Fairness-Aware Range Queries for Selecting Unbiased Data

Suraj Shetiya, Ian Swift, Abolfazl Asudeh, Gautam Das

ICDE 2022

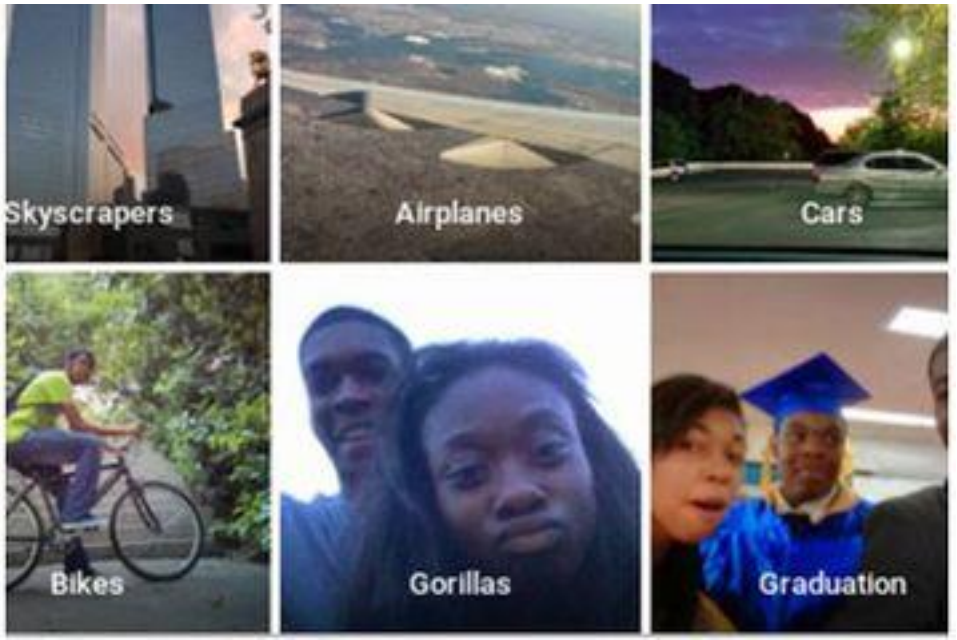
Contents

- Consuming biased data and their consequences
- Fairness-Aware Range Queries
- Algorithms for 1D and general dimensions
- Experimental results

Biased data impact

- Google’s search tags black people as gorilla*

Too few images of black people in training set

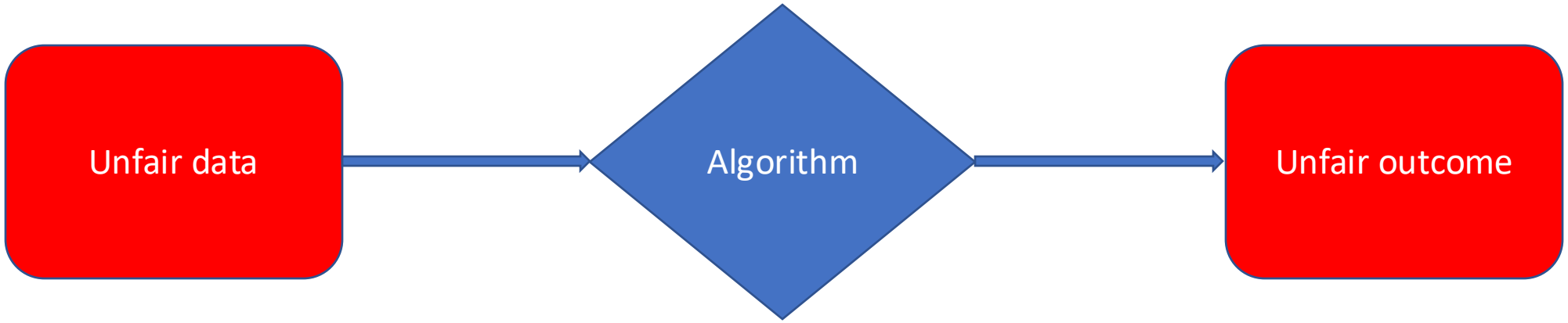


* https://www.wsj.com/articles/BL-DGB-42522

Unfair outcome

- Unfair data leads to unfair outcomes often with grave consequences to the stake holders

