Differentially Private Mobility Modeling with Public Data

James Dreben

Introduction

Models of human mobility have broad utility, but also significant potential privacy loss. To realistically model the movement of people in a city, researchers have necessarily learned by investigating the movements of real individuals. How can we ensure an adversary cannot discover and track a particular individual's movements in models of human mobility? Only by modifying the empirical data and distributions in a differentially private manner, such that any one person's presence does not significantly affect analyses, can we produce a model of human mobility in which unusual behavior is measurably protected and collective trends are still captured.

Previous researchers have constructed an approach, *WHERE*, to modeling the movement of people over time in metropolitan areas. In this and prior work done in collaboration with AT&T, analysis of Call Detail Records (CDRs) has allowed for the discovery of realistic spatiotemporal distributions that capture collective trends in a population. With these spatiotemporal distributions as input, *WHERE* creates a synthetic population and synthetic CDRs, the accuracy of which can then be compared to real CDRs, to produce a realistic model of mass movement.

However, while *WHERE*'s procedure anonymizes the personally identifiable information collected in call data, it does not produce its spatiotemporal distributions from the call data in a differentially private manner. Without differential privacy, the potential to discover and track an individual through the *WHERE* approach remains.

DP-WHERE is a modification of the prior approach that obfuscates the true data learned from the population in a differentially private manner. Prior work has shown the accuracy loss of adding differential privacy to *WHERE* is minimal by comparing to true CDR inputs. Here I first implement and validate that research, drawing my spatiotemporal distributions from Census data and prior research rather than true CDRs.

Originally I had hoped to devise a predictive model of traffic congestion using Google's traffic data to compare the spatiotemporal accuracy of *WHERE* and *DP-WHERE*, as without real CDRs, researchers cannot validate my implementation's distance from truth beyond statistics such as daily range. However, the primary focus of this project became implementing and modifying the differentially private *WHERE* distributions to explicitly use the public data alone.

Research Question

Can I implement and validate *DP-WHERE* using only Census data and published characteristics of human mobility? Can a model of Google traffic congestion be used to measure the effects of differential privacy on spatiotemporal mobility distributions?

All data and code for this project can be found at: https://github.com/jdreben/cs227r_Final_Project

Motivation

The model of differential privacy was developed over a series of papers in the 2000s, culminating in Professor Cynthia Dwork's paper that coined the term [13]. Since this project is for her class on differential privacy I focus my review of prior work on those papers specific to my implementation of *DP-WHERE* from Census data.

Utilizing the individual protection of differential privacy when modeling the interactions between largely polarized populations is especially important in mobility modeling, and was the topic of motivation for this project. Previous work has shown that people spend most of their time at a few places and that information derived from anonymized Call Detail Records (CDRs) can accurately characterize many aspects of human mobility [9, 10, 11]. *WHERE* [2] showed that home, work, and other important locations can be inferred from CDR locations via clustering and regression.

I reached out to Professor Rebecca Wright of *DP-WHERE* [1], to attempt to use the synthetic data her system produced from real CDRs. She wrote me back saying that their agreement with AT&T allowed her to publish the paper, but that her fellow author who was working at AT&T left before they could finalize an agreement to release the learned parameters of their models or synthetic call data. Because of this, I have instead derived my spatiotemporal distributions from Census data and prior work, per their suggestion in *DP-WHERE*. The census provides little or no information about the hourly probabilities of a given location. Therefore, to use *WHERE* only with public data, I must make an assumption about the hourly distributions, drawn from prior work [7, 8].

Paper Contents

In Section 1 I describe the details of my implementation of the *WHERE* distributions drawn from census data and assumptions based on prior research on real CDR patterns. In

Section 2 I describe the details of my implementation of differentially private versions of these spatiotemporal distributions as devised in *DP-WHERE*. In Section 3 I describe the details of my implementation of *WHERE*'s synthetic data generation procedure. Section 4 is my conclusion and thoughts on future work.

