Identifying Bias in Data Using Two-Distribution Hypothesis Tests

William Yik
wyik@hmc.edu
AMISTAD Lab, Department of Computer Science, Harvey Mudd College
Claremont, California, USA

Limnanthes Serafini*
lserafini@hmc.edu
AMISTAD Lab, Department of Computer Science, Harvey Mudd College
Claremont, California, USA

Timothy Lindsey*
tim.lindsey@biola.edu
Department of Math and Computer Science, Biola University
La Mirada, California, USA

George D. Montañez
gmontanez@hmc.edu
AMISTAD Lab, Department of Computer Science, Harvey Mudd College
Claremont, California, USA

ABSTRACT

As machine learning models become more widely used in important decision-making processes, the need for identifying and mitigating potential sources of bias has increased substantially. Using two-distribution (specified complexity) hypothesis tests, we identify biases in training data with respect to proposed distributions and without the need to train a model, distinguishing our methods from common output-based fairness tests. Furthermore, our methods allow us to return a "closest plausible explanation" for a given dataset, potentially revealing underlying biases in the processes that generated them. We also show that a binomial variation of this hypothesis test could be used to identify bias in certain directions, or towards certain outcomes, and again return a closest plausible explanation. The benefits of this binomial variation are compared with other hypothesis tests, including the exact binomial. Lastly, potential industrial applications of our methods are shown using two real-world datasets.

CCS CONCEPTS

• **Mathematics of computing** → **Hypothesis testing and confidence interval computation**; **Computing most probable explanation**; • **Information systems** → *Data analytics*; • **Social and professional topics** → Race and ethnicity; Gender.

KEYWORDS

machine learning, bias, fairness, statistics, hypothesis testing, data analysis

ACM Reference Format:
William Yik, Limnanthes Serafini, Timothy Lindsey, and George D. Montañez. 2022. Identifying Bias in Data Using Two-Distribution Hypothesis Tests. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society (AIES'22), August 1–3, 2022, Oxford, United Kingdom.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3514094.3534169

1 INTRODUCTION

Not every deviation from expectation is meaningful. As the head of human resources for a national organization, you stare at a summary of recent hires and wonder: is the deviation from the expected proportion of hired female applicants merely a sampling fluctuation, or might the company's hiring practices be biased against women? How could you test whether a fair process might have plausibly generated the data in front of you?

Machine learning (ML) algorithms have become increasingly prevalent in a variety of applications with real-world implications for millions of people [3, 17]. The fairness of these algorithms has become an issue of utmost importance, especially in classification tasks. It is well known that such algorithms can produce prejudicial outcomes against certain groups of people if they are trained on *biased* data [1]. Throughout this manuscript, we take **bias** to be over- or under-representation of a specific value (e.g., women) of a protected attribute (e.g., gender) within a given dataset, relative to some expected outcome.

Identifying biases in training data and mitigating their impact in classification tasks has become a central issue in the ML community [7]. In this manuscript, we propose a new method of bias identification using the two-distribution (specified complexity) hypothesis tests of Montañez [20], which distinguish themselves in their ability to statistically rule out whole sets of unbiased possible hypotheses, leaving only biased hypotheses as plausible explanations for the data. As such, they can identify whether an unbiased process is a likely explanation for the data without having to train or examine the results of a model, allowing us to narrow down potential sources of bias in the ML workflow.

Additionally, if it was concluded that a proposed process could not have plausibly produced the data, our methods allow us to return a "closest plausible explanation" which is the explanation closest to the original that is not rejected by our test. This gives the results of our test a unique degree of clarity, since this new explanation acts as a better representation of the process that generated the data than the original proposition.

The remainder of the paper is structured as follows. In Section 2, we discuss other relevant work that has been done in the areas of machine learning bias and hypothesis testing. Section 3 introduces

[*] Both authors contributed equally to this research.