An algorithm is only as good as the data it works with



Different fairness criteria

- C_r, C_b – Number of reds and blues in given range-query
- n_r, n_b – Number of reds and blues in given universe
- Ideal distribution of people would have $\frac{C_r}{n_r} = \frac{C_b}{n_b}$
- Our model based on demographic parity

$$|W_r C_r - W_b C_b| \leq \varepsilon$$

Similarity measure

- Given two range queries, similarity between queries is defined by Jaccard similarity on the objects/tuples that belong to the two queries

$$SIM(Q_1, Q_2) = \frac{out(D, Q_1) \bigcap out(D, Q_2)}{out(D, Q_1) \bigcup out(D, Q_2)}$$

Declarative Fairness-Aware Range Queries

- Find most similar range query to given range query, such that output range query is fair.

```
SELECT ... FROM DATABASE
WHERE
     RANGE-PREDICATES
SUBJECT TO
     |W_r C_r - W_b C_b| <= eps and SIM >= tau
```

Unweighted single predicate range query

- Adding or removing an item from a single predicate range query changes the disparity of the range by 1

- Simple observation: The most similar fair range must have a disparity of δ exactly

- One can thus explore only those ranges which have a disparity of δ. As the left/right end point of the range can move, the sum of the disparity covered by the left and right should add up to δ.

Data Structures – Single Predicate

- Cumulative Sum – Helps search the disparity of any given range in `log(n)` time.

- To enable exploring the ranges which have δ disparity, we maintain a data structure which can help us move the end points efficiently.

- Jump Pointer is a data structure that points to the next location in the dataset which has one additional blue (red resp).

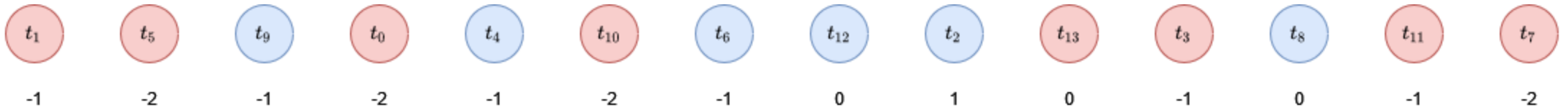
Jump Pointers - Preprocessing

- Create a cumulative sum at each location
- Construct jump pointers using the cumulative sums
- Takes a total O(n log(n)) time

ID	A0	A1
t0	3.1	1.5
t1	0.7	2.3
t2	8	0.65
t3	10.9	1.5
t4	4.4	8.7
t5	1.2	4.1
t6	6.2	6.3
t7	13	5.4
t8	11.3	2.6
t9	2.3	8.4
t10	5.6	4.7
t11	12.7	2.8
t12	7	0.3
t13	9.1	9.4

Cumulative sum

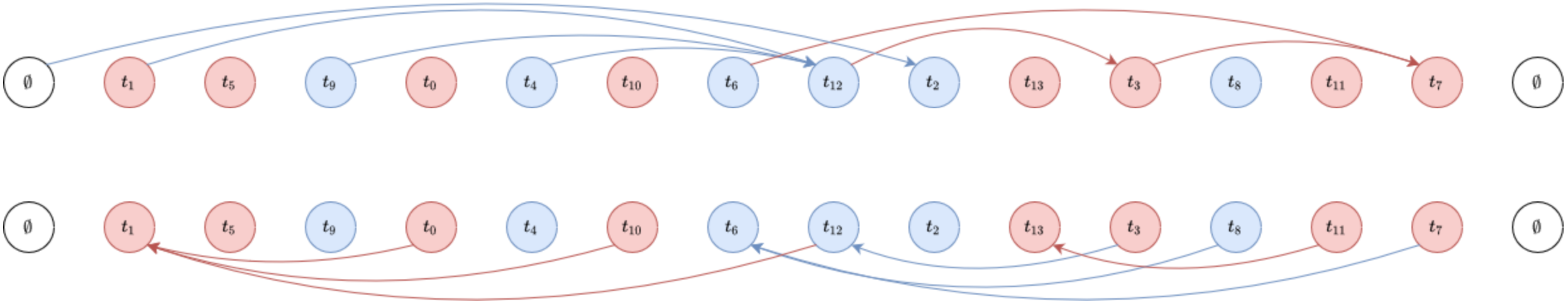
- Sort the elements by the attribute – $\mathtt{A}_0$
- Start from left most location with 0 , blue counts as +1 and red as -1



