2021 IEEE 37th International Conference on Data Engineering (ICDE)

Social Visibility Optimization in OSNs with Anonymity Guarantees: Modeling, Algorithms and Applications

Shiyuan Zheng
Dept. of Computer Science & Engineering
The Chinese University of Hong Kong
syzheng@cse.cuhk.edu.hk

Hong Xie
College of Computer Science
Chongqing University
xiehong2018@cqu.edu.cn

John C.S. Lui
Dept. of Computer Science & Engineering
The Chinese University of Hong Kong
cslui@cse.cuhk.edu.hk

2021 IEEE 37th International Conference on Data Engineering (ICDE) | 978-1-7281-9184-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICDE51399.2021.00201

***Abstract*—Online social network (OSN) is an ideal venue to enhance one's visibility. This paper considers how a user (called requester) in an OSN selects a small number of available users and invites them as new friends/followers so as to maximize his "*social visibility*". More importantly, the requester has to do this under the anonymity setting, which means he is not allowed to know the neighborhood information of these available users in the OSN. In this paper, we first develop a mathematical model to quantify the social visibility and formulate the problem of visibility maximization with anonymity guarantee, abbreviated as "VisMAX-A". Then we design an algorithmic framework named as "AdaExp", which adaptively expands the requester's visibility in multiple rounds. In each round of the expansion, AdaExp uses a query oracle with anonymity guarantee to select only one available user. By using probabilistic data structures like the k-minimum values (KMV) sketch, we design an efficient query oracle with anonymity guarantees. We also conduct experiments on real-world social networks and validate the effectiveness of our algorithms.**

***Index Terms*—Social visibility, social networks analysis, KMV sketch, approximation algorithms**

I. Introduction

Online social network (OSN) is an ideal venue for one to share information and gain attention. There are numerous examples for such claim. In social networking sites like Facebook, users share their opinions, status and likes/dislikes to their friends via the friendship network. In video sharing sites like YouTube, users share their videos to their subscribers via the subscriber network. In an OSN, users with more direct attention e.g., friends, subscribers, followers, etc., can make their contents e.g., opinions, videos, photos, etc. visible to more users in the network, and this may bring higher commercial benefit to these content creators. For example, in the YouTube OSN, generally a user who has more subscribers can get a larger amount of views on the video he published. Note that the viewers not only can be the subscribers, but can also be those who have not subscribed but come across the video via ways like word-of-mouth spreading. Moreover, more views may lead to more advertising exposures, which means a larger reward from YouTube. Informally, we say that those users who have more direct or indirect attention are more *socially visible*, in other words, they have a larger *social visibility*.

One effective way to increase a user's social visibility is to get more direct attention, for instance, attracting new friends, new subscribers, new followers, etc. However, not all the users in an OSN are willing to establish connection with others. Usually, there is a constraint on the scope of users a requester can select from and on the number of users a requester can build new connection with. This implies that the requester needs to judiciously select the targets to establish new connections, so as to maximize the effect of visibility boosting. To illustrate, consider the following examples.

Example 1. *Consider a simple social network with directed edge as illustrated in Fig. 1. We say user v is an incoming neighbor of user u if there is a directed edge from v to u, representing u gets a direct attention from v (e.g., v follows or subscribes u). In this example, we assume that a user is only socially visible to his 1-hop and 2-hop incoming neighbors. For example, in Fig. 1, user 1 is socially visible to users in the set $\{2, 3, 4, 5, 9\}$, which we call the "2-visible set" of user 1. Suppose only a subset of users are willing to establish new connection as requested, named as "available users". Let them be $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ here. Then, suppose user 1 is the requester, who has a quota of adding only one new incoming neighbor to increase his social visibility. After trying all the users who are available and non-trivial, i.e., $\{6, 7, 8, 9, 10, 11\}$, he finds that by adding user 8 as his new incoming neighbor, he can maximize the increase of his 2-visible set.*

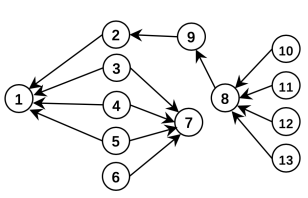


Fig. 1. A directed OSN.

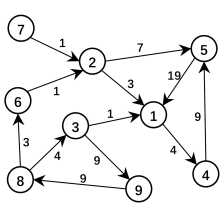


Fig. 2. An OSN model.

978-1-7281-9184-3/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDE51399.2021.00201

Example 1 illustrates that a requester can significantly increase his social visibility by simply adding just one new incoming neighbor. However, the number of available users (n) in a real-world OSN can be billions, and a user may have a quota ($m > 1$) of adding new incoming neighbors, so the number of possible solutions is around $\binom{n}{m}$, which makes it computationally expensive to find the optimal new incoming neighbor set. What makes the problem more challenging and interesting is that, due to privacy constraints, the OSN network topology may not be available to the requester, and other users may not even want to disclose the information of members in their visible sets. To illustrate such anonymity setting, consider the following example.

