

RESEARCH

Open Access



A multi-armed bandit approach for exploring partially observed networks

Kaushalya Madhawa*  and Tsuyoshi Murata

*Correspondence:

kaushalya@net.c.titech.ac.jp

Department of Computer Science,
Tokyo Institute of Technology,
W8-59, 2-12-1, Ookayama,
Meguro-ku, 152-8552 Tokyo, Japan

Abstract

Background: real-world networks such as social and communication networks are too large to be observed entirely. Such networks are often *partially observed* such that network size, network topology, and nodes of the original network are unknown. Analysis on partially observed data may lead to incorrect conclusions.

Methods: We assume that we are given an incomplete snapshot of a large network and additional nodes can be discovered by querying nodes in the currently observed network. The goal of this problem is to maximize the number of observed nodes within a given query budget. Querying which set of nodes maximizes the size of the observed network? We formulate this problem as an exploration-exploitation problem and propose a novel nonparametric multi-armed bandit (MAB) algorithm for identifying which nodes to be queried.

Results: Our proposed nonparametric multi-armed bandit algorithm outperforms existing state-of-the-art algorithms by discovering over 40% more nodes in synthetic and real-world networks. Moreover, we provide theoretical guarantee that the proposed algorithm has sublinear regret.

Conclusions: Our results demonstrate that multi-armed bandit based algorithms are well suited for exploring partially observed networks compared to heuristic based algorithms.

Keywords: Network exploration, Network search, Multi-armed bandits, Incomplete networks

Introduction

Interactions among different entities in many real-world complex systems are often represented by networks, where the entities are represented by nodes and the interactions among them are represented as links between entities. For example, the information contained in online social networks proved to be valuable in advertising applications such as finding influential users to targeted marketing. Usually, data acquisition is done using Application Programming Interfaces (APIs) offered by respective social networking services. Using these APIs is often time consuming and the number of nodes (e.g., profiles) that can be queried within a given time is restricted. A poorly constructed incomplete network will lead to inaccurate findings. This highlights the importance of acquiring more information as possible using a limited number of queries.

Here, we provide an overview of *Adaptive Graph Exploration* problem¹. We formally define it in “[Proposed bandit based probing method](#)” section. Suppose we are given a

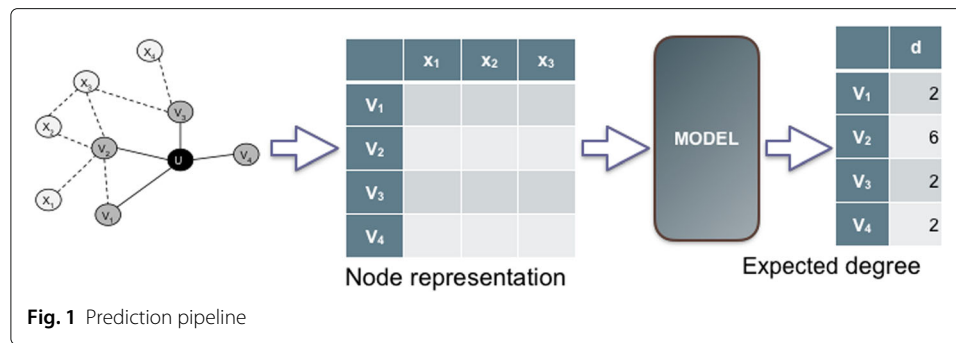
partially observed network. For instance, a sample of a social network collected by a researcher. Since we do not know how this sample is obtained, only way to enhance this sample is by acquiring data belonging to the unseen portion of the network. We use the term *probing* to refer to *querying* a node to retrieve information about it and its neighborhood. As an example, probing a node of a social network corresponds to querying a social network API to obtain information about a user profile and its friends (or followers). Continuing this process for a several rounds introduce new user profiles (nodes) and their friends (neighboring nodes). If we are allowed to continue probing infinitely, we may be able collect information about all the users. Still, when we are done with it, there is a high chance that new users might have joined the network and new friendships have been formed. In reality, we are restricted by constraints enforced by such social network APIs. For example, Twitter limits most of its API requests to a maximum of 15 requests within a 15 min time window². We introduce this constraint as a *probing budget*, the maximum number of times the network is allowed to be probed. Thus, our objective is to enhance the observed graph as much as possible within this *probing budget*.

A straight-forward approach to reduce the incompleteness of a partially observed network is to assume that the network has been generated by a certain network model and use this model to predict the properties of the unobserved portion of the network. For example, Kim and Leskovec (2011) fit a Kronecker model to the observed part of the network and use this fitted model to predict unseen parts of the network. However, this is always not practical for real-world networks as such methods require more structural information (e.g, number of nodes) about the original network. Another approach is to acquire more information of the network by progressively querying the observed network as we name probing in this paper. Existing heuristic algorithms such as maximum observed degree (MOD) probing and maxreach (Soundarajan et al. 2016) require the sample to be obtained in a certain way (e.g., uniform edge sampling). In "Experiments" section, we show that existing probing algorithms can not be generalized for incomplete networks obtained by different sampling techniques. Furthermore, many real-world networks consist of communities, densely connected regions of nodes. With empirical results, we show that heuristic probing algorithms get stuck inside communities, making them worse than probing a node in random.

Contributions

A high level overview of the proposed adaptive probing algorithm is illustrated in Fig. 1. The probing pipeline consists of two major steps, obtaining a feature representation of the observed network and a model which predicts the reward a node will reveal (e.g., the true degree of that node) based on its feature vector. The key assumption of using a learning model is that nodes with similar features in the observed network will result in similar rewards. Our choice of graph features is motivated by previous work on inferring structural role (Henderson et al. 2012) and social status (Zhao et al. 2013) of nodes in social networks.

One property which makes the estimation of rewards different from a normal prediction problem is that our training data is accumulated over the process of probing. A greedy strategy of probing nodes with similar features all the time may result in sub-optimal results. This situation is known in reinforcement learning literature as *exploration-exploitation* trade-off. Multi-armed bandits (Robbins 1952) is a generic way



to approach real-world exploitation-exploration problems. In this context, exploitation corresponds to selecting the node which has the largest expected reward and exploration corresponds to selecting some other node for probing. In this paper, we express adaptive graph exploration problem as a multi-armed bandit problem in a non-stationary environment.

In this manuscript, we extend the approach proposed in Madhawa and Murata (2018). Our contributions are listed below. We mark the contributions which are new additions to the work mentioned in the conference version (Madhawa and Murata 2018) with a *.

- 1 A generic approach for enhancing partially observed networks which does not require any prior knowledge about the network.
- 2 A novel non-parametric upper confidence bound (UCB) algorithm (*i*KNN-UCB) to solve the multi-armed bandit problem (MAB) when the arms are represented in a vector space³.
- 3 We provide a proof that the regret of the proposed bandit algorithm is sublinear.*
- 4 Using *i*KNN-UCB algorithm on synthetic networks and real-world networks from different domains, we demonstrate that our proposed method performs significantly better than existing methods⁴.
- 5 With experiments on different network models, we demonstrate that certain partially observed networks correspond to non-stationary reward distributions.*

Organization

The rest of this manuscript is structured as follows. In “[Background](#)” section, we provide an extensive review of related work. “[Proposed bandit based probing method](#)” section starts with the problem definition and describes our approach in detail. “[Experiments](#)” section explains the experimental setup and the data sets being used. Then, in “[Results](#)” section we present empirical evaluations of our bandit algorithm using real-world networks as well as synthetic networks. Finally, we conclude the paper with “[Conclusions](#)” providing a brief discussion of the proposed bandit approach and a few promising directions as future work.

