Sampling Large Complex Networks: via Maximal Entropy Random Walk

Joon-Hyuk Ko
Dept. of Physics
Astronomy
Seoul National University
jhko725@snu.ac.kr

Sung Hoon Kim
Dept. of Physics
Astronomy
Seoul National University
josephkim11@snu.ac.kr

December 5, 2019

Abstract

In order to analyze properties of a large graph, direct calculation is often expensive, even inhibitive at times. In such situations, graph sampling algorithms provides an efficient alternative. In this paper, we present maximal entropy random walk sampling, whose unique statistical property enables it to effectively extract eigenvector centrality information from the original graph. We also perform a comprehensive benchmark on the this new algorithm, and find that in addition to far outperforming contesting algorithms in node centrality estimation, the algorithm can also well sample the degree distribution and the degree-clustering coefficient distribution of the original graph.

1 Introduction

With the advent of big data, real world graphs are getting ever larger, and hence, the traditional approach of analyzing the entire graph is becoming more and more prohibitive. This is especially the case for research involving the web graphs, as the number of nodes and edges for such graphs verges on the order of millions. (For example, the 2012 YahooWeb contains 1413M nodes and 6636M edges[1].)

Faced with such a problem, there are two avenues of resolution. The first is to analyze the global graph properties in a distributed manner, via frameworks such as MapReduce. While this method directly provides the exact solution to the problem, distributed computations require dedicated computing clusters, which maybe unavailable depending on circumstances.

An alternate, indirect method of solving the large graph problem is by graph sampling. The expensive computation of calculating the global graph properties is circumvented, and rather samples of the original graph are drawn and their properties investigated to infer the characteristics of the original graph.

As is the case with any other sampling task, the most important part of graph sampling is the choice of the sampling algorithm, as the quality of the samples drawn and hence the accuracy of the estimated global graph properties depend heavily on the sampling methodology used. Quite naturally then, the goal for any research regarding graph sampling would be to "find the best sampling algorithm". However, there is a caveat here: the concept of the "best" sampling algorithm is ill-defined because there is no universal sampler that returns the best estimate for every metric. Thus in order to discuss the "best" sampling algorithm, one must first state one's objective, or the graph property of interest.

Pioneering research on this subject was performed by Leskovec and Faloutsos[2], where 10 different sampling algorithms were benchmarked using 14 types of graph properties to gauge which algorithm most accurately describes the global graph for a given sample size. It was shown that while there are universally bad performing algorithms, the best sampler depends on the performance metric. For instance, simple random walk sampling was found to best estimate the clustering coefficient, whereas forest fire sampling best estimated the hop plot of the graph.

In this research, our graph property of interest is the node centrality, which is a measure of a node's influence in the graph. Of course, "influence" can be defined in many different ways, resulting in different centrality measures, such as degree centrality or eigenvector centrality. In this paper, we focus on the eigenvector centrality which, qualitatively, can be interpreted as defining "important" nodes as those connected to other "important" nodes, and is quantitiatively defined as the values of the components of the first eigenvector of the adjacency matrix.

Eigenvector centrality has been shown to be an effective metric in datamining, with PageRank algorithm and Katz centrality being variants of eigenvector centrality. Understanding the eigenvector centrality scores is important in cases where the data transfer in a network occurs over a large effective distance. For instance, recent studies in brain network have shown the role of eigenvector centrality in analyzing responses of the brain to the external stimuli[3], and PageRank has cemented its status as an indispensable tool in Web data mining.

In addition to our objective, we restrict our research to the situations in which global access to the entire graph is not possible. This inaccessibility can arise due to multiple reasons - the graph maybe too large to fit in memory, or one may have only a partial knowledge of the graph. For instance, when analyzing social media graphs, such as Twitter, not only would it require dedicated hardware to grant random access to all the nodes and edges of the graph, but the researcher would be barred from obtaining the data due to company policies.

Our restriction narrows down the search space for the "best" algorithm considerably, as many of the popular sampling algorithms become no longer viable. Random node/edge algorithms cannot be performed due to the absence of the list of node/edge indices and PageRank based heuristics cannot be performed as well since the computation of the scores require the adjacency matrix information. The remaining applicable sampling methods are of the traversal-based sampling algorithms, such as random walk-based algorithms or heuristics-based algorithms such as forest fire sampling.

In this research, we focus on a type of random walk algorithm called maximal entropy random walk and compare its performance with respect to other popular algorithms. First introduced in [4], maximal entropy random walk drew interest from the physics community due to its connections with the path-integral formulation of quantum mechanics[5]. However, we view this algorithm from an alternative viewpoint - as a possible candidate for a graph sampling algorithm that can effectively estimate the eigenvector centrality of a given graph.

In short, the problem we want to solve is the following:

- GIVEN: A large graph to be sampled, but with only local information accessible. That is, given a node in a graph, its neighbors can be accessed, but the entire adjacency matrix is unavailable. In principle, one could recursively parse the node neighbors to obtain the adjacency matrix, but the graph is assumed to be too big for such an approach.
- IMPLEMENT: A new physics-based sampling algorithm - Maximal Entropy Random Walk Sampling
- VERIFY: Whether the MERW sampler shows superior performance in predicting the eigenvector centrality of the given graph.
- ANALYZE: How well the MERW sampler estimates the global property of the given graph, by comparing MERW sampling with existing competitors such as Forest Fire sampling, or General Random Walk sampling. Various graphs - both real and synthetic - will be used for the benchmark.

The contributions of this project are the following:

- We bring focus on maximal entropy random walk, which is an algorithm that has deep physical implications due to its connection with path-integral formulation of quantum mechanics.
- We provide a new, practical viewpoint for the maximal entropy random walk, that it can be used to accurately recover the eigenvector centrality of a given graph.
- We implement a local sampling algorithm based on maximal entropy random walk, extending the theoretical analysis provided in [6].
- We verify that the algorithm show superior performance in estimating eigenvector centrality, and also investigate its performance for sampling other graph properties such as the degree distribution or the clusting coefficient.

2 Survey

Next we list the papers that each member read, along with their summary and critique.

2.1 Papers read by Joon-Hyuk Ko

The first paper was *Sampling from Large Graphs* by Leskovec and Faloutsos [2]

- *Main idea*: The paper points out that with graphs getting larger and larger, it is inevitable for researchers to sample graphs. Different methods of sampling were compared against each other with respect to two criteria - how well they represent the scaled down version of the graph, and how similarly they resemble the past version of the global graph.
- *Use for our project*: As this is one of the first papers to address the question of graph sampling, it is inevitable for us to reference this paper in our work. Also the set of graph properties this paper used to assess the performance of sampling algorithms will also be used in our research.
- *Shortcomings*: While the paper does provide a comprehensive analysis on the performance of different sampling algorithms with respect to a wide array of graph properties, the work is mainly heuristic, and thus is unable to provide a satisfying explanation as to why a particular sampling algorithm samples one class of graph properties well, but lacks accuracy for another class of graph properties. Also, not all the algorithms presented in this paper is applicable when only local access is possible, a point not mentioned in this paper.

The second paper was *Reducing Large Internet Topologies for Faster Simulations* by Krishnamurthy, Faloutsos, Chrobak, Lao, Cui, and Percus. [7]

- *Main idea*: This paper also focuses on the task of reducing a large graph, and assessing how much graph properties are preserved as the graph size is reduced. Again, different graph reduction/sampling strategies were investigated, and some graph parameters such as the average degree or the rank exponent was plotted as a function of the reduced percentage of the original graph.
- *Use for our project*: The plot of graph parameters with respect to the sampled graph size will also be one of the aspects of the sampling algorithms we will be investigating in this research as well. The relation between the sample size and the algorithm runtime is briefly mentioned, which is will be one of the key criteria we will use to assess the different type of sampling algorithms. Also the paper attempts to explain analytically why some sampling strategies can preserve the power law exponent of the original graph. While the exact theory/mathematics will be different, this kind of qualitative analysis of the sampling algorithm is what we aim to pursue in our research as well.
- *Shortcomings*: As can be deduced from the title, the main class of sampling algorithms investigated in this paper is of the reductive type, that is, algorithms that start with the original graph and whittle it down to the desired size. Unfortunately this class of algorithms is ineffective when trying to deal with graphs that are too big/expensive to fit in memory, so in our research, we will not consider algorithms of this type.

The third paper was *Metropolis Algorithm for Representative Subgraph Sampling* by Hubler, Kriegel, Borgwardt, and Ghahramani. [8]

- *Main idea*: The goal of the paper is to develop a graph sampling algorithm that can draw representative subsamples from the original graph. To accomplish the goal, the problem was recast as an optimization problem of finding a subgraph that minimizes some objective function between the original graph and the sampled subgraph. This objective function is designed so that it drives the sampled graph to assume the same value of some graph property of the original graph. This optimization problem was then solved via Metropolis-Hasting random walk algorithm and the results were analyzed for performance.

- *Use for our project*: As this paper also uses a random walk based algorithm for sampling, this paper is worth looking into. In fact, as a comparison against the maximal entropy random walk based sampling that will be presented in our walk, a Metropolis-Hastings random walk sampler will also be implemented and evaluated.
- *Shortcomings*: While the idea of using Metropolis-Hastings random walk for graph sampling purposes deserves some merit, the exact form of the transition probability (which is related to the objective function) used in this paper is impractical, as its calculation requires knowing the some desired property of the global graph. Thus the algorithm is caught in a logical self-loop: in order to draw a sample that accurately predicts some graph property, that graph property needs to be known. In our research, the Metropolis-Hastings random walk sampler we implement will use a different objective function (presented in []), bypassing this issue.