Example 2. *Consider the same setting as Example 1, but for the following conditions on anonymity: (1) Each user only knows his own visible set but does not know the remaining part of the OSN; (2) The requester is not allowed to know the membership of any other user's visible set. The objective of the requester is still to select the optimal new incoming neighbor set while satisfying the anonymity requirement.*

Example 1-2 illustrate the problem of increasing a requester's social visibility by adding new incoming neighbors, as well as the underlying computational challenge and "anonymity guarantee" challenge in selecting the optimal set which can maximize the increase. Note that the social visibility problem is relevant in the real-world and our solution can be used to provide customized visibility boosting service for OSN users. Specifically, it helps requesters who want to increase their visibility to target the most profitable candidates, and avoid wasting time or money on those who can only bring a small increase on their social visibility. Thus, our solution can be applied to many fields, such as self-marketing, advertising service and so on. Motivated by this, we aim to answer the following questions: *(1) How to formulate a mathematical model to quantify social visibility? (2) How to develop computationally efficient algorithms to maximize social visibility of a requester? (3) How to provide anonymity guarantee for our social visibility maximization algorithm?* We answer these questions. Our contributions are:

- We propose a mathematical model to quantify the social visibility and formulate the problem of visibility maximization with anonymity requirement (VisMAX-A).
- We design an algorithmic framework (AdaExp), which adaptively expands the requester's visible set in multiple rounds. In each round of expansion, a *query oracle* returns an estimation of the "best" new incoming neighbor with a guaranteed accuracy.
- We design a query oracle using the KMV sketch technique [1], and prove that our query oracle satisfies the desired accuracy properties.
- We conduct experiments on real-world social network datasets, and the results validate the effectiveness of our framework.

II. Model & Problem Formulation

In this section, we first develop a mathematical model to quantify the social visibility. Then we formulate the problem of visibility maximization with anonymity guarantee.

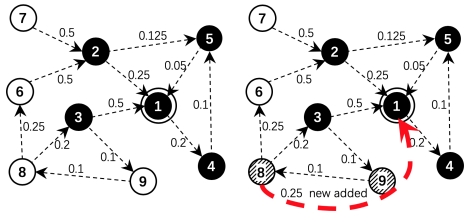
A. The Online Social Network Model

Consider an OSN which is characterized by a weighted directed graph $\mathcal{G} \triangleq (\mathcal{U}, \mathbf{W})$, where $\mathcal{U} = \{1, \ldots, U\}$ denotes a finite set of $U \in \mathbb{N}_+$ users and $\mathbf{W} \triangleq [w_{v\to u} : v, u \in \mathcal{U}] \in \mathbb{R}_+^{U\times U}$ represents a summarization of the weights of edges. We use $w_{v\to u} = 0$ to model that there is no directed edge from v to u. The graph $\mathcal{G}$ does not contain self-loop edges, i.e., $w_{u\to u} = 0, \forall u \in \mathcal{U}$. The weight $w_{v\to u}$ quantifies the *influence strength* of user u over user v. For example, as shown in Fig. 2, in a Twitter-like OSN, a directed edge from v to u can be interpreted as v follows u, and the weight $w_{v\to u}$ corresponds to the frequency that v comments, likes or retweets the tweet posted by u.

B. The Social Visibility Model

Definition 1 (incoming neighbor set). *A user v is an incoming neighbor of user u, if there is a directed edge from v to u. The incoming neighbor set of user u, denoted by $\mathcal{N}(u)$, is the set of all the incoming neighbors of user u, defined as $\mathcal{N}(u) \triangleq \{v | w_{v\to u} > 0, v \in \mathcal{U}\}$.*

We use *"requester"* to refer to the user in the OSN who aims to increase his visibility by requesting others to be his new incoming neighbors, and use *"available users"* to refer to the users in the OSN who are willing to be a new incoming neighbor of the requester if selected and requested. We say available users are available to the requester.



(a) $\mathcal{V}(1, 0.6)$. (b) New $\mathcal{V}(1, 0.6)$.

Fig. 3. The change of 0.6-visible set of user 1.

Definition 2 (visibility distance). *Visibility distance, denoted by $d_{v\to u}$, is a measure to quantify the degree of difficulty in which a user influences his incoming neighbor, which is defined as:*

$$d_{v\to u} = \begin{cases} \infty, & \text{if } v \notin \mathcal{N}(u), \\ D(w_{v\to u}), & \text{otherwise}, \end{cases}$$

where $D : \mathbb{R}_+ \to \mathbb{R}_+$ denotes a distance mapping function, from influence strength (i.e., weight) to visibility distance.