AIES'22, August 1–3, 2022, Oxford, United Kingdom

ACM ISBN 978-1-4503-9247-1/22/08…$15.00
https://doi.org/10.1145/3514094.3534169

the necessary background for our novel hypothesis test. Section 4 explains how we use this hypothesis test to identify bias in data. A binomial variation of this test is introduced and discussed in Section 5. Our experimental setup, including details regarding the datasets we tested, is given in Section 6. Section 7 presents our experimental results. Lastly, Section 8 concludes the paper with discussion on the broader impact of our hypothesis tests. [1]

2 RELATED WORK

Much work has been done to identify bias in training data [4, 19]. In recent years, a wide array of methods have been developed that identify bias within a given dataset by analyzing outcomes generated by models trained on those data. Notable examples include AI Fairness 360 [6] and Aequitas [22]. Other work has been done in the area of natural language processing to examine the location of bias in algorithms by altering individual neurons within a neural network [26].

A few papers use hypothesis testing to detect discriminatory behavior of an algorithm. One such paper, inspired by ideas from optimal transport theory, uses a test statistic based on Wasserstein distance [23]. Others use modified permutation tests based on variations of Pearson's correlation statistic [11, 24]. Whether relating to hypothesis testing or not, the above methods focus on analyzing the *output* of a trained algorithm for bias. However, since the root cause of the problem is the training dataset itself, evaluating a dataset *before* training is the most logical way to address the root of the problem [4].

There are methods that do not train or analyze any model at all and instead evaluate the training data itself. For the task of recognizing textual entailments, biased data can be identified beforehand, and it has been shown that there are tangible performance deficits for models trained on these data [25]. In the realm of classification, most of the relevant work comes from the field of data management. Many papers construct casual networks between attributes (e.g., gender) and outcomes or labels (e.g., hired or not hired) in order to discover discriminatory relationships between them [27, 28]. However, it has also been noted that such methods are often not easily accessible by everyday practitioners and may require in-depth knowledge of causality-based fairness [4].

Methods for combating bias include re-weighting data after examining the output of a biased classifier [16]; altering the dataset itself for parity [12]; using Lagrangian constraints in algorithms to enforce fairness [9]; attempting to remove the causal path between sensitive attributes and decisions [21]; or using maximum entropy principles to modify the statistical rates of protected groups while remaining close to the original distribution [8].

While these methods can remove a certain degree of bias from ML models, the importance of analyzing training data still holds since such options invariably produce less accurate models. In fact, the more unbiased a model is forced to be, the less correct its predictions become [12]. This is because all measures of correctness are tied directly to the training data used. Thus, attention *must* be paid to the data.

Our methods primarily build on previous work dealing with two-distribution (specified complexity) hypothesis tests [20]. Such hypothesis tests have recently been used in several applications including the provision of intention perception to artificial agents, which was shown to provide survival advantages [15]. Other recent work has used similar methods to analyze the hypothesis that a search algorithm's probability of reaching its target is equivalent to blind chance [10].

3 BACKGROUND

Objects that are both improbable and structurally organized are said to have high *specified complexity* [20]. Such objects are both unlikely to occur under a given probability distribution (complex) and fit a predetermined notion of form or functionality (specific). A *specified complexity model* can capture this combination of unlikeliness and conformity. Such models are functions of X which can take on values in the space $\mathcal{X}$ according to probability distribution P (denoted as $X \sim P$) with probability function $p(x)$, which is the component of unlikeliness (complexity) in our specified complexity model. The other component is a specification function $\nu(x)$ which captures how well an observation conforms to a predetermined notion of structure. Following Montañez [20], we formalize these ideas with the following definitions.