Background

Network crawling and sampling

Network sampling methods pose as a potential approach to solve the problem introduced in the above section. However, common sampling techniques such as uniform node sampling and uniform edge sampling are not suitable since uniform sampling depends on

access to the space of all available nodes (Ahmed et al. 2014). It is not practical to assume that we can know the number of nodes or edges of a partially observed network.

In contrast, the objective of our problem is improving a given incomplete network and we have no knowledge of how the sample is being obtained. Particularly, snowball sampling (Lee et al. 2006) and random walk based sampling algorithms (Cooper et al. 2016) can be used when the information about the complete network is not accessible. However, such algorithms suffer from the same drawbacks as of heuristic algorithms; they do not adapt as the observed information updates. As another related problem, link prediction (Liben-Nowell and Kleinberg 2007) can be used to predict missing links on a network, but not capable of predicting missing regions of nodes. The observed sample can be further enhanced by iteratively querying observed nodes and adding their neighboring nodes to the sample. Avrachenkov et al. (2014) propose Maximum Expected Uncovered Degree (MEUD), a greedy algorithm for selecting which node to be probed next. However, this algorithm requires the degree distribution of the original network to be known. When this requirement is not fulfilled, it reduces to Maximum Observed Degree (MOD) algorithm which greedily chooses the node with the largest observed degree. We use MOD as a baseline algorithm in our experiments and show that our proposed algorithm significantly outperforms MOD in synthetic and real-world networks.

Active search

Active search on graphs (Wang et al. 2013; Bilgic et al. 2010) is another related problem with the objective of finding as much *target nodes* as possible possessing a given property. Most of the previous work relating to this problem assume that the complete graph is observable and any node can be queried to find its label (Ma et al. 2015). If only an incomplete view is available, an approach relying only on the observed information may not obtain the best possible reward. In addition to *exploitation* of the best option according to available information, *exploration* of other possible options is performed to achieve better rewards. A common approach to finding a balance between exploitation vs exploration trade-off is formulating it as a multi-armed bandit (MAB) problem (Mahajan and Teneketzis 2008). SN-UCB1 (Bnaya et al. 2013) and NETEXP (Singla et al. 2015) are such MAB based active search algorithms proposed for partially observed networks. NETEXP assumes that probing a node reveals its 2-hop neighborhood, which is not true for real-world social networks. If the observability is restricted to 1-hop neighborhood of nodes, this algorithm reduces to random neighbor probing. SN-UCB1 does not provide a significant improvement over the existing heuristic methods. Recently, Soundarajan et al. (2017) proposed ϵ -WGX, a multi-armed bandit approach to solve Active Edge Probing (AEP) problem in incomplete networks. Though AEP looks similar to the problem discussed in this paper, it is fundamentally different from ours as a node can be probed multiple times and only one neighboring edge is revealed in each probe. Hence, this problem can be considered as a restricted version of the problem we are dealing with in this paper.

Multi-armed bandits

Multi-armed bandits (MAB) (Robbins 1952) is a generic framework used to systematically define exploitation vs exploration trade-off. The classic k-armed bandit problem is modeled after a gambler trying to maximize the profit by choosing which slot machines (known as bandits) to play. Playing a bandit results in a reward which is assumed to be

sampled from a probability distribution specific to that bandit. In a multi-armed problem with a discrete set of available actions, choosing an action corresponds to playing an arm in a multi-armed bandit problem. A variety of bandit algorithms are being used to solve a multitude of real-world optimization problems such as recommender systems (Li et al. 2010) and display advertising (Lu et al. 2010). Out of all existing approaches to MAB problem, upper confidence bound (UCB) (Auer 2002) methods possess the best theoretical guarantee in maximizing the reward. UCB algorithms are based on the principle of “optimism in the face of uncertainty”; actions are chosen based on optimistic guesses of how much reward choosing a particular action may bring in. If choosing that action results in a reward which is less than expected, then the confidence placed on that action is decreased.

Algorithms for classic MAB problems decides which action to play only based on the distribution of rewards observed by choosing that action. Since such algorithms do not use any contextual information into consideration, they are known as *context-free* algorithms. However, in real-world optimization problems such as movie recommendation, an action can be represented by its features. For example, in a movie recommendation problem, a movie has a multitude of features(e.g., genre, year, etc.). A variant of MAB problems, contextual bandits (Li et al. 2010) leverages the features describing an action, known as the context of an action. In addition to the reward, contextual bandits observe the context as a feature vector of each action (bandit). The context vectors are used to calculate the expected reward of each action and the action with the largest expected reward is chosen. As an example, LinUCB (Li et al. 2010) models the expected reward as a linear regression on context vectors.

Proposed bandit based probing method

We start this section with the formal definition of the problem. Then we describe the main components of this work and the multi-armed bandit algorithm in detail.

Problem definition

Suppose there is a large unweighted undirected graph G which cannot be observed fully, instead only a partially observed network G' is available. We denote the initial incomplete network as G'_0 (at time=0). Our goal is to grow this network by probing any of the observed nodes at each time step. Using this notation we denote the observed network at time t as G'_t . Table 1 lists the notation that we will be using in this section.

Definition 1 *Probing a node reveals all links incident to it and the identity of its neighboring nodes.*

The number of times we are allowed to probe the network is constrained by the **probing budget** ($T \in \mathbb{Z}^+$).

Table 1 Table of notations

Symbol	Definition
$G(V, E)$	Original network
$G'_t (V'_t, E'_t)$	Observed network at time t
K_t	Set of candidate nodes at time t
T	Probing budget

Definition 2 At time t , a node in the original network G can belong to any of the following three sets.

