# Information Gathering in Networks via Active Exploration

**Adish Singla**[*]
ETH Zurich
adish.singla@inf.ethz.ch

**Eric Horvitz**
Microsoft Research
horvitz@microsoft.com

**Pushmeet Kohli**
Microsoft Research
pkohli@microsoft.com

**Ryen White**
Microsoft Research
ryen.white@microsoft.com

**Andreas Krause**
ETH Zurich
krausea@ethz.ch

## Abstract

How should we gather information in a network, where each node's visibility is limited to its local neighborhood? This problem arises in numerous real-world applications, such as surveying and task routing in social networks, team formation in collaborative networks and experimental design with dependency constraints. Often the informativeness of a set of nodes can be quantified via a submodular utility function. Existing approaches for submodular optimization, however, require that the set of all nodes that can be selected is known ahead of time, which is often unrealistic. In contrast, we propose a novel model where we start our exploration from an initial node, and new nodes become visible and available for selection only once one of their neighbors has been chosen. We then present a general algorithm NETEXP for this problem, and provide theoretical bounds on its performance dependent on structural properties of the underlying network. We evaluate our methodology on various simulated problem instances as well as on data collected from social question answering system deployed within a large enterprise.

## 1 Introduction

Pioneering work of Stanley Milgram in the 1960's [Milgram, 1967] provided evidence that individuals in a social network, possessing very limited knowledge of the whole network and only being able to access their close acquaintances, can effectively route messages to distant target individuals in the network. Similar phenomena can be observed in many human-powered systems across a spectrum of examples, including the DARPA Red Balloon challenge[1] and co-authorship networks in academic communities. The challenges of local knowledge and limited visibility often arise in various diverse computing systems. This includes querying in peer-to-peer networks constrained by the decentralized network design; information gathering on the internet by web crawlers or humans via following chains of linked documents [White and Singla, 2011] constrained by the inability to directly discover new documents; and users seeking experts via friends in online social networks owing to the privacy constraints. *How can we build systems that* autonomously *explore networks under limited visibility for sake of information gathering?*

**Information gathering.** We formalize these information gathering tasks as actively identifying a set of nodes in a network that maximize a set function quantifying their informativeness. Many natural objectives for this purpose satisfy *submodularity*, an intuitive diminishing returns condition (c.f., Krause and Guestrin (2011)). For instance, in the social Q&A network (or web graph), the problem of finding experts in the network with the desired skills to answer the question (or the documents satisfying the information needs) can be cast as submodular function maximization [El-Arini *et al.*, 2009]. In the message routing problem in the Milgram's experiment, the utility function can be modeled as reduction in distance to the target, measured as the minimal distance from one of the selected nodes to the target.

**Local visibility of the network.** Existing approaches for submodular function optimization are based on the key assumption that the ground set (of all nodes) is known in advance. With this assumption, greedy selection based on the marginal utilities of the nodes provides near-optimal solutions to the problem [Nemhauser *et al.*, 1978]. However, having access to the entire network is unrealistic in many real-world applications for various reasons. For instance, due to privacy concerns, node visibility within social networks (such as Facebook or LinkedIn) is restricted to nodes we already connected to, and only these nodes can be target for routing tasks or posting a question. Even if the whole network is accessible (in a centralized system), the users may be more willing to respond and provide help to solve the task when routed through their acquaintances because of social incentives to help the peers and friends. The fundamental question is how to explore the local neighborhood of the network with the goal of maximizing the utility function over the selected nodes.

### 1.1 Overview of our approach

We present a general approach to information gathering on networks under visibility constraints. In our model, the algorithm starts from an initial node (for instance, the individual in a social Q&A network posting the question or seeking expertise for a task). New nodes become available for selection only once one of their neighbors has been already cho-

[*]Adish Singla performed this research during an internship at Microsoft Research.

[1]http://archive.darpa.mil/networkchallenge/

sen. We model the local visibility constraints via limiting the number of hops within which the neighborhood of selected nodes becomes visible. Given this, the selection algorithm has to choose between exploring the network by selecting high degree nodes to expose new nodes and edges, or exploiting the currently accessible neighborhood by choosing nodes that provide maximal marginal utilities. Our main algorithmic contribution is a novel algorithm NETEXP for this problem. We analyze its performance both in settings where no structural properties of the network are known, and in more specific settings capturing properties of real-world collaborative networks. Our main contributions are:

- A formal model of information gathering in networks with local visibility constraints, capturing several real-world applications, such as task routing in social networks.
- A novel algorithm NETEXP for this problem that actively explores the accessible local neighborhood to increase visibility of the unseen network, while at the same time exploiting the value of the information available in this neighborhood. We analyze the performance of our algorithm and provide theoretical bounds depending on the network structure.
- Evaluation of our approach on data collected from a real-world application of task routing in a social Q&A system deployed within a large enterprise, to show the practical applicability of our methodology.