The fourth paper was *A Survey and Taxonomy of Graph Sampling* by Hu and Lau [9]

- *Main idea*: This paper is a review paper of the research on graph sampling, so there is no ”main idea” that is presented by this paper - rather a comprehensive summary of the different sampling algorithms and their characteristics, as reported in the literature, is given. However, the paper does organize the various types of sampling algorithms into different taxonomies based on their similarities, which is quite instructive.
- *Use for our project*: During our research, this paper will serve as a ”dictionary” of previous literature in case we need such information. The taxonomy presented in this paper has also helped us in narrowing down the scope of the paper to traversal-based sampling methods only, as only these can address the case in which global access to the graph is not possible. Also some of the sampling methods presented in this paper will be implemented by us as well and compared against our algorithm.
- *Shortcomings*: As this is a review article and not a research paper, there are no major shortcomings.

2.2 Papers read by Sung Hoon Kim

The first paper was *Network Sampling From Static to Streaming Graphs* by Ahmed, Neville, and Kompella. [10]

- *Main idea*: The authors bring up the problem that many of the previously introduced algorithms implicitly assume that the graph can be fitted in the local memory, and that global access is possible - which is no longer possible when the graph size becomes large. Therefore the authors propose streaming such large graphs, and propose strategies for graph sampling for streaming graphs. These strategies are then later evaluated in terms of their accuracy with respect to the preservation of the global network statistics.

- *Use for our project*: We share the notion that the relevant sampling methods are those that can operate without the graph being entirely in local memory, and investigate sampling methods that satisfy this criteria. Also this paper provides various ways to compare the distributional properties of the original graph and the sampled graph, which will be employed in our work.
- *Shortcomings*: The main premise of the paper is that the graph is being streamed, and the question addressed is how does one choose a specific edge/node in the current window of the stream. On the other hand, the question of what is the best way to stream the graph itself is out of scope of this paper (as it is assumed to be given) and is not addressed. In our work, we do not consider the streaming/traversing strategy as given, and combine the sampling and streaming procedure into one problem, as finding a good traversal-based sampling algorithm.

The second paper was *Localization of the maximal entropy random walk* by by Burda, Duda, Luck and Waclaw. [4]

- *Main idea*: In simple random walk, the transition probability of the random walker from a node it one of its neighbors is constant for all neighboring nodes. However, one can define a different type of random walk, by choosing a set of transition probabilities that assigns equal statistical weight to two paths of the same length between a fixed starting and endpoint. In this paper, the formula for the transition probability of this random walk is derived and the behavior of the random walker is investigated for the case of regular lattice with sparse defects.
- *Use for our project*: As one of the goals of this project is to develop a new sampling strategy that has novel theoretical support from physics, this concept of maximal entropy random walk is quite appealing. In fact, since this type of random walk maximizes the entropy production of the random walker, this means that the random walker acquires maximal information about the structural property of the graph - a desired property for a sampling algorithm. Hence in this project we will, for the first time in literature, employ maximal entropy random walk to the graph sampling problem.
- *Shortcomings*: The algorithm for the maximal entropy random walk, provided in this paper, requires the entire adjacency matrix of a given graph. While this information is available when simulating the dynamics of the random walker on a relatively small lattice - as was done in this paper - it is often not available when trying to sample large graphs. Thus the proposed algorithm must be modified so that only local node information is used for the transition probabilities. Fortunately, such modification is introduced in [6].

The third paper was *The various facets of random walk entropy* by Burda, Duda, Luck and Waclaw. [5]

- *Main idea*: In this paper, the properties of the maximal entropy random walk is investigated in depth, with connections made to physics theories, in particular path integral formalism and quantum mechanics. This is because just as maximal entropy random walk assigns equal statistical weights to paths of the same length between a fixed start and end point on the graph, the path integral formalism assigns the same value of "action" for paths of same length between a fixed start and endpoint in space. Here, "action" is a fundamental physical quantity which is related to the probability of the particle selecting a particular path, and is analogous to the statistical weight of a given random walk path. Armed with this connection to physics, the authors investigate the behavior of this random walker on graphs, with emphasis on regular lattices and lattices with defects.
- *Use for our project*: The in-depth physical analogues made in this paper will be incorporated into our project, in order to provide a qualitative/quantitative interpretation to the motion of the maximal entropy random walker as well as the quality of the graph sample generated by the sampling methodology. Also, results on the motion of the maximal entropy random walker on regular lattices will serve as a reference while investigating the motion of this random walker on generated/real graphs, a point this paper aims to treat.
- *Shortcomings*: While extensive inquiry is made on the motion of the maximal entropy random walker on lattices - which are of paramount interest in physics since they are intimately related to the study of solids - no analysis is performed for random graphs, which are the prevailing types of graph one encounters outside of the field of physics. We aim to provide such an analysis in our paper.

The fourth paper was *Maximal-entropy random walks in complex networks with limited information* by Sinatra et al. [6]

- *Main idea*: According to the original formulation of the maximal entropy random walk, as proposed by [4, 5], the transition probability of this process involves both the components and the maximum eigenvalue of the adjacency matrix of the graph. This indicates that global information about the graph is required to perform maximal entropy random walk, which is often hard to obtain. In this paper, the authors propose a way to circumvent the problem by assuming no degree correlations between the nodes of the graph, then modifying the original transition probability so that it only requires the local information available at each node. Also the authors show that this no degree correlation assumption can be relaxed to first neighbor correlations and so on, and provide a guideline as to how to further modify the transition probability to suit each case.
- *Use for our project*: Since the goal of the project is to sample the graph via maximal entropy random walk and using only local information, the original formulation of the maximal entropy random walk is incompatible with our project. Therefore in our sampling algorithm, we will use the localized version of transition probabilities for the maximal entropy random walk, provided in this paper.

- *Shortcomings*: While the paper does present a localized version of the transition probabilities even for the case of degree correlations, that formula is not straightforward as it requires ν, the characteristic exponent of the degree correlation. ($< d_j >\propto d_i^{\nu}$, where nodes j are the first neighbors of node i, d denotes the degree, and the brackets indicate the average.) In order to implement the algorithm for this case, estimation of ν is unavoidable, a point overlooked and not addressed in this paper.

3 Proposed Method

3.1 Why Maximal Entropy Random Walk Sampling?

Random walk based sampling algorithms has been a a popular choice for a graph sampling algorithm in literature due to its simplicity and good performance in terms of retrieving the global information of the graph. However, it has a potential drawback: it is biased towards sampling high degree nodes. This is due to the fact that available number of path for passing a given node equals square of its degree. The powerlaw-like sampling of a random graph reported by Lakhina, et.al.[11], directly elucidates this behavior, where they pointed out that traceroute-like methods induces long tail in degree distribution of the sampled graph even when the original graph is a random graph. In other words, a high degree nodes acts like a basin of attraction for the random walker.

By modifying the transition probability of the random walk, this tendency can be alleviated. This is the key idea of the Metropolis-Hastings random walk algorithm, which grants one the power to assign any distribution to the random walk's stationary probability distribution. For instance, by assigning the transition probability from node i to j as $\pi_{ij} = min\{1, d_i/d_j\}$, one can make the random walker visit all nodes with equal probability - that is, assign uniform distribution over all the nodes as the stationary distribution.where the walker has a negated weight against the higher degree nodes.

Of course, whether this random walk is optimal or not entirely relies on the objective of the sampling. In terms of the uniformity of the nodes sampled, this is obviously optimal. However, when it comes to the connected structure of the given graph, simple random walk outperforms Metropolis-Hastings random walk. This intuitively makes senses, as one would expect important graph features, such as triangles, to be centered around the high degree nodes, or nodes of high centrality. In other words, while Metropolis-Hastings is optimal in uniformly sampling nodes, it is not effective in capturing the connected structure of the network.

Enters maximal entropy random walk(MERW) as a strong contender for effectively capturing the eigenvector centrality of the network. Generally, the probability of a random walker to move along the t-step trajectory $\gamma_{i_0 i_t}^{(t)}$, passing through the nodes $(i_0, i_1, ..., i_t)$ is given by

$$P(\gamma_{i_0 i_t}^{(t)}) = P_{i_0 i_1} P_{i_1 i_2} \cdots P_{i_{t-1} i_t} \tag{1}$$

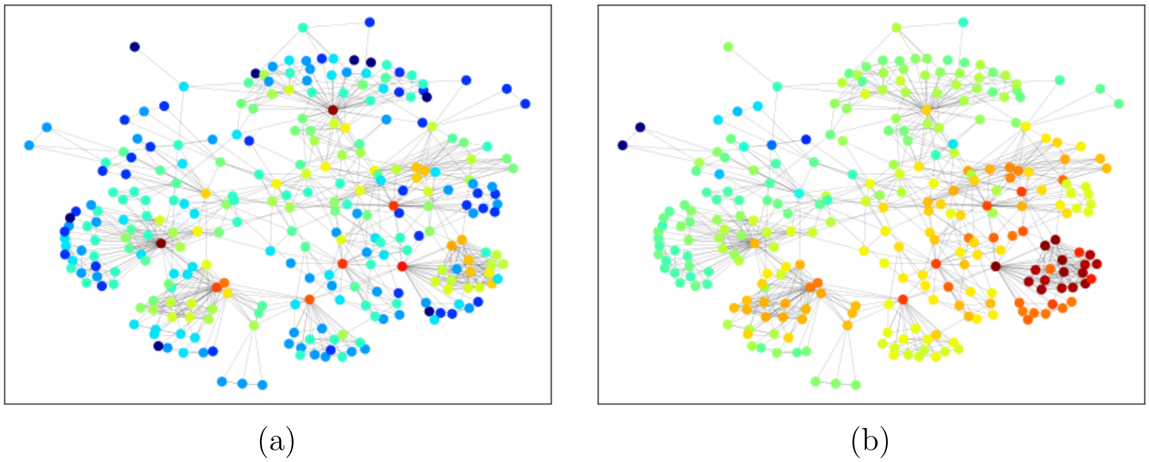
The equation indicates that this probability depends on all the intermediate nodes, and hence two different paths from i_0 to i_t with the same length t, have different probabilities of being traversed.

