Final Project Report

— Analysis of Differentially Private New York City Taxi Dataset generated using PrivBayes

Lai Wei(lw255), Naman Jain(nj83), Yanping Zhang (yz488)

Motivation

The concept of differential privacy is becoming popular as it provides guaranteed privacy in terms of releasing the datasets. Recently, a lot of differentially private algorithms have been developed which give us varying accuracy (performance) and this accuracy varies with the privacy budget. Therefore, it is important to choose a suitable differentially private algorithm for the datasets that need to be released. Our group aims to release the New York City Yellow taxi trip dataset after using a differentially private algorithm on the dataset. The way that this dataset is currently released is not exactly private and can be used for malignant activities. From a layman's perspective, it may look benign as there's no information that identifies the customer or the driver. However, consider the following case. One of the attributes in the dataset is Time and date of pickup and dropoff. There's information about Pickup regions and Dropoff regions. If a person goes to the same place (maybe his/her office) at the same time everyday, it will not be difficult to track down this person. To avoid this, we need to release a differentially private NYC Yellow taxi dataset, so that this information cannot be misused.

Introduction and Problem Statement

The NYC Taxi & Limousine Commission releases the collected Yellow taxi trip records on its website every month which can be used to conduct various analysis and research. However, as talked about earlier, these trip records include sensitive attributes such as pick-up and drop-off dates/times, trip distances, passenger counts etc. Directly releasing this dataset may leak exclusive information about the customers and maybe even the drivers. Therefore it is important to find a differentially private algorithm to release this dataset.

The choice of the algorithm highly depends on the schema of the dataset. In this case, the NYC Taxi dataset is high-dimensional, i.e., it has a lot of attributes (17 to be exact) and over eight million records for a single month. It turns out that most of the differentially private algorithms are not suitable for releasing high dimensional data due to two challenges. The first challenge is the output scalability. Most of the existing algorithms represent the input dataset as a high dimensional vector of size equal to the dataset's domain size.

In high dimensional datasets, the domain size is large. This is because various combinations of data could occur and each one of them has to be taken into account. For example, if there are 10 attributes and each attribute has 10 possible categorical values, we still have 10^{10} combinations. Hence, the domain size is directly proportional to an exponential factor of the number of attributes. This may cause a problem as this implies a slow execution. The second challenge, that researchers generally face in such situations, is maintaining the appropriate signal-to-noise ratio. Signal-to-noise ratio is generally determined by the ratio of the original data size and the scalability of the dataset, which is defined earlier. The average count in each entry in the vector may be really small compared to the injected noise which could make the released data nearly useless.

To address the aforementioned problems and challenges, we decided to use the PrivBayes algorithm to release the NYC Yellow taxi dataset. The PrivBayes algorithm approximates the distribution of the high dimensional data using a set of low dimensional distributions. The algorithm then inject noise to these low dimensional distributions and sample the synthetic data from these distributions. Due to the fact that we are working on low dimensional distributions and that we never compute the actual high dimensional distribution of the original data, we can avoid the output scalability and signal-to-noise problem.

A small problem that we faced while implementation was the Date/Time attribute type. We were unsure about how to approximate this attribute under the PrivBayes algorithm and also how do we inject noise to such an attribute. Our first option was using the Unix Timestamp and then treating each value as an integer. However, as these values are huge, the calculation of bins, used in approximation of this data, would be tedious and time consuming and would not give us any useful result. So, we decided to replace the date/time attributes (both pickup and dropoff) with Day of the Week and Time of the Day attributes, hence, converting the date/time attribute to categorical data type. Also, this is better than publishing the the date/time as this makes the dataset less specific and less useful to adversaries.

In further sections, we will focus on the exact process that constitutes the PrivBayes algorithm. We will also discuss how we adapted the PrivBayes algorithm for the NYC yellow taxi dataset in our methodology and experimental design sections. Finally, we will analyse the accuracy of our algorithm based on the a set of evaluation metrics in our results section.