Definition 1 ($\nu(\mathcal{X})$, Montañez 2018). For any integrable, nonnegative specification function $\nu : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$,

$$\nu(\mathcal{X}) := \begin{cases} \int_{\mathcal{X}} \nu(x)\, dx & \text{if continuous,} \\ \sum_{x \in \mathcal{X}} \nu(x) & \text{if discrete,} \\ \int_{\mathcal{X}} d\nu(x) & \text{in general.} \end{cases} \tag{1}$$

Definition 2 (Common Form and Kardis, Montañez 2018). For any probability distribution P with probability function $p(x)$ on space $\mathcal{X}$, any strictly positive scaling constant $r \in \mathbb{R}_{>0}$ and any nonnegative function $\nu : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$, we define a *common form* model as

$$SC(x) := -\log_2 r \frac{p(x)}{\nu(x)}$$

with *specified complexity kardis*

$$\kappa(x) = r(p(x)/\nu(x)).$$

Definition 3 (Canonical Specified Complexity Model, Montañez 2018). Any common form model constrained such that $\nu(\mathcal{X}) \leq r$ is a *canonical specified complexity model.*

In our work, we primarily focus on canonical specified complexity models and their corresponding kardis values because they possess useful properties and share a close relationship with traditional p-value statistical hypothesis tests [10, 20]. Namely, these models posses level-α properties that allow for hypothesis testing using the kardis as the test statistic. This is formalized in Theorem 1 from Montañez [20].

Theorem 1 (Level-α Property for Canonical Specified Complexity Models, Montañez 2018). *Given $X \sim P$ and significance level $\alpha \in [0, 1]$, let $\kappa(x)$ be the kardis from any canonical specified complexity model. Then* $\Pr(\kappa(X) \leq \alpha) \leq \alpha$.

[1] See source code, experimental notebooks, and relevant datasets at https://github.com/AMISTAD-lab/bias-in-data-source

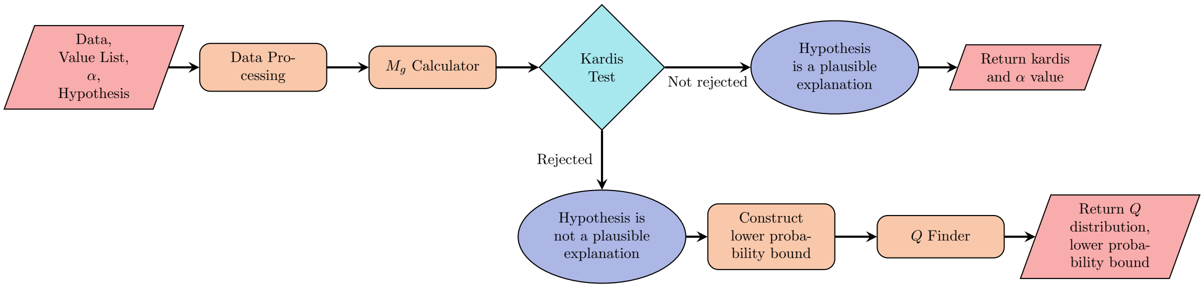
Data,
Value List,
α,
Hypothesis
Data Processing
M_g Calculator
Kardis Test
Not rejected
Hypothesis is a plausible explanation
Return kardis and α value
Rejected
Hypothesis is not a plausible explanation
Construct lower probability bound
Q Finder
Return Q distribution, lower probability bound

Figure 1: FSC Hypothesis Test Workflow.

If a hypothesis P has a sufficiently large $\kappa(x)$ to avoid rejection under a given hypothesis test, we call it a **plausible explanation**. This should not be confounded with other meanings of plausibility such as having a high probability, and users of these hypothesis tests should carefully consider how they interpret the results.

These hypothesis tests based on specified complexity models not only allow us to reject, or fail to reject, a singular proposed hypothesis, but also form rejection regions for whole sets of hypotheses (see Section 4.2).

While there are many canonical specified complexity models that could be used for such hypothesis tests, the Functional Specified Complexity (FSC) model proposed by Montañez [20], based on functional information [13], is particularly useful because it works with finite, discrete data, representative of the datasets nearly all learning algorithms train on, and eliminates the need to estimate the specification normalization factor r [20]. We begin our discussion of FSC by first defining $g : \mathcal{X} \longrightarrow \mathbb{R}_{\geq 0}$ as some function that increases with increasing degrees of extremity for an observation x and $F_g(x)$ as the proportion of events in space $\mathcal{X}$ that exhibit at least the same degree of extremity as the observation. We expand this discussion and formalize extremity in Section 4.1. Following Montañez [20], we define $M_g(x) := |\{x' \in \mathcal{X} : g(x') \geq g(x)\}|$, which gives the *functional specificity*

$$F_g(x) = \frac{M_g(x)}{|\mathcal{X}|}.$$

With this in mind, FSC is formally defined as follows.