A larger $d_{v\to u}$ implies a larger cost/difficulty of spreading content from node u to v. For example, one possible form of

D can be $D(w_{v\to u}) = \frac{1}{1+w_{v\to u}}, v \in \mathcal{N}(u)$. Fig. 3(a) shows the visibility distances of user pairs in Fig. 2, where each real number associated with the dashed edge from v to u represents the visibility distance calculated by $1/(1+w_{v\to u})$. The following assumption states the family of visibility distance function we can have.

Assumption 1. *The visibility distance $D(w_{v\to u})$ decreases in influence strength $w_{v\to u}$, where $v \in \mathcal{N}(u)$.*

Assumption 1 captures that u has a smaller visibility distance from the incoming neighbor v, if his influence strength $w_{v\to u}$ over v is stronger, which means being visible to v is easier.

Let $\vec{p} \triangleq (x_0 \to x_1 \to ... \to x_n)$ denote a directed path in graph $\mathcal{G}$. Let $\mathcal{E}(\vec{p})$ denote the set of all directed edges on path $\vec{p}$, i.e., $\mathcal{E}(\vec{p}) = \{(x_0, x_1), (x_1, x_2), ..., (x_{n-1}, x_n)\}$. We define the visibility distance length $L(\vec{p})$ of the path $\vec{p}$ to be the sum of the visibility distances of all the edges on path $\vec{p}$, i.e., $L(\vec{p}) \triangleq \sum_{(v,u)\in\mathcal{E}(\vec{p})} d_{v\to u}$. For example, consider a path $\vec{p} = (7 \to 2 \to 1)$ in Fig. 3(a). Its visibility distance length can be calculated as $L(\vec{p}) = d_{7\to 2} + d_{2\to 1} = \frac{1}{1+w_{7\to 2}} + \frac{1}{1+w_{2\to 1}} = 0.75$. Let $\mathcal{P}_{v\to u}$ denote the set of all directed paths (without circles) from user v to user u. Base on $\mathcal{P}_{v\to u}$, we define the concept of *τ-visible*, *τ-visible set* and *τ-visibility*.

Definition 3 (visible threshold τ, τ-visible set, τ-visibility). *Let τ denote the visible threshold, $\tau \in \mathbb{R}_+$. User u is τ-visible to user v, if $\min_{\vec{p}\in\mathcal{P}_{v\to u}} L(\vec{p}) \le \tau$. The τ-visible set of user u is the set of all users to whom u is τ-visible, denoted as*

$$\mathcal{V}(u,\tau) \triangleq \left\{ v \middle| v \in \mathcal{U}, \min_{\vec{p}\in\mathcal{P}_{v\to u}} L(\vec{p}) \le \tau \right\}.$$

Lastly, the τ-visibility of user u is the cardinality of user u's τ-visible set.

Namely, user u is τ-visible to user v, if there exists at least one directed path from v to u with the sum of the visibility distances of all the edges on path less than or equal to τ. For example, Fig. 3(a) shows that user 1 is 0.6-visible to user 2, but not user 6. The 0.6-visible set of user 1 is $\mathcal{V}(1, 0.6) = \{2, 3, 4, 5\}$, and thus his 0.6-visibility is 4.

C. The Visibility Maximization Problem

To increase the visibility of a requester r, we can add some new incoming neighbors (new followers, new subscribers, etc.) to r. Formally, we assume that r is given a quota of adding $m \in \mathbb{N}_+$ new incoming neighbors to increase his visibility, and each one of these m incoming edges to r has a default weight $\bar{w} \in \mathbb{R}_+$ representing the default influence strength of a new added link. Let $\mathcal{U}'$ denote the set of available users. Let $\mathcal{M} \subseteq \mathcal{U}'$ denote a set of users selected from $\mathcal{U}'$ with cardinality $|\mathcal{M}| \le m$, and no user in $\mathcal{M}$ is an incoming neighbor of the requester r, i.e., $\mathcal{M} \cap \mathcal{N}(r) = \emptyset$. For the purpose of simplifying presentation, we use $\boldsymbol{\theta}$ to denote the vector of given model parameters (the requester, default weight and visible threshold), i.e., $\boldsymbol{\theta} \triangleq [r, \bar{w}, \tau]$. Lastly, we use $\Delta_{\boldsymbol{\theta}}(\mathcal{M})$ to denote the increase of r's τ-visibility after adding directed edges with default weight $\bar{w}$ from users in $\mathcal{M}$ to r, Fig. 3 illustrates the change of visible set after adding a new incoming neighbor. As shown in Fig. 3(b), when $\mathcal{M} = \{8\}$, $\bar{w} = 3$, there is a new directed edge from user 8 to the requester user 1 with influence strength $w_{8\to 1} = \bar{w}$ and visibility distance $d_{8\to 1} = \frac{1}{1+w_{8\to 1}} = 0.25$. It can be interpreted as user 8 starts to follow user 1. We can find that the expansion of user 1's 0.6-visible set is $\{8, 9\}$ and thus the increase of his 0.6-visibility is $\Delta_{\boldsymbol{\theta}}(\mathcal{M}) = |\{8, 9\}| = 2$.