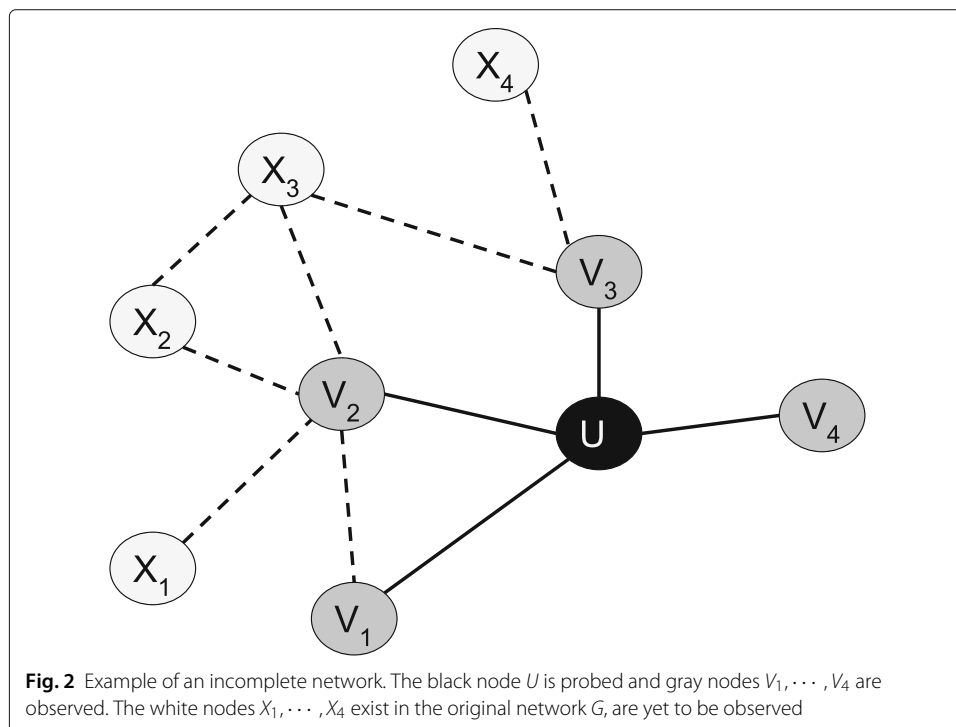
- 1 **unobserved**: existence of these nodes is not visible to the algorithm.
- 2 **observed**: these nodes exist in both G and G_t' , but has not being probed.
- 3 **probed**: the algorithm knows about these nodes and their neighboring nodes.

Figure 2 illustrates an example of an incomplete network. We use bold lines to denote observed links and dash lines to denote unobserved links at the given moment. Even though nodes V_1 and V_2 are observed when node U is probed, $[V_1, V_2]$ link is not observed because neither nodes are probed.

We consider a node which has been probed by the algorithm as an observed node as well. Hence, all the nodes in the graph G_t' are referred to as observed nodes. Any observed node which is not probed is considered as a candidate for probing. Hence, we refer to such nodes as *candidate nodes*. In the beginning, all the nodes in the given sample are candidate nodes. Probing a candidate node reveals a *reward* (e.g., true degree of a node). Our goal is iteratively selecting \mathbf{b} candidate nodes that maximize the cumulative reward (i.e., number of observed nodes).

Calculation of expected reward of a candidate node

Instead of using a heuristic metric to choose a candidate node for probing in each time step, we treat this problem as a learning problem. Similar to an active exploration algorithm, our proposed solution consists of three high-level steps (Pfeiffer III et al. 2014): probing, learning, and prediction. *Probing* a node results in additional information about the observed network. Information about the currently observed network is



leveraged to *learn* a predictive model which *predicts* the expected reward of a given candidate node in future. Our approach assumes that candidate nodes with similar structural neighborhoods will result in similar rewards.

Suppose that the feature vector of a candidate node j at time t is $x_{j,t} \in \mathbb{R}^d$. The learner probes node j at time t and observes the following reward:

$$r_{j,t} = f(x_{j,t}) + \zeta_t,$$

where $f : \mathcal{X} \rightarrow \mathbb{R}$ gives the expected reward of a given node and ζ_t is sub-gaussian white noise. Here, f can be any function which can compute the expected reward of a node given its features (e.g., linear regression).

Assumption 1 (λ -Hölder continuity of f): *There exists constants $C_H > 0$ and $\lambda > 0$ such that $|f(x) - f(x')| \leq C_H \cdot \mathcal{D}(x, x')^\lambda$ for all $x, x' \in \mathcal{X}$. \mathcal{D} is a metric which defines the "distance" between two vectors x and x' .*

Assumption 1 expresses that the difference between the value of regression function f on two points x and x' depends on the distance between the two points $\mathcal{D}(x, x')$. In our problem setting, if two nodes are close with respect to the distance measure \mathcal{D} , their rewards are assumed to be similar. This is a standard smoothness condition used in regression (Chen et al. 2018; Jiang 2019). Lipschitz continuity is a special case of Hölder continuity when $\lambda = 1$. In the following section, we provide a detailed description on how we formulate this problem as a multi-armed bandit problem.

Bandit algorithm

Problem setting

In the classical multi-armed bandit problem, an agent selects one of the K arms (or actions) at each time step and observes a reward depending on the chosen action. The goal of the agent is to play a sequence of actions which maximizes the cumulative reward it receives within a given number of time steps. Classical k -armed bandit problem comes with the following assumptions:

- 1 The set of arms K does not change over time.
- 2 Each arm is independent.
- 3 The rewards are drawn randomly from a probability distribution that is specific to each arm.
- 4 The environment is stationary. The reward distribution of arms does not change over time.

Selecting a node from the set of candidate nodes at time step t for probing is similar to pulling an arm in a multi-armed bandit problem. However, our problem does not satisfy the assumptions mentioned above. For example, the set of candidate nodes change as we keep adding new nodes to the partially observed network during the exploration process. More importantly, probing a node for the second time does not reveal any additional information. Hence, playing an arm once again is a waste of time. In addition, it is not safe to assume that the reward distribution would stay stationary over time. We address all of these concerns in our proposed algorithm.

As independent assumption does not hold in our problem setting, it is more suitable to express it as a *structured bandits* problem, in which reward distributions of arms are not independent, but interrelated. In a structured bandit problem, the agent deduces relationships between arms based on some d -dimensional feature vector $x_a \in \mathbb{R}^d$ belonging to an arm a .

KNN-UCB algorithm for structured bandits

Linear bandits(Rusmevichientong and Tsitsiklis 2010; Dani et al. 2008) model, the simplest among such models, assumes that the reward of choosing an arm is linearly dependent on its features. In linear bandits, the expected reward of an arm is calculated as the inner product of its feature vector and a parameter vector θ . However, real-world data often exhibit more complicated relationships than a linear one. Therefore, we choose k -nearest neighbor (k -NN) regression to estimate the expected reward of arms. To introduce exploration into the solution, we extend Guan and Jiang (Guan and Jiang 2018)’s k -armed KNN-UCB algorithm to the structured setting. As explained in *Multi-armed bandits*, upper confidence bound (Auer 2002) (UCB) algorithms incorporate an exploration term by calculating confidence bound for each arm and choose the action corresponding to the largest confidence bound.

We define k -nearest neighbor upper confidence bound (i KNN-UCB) rule as

$$a_t = \arg \max_i \hat{f}(x_i) + \alpha \cdot U_{t,k}(x_i) \tag{1}$$

where $U_{t,k}(x)$ is the uncertainty score of point x and $\alpha > 0$ is a constant determining the amount of exploration. If $\alpha = 0$ the uncertainty score is ignored, and the algorithm becomes a greedy algorithm which performs exploitation all the time.

To address the issue of non-stationarity of the environment, we consider only the most recent observations within a time window of size τ . In this setting, k -NN regression considers only the τ -most recently observed nodes and their rewards in computing the expected reward for a new node. This approach is motivated by sliding window UCB (SW-UCB) algorithm proposed by Garivier and Moulines (Garivier and Moulines 2011). If $\tau \geq T$, then the resultant UCB algorithm is the same as the usual stationary UCB algorithm.