Definition 4 (Functional Specified Complexity, Montañez 2018)**.** For function g, functional specificity $F_g(x)$, and probability function $p : \mathcal{X} \longrightarrow [0,1]$, the *functional specified complexity kardis* is

$$\kappa(x) := |\mathcal{X}|(1 + \ln|\mathcal{X}|)\frac{p(x)}{F_g(x)^{-1}}.$$

where we have defined $r = |\mathcal{X}|(1 + \ln|\mathcal{X}|)$ and $\nu(x) = F_g(x)^{-1}$.

Since FSC gives concrete formulas for r and $\nu(x)$, it is useful for conducting hypothesis tests using the level-α properties of specified complexity from Theorem 1. We describe our method for conducting these tests in the following section.

4 METHODOLOGY

The workflow for our hypothesis test is visualized in Figure 1. Computing the FSC kardis requires a definition for $\mathcal{X}$. There are many definitions that could be used. Our methodology defines $\mathcal{X}$ as the space of *count vectors*—representations of the frequency of each possible outcome of a random variable. Under this definition, if we rolled a die 10 times and observed the outcome x of 4 ones, 3 twos, 2 threes, 1 four, x would be represented by the count vector $\mathbf{X} = [4, 3, 2, 1, 0, 0]$. This lets us calculate $M_g(x)$ without enumerating every possible sequence, which would become costly as event size increases.

4.1 $M_g(x)$ Calculator

Consider a discrete random variable X which follows a categorical distribution, with m mutually exclusive outcomes and corresponding probabilities $p_1, p_2, \ldots, p_m$. If we observe X over n trials, and X_i represents the number of times the outcome i was observed, then the *count vector*

$$\mathbf{X} = [X_1, X_2, \ldots, X_m]$$

will follow a multinomial distribution. The mean for this distribution will be

$$\mathbb{E}[\mathbf{X}] = [\mathbb{E}[X_1], \mathbb{E}[X_2], \ldots, \mathbb{E}[X_m]]$$

where each $\mathbb{E}[X_i] = np_i$. Let the *distance vector* $\mathbf{D}$ of $\mathbf{X}$ from $\mathbb{E}[\mathbf{X}]$ be defined by

$$\mathbf{D} = [D_1, D_2, \ldots, D_m] \tag{2}$$

where each $D_i = X_i - \mathbb{E}[X_i]$. Note that a bijection exists between $\mathbf{X}$ and $\mathbf{D}$, such that each count vector maps to unique distance vector and vice versa.

Now, we must determine how the function $g : \mathcal{X} \longrightarrow \mathbb{R}_{\geq 0}$ measures the degree to which the count vector $\mathbf{X}$ diverges from the mean count vector $\mathbb{E}[\mathbf{X}]$, or its extremity. Our method of quantifying extremity is the L_1 taxicab distance metric:

$$L_1(\mathbf{X}, \mathbb{E}[\mathbf{X}]) = \sum_{i=1}^{m} |X_i - \mathbb{E}[X_i]|.$$

Using this measure of extremity, g can be formally computed as

$$g(x) = \sum_{i=1}^{m} |X_i - \mathbb{E}[X_i]| = \sum_{i=1}^{m} |D_i|. \tag{3}$$

While other notions of extremity could be used, such as the L_2-norm, this measure is the key for calculating $M_g(x)$, the number of count vectors at least as extreme than the observation $\mathbf{X}$. Specifically, it allows for use of combinatorics to count events for $M_g(x)$, without the need to enumerate those events. The details of this computation are given in Section A of the Appendix.

