

Uniform Data Sampling from a Peer-to-Peer Network

Souptik Datta, Hillol Kargupta*

Department of Computer Science & Electrical Engineering
University of Maryland, Baltimore County, Baltimore, MD, USA
{souptik1,hillol}@cs.umbc.edu

Abstract

Uniform random sample is often useful in analyzing data. Usually taking a uniform sample is not a problem if the entire data resides in one location. However, if the data is distributed in a peer-to-peer (P2P) network with different amount of data in different peers, collecting a uniform sample of data becomes a challenging task. A random sampling can be performed using random-walk, but due to varying degrees of connectivity and different sizes of data owned by each peer, this random walk gives a biased sample. In this paper, we propose a random walk-based sampling algorithm that can be used to sample data tuples uniformly from a large, unstructured P2P network. We model the random walk as a Markov chain and derive conditions to bound the length of the random walk necessary to achieve uniformity. A formal communication analysis shows logarithmic communication cost to discover a uniform data sample.

1. Introduction

Peer-to-peer (P2P) systems such as Gnutella, Napster, Kazaa are increasingly becoming popular for many applications that go beyond sharing media files. Most of the off-the-shelf data analysis/mining techniques do not work well in such highly decentralized, distributed environment. They are mostly designed for centralized applications where all the data are stored in a single central place. That motivated researchers to explore the possibilities of in-network analysis of data[8]without moving any data from their sources. However, complex data analysis procedures are hard to solve in an exact manner in unstructured P2P network without communication of the order of network size. Approximate techniques [5] can solve the problem with results reasonably close to centralized approach with much less communication though. Sampling is one of the basic statistical tools frequently used in approximation, and it is a proven

technique for effective estimation with probabilistic guarantee. Sampling can be very effectively used to derive important information regarding data distributed in a P2P network. For example, average size or playing time of the music files shared or requested can be estimated closely from a uniform sample of shared music files in a P2P file-sharing network, while actually computing it requires the near-impossible task of accessing all the files in the entire network. A uniform sample can be used for more complicated data mining tasks in P2P network like association rule mining and recommendation based on the that. It can also be applied to sensor networks, where multiple sensors observe an attribute from different locations and an average value of the attribute or its distribution over a time-period is of interest. However, collecting a uniform random data sample from a unstructured P2P network is challenging due to irregular nature of the network (varying degrees of connectivity between peers, widely varying amounts of shared data). That motivates us to design an effective uniform random sampling technique suitable for P2P network.

There exists a body of works on uniform sampling of nodes from a graph-like structure using random walk. The main technique used for this is random walk. A detailed survey of random walk on graphs is provided in [9]. Gkantidis *et al.* [6] provide an in-depth analysis of effectiveness of random walk in P2P network in searching and network discovery. Their work shows dependence of random walk's performance on the underlying topology of the network. Cooper *et al.* [4] use random sampling from a class of graph sharing some property (e.g., degree distribution) to model a P2P network. Note that there is a clear difference between the problem they address and the problem we are trying to address, as we are interested in taking a uniform sample of data from a P2P network instead of sample of nodes. Another closely related field of study is sampling web-pages uniformly in World-Wide-Web (WWW). Henzinger *et al.* [7] considered use of random walk on web-graph to approximate uniform sampling the URL-s. Bar *et al.* [15] uses random walk to answer aggregate queries about webpages. Finally, Bash *et. al* [2] propose algorithm

*Also affiliated with AGNIK LLC, Columbia, MD USA.

to sample uniformly in a distributed sensor network based on the geo-spatial information or location of each sensor.

In this paper, we propose an efficient sampling technique that can collect uniform random sample of data distributed in a P2P network taking care of irregular degrees of connectivity and sizes of data shared by different peers within a bounded communication cost. We believe that this is the first attempt for collecting uniform sample of data from a P2P network, and it makes a significant contribution by providing a tool that did not exist so far. It opens up lots of possibilities for analyzing data in a P2P network and enhancing P2P-users' experience.

The paper is organized as follows: The next section defines the problem formally and discusses the background knowledge necessary to understand our proposed sampling approach. In section 3, we propose our main approach of data sampling and show conditions necessary for uniformity of data sample. We also discuss how to bound the length of the random walk. Section 4 shows experimentally that our proposed technique indeed achieve uniformity within reasonable bound. Finally, section 5 concludes the paper.

2. Problem Definition and Background

In this section, we first formally define the specific problem we are interested in. Let there be n nodes in a peer-to-peer system, N_1, N_2, \dots, N_n . Node N_i , $i = 1, 2, \dots, n$, has a set of neighbors to which it is directly connected. This set is denoted by $\Gamma^{(i)}$. Let $X = X^{(1)} \cup X^{(2)} \cup \dots \cup X^{(n)}$ be the full dataset where $X^{(i)} \subset X$, $i = 1, 2, \dots, n$, denotes the subset of the data at node N_i . Let $X^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \dots, x_{n_i}^{(i)}\}$ be the set of n_i points in node i , and its size is given by $n_i = |X^{(i)}|$. Note that dataset is homogeneously distributed, where each node has common set of attributes. Let $n_i \neq n_j \forall j \neq i$, or in other words, data-size at individual peers are different from each other. The problem is how to take a sample $s = \{x_1, x_2, \dots, x_{|s|}\}$ of size $|s|$ from the network so that s represents a uniform, random sample of data tuples taken from the whole data X .