## 2 Related Work

**Search and navigation with local knowledge.** Our work is inspired by the ideas of navigation and search in networks with local knowledge. The seminal work of Kleinberg (2000; 2002) addressed algorithmic questions of network formation and search strategies, to understand when short paths of acquaintances exists and are discoverable by local navigation. In our work, we generalize the task of navigating to a target, to that of information gathering in a network. Adamic *et al.* (2001) and Adamic and Adar (2005) compared different local search strategies, for instance, following high-degree nodes or using proximity in an organizational hierarchy in the context of an email network. Leskovec and Horvitz (2014) tackled the challenge of navigating to a target in a geospatial network from a source using local knowledge of the network and proposed different navigation strategies – based on degree, or being closer in distance to the target, as well as combining these two strategies. Local search and team recruiting have also been studied in context of strategic participants via providing incentives [Kleinberg and Raghavan, 2005; Kota and Narahari, 2010], which also formed the basis of recruiting participants in the Red Balloon challenge[1].

**Task routing.** Richardson and White (2011) developed "IM-an-Expert", a synchronous social Q&A system, where a user can pose a question via instant messaging that is then routed to an "expert" by the centralized system. Each posted question is characterized by the features and required skills; the users in the system have different expertise level, and the central system routes the question to the available experts. The main difference in our setting is the absence of any central authority, leading to the challenge of using local navigation to find the experts. More advanced variants of the social query model are proposed by Horowitz and Kamvar (2010) and Banerjee and Basu (2008), however their proposed models do not provide theoretical guarantees. Jeong *et al.* (2013) proposed "SearchBuddies" that route questions to friends in online networks, but did not study the algorithmic aspects of this routing. The challenge of locating experts in a decentralized manner has been studied [Nushi *et al.*, 2015; Xu *et al.*, 2010], however, no formal analysis has been provided for the proposed techniques. Zhang *et al.* (2012) considered task routing for prediction tasks, by designing routing-based scoring rules and studying their truthfulness and efficiency in the equilibrium. However, we are not focusing on strategic aspects of the participants, and rather the algorithmic aspects of information gathering in networks.

**Explore-exploit dilemma.** Many reinforcement learning and online learning problems are associated with explore-exploit dilemma, and several solutions use the framework of multi-armed bandits (MAB) [Lai and Robbins, 1985]. Motivated by applications in social advertisement, Fang and Tao (2014) considers social-MAB where the actions or arms are the users of the underlying social network. Bnaya *et al.* (2013) considers the application of social network search and targeted crawling by using MAB framework. One of the key differences in our work is that the "exploration" refers to acquiring access to more nodes in the network that were previously inaccessible, and is very different from the usual notion of "exploration" used in MAB literature.

## 3 Problem Statement

We now formalize the problem addressed in this paper.

**Set of nodes and the network.** We consider a set of nodes (*e.g.*, a population of people or users of a system) denoted by set $V = \{v_1, v_2, \ldots, v_{|V|}\}$, of size $|V|$. There is an underlying network over the nodes, denoted by $G = (V, E)$. For now we will not make any assumptions about its structure. Instead we discuss this further during the analysis, as our bounds will depend on the specific assumptions that we assert.

**Task and the utility function.** We denote a task as $T$, associated with an initial node $v_T^o$. For instance, in a social Q&A system, the task $T$ could be a question, and $v_T^o$ is the user posting the question. We associate with each task $T$ a function over the set of nodes, given by $f_T : 2^V \to \mathbb{R}$, quantifying their informativeness/expertise. Given a set of nodes $S$ selected for task $T$, the utility achieved from this set is given by $f_T(S)$. We assume each $f_T$ to be *non-negative*, *monotone* (*i.e.*, whenever $S \subseteq S' \subseteq V$, it holds that $f(S) \leq f(S')$) and *submodular*. Submodularity is an intuitive notion of diminishing returns, stating that, for any sets $S \subseteq S' \subseteq V$, and any given node $v \notin S'$, it holds that $f(S \cup \{v\}) - f(S) \geq f(S' \cup \{v\}) - f(S')$. These conditions are general, and are satisfied by many realistic, as well as complex utility functions for information gathering [Krause and Guestrin, 2011; Krause and Golovin, 2012].

**Local visibility and connectivity of selected set.** We seek algorithms for the setting where the network is revealed incrementally as more nodes are selected. We denote the $l$-hop neighborhood of a node $v$ as $\mathcal{N}(v, l) \subseteq V$, to be the set of all nodes in the network that are connected to $v$ either directly or via at most $(l-1)$ intermediate nodes. For simplicity, we shall

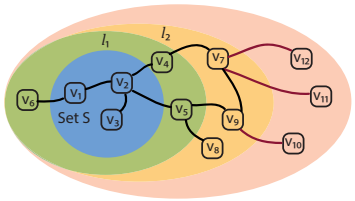



**Figure 1:** Illustration of local visibility constraints for $l_{deg} = 2$, $l_{val} = 2$. The currently selected set is given by $S = \{v_1, v_2, v_3\}$. The 1-hop neighborhood of $S$, given by set $\{v_4, v_5, v_6\}$, is now accessible to the algorithm for selection. Given $l_{val} = 2$, the algorithm can evaluate $f$ to compute marginal utilities for the set $\{v_4, v_5, v_6, v_7, v_8, v_9\}$. Given $l_{deg} = 2$, the 1-hop neighborhood of the nodes $\{v_4, v_5, v_6, v_7, v_8, v_9\}$ is visible to the algorithm in order to find the network they expose. For instance, the algorithm can compute that node $v_7$ exposes the set of nodes $\{v_9, v_{11}, v_{12}\}$

assume that $v$ is also included in $\mathcal{N}(v, l)$. For a set of nodes $S$, we define its $l$-hop neighborhood as $\mathcal{N}(S, l) = \cup_{v \in S} \mathcal{N}(v, l)$. Since the network is unknown in the beginning, we seek an algorithm that incrementally selects nodes always within the 1-hop neighborhood of already selected nodes.