In contrast, maximal entropy random walk is a special type of random walk that allows the any two different paths from i_0 to i_t with the same length t to have the same probability. While this random walk has been studied from physicists' point of view, we focus on an undervalued property of this random walk in the data mining point of view: the stationary probability distribution π_i^* of this random walk is proportional to the square of a node's eigenvector centrality score. That is,

$$\pi_i^* = \psi_i^2 \tag{2}$$

where ψ_i is the i-th element of the principal eigenvector of the adjacency matrix.

This property indicates that if we sample graphs using the maximal entropy random walk, the sampled nodes will be heavily biased towards the nodes with high centrality scores. This in turn indicates that the sampled graph will be likely to preserve the connectivity aspect of the original graph, which makes MERW sampling as a potential "best" algorithm for our goal of effectively estimating the graph's node centrality. Graphed below is the stationary probability distribution for the simple random walk (also called general random walk) and maximal entropy random walk. It is immediately clear that the two random walk methods favor distinctively different parts of the graph.



(a)

(b)

Figure 1: Steady state probability of (a)General Random Walk and (b)Maximal Entropy Random Walk.

3.2 Implementation of MERW Sampling

In the original formulation of the maximal entropy random walk, as given in [4], a random walker requires the adjacency matrix information in order to perform maximal entropy random walk. Such need arises from the fact that the transition probability $p_{i\to j}$ from node i to j is given by

$$p_{i\to j} = \frac{A_{ij}}{\lambda}\frac{\psi_j}{\psi_i} \tag{3}$$

where A, λ, ψ are the adjacency matrix, the principal eigenvalue, and the principal eigenvector respectively.

However, this directly contradicts the setting of the problem this paper addresses, as it is assumed that only local information about the graph is accessible. In this case, an exact version of maximal entropy random walk cannot be performed, but an approximated version is still possible. According to [6], the approximated transition probabilities for the maximal entropy random walk can be given in successive orders.

The 0th order approximation corresponds to the case where all the nodes have the same degree. Then

$$p^{(0)}_{i\to j} = 1/d_i \tag{4}$$

where j is a neighboring node of node i, and d_i is the degree of node i.

The 1st order approximation of the transition probabilities correspond to the assumption of no degree correlation between the nodes, and

$$p^{(1)}_{i\to j} \propto d_j \tag{5}$$

The 2nd order approximation corresponds to the case where degree correlations between a node up to its first neighbors is assumed to exist. In this case, the approximate transition probability is given by,

$$p^{(2)}_{i\to j} \propto d_j^{1-\nu} \tag{6}$$

where ν is the characteristic exponent of the nearest neighbor degree correlation, given by the relation

$$d_i \propto d_{nn,i}^{-\nu} \tag{7}$$

with d_i denoting the degree of node i, and $d_{nn,i}$ denoting the average degree of the nearest neighbors of i. If $\nu > 0$, the network is a disassociative network, and associative vice versa.

In our implementation of maximal entropy random walk(MERW) sampling algorithm, we use Equation (6) for the transition probability of the random walker. Here, it must be noted that ν is a global property of the graph, and hence is not a given input to the algorithm. Thus an implementation of MERW sampling must also address the problem of locally estimating ν.

We solve this problem by noting that a random walk sampling algorithm requires a relaxation stage at the beginning of the algorithm, where the random walker transitions around the graph but no samples are taken in the process. Relaxation stage is necessary for random walk algorithms because in order to sample according to the steady state probability distribution of the walk (which is often the distribution of interest), one must first allow the underlying Markov process to reach equilibrium. In the experiments, the relaxation time was set to 1000 steps.

To exploit this relaxation stage for the estimation of the degree correlation exponent ν, we first initialize the algorithm with $\nu = 0$ - which corresponds to the 1st order approximation of MERW - and allow the random walker to equilibrate for a given amount of steps. During this process, for every node i the random walker visits, $\log d_i$ and $\log d_{nn,i}$ are calculated and stored in a list. At the end of the relaxation stage, linear regression is performed on $\log d_i$ and $\log d_{nn,i}$ to calculate the initial value of ν to be used in the sampling.

Afterwards, the algorithms moves to the sampling stage, where node and edge samples are taken from the trajectory of the random walker. However, as the random walker successively visits new nodes, the values of $\log d_i$ and $\log d_{nn,i}$ are computed for those nodes as well, and linear regression model determining ν is updated by the recursive linear regression algorithm[12]. Recursive linear regression was preferred over recursive least squares, as it is much simpler and faster, not even requiring matrix inversion. The overall algorithm is presented in the box below.

The implementation was done with Python 3.7, with custom class implemented for the representation of the graph data type and fast access of node connection information. To calculate properties of the sampled graph, such as the clustering coefficient, SNAP library[13] was used, in conjunction with various other libraries such as NetworkX for visualization.

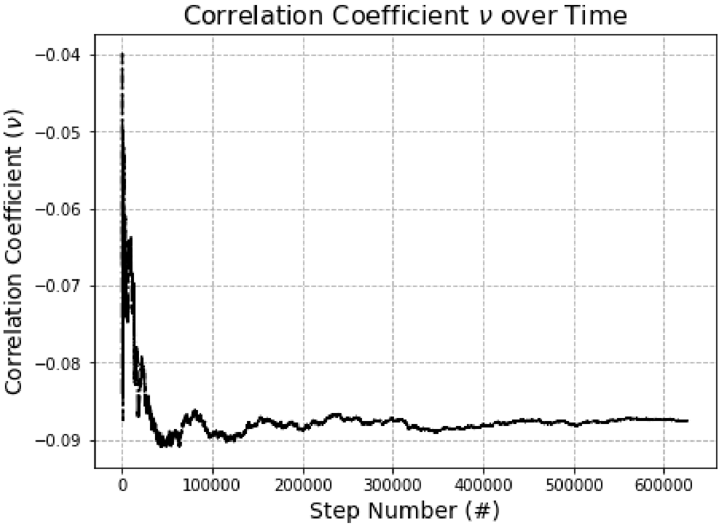


Figure 2: Graph of the adaptively updated ν value during our implementation of maximal entropy random walk sampling. Note that ν settles to an equilibrium value after repeated iterations.

Algorithm 1 Maximal Entropy Random Walk Sampling

```
Input: $G(V, E), V_0 \subset V$, constants $r, t, t_0$
Output: $G_s(V_s, E_s)$
  function MERW($w_i, \nu$)
      $w_f \leftarrow v$ with probability $\propto d_v^{1-\nu}$ for all $v \in N(w_i)$
       return $w_f$
  end function
  function CalcNu($R$)
      for all $[x, y]$ in $R$, perform recursive linear regression i.e. $\log y = -\nu \log x + C$
       return $\nu$
  end function
  while $|G_s| < |G| * r$ do
      $w \leftarrow w \in V_0$
      $\nu \leftarrow 0$
      for $i = 1$ to $t_0$ do
          $w \leftarrow$ MERW($w, \nu$)
          $R \leftarrow R \cup [d_w, (\sum_{v \in N(w)} d_v)/d_w]$
          $\nu \leftarrow$ CalcNu($R$)
      end for
      $w_i \leftarrow w$
      for $i = 1$ to $t$ do
          $w \leftarrow$ MERW($w_i, \nu$)
          $V_s \leftarrow V_s \cup \{w_i\}$
          $E_s \leftarrow E_s \cup \{(w_i, w)\}$
          $R \leftarrow R \cup [d_w, (\sum_{v \in N(w)} d_v)/d_w]$
          $\nu \leftarrow$ CalcNu($R$)
          $w_i \leftarrow w$
      end for
  end while
```

4 Experiments

The two questions we would like to answer from the experiments are as follows:

- *How can we estimate top k eigenvector centrality scores with limited information?*

- *How can we make a 'good' sampling for estimation of global properties of a given graph?*

In this section, we demonstrate that Maximal Entropy Random Walk(MERW) sampling holds the answer to those questions.

4.1 Algorithms and Datasets for Benchmark Comparison

To assess the performance of our algorithm, three commonly used graph sampling algorithms (all in accordance with our problem setting of the unavailability of the adjacency matrix) in literature[2, 9] were chosen to serve as benchmark comparisons. The descriptions of the chosen algorithms are listed below.