We will be taking random walk-based approach to solve the problem of data sampling in P2P network. Note that this discussion is applicable to not only P2P network topology, but to any general, finite, undirected graph. We first introduce the concept of random walk as Markov chain. Let us define some basic notations before the discussion. Let $G = (V, E)$ is a simple, connected, undirected graph with number of nodes, $|V| = n$ and number of edges, $|E| = m$. Let, d_i denotes degree of node i , $1 \leq i \leq n$, which is the number of edges with one end as node i , and d_{max} denotes maximum degree of all the nodes, i.e. $d_{max} \geq (d_i)$ for all $(1 \leq i \leq n)$. Let $P = [p_{ij}]$ denotes the transition probability matrix for G , where $0 \leq p_{ij} \leq 1$ is the probability of moving from node i to node j . Since,

there is no way to move from node i to a node j in one hop without a direct link in between, $p_{ij} = 0$ for all node j that are not neighbor of node i , and sum of the probabilities for a node, including the probability of moving to itself (self-transition probability) should be one, i.e. for any i , $\sum_{j=1}^n p_{ij} = 1$. In other words, P is a row-stochastic matrix.

2.1. Random Walk and Markov Chain

A random walk is a random process consisting of a sequence of discrete steps of fixed length, each in a random direction. In a graph G , a simple random walk is a sequence of nodes visited, where at each step, the next destination node is selected with a fixed probability. The probability of moving from a node i to node j in a simple random walk is given by the transition probability $p_{ij} = 1/d_i$ for all $j \in \Gamma^{(i)}$. A Markov chain, on the other hand, is a discrete-time stochastic process for which the previous states are irrelevant for predicting the subsequent states, given knowledge of the current state. A Markov chain describes at successive times the states of a system, and it is a memoryless system, i.e. it does not 'remember' the states it was in before and decides which future state it will transit purely based on the present state.

If the vertices of a graph can be thought of representing the state at time t , Y_t , in a finite state space, then a random walk on the graph represents a finite stochastic process, and can be modeled by a Markov chain. We can describe this Markov chain by the transition probability matrix of the graph, $P = [p_{ij}]$, as follows:

$$p_{ij} = Pr(Y_{t+1} = j | Y_t = i), \quad i, j = 1, \dots, n \quad (1)$$

The right hand side of the above equation is independent of t , and it shows that probability of moving to node j only depends upon node i (memoryless Markov property). This Markov chain is irreducible and aperiodic if the graph G is connected (any node can be reached from any other node in the graph) and aperiodic (greatest common divisor of all the cycle lengths in G is 1).

Let $\pi(t) \in R^n$ be the probability distribution of the state at time t . It represents probability of being at node N_i at time t . The state distribution satisfies the recursion that transpose of the distribution, $\pi(t+1)^T = \pi(t)^T P$. If P^t represents t -step probability transition matrix, then $\pi(t)^T = \pi(0)^T P^t$. For an aperiodic and irreducible Markov chain, the stationary distribution follows $\pi^T = \pi^T P$, and Motwani *et. al.* [10] shows that i -th element of this distribution, corresponding to node N_i , is given by $\pi_i = d_i/2m$ for all $1 \leq i \leq n$. This implies that for a long enough random walk, the resulting sampling distribution is dependent on the degree of the node d_i irrespective of its point of origin. Since degrees of nodes vary widely in a P2P network (in

other words, the network topology is an *irregular graph*), the sampling distribution is definitely non-uniform.

2.2. Uniform Sampling of Nodes from Graph Using Random Walk

It is clear from the discussion so far that a random walk on an irregular graph results in a non-uniform random sample of nodes depending upon the degrees of the nodes. To make the resulting distribution uniform, the probability transition matrix P needs to be modified to remove the bias due to irregular degrees of nodes. If π_U denotes the stationary uniform distribution, then for a graph G with $|V| = n$, $\pi_U = (1/n)\mathbf{1}$, where $\mathbf{1}$ represents a vector with all elements equals to 1. The random walk can be represented with the equation corresponding to Markov chain by $\pi_U^T = \pi_U^T P$, and plugging in the desired value of π_U , we get $\mathbf{1}^T = \mathbf{1}^T P$, which means sum of the column vectors of P is 1. Since, P is the transition probability matrix, it has to be a row-stochastic matrix as described earlier. The new condition implies that P is a doubly stochastic (row sum and column sum are 1). P is also a symmetric ($P = P^T$) and non-negative ($p_{ij} \geq 0$ for all elements of P) matrix. We summarize all the conditions need to be satisfied by the transition probability matrix P for a random walk of sufficient length to pick up a node uniformly in the following equation:

$$P\mathbf{1} = \mathbf{1}, \mathbf{1}^T P = \mathbf{1}^T, P \geq 0, P = P^T, \quad (2)$$