§1: Spatiotemporal Distributions

Relevant Files

R Writing CSVs.txt / normalize.R *[Pre-Processing]*
home_dist.csv *[from Census, OnTheMap]*
work_dist.csv *[from Census, OnTheMap]*
commute_dist.csv *[from Census, OnTheMap]*
diurnal_pattern.csv *[manual, mirroring prior work]*

§1.1: Home / Work / Commute

I used *OnTheMap* [12], a mapping and reporting tool for Census data, to retrieve counts of homes, job locations, and commute distances per Census grid for counties of New York City (NYC) in 2014. I processed these Shapefiles into into the csvs home_dist, work_dist, and commute_dist. I converted home_dist and work_dist into PDFs by transforming the counts at each grid cell into percent probabilities. It is from these three distributions that I am able to sample a home and a work place for a synthetic user derived from empirical data, which I describe in more detail in Section 3. In *WHERE* they used CDRs to derive these home, work, and commute distributions. I've displayed a density map of the raw home and work counts with their respective legends on the right.

§1.2: Hourly Calls per Location

For each hour of the day, *WHERE* computes a distribution of calls made over the grid cells. Each of those 24 distributions reflects the probability of users being at a given location during that hour. The Hourly distributions are not tied to a specific user, but represent the calling activity across the entire metropolitan area during each hour.

To use *WHERE* only with public data, I must make an assumption about the distribution of hourly calls per location.

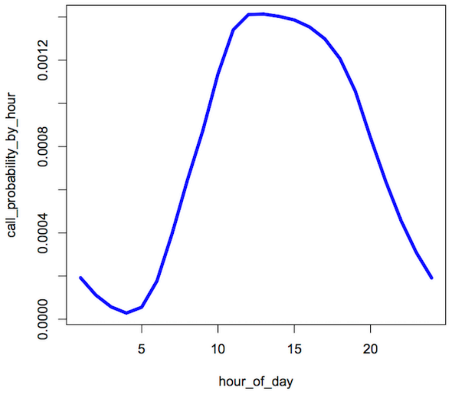
Per the suggestion in *WHERE*, I let the Work distribution double as the hourly calls per location distribution for all hours between 7am and 7pm on Weekdays. Further, I assume the Home distribution provides the hourly distribution at all other times.

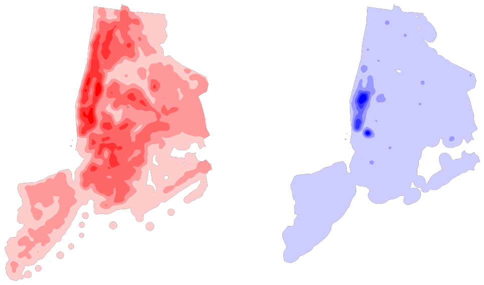
§1.3: Call Times

For each user, *WHERE* computes the distribution of when calls are made throughout the day. Per the suggestion in *WHERE*, I assume a diurnal call time pattern. This seems consistent with results found in *WHERE* and prior work.

More formally, I manually constructed a 24-dimension probability vector (one dimension for each hour) so that each element represents the probability that a user makes calls during that hour, and assigned each user this singular call time class. The smoothed distribution is shown below.

Figure 1: Diurnal Call Distribution





RGB	MinHomeDensity	MaxHomeDensity	Units
255,204,204	5	2934	Homes per square mile
255,204,204	2935	11723	Homes per square mile
255,204,204	11724	26371	Homes per square mile
255,204,204	26372	46877	Homes per square mile
255,204,204	46878	73244	Homes per square mile

RGB	MinJobDensity	MaxJobDensity	Units
204,204,255	5	28880	Jobs per square mile
153,153,255	28881	115508	Jobs per square mile
102,102,255	115509	259888	Jobs per square mile
51,51,255	259889	462019	Jobs per square mile
0,0,255	462020	721903	Jobs per square mile

Figure 2: Home / Work / Hourly Distributions

§1.4: Calls Per Day

The number of calls a user makes on any given day is sampled from a Gaussian (μ, σ) pair that characterizes the user's "Call Class".

In order to move swiftly to the differential privacy aspects of this project, I made some assumptions on the implementation of this distribution, doing my best to draw the analysis from prior work on CDRs. [8] For each synthetic user in my current implementation, I sample uniformly a (μ, σ) pair on the range 1 to 100.

In Section 3 I will discuss how these distributions come together when synthesizing users and CDRs.

