Models and Mechanisms for Fairness in Location Data Processing

Sina Shaham
University of Southern California
Los Angeles, California, USA
sshaham@usc.edu

Gabriel Ghinita
Hamad Bin Khalifa University
Doha, Qatar
gghinita@hbku.edu.qa

Cyrus Shahabi
University of Southern California
Los Angeles, California, USA
shahabi@usc.edu

ABSTRACT

Location data use became pervasive in the last decade due to the advent of mobile apps, as well as smart health and smart cities. At the same time, significant concerns have surfaced with respect to fairness in data processing. Individuals from certain population segments may be unfairly treated when being considered for loan or job applications, access to public resources, or other types of services. In the case of location data, fairness is an important concern, given that an individual's whereabouts are often correlated with sensitive attributes, e.g., race, income, education.

While fairness has received significant attention recently, e.g., in machine learning, there is little focus on achieving fairness when dealing with location data. Due to their characteristics and specific type of processing algorithms, location data pose important fairness challenges that must be addressed in a comprehensive and effective manner. In this paper, we adapt existing fairness models to suit the specific properties of location data and spatial processing. We focus on individual fairness, which is more difficult to achieve, and more relevant for most location data processing scenarios. First, we devise a novel building block to achieve fairness in the form of *fair polynomials*. Then, we propose two mechanisms based on fair polynomials that achieve individual fairness, corresponding to two common interaction types based on location data. Extensive experimental results on real data show that the proposed mechanisms achieve individual location fairness without sacrificing utility.

PVLDB Reference Format:
Sina Shaham, Gabriel Ghinita, and Cyrus Shahabi. Models and Mechanisms for Fairness in Location Data Processing. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/SinaShaham/c-Fair-Polynomials/tree/main

1 INTRODUCTION

In the past decade, location data became an integral part of many applications, e.g., mobile apps, smart health, smart cities. Individual locations are often used in creating user profiles (determining the specific content and advertisements that a user is presented with), or as an input for various decision-making processes (like machine learning), which may affect an individual's access to public resources, loans, etc.

While fairness has been extensively studied recently in ML settings [5] for generic data types [15], little attention has been granted to fairness in location data processing. Addressing fairness for location data presents two specific challenges:

(1) Location data may lead, intentionally or inadvertently, to exercise bias against individuals from disadvantaged backgrounds in a *stealth* fashion, due to their high correlation to sensitive features. While it is illegal to use race or ethnicity in a loan-granting or hiring decision, one may use location of current residence as an input attribute. Even though at first glance location may not seem to be sensitive, it may be used to discriminate against people of a certain ethnicity, simply because on many occasions people from the same ethnic group congregate in certain spatially-focused communities [17]. Similar concerns exist for income or education level, which often exhibit strong correlations with the location where an individual works, lives or travels [8]. Note that, this sort of discrimination may often occur inadvertently, as opaque ML algorithms automatically exploit certain correlations in location data (e.g., higher default rates in certain zipcodes), without realizing their fairness implications.

(2) Fairness is achieved through some data transformation designed to prevent, or limit, the amount of bias in processing. This causes loss of utility, whereby the result of processing can be suboptimal compared to the result obtained on the original data. Achieving fairness requires some utility loss, and the emerging fairness-utility trade-off must be carefully considered when devising a fairness mechanism. In the case of location data, utility has specific formulations, which may impact results in a way that is unique to spatial query processing algorithms. Using generic fairness mechanisms devised for other types of data may lead to poor utility, as seen in [25]. Therefore, it is desirable to design customized mechanisms for fighting bias in location data, such that the utility of spatial information is not significantly decreased.

In this paper, we introduce specific mechanisms targeted at providing fairness in location data processing while preserving utility. We focus on the case of *individual* fairness [11], which is more difficult to achieve, but provides a higher level of fairness guarantees compared to its group-level counterpart. We provide specific definitions of location bias, and carefully characterize how location data can be used to exercise discriminatory decisions.

We introduce a novel construction called *fair polynomials* (Section 3.1) that can be used as building block within mechanisms that achieve fairness in the case of location data[1]. We perform a detailed exploration of fair polynomials in order to understand their properties and the trade-off achieved between enforcing fairness and preserving data utility.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

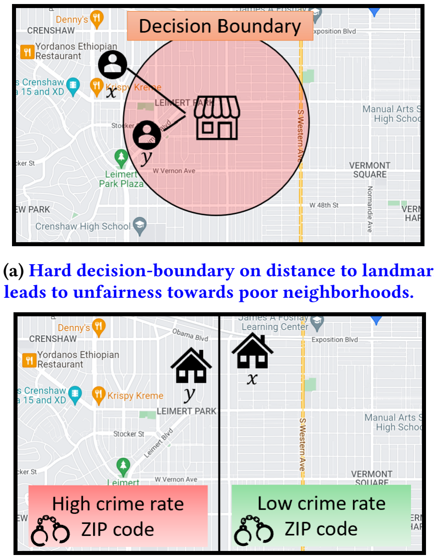
[1] While our focus is on geospatial data, some of our results can be extended for other types of data with continuous domains.

Based on the type of location interaction and query types, we identify two broad categories of scenarios where location bias can occur, and we define protection mechanisms to achieve fairness for each case. Specifically, we consider the following two problems:

- *Location Fairness with respect to Distance to a Reference Point.* In this setting, we investigate how location bias occurs when individuals are characterized with respect to their distance to a reference point. For instance, in the case of location-based advertising, a marketing algorithm may advertise special deals to customers that are nearby a newly-opened health food store. The specific coordinates of the customers may be less relevant for utility, and instead the distance to a landmark is the important factor (i.e., the proximity to the store is a good indicator of likelihood of visiting it). While this may be efficient, it can have fairness repercussions. For instance, if the algorithm chooses the first 100 potential customers to reach based on distance, it is possible that only a rich neighborhood is covered. Customers from a poorer neighborhood that is adjacent to the rich one may never be selected, due to a slightly larger distance threshold. Figure 1a illustrates this case. In this situation, we would like to ensure that individuals from poorer backgrounds also get the chance to benefit from special deals on healthy food, even though they are slightly farther away (i.e., avoid the hard decision boundary phenomenon).
- *Location Fairness with respect to Spatial Coordinates.* In this more general setting, we look at how to ensure fairness of location data processing with respect to actual coordinate values, instead of distances to landmarks. This type of fairness is more broad, in that it can provide fairness with respect to any reference point. Conversely, the amount of data utility sacrificed in the process of achieving fairness may be higher. Figure 1b illustrates this point, where two individual homes are quoted significantly different insurance premiums due to their surrounding characteristics. Ideally, we would like the two residences to have similar premiums, given their close physical proximity.

Our specific contributions are:

- We identify the problem of bias in location data processing, and formalize the notion of location fairness;
- We introduce two definitions of location fairness based on common interaction types; one is customized to a specific type of query and achieves higher utility, whereas the other is more general, at the cost of lower utility;
- We devise a novel concept in the form of *fair polynomials*, which can be used as a building block to obtain mechanisms that achieve fairness for geospatial data;
- We propose two mechanisms based on fair polynomials that enforce location fairness for the two types of studied interactions: namely, with respect to distance to a benchmark, and with respect to spatial coordinates;
- We perform an extensive experimental evaluation on real datasets that shows the effectiveness of the proposed mechanisms, and investigates the fairness-utility trade-offs.



(a) Hard decision-boundary on distance to landmark leads to unfairness towards poor neighborhoods.

(b) Despite close proximity, location x is assigned a much higher insurance premium compared to y.

Figure 1: Examples of two common location bias scenarios.

The rest of the paper is organized as follows: Section 2 formalizes the notions of bias and fairness for location data. Section 3 introduces an individual fairness mechanism with respect to a reference point, whereas Section 4 proposes a mechanism for the more general case of fairness with respect to spatial coordinate values. We survey related work in Section 5. Section 6 presents the results of our experimental evaluation, followed by conclusions in Section 7.

2 SYSTEM MODEL

2.1 Location Bias

The existence of bias, rooted in data or algorithms, is commonly used as a basis for reasoning on unfairness in decision-making. Several sources of bias have been identified in the literature, such as measurement bias [27], behavioral bias [23], etc., many of which are intertwined. In this work, we formalize a type of bias that occurs due to location data. Location bias is formally defined as follows:

Definition 1 (*Location Bias*)**.** Distortion or algorithmic bias generated based on locations of entities in the geospatial domain or their distances to reference points is referred to as *location bias*.

The term *distortion* refers to sources of bias intrinsic to the data. Whereas algorithmic bias [4] refers to a type of bias that is generated purely by processing algorithms. A category of bias closely related to algorithmic bias is called data processing bias [23], and denotes biases introduced in data processing operations such as cleaning, enrichment, and aggregation. Our focus in this work is on location bias sourced in processing algorithms.

Location bias directly or indirectly appears in a variety of applications where distances or locations may cause discrimination against individuals or groups. Consider the running example in Fig. 2 in which a store wants to contact nearby customers with daily offers. A widely known algorithm such as *nearest neighbor* may be used to decide which customers should be contacted. If fairness measures are not considered, and if the store is located in a rich neighborhood, customers who live in less privileged areas of the city may never be contacted, and will be unable to enjoy the special offers.

