# Exploring Partially Observed Networks with Nonparametric Bandits

Kaushalya Madhawa[1] and Tsuyoshi Murata[1]

Tokyo Institute of Technology, Ookayama, Meguro-ku, Tokyo 152-8552, Japan
kaushalya@net.c.titech.ac.jp, murata@c.titech.ac.jp
http://www.net.c.titech.ac.jp

**Abstract.** 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. In this paper we formalize the *Adaptive Graph Exploring* problem. 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-arm bandit (MAB) algorithm for identifying which nodes to be queried. Our contributions include: (1) *i*KNN-UCB, a novel nonparametric MAB algorithm, applies *k*-nearest neighbor UCB to the setting when the arms are presented in a vector space. (2) provide theoretical guarantee that *i*KNN-UCB algorithm has sublinear regret and (3) applying *i*KNN-UCB algorithm on synthetic networks and real-world networks from different domains, we show that our method discovers up to 40% more nodes compared to existing baselines.

**Keywords:** network exploration, network search, multi armed bandits

## 1 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. 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* problem. We formally define it in section 3. 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 obtaining information about a profile and its friends (or followers) using an API or a web service. Several rounds of probing updates the sample with new nodes and links found in the neighborhood of queried nodes. The number of times the network can be probed is restricted by a *probing budget*. Thus, the goal is to enhance the observed graph as much as possible within the *probing budget*.

Two approaches have been proposed to solve the problem of reducing the incompleteness of partially observed networks. First approach involves inferring properties of the unseen part of the network using knowledge of the sample. Such methods infers the missing information by fitting a model of network structure to the observed part [11]. However, this is not practical for real-world networks as such methods require more structural information about the complete network. Second approach is acquiring more information by probing as we propose in this paper. Existing heuristic algorithms such as maximum observed degree (MOD) probing and maxreach [23] require the sample to be obtained in a certain way (e.g., uniform edge sampling). In section 4 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. Heuristic probing algorithms get stuck inside communities, making them worse than probing a node in random.

**Our Work.** A high level overview of the proposed adaptive probing algorithm is illustrated in Figure 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 work on inferring structural role [9] and social status [27] of nodes in social networks.

One property which makes estimation of rewards different from a normal prediction problem is that our training data is accumulated over the process of probing. Probing nodes with similar features all the time may result in suboptimal results. This situation is known in reinforcement learning literature as *exploration-exploitation* trade off. Multi-armed bandits [20] 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.
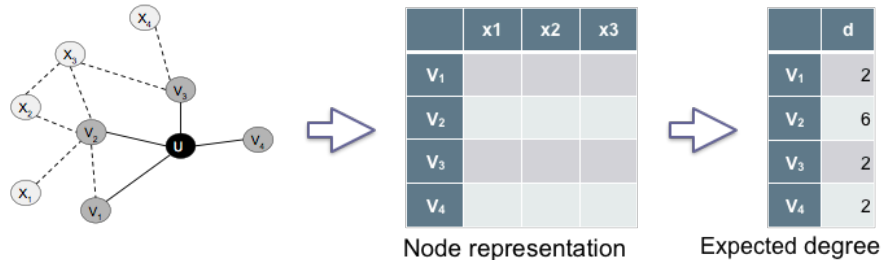
Our contributions are threefold:

**Fig. 1.** prediction pipeline

1. A generic approach for enhancing partially observed networks which does not require any prior knowledge about the network.
2. A novel non-paramteric UCB algorithm ($i$KNN-UCB) to solve the multi-armed bandit problem (MAB) when the arms are represented in a vector space. [1]
3. 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 methods. [2]

The rest of the paper is structured as following. In section 2, we provide an extensive review of related work. section 3 starts with the problem definition and describes our approach in detail. section 4 explains the experimental setup and the data sets being used. Then, in section 5 we present empirical evaluations of our bandit algorithm using real-world networks as well as synthetic networks. Finally, section 6 concludes with a brief discussion of the bandit approach and a few promising directions as future work.

## 2   Related Work

### 2.1   Network Crawling and Sampling

Although this problem looks similar to network crawling and sampling, objective of most sampling algorithms is to select a representative subset of the nodes (or edges) when the entire network is accessible [1]. In contrast, we are improving a given incomplete network and we have no knowledge of how the sample is being obtained. Particularly, snowball sampling [13] can be used when the information about the complete network is not accessible. But it suffers from the same drawbacks as of heuristic algorithms; it does not adapt as the observed information updates. As another related problem, link prediction [15] can predict missing links on a network, but not missing regions of nodes. The only way to enhance the observed sample is by iteratively querying observed nodes and adding their neighboring nodes to the sample.

