Correlated Network Data Publication via Differential Privacy

Priyanshu Behera (21114077 CSE 3rd year)

***Abstract*—The research paper focuses on network data publication via differential privacy mechanisms to prevent node re-identification or edge disclosure. Furthermore, the paper presents to be a viable expansion in the domain of the correlated data setting and the algorithm is also data dependent and can be tuned according to the dataset. The paper also discusses the metrics to measure the utility of the obtained network structure.**

***Index Terms*— Correlation Parameter, Exponential Mechanism, ε-differential privacy, Centrality, Laplace Noise, Sensitivity, KL-Divergence**

I. Introduction

In today's world, information network is growing at an unprecedented rate. With the emergence of social networking sites and applications much of the information encoded in form of individuals as nodes and connection between them as edges can be unveiled or leveraged for a variety of AI methodologies. And also published data can be used to infer some general but important statistics from network structure.

As a naïvely sanitized graph data can place an individual's privacy on the brink of getting exploited. For example, take the case of a 3-isomorphic graph the connection between two people (can be considered sensitive) is vulnerable to exploitation by an adversary if he is able to find out two disjoint sets of 3 vertices of which these victims are part of. These ideas of assuring privacy revolve around the fact that with a small change in the input dataset there is no effect on the output produced, so a node can participate freely in that network without getting its privacy breached as its presence or absence mean the same to the space outside its network.

Some Preliminaries:

1) **Privacy Budget**: Generally denoted by ε, is the budget which specifically describes how related (or different) can the two datasets can be. Higher the privacy budget, more relatable the perturbed data can be to the original one.

2) **Correlation Parameter:** Measures the extent of correlation within samples of a dataset in a directly proportionate way.

3) **KL-Divergence:** A measure of how similar 2 graphs are in terms of their degree distribution.

4) **ε-Differential Privacy:** A dataset D2 which is an output from differentially private algorithm taking D1 as an input will ensure differential privacy if it follows $\Pr[F[D1] = O] <= e^{\varepsilon} * \Pr[F[D2] = O]$ where "F" is some operation or query performed on the dataset.

5) **Sensitivity:** Sensitivity of a function "F" is defined as change in the output due to the change in the input. Mathematically, $S(F) = \max_{D1,\ D2} |F(D1)-F(D2)|$, where S is the sensitivity and D1&D2 are the datasets.

The *DER (Dense Region Exploration and Reconstruction)* algorithm proposed a stronger mechanism in the realm of privacy protection which can be modified according to the dataset used and considers the correlation within the dataset which is its primary functionality.

In this research paper we will evaluate our algorithms on 3 metrics namely frequency of shortest path length, average relative error for a given query size and KL-Divergence all varied with privacy budget or correlation parameter as the x-axis.

II. Procedure or Methodology

In this section, I will discuss the steps that are followed in implementing the ideas of the DER algorithm.

Objective of the Algorithm Implemented:
Aim: Publish useful network data

Input: A network graph data in form of adjacency matrix.

Output: An adjacency matrix which is distorted and altered according to the privacy budget been allocated.

Some pre-requisites for understanding the algorithm:

1) **Centrality**: Term generally used for adjacency matrices which is a measure of how close the non-zero values are to the diagonal. Or, in technical terms a normalized summation of the Euclidean distance of a non-zero entry from the central cell or mathematically

for a graph G, centrality C(G) is defined as:
C(G) = $\sum_{i,j} \frac{G[i][j]}{|V|-2}$ ***(ceil**(abs(i-|V|/2)) +**ceil**(abs(j-|V|/2)))

2) **Noisy-Count**: Number of 1s in the perturbed matrix.

3) **Score**: Number of cells matching in the original matrix G and obtained perturbed matrix $\tilde{G}$.

4) **Exponential Mechanism**: For a score selection 's' each the favorable event term is scaled by a factor of e^s. Hence more specifically Pr[score = s] = (G[s] *exp(s* ε_A/2)) ÷ Σ_s (G[s] *exp(s* ε_A/2)) where G[s] is the number of times a score s occurs among all permutations and ε_A is the privacy budget.

<u>Theorem</u>: Considering a mechanism which achieves a ε/k differential privacy in a dataset will achieve ε differential privacy with a dataset having a correlation parameter k.