Related Work

Analysis on the NYC taxi dataset, similar to ours, has been conducted by researcher Todd W. Schneider. He took the NYC taxi dataset and Citi Bike Ride dataset and tried to find out that at what time of the day it will be better to go for either of these options. Apart from the time of the day, his analysis was based on various other attributes such as the source and the destination of the taxi, which day of the week it is, trip distance, weather conditions (precipitation and other factors) and a few more. The major difference between his analysis and ours is that he did not focus on the privacy aspect. He did not generate the synthetic data and focused only on the analysis. Hence, his methodology (Monte Carlo Simulation) is completely different than ours (PrivBayes Algorithm).

A bunch of research work has been done to release high-dimensional datasets. Some find the subsets of the dimensions. Some post-process the released dataset according to their consistency. Others avoid dimensionality issues by using data reduction techniques. But all of them fail to handle either the scalability challenge or the signal-to-noise challenge or both, which can be solved using PrivBayes algorithm.

Methodology

The PrivBayes algorithm consists of three major steps. First, it uses a differentially private method to construct a k-degree bayesian network *N* based on the attributes in the original dataset D. k-degree bayesian network refers to a bayesian network where each node (in this case attribute) has k number of parents (depends on k attributes). In the second step, the algorithm uses another differentially private method to generate the conditional distributions of the original dataset D using the bayesian network generated in the first step. The method further injects noise to these conditional distributions to form a noisy version of the conditional distribution. Finally in the third step, the algorithm uses the bayesian network created in the first step and the noisy distribution generated in the second step to approximate the distribution of the original dataset D and generates the synthetic dataset by sampling tuples from this distribution. Note that if the input/total privacy budget for the entire algorithm is ε, the privacy budget for the first step, i.e., $\varepsilon_1 = \beta\varepsilon$, where β is some constant in (0,1). Similarly, the privacy budget for the second step,i.e., $\varepsilon_2 = (1-\beta)\varepsilon$. Reasonably, β should be equal to 0.5 in order to provide more stable results related to step 1 and step 2 both. Hence, both first and second step consume ε/2 of the privacy budget. The following paragraphs consist of the details of each steps algorithm.

Step 1: *Construct a k-degree bayesian network N*

The aim of constructing the bayesian network is to use a set of low dimensional distributions to approximate a high dimensional distribution, which influences the accuracy of the synthetic dataset sampled according to the bayesian network. Therefore, it is important to select Attribute - Parent pairs and the degree of bayesian network carefully. KL-divergence from the original distribution Pr[A] and the approximated distribution PrN[A], is used to measure the quality of Bayesian Network.

$$D_{KL}(\Pr[A]||Pr_N[A]) = -\sum_{i=1}^{d} I(X_i, \prod i) + \sum_{i=1}^{d} H(X_i) - H(A)$$

The original data distribution decides the second term, so we only need to construct a bayesian network whose sum of mutual information (represented by I()) of edges is maximum. The optimal 1-degree bayesian network can be constructed by finding the maximum spanning tree, where the weight of edge is mutual information. However, constructing a k-degree bayesian network is NP-hard, when k>1. PrivaBayes uses the following GreedyBayes algorithm to find a high-quality bayesian network. This algorithm ensures that the following two requirements are satisfied. Firstly, the bayesian network turns out to be a directed acyclic graph (DAG). Secondly, the algorithm properly chooses the degree k, which satisfies θ-usefulness and guarantees the noise injected will not overwhelm the data.

ALGORITHM 4: GreedyBayes (D, θ): returns $\mathcal{N}$

initialize $\mathcal{N} = \emptyset$ and $V = \emptyset$;
randomly select an attribute X_1 from $\mathcal{A}$; add $(X_1, \emptyset)$ to $\mathcal{N}$; add X_1 to V;
for $i = 2$ **to** d **do**
    initialize $\Omega = \emptyset$;
    foreach $X \in \mathcal{A} \backslash V$ **do**
        find all maximal parent sets of X, that is,
        $\top(X) = \texttt{MaximalParentSets}(V, \frac{n\varepsilon_2}{2d\theta|\text{dom}(X)|})$;
        if $\top(X) = \emptyset$ **then**
            add $(X, \emptyset)$ to Ω;
        else
            foreach $\Pi \in \top(X)$ **do** add (X, Π) to Ω;
    select (X_l, Π_l) from Ω, using the exponential mechanism with privacy budget $\varepsilon_1/(d-1)$;
    add (X_l, Π_l) to $\mathcal{N}$; add X_l to V;
return $\mathcal{N}$;

However, the mutual information of edges is sensitive, and selecting the best edge is also data-sensitive. So we should use Exponential Mechanism to choose the edge randomly.

