Algorithm Contest Webinar — Sprint 3
April 5, 2021
Gary Howarth (NIST), Christine Task (Knexus Research), Greg Lipstein (DrivenData)
Agenda

❖ Background
❖ Challenge overview
❖ How to participate
❖ Q&A
Public Safety Communications Research Division of the National Institute of Standards and Technology is the primary federal laboratory conducting research, development, testing, and evaluation for public safety communications technologies.
5 Key Research Areas

User Interface User Experience
Location-Based Services
Mission Critical Voice
Public Safety Analytics
Security
Resilient Systems
Cross Cutting Research Areas
Why the Challenge?

- The Public Safety Communications Research Division (PSCR) of the National Institute of Standards and Technology (NIST) is sponsoring this exciting data science competition to help advance research for public safety communications technologies for America’s First Responders.

- As first responders utilize more advanced communications technology, there are opportunities to use data analytics to gain insights from public safety data, inform decision-making and increase safety.

- But… we must assure data privacy.
What’s the Problem?

Public Safety As Data Generators

- As Public Safety entities make enormous gains in cyber and data infrastructure leading to the routine collection of many large datasets.
- Governments and the public are demanding greater protections on individual privacy and the privacy of individual records.
- Open data initiatives are pushing for the release of more information.

Public Safety Generates Sensitive Information

- Included in the data is personally identifiable information (PII) for police officers, victims, persons of interest, witnesses, suspects, etc.
- Studies have found that a combination of just 3 “quasi-identifiers” (date of birth, 5 digit postal code, and gender) uniquely identifies 87% of the population.
What do we mean by Privacy?

Privacy-preserving data-mining algorithms allow trusted data-owners to release useful, aggregate information about their data-sets (such as common user behavior patterns) while at the same time protecting individual-level information.

Intuitively, the concept of making large patterns visible while protecting small details makes sense. You just 'blur' things a bit:

http://fryeart1.weebly.com/journals.html

If we refine this idea into a mathematically formal definition, we can create a standard for individual privacy.
Differential Privacy Prescreening--and Pitfalls to Avoid

Get Prescreened!

In order to be eligible for final scoring, participants must have their approach pre-screened as differentially private. To be considered for pre-screening, submit a brief PDF document below including a written explanation of your algorithm, any additional data sources used other than the provided data set(s), and a clear, correct mathematical proof that the solution satisfies differential privacy. An example pre-screening submission is provided in the competitor pack.

This document will be reviewed and validated by NIST staff or their delegates. Participants will receive “Prescreened” status if their written explanation proves that they have an essentially correct understanding of differential privacy as applied to their submission, or a brief explanation why their algorithm or proof is incorrect.

Watch your Sensitivity!

On the Open Arena the rules are made up and the points don’t matter… especially if you have a perfect score.

Get your proof checked, and get into the Prescreened Arena today!

In privacy, every time you reference the ground truth data in any fashion, you’re incurring a privacy cost. Don’t get into debt! Make sure to keep your accounting straight and pay up with sufficient privacy noise!

A heads up: The maximum trips per individual taxi is 200 in Sprint 3! That means a simple count of the number of trips in a given Community Area in a given shift has a sensitivity of 200. Check out the privacy resources on the drivedata website, and feel free to ask questions on the forum.

Individuals can contribute to multiple time/map segments, which is tricky to keep in mind. Be extra careful when doing parallel composition!
In the Differential Privacy Temporal Map Challenge (DeID2) your task is to **develop algorithms that preserve data utility as much as possible while guaranteeing individual privacy is protected**.

Submissions will be assessed based on

1. their ability to **prove they satisfy differential privacy**; and
2. the **accuracy of output data** as compared with ground truth.