- When a single predicate query is provided, the end points can be searched in

Jump Pointers

- Cumulative sum is processed to obtain blue and red jump pointers

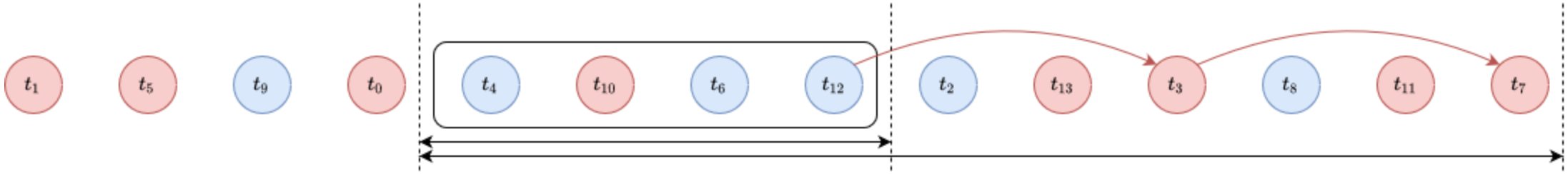


- Jump pointers take a total of `O(n log(n))` time to compute

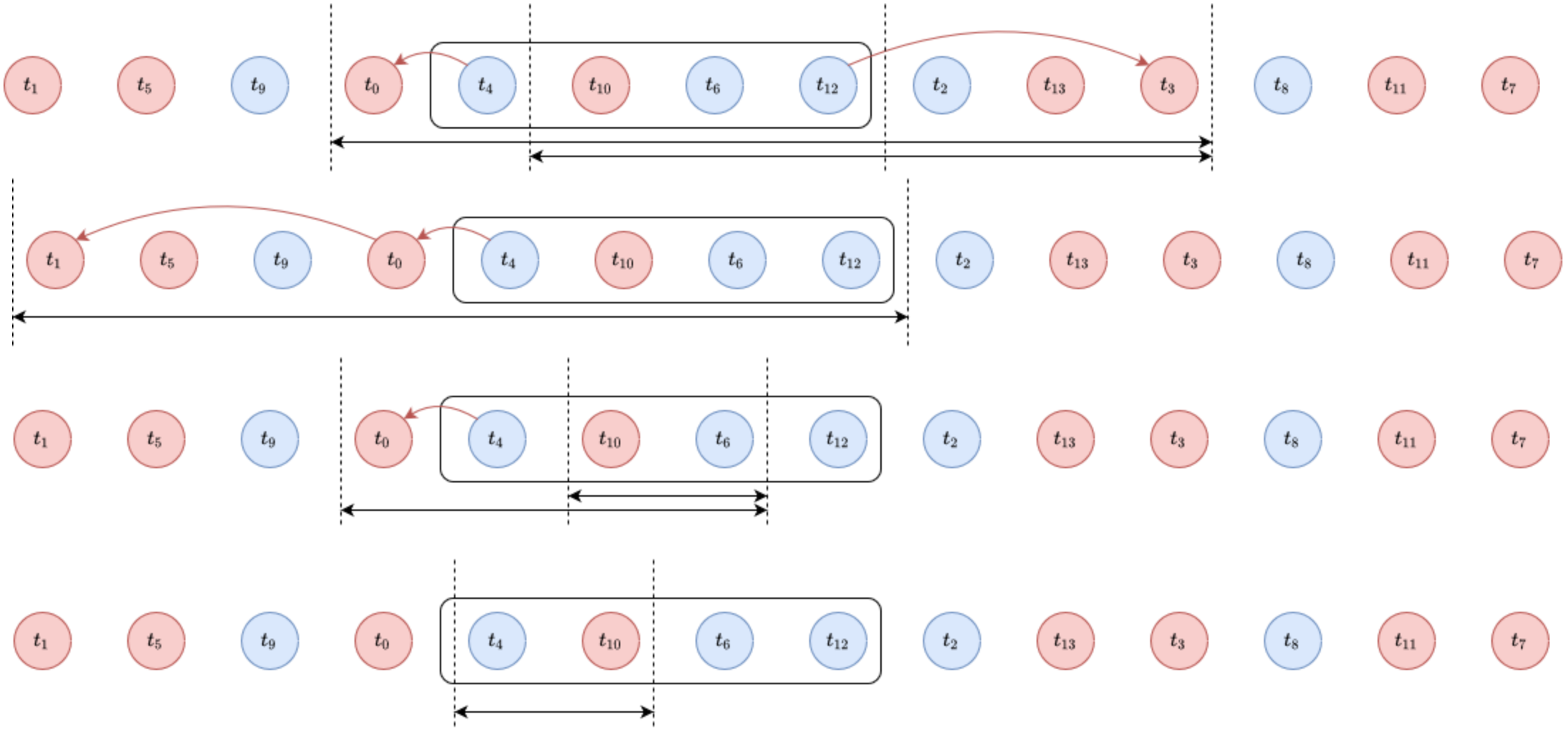
Fair range query example

- Various combinations include – expanding/shrinking from left or right

- For the sample range – [4.4, 7], expanding 2 to the right to find a range with 0 disparity



Fair query – Other windows



Fair range query Complexity

- Preprocessing – $O(n \log(n))$
- Query processing time – $O(\log(n) + disparity)$

Weighted fair range query

- Jump pointers extended to weighted case
- Next pointer points to the location which has a greater/smaller cumulative sum to point to the next blue/red location
- Instead of exact disparity of δ, we check for locations along the pointers which have a disparity less than δ