In order to make informed decisions, we will assume that some information is being revealed to the algorithm while nodes are being selected. In particular, assuming we have already selected nodes $S$, for every node $v$ in $N(S, l_{deg})$, its 1-hop neighborhood $N(v, 1)$ is visible. This essentially means that the algorithm has sufficient information in order to find the new network nodes that will be exposed if $v$ is added to the current set. We furthermore assume that the objective function $f$ can be evaluated for every node $v$ in $N(S, l_{val})$. The constants $l_{deg}$ and $l_{val}$ will be specified later. Figure 1 illustrates this with a simple example for $l_{deg} = 2$ and $l_{val} = 2$. In real-world social networks (such as Facebook or LinkedIn), the visibility is usually restricted to $l_{deg} = 1$ and $l_{val} = 1$ due to privacy settings.

**Optimization problem.** Consider a task $T$ originating at any node $v^o$, and $f$ its associated utility function. We seek to select a set of nodes $S$ of value at least $Q$ in terms of utility. The process starts from the initial node $v^o$ and sequentially adds new nodes maintaining the connectivity of $S$. We are interested in designing an algorithm `Alg` that has only $l_{deg}$ and $l_{val}$ visibility, and achieves $Q$ while minimizing the cost of the selected set. Formally, we define the optimization problem as follows:

$$S^* = \arg\min_{S \subseteq V} |S| \tag{1}$$

$$\text{s.t. } f(S) \geq Q \tag{2}$$

$$S \texttt{ is connected in } G \tag{3}$$

$$v^o \in S \tag{4}$$

$$\texttt{Algorithm Alg} \text{ has } l_{deg}, l_{val} \text{ visibility} \tag{5}$$

Note that due to constraint (5), an optimal solution to this problem is in fact not a fixed set, but a *policy* that specifies which next node to select as a function of the information revealed. We also define a fixed optimal solution set VOPT that achieves value of $Q$, not subject to any computational constraints nor the constraints given by (3), (4) and (5).

# 4 Our Algorithm NETEXP

We begin by describing the high level ideas behind our algorithm NETEXP and then provide details.

## 4.1 Overview of basic approach

To present some of the key insights in designing NETEXP, let us first consider two extreme settings.

**Special case: fully connected network.** Let us first consider the extreme setting where the network graph $G$ is fully connected. This, in turn, makes the constraints (3), (4) and (5) for the optimization problem as posed in Section 3 redundant. In this case, it reduces to an instance of standard submodular optimization [Nemhauser *et al.*, 1978; Krause and Guestrin, 2011]. Ignoring the computational constraints, the size of the fixed optimal solution set is given by $|\text{VOPT}|$, as defined in Section 3. Obtaining this solution is intractable, however a greedy selection based on the marginal utilities of the nodes can provide a solution of size bounded by $O(|\text{VOPT}| \cdot \ln(\frac{1}{\beta}))$ to achieve a utility of $(1-\beta) \cdot Q$ where $\beta$ indicates a tolerance level. If $f$ is integral, any $\beta > 1/Q$ suffices to ensure complete coverage (*i.e.*, exact value $Q$).

**Building the network backbone.** Now suppose the network is arbitrary. One approach would be to first build a connected "backbone" of the network by selecting a set of nodes $S$ that covers the whole set $V$, *i.e.*, $\mathcal{N}(S, 1) = V$. Once this "backbone" is built, the whole network is visible and all the nodes are accessible to be selected by the algorithm. Subsequently, the problem again reduces to that of standard submodular optimization. This "backbone" is called a *connected dominating set* of the graph, and the size of minimal connected dominating set is given by $\gamma_G^c$. Computing such a minimal connected dominating set is intractable, but approximation algorithms are known [Guha and Khuller, 1998]. One naive approach to Problem 1 is hence to first build a complete "backbone", following by greedy selection of nodes based on marginal values. This approach guarantees a feasible solution, but it is obviously undesirable for large networks where $\gamma_G^c$ might be very large. Furthermore, there may be some high value nodes closer to the initial node, and hence building the complete dominating set may be unnecessary.

## 4.2 Explore-exploit dilemma

The two basic extreme settings highlight two key regimes that any policy must be able to handle: *Exploit* an exposed (sub)-graph by selecting nodes to achieve value $Q$, and *explore* an unknown graph to expose nodes for selection. Our algorithm will interleave these steps.

**Exploration of the network.** One natural choice for adding a node to the current set is to select one that provides maximal "exposure" of the unconnected network. The "exposed" neighborhood of the selected set $S$ is the $l = 1$ neighborhood excluding $S$, formally given by $\widetilde{\mathcal{N}}(S, 1) = \mathcal{N}(S, 1) \setminus S$. Assuming we have already selected a set $S$, adding a new node $v$ exposes an additional $\left|\widetilde{\mathcal{N}}(S \cup \{v\}, 1) \setminus \widetilde{\mathcal{N}}(S, 1)\right|$ nodes that becomes available to the algorithm to be selected. This node $v$ may not provide immediate value, however, the additional connections in the newly exposed network might be useful, potentially helping to discover and connect with the required high valued nodes. We call this greedy, value agnostic, choice "exploration".

