ISSN 2395-1621

Effective Use of Sampling Technique for Online Social Networks

[#1]Renuka Satpute, [#2]Prof. N. D. Kale

[1]11.satpute.renuka@gmail.com
[2]ndkpvpit@gmail.com

[#12]Department of Computer Engineering, Savitribai Phule University,
TSSM's PVPIT, Bavdhan, Pune, Maharashtra, India

ABSTRACT

As online social networking comes into view, there has been increased interest to employ the arising structure as well as the accessible information on social peers to improve the information which required to user. Our focus is to improve the sampling results to efficiently explore a user's social network respecting its structure and to quickly approximate quantities of interest. We introducing and analyzing types of the basic sampling method that exploring connections between our samples. We show that our algorithms can improve sampling of nodes in a network structure, assumes detail for each user in the network is available. Also our algorithms can be applied for doing sampling of information in various social networks. Using real and synthetic data sets, we demonstrate the results of our analysis and validates the efficiency of our algorithms be close to quantities of interest. The general methods are described here and can probably be adopted in various types of strategies that efficiently collect the information from a social graph. Our main contribution is to select the samples by making use of sampling-based algorithms (selectivity estimation via k-mean sampling) that given a user in a social network quickly obtain a near uniform random sample of nodes in its neighborhood and generate the graph.

Keywords: Online Social Network, Information Networks, Search Process, Query Processing, Performance Evaluation, Privacy..

ARTICLE INFO

Article History

Received: 18th July 2019

Received in revised form :

18th July 2019

Accepted: 22nd July 2019

Published online :

22nd July 2019

I. INTRODUCTION

The changing tendency in the use of web technology that aims to enhance connectivity, information sharing on the web have +led to the emergence of online social networking services. This is observable by the host of activity and social interaction which discovered in web sites like Facebook, Linkedin, and Twitter etc. At the same time the desire to connect and interact progress far beyond centralized social networking sites and takes the form of ad hoc social networks formed by instant messaging clients, VoIP software, or mobile networks. While interactivity with people beyond one's contact list is currently not possible , the implied social networking structure is in place. Taking of large assumption of these networks, there has been growing the attentiveness to search the underlying information in order to improve on information retrieval tasks of social peers. these tasks are in the center of many application domains. Social search is a type of search method that determine the relevance of search results by sharing the data or handouts of users. The main concept is that we can increase or improved the correctness of search results by collecting and analyzing information from a user's explicit or implicit social network. The social search is done as the following:

1. The user U enters the query to the search engine.
2. The search engine evaluate an ordered list L of which is most related to the results using Global Ranking Algorithm.
3. The search engine getting information that lies in the neighborhood of U and relates to the results in L.
4) The search engine utilizes this information to re-order the list L to a new list L′that is presented to U.

This Project, focus on improving the performance of information collection from the neighborhood of a user in a DSN.

II. LITERATURE REVIEW

In this paper [1], they introduce the potential for using online social networks to enhance Internet search. Analyzed the differences between Web and social networking systems in terms of the mechanisms they use to locate useful information. They explained benefits of integrating the strategies for searching useful and meaningful information in both Web and social networks. Our initial results from a

social networking experiment suggest that such integration has the possible to improve the quality of search experience.
In this paper [2], they proposed Quality-guaranteed Multi-network Sampler approach for jointly sampling of multiple OSNs. It provides a statistical guarantee on the difference between the sampled real dataset and the ground truth. They addresses the issues of sampling data across different OSNs and computed the quality of sampled datasets.
In this paper [3], presents a simple method which produces reasonable estimates for most applications, requiring only a modest amount of hand calculation. The method should prove to be of considerable utility in connection with branch-and-bound approach to combinatorial optimization of D. H. Lehmer.
In this paper [4] they showed the branch-and-bound solution time of an MIP solver can be roughly estimated in the early stages of the solution process. They explained a procedure for this estimation based on parameters of a small sub tree. Our experiments showed that in a relatively short time, we can receive sufficient information to predict the total running time with an error within a factor of five.
In this paper [5] they propose two online approaches for evaluating the size of a backtracking search tree. In the first approach it is basically based on a weighted sample of the branches that visited by chronological backtracking. In the second approach there is a recursive method based on assuming that the unexplored part of the search tree will be similar to the part we have so far explored. They also demonstrate that search tree size can be used to select the algorithm to perform best on a particular problem instance.
In this paper [6] they proposed two algorithms for estimating the size of graphs. Both algorithms depend on nodes being samples from the graph's stationary distribution. Presented both analytically and experimentally that, for social-networks and other small world graphs, these algorithms considerably outperform uniformly sampling nodes. They provide exact estimates using a smaller number of samples. This result is reliable while uniformly sampling nodes is much harder than sampling them according to the stationary distribution.