---

[1] source code available at `https://bitbucket.org/kau_mad/bandits/src/pkdd2018/`
[2] source code available at `https://bitbucket.org/kau_mad/net_complete/src/pkdd2018`

## 2.2   Active Search

Active search on graphs [5, 25] 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 [16]. If only an incomplete view is available, 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 problem (MAB) [17]. SN-UCB1[6] and NETEXP[22] are such MAB based active search algorithms proposed for partially observed networks. Probing a node in NETEXP reveals 2-hop neighborhood, which is not true for real world social networks. SN-UCB1 does not provide a significant improvement over the existing heuristic methods. Soundarajan et al. [24] recently proposed $\epsilon$-WGX, a multi-armed bandit approach to solve Active Edge Probing (AEP) problem in incomplete networks. Though AEP looks similar, it is fundamentally different from ours as a node can be probed multiple times and only one neighboring edge is revealed in each probe.

## 3   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.

### 3.1   Problem Definition

Suppose there is a large unweighted undirected graph $G$ which can not be observed fully, but only a partially observed network $G'$ is available. We denote the initial incomplete network as $G'_0$. 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.

**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 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** $(T \in \mathbb{Z})$
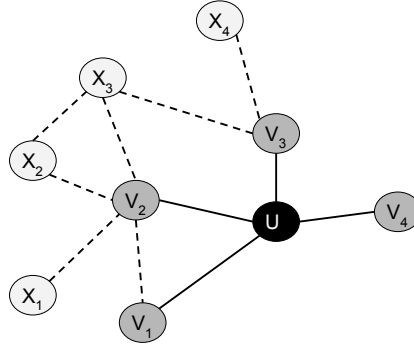


**Fig. 2.** Example of an incomplete network. The black node $U$ is probed and gray nodes $V_1, \cdots, V_4$ are observed. The white nodes $X_1, \cdots, X_4$ exist in the original network $G$, are yet to be observed.

**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 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.

An observed node can either be probed or not probed at the moment. Any observed node which is not probed is considered as a candidate for probing. Hence, we refer such nodes as _candidate nodes_. At the beginning, all the nodes in the given sample are candidate nodes. Probing a candidate node reveals a _reward_ (eg. true degree of a node). Our goal is iteratively selecting **b** candidate nodes that maximizes the cumulative reward (i.e., number of observed nodes).

### 3.2    Calculation of expected reward of candidate nodes

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 [19]: 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 $f : \mathcal{X} \to \mathbb{R}$ gives the expected reward of a given node and $\zeta_t$ is sub-gaussian white noise with mean 0 and variance $\sigma^2$.

**Assumption 1.** *(Lipschitz condition): There exists a constant $L$ such that $|f(x) - f(x')| \leq L \cdot \mathcal{D}(x - x')$ 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 nodes which are similar in terms of their feature vectors will have similar rewards. In the next section, we describe in detail how we formulate this problem as a multi-armed bandit problem.

### 3.3    Bandit Algorithm

**Problem Setting** In the classical contextual 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. In this setting, each arm is assumed to be independent, the rewards are drawn randomly from a probability distribution that is specific to each arm. 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.

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, the classical notion of K-armed bandit problem assumes that the set of $K$ arms would not change over time and requires each arm to be played several times. In contrast, the set of candidate nodes change as probings occur over time. And more importantly, a node can not be probed for a second time.

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 structured bandit problem, the agent deduces relationship between arms based on some $d$-dimensional feature vector $x_a \in \mathbb{R}^d$ assigned to an arm $a$.

**KNN-UCB algorithm for structured bandits** *Linear bandits*[7, 21] the simplest among such models, assumes the reward is linearly dependent on feature vectors and computes the expected reward of an arm by the inner product of its feature vector and a parameter vector $\theta$. But real data often exhibits more complicated relationships than a linear one. Hence, we choose $k$-nearest neighbor (k-NN) regression to estimate the expected reward of arms. We adapt Guan and Jiang [8]'s k-armed KNN-UCB algorithm to the structured setting. Upper confidence bound [2] (UCB) algorithms incorporate an exploration term by calculating a 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

$$a_t = \arg\max_i \hat{f}(x_i) + \alpha \cdot \sigma(x_i) \tag{1}$$

where $\alpha > 0$ is a constant determining the amount of exploration.