Majorly, the algorithm can be divided into 3 major functions with dedicated privacy budget for each of them. The 3 major functions listed as:

A) **Identify vertex labelling**: Implemented as "*minimize_centrality"* function in the python notebook which aims to assign random labels to the vertices such that the centrality is decreased. This can be broken down into parts as:

a. **Swap pairs generation**: Followed the way described in research paper to generate a swap set χ such that each vertex is involved in only 1 swap such that the *sensitivity* of swapping function decreases.

- **Optimization achieved**: The motivation in this idea is that the sensitivity of a swap decreases, or in other way if a vertex is allowed multiple times to participate in swapping, then we may end up diverting very little from the original labelling and yet spending a large computational power.

b. **Minimize centrality**: Will update the adjacency matrix if the swap leads to a decrease in centrality and if it satisfies the constraint of privacy budget.

- **Further Optimization (*OWN IMPLEMNTATION NOT SPECIFIED IN PAPER)*:** The very fact that a change of labelling can only affect the centrality w.r.t max of 2 rows and columns, so instead of computing the centrality again and again, I simply added the change in centrality due to swapping to the pre-computed value. So reduced time complexity of this step from $O(V^2)$ to O(V).

c. **Privacy Budget Allotted(ε_I) (OWN IDEA OF APPLICATION):** It's reciprocal is utilized as the value by which at max the value of centrality can be decreased. The intuition behind choosing strategy is that lesser a privacy budget is, more the change should occur to the original database.

B) **Explore Dense Region**: *Quadtree Data Structure* is deployed for this purpose. The function "*build_quadtree_opt()"* with necessary parameters and the threshold for declaring a region as leaf is called to construct the quad-tree from a given adjacency matrix.

- **Optimization Achieved**: Made use of "*cnt-summary*" matrix. Kinda prefix sum technique which can be used for calculating noisy count in O (1) as compared to previous $O(V^2)$. Hence decision for the leaf region in the quad-tree can be made efficiently.

- A **DFS (Depth First Search)** run of the quadtree is performed to get all leaf regions which will be later useful for reconstruction of edges in a specific dense region.

- **Privacy Budget allocated (ε_E)**: The privacy budget is used in way to return altered noisy count during node creation for a leaf region. More the budget been allocated, lesser is the deviation from the original matrix. In other words, the noise here is sampled from a Laplace distribution where the scale is inversely proportional to the privacy budget.

C) **Edge Rearrangement**: The algorithm being implemented selects a particular score for a given leaf region with probabilities being drawn from exponential distribution and then assign edges to a new NULL matrix.

- Some Terms Useful from perspective of implementation:
  i. R: original graph
  ii. $\tilde{R}$: Distorted graph
  iii. x: $R_{ij} = 0 \wedge \tilde{R}_{ij} = 0$
  iv. y: $R_{ij} = 0 \wedge \tilde{R}_{ij} = 1$
  v. z: $R_{ij} = 1 \wedge \tilde{R}_{ij} = 0$
  vi. w: $R_{ij} = 1 \wedge \tilde{R}_{ij} = 1$

We will be storing the latest value of x, y, z, w associated with a score while iterating through each possible $\tilde{R}$ as this information will be useful for matrix reconstruction.

- Matrix re-construction: Since now we have

*: The synthetic data has 10% of its entire cells filled with 1s.

the information of number of 1s and 0s to be present in the newly formed matrix and a score s which is selected, we have $^{x}C_{y}$ possibilities to fill the new matrix with 1s. We will use the stored values of x, y, z and w to get the indices to be replaced with a value of 1.

- Privacy Budget allocated (ε_A): Here the exponential of privacy budget is taken and then multiplied by the number of favorable events as a result of exponential mechanism.

Limitations on implementation aspect:

a) For a non-interactive implementation of the paper, it is not feasible to assign a dynamic threshold for leaf regions as if it is based on the dimension of the region being explored and it insists a strict lower bound on the number of 1s or it will work only for dense graphs, so in the implementation this threshold can be statically set according to the dataset.

b) In assignment of edges, to apply exponential mechanism, for calculation of favorable events we need to iterate through all the (n*n)! combinations of 0s and 1s if n is the size of a leaf region. So there can be cases where this size can be large or in other words the current time complexity of this function (|L|*(n*n)!) where |L| is the number of leaf regions is not feasible for even a small n = 5, so I put a bound for assigning edges with an exponential mechanism only for n<=3 and for rest I simply pick from a uniform random distribution the x and y co-ordinates for 1s equal in number to the number of 1s in that region of actual graph.