III. PROPOSED APPROACH

A. Problem Statement

It is necessary to concentrate on improving the performance of collecting the information from a node in neighborhood to a user or a dynamic social network. Node's social network to predict correctly we proposed sampling based algorithms to compress interest structure of users and social network.

B. Sampling Methodology

We first describe an idealized approach in which we assume it is possible to quickly get a uniform random sample of D(v). The sampling notation we use is shown in Table1 for reference. Assume that x1, x2; . . . ; xn be the values of the nodes in D (v). Suppose, we could get a uniform random sample S of size n << N with S ∩ D(v) and values x1, x2; . . . ; xn. Assume that X be the sample sum, i.e., X =∑€Sxi. Then it is well known that the quantity X'=x.(N/n) i.e., the sum of sample is scaled by the inverse of the sampling fraction, which is nearest for x . In fact, X' is a random variable whose mean and standard deviation can be approximated (for large N) by the following well-known sampling theorem

$$E[X'] = X,\ sd[X'] = N.б/\sqrt{n}$$

The standard deviation sd[X'] provides an estimate of the error in estimating X by X'. Since б,the standard deviation of values inD(v), is usually not known in advance, it can be itself estimated by computing the standard deviation б' of the sample; thus sd[x'] is estimated as N.б/√n .

Notation	Explanation
N	number of nodes in D(v)
S	set of nodes in sample
n with n << N	number of nodes in sample
x1,x2,….xN	Values of nodes in D(v)
x1,x2,….xn	values of nodes in S
Б	standard deviation of values in D(v)

Table 1: Sampling Notation

Sampling in static network:
We first consider the case where the social network is static, or changes slowly over time. In this case, a simple solution is there, where each node, in a preprocessing phase, performs a complete crawl of its neighborhood D(v) and selects a uniform random sample S of n nodes, whose access paths are then stored at the starting node. At runtime, the value stored at each sample node is accessed and aggregated. This precomputation phase is computationally intensive.

Sampling in dynamic network:
We next consider the case where the network is dynamic, in which the structure of network changes rapidly as the data changes at each node. In such a case, to precompute samples of D (v) as such samples become stale very quickly. Thus, the task of sampling from D (v) has to be deferred to runtime. This problem is challenging because we cannot crawl the entire neighborhood D (v) at runtime (this will be prohibitively slow). There are methods to generate a uniform random subset of nodes of a large graph via random walks. However, in our case, we can improve upon generic random walk methods on graphs as we can leverage the fact that we need to only sample from the neighborhood D(v) of a node v with a small depth d. Consequently, we are able to develop even more efficient random walk procedures. We first make a simplifying assumption that the graph structure of the neighborhood D (v) resembles a tree rooted at v. The solution that we first present will consist of random walks that are initiated from the root of this tree v and follow edges toward the leaves of the tree. Later, we describe how to generalize this basic approach for more general graph structures that are not trees essentially by constraining our random walks to only follow edges of a spanning tree of D (v) rooted at v.

C. Proposed System Overview

Information in Social Networks:
In social networking system information of a registered person will be stored in session. For each user in G we assume that a log accumulated overtime is available.

Endorsement of an item is defined in a generic sense and it may have various instantiations, for example clicking on a url, rating a movie, etc. Endorsements of items by users in the neighborhood of v comprise valuable social information that may be utilized to provide personalized rankings of items to v. In many social information tasks (such as in social search) we are interested in the relative order or ranking of a set of items X in the social network of v. Designing and evaluating a re-ranking algorithm that increases the user satisfaction is out of the scope of this project.