Step 2: *Compute noisy conditional distributions*

After constructing the bayesian network, we need to get a set of noisy low-dimensional conditional distributions to compute the approximate distribution. First, we need to materialize the joint distribution of Attribute-parent Pairs $Pr^*[\mathbf{X}_i, \boldsymbol{\Pi}_i]$, then we inject Laplace noise into the above joint distribution to get a noisy joint distribution $Pr^*[\mathbf{X}_i, \boldsymbol{\Pi}_i]$. Finally, we can derive a noisy conditional distribution $Pr^*[\mathbf{X}_i|\boldsymbol{\Pi}_i]$.

$$\begin{aligned}\Pr[\mathcal{A}] &= \Pr[X_1, X_2, \ldots, X_d] \\ &= \Pr[X_1] \cdot \Pr[X_2 \mid X_1] \cdot \Pr[X_3 \mid X_1, X_2] \ldots \Pr[X_d \mid X_1, \ldots X_{d-1}] \\ &= \prod_{i=1}^{d} \Pr[X_i \mid \Pi_i].\end{aligned}$$

Step 3: *Sampling data from the distribution*

By now, we can sample data from the approximate distribution. With the property of bayesian network, we can sample attributes from their noisy conditional distributions independently. Also, as the bayesian network is a Directed Acyclic Graph, we can sample each attribute in increasing order and when we sample the next attribute, we ensure that we have

sampled all of its parent attributes. This allows us to sample each attribute from noisy conditional distribution $Pr^*[\mathbf{X}_i|\boldsymbol{\Pi}_i]$, given the sampled distributions.

Privacy Guarantee: In step 1, we access original dataset when we select edges between attributes according to the mutual information metric and we adopt exponential mechanism to randomly select them. In step 2, when we materialize the joint distribution of Attribute-Parent pairs, we use the input dataset. So, we use Laplace Mechanism to materialize the distribution. As for step 3, we just sample the data from the noisy distribution and we do not access the original dataset. Hence, it does not consumes privacy budget.

Experimental Design & Results

Test the performance and the accuracy of the released datasets

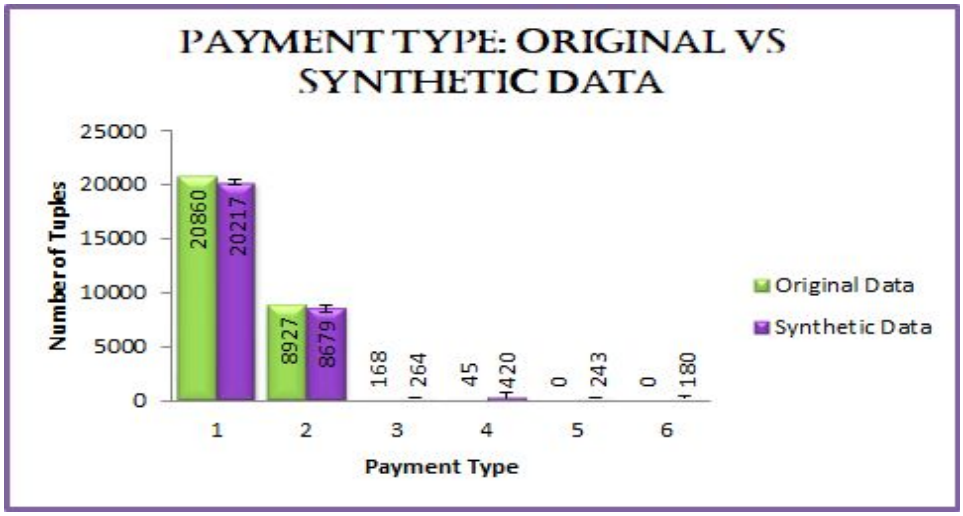
We first plotted the histograms for 5 sensitive attributes that hold significant importance. We ran our algorithm 5 times and computed the average counts for each attribute. The input ε for our experiments is 0.8. Below are the results of our experiments.