Such an algorithmic advantage towards a particular group or towards certain individuals is an example of location bias *with respect to distance to a reference point*. The distances can be represented as a one-dimensional input feature data; however, the source of location bias can be multidimensional, more commonly stored in two or more dimensions, e.g., as latitudes and longitudes. Consider classifying districts in a city as "high risk" or "low risk", where more police presence and government resources are allocated to locations with higher crime rates. A strict boundary between a low-risk and high-risk district dictates that individuals who live nearby but placed oppositely across the border are treated differently, despite their close proximity. The unfairness appears in various forms, such as the number of police patrols, the tolerance of police officers to crime, the cost of insurance, and the approval of home improvement loans. To address location bias in classification tasks, i.e., to prevent individuals with similar features from being treated significantly different, i.e., *unfairly*, we first introduce the notion of location fairness in Section 2.2, and then formulate the problem of achieving location fairness in Section 2.3.

2.2 Location Fairness Definition

There are two categories of fairness definitions, namely *group fairness* and *individual fairness* [12]. The former definition addresses the case where a group with certain features is treated statistically different compared to other groups. The latter approach focuses on treating individuals with similar features in a similar way. We adopt the use of individual fairness for location data, since *(i)* it provides higher fairness guarantees; *(ii)* it is more suitable for continuous domain features such as locations, in contrast to categorical features such as education, race, and gender; and *(iii)* location attributes tend to be dynamic, and hence more relevant on an individual basis, rather than for a group. To this end, we formally present the notion of *individual location fairness* in Definition 2 (our definition is adapted from the individual fairness notion proposed in [11]).

Definition 2. (Individual Location Fairness). Let $\mathcal{L} = \{\boldsymbol{l_1}, \boldsymbol{l_2}, ..., \boldsymbol{l_m}\}$ denote the set of individual locations that need to be classified over the output set $\mathcal{A}$, where $\boldsymbol{l_u} \in \mathbb{R}^k$. A randomized mapping $M: \mathcal{L} \longrightarrow \Delta(\mathcal{A})$ satisfies individual location fairness iff for every two locations $\boldsymbol{l_u}, \boldsymbol{l_v} \in \mathcal{L}$ the (D, d)-Lipschitz constraint holds,

$$D(M(\boldsymbol{l_u}), M(\boldsymbol{l_v})) \leq d(\boldsymbol{l_u}, \boldsymbol{l_v}) \tag{1}$$

Intuitively, the definition states that the evaluation process M for two similar locations should yield similar outcomes. The definition relies on two key distance metrics, (1) similarity distance metric $d: V \times V \longrightarrow \mathbb{R}$, measuring how similar individuals are, and (2) a distance metric $D(.)$ measuring the distance between outcome distributions. The former metric will be defined thoroughly for locations in the upcoming sections, as it is tailored to the specific location interaction type. The latter metric, on the other hand, is commonly defined as *total variation norm* or so-called *statistical distance*. Given two probability distributions P and Q over outcome space $\mathcal{A}$, the statistical distance is calculated as

$$D(P, Q) = \frac{1}{2} \sum_{a \in \mathcal{A}} |P(a) - Q(a)|. \tag{2}$$

Table 1: Summary of notations.

Symbol	Description
$\mathcal{L} = \{\boldsymbol{l_1}, ..., \boldsymbol{l_m}\}$	Set of datapoints in $\mathbb{R}^k$
l_i	Distance from $\boldsymbol{l_i}$ to reference point
m	Number of datapoints
$\|\|.\|\|_p$	p-norm distance
$\mathcal{A}$	Classification output domain
$d(.)$	Distance between datapoints
$D(.)$	Distance between distributions
$M(\boldsymbol{l_i})$	Likelihood score of location $\boldsymbol{l_i}$

Our focus in this work is on binary decision-making tasks, hence, the output space is given by $\mathcal{A} = \{0, 1\}$. We assume a classifier modeled as a randomized mechanism $M : L \longrightarrow \Delta(\mathcal{A})$ mapping individuals over outcomes, where $\Delta(\mathcal{A})$ denotes all possible distributions. Thus, the classification of an individual $\boldsymbol{l_u} \in \mathcal{L}$ over outcome space $\mathcal{A}$ is done according to the distribution of $M(\boldsymbol{l_u})$. To simplify notation, we assume function M to return the likelihood of the positive outcome, i.e., $M(\boldsymbol{l_u}) = M(\boldsymbol{l_u}|a = 1)$.

2.3 Problem Formulation

We formulate two problem instances of individual location fairness, specifically: Problem 1 – individual location fairness with respect to *distance to a reference point (DtR)* , and Problem 2 – individual location fairness with respect to *coordinate values*.

Consider m datapoints $\mathcal{L} = \{\boldsymbol{l_1}, \boldsymbol{l_2}, ..., \boldsymbol{l_m}\}$ located in a k-dimensional space ($\mathbb{R}^k$) with $\boldsymbol{R}$ denoting the reference point. Each location represents an individual. Associated attributes of datapoints are stored in a tabular format such that the ith row of the table is dedicated to $\boldsymbol{l_i}$ (Fig. 2 shows tabular information side by side with the map).

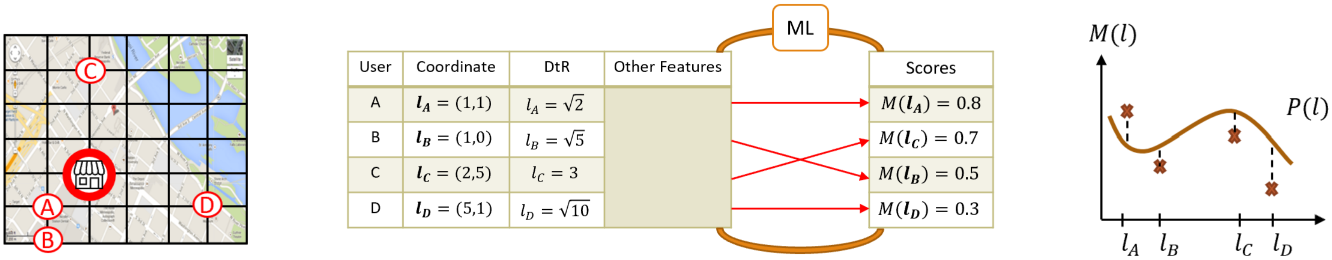
Attribute *Distance to Reference (DtR)* represents the distance of datapoints from the reference point[2]. To keep the discussion applicable to multiple scenarios, we use as DtR metric the *Minkowski* distance of order p (p-norm distance) defined for two datapoints $\boldsymbol{l_u} = (x_1, ..., x_k)$ and $\boldsymbol{l_v} = (x'_1, ..., x'_k)$ as

$$||(\boldsymbol{l_u}, \boldsymbol{l_v})||_p = \sqrt[p]{\sum_{i=1}^{k} (x_i - x'_i)^p}. \tag{3}$$

The u-th entry of the DtR column is associated with datapoint $\boldsymbol{l_u}$ and entails a scalar $l_u = ||(\boldsymbol{l_u}, \boldsymbol{R})||_p / \gamma$, where $\gamma = \underset{i=1...m}{max} ||(\boldsymbol{l_i}, \boldsymbol{R})||_p$ and $\boldsymbol{R}$ denotes the reference point. The constant γ ensures the range of distances is between zero and one ($0 \leq l_i \leq 1, \quad \forall i{=}1...m$). The DtR column is shown with DtR metric set to 2-norm. It is important to note that DtR is based on data representation, and it is not to be confused with the two distance metrics $d(.)$ and $D(.)$, which are crucial elements of individual fairness.

As part of our system model, we assume a *classifier* performing decision-making on top of the data (this model aligns well with current location-based applications, where some machine learning

[2] The proposed approach can be extended to multiple reference points by using a composite cost metric, or an existing locality-sensitive hashing (LSH) approach that produces a single scalar distance value over multiple reference points.



User	Coordinate	DtR	Other Features
A	$\boldsymbol{l_A} = (1,1)$	$l_A = \sqrt{2}$	
B	$\boldsymbol{l_B} = (1,0)$	$l_B = \sqrt{5}$	
C	$\boldsymbol{l_C} = (2,5)$	$l_C = 3$	
D	$\boldsymbol{l_D} = (5,1)$	$l_D = \sqrt{10}$	

Scores
$M(\boldsymbol{l_A}) = 0.8$
$M(\boldsymbol{l_C}) = 0.7$
$M(\boldsymbol{l_B}) = 0.5$
$M(\boldsymbol{l_D}) = 0.3$

(a) Data collection

(b) User locations and classification scores

(c) c-fair polynomials

Figure 2: An example of DtR problem.

is involved in data processing). Given an input individual u, the classifier $M(\boldsymbol{l_u})$ returns a likelihood score for that individual based on her location (e.g., likelihood of receiving a location advertisement). Scores are real values in the range of 0 to 1.

The classifier output scores are shown in the last column of Fig. 2b. For example, user A is located at the coordinate $\boldsymbol{l_A}$ with the calculated distance from the reference point of $\sqrt{1^2 + 1^2} = \sqrt{2}$ normalized to $\sqrt{2}/\sqrt{10}$, and the generated score of $M(\boldsymbol{l_A}) = 0.8$. Other features and attributes used in the model could be race, gender, education, etc. The two problems we seek to address to achieve individual fairness are defined as follows:

Problem 1. (Individual Location Fairness w.r.t. DtR) *For a given location dataset $\mathcal{L}$ with the corresponding DtRs $\{l_1, ..., l_m\}$, and a function $M : \mathcal{L} \longrightarrow [0, 1]$, devise a mechanism to enforce individual fairness (D, d)-Lipschitz constraints with respect to DtRs.*

$$D(M(\boldsymbol{l_i}), M(\boldsymbol{l_j})) \leq d(l_i, l_j) \quad \forall i, j \in [1, ..., m] \tag{4}$$

Problem 2. (Individual Location Fairness w.r.t Coordinates) *For a given location dataset $\mathcal{L} = \{\boldsymbol{l_1}, \boldsymbol{l_2}, ..., \boldsymbol{l_m}\}$, and a function $M : \mathcal{L} \longrightarrow [0, 1]$, devise a mechanism to enforce individual fairness (D, d)-Lipschitz constraints with respect to location coordinates.*

$$D(M(\boldsymbol{l_i}), M(\boldsymbol{l_j})) \leq d(\boldsymbol{l_i}, \boldsymbol{l_j}) \quad \forall i, j \in [1, ..., m] \tag{5}$$

Fairness mechanisms must inherently alter the output likelihood scores in order to achieve the fairness requirement. Hence, there is a cost for such an operation in terms of data utility loss. Since we expect that in numerous cases the output of a fairness mechanism will be used for a learning task, we choose as utility metric *fitting error*, a widely accepted metric in ML for output scores. The utility metric is formally presented in Definition 3.