Loading data set and forming the tree of nodes:
In this section, Users are having authentication and security to access the detail which is presented in the Social network.

Calculating error rate and finding root node for starting sampling:
In this section, the users can send request to another user. If anyone send request means we can view and accept that request that will be added to friend table. In is the main concept and users searching the friends in the search box and they can send request to which they needed.

Sending and Receiving Requests:
In this module, it will display the friend list only if the particular field matches. It is the important module and if we need we can add the particular user to our friend list.

Figure 1 shows, the detailed description of the proposed system.

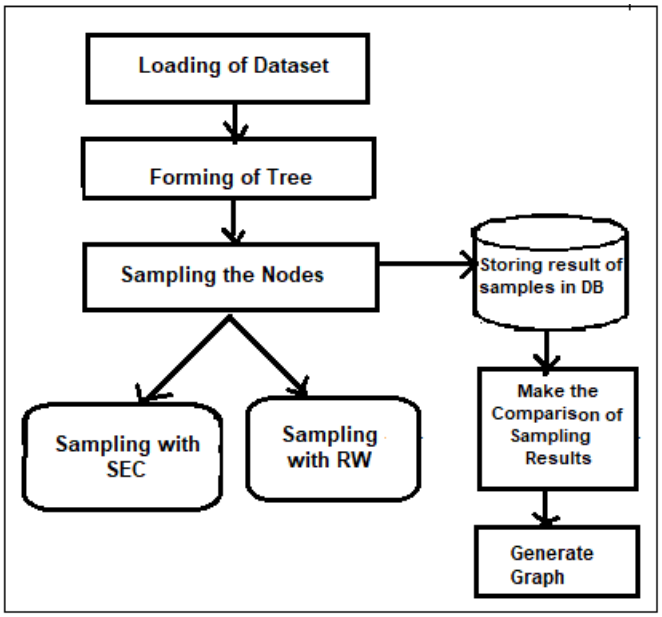

Figure 1.Proposed System Architecture

1) In this system, first loads the datasets as an input.
2) Forming the tree from given dataset.
3) Finding out the root node by using random walksand then getting the sampled nodes from the given dataset.
4) Then, sampling is done by Random walk and our proposed algorithm i.e. sampling estimation via k-mean clustering. Sampled nodes will send the message to all other nodes in the given dataset.
5) After that stores the results of samples in the database
6) Making the comparison by generating the graph.
7) Finally depending on the comparisons an analysis is made.

D. Algorithms

Algorithm 1:

1: procedure SAMPLEDYN (u ,n, d, C)
2: T = NULL, samples = 0, Sample array of size n
3: while samples <= n do
4: if (v = random Walk(u, d, C,T)) != 0 then
5: Sample [samples++] = v
6: end if
7: end while
8: end procedure
9: procedure RANDOMWALK (u, d, C, T)
10: depth = 0, ps = 1
11: while depth < d do
12: pick v € children(u) U u with pv = 1/degree(u) + 1
13: if T U v has no cycle then
14: add v to T

15: ps =ps.pv
16: if v = u then
17: accept with probability C/Ps
18: if accepted then
19: return v
20: else
21: return 0

22: end if
23: else
24: u = v,depth++

25: end if
26: end if
27: end while
28: return 0
29: end procedure

Algorithm2: Selectivity estimation via K-mean clustering
K-means mechanism:
Given a set of observations (x1, x2, …, xn), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k sets ($k \leq n$) S = {S1, S2, …, Sk} so as to minimize the within-cluster sum of squares (WCSS):

$$\arg\min_{\mathbf{s}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

where μi is the mean of points in Si.

D. Mathematical Model:

Mathematical Model for existing work:

Input: u, n, d, C
Where,
u- user node
n-number of nodes in the network

d- depth of the network

C- the constant value parameter

Let T=NULL, samples=0, sample size = n,

Check number of samples and depth d for the network Pick v children(u) with probability pv=1/degree(u) + 1 andT U v should not have cycle.

Mathematical model for proposed work:

Here in the proposed work the two problems are considered and also the comparison is made by showing the results of sampling on various datasets.

