Sampling Representative Users from Large Social Networks

Jie Tang[†♯], Chenhui Zhang[†♯], Keke Cai[‡], Li Zhang[‡], Zhong Su[‡]
[†]Department of Computer Science and Technology, Tsinghua University
[♯]Tsinghua National Laboratory for Information Science and Technology (TNList)
[‡]IBM, China Research Lab
jietang@tsinghua.edu.cn, zh.sherlock@gmail.com, {caikeke, lizhang, suzhong}@cn.ibm.com

Abstract

Finding a subset of users to statistically represent the original social network is a fundamental issue in Social Network Analysis (SNA). The problem has not been extensively studied in existing literature. In this paper, we present a formal definition of the problem of **sampling representative users** from social network. We propose two sampling models and theoretically prove their NP-hardness. To efficiently solve the two models, we present an efficient algorithm with provable approximation guarantees. Experimental results on two datasets show that the proposed models for sampling representative users significantly outperform (+6%-23% in terms of Precision@100) several alternative methods using authority or structure information only. The proposed algorithms are also effective in terms of time complexity. Only a few seconds are needed to sampling 300 representative users from a network of 100,000 users. All data and codes are publicly available.[1]

Introduction

In social networks, a small subset of users (e.g., opinion leaders) usually plays an important role in influencing the social dynamics (behavior and structure). For example, in HCI, for conducting surveys and collecting user feedbacks, selecting representative users is always a key issue (Landauer 1997). On the other hand, in a social network, users may have different *representative degrees* from different aspects (topics). For example, iPhone fans may have a high representative degree on some fashion products, while housewives have a high representative degree on cooking or children's products. One interesting question is: can we design an algorithm to automatically sampling representative users for different topics from large social networks?

Despite several relevant studies, such as those on social influence analysis (Anagnostopoulos, Kumar, and Mahdian 2008; Crandall et al. 2008; Singla and Richardson 2008; Tang et al. 2009), influence maximization (Kempe, Kleinberg, and Tardos 2003; Chen, Wang, and Yang 2009; Scripps, Tan, and Esfahanian 2009), and opinion leader finding (Goyal, Bonchi, and Lakshmanan 2008), there are few theoretical studies on the problem of sampling representative users (SRU). Unlike influence maximization, in which the goal is to find a set of nodes (users) in a social network who can maximize the spread of influence (Richardson and Domingos 2002; Kempe, Kleinberg, and Tardos 2003), the objective of sampling representative users is to identify a few "average" users who can *statistically* represent the characteristics of all users (Landauer 1997). Another type of related work is social influence analysis. Anagnostopoulos et al. (2008) and Singla and Richardson (2008) propose methods to qualitatively measure the existence of influence. Crandall et al. (2008) studies the correlation between social similarity and influence. Tang et al. (2009) presents a method for measuring the strength of such influence. The problem of sampling representative users from social networks is also relevant to graph sampling (Leskovec and Faloutsos 2006; Maiya and Berger-Wolf 2010; 2011; Ugander et al. 2013). However, most existing works focus on studying the network topology and ignore the topic information. Sun et al. (2013) aims to find representative users from the information spreading perspective and Ahmed et al. (2013) studies the network sampling problem in the dynamic environment. Papagelis et al. (2013) presents a sampling-based algorithm to efficiently explore a user's ego network respecting its structure and to quickly approximate quantities of interest. However, the problem itself is different from that sampling representative users for the whole network.

Problem and Our Solution. We use an example from a coauthor network to clearly demonstrate the motivation of this work. In Figure 1, the left figure shows a coauthor network and attributes (or interests) of each researcher. For example, George has a higher interest degree (0.9) on topic "database" than the degree (0.1) on "data mining". The right figure shows the output of topic-based representative users, highlighted with rectangles, where "Ada" and "Frank" are identified as the representative authors on topic "data mining" and "George" and "Eve" are representative users on the "database" topic. The weighted directed edge between the representative user and another user indicates how likely the selected user represents the other user. For example, on topic "data mining", "Ada" has a higher representative degree for "Eve" than "Bob". The problem of sampling representative users from large networks is fundamental for many social network mining tasks. For example, in existing literature,