---

**Objective**

Privacy write-ups
Confirmed by subject matter experts

Algorithm submissions
Evaluated by published performance metric
Challenge Timeline

Algorithm Contest

- 1 Oct 2020
- 6 Jan 2020
- 29 Mar 2021
- 16 Jun 2021
- 27 Oct 2021

Sprint 1
(Baltimore 911 Data)

Sprint 2
(ACS Census Data)

Sprint 3
(Chicago Taxi Data)

Open Source Development Contest

Metric Contest

- Metric development
- 4 Feb 2021
- Potential application of new metrics to Algorithm Contest
# Prize Awards

<table>
<thead>
<tr>
<th>Sprint 1</th>
<th>Sprint 2</th>
<th>Sprint 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct - Dec 2020</td>
<td>Jan - Mar 2021</td>
<td>Apr - Jun 2021</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Place</th>
<th>Prize</th>
<th>Place</th>
<th>Prize</th>
<th>Place</th>
<th>Prize</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>$10,000</td>
<td>1st</td>
<td>$15,000</td>
<td>1st</td>
<td>$25,000</td>
</tr>
<tr>
<td>2nd</td>
<td>$7,000</td>
<td>2nd</td>
<td>$10,000</td>
<td>2nd</td>
<td>$20,000</td>
</tr>
<tr>
<td>3rd</td>
<td>$5,000</td>
<td>3rd</td>
<td>$5,000</td>
<td>3rd</td>
<td>$15,000</td>
</tr>
<tr>
<td>4th</td>
<td>$2,000</td>
<td>4th</td>
<td>$3,000</td>
<td>4th</td>
<td>$10,000</td>
</tr>
<tr>
<td>5th</td>
<td>$1,000</td>
<td>5th</td>
<td>$2,000</td>
<td>5th</td>
<td>$5,000</td>
</tr>
</tbody>
</table>

Progressive prize: 4 @ $1,000

Total: $29,000

Total: $39,000

Total: $79,000

Metric Paper Prizes: $29,000

Open Source Development Prizes: $100,000

Total Prize Purse for Differential Privacy Temporal Map Challenge: $276,000
Algorithm Sprint Structure

This contest sprint will proceed in two phases:

● **Development Phase (Mar 29 - May 10, 2021)**
  ○ Competitors use ‘publicly available’ data to evaluate and refine their algorithms without privacy loss.
  ○ Competitors submit privatized data to be scored on public facing Leaderboard.
  ○ Algorithm write-ups and privacy proofs can be submitted to be **pre-screened as differentially private**.
  ○ Prescreened participants can enter the Prescreened Arena, make executable code submissions to the containerized test harness, and qualify for Final Scoring.
  ○ **Progressive prizes** reward best scores by **Apr 26th**, giving precedence to the Prescreened Leaderboard.

● **Final Scoring Phase (May 10 - 17, 2021)**
  ○ Participants who have passed pre-screening are invited to submit their **final code** (docker container executables) and **write-ups** (privacy proofs, source code and code guides) for final scoring.
  ○ SME panel performs **Final DP Validation** on proofs and source code. Competitors may be asked to fix minor errors and resubmit.
  ○ **Final Scoring** occurs on multiple ‘private’ data sets with the same schema as the ‘publicly available’ data, at multiple epsilon values.
  ○ 1st through 5th Place prizes are based on final scoring.
There are 16 total columns in the ground truth data, including Community Areas, Shift, 12 additional features, and an ID denoting individual taxis. More information is available on the challenge website and in the parameters.json.

- **taxi_id** (int) — Simulated individual ID for each taxi, derived from taxi-weeks in the original data.
- **shift** (int) — Computed variable combining the above two variables into 21 eight-hour segments. Each day has three shifts: "night" (20:00-4:00), "morning" (4:00-12:00), and "afternoon" (12:00-20:00). A "night" shift is contiguous; that is, it includes the last four hours of one day and the first four hours of the following day (e.g. Friday night includes late Friday night and early Saturday morning).
- **company_id** (int) — Company that the taxi works for.
- **pickup_community_area** (int) — Community Area where trip started.
- **dropoff_community_area** (int) — Community Area where trip ended.
- **payment_type** (int) — Type of payment used.
- **trip_day_of_week** (int) — Day of the week that the trip occurred (0 is Monday).
- **trip_hour_of_day** (int) — Hour of the day that the trip occurred on a 24-hour clock.
- **fare** (int) — Fare paid to nearest dollar.
- **tips** (int) — Tips paid to nearest dollar.
- **trip_total** (int) — Total amount paid for the trip.
- **trip_seconds** (int) — Total duration of the trip to the nearest second.
- **trip_miles** (int) — Total length of the trip to the nearest mile.
About the Sprint 3 Data: Community Areas

Chicago has 77 community areas.