**Definition 3.** *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 the 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 term $\sigma(x_i)$ is analogous to the term $T_i(t)$ accounting for the number of times action $i$ has been chosen by the time $t$. The way the network is being probed using $i$KNN-UCB is shown in algorithm 1.

**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 $a_1, a_2, \cdots, a_n$ and we observe rewards $r_{a_1,1}, r_{a_2,2}, \cdots, r_{a_n,n}$. We use the following notion of regret

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

---

**Algorithm 1:** $i$KNN-UCB.

---

**Input**   : incomplete network $G_0' = (V_0', E_0')$, probing budget $b \in \mathbb{N}$, exploration
              parameter $\alpha$, $k$, $T_0$
**Output:** A sequence of $b$ nodes to probe
**Initialize:** *candidate nodes* $= V_0'$

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

---

**Theorem 1.** *Let $M > 0$ be an arbitrary constant. Then the regret is sublinear with, $\mathcal{R}_T \leq M \cdot T^{(1-1/d)}$.*

*Proof.* The regret for bandits in a continuous feature space is

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

Let $\sup_{i \in \mathcal{A}_t} f(x_{i,t})$ be $f_{\pi_t,t}$

$$\mathcal{R}_T = \sum_{t=1}^{T} [f(x_{\pi_t,t}) - f(x_{a_t,t})]$$

Using Lipschitz assumption

$$\mathcal{R}_T \leq \sum_{t=1}^{T} \left[ L \cdot \mathcal{D}(x_{\pi_t,t}, x_{a_t,t}) \right] \tag{5}$$

$$\mathcal{R}_T \leq L \cdot \sum_{t=1}^{T} \left[ \sup_{x \in \mathcal{X}} r_k(x) \right]. \tag{6}$$

From Jiang [10],

$$\sup_{x \in \mathcal{X}} r_k(x) \leq M_1 \cdot \left( \frac{2k}{t} \right)^{1/d} \tag{7}$$

where $M_1 > 0$ is a constant. Using this in eq. (6) results in

$$\mathcal{R}_T \leq L \cdot \Big[ \sum_{t=1}^{T} M_1 \cdot \Big( \frac{2k}{t} \Big)^{1/d} \Big] \tag{8}$$

With $M \geq L \cdot M_1$

$$\mathcal{R}_T \leq M \int_{1}^{T} t^{-1/d} dt \tag{9}$$

$$\leq M \cdot T^{(1-1/d)} . \tag{10}$$

Hence, the regret is sub-linear. $\qquad\qquad\square$

*Remark 1.* If we select $\alpha \geq L$, we can write eq. (5) as

$$\mathcal{R}_T \leq \sum_{t=1}^{T} \alpha \cdot \sigma(x_{a_t,t}) . \tag{11}$$

## 4 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.

1. degree centrality
2. average degree centrality of its neighbors
3. median degree centrality of its neighbors
4. the average percentage of probed neighbors found in the neighborhood

These features are chosen because their effectiveness is shown in previous work on finding structurally similar nodes [9].

### 4.1 Data

We use simulated network data as well as publicly available[3] real-world data sets of social and information networks.

**Synthetic data.** The aim of using synthetic networks is to investigate the behavior of the proposed method on networks with different network configurations. We use two random network models, Barabasi-Albert model (BA) [4] and Lancichinetti-Fortunato-Radicchi (LFR) [12] benchmark to create networks with different characteristics. All these networks have the same number of nodes ($N = 34,546$), the number of nodes in the HepPh citation network. BA model

---

[3] http://snap.stanford.edu/data/index.html

generates networks with power-law degree distributions. But real-world communication networks possess different properties such as homophily [18] which can not be represented by a BA model. We use LFR model to generate networks with community structure. The mixing parameter $\mu$ of LFR model decides the probability of a node linking other nodes belonging to different communities. Low values of $\mu$ 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 $(\mu)$. We created LRF benchmark networks with varying the value of $\mu$ in the range [0.1, 0.5] to investigate the impact of underlying community structure of a network on our method.