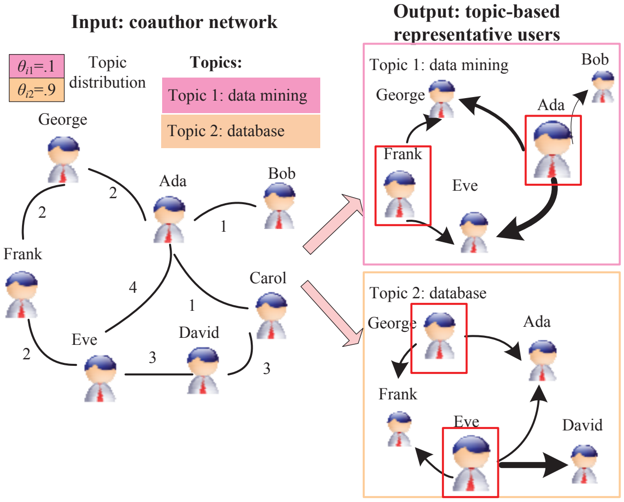


Figure 1: Example of sampling representative users from the coauthor network.

analytic/mining tasks were usually evaluated on a random sampled network. How reliable is the result? It is necessary to design a principled approach to solve the problem.

In this paper, we aim to systematically investigate the problem of sampling representative users from social networks. We formally define the and by linking it to the Dominating Set Problem, we prove that the problem is NP-hard (Cf. Section 3). We design a quality function Q to evaluate the set of sampled users and develop two instantiation models for estimating the Q function. To approximate the optimal solution, we present efficient algorithms to solve the two models with provable approximation guarantees. Finally, we conduct extensive experiments to demonstrate the effectiveness and efficiency of the proposed models on one coauthor dataset and another microblog dataset.

Problem Formulation

Consider a social network $G = (V, E)$, where V is the set of users and $E \subset V \times V$ is the set of directed/undirected links between users. Let $X_i = [x_{ik}]_{k=1\cdots d}$ denote a set of attributes (topics) associated with the user $v_i \in V$, where d is the number of attributes and x_{ik} records the value of the k^{th} attribute of user v_i. For each edge $e_{ij} \in E$, we associate a weight w_{ij}.

Given this, we can define a function to indicate the representative degree of a subset of users T. More precisely, for any subset $T \subseteq V$ and a user $v_i \in V$, we define the *representative degree* of T for v_i on a specific attribute a_j to be $R(T, v_i, a_j)$ with a score ranging in $[0, 1]$. We say T perfectly represents v_i on attribute a_j if $R(T, v_i, a_j) = 1$. For $T = \emptyset$, we have $R(T, v_i, a_j) = 0$. If T contains only one user v, $R(T, v_i, a_j)$ is the degree that v represents v_i on attribute a_j. This score is determined by the input networking data and the way to compute the function R. This formulation is very flexible and can easily incorporate other information. For example, to incorporate the network information, a straightforward method is to treat each node in the network as an attribute and define a neighbor group for each node, i.e., T is equivalent to the set of neighbors for a given user v_i.

Without loss of generality, there are often two types of attributes. The first type is the numeric attribute. For example, in a coauthor network, an attribute can be the interest degree of a user on a topic. Those people with a higher value are more representative. The second type is the non-comparable attribute, for example, the job title. Intuitively, it is better to select users with different job titles (to cover all values of the attribute). To do this, one idea is to divide all people into different groups according to the attribute value. Then the problem is cast as selecting users to represent each group.

Based on this idea, we can define a group of people with attribute value a_{j_k} as $V_k \subseteq V$. We refer to such a group together with the corresponding attribute as an *attribute group*. We use t to denote the number of attribute groups and $\mathbf{G}$ to denote the set of all groups, thus

$$\mathbf{G} = \{(V_1, a_{j_1}), (V_2, a_{j_2}), \dots, (V_t, a_{j_t})\}.$$

Now we give the definition of representative degree for each attribute group as follows.

Definition 1. ***Attribute group representative degree.*** *For an attribute group (V_l, a_{j_l}) $(1 \le l \le t)$, let*

$$P(T, l) = \sum_{v_i \in V_l} R(T, v_i, a_{j_l})$$

be the representative degree of T for attribute group (V_l, a_{j_l}). We say the attribute group (V_l, a_{j_l}) is represented if all people in V_l are perfectly represented on attribute a_{j_l}.

In practice, it might be difficult to find a perfect representation. The representative degree $P(T, l)$ then quantifies how likely the subset T can represent all people in V_l on attribute a_{j_l}.