Definition 3. (Utility). Let $\mathcal{B} : \ M \longrightarrow M'$ be a mechanism that maps every likelihood score M to a likelihood score in M' given that $M, M' : \ \mathcal{L} \longrightarrow \Delta(\mathcal{A})$. The fitting error (utility) of $\mathcal{B}$ is:

$$\sqrt[2]{\frac{1}{m} \sum_{i=1}^{m} (M(\boldsymbol{l_i}) - M'(\boldsymbol{l_i}))^2} \tag{6}$$

As an example, suppose that the output score is mapped to a constant 0.5, i.e., $M'(\boldsymbol{l_i}) = 0.5, \forall i$. The utility loss can then be calculated as $\sqrt[2]{\frac{1}{4}(0.3^2 + 0.2^2 + 0.2^2)} \approx 0.206$.

3 LOCATION FAIRNESS W.R.T. DTR

We propose a location fairness mechanism for Problem 1. Each data point is augmented with a DtR column representing a user's distance from the reference point. Therefore, the most natural similarity distance metric is 1-norm.

$$d(l_i, l_j) = |l_i - l_j| \tag{7}$$

Lemma 3.1 introduces notation and the derivation of statistical distance for the classifier.

Lemma 3.1. *Given the classifier output space of $\mathcal{A} = \{0, 1\}$, statistical distance for every two individuals can be calculated as*

$$D(M(\boldsymbol{l_i}), M(\boldsymbol{l_j})) = |M(l_i) - M(l_j)| \tag{8}$$

Proof.

$$D(M(\boldsymbol{l_i}), M(\boldsymbol{l_j})) = \frac{1}{2} \sum_{a \in \{0,1\}} |P(a) - Q(a)| \tag{9}$$

$$= \frac{1}{2}(|M(\boldsymbol{l_i}) - M(\boldsymbol{l_j})| + |1 - M(\boldsymbol{l_i}) - (1 - M(\boldsymbol{l_j}))|) \tag{10}$$

$$= |M(\boldsymbol{l_i}) - M(\boldsymbol{l_j})| \tag{11}$$

□

3.1 Fair-Polynomials

Despite strong fairness guarantees provided by individual fairness, applying a large number of hard constraints has limited its practicality in real-world scenarios. The most commonly used mechanism in the literature to achieve individual fairness is to come up with an application-specific optimization problem usually referred to as *vendor's utility function* and solve it while imposing individual fairness hard constraints. Unfortunately, two major issues arise with such an approach when applied to location data: (i) In existing approaches, e.g., [25], a constraint optimization problem solver is used to alter input locations such that fairness requirements are met. The number of constraints grows quadratically with the number of data points, which makes their enforcement computationally prohibitive; (ii) The definition of the utility function in most scenarios is not straightforward, confining applicable use cases.

We devise the concept of *fair polynomials*, the intuition behind which is depicted in Figure 2c. A polynomial is fitted to the output scores of the classifier. We construct such polynomials efficiently with a reasonably low *fitting error*. Fair polynomials no longer require enforcement of a large number of hard constraints: given a new data point, its corresponding fair value can be generated by evaluating the polynomial at that point.

Definition 4 (c-Fair Polynomials for DtR)**.** A single variable degree n polynomial $P(x) : \mathbb{R} \rightarrow \mathbb{R}$ is said to be c-Fair if and only if for every two points x and y in its domain

$$|P(x) - P(y)| \leq c|x - y| \tag{12}$$

Given that a fair polynomial providing good estimates of likelihood scores exists, by one-to-one mapping (fitting) of the likelihood scores to polynomials, individual location fairness can be achieved for every two datapoints in the dataset. The constant $c \in [1, +\inf]$ in c-fair polynomials aims to exploit the trade-off between the utility and fairness. On the one hand, when $c = 1$, the optimal individual location fairness is achieved but is usually associated with a higher loss in utility. On the other hand, when the value of c grows larger, the fairness constraint is relaxed, leading to higher utility but lower individual location fairness.

As the DtR problem only involves scalars, our focus is on single variable degree n polynomials. We will extend to multi-variable polynomials to accommodate for multi-dimensional datapoints and address Problem 2 in Section 4.1. In the following, we answer three central questions, (i) what is the sufficient condition for a polynomial to be fair, (ii) how to derive the coefficients of the polynomial by imposing individual fairness constraints, and (iii) how to determine the degree of a fair polynomial.

3.2 Sufficient Condition for Fair Polynomials

There are several families of polynomials that preserve individual fairness over the defined distances for DtR. For example, one such family of polynomials is $P(x) = cx^n/n$ which is proven in Lemma 3.2 to be a c-fair polynomial.

Lemma 3.2. *The polynomial $P(x) = cx^n/n$, is a c-fair polynomial for every two points $x, y \in [-1, 1]$.*

Proof. The proof can be derived by expanding the equation and applying triangle inequality, considering that $|x^i y^j| \leq 1, \forall i, j$.

$$c|\frac{x^n - y^n}{n}| = c|\frac{(x-y)(x^{n-1} + x^{n-1}y + ... + y^n)}{n}| \leq \tag{13}$$

$$\frac{(c|x-y|)(|x^{n-1}| + |x^{n-1}y| + ... + |y^n|)}{n}| \leq c|x-y| \tag{14}$$

□

A fair polynomial must be flexible enough to reduce the error once the likelihood scores are fitted to the polynomial, and not every fair family of polynomials is a viable option. Consider the generic degree n polynomial written as

$$P(x) = a_0 + a_1 x + ... + a_n x^n, \tag{15}$$

where a_i are real numbers. In Theorem 1, we derive a sufficient condition for polynomials of order n to preserve individual fairness.

Theorem 1. *A sufficient condition for a single variable degree n polynomial $P(x) = \sum_{i=0}^{n} a_i x^i$ to be c-fair given that $a_i \in \mathbb{R}$ and $|x| \leq 1$ is to have,*

$$\sum_{i=1}^{n} i|a_i| \leq c \tag{16}$$

Proof. Following the definition of individual location fairness:

$$|P(x) - P(y)| = |\sum_{i=1}^{n} a_i(x^i - y^i)| \leq \sum_{i=1}^{n} |a_i(x^i - y^i)| \tag{17}$$

$$\leq \sum_{i=1}^{n} (|a_i(x-y)|)(|x^{i-1}| + |x^{i-2}y| + ... + |y^i|) \tag{18}$$

$$\leq \sum_{i=1}^{n} |a_i \times i(x-y)| = |x-y| \sum_{i=1}^{n} i|a_i| \tag{19}$$

The above inequality is true based on Jensen's inequality (and also, extended triangle inequality) as well as applying the result from Lemma 3.2. Given that the inequality in Eq. (16) is satisfied, the polynomial is proven to be c-fair based on the definition. □

The theorem indicates that if likelihood scores generated by the model are fitted to a polynomial for which the coefficients are selected such that $\sum_{i=1}^{n} i|a_i| \leq c$, then c-fairness is guaranteed for data entries. The sufficient condition in Theorem 1 can be used directly to learn c-fair polynomials, but the non-linearity existing in the constraint can result in higher computation complexity, as coefficients are unbounded. Theorem 2 addresses this problem by deriving linear constraints over coefficients.

Theorem 2. *A sufficient condition for a 1-variable n-th degree polynomial $P(x) = \sum_{i=1}^{n} a_i x^i$ to be c-fair is to have:*

$$|a_i| \leq \frac{6 \times i \times c}{n(n+1)(2n+1)} \quad \forall\ i \in 1...n \quad (a_i \in \mathbb{R}) \tag{20}$$

Proof. The bound on each a_i value must allow for the maximum degree of freedom while fitting the likelihood scores. Therefore, the condition can be written as an optimization problem.

$$\begin{aligned} &\text{Minimize} && -\sum_{i=1}^{n} a_i \\ &\text{Subject to} && \sum_{i=1}^{n} i|a_i| \leq c \end{aligned} \tag{21}$$

Writing the Lagrangian and applying the stationary condition of Karush–Kuhn–Tucker (KKT) [14],

$$L(a_1, ..., a_n, \lambda) = -\sum_{i=1}^{n} a_i - \lambda(\sum_{i=1}^{n} i|a_i| - c) \tag{22}$$

$$\Rightarrow \frac{\partial L}{\partial a_i} = -1 - \lambda\frac{i}{a_i} = 0 \longrightarrow a_i = -\lambda i \tag{23}$$

λ can be derived from complementary slackness to be

$$\lambda \sum_{i=1}^{n} i^2 = c \longrightarrow \lambda = \frac{6c}{n(n+1)(2n+1)} \tag{24}$$