And, in our data, a mysterious community area -1, which covers missing data and trip origins/destinations outside Chicago.

Here’s a map of Chicago. If you didn’t learn your way around the city when synthesizing the IL data last sprint, now might be a good time for a virtual tour (wikipedia is your friend).

As always in this challenge, different areas will have different populations, habits, incomes, workplaces, etc, and they’ll show up differently in the data.
You’ll notice something different about how we’re handling the temporal aspect in this sprint. Instead of breaking time into months or years, we’re now looking at days and hours. Let’s review the definition for the “shift” variable in our data-set.
You’ll notice something different about how we’re handling the temporal aspect in this sprint. Instead of breaking time into months or years, we’re now looking at days and hours. Let’s review the definition for the “shift” variable in our data-set.

We begin by splitting the 24hr day into three segments:

-- Morning:  4am - 12pm
-- Afternoon: 12pm - 8pm
-- Night:     8pm - 4am

We then check each segment for every day of the week, for a total of $3 \times 7 = 21$ shifts:

<table>
<thead>
<tr>
<th>Monday Morning</th>
<th>Tuesday Morning</th>
<th>Wednesday Morning</th>
<th>...</th>
<th>Saturday Morning</th>
<th>Sunday Morning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday Afternoon</td>
<td>Tuesday Afternoon</td>
<td>Wednesday Afternoon...</td>
<td>...</td>
<td>Saturday Afternoon</td>
<td>Sunday Afternoon</td>
</tr>
<tr>
<td>Monday Night</td>
<td>Tuesday Night</td>
<td>Wednesday Night...</td>
<td>...</td>
<td>Saturday Night</td>
<td>Sunday Night</td>
</tr>
</tbody>
</table>

In life, and in the data, there’s some strong periodicity to the weekly grind—that’s what we’re studying in this sprint. **What is Chicago like on a typical Monday morning?**

A Friday night?

Who’s going where (and how much are they tipping) on Wednesday at rush hour?
You’ll notice something different about how we’re handling the temporal aspect in this sprint. Instead of breaking time into months or years, we’re now looking at days and hours. Let’s review the definition for the “shift” variable in our data-set.

We begin by splitting the 24hr day into three segments:

-- Morning: 4am - 12pm
-- Afternoon: 12pm - 8pm
-- Night: 8pm - 4am

We then check each segment for every day of the week, for a total of 3 x 7 = 21 shifts:

<table>
<thead>
<tr>
<th>Monday Morning</th>
<th>Tuesday Morning</th>
<th>Wednesday Morning</th>
<th>…</th>
<th>…</th>
<th>Saturday Morning</th>
<th>Sunday Morning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday Afternoon</td>
<td>Tuesday Afternoon</td>
<td>Wednesday Afternoon</td>
<td>…</td>
<td>…</td>
<td>Saturday Afternoon</td>
<td>Sunday Afternoon</td>
</tr>
<tr>
<td>Monday Night</td>
<td>Tuesday Night</td>
<td>Wednesday Night</td>
<td>…</td>
<td>…</td>
<td>Saturday Night</td>
<td>Sunday Night</td>
</tr>
</tbody>
</table>

In life, and in the data, there’s some strong periodicity to the weekly grind—that’s what we’re studying in this sprint.

**What is Chicago like on a typical Monday morning?**

A Friday night?
Who’s going where (and how much are they tipping) on Wednesday at rush hour?