Definition 3 (*k*-nearest neighbor regression (Jiang 2019)). Let the k -NN radius of $x \in \mathcal{X}$ be $r_k(x) = \inf \{r : |B(x, r_k(x) \cap X) \geq k|\}$ where $B(x, r) = \{x \in \mathcal{X} : \mathcal{D}(x, x') \leq r\}$. k -NN set of $x \in \mathcal{X}$ be $\mathcal{N}_k(x) := B(x, r_k(x)) \cap X$. Expected reward of arm i , $\hat{f}(x_i)$ is estimated with weighted k -NN regression as

$$\hat{f}(x_i) = \frac{1}{k} \sum_{x_j \in \mathcal{N}_k(x_i)} \frac{y_j}{\mathcal{D}(x_i, x_j)}, \tag{2}$$

where y_j is the observed reward for x_j and $\mathcal{D}(x_i, x_j)$ is euclidean distance between feature vectors x_i and x_j .

We define $\sigma(x)$ as the average distance to points in the k -neighborhood,

$$\sigma(x_i) = \frac{1}{k} \sum_{x_j \in \mathcal{N}_k(x_i)} \mathcal{D}(x_i, x_j). \tag{3}$$

The uncertainty term $U_{t,k}(x_i)$ is analogous to the term $T_i(t)$ in a finite arm MAB problem, the number of times action i has been chosen by the time t . If $U_{t,k}(x_i)$ is large, the k -neighborhood of node i is dispersed over a larger space. On the other hand, if $\sigma(x_i)$ is small, we have already observed nodes close to node i . Hence, the exploration term weights less observed neighborhoods. Algorithm 1 shows how a given network is being probed using the proposed i KNN-UCB algorithm.

Algorithm 1: i KNN-UCB.

Input : incomplete network $G'_0 = (V'_0, E'_0)$, probing budget $T \in \mathbb{Z}^+$, exploration parameter α , temporal window size τ, k, T_0

Output: A sequence of T nodes to probe

Initialize: *candidate nodes* = V'_0

```

1 for  $t \leftarrow 1$  to  $T$  do
2   if  $t \leq T_0$  then
3     sample  $\mathbf{a}_t$  uniformly from  $\mathcal{A}_t$ 
4   else
5     for  $i$  in candidate nodes do
6       calculate the feature vector  $x_i$ 
7       calculate the estimated reward  $\hat{f}(x_i)$  with Eq. 2
8       calculate exploration term  $\sigma(x_i)$  with Eq. 3
9     find the node  $\mathbf{a}_t$  corresponding to the largest UCB with Eq. 1
10    probe node  $\mathbf{a}_t$  in the original graph  $\mathbf{G}$  and observe the reward  $\mathbf{r}_{t,a_t}$ 
11    Add neighboring nodes  $N_{a_t}$  of node  $a_t$  to the incomplete network
12     $\mathbf{G}'_{t-1} \cdot (\mathbf{G}'_t = \mathbf{G}'_{t-1} \cup N_{a_t})$ 
13    remove node  $\mathbf{a}_t$  from candidate nodes

```

Regret

The objective of a bandit algorithm is to select arms so as to maximize the cumulative reward over time. Minimization of total regret is an equivalent way of expressing maximization of cumulative reward. The regret at iteration t equals to the difference between reward of the “optimal” arm and the reward of a suboptimal arm. In simple terms, regret is the loss incurred by the policy for not playing the optimal arm all the times. In T iterations, we pull arms a_1, a_2, \dots, a_n and we observe rewards $r_{a_1,1}, r_{a_2,2}, \dots, r_{a_n,n}$. We use the following notion of regret

$$\mathcal{R}_T = \sum_{t=1}^T \left[\max_a r_{a,t} - r_{a_t,t} \right].$$

Theorem 1 (*Sublinear regret bound*). *Let $M > 0$ be an arbitrary constant. Suppose that Assumption 1 holds. Then the regret is sublinear with,*

$$\mathcal{R}_T \leq M \cdot T^{(\lambda+d)/(2\lambda+d)}.$$

Proof The regret for bandits in a continuous feature space is

$$\mathcal{R}_T = \sum_{t=1}^T [\sup_{x_i \in \mathcal{X}} f(x_{i,t}) - f(x_{a_t,t})]. \tag{4}$$

Using k -nearest neighbor regression rates when f is λ -Hölder continuous (Assumption 1) and $k = O(t^{2\lambda/(2\lambda+d)})$ (Jiang 2019),

$$\sup_{x \in \mathcal{X}} |f(x) - f_k(x)|_\infty \leq \tilde{O}(t^{-\lambda/(2\lambda+d)}). \tag{5}$$

Using this result in Eq. 4, results in

$$\mathcal{R}_T \leq \sum_{t=1}^T L_t \cdot t^{-\lambda/(2\lambda+d)}, \tag{6}$$

where L_t is used as an asymptotic constant.

Let $L_t \leq M$,

$$\mathcal{R}_T \leq M \sum_{t=1}^T t^{-\lambda/(2\lambda+d)} = M \cdot T^{(\lambda+d)/(2\lambda+d)}. \tag{7}$$

Hence, the regret is sub-linear. □

Experiments

We construct the feature vector x_j of candidate node j as a vector of following features. For each feature, the local neighborhood of node j in the observed graph G'_t is considered. All these features vary over time as new nodes and edges are added to the observed graph during the probing process.

- 1 $d(j)$: observed degree centrality of node j .
- 2 $ad(j)$: average degree centrality of its neighbors.
- 3 $md(j)$: median degree centrality of its neighbors.
- 4 $ap(j)$: the average fraction of probed neighbors found in the neighborhood. This feature takes the maximum value 1, if all the neighbors of node j have been probed.

These features are chosen because their effectiveness is shown in previous work on finding structurally similar nodes (Henderson et al. 2012).

Data

We perform experiments on various synthetic networks as well as eight publicly available real-world data sets of social and information networks (Leskovec and Krevl 2014). The datasets are briefly explained below.

Synthetic networks

Real-world networks exhibit a variety of characteristics: different degree distributions, existence of community structure etc. Before delving into real-world networks, we generate synthetic networks with a varying degree of such characteristics. This makes it easier to understand the performance of our proposed approach in terms of network properties.

We use the following network generation models:

Barabasi-Albert (BA) (Barabási and Albert 1999). The BA model generates scale-free networks with power-law degree distributions. A scale-free network contains a few nodes (called hubs) with unusually high degree compared to other nodes. The BA model uses a

network generation process consisting of growth and preferential attachment. This process selects neighbors to a given node with a probability proportional to their degree. This makes sure that the higher the degree a node has it has a higher probability of having edges with other nodes. This phenomenon is responsible for creating hub nodes.