- Complexity of preprocessing and query processing remain same as unweighted case

Multi predicate range query

- Jump Pointers don’t extend to multi-predicate case
- Neighboring range: Two ranges are called neighboring ranges, if the tuples contained by the two ranges differ by one
- Our Approach: Use a Local Search algorithm near the input range to find the closest fair range

Breadth First Search Approach

- Explore ranges near the input range query, to find the most similar fair range query

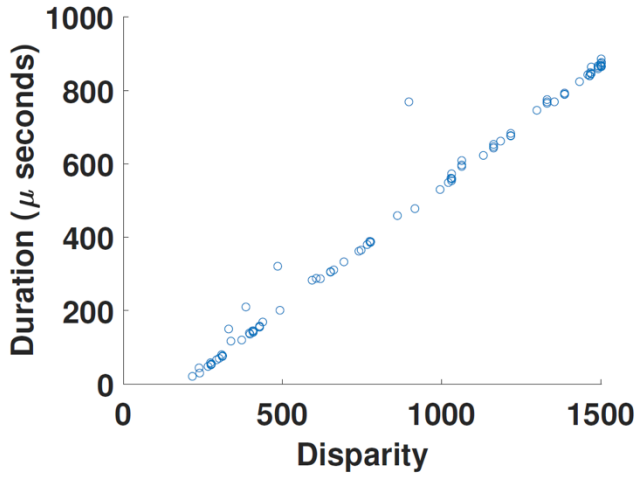
- Number of ranges explored before finding the most similar play a critical role in defining the time taken

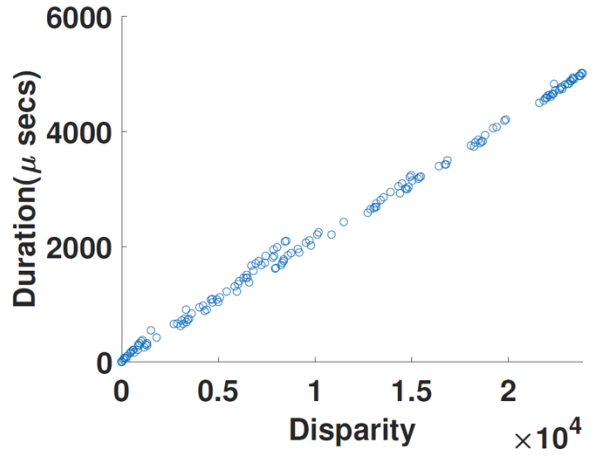
Informed Best First Search

- We define a heuristic to provide an upper bound on the similarity if one of the neighboring range is explored
- Instead of exploring ranges uninformed, A*-based approach to explore ranges based on the heuristic
- Explore those ranges which have more potential to reach optimum before others