**Algorithm 1:** Algorithm NETEXP

1 **Input**:
- Task: $T$; Initial node: $v^o$; Utility function: $f$;
- Local visibility: $\{l_{deg}, l_{val}\}$;
- Exploration rate: $\epsilon$; Value required: $Q$; Tolerance: $\beta$;

2 **Output**: selected set $S$;
3 **Initialize**: $S = \{v^o\}$; $i = 1$; $\epsilon^i = \epsilon$;
4 **while** $f(S) < (1-\beta) \cdot Q$ **do**
5  **if** $(\max_{v \in \widetilde{\mathcal{N}}(S,1)} |\mathcal{N}(S \cup \{v\}, 1)| = |\mathcal{N}(S,1)|)$ **then**
6  $\epsilon^i = 0$; ▷ (*i.e.*, $\mathcal{N}(S,1) = V$) Update $\epsilon^i$
7  With prob. $\epsilon^i$, $a^i \leftarrow$ EXPLORE; else, $a^i \leftarrow$ EXPLOIT;
8  **if** $a^i =$ EXPLORE **then**
9  $\Pi^* = \arg\max_{\Pi_l^S : l \in [1 \ldots l_{deg}]} \frac{|\widetilde{\mathcal{N}}(S \cup \Pi_l^S, 1) \setminus \widetilde{\mathcal{N}}(S,1)|}{l}$ ;
10  $S = S \cup \Pi^*$;
11  **if** $l_{deg} = 1$ **then**
12  Randomly pick $v^* \in \widetilde{\mathcal{N}}(S \cup \Pi^*, 1) \setminus \widetilde{\mathcal{N}}(S,1)$;
13  $S = S \cup \{v^*\}$;
  **else**
14  $\Pi^* = \arg\max_{\Pi_l^S : l \in [1 \ldots l_{val}]} \frac{f(S \cup \Pi_l^S) - f(S)}{l}$ ;
15  $S = S \cup \Pi^*$;
16  $i = i + 1$; $\epsilon^i = \epsilon^{i-1}$;
17 **Output**: $S$

**Exploitation of value.** The other natural choice of selecting the next node is to add the one that provides immediate value to the set in terms of maximal marginal utility value. In context of an already selected set $S$, adding a new node $v$ provides a marginal gain in value, given by $f(S \cup \{v\}) - f(S)$. We call this choice "exploitation", as it greedily maximizes utility given the currently exposed network.

### 4.3 Algorithm NETEXP

**Interleaving network exploration with exploitation.** The key idea is to interleave the two choices of network exploration and exploitation. While the exploration step will continue towards building the connected dominating set (that we need in the worst case anyways as per results in Theorem 3), the exploitation step will greedily select the nodes to maximize the information gathered and terminate the algorithm as soon as the desired level of utility is achieved. Our algorithm NETEXP is illustrated in Algorithm 1. It turns out that this simple approach allows us to derive tight theoretical bounds on the performance, and it also runs quite efficiently for various problem instances. NETEXP interleaves exploration and exploitation with $\epsilon$ probability, as illustrated in Step 7 of Algorithm 1. The $\epsilon$ is constant for the procedure and is provided as input. If further prior information about the network properties or the distribution of features is available, this parameter can be tuned (for instance, whether to do more exploration or more exploitation based on such properties). In Algorithm 1, we simply use a constant $\epsilon$, and when the full network is exposed (or dominating set for the network is already built), we set $\epsilon = 0$ (see Step 6).

**Look-ahead during exploration** As noted above, the exploration steps of the algorithm aim towards building the connected dominating set. Negative results from Guha and Khuller (1998) show that to effectively build connected dominating sets, adding one node at a time based on the criterion of maximal "exposure" is not sufficient, as such a greedy approach may need upto $\Omega(|V|)$ nodes to build a connected dominating set. Our idea is based on the intuition used by Guha and Khuller (1998) on how to effectively add upto two nodes, when $l_{deg} \geq 2$, in a way to be able to build efficient connected dominating sets. We generalize the idea of one-step look ahead for $l_{deg} = 2$ to that of doing a $l-1$ step look ahead for $l_{deg} = l$ during exploration. To formalize this, we introduce the concept of a *chain of length* $l$. Consider the current selected set $S$. We define an $l$-chain from $S$, denoted by $\Pi_l^S$ as an ordered set satisfying the following constraints:

$$\Pi_1^S \in \widetilde{\mathcal{N}}(S,1);$$
$$\Pi_2^S \in \widetilde{\mathcal{N}}(S \cup \Pi_1^S, 1) \setminus \widetilde{\mathcal{N}}(S,1);$$
$$\Pi_{i \geq 3}^S \in \widetilde{\mathcal{N}}(S \cup \Pi_{i-1}^S, 1) \setminus \widetilde{\mathcal{N}}(S \cup \Pi_{i-2}^S, 1);$$

In Step 9, all possible chains of length 1 to $l$ are enumerated to find the set of nodes to add at every iteration. While only considering chains is not required for the analysis, it restricts the search space. Nevertheless, the number of chains to consider is exponential in the lookahead. However, this is not an issue for practical settings where $l \leq 2$ usually.