Mixing-Time of Markov Chain: The length of random walk necessary to converge to a stationary distribution is known as the *mixing time* of Markov chain. Mixing time of Markov chain depends upon the spectral properties of the transition probability matrix [3]. Let, τ denotes the mixing time of Markov chain and $\lambda_1, \dots, \lambda_n$ denote n distinct eigenvalues of P . Using condition given in 2 (the relation $\pi^T = \pi^T P$), π is the left eigenvector of P with eigenvalue 1 and the most dominant eigenvalue is given by 1. So, the eigenvalues of P can be represented by a non-increasing sequence $1 = |\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$. The quantity $|\lambda_2|$ or ‘Second Largest Eigenvalue Modulus’ (SLEM) governs the rate of convergence of a Markov chain. For a non-negative matrix like P , $|\lambda_2| = \lambda_2$. Sinclair [14] showed that the mixing time shares the following relation with eigenvalues and number of states n of the Markov chain

$$\tau = O\left(\frac{\log(n)}{1 - |\lambda_2|}\right) \quad (3)$$

Note that for a random walk on a graph $G(V, E)$, number of states n is nothing but number of vertices of G ($|V| = n$) and mixing time is the length of the random walk. From the above equation, it is evident that smaller the value of $|\lambda_2|$ (SLEM), smaller is the length of random walk necessary to converge to a stationary distribution.

The Metropolis-Hasting Algorithm: One of the algorithms that first tried to achieve uniform sampling using principle of random-walk is a simple adaptation of Metropolis-Hasting [11] algorithm. In this algorithm, node i collects the degree information from all of its immediate neighbors (node j where $j \in \Gamma^{(i)}$) before assigning transition probability, and computes transition probability for each edge based on the node with higher degree that is connected by that edge. For details of the algorithm, refer to [11], [1]. Experimental results shows that Metropolis-Hasting algorithm achieves uniformity [1] with an average length of random walk equals to $10 \cdot \log(n)$ where $n = |V|$ is the network size.

3. Uniform Random Sampling of Data Distributed in a Peer-to-peer Network

We now propose our algorithm for uniform sampling of data distributed homogeneously, but non-uniformly in a P2P network. By ‘homogeneous’, we mean that the data is horizontally partitioned and distributed (each peer has same set of attributes for their data tuple), and by ‘non-uniform’, we mean that each peer has different number of tuples. Note that we are interested in a uniform sample of data from a P2P network, which is different than the problem of uniform sampling of peers (nodes in a graph) described in the previous section.

3.1. Concept: Virtual Data Network

In this section, we first introduce the basic concept we use to take a uniform sample of data from P2P network. Our algorithm is inspired by Metropolis-Hasting algorithm [11]. Uniform sampling of data distributed in a network can be thought as a special case of uniform sampling of nodes in a network where every node has only one data tuple. The main idea is to create a virtual network with each nodes holding only one data tuple and then apply one of the available techniques of uniform node sampling on the virtual graph. The virtual graph can be created by replacing i -th node with n_i data tuples, with n_i virtual nodes, *fully connected* with each other and each connected with all the data tuples of neighbors of N_i (N_j s where $j \in \Gamma^{(i)}$). The links connecting two data tuples belonging to same real node is called *Internal* links. A random walk on such links does not involve any real communication. There are $n_i \times (n_i - 1)/2$ internal links along with n_i virtual nodes per peer N_i . Each actual link E_{ij} connecting nodes N_i and N_j in the real network, on the other hand, is replaced by $(n_i \times n_j)$ *External links*. Note walking along this link involves actually moving from one node to other node in the original graph. Degree of connectivity of each identical virtual node (data tuple)

belonging to peer N_i is given by $\sum_{j \in \Gamma^{(i)}} n_j + (n_i - 1)$. Total number of virtual nodes in the network becomes equal to total number of data tuples, or size of the total dataset $|X| = \sum_{i=1}^n n_i$. Let us denote this virtual network graph by $\overline{G}(\overline{V}, \overline{E})$, where \overline{V} represents set of all virtual nodes and \overline{E} represents set of all edges connecting these virtual nodes.