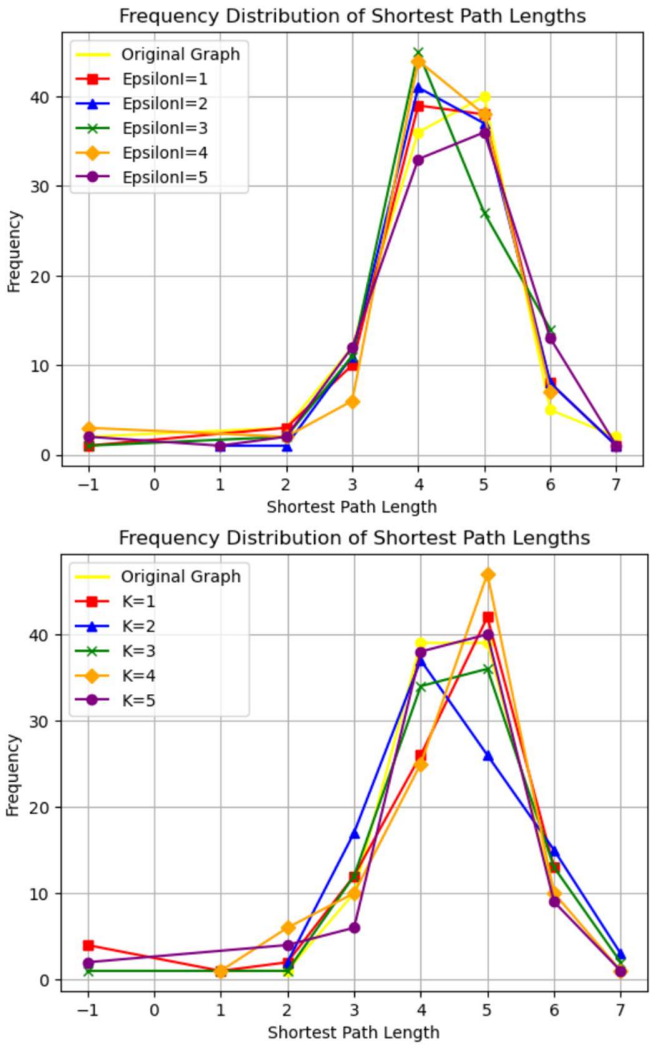
III. Datasets and Results

For the experimental evaluation we have primarily used the dataset Congress Twitter obtained from Stanford Large Network Data Collection (SNAP) which has the information of the twitter interaction between members in the 117th United States Congress and a synthetic dataset* with 1000 nodes and a pre-defined density of 1s in the 2-D adjacency matrix. The density is chosen in a way that there is sufficient depth for the quadtree to explore and demonstrate the working of dense region exploration algorithm which required it to be a little less denser but not that less such that reconstruction has high error associated with it.

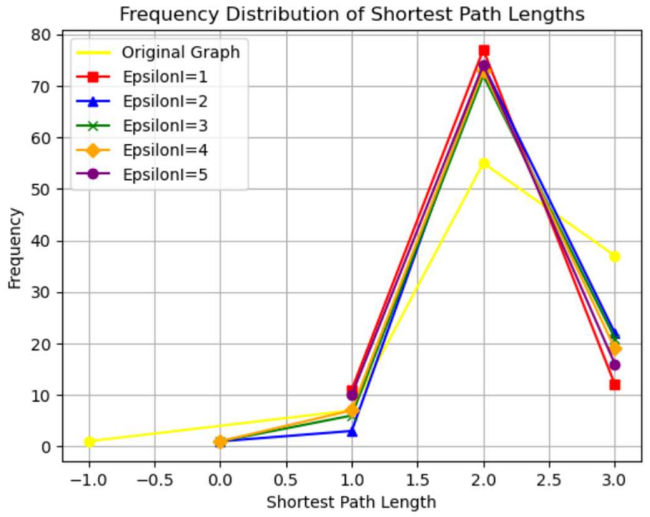
Metrics Used:

a) **Shortest Path Length**: We will plot the frequency of the length of shortest path for 100 randomly chosen pair of nodes versus the Privacy budget(ε_l) been allocated and then we plot it's variation with change in the correlation parameter.