Experiments – single predicate

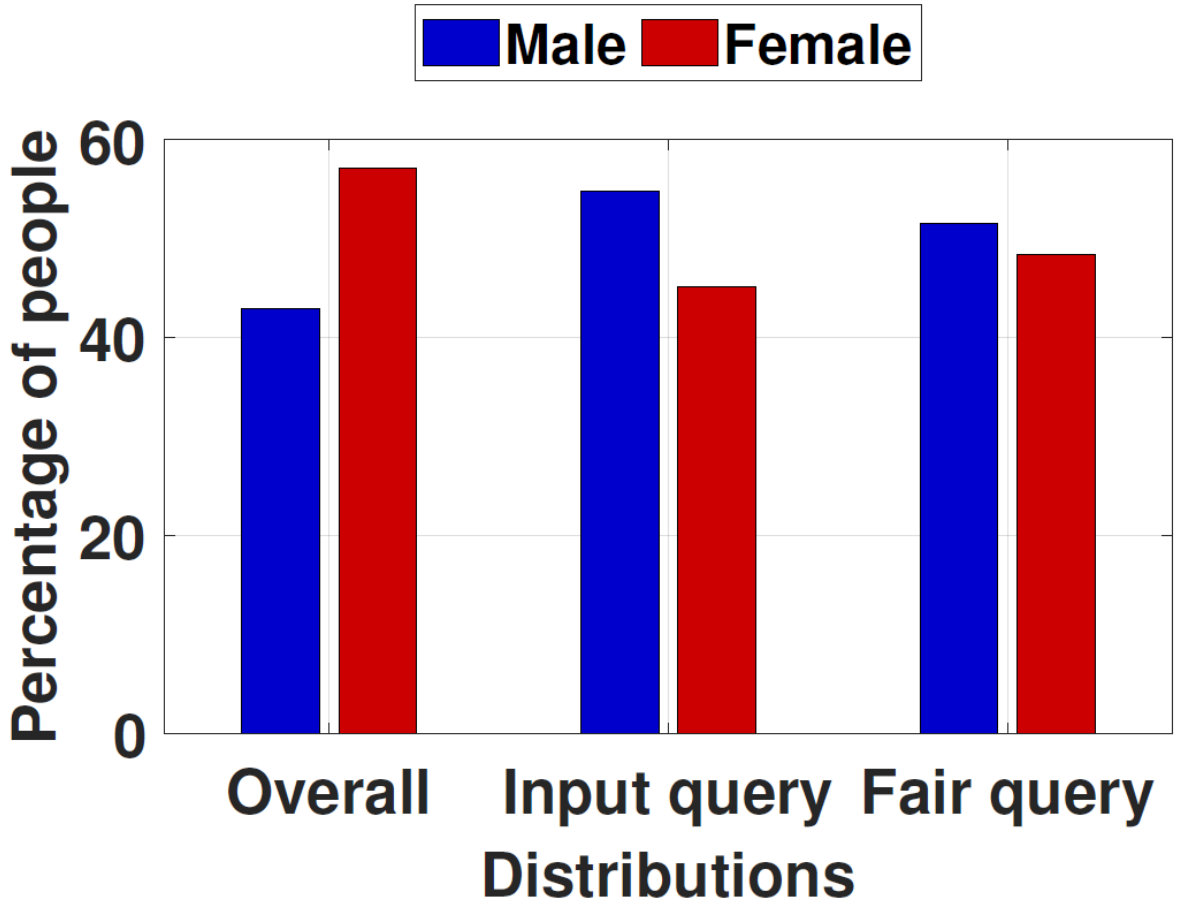
- Single predicate query
- Time taken directly proportional to disparity for both weighted and unweighted