To sample any of the virtual nodes uniformly using random-walk on this virtual graph, the transition probability should satisfy the conditions given by Equation 2. To achieve these conditions, we introduce a transition probability matrix similar to the one proposed by Metropolis-Hasting algorithm[11]. Let us denote the set of virtual nodes belonging to real node N_i as $\{N_i\}$ and neighbors of a virtual node N_K by $\overline{\Gamma}^{(K)}$. A virtual random walk on the virtual graph $\overline{G}(\overline{V}, \overline{E})$ along the virtual edge \overline{E}_{KL} , where virtual node $K \in \{N_i\}$ and virtual node $L \in \{N_j\}$, uses the following transition probability to decide the next step :

$$p_{KL}^V = \begin{cases} 1/\text{Max}((n_{i,K \in \{N_i\}} - 1 + \sum_{g \in \Gamma^{(i)}, K \in \{N_i\}} n_g), \\ (n_{j,L \in \{N_j\}} - 1 + \sum_{h \in \Gamma^{(j)}, L \in \{N_j\}} n_h)) \\ \text{if } L \neq K \text{ and } \overline{E}_{KL} \in \overline{E} \\ 1 - \sum_{L \in \overline{\Gamma}^{(K)}} p_{KL}^V \text{ if } L = K \\ 0 \text{ Otherwise,} \end{cases}$$

Note that, $\sum_{g \in \Gamma^{(i)}, K \in N_i} n_g$ represents total amount of data contained by all the neighbors of real node N_i .

Clearly, the transition probability matrix given by the above equation, which is a $|X| \times |X|$ matrix, satisfies the conditions given by above equation, and a virtual random walk on \overline{G} of length $O(\log(|X|)/(1 - |\lambda_2|))$ will land on a virtual node with probability approaching $1/|X|$ where $|\lambda_2|$ is SLEM of the transition probability matrix P.

3.2. Sampling Algorithm: P2P-Sampling

Given the basic concept behind the uniform sampling technique, we now present our sampling algorithm for uniform random sampling from P2P network. We call this technique as ‘P2P-Sampling’.

Initialization: We assume that before the algorithm begins, each node N_i in the network discovers its immediate neighbor nodes $\Gamma^{(i)}$ (nodes with which it is directly connected), local data sizes of neighbors, N_j -s where $j \in \Gamma^{(i)}$, and total neighborhood data size of each of the neighbors, $\sum_{h \in \Gamma^{(j)}} n_h \forall j \in \Gamma^{(i)}$. Considering a stationary data distribution (where amount of data per node does not change over time in a P2P network), this information can be pre-computed and shared with immediate neighbors before the sampling procedure begins. Each node in its first round of communication discovers its immediate neighbor and its neighborhood data size by a point-to-point handshake message similar to ‘ping’. Every ‘send’ message contains

node id of the sender node, and is acknowledged by the immediate neighbor with an acknowledgement message containing the receiver node id and its own local datasize. Upon receipt of all the messages from all the neighbors, every node N_i computes the neighborhood data size as $\aleph_i = \sum_{g \in \Gamma^{(i)}} n_g$ and stores it. Only one arbitrarily selected node performs the sampling procedure by launching multiple random walks according to our algorithm. Let us name this node as ‘Source Node’, denoted by N_S . This part is shown in pseudocode 3.2.1 under the heading ‘*Initialization*’.

Algorithm: Note that, the concept of virtual network was introduced in previous section to convey the idea of selecting data tuple uniformly from the P2P network. To actually sample data tuples from real P2P network, we are interested in generating transition probability matrix for the real P2P networks which is equivalent to virtual transition probability matrix given by p^V . The important point to note for that is that a transition through any *internal link* (over an edge \overline{E}_{KL} , where both $K, L \in N_i$ for any real node N_i) is equivalent to going to another data tuple in the same node. Keeping that in mind, we transform the transition probability matrix for virtual network to the probability of transition to be used for random walk in a real P2P network. The probability of transition at any real peer N_i is given by :

$$p_{ij}^{p2p} = \begin{cases} n_i/(n_i - 1 + \sum_{g \in \Gamma^{(i)}} n_g) \\ \text{probability of going to another} \\ \text{data tuple randomly in current node } N_i \\ n_j/\text{Max}((n_i - 1 + \sum_{g \in \Gamma^{(i)}} n_g), \\ (n_j - 1 + \sum_{h \in \Gamma^{(j)}} n_h)) \\ \text{probability of going to a data tuple in node } N_j \\ 1 - \sum_{j \in \Gamma^{(i)}} p_{ij}^{p2p} \text{ probability of doing nothing} \\ 0 \text{ for } N_j\text{-s not in } \Gamma^{(i)} \end{cases}$$

The transition matrix given by the above equation is used by the random walk at its current position, node N_i , to decide about its next destination/step. A random walk following this transition probability picks up a data tuple after executing $O(\log|X|/(1 - |\lambda_2|))$ steps, and by virtue of the Markovian property, it is a uniformly sampled data tuple.

The pseudocode for P2P-Sampling is given in 3.2.1. Computation of transition probability by any node N_i requires the knowledge of local data sizes of neighbors, n_j -s for all node $N_j \in \Gamma^{(i)}$, and total neighborhood data size, \aleph_i , which is pre-computed during initialization. The total datasize of neighbors of immediate neighbors (second immediate neighborhood datasize over a particular edge E_{ij} , $\aleph_j = \sum_{h \in \Gamma^{(j)}} n_h$) can be shared during the actual sampling procedure. While the random walk is at any node N_k on its ℓ -th state and N_k wants to decide what should

be the next step according to probabilities given by p^{p2p} , it asks from all its immediate neighbors their neighbor data-size $\sum_{h \in \Gamma(j)} n_h, \forall j \in \Gamma_k$. After receiving the information from all its neighbors, N_k computes the probabilities of picking up another data tuple from itself or moving to any of its neighbors, or doing nothing other than increasing random walk-length counter to $(\ell + 1)$. The whole procedure is repeated until the walk reaches its L_{walk} -th state, where L_{walk} is the pre-specified limit of the random walk ($L_{walk} = O(\log(|X|))$, we will talk about how to decide about L_{walk} in next section). The data-point reached on L_{walk} -th state of the random walk is marked as a uniformly sampled data tuple.