The objective is to judiciously select the set $\mathcal{M} \subseteq \mathcal{U}'$ so as to maximize the increase in the requester's τ-visibility under his *local information* (defined in Definition 4) while preserving *query anonymity* (defined in Definition 5). Note that the requester does not know the whole graph $\mathcal{G}$, but instead, he only has his local information, which is defined as follows.

Definition 4 (local information). *The local information of a user u is defined as:*

1) *the identities and the visibility distance lengths of members in user u's τ-visible set;*
2) *the edges started from or ended at u;*
3) *upon establishing a link with a new incoming neighbor, the updated identities and visibility distance lengths of members in user u's new τ-visible set.*

In practice, due to privacy concerns, a user may not want to disclose the information (e.g., IDs, names, etc.) about members in his τ-visible set. For example, one may hide the viewers of their blogs. To this end, we define the notion of query anonymity.

Definition 5 (query anonymity). *Query anonymity is satisfied when queries about members of any user's τ-visible set at any visible threshold τ is not allowed.*

With the notion of local information and query anonymity defined above, we formulate our problem as follows.

Problem 1 (visibility maximization (VisMAX-A)). *Suppose each user only has his local information defined in Definition 4. Given the graph $\mathcal{G}$, the requester r, the set $\mathcal{U}'$ of available users, the visible threshold τ, the default weight of the new edge $\bar{w}$, and the new incoming neighbor quota m, select a set of users $\mathcal{M}$ as new incoming neighbors of r so to maximize the increase of τ-visibility of r with query anonymity guarantee:*

$$\begin{aligned} &\underset{\mathcal{M}}{\textit{maximize}} && \Delta_{\boldsymbol{\theta}}(\mathcal{M}) \\ &s.t. && |\mathcal{M}| \le m; \mathcal{M} \cap \mathcal{N}(r) = \emptyset; \mathcal{M} \subseteq \mathcal{U}'. \end{aligned}$$

where $\boldsymbol{\theta} = [r, \bar{w}, \tau]$ is the vector of given model parameters.

III. Algorithmic Framework

In this section we present an algorithmic framework to solve VisMAX-A problem based on a *query oracle*. We also analyze how the performance of query oracle will influence the theoretical guarantee. *Due to page limit, the proofs of lemmas, theorems and corollaries are presented in our technical report [2].*

A. Design of the Algorithmic Framework

We first formally define candidate and candidate set. Given a set of users $\mathcal{S}$ who have been newly added as incoming neighbors of the requester r, a user x is a *candidate* if he is (1) $x \in \mathcal{U}'$, (2) $x \notin \mathcal{N}(r) \cup \mathcal{S}$ and, (3) $x \neq r$. We also define the set of all the candidates as *candidate set*, denoted as $\mathcal{P}(\mathcal{S}) \triangleq \mathcal{U}' \setminus (\{r\} \cup \mathcal{N}(r) \cup \mathcal{S})$. Next, we define, analyze and characterize the *marginal gain*.

Definition 6 (marginal gain). *Given a set $\mathcal{S}$ of users who have been newly added as incoming neighbors of the requester r. The marginal gain of adding a candidate $x, x \in \mathcal{P}(\mathcal{S})$ is defined as $\delta(x, \mathcal{S}) \triangleq \Delta_{\boldsymbol{\theta}}(\mathcal{S} \cup \{x\}) - \Delta_{\boldsymbol{\theta}}(\mathcal{S})$.*

Lemma 1. *Given a set $\mathcal{S}$ of users who have been newly added as incoming neighbors of the requester r. The marginal gain of adding a candidate $x, x \in \mathcal{P}(\mathcal{S})$ can be derived as*

$$\delta(x,\mathcal{S}) = \left| \left[\mathcal{V}(x, \tau - D(\bar{w})) \setminus \mathcal{V}(r, \tau)\right] \setminus \left[\cup_{v \in \mathcal{S}} \mathcal{V}(v, \tau - D(\bar{w}))\right] \right|$$

Remark: From Lemma 1, we can observe that the marginal gain of adding a candidate x can be computed from the users' local information defined in Definition 4. Based on the definition of marginal gain, we define the best candidate.