- **General Random Walk** Simplest form of random walk sampling. The transition probability of the random walker moving from node i to j is given by $p_{i\to j} = 1/d_i$. Thus, the transition probability is only depends on the degree of the starting node and is independent of the properties of the final node. The random walk also has a restart probability p - that is, before transition, a biased coin is tossed, the outcome of which determines whether the random walker completes the transition or flies back to its initial starting point. This is repeated until the desired number of nodes are sampled. For the rare case of the random walker being stuck in a dead-end, it is "rescued" by allowing it to jump to a position in the graph. The restart probability was set to p = 0.15, in accordance with [2].

- **Metropolis-Hastings Random Walk** Random walk version of the uniform node sampling algorithm[14]. The equilibrium distribution of the random walk is set to be a uniform distribution over the nodes via the Metropolis-Hastings algorithm. The transition probability from node i to j is given as $\pi_{ij} = min\{1, d_i/d_j\}$. Probability for not transitioning also exists, with value of $p_{i\to i} = 1 - \sum_j p_{i\to j}$. The algorithm is also equipped with a rescue mechanism for the random walker meeting a dead end. Same was the MERW algorithm, a relaxation stage is required prior to sampling for the underlying Markov process to equilibrate. The length of this relaxation stage was set to be 1000 steps, same as the value used in the MERW sampling.

- **Forest Fire** Inspired by temporal graph evolution model, Forest Fire sampling algorithm picks a random initial node and burns its edges with probability p independently. When an edge is burned, the fire is spread to the node on the other side, and edge burning continues in the same manner recursively. The burning probability was set to be 0.7, following the specifications of [2].

The dataset for the experiments involved both synthetic data and real world graph datasets. The real world graphs were sourced form the SNAP dataset[15] while the synthetic graphs were generated using random graph models such as Barabasi-Alberts or Erdos-Renyi models. The exact specifications for the datasets used are shown in Table 1.

Dataset Name	**Nodes**	**Edges**	**Specification**
Barabási–Albert	10000	99945	Generated graph using Barabási–Albert model
ca-CondMat	21363	91342	GCC of arXiv Condensed Matter collaboration network
AS	6474	13895	Network of Internet routers(Autonomous Systems)
Erdős–Rényi	10000	99952	Generated graph using Erdős–Rényi model
RoadNet	20010	27038	Road network of California

Table 1: Specifications of Dataset

4.2 Results

The section is composed of two parts: 1) Estimation of top Eigenvector Centrality scores, and 2) Estimation of global properties using Maximal Entropy Random Walk sampling. In the first part, we will show that while other algorithms fail to retrieve the eigenvector centrality information for a given graph, MERW sampling allows for an almost exact estimation of the top eigenvector centrality scores. In the second part, the capability of the MERW sampling in estimating graph global properties will be compared to other algorithms. MERW sampling will be tested comprehensively using different global properties with discussions about the behaviors of the sampling algorithms.

4.2.1 Estimation of top Eigenvector Centrality scores

To estimate the top eigenvector centrality scores, each of the graphs in the aforementioned datasets was sampled with 4 different algorithms - general random walk(GRW), Metropolis-Hastings random walk(MHRW), maximal entropy random walk(MERW), and forest fire(FF). The size of the sampled graphs was chosen to be 20% of the original graph. Then the top 10 eigenvector centrality scores with the corresponding nodes of the original and sampled graphs are calculated. We present the result for the arXiv Cond. Mat. GCC data below. The results for the rest of the datasets can be found in the appendix.

Rank	Node ID	Scores
1	2092	0.2254
2	3880	0.2174
3	2116	0.1672
4	949	0.1634
5	1637	0.1391
6	4068	0.1384
7	2093	0.1159
8	2119	0.1140
9	2114	0.1134
10	2091	0.1064

Table 2: Centrality scores of ca-CondMat GCC dataset

Rank	Node ID	Scores
1	4574	0.4011
2	6810	0.3714
3	4575	0.3669
4	4571	0.3580
5	6807	0.3029
6	6808	0.2702
7	6811	0.2669
8	6806	0.2381
9	4572	0.2202
10	6809	0.1808

Table 3: Centrality scores of MHRW sampled graph

Rank	Node ID	Scores
1	17877	0.6024
2	17875	0.4464
3	2883	0.3919
4	4106	0.3486
5	5207	0.2251
6	16305	0.2002
7	13164	0.2002
8	17870	0.1303
9	17869	0.0748
10	260	0.0335

Table 4: Centrality scores of FF sampled graph

Rank	Node ID	Scores
1	1275	0.3440
2	760	0.3253
3	750	0.2697
4	778	0.2389
5	788	0.2226
6	7801	0.2197
7	512	0.1871
8	769	0.1768
9	12191	0.1644
10	12190	0.1644

Table 5: Centrality scores of GRW sampled graph

Rank	Node ID	Scores
1	2092	0.2526
2	3880	0.2486
3	949	0.2200
4	2116	0.1834
5	1637	0.1692
6	4068	0.1570
7	2119	0.1303
8	2091	0.1270
9	585	0.1254
10	2093	0.1236

Table 6: Centrality scores of MERW sampled graph

Table 2 shows the top 10 eigenvector centrality nodes with their centrality scores. From the table, it can be seen that the most central node, 2092, has the score of 0.2254, while the 10th node, 2091, has 0.1064, which is less than half of the former.

For the MHRW and FF sampling(Table 3 & 4), the scores and high-ranked the node ID is quite different from the original graph. This is unsurprising as MHRW tends to visit all nodes with equal probability, and FF samples radially from a randomly chosen initial node, and thus has no built-in mechanism to navigate to the high centrality nodes, especially if they are far from the starting point. Centrality scores for the GRW(Table 5) are also not promising as well. However, there were some cases in which the GRW did perform - the most notable is the result for the AS dataset, where GRW was able to find many of the top ranking nodes, and approximate the score distribution(Table 10).

Inspecting the results for the maximal entropy random walk sampling, it can be seen that the algorithm captures the top scoring nodes with much higher accuracy. Many of the nodes in MERW sampled results(Table 6) can be seen in the original graph, with slight shuffling in the rankings. Such ability of the MERW sampling to accurately capture the original eigencentrality distribution holds for other datasets as well, except for the RoadNet dataset where MERW failed to correctly find the top ranking nodes.

To better quantify the quality of the estimated centrality scores, the root-mean-square-error(RMSE) between the original scores and the estimated scores was computed for each of the datasets.

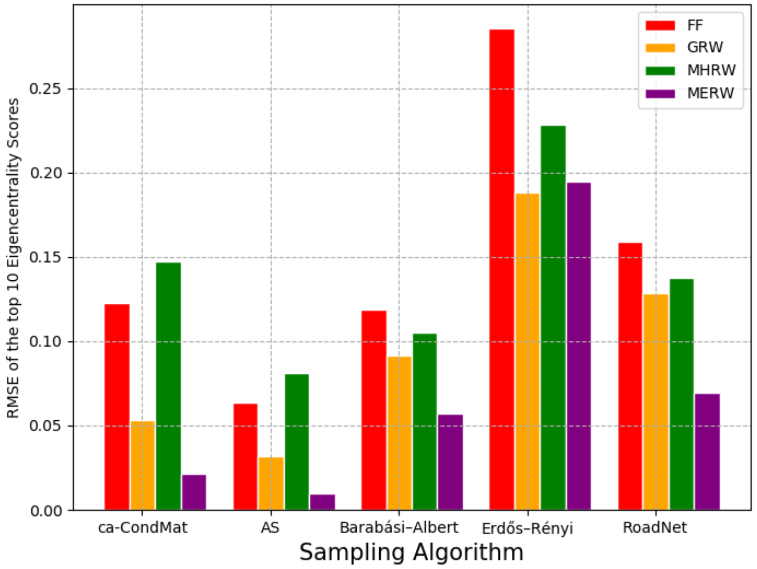


Figure 3: RMSE of the estimation of the top 10 eigenvector centrality scores

From the graph, is is immediately clear that overall, MERW is the best performing algorithm for estimating the top 10 eigencentrality scores, with an less than 5% error for real world datasets. An intriguing point is that MERW managed to achieve low error for the RoadNet dataset as well, even though it failed to find the correct high centrality nodes (as can be seen from the discrepancies between the Node IDs in Table 12 and Table 26). Whether this uncanny estimation of the centrality score is coincidental or not needs more in-depth investigation, but the phenomenon itself is quite interesting nonetheless.

For the case of the Erdős–Rényi dataset, it can be seen that the GRW sampling outperforms all other algorithms. This intuitively makes sense, as an Erdős–Rényi random graph is almost homogeneous and GRW sampling with restart samples the neighborhood of its starting point. Due to the homogeneity, the graph statistics of the sampled neighborhood will closely resemble that of the original graph, resulting in low error in the estimated centrality scores.

Summing up, it is clear that MERW does perform an eigencentrality-based sampling as initially desired, resulting in samples that much better resemble the original centrality of the graph. In the following section, we will investigate whether MERW also performs well in estimating other types of graph properties.

4.2.2 Estimation of global properties using Maximal Entropy Random Walk sampling

In this section the performance of the MERW sampling was benchmarked against other algorithms with respect to the following criteria:

- **Rank - Eigenvector centrality distribution** Distribution of component of the first eigenvector of the adjacency matrix, with respect to ranking of that component
- **Degree distribution** The degree-count distribution of the graph.
- **Degree - Clustering coefficient distribution** The distribution of a node's degree its average clustering coefficient
- **Hop plot** Define maximum hop of a node as the minimum number of steps required to reach all other reachable nodes from the starting node. Hop plot is the maximum hop - count distribution.
- **Eigenvalue distribution** The eigenvalue-rank distribution. To properly define the D-statistics on this distribution, the absolute values of the eigenvalues were used as the y-axis, while the rank refers to the ranking of the original eigenvalues. As the full computation of all the eigenvalues of the graph takes too much time, only the top 50 eigenvalues were considered.