**Look-ahead during exploitation** The similar idea of look-ahead as used above is also employed during the exploitation step, and is done in Step 14. This look-ahead can also be thought of as moving in the direction of the utility function's (discrete) "gradient".

**The case of $l_{deg} = 1$.** Recently, Borgs *et al.* (2012) introduced a simple randomization technique to efficiently build up connected dominating sets when "look-ahead" is not possible, *i.e*, for the case $l_{deg} = 1$. Specifically, after a node $v$ is added, another randomly selected node from its neighborhood $\mathcal{N}(v,1)$ that is newly exposed is also added. We use the same trick in NETEXP, in Step 12 of Algorithm 1.

## 5 Performance Analysis

We now analyze the performance of NETEXP. The proofs and details are available in the extended version of the paper [Singla *et al.*, 2015].

### 5.1 Analysis for general settings

We first analyze NETEXP for general settings, without any assumptions on the network structure or the utility function. We will state our results in terms of two important network properties: *i)* the maximum degree of any node in the graph $G$, which we denote as $\Delta_G$, and *ii)* smallest size of any *connected* dominating set $\gamma_G^c$ as defined in Section 4.1. We start with analyzing the setting where $(l_{deg} = 2, l_{val} = 1)$ (similar results hold for $l_{deg} \geq 2, l_{val} \geq 1$ as well):

**Theorem 1.** *For $l_{deg} = 2$, $l_{val} = 1$,* NETEXP *terminates with set $S$ achieving utility of at least $(1-\beta) \cdot Q$, satisfying the constraints of* (3)*,* (4) *and* (5) *for Problem 1, with the following bound on the size of $S$ in expectation (over the coin flips), given by $\mathbb{E}[|S|] \leq$*

$$\Big(\frac{1}{\epsilon} \cdot \big(2 + 2\ln(\Delta_G)\big) \cdot \gamma_G^c\Big) + \Big(\frac{1}{1-\epsilon} \cdot |\text{VOPT}| \cdot \ln(\frac{1}{\beta})\Big)$$

Common social networks only satisfy $l_{deg} = 1$ (*i.e.*, nodes can see their friends' friends). For this more challenging case we can still prove the following (slightly weaker) result:

**Theorem 2.** *For $l_{deg} = 1$, $l_{val} = 1$,*NETEXP *terminates with set $S$ achieving utility of at least $(1-\beta)\cdot Q$, satisfying the constraints of* (3)*,* (4) *and* (5) *for Problem 1, with the following bound on the size of $S$ in expectation (over the coin flips) that holds with probability at least $1 - e^{-\gamma_G^c}$, given by*

$$\mathbb{E}[|S|] \leq \Big(\frac{1}{\epsilon}\cdot\big(4 + 2\cdot\ln(\Delta_G)\big)\cdot\gamma_G^c\Big) + \Big(\frac{1}{1-\epsilon}\cdot|\text{VOPT}|\cdot\ln(\frac{1}{\beta})\Big)$$

The proof of Theorems 1 and 2 involves separately analyzing the sequence of explore and exploit actions during the execution of the procedure, and ensuring that interleaving these sequences preserves the analysis.

For the general setting, we also have the following lower bound, showing that the dependency on $\gamma_G^c$ is unavoidable.

**Theorem 3.** *For any bounded $l_{deg}$ and $l_{val}$, there exists a problem instance for which any feasible policy will need to select a set of size at least $|S| \geq \max\Big(\gamma_G^c,\ |\text{VOPT}|\Big)$*

The proof of Theorem 3 follows from the arguments of Section 4.1 and by crafting a worst-case instance of graph structure and distribution of node values for any given policy.

### 5.2 Analysis for realistic settings

We now analyze the results for the specific setting motivated by real-world collaborative networks such as co-authorship networks in academic communities.

**Feature distribution and network structure.** The nodes are associated with subsets of a feature set $X$ of size $|X|$. Considering the specific domain of co-authorship networks, a feature $x \in X$ could, *e.g.*, be an indicator variable denoting whether a user has published a paper in an AI conference. For each feature $x \in \mathcal{X}$, let the set of nodes possessing that feature be given by $V_x$. We consider each feature $x \in \mathcal{X}$ as a "social dimension" that induces a network given by $G_x = (V_x, E_x)$. In this model we assume that each network $G_x$ is formed by a preferential attachment process [Barabási and Albert, 1999] leading to a power law distribution of the degrees. The final network that we observe is obtained by an overlay of these networks, given by $G = \cup_{x\in\mathcal{X}} G_x$. For a node $v$, we denote the value of its feature $x \in X$ as $x_v$ and is proportional to the rank order dictated by the degree of node $v$ in graph $G_x$. This notion of feature values essentially captures the "authority" of a node over a particular feature (for instance, a node gains expertise in "AI" if it has high degree of connections for this particular network).

**Characteristics properties of utility function.** For a given task $T$, we consider a separable utility function given by: $f(S) = \sum_{x\in X} w^x \cdot f^x(S)$, where $w^x$ denotes the weight of function $f^x$. Here, the function $f^x$ depends only on nodes' features $x$. The maximum value of function $f^x(S) = 1$ is achieved after including a top valued node with feature $x$.