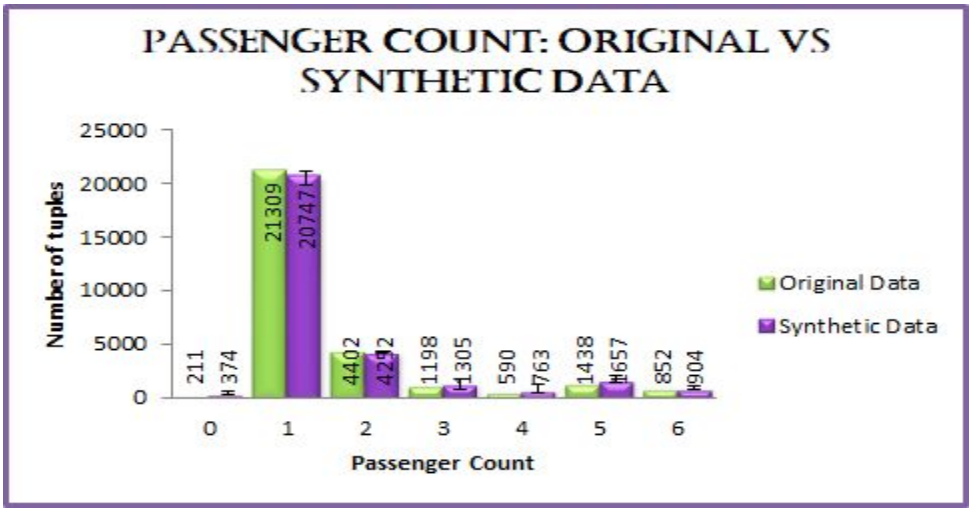
Bayesian Network constructed from step one of the algorithm:

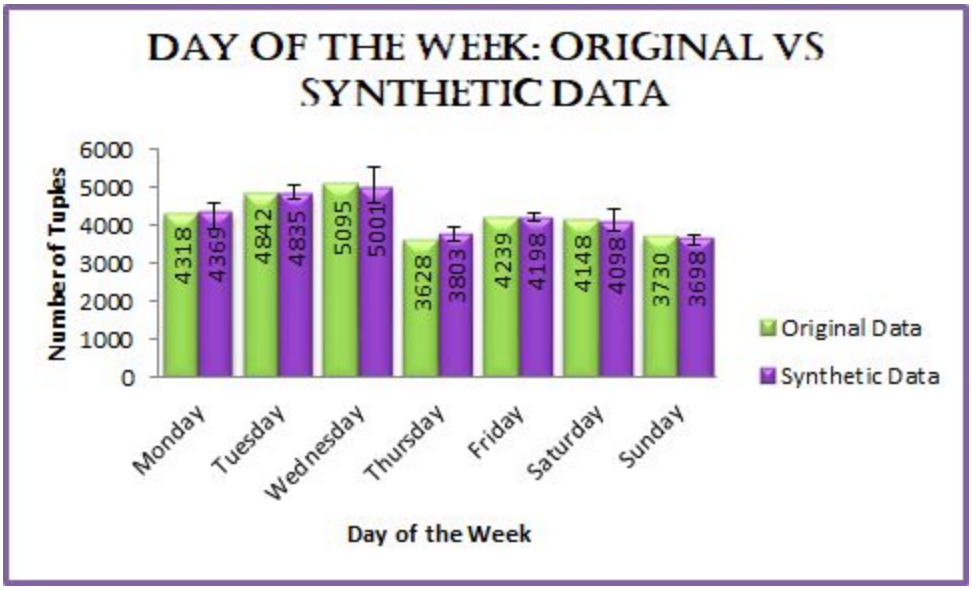
Attributes	Parent(s)
improvement_surcharge	None
mta_tax	improvement_surcharge
tolls_amount	ma_tax, improvement_surcharge
RatecodeID	tolls_amount
extra	RatecodeID, mta_tax, improvement_surcharge
time_of_the_day	extra , tolls_amount
DOLocationID	time_of_the_day
PULocationID	DOLocationID
trip_distance	PULocationID, DOLocationID
fare_amount	PULocationID, DOLocationID
day_of_the_week	extra, improvement_surcharge, mta_tax
total_amount	fare_amount