Our goal in this paper is to find the representative set T so that it has the highest representative degree $P(T, l)$ for all attributes. In general, given a social network G, the attributes distributions X, the attribute groups $\mathbf{G}$ and the representative set T, we can formally define the following problem:

Problem 1. ***Sampling Representative Users.*** *Given (1) a social network $G = (V, E)$, where V is the set of users and E is the set of directed edges, (2) d attributes associated with each user, a desirable number k of users to be selected, (3) t attribute groups to be represented, (4) a quality function Q, how to find the set T of k representative users with maximal quality? Formally, we have the following optimization problem:*

$$\arg\max_{T \subseteq V, |T| = k} Q(G, X, \mathbf{G}, T).$$

Proposed Models

In this section, we first perform a theoretical investigation of the problem of sampling representative users from a social network. We link this problem to the Dominating Set Problem and prove its NP-hardness. Then we develop two instantiation models for the problem.

Theoretical Basic

Given a predefined quality function $Q(G, X, \mathbf{G}, T)$, the problem of sampling the representative set with maximal quality is referred to as a *Q-evaluated Representative Users Sampling* Problem (shortly Q-SRU). For any specified Q, the Q-evaluated representative users sampling problem is a subproblem of the original representative users sampling problem. The following theorem shows the NP-hardness of this problem.

Theorem 1. *For any fixed Q, the Q-evaluated Representative Users Sampling problem is NP-hard.*

Proof. We prove the theorem by a reduction to the Dominating Set Problem (Garey and Johnson 1979; Karp 1972). We construct an instance of the Q-SRU problem as follows:

Consider in a network G, if there is only one attribute (i.e., a_1) and only one attribute group (V, a_1). For $v_i \neq v_{i'} \in V$, assign $R(\{v_{i'}\}, v_i, a_1) = 1$ if edge $e_{i,i'} \in E$; assign $R(\{v_{i'}\}, v_i, a_1) = 0$ otherwise. For every $v_i \in V$, we also assign $R(\{v_{i'}\}, v_i, a_1) = 1$. We can see that for a given subset of V, a person is perfectly represented if and only if the corresponding vertex is dominated. Therefore the Dominating Set Problem is equivalent to the problem of evaluating whether there exists a k-element representative set to represent all users perfectly.

Regarding the quality function, Q can be rewritten as a t-variable function f taking $P(T, l)$ $(1 \leq l \leq t)$ as parameters. When every $P(T, l)$ reaches its maximal value $|V_l|$, the value of f is maximized, and this must be the only maximal point of f (distinguishable). If we can find the maximal value of Q, we can accordingly determine whether f (equal to Q) reaches the maximal value, which allows us to further determine whether there exist a k-element dominating set. Based on this analysis, we can conclude that the Q-evaluated Representative Users Sampling problem is NP-hard. □

Two Sampling Models

In social science, two basic principles to select representative users are synecdoche (in which a specific instance stands for the general case) and metonymy (in which a specific concept stands for another related or broader concept) (Landauer 1997). In effect, we treat one user as being a synecdochic representative of all users, and we treat one measurement on that user as being a metonymic indicator of all of the relevant attributes of that user and all users. Based on the above principles, we can consider many different methods to choose the representative user, such as statistics (Landauer 1997), grounded theory (Strauss and Corbin 1990), political theory (Schuler and Namioka 1993), and design practice. Here, we develop two practical instantiation models: Statistical Stratified Sample (S^3) and Strategic Sampling for Diversity (SSD) models.

Model 1: Statistical Stratified Sample (S^3) Maximizing the representative degree of all attribute groups is infeasible in practice. Some trade-offs should be considered as we may need to choose some less representative users on some attributes in order to increase the global representative degree on all attributes groups.

For each attribute group $(V_l, a_{j_l}) \in \mathbf{G}$, we assign a score m_l $(m_l > 0)$ to indicate the "importance" of this group. Usually we can choose m_l to be related with the size of V_l. The groups with a higher m_l value should be better represented. We do not want to finally have an attribute group with only one single user. It is preferable to have some "bias" to select user who can represent large attribute groups. We use a value $\beta \in (0, 1]$ to denote this bias, and give the following quality function:

$$Q(G, X, \mathbf{G}, T) = \sum_{(V_l, a_{j_l}) \in \mathbf{G}} m_l \{P(T, l)\}^{\beta}. \qquad (1)$$

To explain the "bias" clearly, we consider an extreme but intuitive case: The attribute groups have no intersection ($V_{l_1} \cap V_{l_2} = \emptyset$ for all l_1, l_2), and $R(T, v_i, a_j) = 1$ if $v_i \in T$, $R(T, v_i, a_j) = 0$ otherwise. Then for each group $1 \leq l \leq t$, the representative degree $P(T, l)$ is exactly the number of users in both T and V_l. Therefore the summation of all $P(T, l)$ is the fixed value $|T| = k$.