Theorem 4 states the main result for these settings, and is proved using the results on the navigation properties of "small-world" networks [Bollobás, 2003; Borgs *et al.*, 2012]. Compared to the general settings of Theorem 1 and 2, the "small-world" networks are much easier to navigate as can be seen in the polylogarithmic bounds.

**Theorem 4.** *Consider $l_{deg} = 1$, $l_{val} = 1$ and a task $T$ originating from user $v^o$ possessing a feature $x^o$. With probability of at least $1 - o(1)$,* NETEXP *terminates with set $S$ achieving utility of at least $(1-\beta)\cdot Q$, with the following bound on the size of $S$ in expectation, given by $\mathbb{E}[|S|] \leq$*

$$\Big(\frac{1}{\epsilon}\cdot O\big(\ln^4(|V_{x^o}|)\big)\Big) + \Big(\frac{1}{1-\epsilon}\cdot O\big(\sum_{x\in X}\ln^4(|V_x|) + |\text{VOPT}|\big)\Big)$$

Next, we state the lower bound in Theorem 5, which follows from the expected diameter of the graph obtained from preferential attachment [Bollobás and Riordan, 2004].

**Theorem 5.** *For any bounded $l_{deg}$ and $l_{val}$, there exists a problem instance for which any feasible policy will need to select a set of size at least $\mathbb{E}[|S|] \geq O\Big(\frac{\ln(|V|)}{\ln\ln(|V|)}\Big)$*

## 6 Experimental Evaluation

We now report on the results of our experiments.

### 6.1 Experimental Setup

**Benchmarks and Metrics.** We compare the performance of our procedure NETEXP against the following three baselines: *i*) RANDOM is a trivial baseline that randomly selects users from the exposed network to add to the set; ; *ii*) DEG is equivalent to running NETEXP with $\epsilon = 1$; and *iii*) VAL is equivalent to running NETEXP with $\epsilon = 0$. The primary metric that we compare is the utility achieved as a function of the size of the selected set $S$. We further carried out experiments by varying $\epsilon$, varying the distribution of node values and comparing the size of the exposed network.

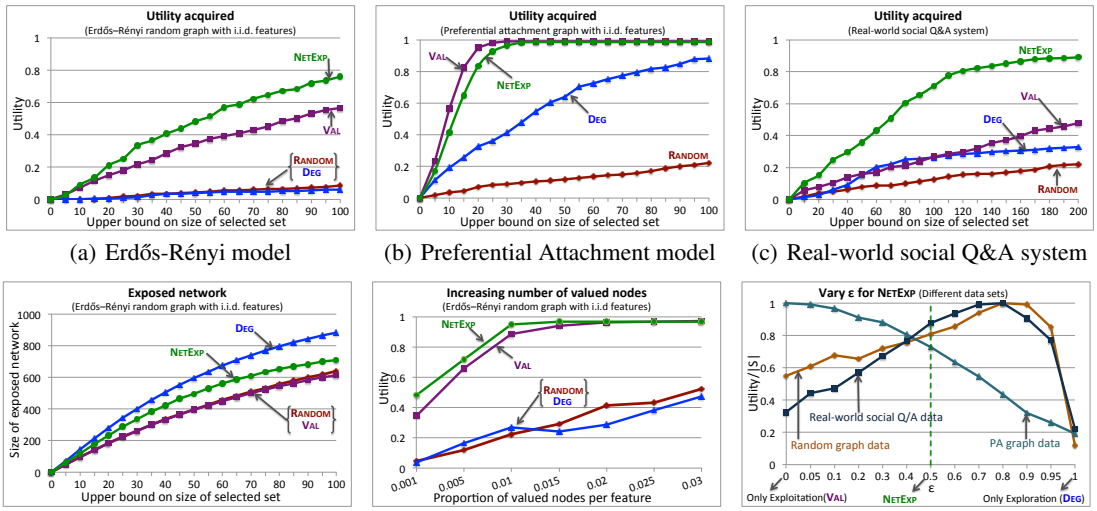
**Utility function and features.** We consider a feature set $X$ of size $|X|$ associated with the nodes $V$, and denote the value of feature $x \in X$ for node $v \in V$ as $x_v$. For a given task $T$, we consider separable utility functions per feature and can write $f(S) = \frac{1}{\sum_{x\in X} w^x}\sum_{x\in X} w^x \cdot f^x(S)$, with weights given by $w^x \in \{0, 1\}$. We consider a probabilistic notion of utility derived from the feature values and define a submodular utility function per feature $f^x(S) = \Big(1 - \prod_{s\in S}(1 - x_s)\Big)$, inspired by El-Arini *et al.* (2009).

**Implementation choices.** We considered a realistic setting of $l_{deg} = 1$ and $l_{val} = 1$ visibility. For running different algorithms, we considered value quota $Q = 1$, with tolerance of $\beta = 0.05$. We used $\epsilon = 0.5$ for NETEXP across all of the datasets without any further tuning. For the implementation of the NETEXP, we report results where we skipped the Step 12 of Algorithm 1 of adding a random user after every exploration step for $l_{deg} = 1$ case. While this step is useful to handle the worst case scenarios of the graph configuration to achieve desirable bounds, for practical purposes, it did not provide any additional benefit or performance gain.