Therefore, bounds on the coefficients are given as

$$|a_i| \leq \frac{6ic}{n(n+1)(2n+1)} \quad \forall\ i \in 1...n \tag{25}$$

□

3.3 Derivation of Fair Polynomials

We employ a simple ML model to compute polynomial coefficients, where each location distance represents a training sample used to fit the likelihood scores to a polynomial. The training set can be assembled by choosing at random a number of locations from the same data domain as the application (e.g., residential locations within a city). In cases where the target user population is already known (e.g., the coordinates of customers for a store that is using location-based advertising), this user set can be directly used for training. In the following, to simplify notation, we assume the latter. For a given training input l_i, the polynomial output is derived as

$$P(l_i) = a_0 + a_1 l_i + ... + a_n l_i^n, \tag{26}$$

We denote the matrix of all training examples as

$$L = \begin{bmatrix} 1 & l_1 & l_1^2 & ... & l_1^n \\ 1 & l_2 & l_2^2 & ... & l_2^n \\ . & . & . & . & . \\ 1 & l_n & l_n^2 & ... & l_m^n \end{bmatrix}.$$

Recall that m is the number of training examples, and the variables that we learn are the a_is, which define the fair polynomial fitted to the data. The vector of coefficients can be written as

$$\boldsymbol{a}^T = [a_0, a_1, a_2, ..., a_n] \tag{27}$$

and the likelihood scores are vectorized as

$$\boldsymbol{b}^T = [M(l_1), M(l_2), ..., M(l_m)] \tag{28}$$

The convex optimization problem to learn $\boldsymbol{a}$ is formulated as:

$$\begin{aligned} &\text{Minimize} && ||L\boldsymbol{a} - \boldsymbol{b}||_2 \\ &\text{Subject to} && |a_i| \leq \frac{6 \times i \times c}{n(n+1)(2n+1)} \end{aligned} \tag{29}$$

This is equivalent to the least square problem with linear constraints and can be solved efficiently with algorithms such as Trust Region Reflective [10] and Bounded-variable least-squares [26], with complexity linear to the order of the polynomial n. Existing work [25] requires enforcing $O(m^2)$ hard constraints ($m >> n$).

The selection of the polynomial degree n can be conducted based on a trial and error methodology. The optimal degree is the one that results in the minimum variance of error between likelihood scores and their corresponding values on the polynomial. Formally, let e_i denote the error between $M(\boldsymbol{l_i})$ and $P(l_i)$, i.e.,

$$e_i = |M(\boldsymbol{l_i}) - P(l_i)| \tag{30}$$

Then, the value of $n \geq 1$ is selected such that

$$n = argmin(\sum_{i=1}^{m} e_i^2)/(m - n - 1) \tag{31}$$

4 LOCATION FAIRNESS W.R.T. COORDINATES

We extend fair polynomials to preserve individual fairness with respect to data coordinates. The intuition is that the deviation of outcome scores achieved for locations in a classification task should not differ significantly from their distance in space. Revisiting the example in Figure 2, suppose that two users A and B both apply for a home improvement loan, and despite living in close proximity, one is categorized in an underdeveloped area and the other in a developed region due to geographic segmentation. A bank applies a classifier to decide whether an applicant should be granted a loan. The applicant whose home is in the underdeveloped category might be disadvantaged, as the location category can significantly impact the output of the classifier. Individual fairness argues that if two users are located close to each other, their output likelihood scores should not differ significantly.

An advantage of the DtR problem is that a single variable c-fair polynomial can fit output scores due to scalar distances. For multi-dimensional datapoints, Definition 4 is no longer directly applicable. To address this problem, we extend the definition to multivariate polynomials to achieve individual fairness for higher dimensional datapoints. The number of variables involved in fair polynomials is equal to the dimensionality of data points (k).

There are three key variables involved in finding an efficient family of fair polynomials that can fit the output likelihood scores with the minimum loss in utility: *(i)* dimensionality of data k; *(ii)* the distance metric $d(.)$ and *(iii)* fair polynomial degree n. The individual fairness problem can be characterized with respect to these criteria as follows:

- One-dimensional data representation (scalars), 1-norm distance, flexible order polynomial. This corresponds to the DtR scenario considered in the previous section.
- 2-Dimensional data representation, 2-norm distances; order 1 polynomial. This is the most common scenario for locations where the attribute columns include 2D coordinates, and the fairness must be achieved with respect to Euclidean distance between individuals.
- k-Dimensional data representation, 2-norm distances; order 1 polynomial.
- k-Dimensional data representation, p-norm distances; order 1 polynomial.
- k-Dimensional data representation, p-norm distances; flexible order polynomial.

We formulate and derive the sufficiency condition to guarantee individual fairness for each mentioned scenario. The optimization problem in Eq. (29) is formulated with the derived constraints. We omit the vectorization process for conciseness. For several of the proofs used in this section, we make use of Generalized Titu's Lemma provided in Lemma 4.1.

LEMMA 4.1 (GENERALIZED TITU'S LEMMA). *Let m be an integer greater than or equal to 2, a_i^m a non-negative real number, and x_i a positive real number. Then,*

$$n^{m-2} \sum_{i=1}^{n} \frac{a_i^m}{x_i} \geq \frac{(\sum_{i=1}^{n} a_i)^m}{\sum_{i=1}^{n} x_i} \tag{32}$$

PROOF. Please see proof in Appendix B of our extended version[3]. □

4.1 2-norm, 2 dimensional data, Order 1 polynomial

For higher dimensional datapoints, the most common scenario happens when datapoints are in 2D, and the order of the polynomial is one. In practice, datapoints represent coordinates of locations on the map. Consider two locations $\boldsymbol{l_1} = (x_1, x_2)$ and $\boldsymbol{l_2} = (x_1', x_2')$

[3] https://github.com/SinaShaham/c-Fair-Polynomials/tree/main

in $\mathbb{R}^2$, where x_1 and x_1' are the x-axis coordinates, while x_2 and x_2' denote y-axis coordinates. To achieve individual fairness with respect to locations, the hard Lipschitz constraints dictate that:

$$D(M(\boldsymbol{l_i}), M(\boldsymbol{l_j})) \leq d(\boldsymbol{l_i}, \boldsymbol{l_j}) \quad \forall i, j \in 1...m \tag{33}$$

The distance between distribution scores, i.e. $D(.)$, is calculated as before based on Equation (3.1) and the distance between locations is the 2-norm of data points (Euclidean distance), calculated as:

$$d(\boldsymbol{l_i}, \boldsymbol{l_j}) = \sqrt[2]{(x_1 - x_1')^2 + (x_2 - x_2')^2} \tag{34}$$

We start by showing how a fair-polynomial can be derived for the Euclidean similarity distance. Then, we relax the assumptions and generalize the approach for arbitrary distance norms as well as n-dimensional datapoints. Recall that as location data are stored in 2D, the fair polynomial consists of two variables. The generalized definition of fair polynomials for order n polynomials and k dimensional data is provided in Definition 5.

Definition 5 (Generalized c-Fair Polynomial)**.** Polynomial $P(x_1, x_2, ..., x_k) : \mathbb{R}^k \rightarrow \mathbb{R}$ with real coefficients is said to be c-fair if and only if for every two points $x = (x_1, x_2, ..., x_m)$ and $x' = (x_1', x_2', ..., x_m')$ in its domain

$$\begin{aligned} |P(x_1, x_2, ..., x_m) - P(x_1', x_2', ..., x_m')| \leq \\ c \times d(x, x') = c \times ||(x, x')||_p \end{aligned} \tag{35}$$

In the case of 2-dimensional locations and Euclidean distance, fair polynomials imply that for every two locations $\boldsymbol{l_1} = (x_1, x_2)$ and $\boldsymbol{l_2} = (x_1', x_2')$, we must have,

$$|P(x_1, x_2) - P(x_1', x_2')| \leq c \times \sqrt[2]{(x_1 - x_1')^2 + (x_2 - x_2')^2} \tag{36}$$

Where the polynomial is denoted by

$$P(x_1, x_2) = a_0 + a_1 x_1 + a_2 x_2 \tag{37}$$

The goal is to learn the coefficients a_i such that the polynomial $P(.)$ can model the output scores $M(.)$ and preserve fairness with respect to Euclidean distance. Theorem 3 provides the sufficiency condition for a two-variables order one polynomial to be fair.

Theorem 3. *A sufficient condition for a 2-variable first degree polynomial $P(x_1, x_2) = a_0 + a_1 x_1 + a_2 x_2$ to be c-fair defined over 2-norm similarity distance is to have:*

$$|a_1|, |a_2| \leq c/\sqrt{2} \quad (a_1\ a_2 \in \mathbb{R}) \tag{38}$$

Proof. On the one hand, based on Lemma 4.1, a lower bound for Euclidean distances can be written as

$$d(\boldsymbol{l_i}, \boldsymbol{l_j}) = \sqrt[2]{(x_1 - x_1')^2 + (x_2 - x_2')^2} \geq \tag{39}$$

$$\sqrt[2]{(|x_1 - x_1'| + |x_2 - x_2'|)^2 / 2} \tag{40}$$

On the other hand, for the polynomial one can write

$$|P(x_1, x_2) - P(x_1', x_2')| = |a_1(x_1 - x_1') + a_2(x_2 - x_2')| \tag{41}$$

$$\leq |a_1||(x_1 - x_1')| + |a_2||(x_2 - x_2')| \tag{42}$$

By combining the two equations the sufficiency condition in Equation (38) can be derived from the following inequality

$$\begin{aligned} |a_1||(x_1 - x_1')| + |a_2||(x_2 - x_2')| \leq \\ c \times (|x_1 - x_1'| + |x_2 - x_2'|)/\sqrt{2} \end{aligned} \tag{43}$$

□

The above theorem indicates that if the coefficients of polynomials fitted to data are chosen such that $|a_1|, |a_2| \leq c/\sqrt{2}$, the individual fairness is guaranteed for every two locations in the domain. The sufficiency condition for first degree polynomials is generalized for k-dimensional datapoints in space in Theorem 4. Recall that the number of variables in the polynomial is equal to the number of dimensions.