Algorithm 3.2.1 Uniform P2P-Sampling

▷ **Initialization:** At each node N_i

for all $j \in \Gamma_i$ where n_j is NOT known **do**

 Send message to N_j containing sender id and data size i, n_i .

 Receive response containing local datasize of j, n_j .

end for

 Calculate neighborhood datasize $\aleph_i = \sum_{g \in \Gamma(i)} n_g$.

▷ **P2P-Sampling:**

 Current node $N_i =$ Source Node N_S

 Walk length $\ell =$ Walk limit $L_{walk} = O(\log(\text{Estimated Total Datasize}))$.

while $\ell > 0$ **do**

for all $j \in \Gamma_i$ **do**

 Get neighborhood datasize \aleph_j .

 Calculate transition probability to node $N_j, p_{ij} = n_j / \text{Max}((n_i - 1 + \aleph_i), (n_j - 1 + \aleph_j))$.

end for

 Calculate probability of picking up another data tuple from $N_i, p_{ii} = n_i / (n_i - 1 + \aleph_i)$.

 Calculate probability of ‘No Change’, $p_{i0} = 1 - p_{ii} - \sum_{j \in \Gamma_i} p_{ij}$.

 Decides the next state based on transition probability p_{ix} .

if Next state is N_j **then**

 Set current node $N_i = N_j$.

end if

 Reduce walklength $\ell = \ell - 1$.

end while

 Mark one data tuple $x_i \in X^{(i)}$ uniformly randomly in current node N_i as a sample.

x_i is sent to source node N_S via shortest path in P2P network.

▷ **Data Sampling:** P2P Sampling is repeated $|s|$ times by source node N_S to collect a uniform sample of size $|s|$.

To collect a data sample of size $|s|$, N_S launches $|s|$ such random walks, each of length L_{walk} and collects one data tuple per random walk. Each random walk marks the data tuple at which it terminates as a sample, and then they are transmitted to the source node N_S via direct point-to-point communication between the owner node of that data tuple and N_S .

3.3. Communication Topology for Efficient Sampling

Our proposed P2P sampling procedure is conceptually equivalent to a Markov-chain on a virtual graph with number of nodes equals to total data size in the network, $|X|$. It is already known that the length of the random walk necessary for uniform sampling is $O(\log|X|/(1 - |\lambda_2|))$. Now we derive a bound for the ‘spectral gap’ of the transition probability matrix P to limit the desired length of the random walk. Estimating the second largest eigenvalue of the entire transition probability matrix (λ_2) to evaluate the spectral gap is a hard problem. We will not delve into how to estimate λ_2 exactly here. Instead, we will show how to bound the inverse of the spectral gap of P .

Bounding the Spectral Gap: The transition probability matrix P is a double stochastic matrix, and hence, the dominant most eigenvalue is 1, with $\mathbf{1}^T$, the row of ones, as its eigenvector. We are looking for an upper bound for second largest eigenvalue λ_2 with column eigenvector X . Let us form the column vector C by taking the largest element from each row of P . Each element of C , the maximum row-element of P , is nothing but the probability of going to another virtual node (data tuple) belonging to same peer, i.e. $1/(n_i - 1 + \sum_{g \in \Gamma(i)} n_g)$ for a virtual node belonging to peer N_i with n_i data tuples.

Given, $\mathbf{1}^T X = 0$ and $PX = \lambda_2 X$, we can write $(P - C\mathbf{1}^T)X = \lambda_2 X$. Taking Gerschgorin disks from the column sums, we see that the column sums of P are unity and those of C are all equal and equal to the sum of the maximum elements of each row. Thus

$$\begin{aligned}
 |\lambda_2| &\leq \left(\sum_{i=1}^{|X|} C_i \right) - 1 \\
 &\leq \sum_{i=1}^n \frac{1}{1 + \sum_{g \in \Gamma(i)} n_g / n_i} - 1 \quad (4)
 \end{aligned}$$