For $0 < \beta < 1$, by the concave property of function x^{β}, Q reaches the maximal value if and only if $P(T, l_1) : P(T, l_2) = m_{l_1}^{\frac{1}{1-\beta}} : m_{l_2}^{\frac{1}{1-\beta}}$ for any $1 \leq l, l' \leq t$. If we select $m_l = |V_l|^{1-\beta}$, the proportion of the selected users from each group tends to be the same. For $\beta \to 0$, the bias is linear in the sense that for every $1 \leq l \leq t$, the number of representative users for (V_l, a_{j_l}) is the same proportion of m_l. For $\beta = 1$, the bias tends to infinity in the sense that all the selected users represent the most important attribute group.

Model 2: Strategic Sampling for Diversity (SSD) In this model our goal is the diversification of delegates. When k is small, diversity means to find some users to cover (having a non-zero representative degree) as many groups as possible. We would ignore this case here, because the users to be selected are usually more than the number of attribute groups so that we can pick one user for each group to insure all groups are covered. Assume that there is a representative set T such that $P(T, l) > 0$ for every $1 \leq l \leq t$. "Diversity" here means that we want to sample users such that we have a balanced $P(T, l)$ for all attributes (by avoiding extremely large or small $P(T, l)$).

In practice, the size of attribute groups may vary significantly. For a large group V_l, $P(T, l)$ should be also relatively large. The objective is that all these values can be balanced. For example, an extreme case is that all the values are identical or the size of each group is the same. Simply requiring the same distribution is not enough to achieve this goal. Moreover, the practical case is much more complex. Consider the case that every attribute group has the same size and we want all $P(T, l)$ to be identical. In this case, any representative set is no better than the empty set, which represents every group with a same degree. This is not useful to solve the problem. We still require every attribute group to be well represented, thus we give the following quality function:

$$Q(G, X, \mathbf{G}, T) = \min_{(V_l, a_{j_l}) \in \mathbf{G}} \{\lambda_l \cdot P(T, l)\}, \qquad (2)$$

```
Input: The set of users $V$; the set of attributes $A$; the attributes
       values $X$; the values $R(T, v_i, a_j)$ for every $|T| = 1$
       and $v_i \in V$, $a_j \in A$; the number of users to find $k$.
Output: A set $T$ of $k$ representative users.
$T = \emptyset$;
while $|T| < k$ do
    $max = -1$;
    foreach $v_i \notin T$ do
        foreach $(V_l, a_{j_l}) \in \mathbf{G}$ that $v_i$ can contribute do
            foreach $v_{i'} \in V_l$ that $R(\{v_i\}, v_{i'}, a_{j_l}) > 0$ do
                Compute the increment of $R(T, v_{i'}, a_{j_l})$;
            end
            Compute the total increment of $P(T, l)$;
        end
        Compute the increment of quality by adding $v_i$;
        if $increment > max$ then
            $v = v_i$;
            Update $max$;
        end
    end
    $T = T \cup \{v\}$;
    Update the values $R(T, v_{i'}, a_j)$ and $P(T, l)$;
end
return $T$;
```

Algorithm 1: Approximate algorithm for S^3 model.

where λ_l is a positive constant associated with the attribute group (V_l, a_{j_l}). Usually we choose $\lambda_l = |V_l|^{-1}$. We call the attribute group with smallest $\lambda_l \cdot P(T, l)$ the *poorest group*. The quality function depends on the performance of T for the poorest group.

Approximate Algorithms

In this section, we present efficient algorithms with provable approximation guarantee to solve the proposed models.

Approximate Algorithm for S^3 Model

We give a greedy heuristic algorithm. Each time we traverse all users and find the one that mostly increases the quality function Q. We use arrays to store how each user is represented on each attribute, and how each attribute group is represented. For the increase in quality achieved by adding a user $v_i \in V$, we only need to consider the attribute groups that v_i can contribute to (at most t) and v_i's neighbors (we say $v_{i'}$ is v_i's neighbor if $R(\{v_i\}, v_{i'}, a_j) > 0$ for some $a_j \in A$) in those attribute groups. The algorithm is summarized in Algorithm 1 and its complexity is $O(t \times k \times |E|)$.