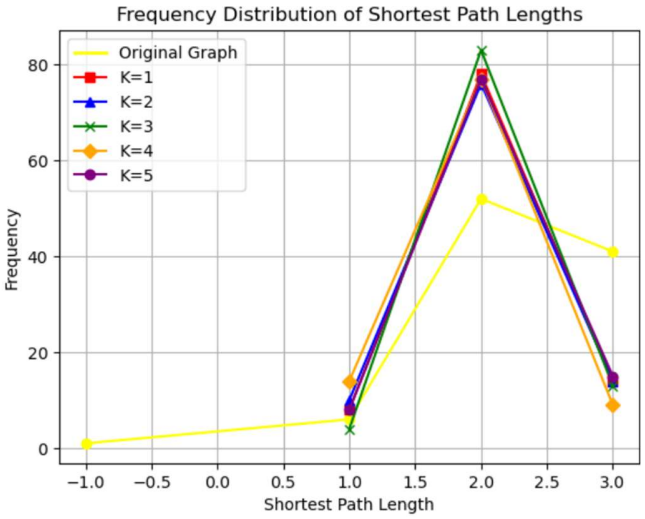
- Plot for the synthetic data*



- Plot for Congress-Twitter Data:



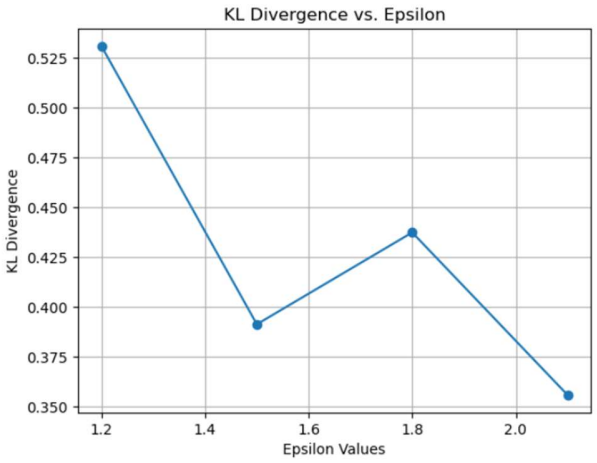
*: The synthetic data has 10% of its entire cells filled with 1s.



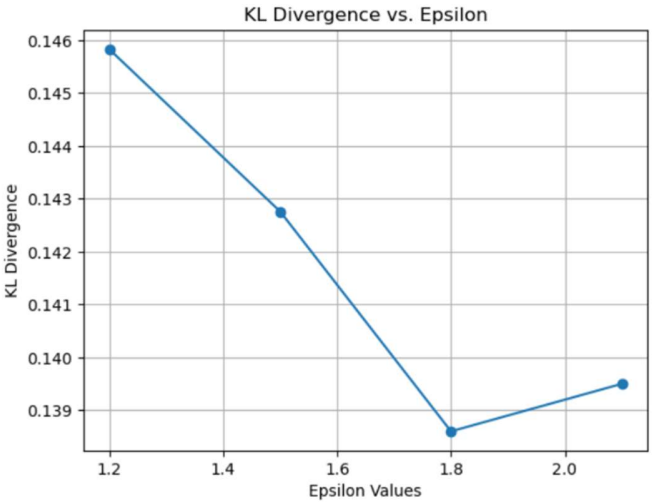
Proof of Hypothesis: From the above results, we can state that our algorithm has a very good utility for queries of shortest path length along with preserving privacy at the same time. Also, we can say that with more privacy being allocated, the results get similar to the original one.

b) **KL-Divergence**: A utility metric used to compare the similarity in degree distribution in original and the distorted adjacency matrix. Now consider original network as N and the newly generated output network is $\tilde{N}$, and their degree distributions are given by D(N) and $D(\tilde{N})$ respectively, then $KL_{N,\tilde{N}} = \sum_{i=0}^{i=|V|-1} D(N)[i] * \log\left(\frac{D(N)[i]}{D(\tilde{N})[i]}\right)$

- Plot for synthetic data*:

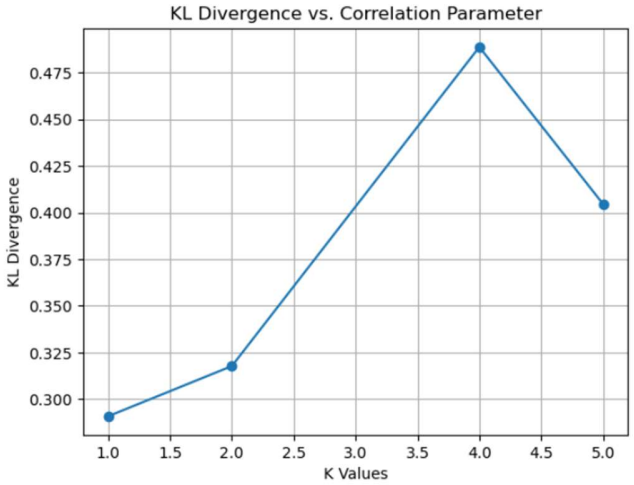


- Plot for Congress Twitter Data:

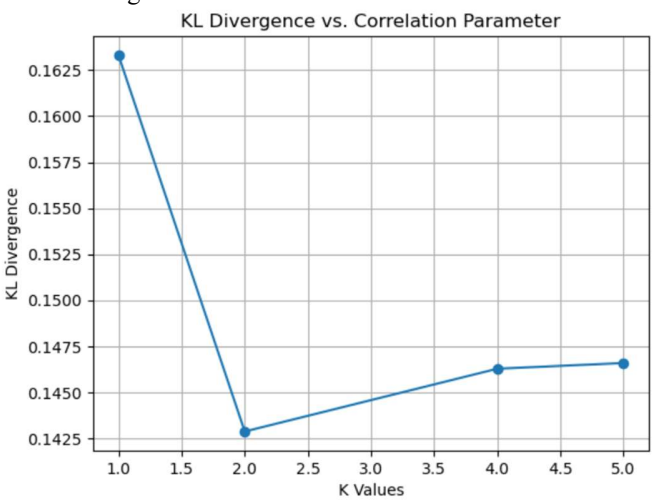


Proof of hypothesis: As we can see a general trend of decrement in KL-Divergence with increase in epsilon values which signifies the fact that with increase in epsilon value the similarity between original and distorted graph can increase.

Plot for synthetic data*:



Plot for Congress Twitter Data:



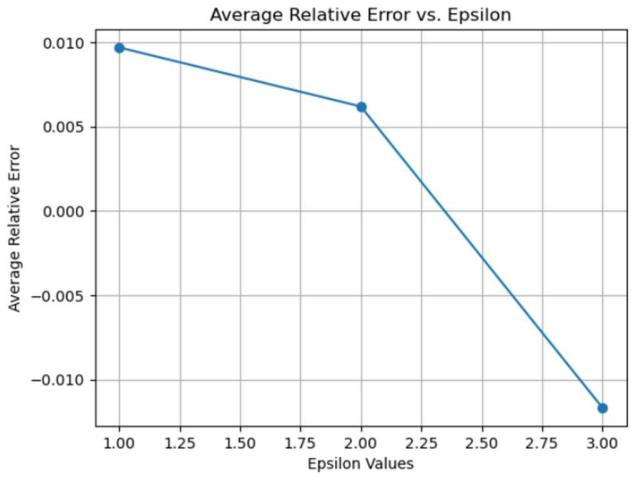
Proof of Hypothesis:
Our hypothesis of decrease in the utility of data with the

*: The synthetic data has 10% of its entire cells filled with 1s.

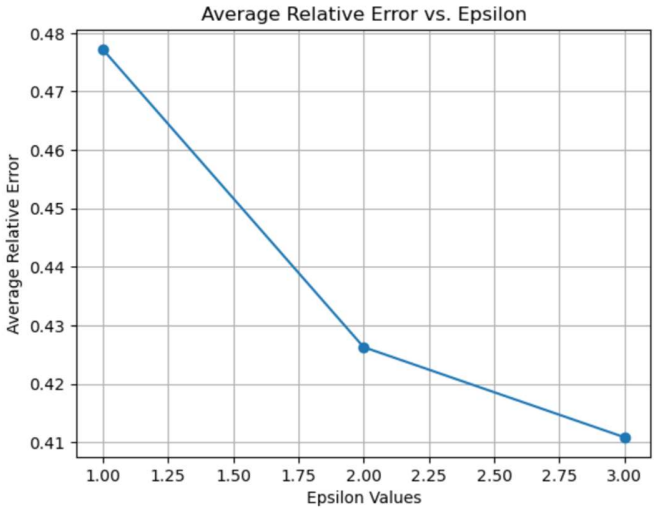
increment in the correlation is verified here. We can see a trend with increment in correlation parameter which will result in achieving a k-times more stricter privacy constraint corresponding to an increment in the dissimilarity between the network graphs measured by the KL-Divergence values.

c) **Cut-Query**: Used to measure similarity in the network structure of two graphs. Lesser the average relative error of a graph w.r.t original graph, the more it resembles the structural characteristics of the original graph. Average relative error while measuring the number of edges crossing a given cut defined as:

- Error $(Q_{S,T}(\tilde{G})) = |Q_{S,T}(\tilde{G}) - Q_{S,T}(G)|/ Q_{S,T}(G)$
  Where S, T are the set vertices included in the cut and outside it respectively. $Q_{S,T}$ refers to the number of edges crossing the cut.
- The relative error is averaged over a series of 10 runs of the algorithm versus the privacy budget allocated.
- Plot for synthetic data* (threshold = 1)



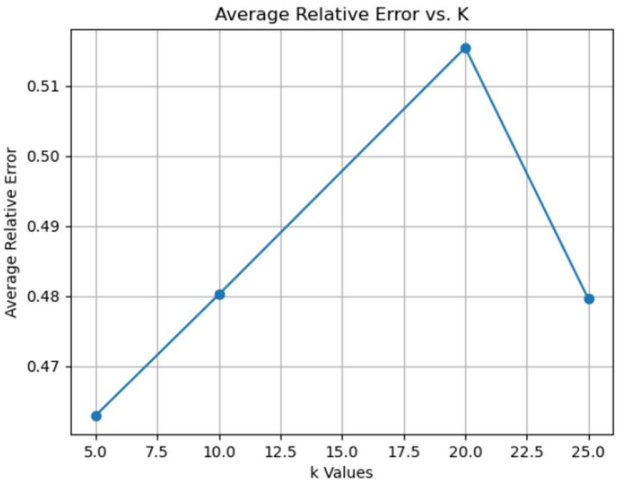
- Plot for Congress Twitter Data:



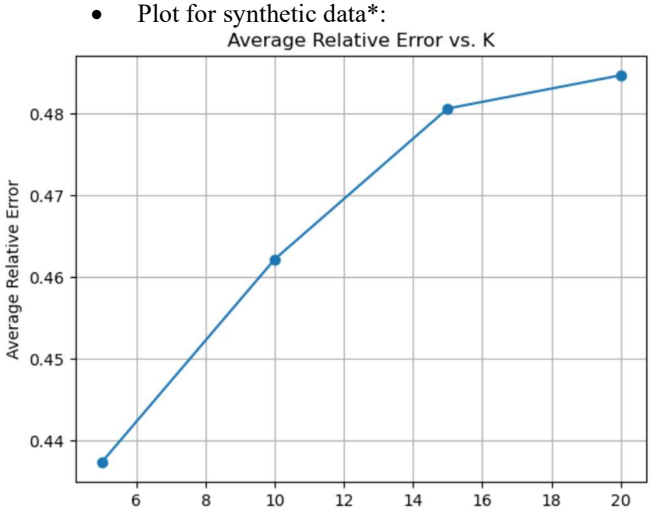
Here we can see the total privacy budget allocated on the x-axis and see the variation of the average relative error with it.

Proof of Hypothesis:
The results strengthen our hypothesis of with increment in privacy budget the similarity between distorted and original graph can increase and hence they are more likely to exhbit network properties such as one taken here (average number of edges crossing a cut of given size) similar to each other resulting in a decrease in the average relative error.

- Plot for Congress Twitter Data:



- Plot for synthetic data*:



Here on x-axis we have 4 different k-values[5, 10, 15, 20] and a $\varepsilon = 10$ and from Theorem 1 mentioned above the total budget allocated is 10/k.

Proof of Hypothesis:
Here as value of K increases, the correlation increases and hence the privacy constraint becomes more stringent with lesser effective privacy budget being allocated leading to a worse utility marked by increase in average relative error.

Conclusion:

In brief, we came to know the importance of sanitization algorithms and what is the main contribution given in this field

*: The synthetic data has 10% of its entire cells filled with 1s.

by our algorithm. We then dealt with all the 3 major sub-algorithms of DER and implications of privacy budget allocated to them right from identifying a good label for vertices, to explore dense regions and then finally assigning edges using exponential mechanism.

On a concluding remark, we accessed the utility of the sanitized graph or the network data for answering queries while at the same time ensuring differential privacy. Also, we also depicted that if the correlation in the data can be measured then, still we can use the given differential privacy algorithm with a slight change in privacy budget.

References:

1. Chen, Rui & Fung, Benjamin & Yu, Philip & Desai, Bipin. (2014). Correlated network data publication via differential privacy. The VLDB Journal. 23. 653-676. 10.1007/s00778-013-0344-8.

*: The synthetic data has 10% of its entire cells filled with 1s.