Theorem 4. *A sufficient condition for a k-variable first degree polynomial $P(x_1, ..., x_k) = a_0 + \sum_{i=1}^{k} a_i x_i$ to be c-fair defined over 2-norm similarity distance is to have:*

$$|a_i| \leq c/\sqrt[2]{k}, \quad \forall i = 1...k, \quad (a_i \in \mathbb{R}) \tag{44}$$

Proof. Please see proof in Appendix B of our extended version. □

4.2 p-norm, k dimensional, Order 1 polynomial

We now relax the similarity distance metric for arbitrary p-norm distance, calculated for datapoints $\boldsymbol{l_i} = (x_1, ..., x_k)$ and $\boldsymbol{l_j} = (x_1', ..., x_k')$ as

$$d(\boldsymbol{l_i}, \boldsymbol{l_j}) = \sqrt[p]{\sum_{q=1}^{k} (x_q - x_q')^p} \tag{45}$$

Theorem 5. *A sufficient condition for a k-variable first degree polynomial $P(x_1, ..., x_k) = a_0 + \sum_{i=1}^{k} a_i x_i$ to be c-fair defined over p-norm similarity distance is to have:*

$$|a_i| \leq c/\sqrt[p]{k^{p-1}}, \quad \forall i = 1...k \tag{46}$$

Proof. Based on generalized Titu's Lemma, we have on the one hand a lower bound for Euclidean distances:

$$\begin{aligned} d(\boldsymbol{l_i}, \boldsymbol{l_j}) = \sqrt[p]{\sum_{q=1}^{k} (x_q - x_q')^p} \geq \\ \sqrt[p]{(\sum_{q=1}^{k} |x_q - x_q'|)^p / k^{p-1}} = (\sum_{q=1}^{k} |x_q - x_q'|)/\sqrt[p]{k^{p-1}} \end{aligned} \tag{47}$$

On the other hand, for the polynomial one can write

$$|P(x_1, ..., x_k) - P(x_1', ..., x_k')| = |\sum_{q=1}^{k} a_q (x_q - x_q')| \tag{48}$$

$$\leq \sum_{q=1}^{k} |a_q||(x_q - x_q')| \tag{49}$$

Combining the two, we obtain:

$$\sum_{q=1}^{k} |a_q||(x_q - x_q')| \leq \sum_{q=1}^{k} |x_q - x_q'|/\sqrt[p]{k^{p-1}} \tag{50}$$

The inequality is satisfied when $|a_q| \leq 1/\sqrt[p]{k^{p-1}}$. □

4.3 p-norm, k dimensional, Order n polynomial

So far, the sufficiency condition for c-fair polynomials was derived for arbitrary norms in k-dimensional space based on order 1 polynomials. Moreover, in the DtR problem, c-fair polynomials were developed for 1-dimensional distance using arbitrary degree n polynomial. This subsection provides the theoretical background for the generalized scenario in which the location data are in k dimensions with the norm set to p, and degree n polynomials.

Although by increasing the degree of polynomials, a better fit to likelihood scores can be achieved, the existence of monomials in which multiple variables are involved leads to complexity in the derivation of sufficiency conditions. To address this, we assume that the monomials in the multivariable polynomial consist of only a single variable. Making such an assumption comes with the cost of utility loss; however, it greatly reduces the complexity of the generic case. We assume that the degree n polynomial is expressed as the summation of k univariate polynomials.

$$P(x_1, x_2, ..., x_k) = \sum_{i=1}^{k} P_i(x_i), \tag{51}$$

where $P_i(x_i) = \sum_{j=1}^{n} a_{ij} x_i^j$ is a degree n univariate polynomial with its input being x_i, the ith variable in the original polynomial. The assumption helps to remove existence of monomials with multiple variables such as $x_i^3 x_j^2 x_k^3$ and to simplify derivation of location fairness sufficiency conditions provided in Theorem 6.

Theorem 6. *A sufficient condition for a k-variable n-th degree polynomial $P(x_1, x_2, ..., x_k) = \sum_{i=1}^{k} P_i(x_i)$ to be c-fair defined over p-norm similarity distance is to have:*

$$|a_{ij}| \leq \frac{6 \times j \times c \sqrt[p]{k^{p-1}}}{n(n+1)(2n+1)}, \quad \forall i = 1...k \;\&\; \forall j = 1...n \tag{52}$$

Proof. We write the equation in its component form shown in Equation 51. An upper bound for $P_i(x_i)$ can be derived as,

$$|P(x_1, ..., x_k) - P(x'_1, ..., x'_k)| = |\sum_{i=1}^{k} P_i(x_i) - \sum_{i=1}^{k} P_i(x'_i)| \tag{53}$$

$$= |\sum_{i=1}^{k} (P_i(x_i) - P_i(x'_i))| \leq \sum_{i=1}^{k} |P_i(x_i) - P_i(x'_i)| \tag{54}$$

An upper bound for the sub-terms for all $j = 1...k$ can be derived as

$$|P_i(x_i) - P_i(x'_i)| = |\sum_{j=1}^{n} a_{ij}(x_i^j - x_i'^j)| \leq |x_i - x'_i| \sum_{j=1}^{n} j|a_{ij}| \tag{55}$$

The above inequality is derived based on Eq. (17) setting $|x_i| \leq 1$. We also use the lower bound derived in Eq. (47)

$$c \times d(\boldsymbol{l_u}, \boldsymbol{l_v}) \geq c \sum_{q=1}^{k} |x_q - x'_q| / \sqrt[p]{k^{p-1}} \tag{56}$$

Putting the derived upper bound and lower bound together, the following inequality is satisfied, and c-fairness is guaranteed.

$$|x_i - x'_i|(\sum_{j=1}^{n} j|a_{ij}|) \leq c|x_j - x'_j| / \sqrt[p]{k^{p-1}} \tag{57}$$

$$\sum_{j=1}^{n} j|a_{ij}| \leq c / \sqrt[p]{k^{p-1}} \tag{58}$$

By applying the method used in DtR, the bounds are linearized to,

$$|a_{ij}| \leq \frac{6 \times j \times c}{n(n+1)(2n+1)\sqrt[p]{k^{p-1}}} \quad \forall\, i \in 1...n \;\; (a_i \in \mathbb{R}) \tag{59}$$

□

5 RELATED WORK

Fairness notions can be grouped into two broad categories of Group Fairness and Individual Fairness [11]. In group fairness, a protected attribute of the dataset, such as race or gender, which is considered to be critical in decision-making outcomes, partitions individuals into groups. The ML model used for a decision-making task on the dataset is considered to be fair if it achieves some statistical measure across groups. A few of the key statistical measures include statistical parity [20][11], equalized odds [16], treatment equality [6], and test fairness [9]. Individual fairness aims to give similar predictions to similar outcomes, focusing on fairness for individuals as opposed to groups. Group fairness notions are generally weaker than individual fairness notions [18]. Despite higher fairness guarantees provided by individual fairness and fragility of group fairness notions, group fairness notions are widely studied in the literature due to their easier enforcement [22]. Only a handful of approaches exist in the literature to achieve fairness in the geospatial domain.

The current state-of-the-art approach to enforce individual location fairness is to define a linear loss function once the likelihood scores are generated and solve optimization under individual fairness Lipschitz constraints. Let I be an instance of our problem consisting of a metric $d : \mathcal{L} \times \mathcal{L} \longrightarrow \mathbb{R}$, and a loss function $L : \mathcal{L} \times A \longrightarrow \mathbb{R}$, the optimization problem is defined as,

$$opt(I) = \min_{\{M(x)\}_{x \in L}} \; \underset{x \sim L}{\boldsymbol{E}} \; \underset{a \sim M(x)}{\boldsymbol{E}} \; L(x, a) \tag{60}$$

$$\text{Subject to: } \forall x, y \in: \;\; D(M(x), M(y)) \leq d(x, y) \tag{61}$$

$$\forall x \in L : \;\; M(x) \in \Delta(A) \tag{62}$$

One can see that the number of constraints in this mechanism grows quadratically with the number of individuals, imposing a large computational complexity on the system. The authors in [25] formulate the loss function for location-based advertisements in social media. Locations visited on the map are shown as binary strings, and a classifier is used to predict whether a user should receive a targeted advertisement. Moreover, not directly related to locations, but for general purpose advertisement and auctions, individual fairness is applied in [12]. Another application over which the loss function has been defined is achieving individual fairness in ranking and recommendation systems [24]. In ranking systems, the amount of unfairness with respect to individuals is measured after ranking, and a loss function aims to reorder ranking such that the amount of individual unfairness is minimized [7].

Several attempts have also been made to apply the individual fairness notion for clustering datapoints in Cartesian space. The notion in [19] defines clustering conducted for a point in space as fair if the average distance to the points in its own cluster is not greater than the average distance to the points in any other cluster. The authors in [21] focus on defining individual fairness for k-median and k-means algorithms. Clustering is defined to be individually fair if every point expects to have a cluster center within a particular radius. To the best of our knowledge, no work has directly defined individual fairness with respect to locations.