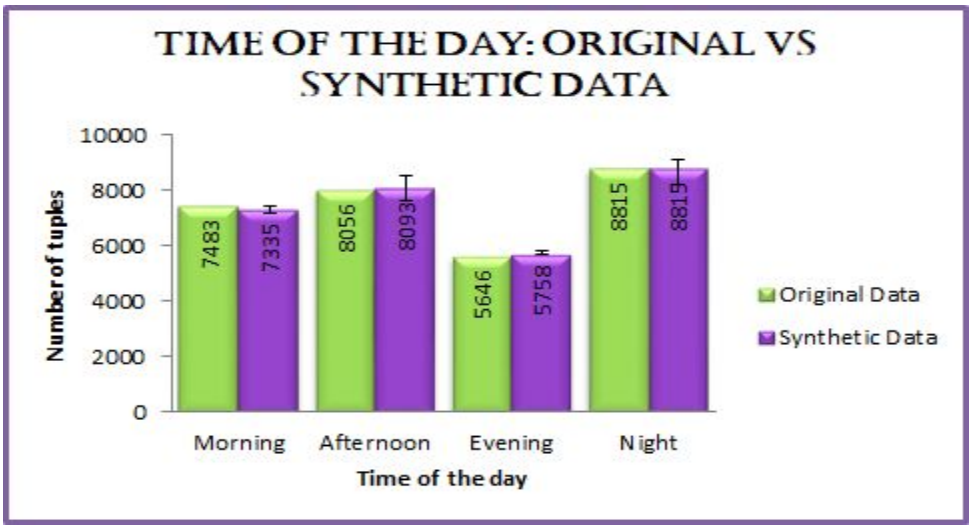
VendorID	PULocationID, extra, improvement_surcharge
passenger_count	VendorID, PULocationID, fare_amount
tip_amount	PULocationID, DOLocationID
payment_type	total_amount
store_and_fwd_flag	PULocationID, DOLocationID, passenger_count

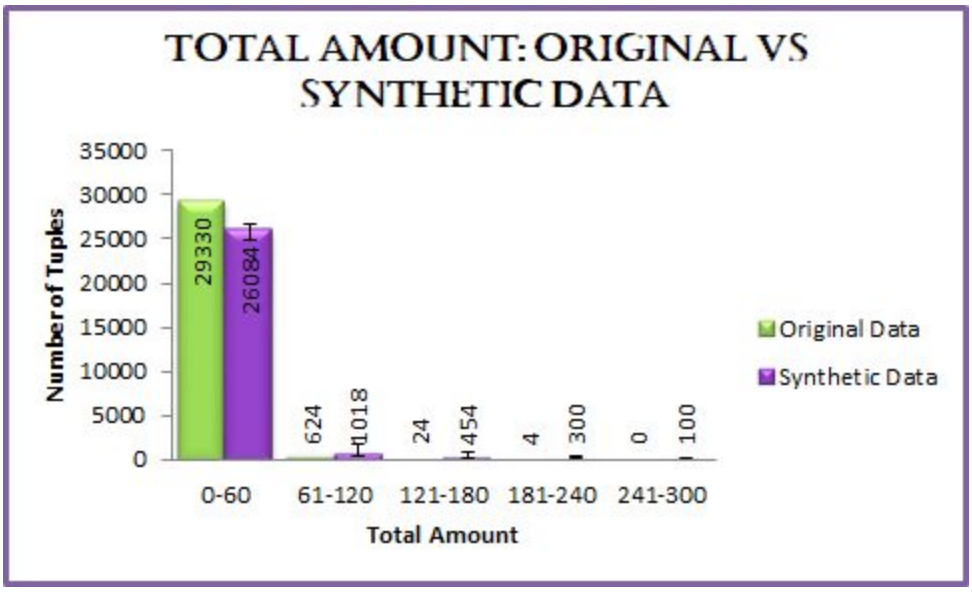
The bayesian network constructed above demonstrates that the first step of the algorithm is able to capture the relationships between the attributes in the NYC yellow taxi datasets. For example, it is reasonable to infer that the parents of trip_distance are PULocationID and DOLocationID because the trip distance directly depends on the pickup/dropoff locations. It is also reasonable to infer that the parent of total_amount is fare_amount since fare_amount takes up a large portion of the total amount. Following are the histograms of the released synthetic datasets (average calculated over 5 different synthetic datasets; error is shown with error bars).