Definition 7 (the best candidate $v^*(\mathcal{S})$). *Given a set $\mathcal{S}$ of users who have been added, we define the best candidate for current $\mathcal{S}$ as $v^*(\mathcal{S}) \in \arg\max_{x \in \mathcal{P}(\mathcal{S})} \delta(x, \mathcal{S})$.*

We denote the marginal gain associated with the best candidate $v^*(\mathcal{S})$ by $\delta^*(\mathcal{S}) \triangleq \max_{x \in \mathcal{P}(\mathcal{S})} \delta(x, \mathcal{S})$. In the following we define an oracle to query about the best candidate.

Definition 8 (query oracle). *A query oracle denoted by $\texttt{QueryOracle}(\epsilon, \mathcal{S})$ is a function which outputs a randomized estimation of the best candidate which satisfies*

$$\mathbb{E}[\delta^*(\mathcal{S}) - \delta(\hat{v}^*, \mathcal{S})] \leq \epsilon,$$

where $\hat{v}^* = \texttt{QueryOracle}(\epsilon, \mathcal{S})$ denotes the output of the oracle.

Algorithm 1 outlines our framework. The key idea is doing adaptive expansion on visible set, i.e., sequentially selecting and connecting with m users. In each round, only one candidate is added as a new incoming neighbor, and this candidate is directly decided by query oracle $\texttt{QueryOracle}(\epsilon, \mathcal{S})$. Combined with Definition 8, we know that this candidate is the best candidate estimated by the query oracle $\texttt{QueryOracle}(\epsilon, \mathcal{S})$.

Algorithm 1 Adaptive Expansion for VisMAX-A (AdaExp)

1: **Input:** the requester r, new incoming neighbor quota m
2: **Output:** new incoming neighbors set $\mathcal{M}_{AE}$
3: $\mathcal{S} \leftarrow \emptyset$
4: **while** $|\mathcal{S}| < m$ **do**
5: $\quad \hat{v}^* \leftarrow \texttt{QueryOracle}(\epsilon, \mathcal{S})$
6: $\quad \mathcal{S} \leftarrow \mathcal{S} \cup \{\hat{v}^*\},\ \mathcal{N}(r) \leftarrow \mathcal{N}(r) \cup \{\hat{v}^*\}$
7: **end while**
8: **return** $\mathcal{M}_{AE} \leftarrow \mathcal{S}$

B. Theoretical Guarantee

The "performance gap" of Algorithm 1 (AdaExp) arises from two parts. One is from the adaptive expansion and the other one is from the randomness caused by the query oracle. The following theorem presents the theoretical guarantee for Algorithm 1 (AdaExp).

Theorem 1. *Let $\mathcal{M}_{AE}$ denote the output of the Algorithm 1. Then we have*

$$\mathbb{E}\left[\Delta_{\boldsymbol{\theta}}(\mathcal{M}_{AE})\right] \geq \left(1 - \frac{1}{e}\right) \Delta_{\boldsymbol{\theta}}(\mathcal{M}^*) - m\epsilon,$$

where $\mathcal{M}^*$ denotes the optimal solution of VisMAX-A via exhaustive search.

Remark: Theorem 1 states that the approximation ratio decreases as the bound of expected error ϵ of the query oracle increases. Our solution can achieve a high theoretical guarantee when ϵ is small. To analyze the "performance gap" caused by the adaptive selection, we prove the monotonicity and submodularity of the objective function $\Delta_{\boldsymbol{\theta}}(\mathcal{M})$ in our technical report [2]. We next design a framework to implement the query oracle.

IV. Query Oracle Design

In this section, we design an algorithm to implement the query oracle.

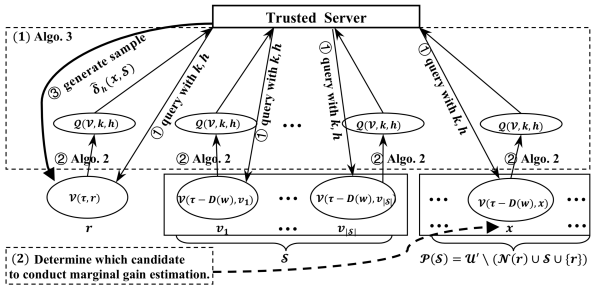


Fig. 4. The framework to estimate $v^*(\mathcal{S})$ with anonymity guarantee.