§2: Differentially Private *WHERE*:

Relevant Files
dp_home_dist.R [§2.1] [*successfully implemented]*
dp_work_dist.R [§2.1] [*successfully implemented]*
dp_commute_dist.R [§2.2] [*implemented, but inefficient]*
dp_callsperday_dist.R [§2.3] *[successfully implemented]*

§2.1: DP-Home / DP-Work

Let the privacy budget be **ε** home or **ε** work.

Each distribution has a global sensitivity of 2 as any change by an individual from one grid cell to another affects two counts by one.

Looping through each cell, I modify its true count (number of homes / work places in grid cell) by adding Lap(0, 2 / **ε** home or **ε** work) noise. As we learned in class (published result in [4]) this procedure is **ε** - d.p. by the Laplace mechanism.

Post Processing

Post-processing as in [3] must be done as a final step to ensure the distribution is well formed, i.e. monotonically decreasing. As we learned in class, post-processing of the distribution does not diminish the privacy guarantee.

For this procedure I used an implementation written in C++ with an R wrapper provided to me by George Kellaris, but include here a brief summary of the mechanism as I understand it from Hay et. al. [3]

Formally, given some noisy answer s to a query, the objective is to find s′ that minimizes $\|s - s'\|_2$ subject to the constraints.

$s'[i] \leq s'[i+1]$ for $1 \leq i \leq n$, where n is the length of s.
Let $s[i, j]$ be the subsequence of j-i+1 elements:
$<s[i], s[i+1], \ldots s[j]>$

Let $M[i, j]$ be the mean of these elements.

Let $L_k = \min_{j \in [k, n]} \max_{i \in [1, j]} M[i, j]$
Let $U_k = \max_{i \in [1, k]} \min_{j \in [i, n]} M[i, j]$.

The minimum L_2 solution s, is unique and given by:
$s[k] = L_k = U_k$.

For example, if s were < 9, 14, 10 >, the last two elements would be out of order. The optimal modified sequence s′ would be < 9, 12, 12 >.

§2.2: DP-Commute Distance

Let the privacy budget be **ε** commute.

Implications of Census vs. CDR Data

For each Census block, *OnTheMap* gives us the total number of jobs reported by those with a home in that block, the number of those jobs that are less than 10 miles away, the number that are 10 to 24 miles way, the number that are 25 to 50 miles away, and finally the number greater than 50 miles.

DP-WHERE computes the commute distance distribution directly from CDRs instead of Census data as I do. They derive data dependent histogram bin ranges, and so must split their privacy budget in half to 1) compute those ranges and 2) compute the counts themselves.

If the histogram bin ranges *OnTheMap* provides (< 10 miles, 10-24 miles, 25-50 miles, > 50 miles) are data independent, then I could more simply add Lap(0, 2 / **ε** commute) noise to each bin, as an individual could at most change the count of 2 bins by changing their commute distance (i.e. GS of 2).

However, for the sake of demonstration, I have followed their method and implemented the **ε** - d.p. commute distance distribution as described in *DP-WHERE*, with two modifications:

1) The authors imposed a d_c x d_c commute grid on the geographical area. I found myself bogged down in attempting to implement the commute grid, so in order to simplify and focus on the differential privacy aspect of this distribution, I instead treat each Census block as a cell.

2) Since I do not have the true commute distance for each indvidual, but rather counts for each (<10 miles, 10-24 miles, >50 miles) bin. I first sample a commute distance uniformly for each user along the ranges [0 - 10), [10 - 24], [25 - 50], (50, 100], so that in expectation the average commute distance for users in each bin will be centered, and so that I can calculate the frequencies and store them in the data structure D_i. I base the cap of 100 from the fact that NYC is ~14,000 mi^2.

The following is my procedure to create a per-commute-grid-cell data-dependent histogram of commute distances in a differentially private way, and then sample from this (normalized) histogram.

Exponential Mechanism (dpmedian)

The differentially private approximation to the median (dpmedian) of the commute distances in grid cell i is computed using the exponential mechanism.

For this implementation I use the following median scoring function:

$score(D_i, r) = -min\ |D_i \oplus X|$
$s.t.\ median(X) = r$

I now analyze the sensitivity. The best possible score for this function comes from the true data, median pair, so if X = D_i then $|D_i \oplus X| = 0$. Similarly, any r ≠ med(X) will be ≤ 0.