Error Bound Analysis: We show that the algorithm can guarantee a $(1 - 1/e)$-approximate.

We first consider the submodular property of the quality function defined in Eq. 1. A function $f : V \to \mathbb{R}^+$ is submodular if for all sets $S \subseteq T \subseteq V$ and every element $v \in V$, it has

$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$$

Function f is monotone increasing if for all sets $S \subseteq T \subseteq V$, there is

$$f(T) \geq f(S)$$

For a function f that is both monotonically increasing and submodular, we could add elements one-by-one into a set T. Suppose the elements are $v_1, v_2, \ldots, v_k$, and we use T_i to denote the set of i-th step $\{v_1, v_2, \ldots, v_i\}$ $(1 \leq i \leq k)$. We see that each time we add an element into T, there is an increment of $f(T)$. If some of the previous elements were not added, this increment becomes larger or stays the same. At each step, we choose to add an element $v \in V$ that maximizes $f(T \cup \{v\}) - f(T)$. In this way, we can use a greedy heuristic, that is each time we choose the element that increases $f(T)$ the most, i.e.,

$$f(T_2) - f(T_1) \geq f(T_3) - f(T_2) \geq \cdots \geq f(T_k) - f(T_{k-1})$$

Intuitively, the greedy algorithm can generate a good approximate solution for the problem of sampling a k-element set T that maximizes $f(T)$. Suppose the generated set is T and the optimal set is T^*. We consider the set $T \cup T^*$, whose function value is larger than (or worse case equal to) that of T^* according to the monotonic property of f. We can construct $T \cup T^*$ as follows: we first use greedy heuristic and add k elements of T into it, then we add elements in $T^* - T$ one-by-one. We see that for every element in $T^* - T$, the increment is not larger than the increment of any element in T. Since $l \leq k$ there must be $2 \cdot f(T) \geq f(T \cup T^*) \geq f(T^*)$. A tighter bound is reported in (Nemhauser, Wolsey, and Fisher 1978). For every monotonically increasing, submodular, non-negative function f on V, the set generated by the greedy heuristic is at least $(1 - 1/e)$ of the optimal solution.

Approximate Algorithm for SSD Model

For the SSD model, according to its definition of the quality function, intuitively we can design a greedy algorithm as follows: each time we choose the poorest group, and select a user to increase the representative degree of this group.

Similar to Algorithm 1, we use arrays to store the current values of $R(T, v_i, a_j)$ and $P(T, l)$ for every $v_i \in V$, $a_j \in A$ and $1 \leq l \leq t$. At the beginning of this algorithm, all groups are considered to be the poorest. Then we process with greedy heuristics as follows: We combine the procedure of finding the poorest group and the procedure of finding the user to improve this group together. We traverse every user v_i, and every group (V_l, a_{j_l}) to which v_i contributes. Then we compute the improvement of $P(V_l, l)$ by adding v_i (similar to Algorithm 1). If (V_l, l) is even poorer than the poorest ever seen, we mark v_i as the best candidate. If this group is as poor as the poorest group, we consider whether the improvement of $\lambda_l P(V_l, l)$ is larger than the previous poorest group. If so we mark v_i as the best candidate. Details of the algorithm and its error bound analysis is omitted for brevity.

Experimental Results

We conduct various experiments to evaluate the effectiveness of the proposed approach. All data sets, codes, and tools to visualize the mining results are publicly available. [1]

[1] http://arnetminer.org/repuser/

Experimental Setup

Datasets. We perform experiments on two different genres of datasets. The first type is author datasets: DB and DM. Each consists of authors and coauthor relationships extracted from major conferences in a specific domain, where the goal is to find who are representative authors in the corresponding domain. For the Database domain, we extract all authors from papers published on the three major conferences (SIGMOD, VLDB, ICDE) during 2007-2009. In total, we collect 8,027 authors and 23,770 coauthor relationships. For the Data Mining domain, we extract authors from papers published on the three major conferences (SIGKDD, ICDM, CIKM) during 2007-2009. In total, we collect 6,394 authors and 12,454 coauthor relationships.