We’ve taken a year of data and broken it into 52 weeks (so each ground truth taxi driver from the original data is split into 52 taxi_id individuals in our data set, and we’re effectively providing those original drivers with privacy on one week level). Here’s an example of an aggregated “activity vector” for one of our taxi_ids. This driver doesn’t work weekends, and gets most of their business in the evenings.
There’s a lot of information in this data set, and we’d like your synthetic data to preserve as much of it as possible.

That means we want to know what the community areas are like in each time segment:
About the Sprint 3 Data: What are you preserving?

There’s a lot of information in this data set, and we’d like your synthetic data to preserve as much of it as possible.

That means we want to know what the community areas are like in each time segment:

We also want to know what the transit patterns are like (correlations across map segments):
There’s a lot of information in this data set, and we’d like your synthetic data to preserve as much of it as possible.

That means we want to know what the community areas are like in each time segment:

We also want to know what the transit patterns are like (correlations across map segments):

...And we also want to know what the *individual taxi drivers* are like:
There’s a lot of information in this data set, and we’d like your synthetic data to preserve as much of it as possible.

In the first sprint we asked for simple aggregation of temporal map data.

And in the second sprint we asked you to manage that with a much more complex feature set.

In this third sprint, we’ll finally be giving you real world individual data (no more simulated linkages!)

And we’ll be judging whether you’ve created complete synthetic individuals that are realistic across all time segments.

To check this, we’ll be using the Higher Order Conjunction metric from the previous NIST Synthetic Data challenge, more on that later.
As compared to Sprint #2, there is a much higher maximum number of records per individual: 200.

The output will be Synthetic Data, including values for taxi_id that represent realistic individuals.

Queries on Taxis will be more manageable than queries on trips

However we’ve simplified epsilon values to just: 10 and 1

**Trip Records:**
taxi_ID, shift, pick-up community, drop-off community, miles, tips.....

**Evaluation Space:**
- Trip Record Data, Aggregated in Communities by Shifts (tips, miles, etc)
- K-Marginal
- Pick-up / Drop-off Patterns between map segments.
- K-Marginal
- Distribution of Individual Taxi Activity Vectors
- HoC
As compared to Sprint #2, there is a much higher maximum number of records per individual: 200.

The output will be Synthetic Data, including values for taxi_id that represent realistic individuals.

Queries on Taxis will be more manageable than queries on trips

However we’ve simplified epsilon values to just: 10, 1

Examples of queries:

“How many trips happened on monday?”
Histogram Counting Trips (sensitivity 200)

“How many taxis took at least 10 trips on monday?”
Histogram Counting Taxis (sensitivity 1)

Evaluation Space:

- Trip Record Data, Aggregated in Communities by Shifts (tips, miles, etc)
- K-Marginal

- Pick-up / Drop-off Patterns between map segments.
- K-Marginal

- Distribution of Individual Taxi Activity Vectors
- HoC
About the Problem: Taxis Not Trips!

What does our data look like?

It’s handy to remember that the taxi_Id variable represents one week of a real taxi-driver’s work.

Each original ground truth driver contributed 52 weeks, and we piled them all together to make this data set.

So you’ll see some interesting similarity patterns between taxi_Ids: Drivers that only work evenings or never work mondays will have similar data.

And some weeks contain St. Patrick’s Day

Evaluation Space:

- Trip Record Data, Aggregated in Communities by Shifts (tips, miles, etc)
- K-Marginal

- Pick-up / Drop-off Patterns between map segments.
- K-Marginal

- Distribution of Individual Taxi Activity Vectors
- HoC
About the Problem: Taxis Not Trips!

What does our data look like?

Evaluation Space:

- Trip Record Data, Aggregated in Communities by Shifts (tips, miles, etc)
- K-Marginal
- Pick-up / Drop-off Patterns between map segments.
- K-Marginal
- Distribution of Individual Taxi Activity Vectors
- HoC

It's also a good idea to understand the map of Chicago.

It's a very diverse city with a long history. And many well-defined communities that have different transit patterns.

You'll see that (unlike uber or Lyft drivers) many taxi_iD generally have a home-base region they work in.