The key idea is to design a framework to *estimate* the best candidate $v^*(\mathcal{S})$ with query anonymity guarantee and error bound guarantee, so that we can implement the query oracle and make the AdaExp algorithm feasible. We consider the existence of a *trusted server*. Fig. 4 (1) outlines the framework of generating one *sample* of marginal gain $\delta(x, \mathcal{S})$ if user x is connected to requester r as a new incoming neighbor. First, the trusted server uses a probabilistic data structure to query involved users about their visible sets while preserving query anonymity. Then those queried users return their results which would not reveal their local information. After that, the trusted server uses these *query outcomes* to produce one *unbiased sample* $\hat{\delta}(x, \mathcal{S})$ on marginal gain $\delta(x, \mathcal{S})$.

First, we introduce how to query one user with query anonymity guarantee. We use the data structure KMV sketch introduced in [1]. For simplicity of presentation, let $\tilde{\mathcal{V}}$ denote

Algorithm 2 Response query with KMV sketch.

Input: hash function h, sketch size k, visible set $\tilde{\mathcal{V}}$.
Output: the KMV sketch $\mathcal{Q}(\tilde{\mathcal{V}}, k, h)$.
Notation: $max(\mathcal{K})$ // return the largest value in $\mathcal{K}$
$\mathcal{K} \leftarrow \emptyset$
for each $v \in \tilde{\mathcal{V}}$ **do**
  $val = h(v)$
  if $|\mathcal{K}| < k$ **then**
    insert val into $\mathcal{K}$
  end if
  if $|\mathcal{K}| \geq k$ and $val < max(\mathcal{K})$ **then**
    remove $max(\mathcal{K})$
    insert val into $\mathcal{K}$
  end if
end for
return $\mathcal{Q}(\tilde{\mathcal{V}}, k, h) \leftarrow \mathcal{K}$

Algorithm 3 Generate a sample of the marginal gain $\delta(x, \mathcal{S})$.

Input: hash function h, sketch size k, users $\{x, r\} \cup \mathcal{S}$.
Output: estimated marginal gain $\hat{\delta}_h(x, \mathcal{S})$.
Apply Algo. 2 to generate the following query outcomes
  $\mathcal{Q}(\mathcal{V}(v, \tau - D(\bar{w})), k, h), \forall v \in \{x\} \cup \mathcal{S}$,
  $\mathcal{Q}(\mathcal{V}(r, \tau), k, h)$.
$\mathcal{F} \leftarrow [\mathcal{Q}(\mathcal{V}(x, \tau - D(\bar{w})), k, h) \setminus \mathcal{Q}(\mathcal{V}(r, \tau), k, h)]$
  $\setminus \cup_{v \in \mathcal{S}} \mathcal{Q}(\mathcal{V}(v, \tau - D(\bar{w})), k, h)$
$q \leftarrow$ the k-th smallest value in
  $\cup_{v \in S \cup \{x\}} \mathcal{Q}(\mathcal{V}(v, \tau - D(\bar{w})), k, h) \cup \mathcal{Q}(\mathcal{V}(r, \tau), k, h)$
return $\hat{\delta}_h(x, \mathcal{S}) \leftarrow \frac{|\mathcal{F}|}{k} \frac{k-1}{q}$

the visible set that the trusted server queries for. Algorithm 2 outlines how the queried user generates the outcome, i.e., a KMV sketch, which is denoted by $\mathcal{Q}(\tilde{\mathcal{V}}, k, h)$. The one-way hash function h and sketch size k are given by the trusted server, where h is used to map the ID of each element $v \in \tilde{\mathcal{V}}$ (the ID of the user lies in $[U]$) into a real number in $[0, 1)$. In this work, we assume we are given a family of hash functions $\mathcal{H}$, such that $\{[h(x), x \in \mathcal{U}], h \in \mathcal{H}\} = [0, 1)^{|\mathcal{U}|}$.

Next, we introduce how the trusted server generates unbiased samples on the marginal gain function $\delta(x, \mathcal{S})$ using the query outcomes. Before that, we give the following theorem which shows a good property of the KMV sketch.

Theorem 2. *Suppose the trusted server queries n involved users for wanted visible sets $\tilde{\mathcal{V}}_i, \forall i \in [n]$, and receives query outcomes $\mathcal{Q}(\tilde{\mathcal{V}}_i, k, h), \forall i \in [n]$ (i.e., the KMV sketches). Then we have*

$$\mathbb{E}_{h \sim Uniform(\mathcal{H})} \left[\frac{|\oplus(\mathcal{Q}(\tilde{\mathcal{V}}_1, k, h), \ldots, \mathcal{Q}(\tilde{\mathcal{V}}_n, k, h))|}{k} \frac{k-1}{q} \right] = |\oplus(\tilde{\mathcal{V}}_1, \ldots, \tilde{\mathcal{V}}_n)|, \quad (1)$$

where the function $\oplus(\cdot, \ldots, \cdot)$ denotes the result of a sequence of set operations over its parameters, and q is the k-th smallest value in the union of queries outcomes, i.e., $\cup_{i \in [n]} \mathcal{Q}(\tilde{\mathcal{V}}_i, k, h)$.