As for the attributes of authors, we extract keywords from the papers in each dataset. For each author, we then define the value of an attribute as the number of times she/he uses the corresponding keyword in the authored papers. For each dataset, we first downcase all the keywords and then extract 200 most frequently used keywords. For evaluation, it is difficult to find a standard dataset with ground truth. Finally, we take program committee (PC) members of the conferences in each dataset during 2007-2009 as representative users. Typically, program committee members of a conference not only include top experts in a domain, but also include authors with expertise covering all subareas of the domain. Therefore, our goal is to predict who are representative users (PC members of the three major conferences) for the two research domains, respectively. Finally, we collect 291 representative users (by removing those who are not coauthors papers we collected) in the Database dataset and 373 representative users in the Data Mining dataset. Table 1 shows detailed statistics of the two datasets.

The second type of data is a mibroblog network. We crawled the messages from Weibo and classify them by keywords into 4 datasets, namely, program, food, student and public welfare. Then, we extracted the senders and the sender-follower relationships for each message. For program domain, we find 330 messages and collect 19,152 senders as well as 19,225 sender-follower relationships. There are 2,956 messages including 189,176 senders and 204,863 sender-follower relationships in food domain. 859 messages with 79,052 senders and 76,559 sender-follower relationships are collected for student domain while 2,410 messages with 324,594 senders and 383,702 sender-follower relationships are collected for public welfare domain. We consider the basic information of each user as the attributes, including location, gender, registration date, verified type, status, description and the number of friends or followers. All the attributes are classified into several categories denoting by numbers and the value of an attribute is defined as those number. As there is no ground truth for the Weibo data, we mainly use this dataset for evaluating efficiency and qualitative evaluation.

Evaluation Measures, Baseline Methods. For comparison purposes, we define the following baseline methods:

- *InDegree*. It simply takes the number of links as the criteria to select the representative users who have the highest indegree scores.

Table 1: Statistics of the two datasets.

Dataset	Conf.	#authors	#coauthorships	#representatives
Database	SIGMOD	3,447	9,507	256
	VLDB	3,606	8,943	251
	ICDE	4,150	9,120	244
Sum	ALL	8,027	23,770	291
Data Mining	SIGKDD	2,494	4,898	243
	ICDM	2,121	3,452	211
	CIKM	2,942	4,921	205
Sum	ALL	6,394	12,454	373

- *HITS_h and HITS_a*. It applies the HITS algorithm (Kleinberg 1999) to the author networks to calculate two values for each node: *authority* representing a score of authority of the node, and *hub* value representing a score based on the node's out-links to other nodes, thus resulting in two baselines, HITS_h and HITS_a. The former selects nodes with the highest hub scores while the latter selects nodes with the highest authorities.
- *PageRank*. It employs the PageRank algorithm (Page et al. 1999) to estimate the importance of each node in a network via a random process. This method selects nodes with the highest PageRank scores as representative users.

In our method, we consider the network information by treating each node as an attribute. We implement all the algorithms C++. In all experiments, we conduct evaluations for all methods in terms of P@10, P@50, P@100, R@50, R@100, and F1-measure (Buckley and Voorhees 2004).

Results

We evaluate the performance of the proposed models for sampling representative users on the two datasets. Let f_l be the total frequency of the l-th keyword ($1 \leq l \leq 200$). We assign $m_l = f_l^{0.5}$ in the S^3 model and $\lambda_l = f_l^{-1}$ in the SSD model. The parameter β in the S^3 model is set to be $\beta = 0.7$, by tuning from 0.1 to 1 with interval 0.1. We also compare our approach to the baseline methods.

Table 2 and 3 list the results of the comparison methods on the two datasets with the following observations:

High Precision. The proposed models achieve the best performance on both datasets. In terms of P@10, P@50, and P@100, the S^3 model significantly outperforms the other methods (+18.0%-28.0% by P@50 for example) on the Database, and performs best as well on Data Mining in terms of P@10, P@50. On the Database the SSD model finally achieves the best accuracy by Precision, Recall, and F1-measure, while on Data Mining S^3 model achieves the best accuracy.

Precision-Recall Curve. Figure 2 illustrates the precision-recall curve of the different methods. It can be seen that on the Database dataset, our S^3 model clearly outperforms the baseline methods. The SSD underperforms the InDegree algorithm when the recall is small, but when increasing the recall, for example up to 40%, the SSD model performs the best, even better than the S^3 model. On the Data Mining

Table 2: Performance of different methods on the Database dataset (%). (In Model 1, we set $\beta = 0.6$.)