### 6.2 Datasets

We performed experiments with three different datasets.

**Erdős-Rényi random graph.** We created random graphs using the random graph model of Erdős and Rényi (1959), with ground set of size $|V| = 1000$ and the probability of an edge existing between any two nodes given by $p_{edge} = 0.01$, a threshold probability above which the graph is connected with high probability. We then considered a total of 5 features (*i.e.*, $|X| = 5$), distributed *i.i.d.* across the nodes with probability given by $p_{val} = 0.001$. Note, that this probability is essentially equivalent to having one non-zero valued node in the set per feature on expectation. For these *non-zero* valued nodes per feature, the actual values of the features are then




(a) Erdős-Rényi model (b) Preferential Attachment model (c) Real-world social Q&A system

(d) Exposed network (e) Varying fraction of experts (f) Varying $\epsilon$ for NETEXP

**Figure 2:** Figure 2(a), 2(b) and 2(c) shows the utility acquired as the size of the selected set $S$ increases. Results show robustness of NETEXP for different network structures and feature distributions. For the Erdős-Rényi graph model, Figure 2(d) shows the exposed network for different algorithms and Figure 2(e) shows the results of varying the proportion of valued nodes. Figure 2(f) shows the results of varying the $\epsilon$ for NETEXP, showing the robustness of the NETEXP compared to explore-only ($\epsilon = 1$) and exploit-only ($\epsilon = 0$) methods.

uniformly sampled and scaled to lie in the range $(0, 1]$. We also vary $p_{val}$ used by results in Figure 2(e). We generated a set of the tasks, each associated with 3 required features out of 5 (*i.e.*. 3 functions have non-zero weights $w^x$). Distribution of tasks is generated by a random sampling of the required features, as well as the initial user $v^o$ per task.

**Preferential attachment graph per skill.** As a second dataset, we used settings similar to those discussed in Section 5.2. We considered a ground set of size $|V| = 100,000$ and total of 10 distinct features (*i.e.*, $|X| = 10$). The features are distributed *i.i.d.* with probability 0.2 (*i.e.*, on the expectation, every node is associated with two features). The distribution of the tasks is generated similar to the first dataset, with three randomly selected relevant skills per task.

**Real-world social Q&A system.** As a third study, we used a real-world dataset from a synchronous social Q&A system named IM-an-Expert (IMX), which routes incoming questions to candidate answerers via instant messaging [Richardson and White, 2011]. We obtained usage data for this system, from a deployment inside a large enterprise over a period of two years (May 2012 to April 2014). During that time, the system was used by about 5,000 subscribed users (the "experts"), embedded in the network defined by the organizational tree hierarchy of over $180,000$ nodes. These "experts" each have a profile that had been inferred from their homepages, email distribution list subscriptions, and a self-described set of keywords describing their expertise. Users can pose a question in IMX via text, for example, *Excel: How do I set default pivot table to "Classic"?*. For each such question, the system assigns an expertise score to the subscribed users, in the range $(0, 1]$, based on matching the content of question keywords against their profiles. While the deployed IMX system operates in a centralized manner with questions being routed to any expert, we simulated the system as only being able to locate experts using local search.

### 6.3 Results

**Utility acquired w.r.t set size.** Figure 2(a),2(b),2(c) illustrates the utility acquired as the specified upper bound on the selected set $S$ is increased, showing the robustness of NETEXP. The difference in the rate at which the utility is acquired for the random graph (Figure 2(a)) and preferential attachment graph models (Figure 2(b)) is in line with the theoretical results we had studied for worst-case as well as scale-free networks in Theorem 2 and Theorem 4, respectively. The results on the real-world IMX dataset in Figure2(c) shows a large positive gap in acquired utility for NETEXP compared to the baselines. Note that the properties of the IMX dataset are very different from the two synthetic datasets, since the users are embedded in an organizational tree hierarchy.

**Exposed network and varying feature distribution.** The results in Figure 2(d) shows that NETEXP also maintains a good balance of the exposed network in comparison to VAL and to DEG (*i.e.*, specifically optimizing for maximizing this exposure). Figure 2(e) compares the different algorithms (with $|S| = 50$), as the proportion of valued nodes in the network are increased in the first dataset.

**Varying $\epsilon$ for NETEXP.** Figure 2(f) shows the results for NETEXP with varying $\epsilon$ on different three datasets, by restricting $|S|$ to 100, 100 and 200 respectively. The results show the robustness of NETEXP compared to explore-only ($\epsilon = 1$) and exploit-only ($\epsilon = 0$) methods. The results plot the utility acquired divided by the size of selected set, (*i.e.*, $f(S)/|S|$), normalized to the same scale across datasets.

## 7 Conclusions and Future Work

We formalized the process of selecting nodes in a network for information gathering, limited by local knowledge and visibility of the network. Our model captures many real-world applications that have been previously studied only under the assumption that the underlying network and complete set of nodes is known and accessible in advance. We developed

a general algorithm NETEXP that provides a simple way to trade-off between network exploration and the exploitation of value in the exposed neighborhood. We proved strong theoretical guarantees on its performance, and evaluated our approach on several simulated instances as well as using data collected from a real-world social Q&A system. We see several interesting directions in which the presented work can be extended. We assumed that the utility of a node can be observed perfectly. Extending the proposed approach for the case of noisy/stochastic function evaluations would be practically useful. Designing algorithms that adjusts the rate of exploration $\epsilon$ and take into account for some prior information available about the network structure and utility distribution is an interesting direction for future work.