Note that the sum is transformed to sum over the number of peers (n) in P2P network as there are n_i identical virtual nodes per peer N_i , each having same maximal transition probability. Let us denote the ratio $\sum_{g \in \Gamma(i)} n_g / n_i$ for node N_i as ρ_i , the ratio of total neighborhood data and local data. This is an important factor in deciding about the walk length necessary to achieve uniformity using P2P-Sampling. From Equation 4, a lower bound of spectral gap can be derived as $(1 - |\lambda_2|) \geq (2 - \sum_{i=1}^n \frac{1}{1 + \rho_i})$. Now, if ρ_i is above a specified threshold value, $\hat{\rho}$, for all the peers in the network ($1 \leq i \leq n$), then the entire sum, $\sum_{i=1}^n 1/(1 + \rho_i)$ is upper bounded by the term $n/(1 + \hat{\rho})$. That gives the following bound:

$$\frac{1}{1 - |\lambda_2|} \leq \frac{1}{2 - n/(1 + \hat{\rho})} \quad (5)$$

Equation 5 can be used to decide about the length of the random walk in P2P Sampling. Since it is known that the length of the walk is $O(\log(|X|)/(1 - |\lambda_2|))$, the random walk with length $L_{walk} = O(\log(|X|))$ will select a data tuple uniformly randomly upon termination provided for all peers $\hat{\rho} = O(n)$.

Effect on Communication Topology: This above condition implies that the neighborhood datasize for each peer has to be at least $O(n)$ times the local datasize. Thus, to achieve uniformity by a random walk of length $c \cdot \log(|X|)$ (c is a small integer constant), each peer N_i where the random walk lands, needs to discover neighbors until $\rho_i = O(n)$. This is how the ‘communication topology’ of each peer is formed. Then the peer decides about the the next step of the random walk based on our proposed algorithm.

In a real world P2P network, the data distribution often follows power law distribution[13], with very few peers sharing most of the data, and most of the peers sharing very little data. In such a network, peers sharing very few data can achieve the $O(n)$ ratio easily by forming communication link with few of the peers sharing most of the data. In that case, the communication topology of the P2P network takes the form of a central hub consisting of few peers sharing most of the data, and rest of the peers sharing few data are directly connected to this hub. A random walk in such network is likely to enter the ‘data hub’ quickly as most of the virtual nodes are either directly connected to the hub, or belong to the hub. Once in, the walk also stays inside the hub longer as larger the local datasize, more the probability of picking up another data tuple from the same peer.

For a power law data distribution, the peers forming the ‘hub’ will have difficulty achieving the $O(n)$ bound for data ratio threshold though, since their own datasize (the denominator of the ratio ρ_i) is large. As a solution to this problem, we suggest to divide each peer holding large amount of data into a number of virtual peers fully connected with each other, each holding smaller amount of data, so that the $O(n)$ ratio threshold can be achieved by each virtual peer. Note that this just creates some extra virtual links and a walk through these links does not incur any real communication.

Although total datasize ($|X|$) may not be known to the node running the sampling a priori, it does not pose a big challenge in estimating L_{walk} . An overestimate of total datasize works just fine as the effect of it is logarithmic on the entire walklength. For example, an overestimate of 1G data for a network holding 1M data just affects the walklength by $3 \cdot c$ extra steps for a random walklength of $L_{walk} = c \cdot \log(|\bar{X}|)$ where $|\bar{X}|$ is an estimated upper bound of total datasize. An underestimate is not a big problem either, as long as it is not too small ($< 0.1\%$ of the actual datasize).

3.4. Analysis of Communication Cost

The communication complexity of the sampling procedure has two components (1) communication cost to discover the data tuple by random walk and (2) communication cost to transport the data tuple sampled to the node operating the sampling procedure. We assume that the second part happens by direct point-to-point connection when source node id and destination node id are known and is taken care of by network protocol. Here we just analyze the communication cost for discovering the sample data tuples.

In our formal analysis of communication, we do not take in to account communication due to sender and receiver id contained in every message as that is assumed to be taken care of at the network protocol adapted for communication. We assume during sample discovery, the random walk just contains source node id (integer, 4 bytes) and current walk-length counter (again integer, 4 bytes).

During the initialization phase of P2P Sampling, each node exchanges its local datasize with all its immediate neighbors, which accounts for 2 integers exchanged per edge (for datasizes of two nodes at two ends of the edge). Total communication is thus $(2 \times |E| \times 4)$ bytes, where $|E|$ denotes total number of edges in the entire network.

After the source node launches the random walk with length L_{walk} , and the walk lands on a new node N_k on its ℓ -th step, it needs to receive the datasizes of d_k immediate neighbor nodes to decide about the transition probability of the next step (d_k is the degree of connectivity of node N_k), which involves $(d_k \times 4)$ bytes communication. Based on the transition probabilities, if the random walk lands in a new node in the next step, it carries 8 bytes (2 integers) with it to a new node. The random walk actually takes $\bar{\alpha} \times L_{walk}$ real steps ($0 \leq \bar{\alpha} \leq 1$), where $\bar{\alpha}$ represents average probability of taking an actual (*external*) link in one step of the random walk. So, total communication necessary to discover a uniformly sampled data tuple from the P2P network is $(\bar{\alpha} \times c \cdot \log(|\bar{X}|) \times (\bar{d} + 2) \times 4)$ bytes, where \bar{d} is average degree of the nodes in the network. Average degree of even a power law graph is usually constant[12], which lets us assume \bar{d} is a constant. Thus, P2P sampling discovery of one data tuple from the P2P network requires $O(\log(|\bar{X}|))$ bytes of communication over the entire network.