Methods	P@10	P@50	P@100	R@50	R@100	F1
InDegree	50.0	36.0	40.0	6.2	13.7	33.0
HITS_h	40.0	30.0	27.0	5.2	9.3	20.3
HITS_a	30.0	30.0	30.0	5.2	10.3	24.1
PageRank	40.0	26.0	27.0	4.5	9.3	24.7
S^3	70.0	46.0	46.0	7.9	15.8	41.9
SSD	10.0	36.0	39.0	6.2	13.4	32.6

Table 3: Performance of different methods on the Data Mining dataset (%). (In Model 1, we set $\beta = 0.6$.)

Methods	P@10	P@50	P@100	R@50	R@100	F1
InDegree	80.0	62.0	45.0	8.3	12.1	28.4
HITS_h	70.0	42.0	30.0	5.6	8.0	21.4
HITS_a	70.0	48.0	30.0	6.4	8.0	18.8
PageRank	50.0	38.0	30.0	5.1	8.0	24.9
S^3	80.0	64.0	53.0	8.6	14.2	36.7
SSD	20.0	32.0	22.0	4.3	5.9	22.5

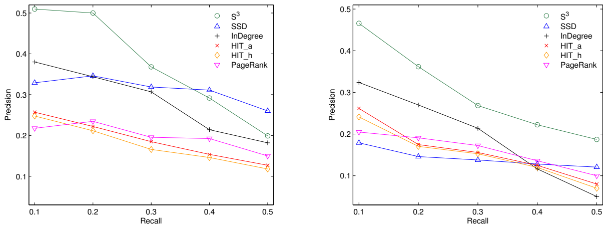
dataset, the S^3 also model clearly performs best. The SSD model starts to perform better when increasing recall.

Efficiency. We further compare the efficiency of the proposed methods with the baseline methods. Table 4 lists the CPU time required by the different methods to find representative users respectively from the coauthor datasets and the Weibo dataset. Our methods are found to be very efficient, for the SSD model only requires 0.54s for Database, 0.74s for Data Mining and 0.67s for Weibo in program domain, and is faster than the S^3 model.

Effect of Parameter β for S^3. On both datasets, the S^3 performs the best. We set the parameter β in the S^3 model as $\beta = 0.7$. We then analyze how the parameter affects the model performance. We test the performance of the S^3 model by with the parameter β varied. We found that the model is not very sensitive with β, although the performances of S^3 with different values for β are a bit different. Generally, S^3 achieves the best performance on Database at $\beta = 0.4$, and on Data Mining at $\beta = 0.7$. This confirms the effectiveness of our proposed methods.

Case Study. We demonstrate here the effectiveness of the two proposed models for representative user identification on the Database and Data Mining datasets.

Table 5 shows representative authors found by our methods and one baseline method (PageRank) from the two coauthor datasets. For each dataset, we list the top 10 representative authors found by the different methods (more details about the results is online available.). Compared with the authority-based baseline method (PageRank), our method has several distinct advantages: First, the baseline method can only measure the similarity between nodes, but does not consider the diversity of attributes, which is necessary for the statistical representation. Second, the baseline method cannot tell which users can be represented by a selected representative user on a specific attribute, while our methods have the capacity to do so.



(a) Database

(b) Data Mining

Figure 2: Precision-recall curves of the comparison methods on the Database and Data Mining datasets.

Table 4: Efficiency performance of different methods on the three datasets. The CPU time does not include the time of loading the network data. (Second)

Methods	Database (291)	Data Mining (373)	Weibo (200)	
			Program	Food
InDegree	0.10	0.16	0.01	0.02
HITS_h	2.44	1.58	0.99	45.66
HITS_a	2.44	1.58	0.95	45.60
PageRank	6.89	2.95	0.51	18.33
S^3	1.68	3.62	4.45	50.71
SSD	0.54	0.74	0.67	10.81

Conclusions

In this paper, we study a novel problem of sampling representative users from social networks, with the objective of finding a small subset of users to statistically represent all users in the original social network. We formally define this problem and perform a theoretical investigation of the problem, and prove its NP-hardness. Approximate algorithms for the two models have been developed to efficiently choose the set of representative users. Experimental results on two author datasets demonstrate the effectiveness and efficiency of the proposed models.

Acknowledgements. Jie Tang and Chenhui Zhang are supported by National 863 project (2014AA015103), National 973 projects (2014CB340506, 2012CB316006, 2011CB302302), NSFC (61222212), and a research fund from IBM.

Table 5: Example of representative author sampling from the coauthor datasets. From each dataset, we list the top 10 representative users found by the two methods respectively.