Theorem 2 states that the KMV sketch can be applied to produce an unbiased estimator on the cardinality of the operations (i.e., union, intersection, etc.) of multiple sets. Recall that Lemma 1 provides a closed form for the marginal gain function $\delta(x, \mathcal{S})$, which is the cardinality of the set obtained by set operations over several involved sets. With this observation and Theorem 2, we design Algorithm 3 and have the following corollary.

Corollary 1. *The sample $\hat{\delta}_h(x, \mathcal{S})$ generated by Algorithm 3 is unbiased, i.e, $\mathbb{E}_{h \sim Uniform(\mathcal{H})} \left[\hat{\delta}_h(x, \mathcal{S}) \right] = \delta(x, \mathcal{S})$.*

As shown in Algorithm 3, the trusted server generates an unbiased sample $\hat{\delta}_h(x, \mathcal{S})$ using the query outcomes returned by involved users. Finally, the trusted server sends the sample $\hat{\delta}_h(x, \mathcal{S})$ to the requester r, and r may refer to sampling history to select the next candidate x using certain strategies, e.g., multi-armed bandits (MAB) strategy introduced in [3]. Note that the unbiasedness of the sample $\hat{\delta}_h(x, \mathcal{S})$ generated by Algorithm 3 implies that one can estimate $\delta(x, \mathcal{S})$ accurately by generating a sufficiently large number of IID samples $\hat{\delta}_h(x, \mathcal{S})$ with different hash functions h. Also note that the sample $\hat{\delta}_h(x, \mathcal{S})$ preserves the anonymity as it is only a real number.

V. Performance Evaluation

In this section, we conduct experiments on real-world datasets to evaluate the performance of our algorithm.

A. Experimental Settings

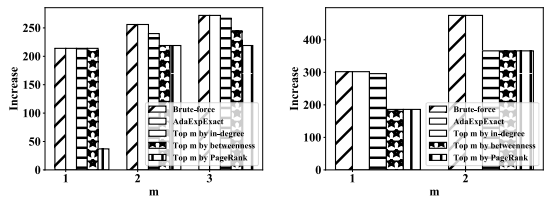
We use the following datasets for evaluation.

- **Blogs [4]:** it contains front-page hyperlinks between blogs in the context of 2004 US election, with 1,224 nodes and 19,025 directed and unweighted edges. Nodes correspond to blogs and edges correspond to hyperlinks between blogs.
- **DBLP [5]:** it is a co-author network extracted from the DBLP Bibliography, with 10,000 nodes and 55,734 undirected and weighted edges. Nodes correspond to scholars who have published papers in major conferences and edges correspond to co-author relationships between scholars. The weights of edges indicate the number of cooperations between two scholars.

For all above OSNs, we consider the most computationally challenging case where all users are available, i.e., $\mathcal{U}' = \mathcal{U}$. In the experiments, we use MAB strategy (best arm version) introduced in [3] as the requester's strategy to select the next candidate to do sampling, which is indicated in Fig. 4 (2). The introduction to MAB and the implementation details are in our technical report [2].

B. Experimental Results

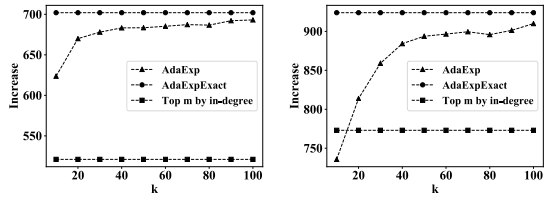
Evaluate AdaExpExact. We assume the case that the requester is able to get the exact answer via the query oracle (i.e., $\epsilon = 0$ in Definition 8), which can also be regarded as a baseline where query anonymity is not required. We name it as "AdaExpExact". We compare it with the brute-force method and heuristic methods such as picking m candidates with the largest in-degree centrality, betweenness centrality



(a) *Blogs*, $\tau = 3, D(\bar{w}) = 1$ (b) *DBLP*, $\tau = 0.7, D(\bar{w}) = 0.3$

Fig. 5. Comparison of AdaExpExact, brute-force and heuristic methods.

and PageRank value. Fig. 5 shows that AdaExpExact achieves the same increase in visibility as the brute-force method for all the datasets and for all values of m, which validates the error arises from the "adaptive" part (mentioned in III-B) of AdaExp is very small. In addition, our algorithm outperforms the heuristic methods.