4.2 FSC Test and Constructing s Lower Bound

Alongside $M_g(x)$, we also need $|\mathcal{X}|$ to compute the $\nu(x)$ and r components of our FSC model as per Definition 4. Letting m denote the number of possible categories, $\mathcal{X}$ is the space of all possible count vectors $\mathbf{X} = [X_1, X_2, \ldots, X_m]$, on which there are certain helpful constraints. Namely,

$$\sum_{i=1}^{m} X_i = n$$

where X_i is the frequency of event i and n is the length of the observation x. Thus, $|\mathcal{X}|$ is the number of ways m non-negative integers can sum to n which is given by Lemma 2 in Section A of the Appendix,

$$|\mathcal{X}| = \binom{n+m-1}{m-1}. \tag{4}$$

Once $|\mathcal{X}|$ is used to calculate $\nu(x)$ and r, the last component we need to compute $\kappa(x)$ is $p(x)$. As stated in the previous subsection, the count vectors $\mathbf{X}$ follow a multinomial distribution. Thus, the probability of seeing event x is given by the probability mass function (PMF) of a multinomial distribution,

$$p(x) = \frac{n!}{X_1! \cdots X_m!} p_1^{X_1} \cdots p_m^{X_m} \tag{5}$$

where each p_i is the probability of observing outcome i under distribution P. These values can now be used to compute the kardis $\kappa(x) = r(p(x)/\nu(x))$. If $\kappa(x) \leq \alpha$, we can reject the null hypothesis. We then follow the method of Montañez [20] to construct a lower bound on how much an explanation needs to boost the probability of observing x in order to be considered a plausible explanation. Any plausible explanation must boost the probability of observing x by a factor of

$$s \geq \frac{\alpha \nu(x)}{r p(x)} \tag{6}$$

over the proposed explanation given by P [20]. Let

$$s_{min} = \frac{\alpha \nu(x)}{r p(x)}. \tag{7}$$

In order to even be considered as a plausible explanation for the data, a new distribution Q which confers $q(x)$ probability on event x must satisfy the condition

$$q(x) \geq s_{min} \cdot p(x). \tag{8}$$

This lower bound allows us to rule out whole sets of explanations which do not sufficiently increase the chance of observing x. In practice, this may allow users to eliminate unbiased hypotheses as a whole, leaving only biased hypotheses as plausible explanations.

4.3 Q Finder

If a proposed hypothesis P is rejected, one may wish to find *plausible* explanations. The methods discussed previously allow us to return the closest possible explanation to the proposed one which is *not* rejected by the hypothesis test, and thus is a *plausible explanation* relative to that test. Denote this explanation as Q. Since Q is a probability distribution which gives $q_1, \ldots, q_m$ probability to each possible value $i = 1, \ldots, m$ of random variable X, we must have

$$\sum_{i=1}^{m} q_i = 1. \tag{9}$$

Furthermore, Q must satisfy the lower bound condition of Equation 6. If $q(x)$ is the probability of observing x under distribution Q defined similarly to $p(x)$ as

$$q(x) = \frac{n!}{X_1! \cdots X_n!} q_1^{X_1} \cdots q_n^{X_n},$$

then we must also have

$$q(x) \geq s_{min} \cdot p(x) \tag{10}$$

We can define the "closeness" of Q to P using KL-divergence,

$$D_{\mathrm{KL}}(Q \| P) = \sum_{i=1}^{m} q_i \log\left(\frac{q_i}{p_i}\right). \tag{11}$$

However, the choice of KL-divergence is not mandatory; other measures of distance, such as Jensen-Shannon divergence or earth mover's distance, could be used with similar results. With Equation 11 in mind, the task of finding the closest distribution Q to P such that our hypothesis test fails to reject Q is simply a constrained optimization problem, with Equation 11 as the cost function subject to the constraints defined by Equations 9 and 10. To solve this problem, we use the sequential quadratic programming method of Kraft [18]. We provide illustrative examples of closest plausible distributions for a real-world dataset in Section 7. It should be noted that the fairness of the closest plausible distribution Q should be evaluated by the user in the *exact same* way the fairness of the proposed hypothesis P is, and that this method for determining the fairness of a distribution should be determined *beforehand* by the user. Otherwise, there is potential for misinterpretation of the test's results.

