varying sizes (200, 400, 600, 800, 1000 nodes), sampling the nodes randomly. As is shown in Fig. 4(c), the runtime of RFM increases significantly as the graph grows in size (cf. Figs. 4(a) and 4(c)). Although the AUC of ORFP-cp is comparable with that of RFM (cf. Fig. 4(b)), ORFP-cp significantly outperforms RFM in terms of efficiency. ORFP-cp also runs faster than ORFP, which demonstrates that game theory can also improve the computational efficiency. In addition, the runtime of ORFP and ORFP-cp increases smoothly as the graph gets larger (node sizes from 200 to 1000). We also evaluate the efficiency on the big Enron and Slashdot datasets. The average runtime of ORFP-cp employing different sampling strategies is several times faster than ORFP without game theory (see node sizes larger than 1000 in Fig. 4(c)).

VI. Conclusion and Outlook

We have proposed ORFP, an online algorithm for latent Gaussian processes on exchangeable graphs, which also exploits game theory by using utility functions to sample genuine zero links. This work improves the scalability of the Random Function Prior Method—a powerful network modeling method proposed by [19], which uses MCMC for inference and thus is only applicable to datasets with hundreds of nodes. We develop an online learning strategy with variational Expectation-Maximization to greatly improve the scalability, as shown in Fig. 4(c). We also observe that in network modeling, most of the non-existing links in the graph are not useful for link prediction. Therefore we introduce network formation games to select informative non-existing links for training. As shown in Fig. 4(b), this can help to further reduce the computational complexity and improve the prediction performance simultaneously. The experimental results on seven real network datasets demonstrate the advantages of ORFP in both predictive performance and computational efficiency over the existing approaches. We plan to develop a distributed learning algorithm to further scale up our model to even larger dataset.

Acknowledgement

This paper was in part supported by Grants from the Natural Science Foundation of China (No. 61503217, 61572111), the National High Technology Research and Development Program of China (863 Program) (No. 2015AA015408), a 985 Project of UESTC (No. A1098531023601041), a Fundamental Research Fund for the Central Universities of China (No. ZYGX2014J058), and a scholarship from China Scholarship Council (CSC) (No. 201606220187).

References

[1] Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. J. Mach. Learn. Res. **9**, 1981–2014 (2008)
[2] Hoff, P.: Modeling homophily and stochastic equivalence in symmetric relational data. In: NIPS, pp. 657–664 (2007)
[3] Hoff, P.: Modeling homophily and stochastic equivalence in symmetric relational data. In: Advances in Neural Information Processing Systems, pp. 657–664 (2008)
[4] Huang, H., Tang, J., Liu, L., Luo, J., Fu, X.: Triadic closure pattern analysis and prediction in social networks. TKDE **27**(12), 3374–3389 (2015)
[5] Jackson, M.O.: A survey of network formation models: stability and efficiency. Group Formation in Economics: Networks, Clubs, and Coalitions pp. 11–49 (2005)
[6] Jackson, M.O., Wolinsky, A.: A strategic model of social and economic networks. Journal of economic theory **71**(1), 44–74 (1996)
[7] Jackson, M.O., Zenou, Y.: Games on networks. Handbook of game theory **4** (2014)
[8] Kallenberg, O.: Symmetries on random arrays and set-indexed processes. Journal of Theoretical Probability **5**(4), 727–765 (1992)
[9] Kemp, C., Griffiths, T.L., Tenenbaum, J.B.: Discovering latent classes in relational data. Tech. Rep. AI Memo 2004-019, MIT (2004)
[10] Kemp, C., Tenenbaum, J.B., Griffiths, T.L., Yamada, T., Ueda, N.: Learning systems of concepts with an infinite relational model. In: Proceedings of AAAI 2006, pp. 381–388 (2006)
[11] Lawrence, N.D.: Gaussian process latent variable models for visualisation of high dimensional data. pp. 329–336. of (2004)
[12] Lawrence, N.D., Urtasun, R.: Non-linear matrix factorization with gaussian processes. In: ICML, pp. 601–608 (2009)
[13] Lee, C., Pham, M., Kim, N., Jeong, M.K., Lin, D.K., Chavalitwongse, W.A.: A novel link prediction approach for scale-free networks. In: WWW, pp. 1333–1338 (2014)
[14] Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: WWW. ACM (2010)
[15] Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: Densification and shrinking diameters. ACM Trans. Knowl. Discov. Data **1**(1) (2007)
[16] Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Internet Mathematics **6**(1), 29–123 (2009)
[17] Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American society for information science and technology **58**(7), 1019–1031 (2007)
[18] Lichtenwalter, R.N., Lussier, J.T., Chawla, N.V.: New perspectives and methods in link prediction. In: SIGKDD, pp. 243–252. ACM (2010)
[19] Lloyd, J., Orbanz, P., Ghahramani, Z., Roy, D.M.: Random function priors for exchangeable arrays with applications to graphs and relational data. In: NIPS, pp. 1007–1015 (2012)
[20] Lovász, L., Szegedy, B.: Limits of dense graph sequences. Journal of Combinatorial Theory, Series B **96**(6), 933–957 (2006)
[21] Miller, K.T., Griffiths, T.L., Jordan, M.I.: Nonparametric latent feature models for link prediction. In: NIPS, pp. 1276–1284 (2009)
[22] Orbanz, P., Roy, D.M.: Bayesian models of graphs, arrays and other exchangeable random structures. IEEE Trans. Pattern Anal. Mach. Intell. **37**(2), 437–461 (2015)
[23] Palla, K., Knowles, D.A., Ghahramani, Z.: An infinite latent attribute model for network data. In: ICML (2012)
[24] Pan, R., Zhou, Y., Cao, B., Liu, N.N., Lukose, R., Scholz, M., Yang, Q.: One-class collaborative filtering. In: ICDM, pp. 502–511. IEEE (2008)
[25] Rényi, A., Erdős, P.: On random graphs. Publicationes Mathematicae **6**(290-297), 5 (1959)
[26] Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: NIPS, pp. 1257–1264 (2007)
[27] Schmidt, M.: Graphical model structure learning with l1 -regularization. Ph.D. thesis, University of British Columbia (2010)
[28] Tang, J., Gao, H., Hu, X., Liu, H.: Exploiting homophily effect for trust prediction. In: WSDM, pp. 53–62. ACM (2013)
[29] Tang, J., Gao, H., Liu, H.: mtrust: Discerning multi-faceted trust in a connected world. In: WSDM, pp. 93–102. ACM (2012)
[30] Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. nature **393**(6684), 440–442 (1998)
[31] Xu, Z., Yan, F., Qi, Y.A.: Sparse matrix-variate t Process blockmodels. In: Proceedings of AAAI 2011 (2011)
[32] Xu, Z., Yan, F., Qi, Y.A.: Bayesian nonparametric models for multiway data analysis. IEEE Trans. Pattern Anal. Mach. Intell. **37**(2) (2015)
[33] Yan, F., Xu, Z., et al.: Sparse matrix-variate gaussian process blockmodels for network modeling. arXiv preprint arXiv:1202.3769 (2012)
[34] Zhang, J., Fang, Z., Chen, W., Tang, J.: Diffusion of following links in microblogging networks (2015)
[35] Zhao, T., Zhao, H.V., King, I.: Exploiting game theoretic analysis for link recommendation in social networks. In: CIKM. ACM (2015)