(a) The impact of k ($m = 5$) (b) The impact of k ($m = 10$)

Fig. 6. The impact of k on AdaExp. (DBLP)

Impact of sketch size k. We consider $m = 5$ and $m = 10$, respectively. We vary the sketch size k from 10 to 100 to study its impact on the performance of AdaExp. Fig. 6 shows the visibility increase achieved by AdaExp, AdaExpExact and heuristic method based on in-degree centrality. Fig. 6(a) shows that as the sketch size k increases, AdaExp achieves a larger visibility increase. When the sketch size is around 50, AdaExp achieves a visibility increase close to AdaExpExact and starts to grow very slowly. This is important because it implies that AdaExp only requires a small sketch size to generate a good estimation. AdaExp almost always outperforms heuristic methods unless the sketch size k is very small. Fig. 6(b) further validates this observation when $m = 10$.

VI. Related Work

From an application perspective, our work is related to friendship recommendation [6]–[9], link prediction [10]–[12] and influence maximization [13]–[15]. However, different from all above, our work is built on a social visibility model, and thus the approaches to solving those problems can not be applied to the VisMAX-A problem. Moreover, our framework provides anonymity guarantee, while those problems do not. From a methodology perspective, the KMV sketch technique has been widely used to estimate the cardinality of record size [16], [17]. In [18], Cohen *et al.* introduce a new estimator for the size of sets intersection based on the MinHash sketch technique. A general unbiased estimation over a sequence of set operations is proposed in [1]. In our framework, we use the KMV sketch to design queries while preserving query anonymity.

VII. Acknowledgments

The work of John C.S. Lui is supported in part by the GRF 14200420. The work of Hong Xie is supported by the Fundamental Research Funds for the Central Universities (2020CDJ-LHZZ-057). Hong Xie is the corresponding author.

VIII. Conclusions

In this paper, we develop a mathematical model to quantify social visibility and formulate the social visibility maximization problem (VisMAX-A). Based on a query oracle, we develop a computationally efficient algorithm AdaExp to address the VisMAX-A problem with an approximation ratio slightly lower than $(1 - 1/e)$. Then, we propose a query oracle to estimate the best candidate in each iteration of AdaExp with anonymity guarantee and error bound guarantee. Finally, we conduct experiments on real-world social network datasets to validate the effectiveness of our framework.

References

[1] K. Beyer, P. J. Haas, B. Reinwald, Y. Sismanis, and R. Gemulla, "On synopses for distinct-value estimation under multiset operations," in *Proc. ACM SIGMOD*. ACM, 2007.

[2] S. Zheng, H. Xie, and J. C. Lui, *Technical report*, 2020, www.dropbox.com/sh/bmk98a0x8co6tbx/AAAzRcv4aAldUl3FlAOSlBcSa.

[3] J.-Y. Audibert and S. Bubeck, "Best arm identification in multi-armed bandits," 2010.

[4] J. Kunegis, "KONECT – The Koblenz Network Collection," in *Proc. Int. Conf. on World Wide Web Companion*, 2013, pp. 1343–1350.

[5] W. Nawaz, K.-U. Khan, Y.-K. Lee, and S. Lee, "Intra graph clustering using collaborative similarity measure," *Distributed and Parallel Databases*, vol. 33, no. 4, pp. 583–603, 2015.

[6] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–59, 1997.

[7] H. Ma, "On measuring social friend interest similarities in recommender systems," in *Proc. ACM SIGIR*. ACM, 2014.

[8] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. WSDM*. ACM, 2011.

[9] X. Xie, "Potential friend recommendation in online social network," in *Proc. IEEE/ACM CGCC*. IEEE Computer Society, 2010.

[10] L. L and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, p. 11501170, Mar 2011.

[11] V. Martínez, F. Berzal, and J.-C. Cubero, "A survey of link prediction in complex networks," *ACM CSUR*, vol. 49, no. 4, p. 69, 2017.

[12] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[13] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. ACM KDD*, 2003.

[14] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. ACM KDD*. ACM, 2001.

[15] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proc. ACM KDD*. ACM, 2002.

[16] G. Cormode, M. Garofalakis, P. J. Haas, C. Jermaine *et al.*, "Synopses for massive data: Samples, histograms, wavelets, sketches," *Foundations and Trends® in Databases*, vol. 4, no. 1–3, pp. 1–294, 2011.

[17] X. Wang, Y. Zhang, W. Zhang, X. Lin, and W. Wang, "Selectivity estimation on streaming spatio-textual data using local correlations," *Proc. VLDB*, 2014.

[18] R. Cohen, L. Katzir, and A. Yehezkel, "A minimal variance estimator for the cardinality of big data set intersection," in *ACM KDD*, 2017.