5 BINARY HYPOTHESIS TESTING

While specified complexity hypothesis tests may have been previously unknown to the reader, many have already used them without realizing it. Montañez has pointed out that every p-value hypothesis test corresponds to a specified complexity hypothesis test [20], and we now show that a binomial probability mass function itself is a specified complexity kardis. Thus, a binomial probability mass function can be used to perform a specified complexity hypothesis test, without having to compute an exact tail probability. We then illustrate how this new binomial specified complexity test could be used to detect bias in binary scenarios and return a closest possible explanation akin to that of the previous section. Lastly, we compare the statistical power of the traditional FSC test, binomial specified complexity test, and exact binomial test, giving examples of when each may be useful for identifying bias in data.

5.1 Binomial Specified Complexity Test

If X is a binomially distributed random variable (denoted $X \sim \mathrm{Bin}(n, p)$) with parameters n (the number of trials) and probability

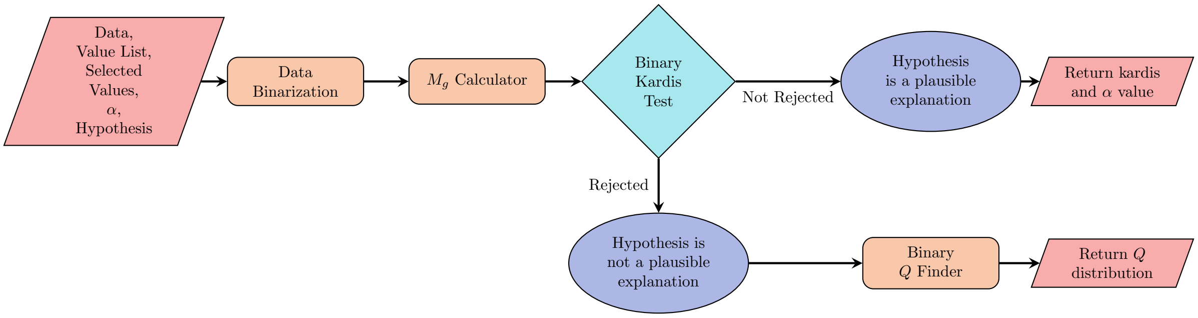

Figure 2: Binary Hypothesis Test Workflow.

of success $p \in [0,1]$, then its probability mass function (PMF) is

$$f(k_x, n, p) = \Pr(X = k_x) = \binom{n}{k_x} p^{k_x}(1-p)^{n-k_x}$$

where k_x is the number of successes in observation x. We can see that certain components of this function parallel that of a specified complexity model. Namely, $p^{k_x}(1-p)^{n-k_x}$ mirrors the probability distribution component of a specified complexity model, $p(x)$, and the $\binom{n}{k_x}$ component appears to capture some amount of specificity of a sequence. These parallels are formalized in Proposition 1.

Proposition 1. The PMF of a random variable $X \sim \text{Bin}(n,p)$ is a common form kardis with

$$\kappa(x) = \Pr(X = k_x) = \binom{n}{k_x} p^{k_x}(1-p)^{n-k_x} = r\frac{p(x)}{\nu(x)},$$

with $p(x) = p^{k_x}(1-p)^{n-k_x}$, $r = 1$, and $\nu(x) = \binom{n}{k_x}^{-1}$. Furthermore, the Shannon surprisal of the PMF is a common form specified complexity model, namely,

$$SC(x) = -\log_2 \Pr(X = k_x) = -\log_2 r\frac{p(x)}{\nu(x)}.$$

The proposition follows directly from the definitions of the binomial PMF and the common form (Definition 2). It is important to note that this model is not canonical in general, since $\nu(\mathcal{X})$ may exceed r. However, we can define r such that $\nu(\mathcal{X}) \le r$, turning our model into a canonical one with useful level-α properties. Since ν is discrete, by Definition 1,

$$\nu(\mathcal{X}) = \sum_{x\in\mathcal{X}} \nu(x) = \sum_{x\in\mathcal{X}} \binom{n}{k_x}^{-1},$$

the formula for which is given by Lemma 1 with proof in Section C of the Appendix.