Lancichinetti-Fortunato-Radicchi (LFR) (Lancichinetti et al. 2008). In addition to power-law degree distributions, real-world social and communication networks possess additional phenomena such as the existence of communities (Fortunato and Castellano 2012). Groups of nodes which are densely connected within the same group compared to nodes belonging to other groups are known as communities. Modularity is a popular metric used to measure the quality of a particular division of a network into constituent communities. Most of the community detection algorithms are based on the principle of modularity maximization (Newman 2004; Blondel et al. 2008). Thus, higher modularity is an indication of the existence of community structure.

BA network model is not capable of generating networks having community structure. We use Lancichinetti–Fortunato–Radicchi (LFR) benchmark to generate networks with community structure. Node degree and community sizes of networks generated by LFR benchmark have power law distributions with different exponents. The mixing parameter μ of LFR model decides the probability of a node linking to a node belonging to another community. Low values of μ will result in dense communities as the chance of having intra-community links ($1 - \mu$) is higher compared to the chance of inter-community links (μ). We create LFR benchmark networks with varying the value of μ in the range [0.1, 0.5] to investigate how our proposed model performs on networks with varying degree of community structure. Mixing parameter and modularity of a network are inversely related.

To make it easier to compare performance across networks generated by different algorithms, we generate all the networks with the same number of nodes ($N = 34,546$), the number of nodes in the HepPh citation network (described in the following section).

Real-world networks

Table 2 gives a summary of the seven real-world network data sets we use. We use real-world networks obtained from various domains as detailed below.

Citation networks. A citation network contains an undirected edge connecting paper i and paper j , if the paper i cites another paper j .

Co-authorship networks. Similarly, in a co-authorship network authors are represented as nodes. Two authors are connected if they have published at least one paper together.

Social networks. A social network consists of users and relationships between them. We use a network data set obtained from the social network Twitter. This network is made

Table 2 Description of real-world network data sets

	HepPh	HepTh	Epinions	Twitter	Stanford	AstroPh	DBLP	Slashdot
Type	Citation	Citation	Web	Social	Web	CA	CA	Web
Nodes	34546	27770	75789	81306	281903	18772	317080	82168
Edges	421578	352807	508837	1768149	2312497	198110	1049866	549202
ACC	0.2848	0.3120	0.1378	0.5653	0.5976	0.6306	0.6324	0.0603

(CA = co-authorship, ACC = Average Clustering Coefficient)

of 1000 ego-networks consisting of 4869 Twitter lists (Leskovec and McAuley 2012). An ego-network is a social circle formed among a user and her friends.

Web networks. These networks represent users and links between them, similar to a social network. However, the networks we consider here, Epinions and Slashdot represent who-trust-whom data of users instead of the relationships or interaction among users. Hence, we categorize them as web networks. In these networks, a user tags another user as trustworthy or not. They are sparse compared to online social networks.

Impact of initial sampling method

To investigate how the sampling method used to acquire the initial sample influence the probing methods, we generate graph samples using two sampling methods. These are the methods we use:

- 1 Random node sampling (RN): At each step we randomly choose one neighbor of a node already in the sample.
- 2 Breadth-first search (BFS): Nodes are added to the sample in the order they are observed.

We induce a subgraph on a sample of nodes obtained by any of the above methods. These subgraphs are used as the initial sample for the adaptive graph exploring problem and probing algorithms are applied to acquire more information on the original network.

Methods

We compare the performance of our algorithm against the following algorithms.

Algorithms that do not use node features

- **Random node (RW).** In this trivial baseline, we select one of the candidate nodes randomly for probing.
- **Maximum observed degree (MOD).** This greedy algorithm chooses the node having the maximum observed degree. MOD is the MEUD algorithm proposed in (Avrachenkov et al. 2014) adapted to one-hop neighborhood visibility.

Algorithms that use node features

- **Lin-UCB.** This applies the UCB algorithm by Dani and Kakade (Dani et al. 2008) assuming that the reward of an arm is linearly dependent on its feature vector.
- **KNN-greedy.** This algorithm chooses the arm corresponding to the largest expected reward calculated by k -NN model.
- **KNN- ϵ -greedy.** This algorithm chooses a random arm with probability ϵ , chooses the arm corresponding to the largest expected reward calculated by k -NN model with probability $(1 - \epsilon)$.
- **i KNN-UCB** This is our proposed algorithm, Algorithm 1. For all algorithms based on k nearest neighbor regression, we limit number of neighbors to 20 ($k=20$).

Results

Analysis on synthetic networks

We probe incomplete BA and LFR networks obtained by RN and BFS sampling for 1000 iterations ($T = 1000$). We perform each experiment 10 times with different initial samples

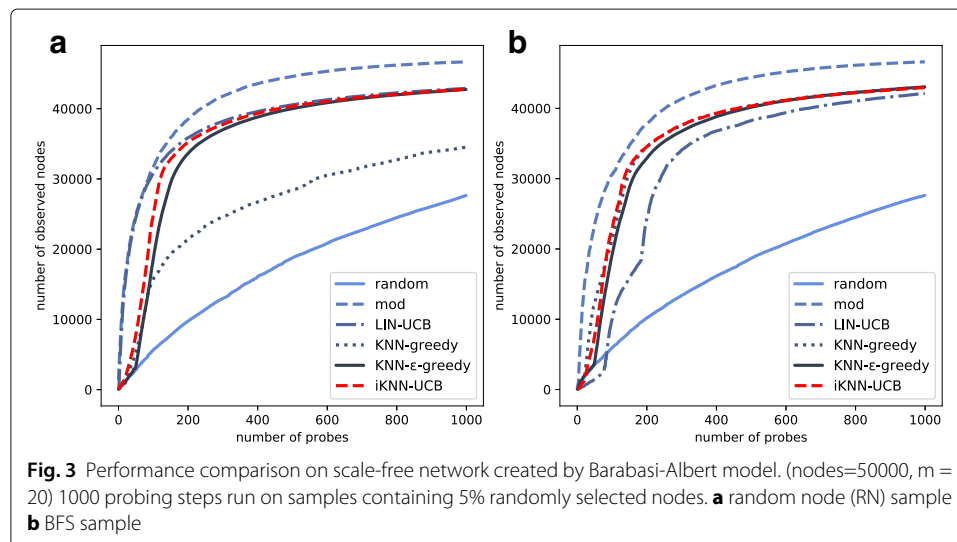
and report the average in this section. The number of nodes observed in the BA network is shown in Fig. 3. For all networks generated by Barabasi-Albert (BA) model, MOD could observe more nodes than the bandit algorithm. This observation confirms Avrachenkov et al. (2014)'s claim that MOD probing can achieve the best connected network cover for networks generated by preferential attachment processes. The recent work by LaRock et al. (LaRock et al. 2018) further generalize this claim as learning and predicting the rewards is not necessary for networks generated by BA model.