For any (X, r), let s = score(X, r). By the definition of the scoring function, we know that there must be some database B such that $|X \oplus B| = -s$ and med(B) = r.

If X,Y are neighboring databases s.t. $|X \oplus Y| = 1$, then $|Y \oplus B| \leq |Y \oplus X| + |X \oplus B| \leq -s + 1$. Since med(B) = r, then we know that score(Y, r) ≥ s - 1.

Finally, | score(X, r) – score(Y, r) | ≤ | s – (s – 1) | = 1 meaning a global sensitivity of 1.

Thus, the exponential mechanism $\mathcal{E}$ is defined to be:
$\mathcal{E}\ (D_i,\ score,\ \boldsymbol{\varepsilon}\ commute\ /\ 2) = output\ r\ with\ probability\ proportional\ to\ e^{(\boldsymbol{\varepsilon}\ commute\ /\ 2\ *\ score(D_i,\ r))}$

In class we showed that the exponential mechanism with a scoring sensitivity of 1 is **ε** - d.p., and thus, as we've only allocated to this procedure half our budget, **ε** commute / 2 is used to calculate dpmedian.

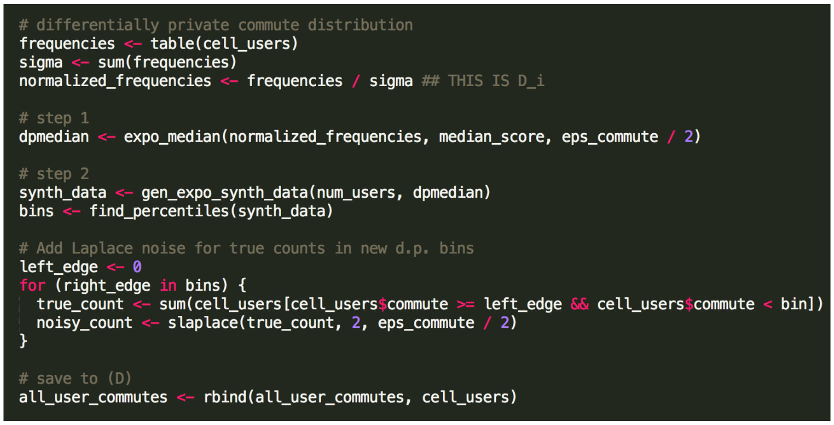
Histogram Bins

In *DP-WHERE*, they suggest adding two commute distances (0 and 0.1 miles) to each grid cell to avoid having empty cells. Now, using the remaining half of the privacy budget, I can determine histogram bins by first generating the synthetic data.

Let η(x) be the (normalized) frequency of the distance x in the dataset Di. If η(x) follows an exponential distribution with rate parameter λ, then $\eta(x) = \lambda e^{-\lambda x}$. The literature shows that this is a popular model for positively skewed distributions such as commute distances.

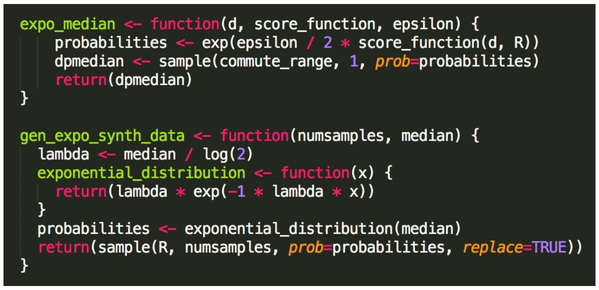
Sample from η(x) with parameter λ = dpmedian / log(2), then determine the 10, 20, 30, …, 90, 95 percentiles of this set.

