Leveraging History for Faster Sampling of Online Social Networks
Zhuojie Zhou1, Nan Zhang1
1Computer Science Department, George Washington University

Introduction

- **Motivation**: To enable third-party analytical applications of Online Social Networks (OSNs), one must be able to accurately estimate big-picture aggregates (e.g., the AVG age of users, the COUNT of user posts that contain a given word) by issuing a small number of individual-user queries through the social network’s web interface.

- **Problem Definition**: How to sample nodes from large graphs using random walks via graph browsing interface with limited query cost while obtaining as accurate estimation as possible?

- Our ideas: History-Aware Random Walks. The focus of this paper is to offer a “drop-in” replacement for this core design (of random walk), such that existing sampling-based analytics techniques over online social networks, no matter which analytics tasks they support or graph topologies they target, can have a better efficiency by leverage random walks’ history.

Preliminaries

- **Graph Browsing Interface**: The only access channels we have over the data is the web and/or API interface provided by OSNs. While the design of such interfaces varies across different real-world online social networks, most of all them support queries that take any user ID as input and return two types of information about u:
  - \( N(u) \), the set of all neighbors of u, and
  - all other attributes of u (e.g., user self-description, profile, posts).

- **Random Walk**: Simple Random Walk (SRW). Given graph \( G(V, E) \), and a node \( v \in V \), a random walk is called Simple Random Walk if it chooses uniformly at random a neighboring node \( u \in N(v) \) and transit to \( u \) in the next step.

  \[
  P_{vu} = \begin{cases} 
  1/k_u & \text{if } u \in N(v), \\
  0 & \text{otherwise}
  \end{cases}
  \]

  That is, SRW selects each node in the graph with probability proportional to its degree.

- **History-Aware Random Walks (Higher order Markov Chain)**:

  \[
  \Pr(X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_1 = x_1) = \Pr(X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_{m-n} = x_{m-n})
  \]

  \[
  \Pr(X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_1 = x_1) = \frac{1}{k_u}
  \]

  Figure: A demo shows a History-aware random walks on-the-fly.

CNRW: Circulated Neighbors Random Walk

- **Key idea**: The key idea of CNRW is to replace such a memoryless transition of SRW to a stateful process. Specifically, given the previous transition of the random walk \( u \rightarrow v \), instead of selecting the next node to visit by sampling with replacement from \( N(v) \), i.e., the neighbors of \( v \), we perform such sampling by circulating all \( u \)’s neighbors without replacement.

  **Theorems**
  - Theorem 1. CNRW has the same stationary distribution \( \pi(v) = k_v/|E| \) as SRW’s.
  - Theorem 2. The asymptotic variance of CNRW is no greater than SRW’s.

  \[
  V_c(\mu) \leq V_s(\mu).
  \]

  **Theorem 3.** For a ball graph, the transition probability

  \[
  \frac{P_{CNRW}}{P_{SRW}} > \frac{|G_1| - \ln |G_1|}{|G_1| - 1}
  \]

  Figure: A demo of CNRW, it chooses the next candidate from the set \( N(v) \) in a round-robin manner, e.g. circulating the nodes \( \{u, q, w\} \).

  Segments of alternating path blocks. They are circulated with a period of 3.

  In a round-robin path blocks. They are circulated with a period of 3.

  Figure: Comparisons of the block distribution in CNRW and SRW. CNRW creates alternating stratified path blocks that boost the sampling performance.

GNRW: Groupby Neighbors Random Walk

- **Key idea**: GNRW is a natural extension of CNRW. Instead of performing the circulation at the granularity of each neighbor of \( v \), we propose to stratify the neighbors of \( v \) into groups, and then circulate the selection among all groups.

  **Theorems**
  - Theorem 1. GNRW has the same stationary distribution \( \pi(v) = k_v/|E| \) as SRW’s.
  - Theorem 2. The asymptotic variance of GNRW is no greater than SRW’s.

  \[
  V_c(\mu) \leq V_s(\mu).
  \]

  Figure: An example of partitioning a node’s neighbors into 3 groups. GNRW chooses a next group by circulating \( \{S_1, S_2, S_3\} \).

  Segments of alternating path blocks. They are circulated with a period of 4.

  Figure: A demo of GNRW: it makes higher order of stratifications of the path blocks by grouping them into strata.

Experimental Results

- **KL-distance**

  \[
  KL-Divergence
  \]

  \[
  S1 \quad S2 \quad S3 \quad S4 \quad S5 \quad S6 \quad S7 \quad S8 \quad S9 \quad S10
  \]

  Figure: A demo of GNRW: it makes higher order of stratifications of the path blocks by grouping them into strata.

- **Estimation Error**

  \[
  Estimation Error
  \]

  \[
  0.04 \quad 0.05 \quad 0.06 \quad 0.07 \quad 0.08
  \]

  Figure: Comparisons of the block distribution in CNRW and SRW. CNRW creates alternating stratified path blocks that boost the sampling performance.

Conclusion

In this paper, we developed two algorithms: (1) CNRW, which replaces the memoryless transition in simple random walk with a memory-based, sampling-without-replacement, transition design, and (2) GNRW, which further considers the observed attribute values of neighboring nodes in the transition design. We proved that while CNRW and GNRW achieve the exact same target (sampling) distribution as traditional simple random walks, they offer provably better (or equal) efficiency no matter what the underlying graph topology is.

Selected References