Lemma 1. *If $\mathcal{X}$ is the space of all possible outcomes of $X \sim Bin(n,p)$, then*

$$\sum_{x\in\mathcal{X}} \binom{n}{k_x}^{-1} = n+1.$$

where k_x is the number of successes in event x.

Thus, if we define $r = n+1$, then $\nu(\mathcal{X}) \le r$, and we can form a new canonical specified complexity model for binary scenarios.

Definition 5 (Binomial Specified Complexity (BSC))**.** For a random variable $X \sim \text{Bin}(n,p)$, and number of successes k_x, the *binomial specified complexity kardis* is

$$\begin{aligned}\kappa(x) &:= (n+1)\Pr(X = k_x)\\ &= (n+1)\binom{n}{k_x} p^{k_x}(1-p)^{n-k_x}\end{aligned}$$

where we have defined $r = n+1$, $p(x) = p^{k_x}(1-p)^{n-k_x}$, and $\nu(x) = \binom{n}{k_x}^{-1}$.

Since BSC is a canonical model, we can use the level-α of Theorem 1 to conduct a hypothesis test similar to the FSC test of Section 4, the workflow for which is shown in Figure 2. Specifically, we reject a proposed hypothesis P, which has probability of success $p \in [0,1]$, at a given α value if, for $X \sim P$,

$$\kappa(X) = (n+1)\Pr(X = k_x) \le \alpha.$$

Further paralleling the FSC test of Section 4, we can also form rejection regions for proposed hypotheses, providing a lower bound for the probability mass an explanation Q must give to the observed event in order for it to be considered plausible. By Equation 7, Q must boost the probability of observing x by at least

$$\begin{aligned}s_{min} = \frac{\alpha\nu(x)}{rp(x)} &= \frac{\alpha\binom{n}{k_x}^{-1}}{(n+1)(p^{k_x}(1-p)^{n-k_x})}\\ &= \frac{\alpha}{(n+1)\Pr(X = k_x)}.\end{aligned}$$

Thus, by Equation 10 in order for an explanation Q to be considered plausible, it must impart

$$\begin{aligned}q(x) &\ge s_{min}\cdot p(x)\\ &= \frac{\alpha}{(n+1)\Pr(X = k_x)}p(x)\\ &= \frac{\alpha}{(n+1)\binom{n}{k_x}}\end{aligned}$$

probability mass on event x. However, by Definition 5, $q(x) = q^{k_x}(1-q)^{n-k_x}$ where $q \in [0,1]$ is the probability of success under Q. Thus, we must have

$$\begin{aligned}q(x) &\ge \frac{\alpha}{(n+1)\binom{n}{k_x}}\\ q^{k_x}(1-q)^{n-k_x} &\ge \frac{\alpha}{(n+1)\binom{n}{k_x}}.\end{aligned}$$

This formulation allows us optimize for q similarly to the method Section 4.3 and return Q, the closest plausible distribution to P which is not rejected by the BSC test.

5.2 Exact Binomial Test

The main BSC test sacrifices exact values for speed. In contrast, one could use an exact binomial test to compute the tail probability explicitly, trading speed for precision. Our method for doing so is as follows. First we binarize the data, then perform a one-sided (greater) binomial test using the following equation:

$$\sum_{k=i}^{n} \binom{n}{k} p_b^k (1-p_b)^{n-k}$$

where n is the length of the dataset, i is the count of selected values, and p_b is the user-given baseline probability of obtaining a selected value. In order to accommodate realistically-sized datasets in reasonable amounts of time, we use Stirling's approximation to calculate ${}_nC_k$, namely,

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

yielding

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \approx \frac{\sqrt{n} \times n^n}{\sqrt{2\pi k(n-k)} \times k^k (n-k)^{(n-k)}}.$$