Figure 3: DP-Commute Distribution Code Snippet (1)



```
# differentially private commute distribution
frequencies <- table(cell_users)
sigma <- sum(frequencies)
normalized_frequencies <- frequencies / sigma ## THIS IS D_i

# step 1
dpmedian <- expo_median(normalized_frequencies, median_score, eps_commute / 2)

# step 2
synth_data <- gen_expo_synth_data(num_users, dpmedian)
bins <- find_percentiles(synth_data)

# Add Laplace noise for true counts in new d.p. bins
left_edge <- 0
for (right_edge in bins) {
  true_count <- sum(cell_users[cell_users$commute >= left_edge && cell_users$commute < bin])
  noisy_count <- slaplace(true_count, 2, eps_commute / 2)
}

# save to (D)
all_user_commutes <- rbind(all_user_commutes, cell_users)
```

Figure 4: DP-Commute Distribution Code Snippet (2)



```
expo_median <- function(d, score_function, epsilon) {
    probabilities <- exp(epsilon / 2 * score_function(d, R))
    dpmedian <- sample(commute_range, 1, prob=probabilities)
    return(dpmedian)
}

gen_expo_synth_data <- function(numsamples, median) {
  lambda <- median / log(2)
  exponential_distribution <- function(x) {
    return(lambda * exp(-1 * lambda * x))
  }
  probabilities <- exponential_distribution(median)
  return(sample(R, numsamples, prob=probabilities, replace=TRUE))
}
```

The distances corresponding to these percentiles form the edge of the histogram bins.

Each bin has a global sensitivity of 2, per my earlier discussion in this section, so applying the Laplace mechanism yields an **ε** commute / 2 - d.p computation of the approximate histogram count.

Each user appears in only one grid cell, so by serial and parallel composition, we achieve **ε** commute - d.p.

§2.3: DP-Calls Per Day

Let the privacy budget be **ε** cpday.

The mean number of calls per day for any user is from the set M = {μ min, … , μ max }.

Standard deviation of calls per day for any user is from set Σ = {σ min, … , σ max }.

Each user's pair of values are rounded to the nearest value in this set (for sensitivity reasons I will discuss below).

Let CountAvgStd(μ, σ) be a | M | x | Σ | matrix with each cell a count of the number of users with a given mean and standard deviation.

Any addition or deletion of calls by a single user can change the mean / standard deviation pair to another pair, decreasing the count for at most one element of the matrix CountAvgStd by 1 (hence the need for rounding) and increasing the count for another element by 1.

This means the global sensitivity of the vector

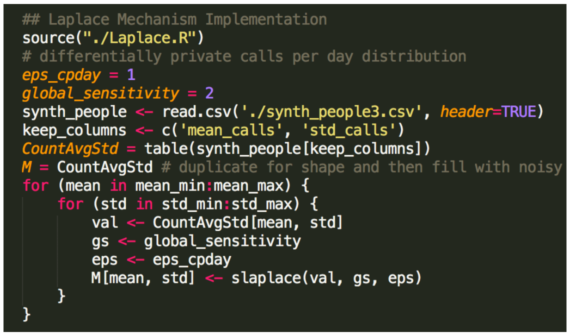
$\langle \mathrm{CountAvgStd}(\mu_{\mathrm{min}}, \sigma_{\mathrm{min}}), \ldots \mathrm{CountAvgStd}(\mu_{\mathrm{max}}, \mu_{\mathrm{min}}) \rangle$ is 2.

Noise Addition

For each (μ, σ) pair, add Lap (0, 2 / **ε** cpday) to the true count of individuals with that mean / standard deviation.

This procedure is **ε** - d.p by the Laplace mechanism and parallel composition.

Figure 5: DP-Calls Per Day Code Snippet



```r
## Laplace Mechanism Implementation
source("./Laplace.R")
# differentially private calls per day distribution
eps_cpday = 1
global_sensitivity = 2
synth_people <- read.csv('./synth_people3.csv', header=TRUE)
keep_columns <- c('mean_calls', 'std_calls')
CountAvgStd = table(synth_people[keep_columns])
M = CountAvgStd # duplicate for shape and then fill with noisy
for (mean in mean_min:mean_max) {
    for (std in std_min:std_max) {
        val <- CountAvgStd[mean, std]
        gs <- global_sensitivity
        eps <- eps_cpday
        M[mean, std] <- slaplace(val, gs, eps)
    }
}
```

§2.4: DP-Call Times

In *DP-WHERE*, they clustered users into one of two classes using differentially private k-means clustering (rather than X-means as used in *WHERE*). From the CDR database, just as in WHERE, they computed the number of calls each user made during each hour of the day and then classified users based on this 24-dimension probability vector. In my case, all users had the same probability vector.