The distribution plots for each of the distributions for the arXiv Cond. Mat. GCC data] is as the following. The sample size was taken to be the same as that of the previous section (20% of the original graph size). The distribution plots for the other datasets are listed in the appendix.

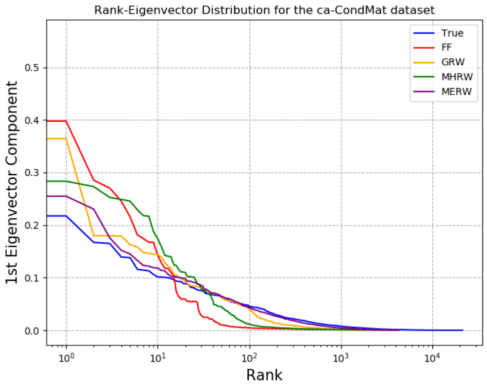


Figure 4: Eigenvector distribution

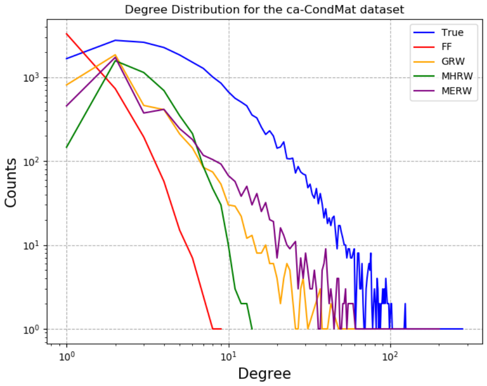


Figure 5: Degree distribution

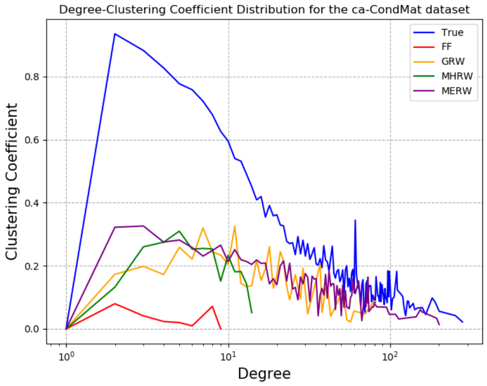


Figure 6: Clustering coefficient distribution

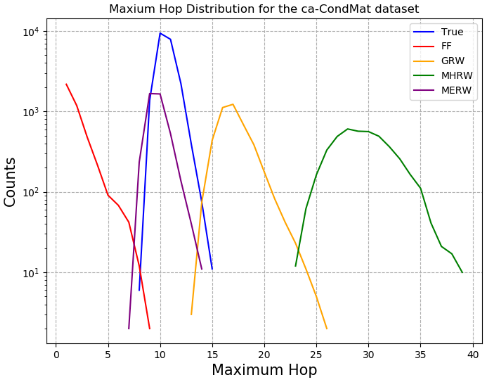


Figure 7: Hop plot

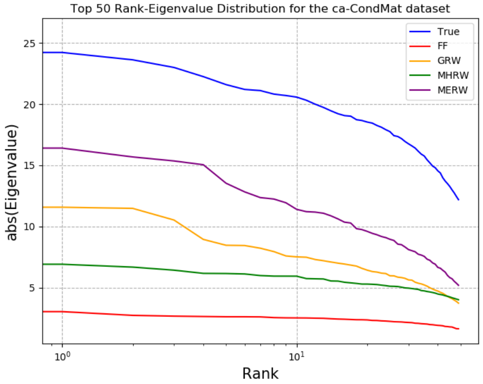


Figure 8: Top 50 eigenvalue distribution

In order to quantitatively describe how closely the sampled distributions match the original distribution, the Kolomogorov-Smirnov D-Statistic was chosen as the metric. Given two probability distributions of x, $P(x)$ and $P'(x)$ and their corresponding cumulative distribution functions $F(x)$ and $F'(x)$, the D-statistic is defined as

$$D = max_x\{|F'(x) - F(x)|\} \tag{8}$$

The D-statistic is always in the range [0, 1] and lower D value indicated better agreement between the two distributions. Listed below are the histograms of the D-statistics of sampled distributions for all the datasets.

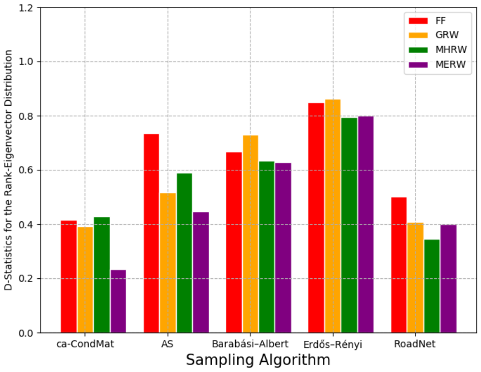
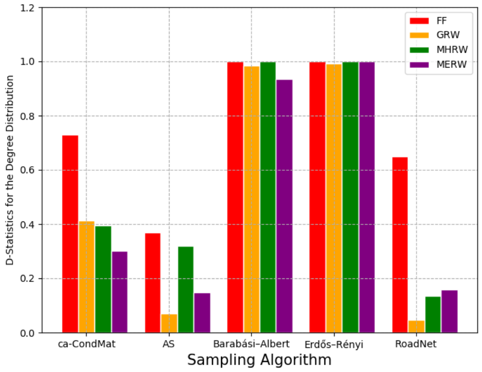


Figure 9: D-statistics for the rank-eigenvector distribution



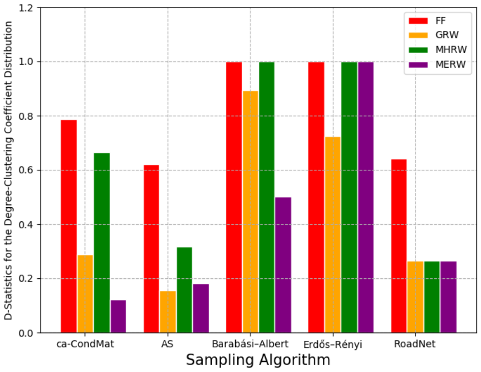
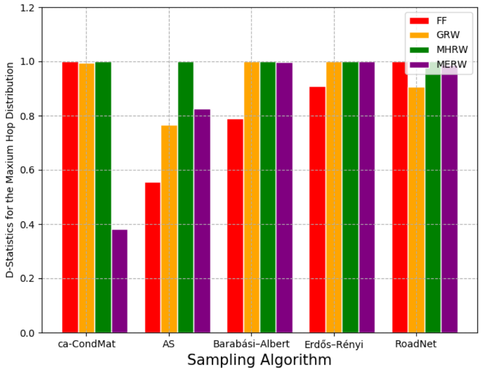


Figure 10: D-statistics for the degree distribution

Figure 11: D-statistics for the degree - clustering coefficient distribution



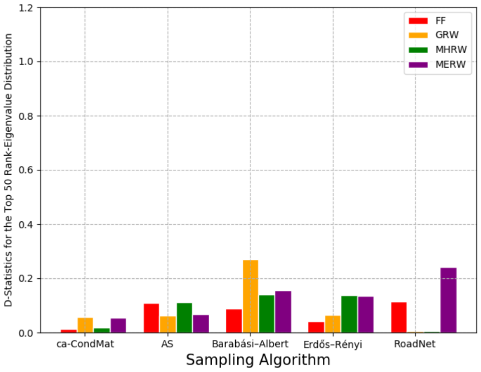


Figure 12: D-statistics for the hop plot

Figure 13: D-statistics for the top 50 eigenvalue distribution

The distribution plots as well as the D-statistic histograms display a number of interesting results worth analyzing. First, Figure 4 is the distribution of the first eigenvector of the sampled graphs, corresponding to the full version of the tables presented in the above section. From the distribution plot, it can be seen that out of the four sampling algorithms used, MERW sampling provided the best estimation of the true distribution.

Looking at Figure 9, it is clear that MERW is usually the best choice for the eigenvector estimation, except for the RoadNet, which was commented above. It needs to be reminded that even though for BA and ER graphs MHRW sampling also provides a similar level of accuracy, MHRW sampling fails to properly retrieve the correct IDs that correspond to the eigencentrality scores. This means that MHRW samples an induced subgraph that "looks like" the original graph centrality-wise, but is not sampling the part of the original graph that was actually high in the eigencentrality scores.

From Figure 10, MERW seems to perform nicely compared to its competitors. It does struggle for RoadNet, but RoadNet seems to the edge case for maximal entropy random walk, as all the MERW sampled graph characteristics for RoadNet is unsatisfactory. Another trend worth commenting is that none of the algorithms managed to properly capture the degree distribution of teh ER graph, as all the D-values are very close to 1. The answer lies in Figure 25: while the original ER graph consisted of only very high degree nodes, all the sampled graphs - probably due to their induced nature - had relatively small degree nodes, resulting in extreme discrepancy between the original and the sampled degree distributions.