To understand how the existence of community structure impacts the probing, we evaluate the performance of all algorithms on synthetic networks generated by different configurations of LFR benchmark model (Lancichinetti et al. 2008). We vary the mixing parameter μ from 0.1 to 0.5 keeping all other parameters of the model constant ($\gamma = 3, \beta = 1.3$, average degree = 25). *i*KNN-UCB significantly outperforms the baseline for networks with a smaller μ . The results are shown in Fig. 4. When the initial sample is obtained by breadth-first walk (BFS), *i*KNN-UCB outperforms all baselines by a significant margin. The gap between *i*KNN-UCB and the baselines is larger when the mixing parameter is small, the network has significant community structure. BFS sampling results in dense network samples. This is evident from the significantly high clustering coefficients of BFS sample networks compared to RN samples of the same size. Probing a few nodes in such a densely connected region is enough to acquire the most information about that region. However, greedy algorithms such as MOD are not capable of learning this reality and keep probing nodes which won't result in high rewards.

The experimental results on synthetic networks suggest that *i*KNN-UCB algorithm can adapt for incomplete networks obtained by different sampling techniques and networks with structural properties such as community structure.

The importance of exploration

The parameter α in the proposed UCB algorithm determines the amount of exploration performed by the algorithm. Therefore, when $\alpha = 0$, the resultant algorithm is equivalent to the greedy algorithm which chooses the node with the largest expected reward. In this section, we perform experiments by varying the value of α and results are shown in



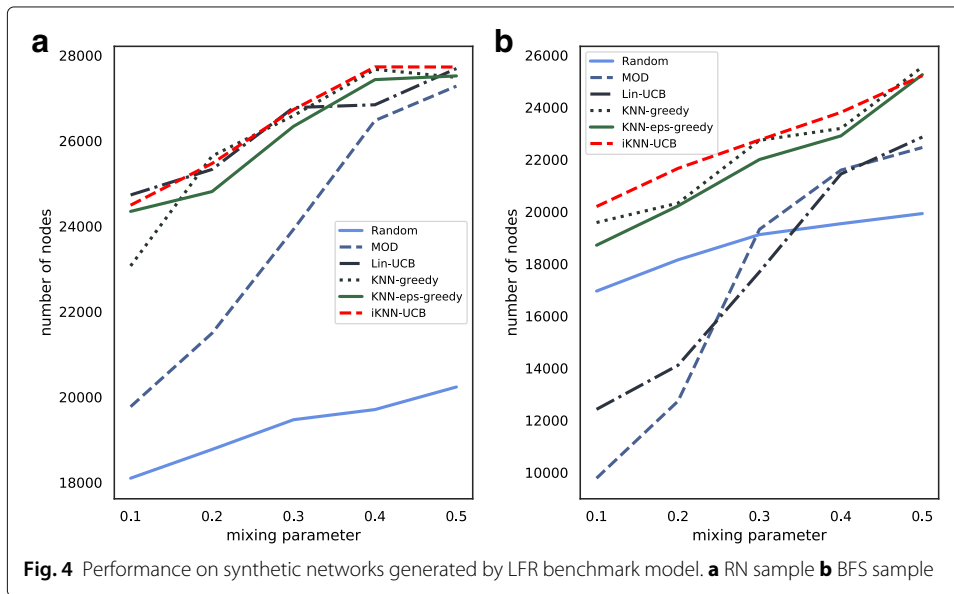
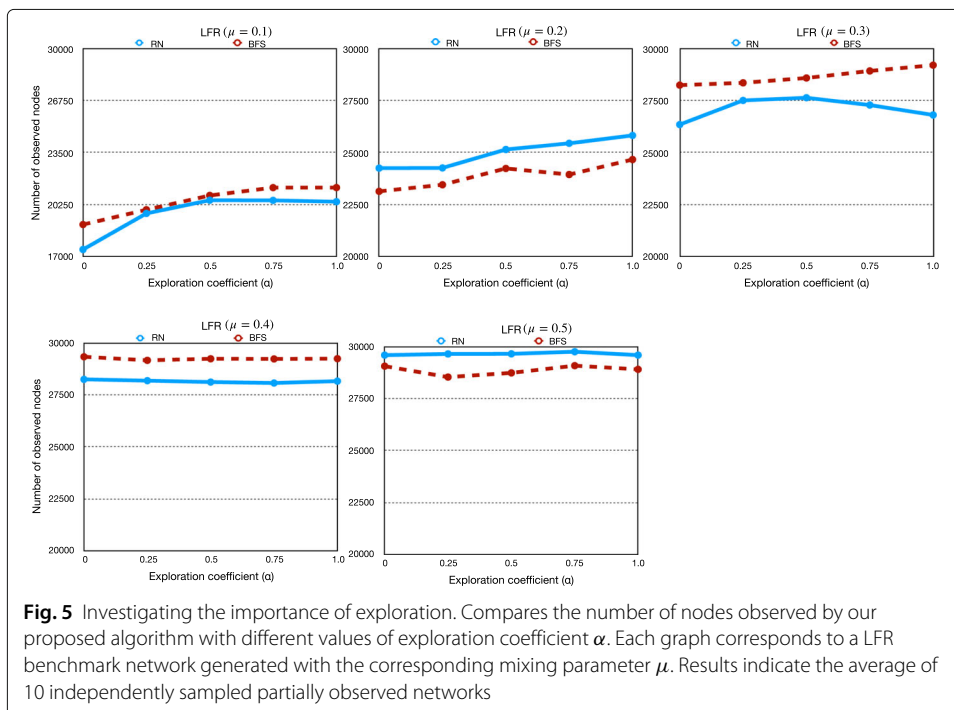


Fig. 5. We observe that more exploration corresponds to better results for networks with stronger community structure (smaller μ). However, the exploration is less important for networks with weaker community structure (larger μ). This is evident in Fig. 4 as well. We perform all the subsequent experiments keeping the exploration coefficient as a constant ($\mu = 1$).