Considering my call time distribution was constructed independent of any individual and is consistent across all users, there is no privacy loss in using it directly. However, the procedure can still be done, and is interesting to consider, as clustering to a single class in a differentially private manner from a known class is in a way a direct measure of the effect of differential privacy on the distribution.

The other distributions were a higher priority in implementation as many of them I had to successfully implement in order to continue progressing, or were particularly interesting in terms of differential privacy, whereas I could already use my non-d.p. call time distribution for synthetic data generation. Beacuse of this, I was unfortunately not able to get to demonstrating the addition of differential privacy to the call times distribution.

§3: Synthetic Data Generation:

Relevant Files
WHERE.R *[**Create**, **Move**]*
synth_CDRs.csv *[ex. product of move]*
synth_people(1-3).csv *[ex. product of create]*

For synthetic data generation, I use a slight variant of the **Two-Place Model: Work and Home**, described in *WHERE*. I used their Two Locations test case, but they also describe an extension of the algorithm for additional places.

Create (minimally modified from *WHERE*)

First, sample a home from home distribution.
Next, sample d from the commute distance distribution conditioned on the given user's home location.
Next, sample a workplace from work distribution.Next, I assign a diurnal call time class (in the paper they used multiple classes clustered from CDRs)
Next, I sample mean and standard deviation from the calls per day distribution, (in the paper they derived this distribution from CDRs, I derive it from their prior work)

I then use their **Move** algorithm exactly as published in *WHERE:*

Algorithm 2 Move

```
for $user = 0 \rightarrow N$ do
    for $day = 0 \rightarrow D$ do
        $callstoday \leftarrow$ normal random number with $\mu$ and $\sigma$ from $pop[user].callsperday$ distribution
        for $call = 0 \rightarrow callstoday$ do
            $calltime \leftarrow$ time from $pop[user].callsbehavior$
            $location \leftarrow$ location using probabilities of $pop[user].work$ and $pop[user].home$ at time $calltime$ from the $Hourly$
            print $user, day, calltime, location$
        end for
    end for
end for
```

Synthetic users are moved between home and work according to **Move**. Movement occurs based on synthetic CDRs representing calls made at different locations at different times. For each day, the number of calls the user makes is sampled from the user's average (μ) and standard deviation (σ) per day. The time in which the call is made is goverened by the diurnal call time distribution. When a call is made, the location of the call is determined to be either home or work according to the probability of a person being in the location at that time of day (i.e., the probability is determined according to the distributions in Hourly as discussed in §1.3).

Each row of the CDRs have columns: user id, day, call time, and location. In *DP-WHERE* they generated 10,000 synthetic users over 30 days in an area of 14,000 mi^2 around New York City. I was also able to generate 10,000 synthetic users over 30 days, but attempting to generalize my code for a coordinate grid rather than the data's natural tract structure was not as relevant to this project as other aspects I could focus on, so I decided to get through the synthetic data generation portion of this project so that I could spend as much time as I could on studying the aspects of differential privacy.

§4: Conclusion / Future Work

R vs. Python

Developing in R proved to be immensely more useful than Python for this project. I'd only ever gone through a tutorial on R before, but between being able to talk with George Kellaris about implementations and the language's built-in statistical functions like `sample`, `quantile`, and `table`, it made me a lot more productive, and I look forward to using it in future projects.

Further Efficiency in Exponential Mechanism

In implementing the exponential mechanism, my algorithm was significantly slowed by the process of generating noisy commute distance rows with median equal to a given candidate. Perhap using a measure of rank rather than xor would have given me greater insight. Since implementing I've had a number of thoughts as to how to make that more efficient, primarily along the line of sorting the true commute distances and breaking possible outputs into ranges with equivalent utility scores and therefore proportional probabilities. Then with a range selected from the mechanism, I could sample an output uniformly from it with a time cost proportional to the number of ranges.

Clustering Call Times in a Differentially Private Manner

I'd very much like to continue working on this project, and an immediate next step will be clustering the call time distributions (though I will cluster into a single class rather than the two they use in *DP-WHERE*). With that distribution made differentially complete, I can truly validate the accuracy claims made in *DP-WHERE*, and develop a new measure of spatiotemporal accuracy via Google traffic data, as I had initially hoped.