And also... some community areas contain O'Hare.
About Sprint 3 Scoring: K-Marginal Scoring

**Baseline 3-Marginal Score:**

The k-marginal evaluation tool is a randomized heuristic that measures similarity between two high dimensional data sets, by considering all correlations of k or fewer variables (we often use k = 3). It was developed to score solutions in the NIST DelD1 Synthetic Data Challenge.

**Here's how it works:**

1. Numerical features are grouped into range bins.
2. Select a set of k-marginals to score according to the picking strategy specified in the configuration (for example—choose marginals uniformly random at a sampling rate of 0.1).
3. For each selected k-marginal, compute absolute difference between the two densities
4. Average scores across all selected k-marginals to create a total score.
5. Compute the human-readable NIST score = \((2 - \text{raw_score})/2\) x 1000.

An example dataset with three features selected: B, E, G.

Marginal density on the real and synthetic data for selected features.
About Sprint 3 Scoring: K-Marginal Scoring

**Baseline 3-Marginal Score:**

The k-marginal evaluation tool is a randomized heuristic that measures similarity between two high dimensional data sets, by considering all correlations of k or fewer variables (we often use k = 3). It was developed to score solutions in the NIST DeID1 Synthetic Data Challenge.

**Here’s how it works:**

1. Numerical features are grouped into range bins.
2. Select a set of k-marginals to score according to the picking strategy specified in the configuration (for example—choose marginals uniformly random at a sampling rate of 0.1).
3. For each selected k-marginal, compute absolute difference between the two densities
4. Average scores across all selected k-marginals to create a total score.
5. Compute the human-readable NIST score = ((2 - raw_score)/2) x 1000.

See the DrivenData website and competitor’s pack for more details.
The k-marginal evaluation tool is a randomized heuristic that measures similarity between two high dimensional data sets, by considering all correlations of k or fewer variables (we often use k = 3). It was developed to score solutions in the NIST DeID1 Synthetic Data Challenge.

**Here’s how it works:**

1. Numerical features are grouped into range bins.
2. Select a set of k-marginals to score according to the picking strategy specified in the configuration (for example—choose marginals uniformly random at a sampling rate of 0.1).
3. For each selected k-marginal, compute absolute difference between the two densities
4. Average scores across all selected k-marginals to create a total score.
5. Compute the human-readable NIST score = ((2 - raw_score)/2) x 1000.

See the DrivenData website and competitor’s pack for more details.

Continue across many randomly selected marginals, average results.

NIST Score: 982
About Sprint 3 Scoring: HOC Scoring

Like k-marginal, the Higher Order Conjunction (HOC) metric was developed for the first NIST Differential Privacy Synthetic Data Challenge (DeID1). Where the k-marginal metric tests for distributional similarity across a few columns at a time, the HOC metric checks for similarity across full records.

In order to evaluate taxis, the HOC metric will operate over an aggregation of taxi activity data by assigning each taxi a 99-dimension vector counting up the number of trips it takes from each pickup_community_area (78 buckets) and each shift (21 buckets).
Where a k-marginal test begins by selecting k columns in the data, an iteration of the HOC metric begins by randomly selecting one target record (taxi activity vector) in the ground truth data. This represents the taxi 'type' we’re testing. It then generates a randomized similarity constraint for this test, by generating an array of 99 uniformly random integers (in the range [5,50] for this sprint). Using this similarity constraint array, a taxi is defined to be 'similar' to the target taxi if, for each of the 99 columns, the absolute difference between the taxi's activity vector and the target taxi's activity vector is less than similarity constraint value for that column.

The percent (pool) of taxis that are 'similar' to the target taxi type in both the ground truth data and the privatized synthetic data are then computed. The score will take the absolute difference between these densities and the average across all of the tests.
Welcome Aboard to Sprint 3!

- The obstacles you'll run into on this challenge—sparse histograms, sparse map segments, longitudinal privacy, positive bias, complex and heterogeneous map segments.

- And the data properties you may leverage to help address them—similarity of features, trends across time, similarity of individuals, and many others…

- Are problem features that show up in critical real world differential privacy applications.