**Real-world data.** Table 2 gives a summary of the seven real-world network data sets we use. In citation networks, if a paper $i$ cites another paper $j$, the network contains an undirected edge connecting paper $i$ and paper $j$. Similarly, co-authorship networks represent authors as nodes and two authors are connected if they have published at least one paper together. Nodes of the network Enron-email are email addresses of Enron employees. If user $i$ has sent at least one email to the user $j$, nodes $i$ and $j$ are connected by an undirected edge. Twitter data set is made of 1000 ego-networks consisting of 4,869 Twitter lists [14]. Epinions, and Slashdot can be considered as web of trust networks. Even though Epinion and Slashdot networks are often labeled as online social networks, they differ from the usual notion of social networks as they represent who-trust-whom data of users instead of the relationships or interaction among users. In these networks, a user tags another user as trustworthy or not. They are sparse compared to online social networks.

**Table 2.** Description of data sets. (CA = co-authorship)

|  | HepPh | HepTh | Epinions | Twitter | Stanford | AstroPh | DBLP | Slashdot |
|---|---|---|---|---|---|---|---|---|
| Type | citation | citation | web | social | web | CA | CA | web |
| Nodes | 34,546 | 27,770 | 75,789 | 81,306 | 281,903 | 18,772 | 317,080 | 82,168 |
| Edges | 421,578 | 352,807 | 508,837 | 1,768,149 | 2,312,497 | 198,110 | 1,049,866 | 549,202 |
| Avg Clustering | 0.2848 | 0.3120 | 0.1378 | 0.5653 | 0.5976 | 0.6306 | 0.6324 | 0.0603 |

### 4.2   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 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.

### 4.3   Methods

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

**Algorithms that do not use node features**

- **Random walk (RW).** In this trivial baseline, we select one of the candidate nodes randomly for probing. This is equivalent to running our Bandit Explorer algorithm with only one cluster and using the random strategy for node selection.
- **Maximum observed degree (MOD).** This greedy method proposed in [3] is the current state-of-the-art algorithm for finding the network cover in an online manner.

**Algorithms that use node features**

- **Lin-UCB.** This applies the UCB algorithm by Dani, Hayes, and Kakade [7] 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-$\epsilon$-greedy.** This algorithm chooses a random arm with probability $\epsilon$ while selecting the arm with k-NN regression selects the arm rest of the times.
- *i***KNN-UCB** This is our proposed algorithm, algorithm 1.

## 5   Results

### 5.1   Analysis on Synthetic Networks

We probe incomplete BA and LFR networks obtained by RN and BFS sampling for 1,000 iterations ($T = 1000$). Number of nodes observed in the BA network is shown in Figure 3. For all networks generated by Barabasi-Albert (BA) model, MOD could observe more nodes than bandit algorithm. This confirms Avrachenkov et al. [3]'s claim that MOD probing can achieve the best connected network cover for networks generated by preferential attachment processes.

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 [12]. We vary the mixing parameter $\mu$ 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 smaller $\mu$. When the initial sample is obtained by BFS sampling, $i$KNN-UCB outperforms all baselines by a significant margin. The gap between $i$KNN-UCB and the baseline is larger when the mixing parameter is small, network has significant community structure. 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.
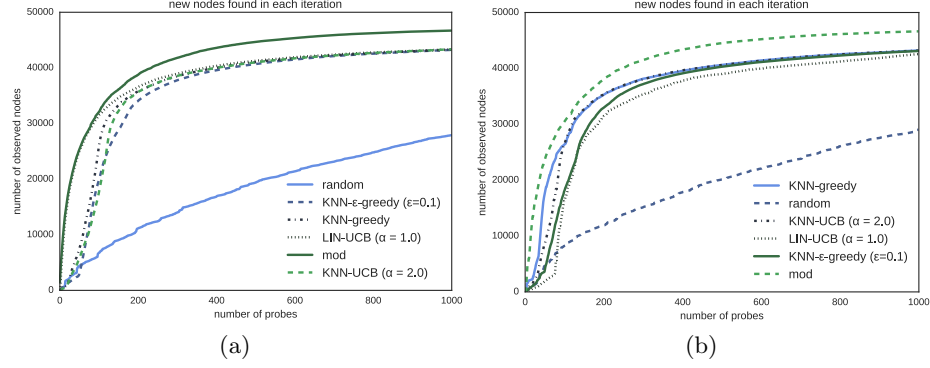
**Fig. 3.** Scale-free network created by Barabasi-Albert model. (nodes=50,000, m = 20) (a) random node (RN) sample (b) BFS sample
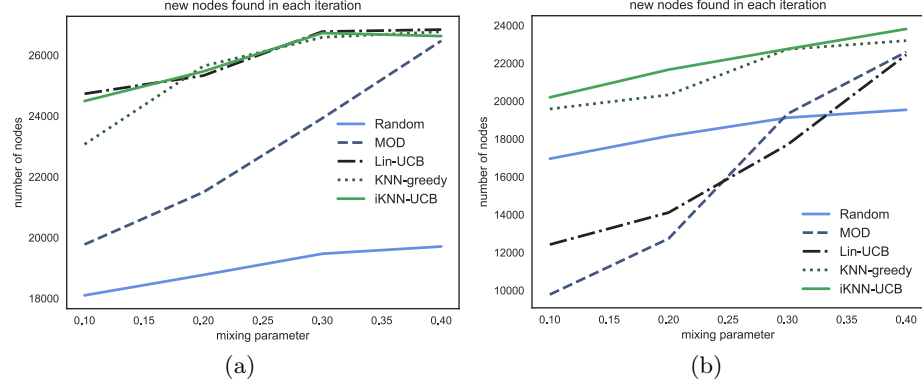