4. Experimental Results

We experimented P2P-Sampling in a simulated P2P environment using P2P network topology created by network simulator BRITE. As pointed out by [13], most of the real world P2P systems follow a power-law topology where degrees of connectivity of the peers follow a power-law distribution. That is why we used *Router-Barabasi-Albert* model in BRITE to generate the network topology. Default set-

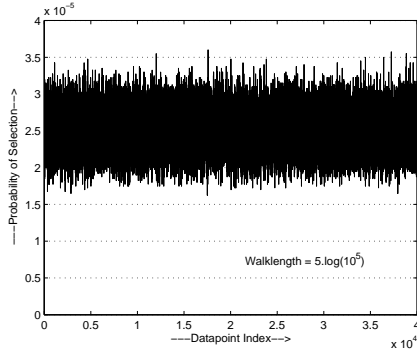


Figure 1. Probability of selection of data tuple in a 1000-node P2P network with 40,000 data, distributed following power law. Theoretical uniform probability of selection is 2.5×10^{-5} .

ting (Refer to BRITE documentation for details available at www.cs.bu.edu/brite) is used with BA model in BRITE to generate a P2P network topology with 1,000 ($= n$) peers altogether. One node is selected arbitrarily as source node that launches the random walks to collect uniform sample of data.

To evaluate uniformity of P2P-Sampling algorithm, we measure the KL-distance between theoretical uniform distribution (theoretical probability of selection of each data tuple is $q_i = 1/40,000 \forall i$) and actual distribution observed by our sampling experiments¹. We run our algorithm on the 1,000-node network topology described earlier with 40,000 data distributed over all the nodes following power law distribution with a power law coefficient value of 0.9. Random walk length is set at $L_{walk} = 25$, with an estimated datasize of 100K ($|\bar{X}| = 100,000, c = 5$ and $L_{walk} = c.log(|\bar{X}|)$). To get the experimental distribution, we count frequency of selection of each data tuple for multiple sampling run over the entire data and converted that to average probability of selection of each data tuple. Figure 1 shows the experimental selection probability of each data tuple with these conditions. The KL-distance between this distribution and theoretical distribution comes out to be 0.0071 bits, which shows that experimental distribution is very close to theoretical uniform distribution.

To verify experimentally whether uniformity is guaranteed irrespective of nature of the underlying distribution of data in the network, we run P2P-Sampling with different types of underlying data distribution in a 1000-node P2P network with total 40,000 data. For each type of distribution, we distribute data once with correlation to the degree

¹If p_i denotes the experimental probability of selection of i -th data tuple and q_i denotes the theoretical probability for a distribution with n points, then $KL(p, q) = \sum_{i=1}^n p_i.log_2(p_i/q_i)$ bits

of the nodes (nodes with highest degree gets maximum data and so on), and once without any correlation (data gets distributed randomly irrespective of the degree of the nodes). We experimented with power law distribution with coefficients 0.9 (heavy-skew) and 0.5 (lighter skew), exponential distribution with parameter 0.008 (so that each of the 1000 nodes gets some data), normal distribution with mean 500 and standard deviation 166 and random distribution (total data is randomly distributed amongst 1000 nodes). KL-distance is shown for different distributions in Figure 2 as a measure of uniformity as before. It is clear that irrespective of distribution of data, P2P-Sampling achieve very good uniformity, and it is not affected by any correlation between degree of nodes and its datasize.

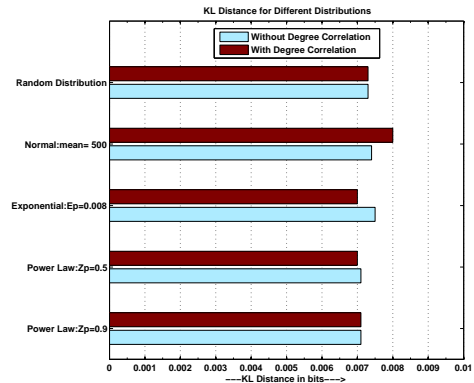


Figure 2. KL distance between theoretical uniform distribution and experimental distribution for different underlying distribution of data with and without any correlation to degree of the nodes.

Finally, we report the average number of real communication steps taken by random walk in our experiments to discover a random sample with different types of data distribution. The data distributions used are the same as mentioned in last paragraph. Random walk length is set at $L_{walk} = 25$ as before, and the actual number of communication steps (when the walk goes from one real node to another real node, that is considered as one real communication step) taken by the walk on an average is expressed as a percentage of the specified walk-length and reported in Figure 3. It is observed that for all the data distributions, that the random walk takes less than 50% of pre-specified steps (L_{walk}) on an average to discover a random sample, irrespective of the data distribution. For highly skewed distribution like power law and exponential, if the nodes having higher degree gets more data (data distributed with degree correlation), the average number of steps required is more in comparison to random data distribution. We believe this

is due to the fact that if the random walk lands up in a node with small amount of data and small degree of connectivity, it is almost certainly going to a different node in its next step, causing real communication.