DAY OF THE WEEK: ORIGINAL VS SYNTHETIC DATA
Number of Tuples
6000
5000
4000
3000
2000
1000
0
4318
4369
4842
4835
5095
5001
3628
3803
4239
4198
4148
4098
3730
3698
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day of the Week
Original Data
Synthetic Data


TIME OF THE DAY: ORIGINAL VS SYNTHETIC DATA
Number of tuples
10000
8000
6000
4000
2000
0
7483
7335
8056
8093
5646
5758
8815
8819
Morning
Afternoon
Evening
Night
Time of the day
Original Data
Synthetic Data


TOTAL AMOUNT: ORIGINAL VS SYNTHETIC DATA
Number of Tuples
35000
30000
25000
20000
15000
10000
5000
0
29330
26084
624
1018
24
454
4
300
0
100
0-60
61-120
121-180
181-240
241-300
Total Amount
Original Data
Synthetic Data

Using single synthetic dataset gives us comparatively worse results than using the average of 5 synthetic datasets. Error bars represent the range of values taken by the attribute with respect to the average value. From the graphs above, it is evident that this range is small and hence, the accuracy is very good for privacy budget = 0.8, which is a very practical privacy budget.

Comparison between two different bayesian network

We further compare the results of two different bayesian networks with the same degree $k = 4$ and input $\varepsilon = 0.8$. Below are the subsets of the graphs that show that they are different from each other:

Bayesian Network 1

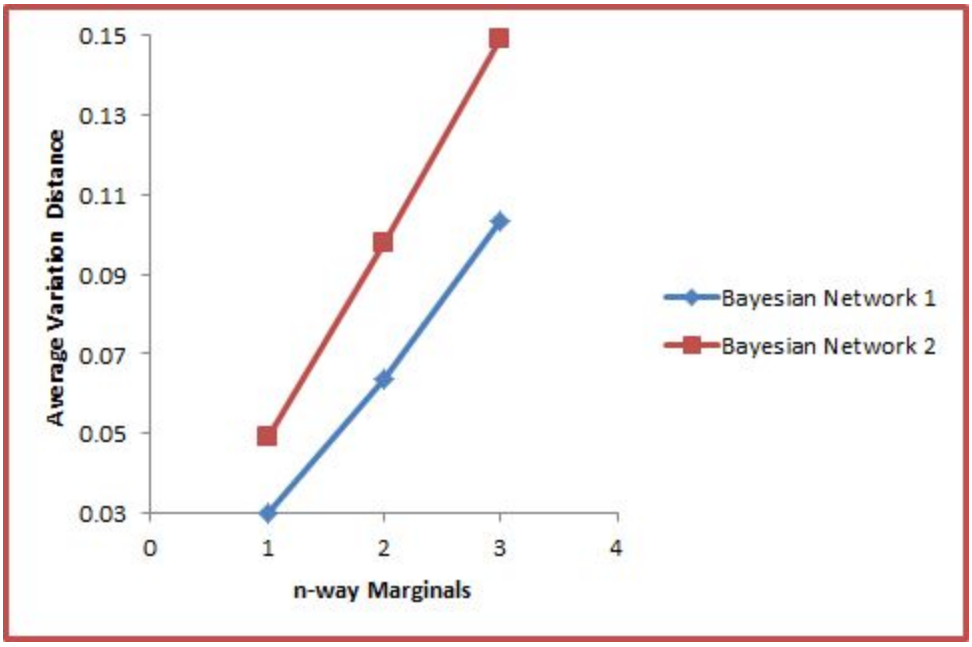
Attributes	Parent(s)
passenger_Count	None
fare_amount	trip_distance
payment_type	total_amount
tip_amount	fare_amount, trip_distance
PULocationID	DOLocationID