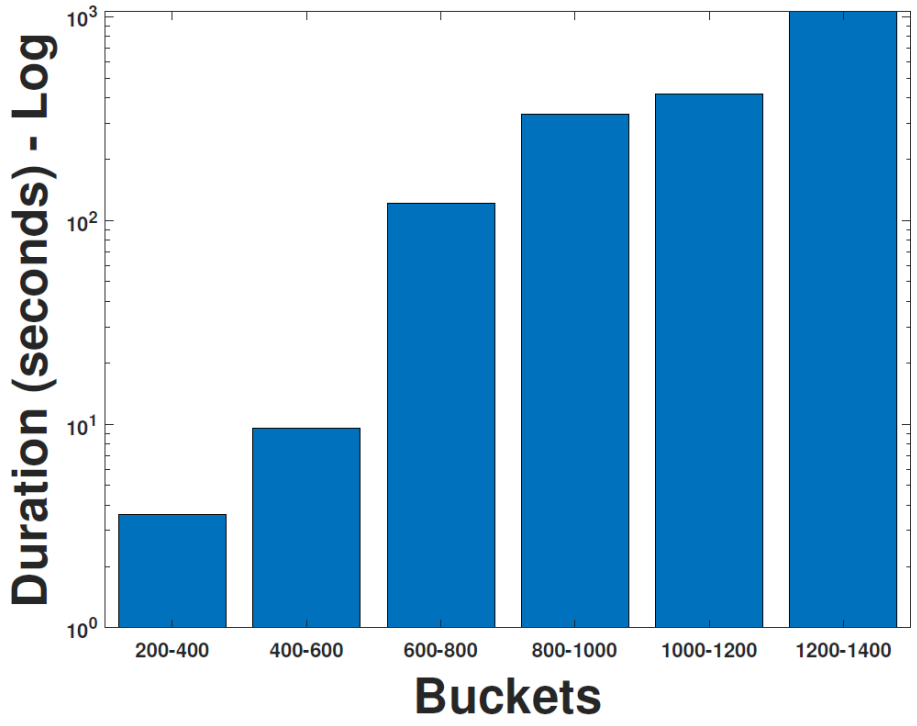


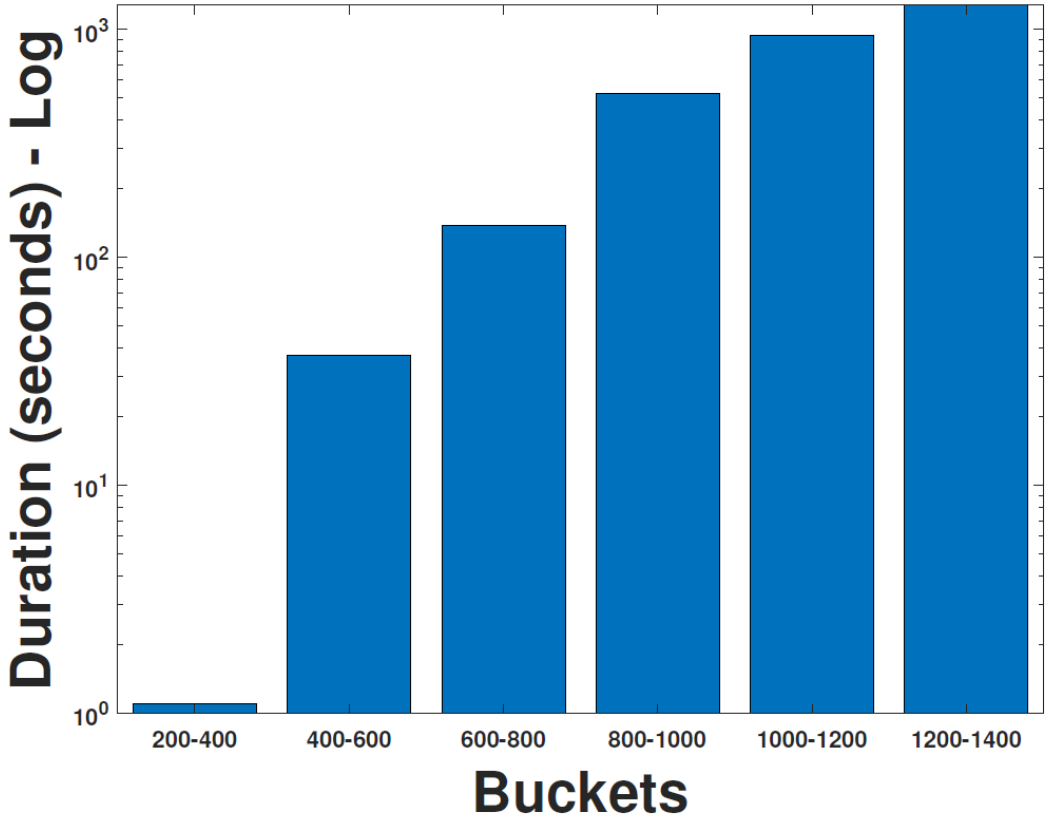
Experiments PoC

- Texas Tribune dataset
- Change in demography in input query, overall population and our query



Experiments multi-predicate





Thank you