S^3		PageRank	
Database	Data Mining	Database	Data Mining
Jiawei Han	Philip S. Yu	Serge Abiteboul	Philip S. Yu
Jeffrey F. Naughton	Jiawei Han	Rakesh Agrawal	Jiawei Han
Beng Chin Ooi	Christos Faloutsos	Michael J. Carey	Jian Pei
Samuel Madden	ChengXiang Zhai	Jiawei Han	Christos Faloutsos
Johannes Gehrke	Bing Liu	Michael Stonebraker	Ke Wang
Kian-Lee Tan	Vipin Kumar	Manish Bhide	Wei-Ying Ma
Surajit Chaudhuri	Jieping Ye	Ajay Gupta	Jianyong Wang
Elke A. Rundensteiner	Ming-Syan Chen	H. V. Jagadish	Jeffrey Xu Yu
Divyakant Agrawal	Padhraic Smyth	Surajit Chaudhuri	Haixun Wang
Wei Wang	C. Lee Giles	Warren Shen	Hongjun Lu

References

Ahmed, N. K.; Neville, J.; and Kompella, R. 2013. Network sampling: From static to streaming graphs. *ACM Trans. Knowl. Discov. Data* (8).

Anagnostopoulos, A.; Kumar, R.; and Mahdian, M. 2008. Influence and correlation in social networks. In *KDD'08*, 7–15.

Buckley, C., and Voorhees, E. M. 2004. Retrieval evaluation with incomplete information. In *SIGIR'2004*, 25–32.

Chen, W.; Wang, Y.; and Yang, S. 2009. Efficient influence maximization in social networks. In *KDD'09*, 199–207.

Crandall, D.; Cosley, D.; Huttenlocher, D.; Kleinberg, J.; and Suri, S. 2008. Feedback effects between similarity and social influence in online communities. In *KDD'08*, 160–168.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company.

Goyal, A.; Bonchi, F.; and Lakshmanan, L. V. 2008. Discovering leaders from community actions. In *CIKM'2008*, 499–508.

Karp, R. M. 1972. *Reducibility Among Combinatorial Problems. In the book of Complexity of Computer Computations, R. E. Miller and J. W. Thatcher (editors)*. New York, USA: Plenum.

Kempe, D.; Kleinberg, J.; and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *KDD'03*, 137–146.

Kleinberg, J. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46(5):604–632.

Landauer, T. K. 1997. Behavioural research methods in human-computer interaction. *M. Helander, T.K. Landauer, and P. Prabhu, (Eds.), Handbook of Human-Computer Interaction*.

Leskovec, J., and Faloutsos, C. 2006. Sampling from large graphs. In *KDD'06*, 631–636.

Maiya, A. S., and Berger-Wolf, T. Y. 2010. Sampling community structure. In *WWW'10*, 701–710.

Maiya, A. S., and Berger-Wolf, T. Y. 2011. Benefits of bias: Towards better characterization of network sampling. In *KDD'11*, 105–113.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* 14(1):265–294.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report SIDL-WP-1999-0120, Stanford University.

Papagelis, M.; Das, G.; and Koudas, N. 2013. Sampling online social networks. *IEEE TKDE* 25(3):662–676.

Richardson, M., and Domingos, P. 2002. Mining knowledge-sharing sites for viral marketing. In *KDD'02*, 61–70.

Schuler, D., and Namioka, A. 1993. *Participatory Design: Principles and Practices*. Hillsdale, NJ, USA: Erlbaum.

Scripps, J.; Tan, P.-N.; and Esfahanian, A.-H. 2009. Measuring the effects of preprocessing decisions and network forces in dynamic network analysis. In *KDD'2009*, 747–756.

Singla, P., and Richardson, M. 2008. Yes, there is a correlation: - from social networks to personal behavior on the web. In *WWW'08*, 655–664.

Strauss, A. L., and Corbin, J. M. 1990. *Basics of qualitative research : grounded theory procedures and techniques*. Newbury Park CA USA: Sage.

Sun, K.; Morrison, D.; Bruno, E.; and Marchand-Maillet, S. 2013. Learning representative nodes in social networks. In *PAKDD'13*, 25–36.

Tang, J.; Sun, J.; Wang, C.; and Yang, Z. 2009. Social influence analysis in large-scale networks. In *KDD'09*, 807–816.

Ugander, J.; Karrer, B.; Backstrom, L.; and Kleinberg, J. 2013. Graph cluster randomization: Network exposure to multiple universes. In *KDD'13*, 329–337.