- You can check out the Sprint #1 & Sprint #2 winners’ announcement to see some of the creativity and intuition that’s been developed so far on these problems. And you can look at the NIST Privacy Engineering Collaboration Space to see some of the solutions that were successful in the NIST Differential Privacy Synthetic Data Challenge.

- You’ve handled higher sensitivity and larger feature sets in past sprints. Let’s tackle the final part of this problem and produce realistic temporal/geographic synthetic individuals in Sprint 3#!
What are you doing with your summer vacation? Would it be improved by earning $10K?
A Brief Note on Your Summer Vacation

What are you doing with your summer vacation? Would it be improved by earning $10K?

We are inviting anyone who has participated in any sprint of the challenge and had their solutions pass differential privacy validation, to release their code as open source and also to put a bit of time in this summer cleaning up their code and making it suitable for use beyond the challenge. We know the hectic pace of the challenge doesn’t always leave a lot of space for good software engineering practices. We'll be rewarding:

- $4K for open source releases (due July 5th)
- $1K for software development plans at (due July 5th)
- Up to $5K for fully executed plans and production software (due Oct 8th)

And we may be helping you partner with real world public safety data analysts to help ensure your systems suit their needs.

See your hard work pay off, and your solutions grow and thrive to help others well beyond the challenge. We’ll be posting to the challenge forum soon with more details.
What are you doing with your summer vacation? Would it be improved by earning $10K?

The development plan must be no more than 4 pages (as a PDF document) and should contain:

- A set of **project goals**. Each goal should be: specific, measurable, attainable, relevant, and accomplishable within 90 days,
- **Milestones with dates** describing the progression of accomplishing the goals that the team intends to achieve within the 90 days,
- A description of how their developed code will **benefit the public safety community** after development. For example, will it be developed into a commercial product? Will it be deposited in an open source repository?, and
- A brief proposal to **de-identify a dataset relevant to public safety**. Teams may identify an existing dataset or they may identify a public safety agency with whom they intend to collaborate.

Development plans will be evaluated on their appropriate scope and planning, adherence to established standards, improvements to code utility, public safety benefit.
Important Dates

Sprint 3 starts  
Mar 29, 2021

Deadline for pre-screening submissions to be reviewed before Progressive Prizes  
Apr 20, 2021

Progressive Prize rankings determined  
Apr 26, 2021

Development phase closes  
May 10, 2021

Final code submissions and write-ups due  
May 17, 2021

Winners announced  
Jun 16, 2021
A **competitor pack** is provided to help participants get started! The contents include:

<table>
<thead>
<tr>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Publicly available” <strong>taxi records data</strong></td>
</tr>
<tr>
<td><strong>Parameters</strong> of the data, including schema and configuration inputs</td>
</tr>
<tr>
<td>Naive <strong>baseline solution</strong></td>
</tr>
<tr>
<td>Sample <strong>privacy write-up</strong></td>
</tr>
<tr>
<td><strong>Scoring metric</strong> implementation for local testing</td>
</tr>
</tbody>
</table>
Note on Eligibility

❖ The challenge is open to international participants, however to be eligible to receive a prize the Team Lead must be age 18 or older and a US citizen or permanent resident of the US or its territories.

❖ The Team Lead is the sole person who will accept the cash prizes on behalf of the team.

❖ Teams are welcome and encouraged to include participants from around the globe. The forum can be a great way to find teammates.

❖ Check out the official rules on challenge.gov for further information on eligibility.
Let’s Get Going!

Overview

Large data sets containing personally identifiable information (PII) are exceptionally valuable resources for research and policy analysis in a host of fields supporting America’s First Responders such as emergency planning and epidemiology.

https://deid.drivendata.org/
Questions?
Competition Details and Official Rules

Challenge.gov
https://www.challenge.gov/challenge/
differential-privacy-temporal-map-challenge/

DrivenData
https://deid.drivendata.org/

Challenge Questions
gary.howarth@nist.gov OR
Challenge Forum

Thank you!