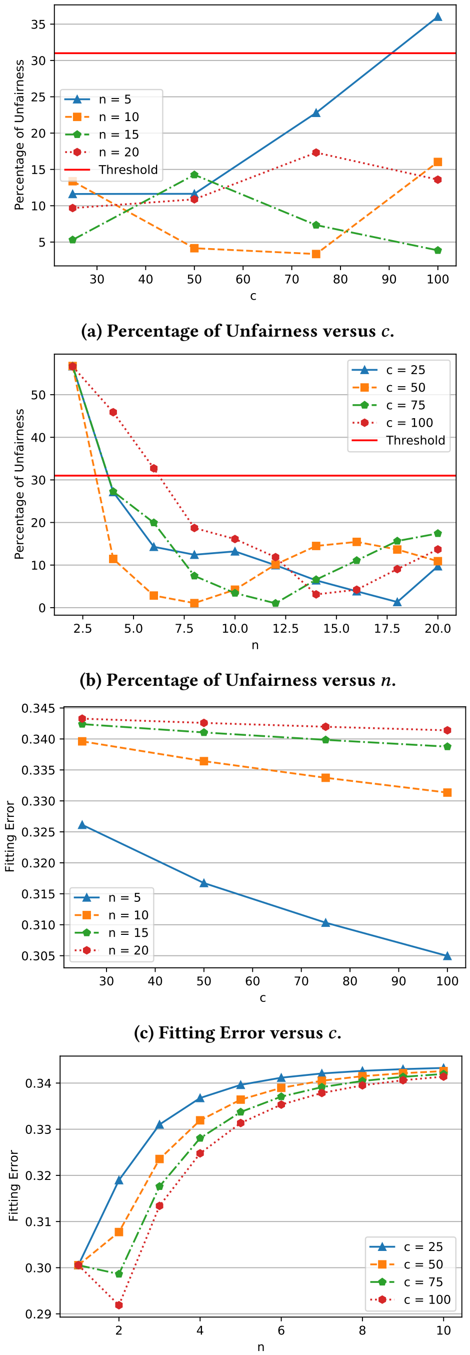
6 EXPERIMENTAL EVALUATION

We evaluate our proposed c-fair polynomials-based fairness mechanisms in the two studied scenarios. For the DtR case, we use a dataset of taxi fares from New York City; for the multi-variate case, we consider budget allocation to police departments according to the Chicago crime occurrence dataset. We ran experiments on a 3.40GHz core-i7 Intel processor with 8GB RAM. The code is implemented in Python and uses the Trust Region Reflective least square implementation from Scipy [3] (the maximum number of iterations for convergence is set to 300, and the default tolerance threshold value of $1e$-2 is used to stop the optimization iterations).

6.1 Fairness w.r.t. DtR

We sampled 120,000 records from the NYC taxi dataset [1] providing over 55 million trips and their associated fares. We deployed an Artificial Neural Network (ANN) to assess the likelihood of taxi fares being fair in the system. Specifically, we seek to capture whether there is bias in the setting of fares based on the specific neighborhoods where the trip starts/ends. Ideally, the trip distance should be the only factor determining price (we carefully pre-process the dataset such that trips are clustered according to the time of week/day, such that differences in fare due to demand status and traffic causes are eliminated). Our goal is to first understand the percentage of records for which the individual fairness constraints do not hold with respect to traveled distances. Then, we analyze the performance of the proposed c-fair mechanism once it is applied to achieve individual location fairness.

ML Model for Fairness Characterization. Our ANN model consists of two hidden layers with 200 and 100 neurons and an output layer with two neurons representing the binary classification task. The activation function used in the model is RELU, the dropout probability for each layer set to 0.4, and cross-entropy is used as the loss function. The accuracy of the model is 92%. The input features include pick-up date and time (categorical hour, AM or PM, weekday, EDT date), pick-up longitude, pick-up latitude, drop-off longitude, drop-off latitude, passenger count, and distance traveled in kilometers. The ride fares have a mean of 10 dollars with a standard deviation of 7 dollars, and the average traveled distance is 3.31 km with a standard deviation of 3.2 km. For model training purposes, we split the data into training, validation and test datasets with 96, 000, 12, 000, and 12, 000 records, respectively. To generate the ground truth for the training dataset, we have used price per kilometer traveled as the indicator of how fair the associated traveled fares are. For every hour of the day, the average price per kilometer is calculated as the hard threshold between fair and unfair travels. The trips above the threshold are classified as unfair, and the values less than the threshold are assumed to be fair. This results in a total of 21, 928 trips being categorized as unfair.



(a) Percentage of Unfairness versus c.

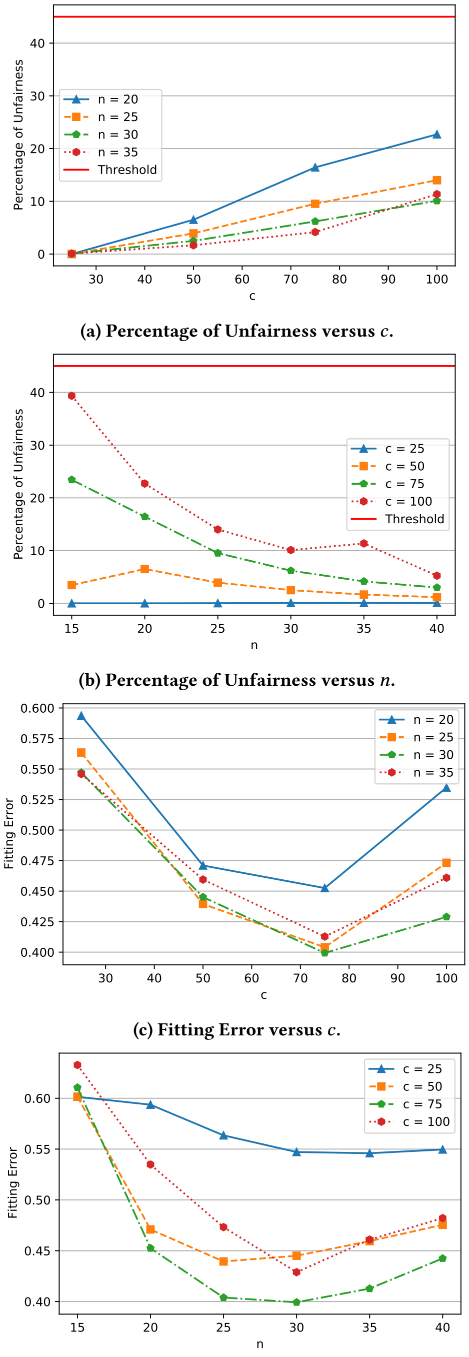
(b) Percentage of Unfairness versus n.

(c) Fitting Error versus c.

(d) Fitting Error versus n.

Figure 3: DtR Mechanism evaluation, New York taxi dataset

Once the ANN model is trained, we predict the likelihood of each trip fare being fair on the test dataset. For every two records, the individual fairness constraint is evaluated to reveal whether fares are fair with respect to travel distances. In the absence of any fairness mechanisms being deployed, 32% of constraints are



(a) Percentage of Unfairness versus c.

(b) Percentage of Unfairness versus n.

(c) Fitting Error versus c.

(d) Fitting Error versus n.

Figure 4: Fairness w.r.t. coordinates values

not satisfied, hence those trips are unfair. The 32% threshold is highlighted with a red horizontal line in the experimental plots, to highlight the fairness improvement of the considered approaches.

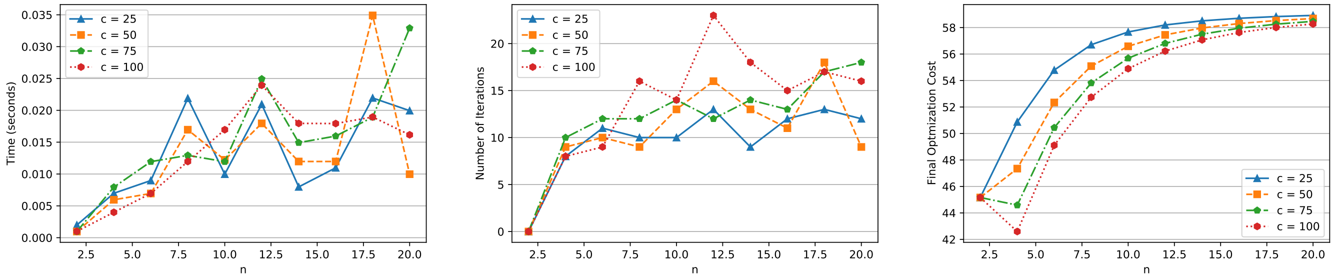
Next, we apply the proposed c-fair mechanism to achieve individual location fairness with respect to DtR. Our experiments evaluate the performance based on four key metrics: percentage of unfairness (constraints were not satisfied), the degree of c-fair polynomial (n), the parameter c, and the root mean square (RMS) of fitting error to likelihood scores.

Percentage of Unfairness. Fig. 3a shows the impact of increasing c on reducing unfairness when the degree of the polynomial is 5, 10, 15, and 20. As expected, lower values of c result in higher fairness in the system, with maximum fairness achieved when c is equal to one. For the maximum fairness scenario, the percentage of unfairness is zero, meaning that all individual fairness constraints are satisfied for every two records in the dataset. By increasing the value of c, fair polynomials would have more room for maneuver and fitting to likelihood scores, but it comes with the cost of higher unfairness. Such behaviour demonstrates the utility-fairness trade-off captured by the constant c. Increasing the polynomial degree can be seen to improve the percentage of unfairness until it reaches the point where it overfits the likelihood scores, and the performance deteriorates. Fig. 3b shows more clearly the impact of increasing the value of n on unfairness. The figure substantiates that lower c values result in a lower percentage of unfairness for all polynomial degrees.