Again for the degree-clustering coefficient distribution, MERW triumphs over its competitors (Figure 11). This is not unexpected because MERW is biased towards nodes with high centrality, which increases the chance of the more salient part of the graph, connectivity-wise, to be sampled, resulting in a more accurate depiction of the cluster coefficient distribution. Again all algorithms fail to properly sample the ER graph, due to the same reason as that of the degree distribution.

For the hop plot and the eigenvalue distribution, there is no clear trend of MERW dominating over other algorithms. Out of the different datasets, MERW performs well for Cond. Mat. and AS datasets, so there maybe a possibility of MERW interacting well with certain types of real world networks, but this need to be further investigated with more data.

Still, while MERW does not excel, none of the other algorithms do either. Especially, the D values for the hop plot is quite high for almost all the cases, meaning none of the sampling algorithms could sample anything resembling the original distribution. Looking at the actual distribution plots(Figures 7, 17, 22, 27 and 32), we can see that this is a combination of two effects. First, forest fire underestimates the maxmum hop distribution, due to its radially spreading nature which produces a much tighter-knit graph than the actual graph. On the other hand, GRW, MERW, and MHRW samplers wander around the graph, resulting in a chain like structure that is prone to having a larger radius than the original graph.

In short, it can be concluded that even with eigencentrality estimation excluded from the evaluation criteria- since MERW should do this well due to its steady state distribution - MERW sampling is still a competitive algorithm, especially for connectivity-related graph properties such as the degree distribution or the clustering coefficient distribution. As for the hop plot and the eigenvalue distribution, a better sampling algorithm is in need, as none of the algorithms tested in this study was able to show a consistently good result across all the datasets.

4.3 Additional Discussion

4.3.1 Why did Forest Fire perform badly?

Overall, Forest Fire algorithm did not perform well in our experiments regardless of the different dataset and different evaluation criteria. It could be that the sampling ratio used for the study was not optimal for the algorithms, but even with this taken into account, the poor performance seems to contradict the result of former studies, such as Leskovec, et al[2]. To resolve the apparent mismatch, a deeper inquiry was made.

Looking at the sampled degree distribution plots for the different datasets(Figures 5, 15, 20, 25 and 30), it is quite clear that regardless of what the original degree distribution looks like, forest fire sampling produces a power law sample. This is unsurprising, because the forest fire algorithm itself is motivated from the preferential attachment model, which produces the graph. In particular, Figure 25 and Figure 30 makes the most striking contrast: while the original degree distributions of the two datasets are completely different, the former being ∩-shaped and the latter displaying power law distribution,the forest fire sampled result shows the same power law characteristic.

From this, it is clear that forest fire is not a good choice for graphs that deviate from the power law, as the sample will not properly reflect the original graph characteristics. However, in our study, forest fire did not perform well even for the power law graph of the BA model. From Figure 30, it is seen that the number of degrees was vastly underestimated, which is the result of a suboptimal value of the burn probability. If this parameter is adjusted, FF sampling would return satisfactory samples for the power law graphs.

From the above observations, it can concluded that since Forest Fire is inclined to producing scale-free networks, it should be applied to power law graphs for best performance. Also the burn probability must be adjusted to a near optimal value, either by parameter tweaking or through heuristics. However, for the case of sampling arbitrary networks with unknown degree distribution one must be wary of its power law producing side effects.

5 Conclusions

Sampling properties from large graphs is an efficient and useful technique for analyzing large scale graphs. In this research, the main focus was to effectively estimate the node centrality information under the restriction that the entire graph cannot be accessed. To solve the problem, a variety of random walk called maximal entropy random walk was newly interpreted from a data mining perspective, and formulated in to a local sampling algorithm.

By assessing the performance of our maximal entropy random walk sampling against other existing algorithms over multiple datasets showed that MERW sampling did far outperform all other algorithms in terms of extracting the eigenvector centrality information from the original graph. Additional benchmarks showed that MERW is also well-capable of estimating both the degree distribution and degree-clustering coefficient distribution, as they are both connectivity related graph properties and correlated with nodes with high centrality. We expect this new algorithm t become a powerful addition in web page analysis, as it could be used to find pages with high importance without having to deal with the entire adjacency matrix.

References

[1] U. Kang, C. E. Tsourakakis, and C. Faloutsos, "Pegasus: A peta-scale graph mining system implementation and observations," in *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, (Washington, DC, USA), pp. 229–238, IEEE Computer Society, 2009.

[2] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, (New York, NY, USA), pp. 631–636, ACM, 2006.

[3] G. Lohmann, D. S. Margulies, A. Horstmann, B. Pleger, J. Lepsien, D. Goldhahn, H. Schloegl, M. Stumvoll, A. Villringer, and R. Turner, "Eigenvector centrality mapping for analyzing connectivity patterns in fmri data of the human brain," *PLOS ONE*, vol. 5, pp. 1–8, 04 2010.

[4] Z. Burda, J. Duda, J. M. Luck, and B. Waclaw, "Localization of the maximal entropy random walk," *Phys. Rev. Lett.*, vol. 102, p. 160602, Apr 2009.

[5] Z. Burda, J. Duda, J. M. Luck, and B. Waclaw, "The various facets of random walk entropy," 2010.

[6] R. Sinatra, J. Gómez-Gardeñes, R. Lambiotte, V. Nicosia, and V. Latora, "Maximal-entropy random walks in complex networks with limited information," *Phys. Rev. E*, vol. 83, p. 030103, Mar 2011.

[7] V. Krishnamurthy, M. Faloutsos, M. Chrobak, L. Lao, J. H. Cui, and A. G. Percus, "Reducing large internet topologies for faster simulations," in *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems* (R. Boutaba, K. Almeroth, R. Puigjaner, S. Shen, and J. P. Black, eds.), (Berlin, Heidelberg), pp. 328–341, Springer Berlin Heidelberg, 2005.

[8] C. Hübler, H. Kriegel, K. Borgwardt, and Z. Ghahramani, "Metropolis algorithms for representative subgraph sampling," in *2008 Eighth IEEE International Conference on Data Mining*, pp. 283–292, Dec 2008.

[9] P. Hu and W. C. Lau, "A survey and taxonomy of graph sampling," 2013.

[10] N. K. Ahmed, J. Neville, and R. Kompella, "Network sampling: From static to streaming graphs," *ACM Trans. Knowl. Discov. Data*, vol. 8, no. 2, pp. 1–56, 2013.

[11] A. Lakhina, J. W. Byers, M. Crovella, and P. Xie, "Sampling biases in ip topology measurements," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, vol. 1, pp. 332–341 vol.1, March 2003.

[12] J. H. Klotz, "Updating simple linear regression," *Statistica Sinica*, vol. 5, no. 1, pp. 399–403, 1995.

[13] J. Leskovec and R. Sosič, "Snap: A general-purpose network analysis and graph-mining library," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 1, p. 1, 2016.

[14] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 04 1970.

[15] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection." http://snap.stanford.edu/data, June 2014.

A Appendix

A.1 Additional Data

A.1.1 Eigencentrality scores for other datasets

AS Dataset

Rank	Node ID	Scores
1	701	0.5275
2	1239	0.2809
3	3561	0.2418
4	1	0.1548
5	7018	0.1459
6	2914	0.1242
7	2828	0.0984
8	2548	0.0967
9	293	0.0910
10	209	0.0823

Table 7: Centrality scores of the AS dataset

Rank	Node ID	Scores
1	3786	0.6044
2	6461	0.2663
3	4766	0.2414
4	5051	0.1447
5	9529	0.1407
6	7564	0.1407
7	9532	0.1407
8	7620	0.1407
9	9526	0.1407
10	4670	0.1394

Table 8: Centrality scores of MHRW sampled graph

Rank	Node ID	Scores
1	701	0.3863
2	7963	0.1157
3	10826	0.1120
4	3674	0.1088
5	1239	0.1080
6	6469	0.1028
7	13516	0.1026
8	14134	0.1026
9	6456	0.1026
10	13889	0.1026

Table 9: Centrality scores of FF sampled graph

Rank	Node ID	Scores
1	701	0.5674
2	1239	0.2301
3	3561	0.1972
4	1	0.1731
5	2914	0.1507
6	7018	0.1308
7	286	0.1112
8	293	0.1021
9	3356	0.0862
10	1800	0.0788

Table 10: Centrality scores of GRW sampled graph

Rank	Node ID	Scores
1	701	0.5303
2	1239	0.3082
3	3561	0.2192
4	1	0.1801
5	2914	0.1536
6	7018	0.1517
7	2828	0.1228
8	293	0.1152
9	2548	0.1043
10	209	0.0993

Table 11: Centrality scores of MERW sampled graph

RoadNet Dataset

Rank	Node ID	Scores
1	1607873	0.1804
2	1499044	0.1423
3	1499041	0.1415
4	1607872	0.1334
5	1607753	0.1310
6	25154	0.1308
7	1499043	0.1272
8	1607698	0.1244
9	1607866	0.1240
10	25585	0.1219

Table 12: Centrality scores of the RoadNet dataset

Rank	Node ID	Scores
1	2479	0.3957
2	2473	0.3501
3	2478	0.3476
4	2529	0.2935
5	2530	0.2912
6	2522	0.2546
7	2469	0.1987
8	2472	0.1928
9	2718	0.1914
10	2518	0.1885