**Fig. 4.** Performance on synthetic networks generated by LFR benchmark (a) RN sample (b) BFS sample

### 5.2   Results on Real World Networks

We use 8 real-world networks mentioned in Table 2 and generate RN and BFS samples containing 5% nodes of the original network $G$. Then 1,000 probing steps are performed. We perform each experiment five times initialized with different random seeds and report the average number of additional nodes which were observed in Figure 5 and Figure 6.

$i$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.
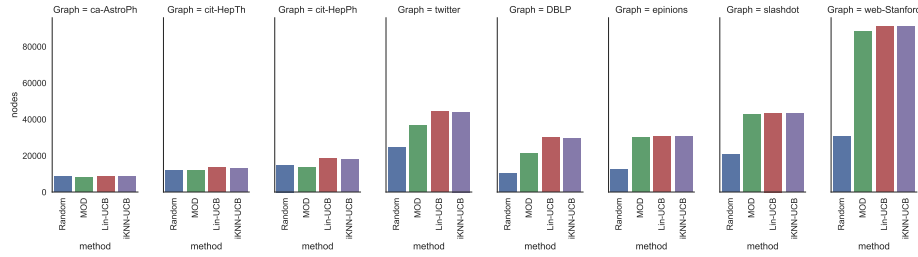
**Fig. 5.** Comparison against baselines: 1000 probes run on 5% nodes of each network. Each sample is created by performing a random walk on the original network
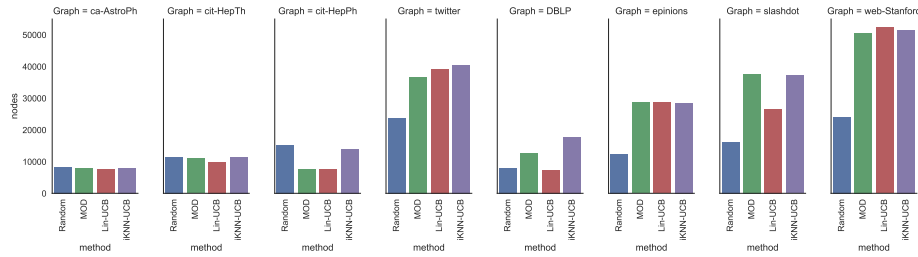


**Fig. 6.** Comparison against baselines: 1000 probes run on 5% nodes of each network. Each sample is created by performing a breadth first walk on the original network

## 6    Conclusions

In this paper, we introduced a bandit based exploration algorithm for partially observed incomplete networks. We proposed a novel nonparametric multi-armed bandit algorithm *i*KNN-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, *i*KNN-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 variety of real-world networks. Using different configurations of LFR benchmark networks, we observed that our algorithm outperforms all other baselines significantly when the network exhibits community structure prominently. Since the reward function is independent from the probing procedure, it is easy to define a new reward function to solve a different graph exploration problem (eg. finding a particular type of nodes).

In this problem, 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 [4]). As future work, we would explore how this current approach can be changed for such different settings of the same problem.

---

[4] https://dev.twitter.com/rest/reference/get/followers/ids

# References

[1]  Nesreen K Ahmed, Jennifer Neville, and Ramana Kompella. "Network sampling: From static to streaming graphs". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 8.2 (2014), p. 7.