Input: u, n, d, C

Where,

u- user node

n-number of nodes in the network

d- depth of the network

C- the constant value parameter

Calculate the degree of nodes in the dataset which is denoted by degree(v)

The probability of random walk returning any specific leaf node bi is P(bi)=1/N

where N number of nodes in the vicinity of v Let T=NULL, samples=0, sample size = n,

Check number of samples and depth d for the network Pick v children(u) with probability

Pv=1/degree(u) + 1

T U v should not have cycle

Sampling of nodes in the social networks as well as sampling of information in the social networks is done by using random walks.

IV. Result and Discussion

A. *Experimental Setup:*

The system is built using Java framework on Windows platform. The Net bean IDE is used as a development tool. The system doesn't require any specific hardware to run; any standard machine is capable of running the application.

B. *Dataset:*

We have to use AOL dataset.

C. *Expected Result:*

In this section discussed the experimental result of the proposed system.

Table 1 shows, the comparison between the existing and proposed system algorithm. Figure 2 shows, comparison between the Random walk and SEC Sampling. From the graphs, it is concluded that the proposed SEC Method is more reliable than the Random walk.

Methods	No. of. Samples
Random Walk	17
SEC	7

Table 1: Comparison between the random walk and SEC

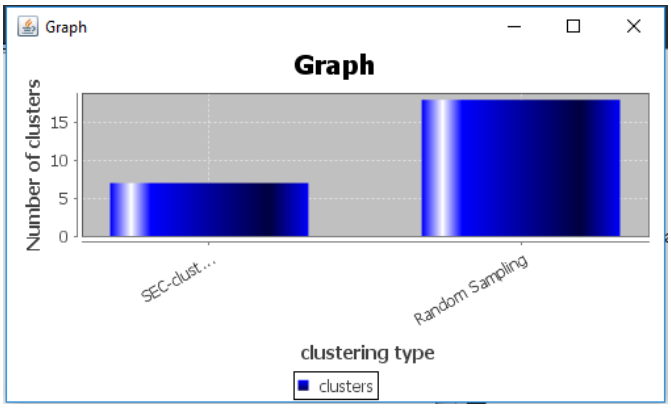

Figure 2. Comparison Graph

V. Conclusion

In our research proposes methods which collect information in time from the neighborhood of a node in a dynamic social network where knowledge of its structure is not available or else is limited. Methods given are based on sampling. Using sampling it is not necessary to visit all nodes in the area of user and hence get improved performance. The results of random sampling are analyzed. The proposed algorithm is used to calculate the count of sampled nodes. Our algorithm is selectivity estimation via k-mean clustering We studied what is low selectivity problem and tried to find the solution for the same. This problem can be minimized or the accuracy of sampling nodes can be increased. In our case data sets are used e.g. AOL dataset. Here we assumed that information for each user the network is available. Improving the sampling process is the major concern of our project.

ACKNOWLEDGEMENT

The authors would like to thank the researchers as well as publishers for making their resources available and teachers for their guidance.

REFRENCES

[1] A. Mislove, K.P. Gummadi, and P. Druschel, "Exploiting Social Networks for Internet Search," Proc. Fifth Workshop Hot Topics in Networks (HotNets), 2006.

[2] Hong-Han Shuai, "QMSampler: Joint Sampling of Multiple Networks with Quality Guarantee," DOI 10.1109/TBDATA. 2017.

[3] D.E. Knuth, "Estimating the Efficiency of Backtrack Programs," Math. of Computation, vol. 29, no. 129, pp. 121-136, 1975.

[4] G. Cornujols, M. Karamanov, and Y. Li, "Early Estimates of the Size of Branch-and-Bound Trees," INFORMS J. Computing, vol. 18, pp. 86-96, 2006.

[5] P. Kilby, J. Slaney, S.Thie´baux, and T. Walsh, "Estimating Search Tree Size," Proc. Nat'l Conf. Artificial Intelligence (AAAI), 2006.

[6] L. Katzir, E. Liberty, and O. Somekh, "Estimating Sizes of Social Networks via Biased Sampling," Proc. 20th Int'l Conf. World Wide Web (WWW), 2011.