Table 13: Centrality scores of MHRW sampled graph

Rank	Node ID	Scores
1	6524	0.5572
2	8731	0.5572
3	8733	0.4350
4	6519	0.2175
5	39222	0.2175
6	8734	0.2175
7	8724	0.2175
8	2250	0.0137
9	4096	0.0122
10	5486	0.0105

Table 14: Centrality scores of FF sampled graph

Rank	Node ID	Scores
1	25154	0.2789
2	25585	0.2599
3	25268	0.2379
4	25272	0.2273
5	25273	0.2116
6	25580	0.2011
7	25079	0.1877
8	25581	0.1668
9	25274	0.1638
10	25266	0.1633

Table 15: Centrality scores of GRW sampled graph

Rank	Node ID	Scores
1	1492596	0.2340
2	1492600	0.1874
3	1481369	0.1835
4	1481373	0.1810
5	1492601	0.1782
6	1481436	0.1719
7	1481435	0.1633
8	1481367	0.1608
9	1481365	0.1604
10	1481370	0.1603

Table 16: Centrality scores of MERW sampled graph

Erdős–Rényi Model

Rank	**Node ID**	**Scores**
1	8759	0.0200
2	5139	0.0194
3	3007	0.0192
4	8808	0.0191
5	9605	0.0189
6	5950	0.0185
7	751	0.0185
8	4861	0.0183
9	7642	0.0182
10	2963	0.0181

Table 17: Centrality scores of the Erdős–Rényi model

Rank	**Node ID**	**Scores**
1	7720	0.5692
2	3379	0.4349
3	9574	0.3818
4	7669	0.2137
5	7547	0.2072
6	4923	0.1877
7	9742	0.1812
8	2900	0.1536
9	4554	0.1392
10	7007	0.1252

Table 18: Centrality scores of MHRW sampled graph

Rank	**Node ID**	**Scores**
1	6478	0.3953
2	1831	0.3602
3	8642	0.3535
4	3501	0.3108
5	1167	0.3108
6	4662	0.3108
7	6648	0.3108
8	9247	0.3108
9	1798	0.3108
10	5596	0.0628

Table 19: Centrality scores of FF sampled graph

Rank	Node ID	Scores
1	6970	0.5800
2	349	0.1780
3	7655	0.1682
4	8058	0.1588
5	3839	0.1586
6	9108	0.1469
7	9797	0.1454
8	1944	0.1432
9	6577	0.1393
10	8630	0.1363

Table 20: Centrality scores of GRW sampled graph

Rank	Node ID	Scores
1	804	0.5511
2	2797	0.3391
3	962	0.2870
4	9805	0.2624
5	4249	0.2363
6	4241	0.2206
7	7791	0.2125
8	6253	0.2107
9	5133	0.2106
10	8041	0.1355

Table 21: Centrality scores of MERW sampled graph

Barabási–Albert Model

Rank	Node ID	Scores
1	4	0.1652
2	1	0.1591
3	13	0.1584
4	10	0.1579
5	17	0.1559
6	9	0.1554
7	8	0.1475
8	2	0.1456
9	5	0.1361
10	3	0.1334

Table 22: Centrality scores of the Barabási–Albert model

Rank	Node ID	Scores
1	5115	0.5748
2	8341	0.3155
3	3833	0.3137
4	878	0.3097
5	7353	0.2583
6	4664	0.2544
7	1095	0.1630
8	6429	0.1508
9	6017	0.1399
10	5953	0.1254

Table 23: Centrality scores of MHRW sampled graph

Rank	Node ID	Scores
1	5415	0.5312
2	984	0.5211
3	6731	0.2349
4	656	0.2310
5	171	0.2050
6	324	0.2050
7	1672	0.2050
8	1443	0.2050
9	10	0.2004
10	193	0.1717

Table 24: Centrality scores of FF sampled graph

Rank	Node ID	Scores
1	6390	0.4508
2	76	0.2802
3	832	0.2432
4	210	0.2286
5	1246	0.1788
6	2424	0.1738
7	2719	0.1714
8	2778	0.1436
9	2839	0.1398
10	5227	0.1355

Table 25: Centrality scores of GRW sampled graph

Rank	Node ID	Scores
1	4	0.2585
2	9	0.2278
3	13	0.2259
4	1	0.2252
5	10	0.2152
6	8	0.1970
7	5	0.1929
8	17	0.1920
9	3	0.1919
10	2	0.1778

Table 26: Centrality scores of MERW sampled graph

A.1.2 Sampled distributions for other datasets

AS Dataset

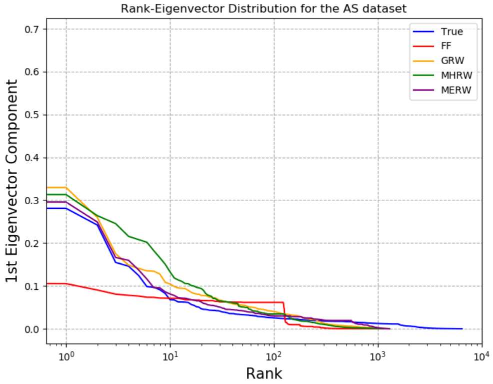


Figure 14: Eigenvector distribution

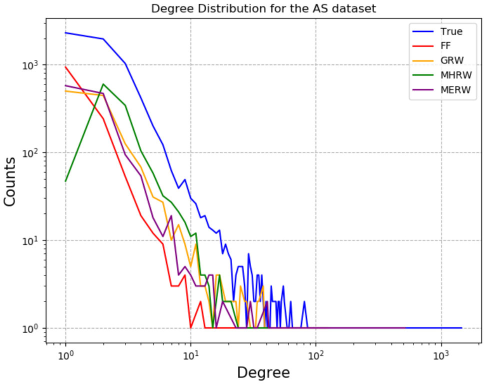


Figure 15: Degree distribution

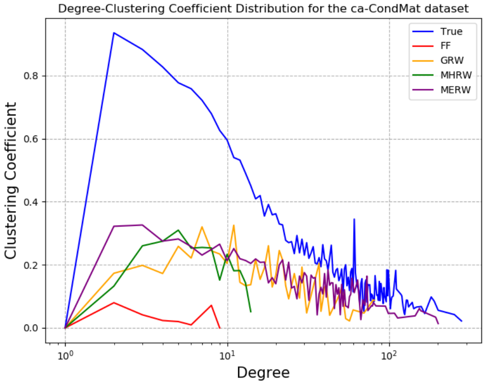


Figure 16: Clustering coefficient distribution

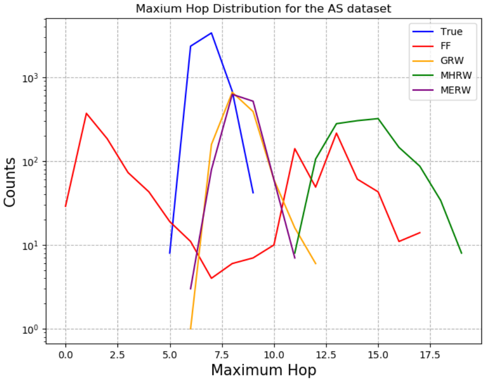


Figure 17: Hop plot

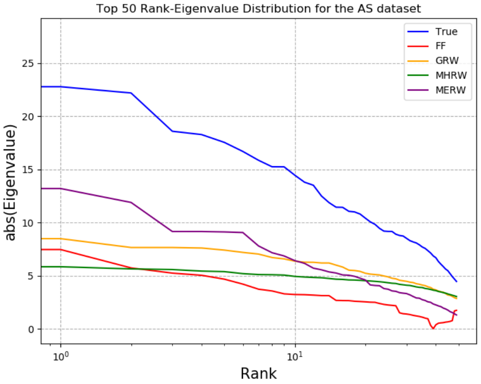


Figure 18: Top 50 eigenvalue distribution

RoadNet Dataset

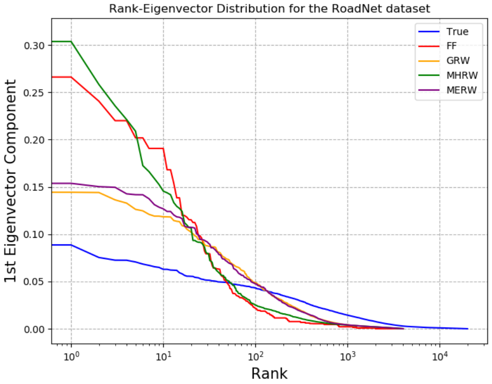
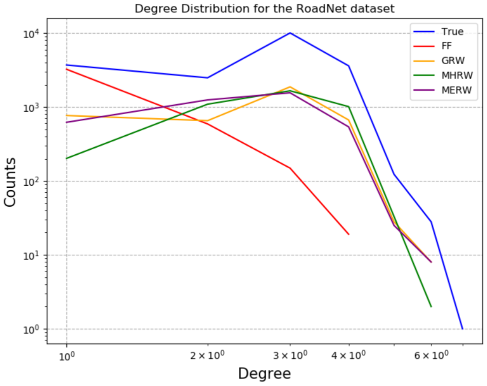