[2]  Peter Auer. "Using confidence bounds for exploitation-exploration trade-offs". In: *Journal of Machine Learning Research* 3.Nov (2002), pp. 397–422.

[3]  Konstantin Avrachenkov et al. "Pay few, influence most: Online myopic network covering". In: *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on.* IEEE. 2014, pp. 813–818.

[4]  Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks". In: *science* 286.5439 (1999), pp. 509–512.

[5]  Mustafa Bilgic, Lilyana Mihalkova, and Lise Getoor. "Active learning for networked data". In: *Proceedings of the 27th international conference on machine learning (ICML-10).* 2010, pp. 79–86.

[6]  Zahy Bnaya et al. "Social network search as a volatile multi-armed bandit problem". In: *HUMAN* 2.2 (2013), pp. 84–98.

[7]  Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. "Stochastic Linear Optimization under Bandit Feedback". In: *21st Annual Conference on Learning Theory.* 2008, pp. 0–20.

[8]  Melody Y Guan and Heinrich Jiang. "Nonparametric Stochastic Contextual Bandits". In: *AAAI Conference on Artificial Intelligence.* 2018.

[9]  Keith Henderson et al. "Rolx: structural role extraction & mining in large graphs". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM. 2012, pp. 1231–1239.

[10]  Heinrich Jiang. "Rates of Uniform Consistency for k-NN Regression". In: *arXiv preprint arXiv:1707.06261* (2017).

[11]  Myunghwan Kim and Jure Leskovec. "The network completion problem: Inferring missing nodes and edges in networks". In: *Proceedings of the 2011 SIAM International Conference on Data Mining.* SIAM. 2011, pp. 47–58.

[12]  Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. "Benchmark graphs for testing community detection algorithms". In: *Physical review E* 78.4 (2008), p. 046110.

[13]  Sang Hoon Lee, Pan-Jun Kim, and Hawoong Jeong. "Statistical properties of sampled networks". In: *Physical Review E* 73.1 (2006), p. 016102.

[14]  Jure Leskovec and Julian J Mcauley. "Learning to discover social circles in ego networks". In: *Advances in neural information processing systems.* 2012, pp. 539–547.

[15]  David Liben-Nowell and Jon Kleinberg. "The link-prediction problem for social networks". In: *journal of the Association for Information Science and Technology* 58.7 (2007), pp. 1019–1031.

[16]  Yifei Ma, Tzu-Kuo Huang, and Jeff G Schneider. "Active Search and Bandits on Graphs using Sigma-Optimality." In: *UAI.* 2015, pp. 542–551.

[17]  Aditya Mahajan and Demosthenis Teneketzis. "Multi-armed bandit problems". In: *Foundations and Applications of Sensor Management* (2008), pp. 121–151.

[18]  Miller McPherson, Lynn Smith-Lovin, and James M Cook. "Birds of a feather: Homophily in social networks". In: *Annual review of sociology* 27.1 (2001), pp. 415–444.

[19]  Joseph John Pfeiffer III, Jennifer Neville, and Paul N Bennett. "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. 2014, pp. 639–648.

[20]  Herbert Robbins. "Some aspects of the sequential design of experiments". In: *Bulletin of the American Mathematical Society* 58.5 (1952), pp. 527–535.

[21]  Paat Rusmevichientong and John N Tsitsiklis. "Linearly parameterized bandits". In: *Mathematics of Operations Research* 35.2 (2010), pp. 395–411.

[22]  Adish Singla et al. "Information Gathering in Networks via Active Exploration." In: *IJCAI*. 2015, pp. 981–988.

[23]  Sucheta Soundarajan et al. "MaxReach: Reducing network incompleteness through node probes". In: *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE. 2016, pp. 152–157.

[24]  Sucheta Soundarajan et al. "$\varepsilon$-WGX: Adaptive Edge Probing for Enhancing Incomplete Networks". In: *Proceedings of the 2017 ACM on Web Science Conference*. ACM. 2017, pp. 161–170.

[25]  Xuezhi Wang, Roman Garnett, and Jeff Schneider. "Active search on graphs". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2013, pp. 731–738.

[26]  Duncan J Watts and Steven H Strogatz. "Collective dynamics of'small-world'networks". In: *nature* 393.6684 (1998), p. 440.

[27]  Yuchen Zhao et al. "Inferring social roles and statuses in social networks". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2013, pp. 695–703.