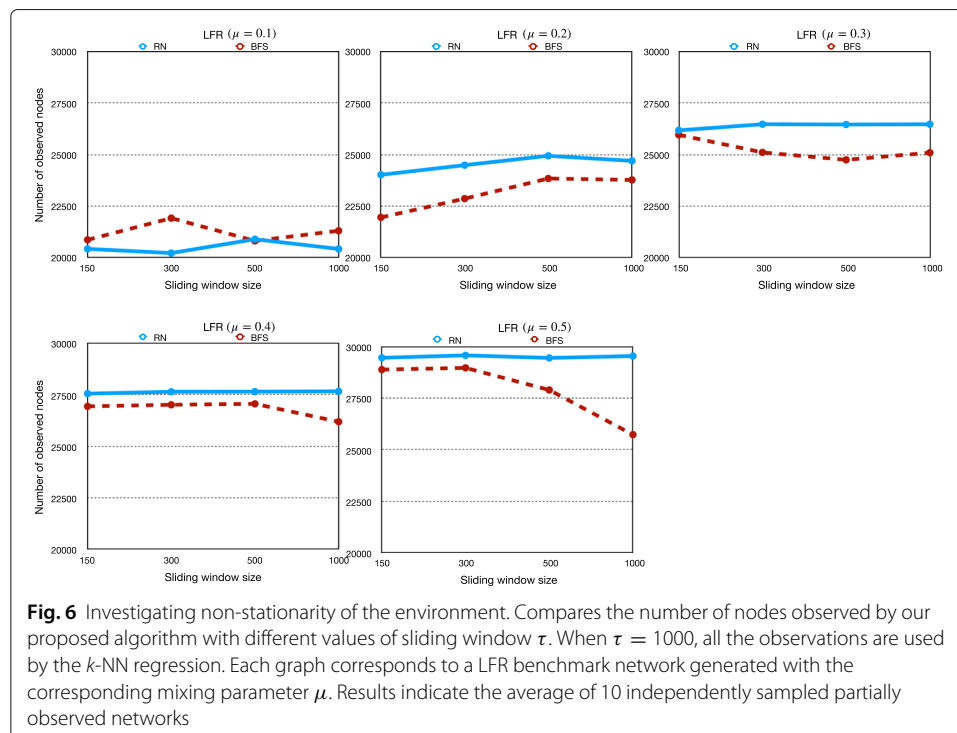
Non-stationarity of the environment

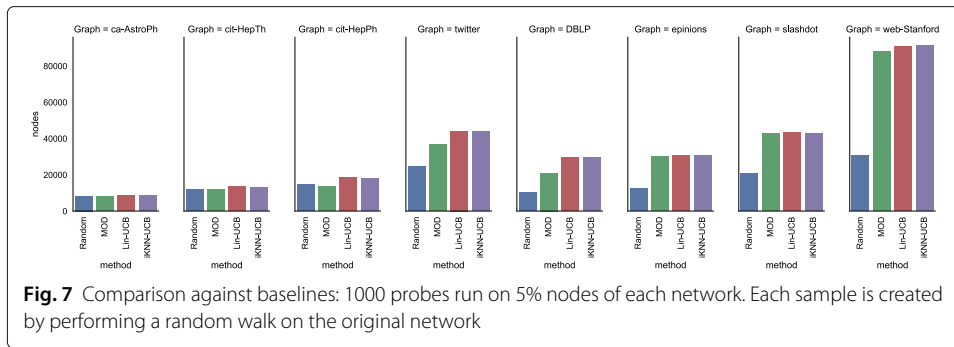
In this section, we investigate the non-stationarity of reward distribution by varying the sliding window size τ . If the reward distribution is stationary, then the variation of performance between different values of τ should be minimum. However in Fig. 6, we observe larger variance among the cumulative reward of LFR benchmark graphs with a larger mixing parameter, especially when sampled by a BFS walk. Larger mixing parameter corresponds to networks with weaker community structures, hence smaller modularity. Based on these observations, for network samples with low modularity and high clustering, it is desirable to run *i*KNN-UCB algorithm with a smaller sliding window; relying only on the most recent observations.

Results on real-world networks

We use eight real-world networks mentioned in Table 2 and generate RN and BFS samples containing 5% nodes of the original network G . Then we perform 1000 probing steps on each graph. We perform each experiment five times initialized with different random seeds and report the average number of additional nodes which were observed in Figs. 7 and 8.

*i*KNN-UCB and Lin-UCB bandit algorithms outperform all baseline methods in networks generated by both RN and BFS sampling. Even though Lin-UCB bandit algorithm observes as much nodes as *i*KNN-UCB for RN samples, its performance is worse for BFS samples. This shows that linear model in Lin-UCB is not capable of learning the relationship between observed node features and the true degree of a node if the sample is constructed by a BFS.





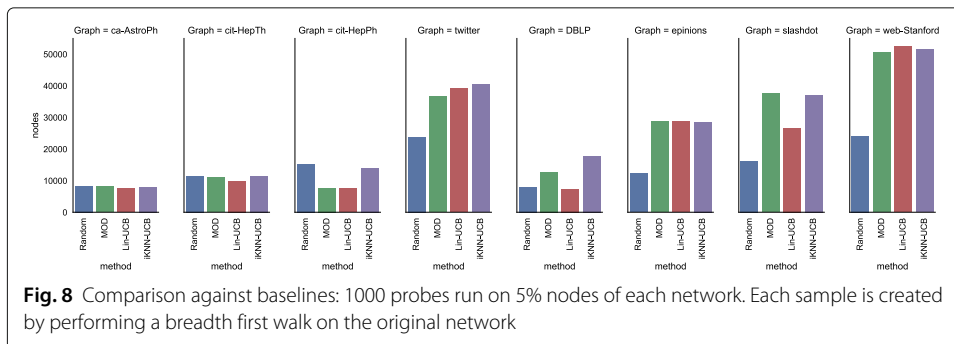
Conclusions

In this paper, we introduced a multi-armed bandit based exploration algorithm for partially observed incomplete networks. We proposed a novel nonparametric multi-armed bandit algorithm *iKNN-UCB* with sublinear regret. Compared to existing solutions for the *Adaptive Graph Exploring* problem, the proposed method does not depend on a specific heuristic. Additionally, *iKNN-UCB* bandit algorithm outperforms the baseline methods irrespective of how the initial incomplete network is obtained. We provided experimental evidence for our approach using synthetic networks and a variety of real-world networks. Using different configurations of LFR benchmark networks, we observed that our algorithm outperforms all other baselines significantly, especially when the network exhibits community structure prominently. Since the reward function is independent of the probing procedure, it is easy to define a new reward function to solve a different graph exploration problem (e.g., finding a particular type of nodes).

In this paper, we assumed that probing a node would reveal all its neighboring nodes. However, in some real-world scenarios, only a certain number of neighbors is revealed (e.g., follower limit in Twitter API ⁵). As future work, we intend to explore how this current approach can be extended for such different settings of the same problem.

Endnotes

- ¹ We use the terms network and graph interchangeably.
- ² <https://developer.twitter.com/en/docs/basics/rate-limits.html>
- ³ source code available at https://bitbucket.org/kau_mad/bandits
- ⁴ source code available at https://bitbucket.org/kau_mad/net_complete
- ⁵ <https://dev.twitter.com/rest/reference/get/followers/ids>



Abbreviations

AEP: Active edge probing; API: Application programming interface; BA: Barabasi-Albert model; BFS: Breadth first search; k -NN: k -nearest neighbor algorithm; LFR: Lancichinetti-Fortunato-Radicchi benchmark model; MAB: Multi-armed bandits; MEUD: Maximum expected uncovered degree; MOD: Maximum observed degree; RN: Random node sampling; RW: Random walk; UCB: Upper confidence bound

Acknowledgments

Not applicable.

Funding

This work was supported by JSPS Grant-in-Aid for Scientific Research(B) (Grant Number 17H01785) and JST CREST (Grant Number JPMJCR1687).

Availability of data and materials

All datasets are publicly available at <http://snap.stanford.edu/data/index.html> (Leskovec and Krevl 2014). Source code used to perform experiments is available here: https://bitbucket.org/kau_mad/net_complete/src/9cbce0522fb48b0403a495c6c8f505bb817f7c6/mab_explorer/?at=master.