Figure 19: Eigenvector distribution



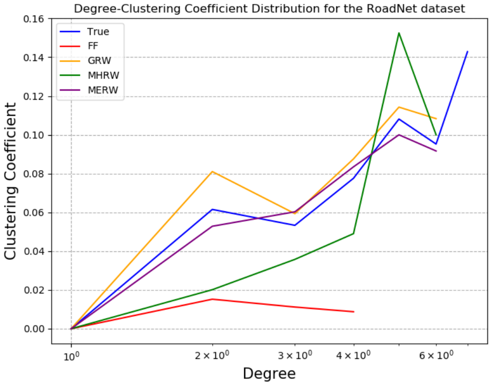
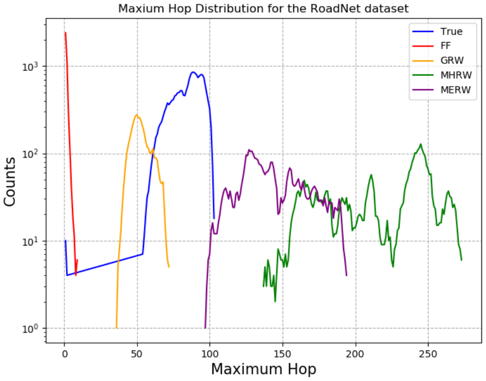


Figure 20: Degree distribution

Figure 21: Clustering coefficient distribution



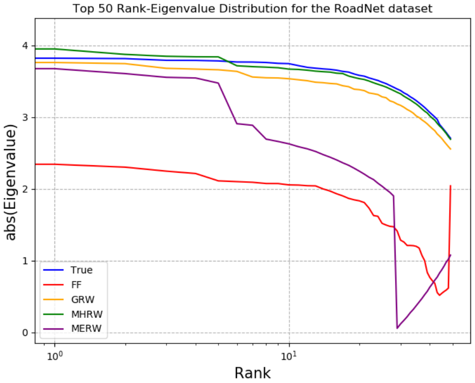


Figure 22: Hop plot

Figure 23: Top 50 eigenvalue distribution

Erdős–Rényi Model

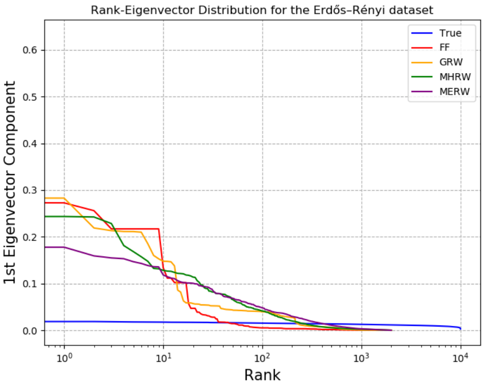


Figure 24: Eigenvector distribution

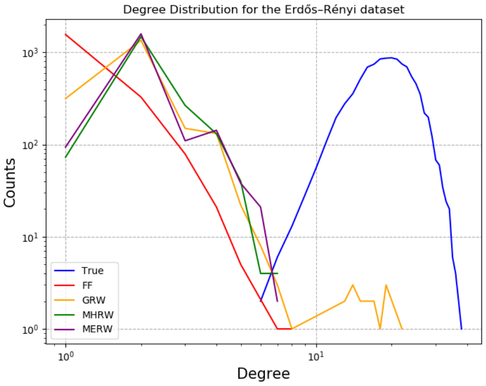


Figure 25: Degree distribution

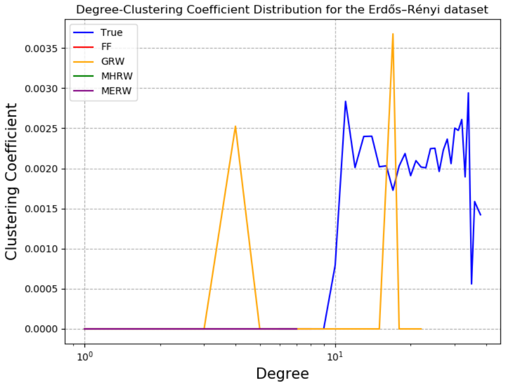


Figure 26: Clustering coefficient distribution

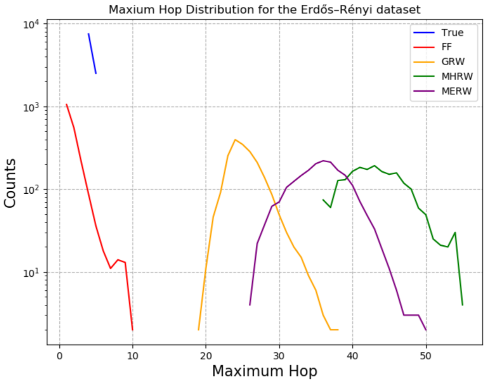


Figure 27: Hop plot

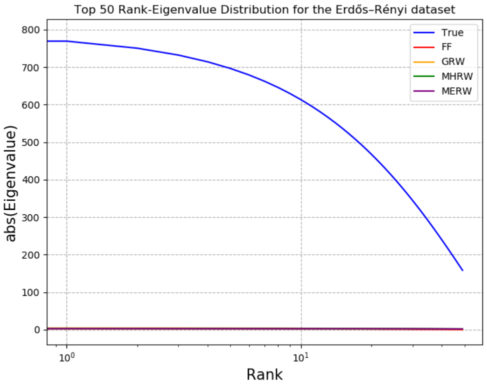


Figure 28: Top 50 eigenvalue distribution

Barabási–Albert Model

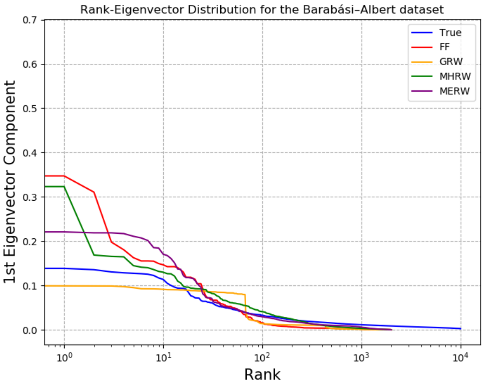


Figure 29: Eigenvector distribution

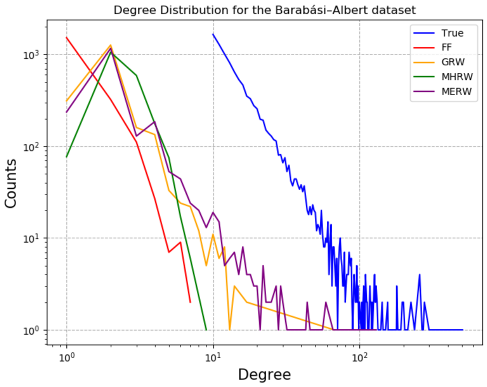


Figure 30: Degree distribution

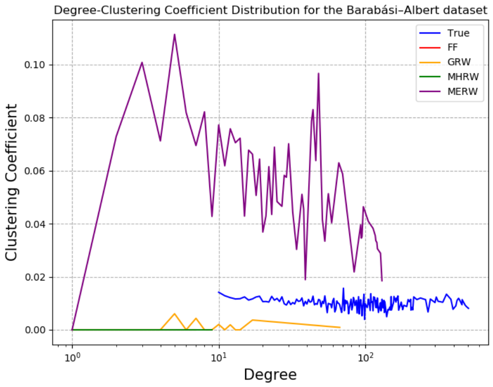


Figure 31: Clustering coefficient distribution

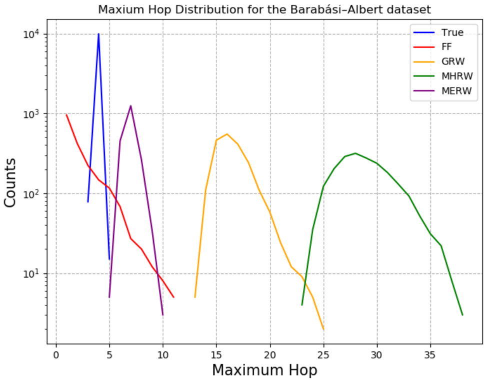


Figure 32: Hop plot

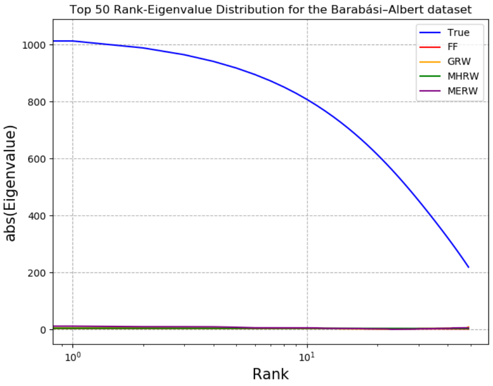


Figure 33: Top 50 eigenvalue distribution

A.2 Labor Division

The team performed the following tasks
Proposal

- Brainstorm research topic, literature research, proposal write-up [all]

Progress report

- Devise Maximal Entropy Random Walk sampling algorithm[all]
- Validate the applicability of the algorithm theoretically[Joon-Hyuk]
- Write codes for Maximal Entropy Random Walk sampling and graph generation [Sung Hoon]

- Optimize the algorithm codes and write codes for other benchmark sampling algorithms [Joon-Hyuk]
- Run codes and analyze results [Sung Hoon]
- Final report writeup [Joon-Hyuk]
- Organize the codes into a packaged form ready for distribution [Joon-Hyuk]

A.3 Full disclosure wrt dissertations/projects

Joon-Hyuk Ko: His current topic is on using deep-learning on data from physical systems, and thus has no relations to this project.

Sung Hoon Kim: His current topic is the same as that of Joon-Hyuk Ko, and thus, not related to this project

Contents