Fitting Error. Figs. 3c and 3d demonstrate the amount of utility loss in data due to fitting likelihood scores to a c-fair polynomial. Two key trends can be observed from the figures. First, increasing the value of c lowers the fitting error. This is expected, as higher c allows more flexibility for selecting coefficients and better fitting performance. Second, increasing the value of n for the same value of c raises the fitting error. To understand this behavior, one can intuitively look at the problem as allocating the same amount of budget among several buckets representing coefficients. Although increasing the degrees of freedom provides better fitting performance as higher degree monomials exist, it further restricts the budget for each coefficient. Thus, the lower degree monomials, which have a more significant impact on the performance, are allocated a lower amount of budget, negatively affecting the performance.

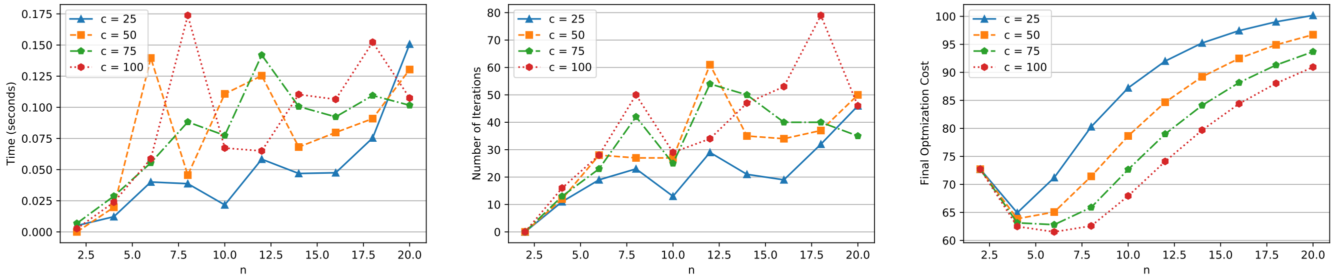
Computational Complexity. We measure the computational overhead of c-fair polynomials in terms of time complexity, number of iterations before optimization convergence, and final optimization cost (the final optimization cost represents the value of the Scipy cost function upon reaching the solution [3]). Fig. 5 shows the results. In each graph, the overhead is shown for four values of $c = 25, 50, 75, 100$ plotted for varying polynomial degrees. Overall, the time complexity is in the order of milliseconds and does not limit the practical deployment of c-fair polynomials. The second graph illustrates the number of iterations before reaching the optimal point. The optimization process stops either by reaching the maximum number of iterations (300) or when the relative change in optimization cost remains below the tolerance threshold (1e-2). As explained previously, the slight oscillation in the performance is due to selecting a random start point for the optimization.

Note that, increasing the degree of polynomial n results in higher computational overhead. This is expected, as more degrees of freedom (coefficients) lead to more effort for finding the optimal point. Another consistent behavior across all three figures is that increasing c on average reduces the computation complexity cost and facilitates reaching the near-optimal point. The trend is more apparent in the final optimization cost figure, in which it can be clearly seen that a higher c value leads to a lower cost.



(a) Time Complexity **(b) Number of Iterations Before Convergence** **(c) Optimization Final Cost**

Figure 5: Computation Overhead Analysis on New York Taxi Dataset.



(a) Time Complexity **(b) Number of Iterations Before Convergence** **(c) Optimization Final Cost**

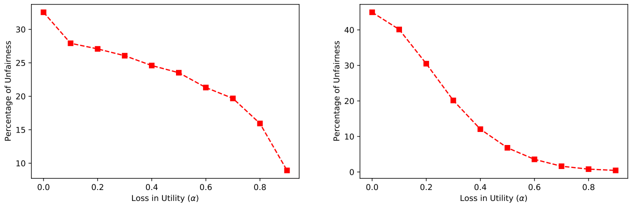
Figure 6: Computation Overhead Analysis on Chicago Crime Dataset.

6.2 Fairness w.r.t. coordinate values

For this scenario, we consider the case of budget allocation to different areas of Chicago, USA, based on the measured crime rates. We use the dataset provided by the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system [2], consisting of reported crime incidents in Chicago. A 1024×1024 grid is overlaid on top of the Chicago map, and the goal is to fairly allocate the budget such that neighborhoods that are close to each other are treated similarly. We have selected seven major crime categories of sexual assault, homicide, kidnapping, sex offense, motor vehicle theft, criminal damage, and narcotics among the reported crimes, and trained a logistic regression model to infer the likelihood of crime occurrence in each cell. The training dataset includes the crime data from January to November 2015, and the December data is chosen as the test dataset. The accuracy of the model is 94% and its output is a set of likelihoods indicating the probability of crime occurrence. The budget allocated to each cell is proportional to the likelihood score derived by the classifier.

Once the likelihood scores are generated, they are used with X and Y cell coordinates to achieve individual location fairness with the distance metric set to 2-norm. In absence of any fairness mechanism, we determine the percentage of individual location fairness constraints not being satisfied at 44.0%.

To understand if the expansion to higher polynomial degrees is essential, we started our experiments by focusing on degree one c-fair polynomials and applying the results in Theorem 3. As expected, the fitting error was rather high, and the utility was insufficient. We also noticed that for degree one polynomials, the optimal solution is achieved even when the value of c is equal to one. Therefore, increasing c does not help with improving the fitting error. Thus, it is crucial to use higher degree polynomials for this purpose.

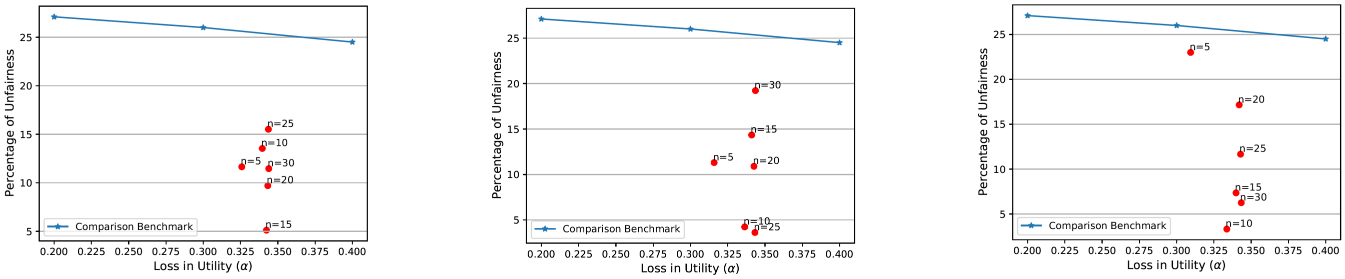


(a) DtR Mechanism evaluation, New York taxi dataset **(b) Fairness w.r.t. coordinates evaluation, Chicago crime dataset**

Figure 7: Comparison Benchmark Performance.

Next, we apply the optimization formulation derived in Theorem 6, the most generalized formula allowing each dimension to contribute in fitting with a degree n polynomial. Fig. 4 demonstrates the performance of c-fair polynomials for achieving individual location fairness on the crime dataset. The patterns are generally consistent with the DtR problem considered in the New York taxi fares experiment. Figs. 4a and 4b show the impact of c and n on the percentage of unfairness and Figs. 4c and 4d show the performance with respect to utility. The red line is used as the reference point representing the percentage of unfairness in the original data, in the absence of fairness mechanisms.

Increasing the value of c results in a lower degree of fairness and higher fitting error once the degree of polynomial reaches an acceptable level. This result further substantiates the fairness-utility trade-off in the system, also observed in the DtR problem. Based on the experiments, in a 2-dimensional space, using a degree of 10 and above polynomial to model each dimension can ensure that scores

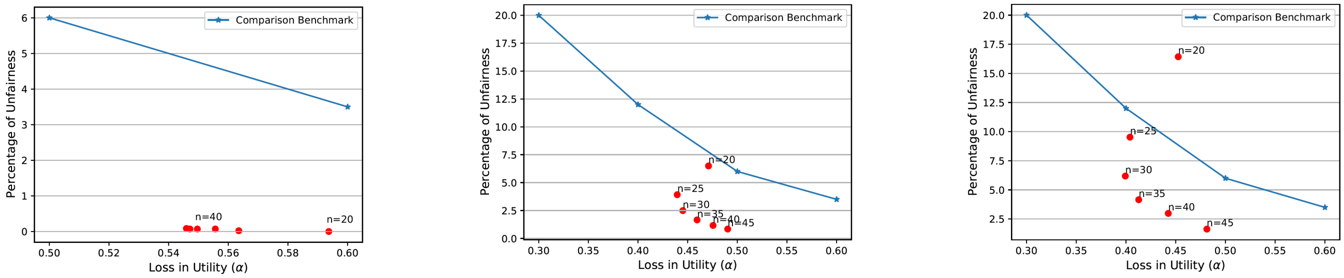


(a) $c = 25$

(b) $c = 50$

(c) $c = 75$

Figure 8: DtR Mechanism Benchmark Comparison, New York taxi dataset



(a) $c = 25$

(b) $c = 50$

(c) $c = 75$

Figure 9: Fairness w.r.t. coordinates Benchmark Comparison, Chicago crime dataset

are under-fitted, preventing high fitting error values. In summary, the amount of fairness achieved with the fair polynomials, even for a reasonably low degree for polynomials such as 15 and values of c greater than 10, can be seen to be over 70%.

Fig. 6 shows the computation complexity of the proposed mechanism. The first point to notice is that a relatively high amount of time is required to achieve individual location fairness w.r.t. coordinates compared to the DtR setting. However, computational complexity is still low in absolute value, and not an obstacle for practical deployment, with sub-second execution time.