**Acknowledgments.** We'd like to thank Johan Ugander for helpful discussions. Andreas Krause was supported in part by a Microsoft Research Faculty Fellowship and ERC StG 307036.

## References

[Adamic and Adar, 2005] Lada A. Adamic and Eytan Adar. How to search a social network. *Social Networks*, 2005.

[Adamic *et al.*, 2001] Lada A. Adamic, Rajan M. Lukose, Amit R. Puniyani, and Bernardo A. Huberman. Search in power-law networks. *Physical review E*, 2001.

[Banerjee and Basu, 2008] Arindam Banerjee and Sugato Basu. A social query model for distributed search. *University of Minnesota, Minneapolis*, 2008.

[Barabási and Albert, 1999] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[Bnaya *et al.*, 2013] Zahy Bnaya, Rami Puzis, Roni Stern, and Ariel Felner. Social network search as a volatile multi-armed bandit problem. *HUMAN*, 2(2):pp–84, 2013.

[Bollobás and Riordan, 2004] Béla Bollobás and Oliver Riordan. The diameter of a scale-free random graph. *Combinatorica*, 24(1):5–34, 2004.

[Bollobás, 2003] B. Bollobás. Mathematical results on scale-free random graphs. In *Handbook of Graphs and Networks*, pages 1–37. Wiley, 2003.

[Borgs *et al.*, 2012] Christian Borgs, Michael Brautbar, Jennifer Chayes, Sanjeev Khanna, and Brendan Lucier. The power of local information in social networks. In *Internet and Network Economics*, pages 406–419. Springer, 2012.

[El-Arini *et al.*, 2009] Khalid El-Arini, Gaurav Veda, Dafna Shahaf, and Carlos Guestrin. Turning down the noise in the blogosphere. In *SIGKDD*, pages 289–298, 2009.

[Erdős and Rényi, 1959] P Erdős and A Rényi. On random graphs. *Publicationes mathematicae*, 6:290–297, 1959.

[Fang and Tao, 2014] Meng Fang and Dacheng Tao. Networked bandits with disjoint linear payoffs. In *SIGKDD*, pages 1106–1115, 2014.

[Guha and Khuller, 1998] Sudipto Guha and Samir Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.

[Horowitz and Kamvar, 2010] Damon Horowitz and Sepandar D Kamvar. The anatomy of a large-scale social search engine. In *WWW*, pages 431–440, 2010.

[Jeong *et al.*, 2013] Jin-Woo Jeong, Meredith Ringel Morris, Jaime Teevan, and Daniel J Liebling. A crowd-powered socially embedded search engine. In *ICWSM*, 2013.

[Kleinberg and Raghavan, 2005] J. Kleinberg and P. Raghavan. Query incentive networks. In *FOCS*, 2005.

[Kleinberg, 2000] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *STOC*, 2000.

[Kleinberg, 2002] Jon Kleinberg. Small-world phenomena and the dynamics of information. In *NIPS*, 2002.

[Kota and Narahari, 2010] Nagaraj Kota and Y Narahari. Threshold behavior of incentives in social networks. In *CIKM*, pages 1461–1464, 2010.

[Krause and Golovin, 2012] Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3:19, 2012.

[Krause and Guestrin, 2011] Andreas Krause and Carlos Guestrin. Submodularity and its applications in optimized information gathering. *TIST*, 2(4):32, 2011.

[Lai and Robbins, 1985] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[Leskovec and Horvitz, 2014] J. Leskovec and E. Horvitz. Geospatial structure of a planetary-scale social network. *IEEE Transactions on Computational Social Systems*, 2014.

[Milgram, 1967] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[Nemhauser *et al.*, 1978] G.L. Nemhauser, L.A. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Math. Prog.*, 1978.

[Nushi *et al.*, 2015] Besmira Nushi, Omar Alonso, Martin Hentschel, and Vasileios Kandylas. Crowdstar: A social task routing framework for online communities. In *ICWE*, 2015.

[Richardson and White, 2011] Matthew Richardson and Ryen W. White. Supporting synchronous social Q&A throughout the question lifecycle. In *WWW*, 2011.

[Singla *et al.*, 2015] Adish Singla, Eric Horvitz, Pushmeet Kohli, Ryen W. White, and Andreas Krause. Information gathering in networks via active exploration (extended version). `http://arxiv.org/abs/1504.06423`, 2015.

[White and Singla, 2011] Ryen W. White and Adish Singla. Finding our way on the web: Exploring the role of waypoints in search interaction. In *WWW*, 2011.

[Xu *et al.*, 2010] Kuang Xu, Jing Xie, and Victor OK Li. Locating experts via online social networks. In *IEEE International Conference on Communications*, pages 1–5, 2010.

[Zhang *et al.*, 2012] Haoqi Zhang, Eric Horvitz, Yiling Chen, and David C Parkes. Task routing for prediction tasks. In *AAMAS*, pages 889–896, 2012.