Authors' contributions

All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 1 March 2019 Accepted: 12 May 2019

Published online: 31 May 2019

References

- Ahmed NK, Neville J, Kompella R (2014) Network sampling: From static to streaming graphs. *ACM Trans Knowl Discov Data (TKDD)* 8(2):7
- Auer P (2002) Using confidence bounds for exploitation-exploration trade-offs. *J Mach Learn Res* 3(Nov):397–422
- Avrachenkov K, Basu P, Neglia G, Ribeiro B, Towsley D (2014) Pay few, influence most: Online myopic network covering. In: *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*. IEEE, pp 813–818. <https://doi.org/10.1109/INFCOMW.2014.6849335>
- Barabási A-L, Albert R (1999) Emergence of scaling in random networks. *Science* 286(5439):509–512
- Bilgic M, Mihalkova L, Getoor L (2010) Active learning for networked data. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. Omnipress, Haifa, pp 79–86. <http://www.icml2010.org/papers/544.pdf>
- Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. *J Stat Mech Theory Exp* 2008(10):10008
- Bnaya Z, Puzis R, Stern R, Felner A (2013) Social network search as a volatile multi-armed bandit problem. *HUMAN* 2(2):84–98
- Chen GH, Shah D, et al (2018) Explaining the success of nearest neighbor methods in prediction. *Found Trends® Mach Learn* 10(5-6):337–588
- Cooper C, Radzik T, Siantos Y (2016) Fast low-cost estimation of network properties using random walks, Vol. 12. <https://doi.org/10.1080/15427951.2016.1164100>
- Dani V, Hayes TP, Kakade SM (2008) Stochastic linear optimization under bandit feedback. In: *21st Annual Conference on Learning Theory*, pp 0–20. <http://colt2008.cs.helsinki.fi/>. <http://shop.omnipress.com/index.asp?PageAction=VIEWPROD&ProdID=157>
- Fortunato S, Castellano C (2012) Community structure in graphs. *Comput Complex Theory Tech Appl*:490–512. https://doi.org/10.1007/978-1-4614-1800-9_33
- Garivier A, Moulines E (2011) On upper-confidence bound policies for switching bandit problems. In: *International Conference on Algorithmic Learning Theory*. Springer, Berlin, pp 174–188
- Guan MY, Jiang H (2018) Nonparametric stochastic contextual bandits. In: *AAAI Conference on Artificial Intelligence*. <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewPaper/16944>
- Henderson K, Gallagher B, Eliassi-Rad T, Tong H, Basu S, Akoglu L, Koutra D, Faloutsos C, Li L (2012) Rolx: structural role extraction & mining in large graphs. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, New York, pp 1231–1239. <http://doi.acm.org/10.1145/2339530.2339723>. <https://doi.org/10.1145/2339530.2339723>
- Jiang H (2019) Non-Asymptotic Uniform Rates of Consistency for k -NN Regression. In: *AAAI Conference on Artificial Intelligence*. https://aaai.org/Conferences/AAAI-19/wp-content/uploads/2018/11/AAAI-19_Accepted_Papers.pdf
- Kim M, Leskovec J (2011) The network completion problem: Inferring missing nodes and edges in networks. In: *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, pp 47–58. <https://doi.org/10.1137/1.9781611972818.5>. <https://epubs.siam.org/doi/abs/10.1137/1.9781611972818.5>
- Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. *Phys Rev E* 78(4):046110
- Lee SH, Kim P-J, Jeong H (2006) Statistical properties of sampled networks. *Phys Rev E* 73(1):016102
- Leskovec J, Krevl A (2014) SNAP Datasets: Stanford Large Network Dataset Collection. <http://snap.stanford.edu/data>
- Leskovec J, McAuley JJ (2012) Learning to discover social circles in ego networks. In: *Advances in neural information processing systems*. Curran Associates, Inc, pp 539–547. <http://papers.nips.cc/paper/4532-learning-to-discover-social-circles-in-ego-networks.pdf>

- Li L, Chu W, Langford J, Schapire RE (2010) A Contextual-bandit Approach to Personalized News Article Recommendation. In: Proceedings of the 19th International Conference on World Wide Web. WWW '10. ACM, New York. pp 661–670. <http://doi.acm.org/10.1145/1772690.1772758>. <https://doi.org/10.1145/1772690.1772758>
- Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. *J Assoc Inf Sci Technol* 58(7):1019–1031
- Lu T, Pál D, Pál M (2010) Contextual multi-armed bandits. In: Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research. Vol. 9. PMLR, Sardinia. pp 485–492
- Ma Y, Huang T-K, Schneider JG (2015) Active search and bandits on graphs using sigma-optimality. In: UAI. AUAI Press. pp 542–551
- Madhawa K, Murata T (2018) Exploring partially observed networks with nonparametric bandits. In: International Conference on Complex Networks and Their Applications. Springer International Publishing. pp 158–168. https://doi.org/10.1007/978-3-030-05414-4_13
- Mahajan A, Teneketzis D (2008) Multi-armed bandit problems. *Found Appl Sensor Manag*:121–151. https://doi.org/10.1007/978-0-387-49819-5_6
- Newman ME (2004) Fast algorithm for detecting community structure in networks. *Phys Rev E* 69(6):066133
- Pfeiffer III JJ, Neville J, Bennett PN (2014) Active exploration in networks: Using probabilistic relationships for learning and inference. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM, New York. pp 639–648. <http://doi.acm.org/10.1145/2661829.2662072>. <https://doi.org/10.1145/2661829.2662072>
- Robbins H (1952) Some aspects of the sequential design of experiments. *Bull Am Math Soc* 58(5):527–535
- Rusmevichientong P, Tsitsiklis JN (2010) Linearly parameterized bandits. *Math Oper Res* 35(2):395–411
- Singla A, Horvitz E, Kohli P, White R, Krause A (2015) Information gathering in networks via active exploration. In: IJCAI. AAAI Press. pp 981–988
- Soundarajan S, Eliassi-Rad T, Gallagher B, Pinar A (2016) Maxreach: Reducing network incompleteness through node probes. In: Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference On. IEEE. pp 152–157. <https://doi.org/10.1109/ASONAM.2016.7752227>
- Soundarajan S, Eliassi-Rad T, Gallagher B, Pinar A (2017) ϵ -WGX: Adaptive edge probing for enhancing incomplete networks. In: Proceedings of the 2017 ACM on Web Science Conference. ACM, New York. pp 161–170. <http://doi.acm.org/10.1145/3091478.3091492>. <https://doi.org/10.1145/3091478.3091492>
- LaRock T, Sakharov T, Bhadra S, Eliassi-Rad T (2018) Reducing network incompleteness through online learning: A feasibility study. In: The 14th International Workshop on Mining and Learning with Graphs. http://www.mlgworkshop.org/2018/papers/MLG2018_paper_40.pdf
- Wang X, Garnett R, Schneider J (2013) Active search on graphs. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '13. ACM, New York. pp 731–738. <http://doi.acm.org/10.1145/2487575.2487605>. <https://doi.org/10.1145/2487575.2487605>
- Zhao Y, Wang G, Yu PS, Liu S, Zhang S (2013) Inferring social roles and statuses in social networks. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '13. ACM, New York. pp 695–703. <http://doi.acm.org/10.1145/2487575.2487597>. <https://doi.org/10.1145/2487575.2487597>

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