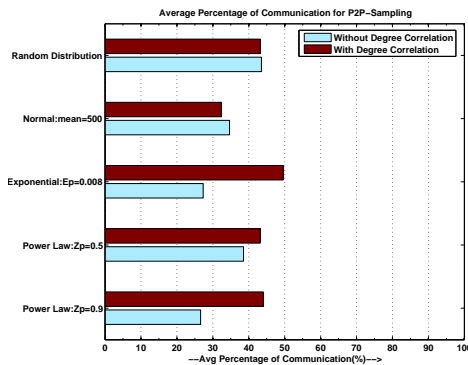


Figure 3. Average number of real communication steps taken by random walk expressed as a percentage of pre-specified length of random walk (L_{walk}) for different types of underlying data distribution.

5. Conclusion

In this paper, we have addressed the problem of uniform data sampling in a P2P network where data is distributed homogeneously and non-uniformly amongst different peers. Our sampling-technique is a random-walk based approach where a random walk is launched by a node to collect a data tuple uniformly randomly from the entire network. The non-uniform datasizes and varying degrees of connectivity for different peers in a P2P network causes a biased sample if standard random walk is used for sampling. That is why we propose our sampling algorithm ‘P2P-Sampling’, which can take care of the bias due to varying datasizes and degrees of different peers at each transition step of random walk and pick up a data tuple uniformly within a finite walklength. We also provide a bound on the spectral gap of the transition probability matrix and use that to provide necessary connectivity condition for each peer in a P2P network. Experimental results show that our proposed sampling method achieves uniformity within acceptable tolerance. Given the sample data discovery process takes communication of the order of logarithm of total data contained in the entire network, we believe P2P-Sampling provides a very handy tool for analyzing data distributed in a P2P network.

6. Acknowledgement:

This work is supported by CAREER award IIS-0093353. We specially thank Chris Giannella (affiliated with Department of Mathematics and Computer Science, Goucher College, Baltimore, MD) for his valuable contributions.

References

- [1] A. Awan, R. Ferreira and S. Jagannathan. Distributed uniform sampling in unstructured peer-to-peer networks. In *proceedings of 39th International Conference on System Sciences*, Kauai, Hawaii, 2006.
- [2] B. A. Bash, J. W. Byers, and J. Considine. Approximately uniform random sampling in sensor networks. In *proceedings of the 1st Workshop on Data Management for Sensor Networks*, August, 2004 (<http://db.cs.pitt.edu/dmsn04/>).
- [3] P. Brmaud. *Markov Chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer Verlag, 1999.
- [4] C. Cooper, M. Dyer, and C. Greenhill. Sampling regular graphs and a peer-to-peer network. In *proceedings of the 16th annual ACM-SIAM symposium on Discrete algorithms*, pages 980–988, Philadelphia, PA, USA, 2005.
- [5] S. Datta, C. Giannella, and H. Kargupta. K-means clustering over a large, dynamic network. In *proceedings of the 2006 SIAM International Conference on Data Mining (SDM’06)*, April 2006.
- [6] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks. In *proceedings of IEEE INFOCOM, 2004.*, pages 120–130, 2004.
- [7] M. Henzinger, A. Heydon, M. Mizenmacher, and M. Najork. On near-uniform url sampling. In *proceedings of the World Wide Web Conference*, 2000.
- [8] Kowalczyk W., Jelasity M., and Eiben A. Towards Data Mining in Large and Fully Distributed Peer-To-Peer Overlay Networks. In *proceedings of BNAIC’03*, pages 203–210, 2003.
- [9] L. Lovasz. Random walks on graphs: A survey. In *Combinatorics Inos Bolyai Mathematical Society, Budapest*, pages 353–398, 1996.
- [10] R. Motwani and P. Raghavan. Randomized algorithms. *ACM Computing Survey*, 28(1):33–37, 1996.
- [11] N. Metropolis, A. W. Rosenbluth and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(2):1087–1092, 1953.
- [12] P. Orponen and S. E. Schaeffer. Efficient algorithms for sampling and clustering of large nonuniform networks. *Technical Report, cond-mat/0406048*, 2004.
- [13] S. Saroiu, K. P. Gummadi, and S. D. Gribble. Measuring and analyzing the characteristics of napster and gnutella hosts. In *Multimedia Systems.*, 9(2):170–184, 2003.
- [14] A. Sinclair. Improved bounds for mixing rates of marked chains and multicommodity flow. In *proceedings of the 1st Latin American Symposium on Theoretical Informatics*, pages 474–487, London, UK, 1992. Springer-Verlag.
- [15] Z. Bar-Yossef, A. Berg and D. Weitz. Approximating aggregate queries about web pages via random walks. In *proceedings of the 26th International Conference on Very Large Databases (VLDB)*, pages 535–544, 2000.