6.3 Comparison with benchmarks

We derive a threshold-based benchmark from the fairness mechanism in [11] and the binary evaluation approach in [13]. Given constant threshold t, let polynomial $P(l_i) = t$, and let parameter α define by how much scores can be altered (e.g., if $\alpha = 0.1$, each score can be altered by at most ±0.1). For DtR, the benchmark cycles through each score M_i and pushes it towards polynomial $P(l_i)$ with an allowance of α:

$$M(l_i) \leftarrow M(l_i) + sign(P(l_i) - \alpha) \times \min(\alpha, |P(l_i) - \alpha|) \qquad (63)$$

The *sign* operation ensures the direction of change is favorable to the benchmark, and *min* ensures that the change in score does not overshoot $P(l_i)$. When the value of α is zero, no utility loss exists, and the unfairness percentage is the same as the original. As α grows, the flexibility margin results in more constraints being satisfied until absolute fairness is achieved (all scores equal to $P(l_i)$).

For the 2D fairness case, we use $P(x, y) = (x + y)/\sqrt{2}$, easily extensible to higher dimensions. Both the DtR and multi-dimensional polynomials can be shown easily to preserve fairness.

Figure 7 shows the utility-fairness trade-off obtained by the baseline for both DtR and multi-variate case. Unfairness is monotonically decreasing as one allows a higher α alteration to the score. We notice a plateau of desirable behavior concentrated between $\alpha = [0.2, 0.6]$, where the utility-fairness trade-off is good (for α values below 0.2 unfairness is too high, whereas for values higher than 0.6 the fairness gain fades in the multi-variate case).

Next, we focus our attention to this desirable α range, and we plot the relative performance of the benchmark versus our proposed fair polynomials approach for various values of c and n. The scatter plots in Figs. 8 and 9 show that our approach provides a consistently superior trade-off compared to the benchmark. The only case where the benchmark outperforms is for the multi-variate fairness case on the Chicago dataset, and only for the $n = 20$ setting. In all other cases, fair polynomials provide either a vastly improved utility, or better fairness.

7 CONCLUSION

We studied in depth the problem of individual fairness for location data, and we identified sources of location bias that can occur in practical settings. We formulated two distinct problems that are relevant to location-based applications, and we devised specific techniques to achieve fairness while preserving utility, with the help of a novel construction called fair polynomials. While our focus is on geospatial data and applications, fair polynomials have the potential to provide useful building blocks for fairness in other application domains. In future work, we plan to study more complex types of location-based interaction. At the same time, we plan to study the effect of fairness mechanisms in conjunction with other constraints, such as privacy, e.g., devise mechanisms that can achieve both fairness and privacy.

REFERENCES

[1] [n.d.]. New York City taxi fare prediction. https://www.kaggle.com/c/new-york-city-taxi-fare-prediction/data

[2] 2015. Chicago Crime Dataset. https://data.cityofchicago.org/Public-Safety/Crimes-2015/vwwp-7yr9

[3] 2021. SciPy Documentation Version: 1.7.1 (lsq linear). https://docs.scipy.org/doc/scipy/

[4] Ricardo Baeza-Yates. 2018. Bias on the web. *Commun. ACM* 61, 6 (2018), 54–61.

[5] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2017. Fairness in machine learning. *Nips tutorial* 1 (2017), 2.

[6] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2021. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research* 50, 1 (2021), 3–44.

[7] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *The 41st international acm sigir conference on research & development in information retrieval*. 405–414.

[8] Enrico Cantoni. 2020. A precinct too far: Turnout and voting costs. *American Economic Journal: Applied Economics* 12, 1 (2020), 61–85.

[9] Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5, 2 (2017), 153–163.

[10] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. 2000. *Trust region methods*. SIAM.

[11] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[12] Cynthia Dwork and Christina Ilvento. 2018. Individual fairness under composition. *Proceedings of Fairness, Accountability, Transparency in Machine Learning* (2018).

[13] Alessandro Fabris, Stefano Messina, Gianmaria Silvello, and Gian Antonio Susto. 2022. Algorithmic Fairness Datasets: the Story so Far. *arXiv preprint arXiv:2202.01711* (2022).

[14] Benyamin Ghojogh, Ali Ghodsi, Fakhri Karray, and Mark Crowley. 2021. KKT Conditions, First-Order and Second-Order Optimization, and Distributed Optimization: Tutorial and Survey. *arXiv preprint arXiv:2110.01858* (2021).

[15] Sara Hajian, Francesco Bonchi, and Carlos Castillo. 2016. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2125–2126.

[16] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016), 3315–3323.

[17] Kamyoung Kim. 2020. A Spatial Optimization Approach for Simultaneously Districting Precincts and Locating Polling Places. *ISPRS International Journal of Geo-Information* 9, 5 (2020), 301.

[18] Michael P Kim, Aleksandra Korolova, Guy N Rothblum, and Gal Yona. 2019. Preference-informed fairness. *arXiv preprint arXiv:1904.01793* (2019).

[19] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. 2020. A Notion of Individual Fairness for Clustering. *arXiv preprint arXiv:2006.04960* (2020).

[20] Matt J Kusner, Joshua R Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. *arXiv preprint arXiv:1703.06856* (2017).

[21] Sepideh Mahabadi and Ali Vakilian. 2020. Individual fairness for k-clustering. In *International Conference on Machine Learning*. PMLR, 6586–6596.

[22] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–35.

[23] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kıcıman. 2019. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data* 2 (2019), 13.

[24] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. 2021. Fairness in Rankings and Recommendations: An Overview. *arXiv preprint arXiv:2104.05994* (2021).

[25] Christopher Riederer and Augustin Chaintreau. 2017. The Price of Fairness in Location Based Advertising. (2017).

[26] Philip B Stark and Robert L Parker. 1995. Bounded-variable least-squares: an algorithm and applications. *Computational Statistics* 10 (1995), 129–129.

[27] Harini Suresh and John V Guttag. 2019. A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002* 2 (2019).

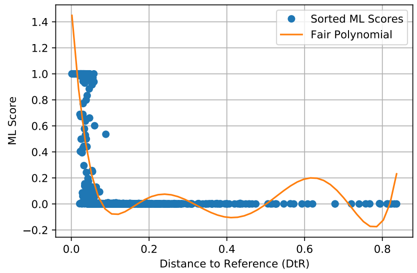


Figure 10: An example of c-fair polynomial applied to the NYC taxi dataset.

A APPENDIX (PROOF OF LEMMAS)

LEMMA A.1 (*GENERALIZED TITU'S LEMMA*). *Let m be an integer greater than or equal to 2, a_i^m a non-negative real number, and x_i a positive real number. Then,*

$$n^{m-2}\sum_{i=1}^{n}\frac{a_i^m}{x_i} \geq \frac{(\sum_{i=1}^{n} a_i)^m}{\sum_{i=1}^{n} x_i} \tag{64}$$

PROOF. By Holder's inequality

$$(\sum_{i=1}^{n} 1)^{\frac{m-2}{m}}(\sum_{i=1}^{n}\frac{a_i^m}{x_i})^{\frac{1}{m}}(\sum_{i=1}^{n} x_i)^{\frac{1}{m}} \geq \tag{65}$$

$$\sum_{i=1}^{n} 1^{\frac{m-2}{m}}(\frac{a_i^m}{x_i})^{\frac{1}{m}} x_i^{\frac{1}{m}} \tag{66}$$

$$\rightarrow n^{\frac{m-2}{m}}(\sum_{i=1}^{n}\frac{a_i^m}{x_i})^{\frac{1}{m}}(\sum_{i=1}^{n} x_i)^{\frac{1}{m}} \geq \sum_{i=1}^{n} a_i \tag{67}$$

$$\rightarrow n^{m-2}\sum_{i=1}^{n}\frac{a_i^m}{x_i} \geq \frac{(\sum_{i=1}^{n} a_i)^m}{\sum_{i=1}^{n} x_i} \tag{68}$$

□

THEOREM 7. *A sufficient condition for a k-variable first degree polynomial $P(x_1, ..., x_k) = a_0 + \sum_{i=1}^{k} a_i x_i$ with real coefficients ($a_i \in \mathbb{R}$) is to have: Euclidean distance n variables.*

$$|a_i| \leq c/\sqrt[2]{k}, \quad \forall i = 1...k \tag{69}$$

PROOF. For every two locations $\boldsymbol{l_1} = (x_1, ..., x_k)$ and $\boldsymbol{l_2} = (x'_1, ..., x'_k)$, a lower bound can be derived based on Generalized Titu's Lemma.

$$d(\boldsymbol{l_1}, \boldsymbol{l_2}) = \sqrt[2]{\sum_{i=1}^{k}(x_i - x'_i)^2} \geq \sqrt[2]{(\sum_{i=1}^{k}|x_i - x'_i|)^2/k} \tag{70}$$

On the other hand, the following inequality can be written for the polynomial.

$$|P(x_1, ..., x_k) - P(x'_1, ..., x'_k)| = |\sum_{i=1}^{k} a_i(x_i - x'_i)| \tag{71}$$

$$\leq \sum_{i=1}^{k}|a_i||(x_i - x'_i)| \tag{72}$$

By combining two equations the sufficiency condition can be derived from,

$$\sum_{i=1}^{k}|a_i||(x_i - x'_i)| \leq c \times (\sum_{i=1}^{k}|x_i - x'_i|)/\sqrt{k}$$

□

B APPENDIX (POLYNOMIAL VISUALIZATION)

Figure 10 presents the real-world example of Fig. 2 for NYC taxi dataset. It can be seen in the figure that a fair polynomial is fitted to output ML scores shown by blue dots. The degree of the polynomial is 6, and the value of c is set to 100.