Measuring Accuracy Earth Mover's Distance (EMD)

Their measurement of choice, earth mover's distance converted to miles, is particularly fascinating and deserves mention in regard to quantifying accuracy of spatiotemporal distributions. There is a github python package that has implemented EMD <https://github.com/wmayner/pyemd>.

Here is an intuitive explanation: Given two distributions, one can be thought of as a mass of earth spread out in space, the other as a collection of holes. The EMD measures the *least amount of work* needed to fill the holes with earth. Here, a unit of work corresponds to transporting a unit of earth by a measure of ground distance. Further detail is outlined in prior work [1, 2] and readily available online.

The distance between distributions can also be measured in daily range, or the distance between any two points a person visits in a day, and I could also compare via this metric, which they separately release their results for and say characterizes mobility patterns.

With Census data alone, *DP-WHERE* showed that the methods I've outlined in greater detail in this paper can generate synthetic data with accuracy of up to about 8 miles when compared to real CDRs. They also showed they could get an accuracy of about 3 miles with the synthetic data generated by their models learned from the real call data.

They only measured the accuracy of differing levels of epsilon differential privacy on their synthetic data learned from real CDRs. If I went back and imposed a GPS grid on my distributions, I could perform that analysis on the all public version of *DP-WHERE*.

Closing Remarks

There is no question that we have been and will continue to be in the years to come moving toward unprecedented levels of indvidual data collection and tracking at a mass scale. The ability to predict human mobility in metropolitan areas has incredibly varied applications, as do many forms of prediction based on collective human characteristics.

The advent of differential privacy is a truly significant landmark for the modern age. It is a privilege to have studied it, and it's clear to me now that it really is of utmost importance that those who can take the time to understand and explain it do so clearly and often so that people are aware. Perhaps there would be less fear for the future of big data collection.

This was an incredibly fun and challenging final project and class of my college career. I look forward to continuing to learning about differential privacy and am deeply excited to continue developing my model of human mobility.

References

1. [DP-WHERE] Mir DJ, Isaacman S, Cáceres R, Martonosi M, Wright RN (2013) DP-WHERE: differentially private modeling of human mobility. In: IEEE international conference on big data. IEEE Press, New York, pp 580-588.
2. [WHERE] S. Isaacman, R. A. Becker, R. Caceres, M. Martonosi, J. Rowland, A. Varshavsky, and W. Willinger, “Human mobility modeling at metropolitan scales,” in MobiSys ’12.
3. [POST-PROCESSING] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially private histograms through consistency. Proc. VLDB Endow., vol. 3, no. 1-2, pp. 1021–1032, Sep. 2010.
4. [LAPLACE] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In TCC, 2006.
5. [EXPO. MECH] F. McSherry and K. Talwar, “Mechanism design via differential privacy,” in FOCS ’07.
6. [EFFICIENT EXPO. MECH] G. Cormode, C. M. Procopiuc, D. Srivastava, and T. T. L. Tran, “Differentially private summaries for sparse data,” in ICDT ’12.
7. [CALL TIME PUBLISHED STATISTICS] Almeida, J. Queijo, and L. Correia. Spatial and temporal traffic distribution models for gsm. In Vehicular Technology Conference, Sept. 1999
8. [CALL TIME PUBLISHED STATISTICS] J. Candia, M. C. González, P. Wang, T. Schoenharl, G.Madey, and A.-L. Barabási. Uncovering individual and collective human dynamics from mobile phone records. MATH.THEOR., 41:224015, 2008.
9. [MOBILITY CHARACTERISTICS] M. C. González, C. A. Hidalgo, and A.-L. Barabási. Understanding individual human mobility patterns. Nature, 453, 2008
10. [MOBILITY CHARACTERISTICS] S. Isaacman, R. Becker, R. Cáceres, S. Kobourov, M. Martonosi, J. Rowland, and A. Varshavsky. Identifying important places in people’s lives from cellular network data. In 9th International Conf. on Pervasive Computing, 2011.
11. [MOBILITY CHARACTERISTICS] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. Science, 327, 2010.
12. [ONTHEMAP] Wu, J.S. and Graham, M.R. (2008) “OnTheMap: An Innovative Mapping and Reporting Tool.” U.S. Census Bureau.
13. [DIFFERENTIAL PRIVACY] C. Dwork. Differential Privacy. ICALP, 2006.