Bayesian Network 2

Attributes	Parent(s)
Passenger_count	VendorID, extra
fare_amount	None
payment_type	PULocationID
tip_amount	day_of_the_week
PULocationID	extra, tolls_amount

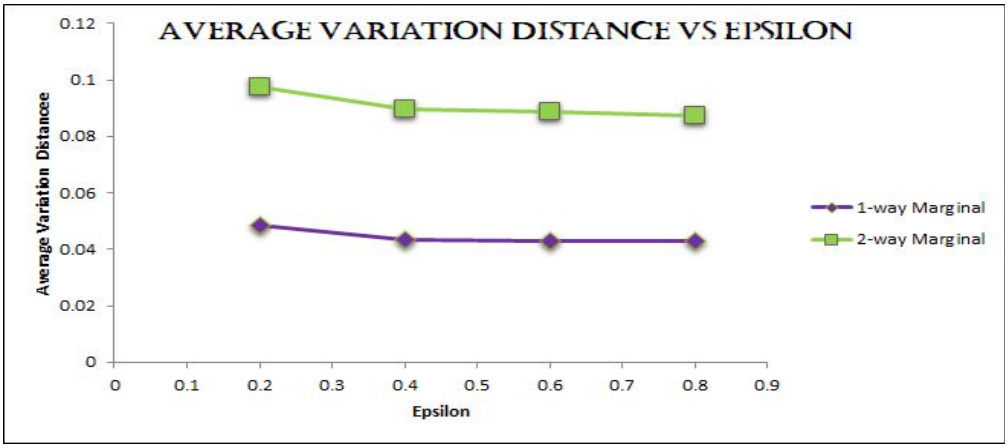
From the tables above, we can infer that bayesian network 1 is more logical than bayesian network 2. The reason is that it is rational to think that payment_type depends on the total

amount since passenger may pay cash when the total amount is small and with credit card when the total amount is large. It is reasonable to infer that the tip_amount depends on the fare_amount since it is generally a percentage of the fare_amount. However, in bayesian network 2, most of the relationships are not logical and do not make sense like the pickup location ID depending on the toll amount. The vice-versa may still make some sense. Below is the result for the average variation distance versus the alpha-way marginals.



We see that the accuracy of bayesian network 1 is better than the accuracy of bayesian network 2. The accuracy of noisy marginals is measured by computing the half of the L1 distance between the noisy marginals and the noise-free marginals, which is the total variation distance. The error metric is the average accuracy over all marginals, called average variation distance.

Change in Average Variation Distance for n-way marginals with change in epsilon



The graph above demonstrates that the average variation distance of 1-way marginal is smaller than the average variation distance of 2-way marginal under different epsilons.

Conclusions

The results of our experiment demonstrate that PrivBayes algorithm achieves great accuracy for most of the attributes in the NYC yellow taxi data based on the observation from the histograms. We also demonstrated that the variations in the bayesian network can affect the accuracy. A bayesian network that is logical often results in a better accuracy. We showed that the change in average variation distance of n-way marginals is stable and remains low, which is acceptable.

Open Directions

We know that PrivBayes gives very good results for all queries in general. However, due to this property of query-independence, it may not give us the best results if we want to focus only on specific queries. Although, it is mentioned in the PrivBayes paper that this "error" is negligible, we think that there's scope for improvement. Hence, one possible future direction is to customize the PrivBayes analysis to a query workload taken as input,i.e., making the algorithm query-dependent. Also, there are various different datasets available on the NYC Taxi and Limousine Commission's website, like Green taxis and others. Researchers can analyse how this algorithm can be applied on those datasets and then can draw a comparison between these taxi datasets to see when will it be a better option to use either one of the taxis. Lastly, it would be interesting to follow the analysis undertaken by researcher Todd W. Schneider (maybe even extend it) but on the generated synthetic data and then compare the accuracy of the results.