Once we calculate the tail probability, it is compared to the given α value. If the probability is smaller than α, the hypothesis is naturally rejected. However, the algorithm also considers the case in which the tail probability is greater than $1-\alpha$: a situation in which the dataset is likely biased *against* the selected values. If this is the case, i and p_b are redefined to be the count and probability of the non-selected values. For either of these rejection cases, the algorithm then calculates the coefficient closest to one on p_b that is necessary to produce a valid hypothesis. This coefficient is called s': like s, it is a scalar value that raises probability. However, s is a coefficient on a probability mass function, not on the probability of specific values. The upper bound on s' may easily be calculated as $1/p_b$. The s' value is lowered from there so that the binomial tail constructed using the probability of $s' \cdot p_b$ is as close to α as possible while still being a valid explanation. This is done by manipulating one power of ten at a time - first subtracting 10^0 until going further would drop the tail probability below α. This process is then repeated for 10^{-1}, then 10^{-2}, until a user-defined number of significant figures have been optimized. The s' value is multiplied by the relevant probability, and the resulting distribution is returned to the user alongside a boolean value representing whether the hypothesis was rejected.

5.3 FSC for Binary Scenarios

It should be noted that the FSC hypothesis test of Section 4 could still be used for the binary scenarios described in this section. Since there are only two possible outcomes, by Equation 4 we have

$$|\mathcal{X}| = \binom{n+2-1}{2-1} = n+1$$

giving

$$r = |\mathcal{X}|(1 + \ln|\mathcal{X}|) = (n+1)(1+\ln(n+1)).$$

Additionally, by Equation 5

$$\begin{aligned} p(x) &= \frac{n!}{X_1! \cdots X_m!} p_1^{X_1} \cdots p_m^{X_m} \\ &= \frac{n!}{k_x!(n-k_x)!} p^{k_x}(1-p)^{1-k_x} \\ &= \binom{n}{k_x} p^{k_x}(1-p)^{1-k_x} \\ &= \Pr(X = k_x), \end{aligned}$$

where k_x is the number of successes in observation x, n is the total number of trials, and p is the probability of success.

$M_g(x)$ can also be easily calculated for a binary scenario, since any event whose number of successes is farther from the mean (above *or* below) than the observation is at least as extreme as that observation. The number of events *less* extreme than the observation is

$$2|k_x - np| - 1.$$

For example, if $k = 60$, $n = 100$, and $p = 0.5$, then there would be $2|60-50| - 1 = 19$ events less extreme than the observation, corresponding to the events with 41 to 59 successes, inclusive. It follows that the number of sequences at least as extreme as the observation is

$$\begin{aligned} M_g(x) &= |\mathcal{X}| - (2|k_x - np| - 1) \\ &= (n+1) - (2|k_x - np| - 1) \\ &= n - 2|k_x - np| + 2. \end{aligned}$$

Thus,

$$\begin{aligned} \nu(x) &= F_g(x)^{-1} \\ &= \frac{|\mathcal{X}|}{M_g(x)} \\ &= \frac{n+1}{n - 2|k_x - np| + 2}, \end{aligned}$$

and we have an FSC kardis of

$$\begin{aligned} \kappa_{\text{FSC}}(x) &= r\frac{p(x)}{\nu(x)} && (12) \\ &= (n+1)(1+\ln(n+1))\frac{\Pr(X=k_x)}{(\frac{n+1}{n-2|k-np|+2})} && (13) \\ &= (n-2|k_x - np| + 2)(1+\ln(n+1))\Pr(X=k_x). && (14) \end{aligned}$$

5.4 Comparing Hypothesis Tests for Binary Cases

While the exact binomial test has the most statistical power of the three hypothesis tests for binary scenarios presented thus far, it poses performance challenges for large datasets, especially when one seeks to find the next best explanation using the method of Subsection 5.2. As such, the FSC and BSC tests, while statistically less powerful (namely, they return larger tail probabilities for the same observations), become more applicable in such scenarios. Comparing the strengths of these two tests, we see that for an observation of n trials and k_x successes, the BSC test returns a smaller tail probability bound than the FSC test until k_x reaches some extreme value far away from the mean. This is formalized in Theorem 2, with proof in Section C of the Appendix.

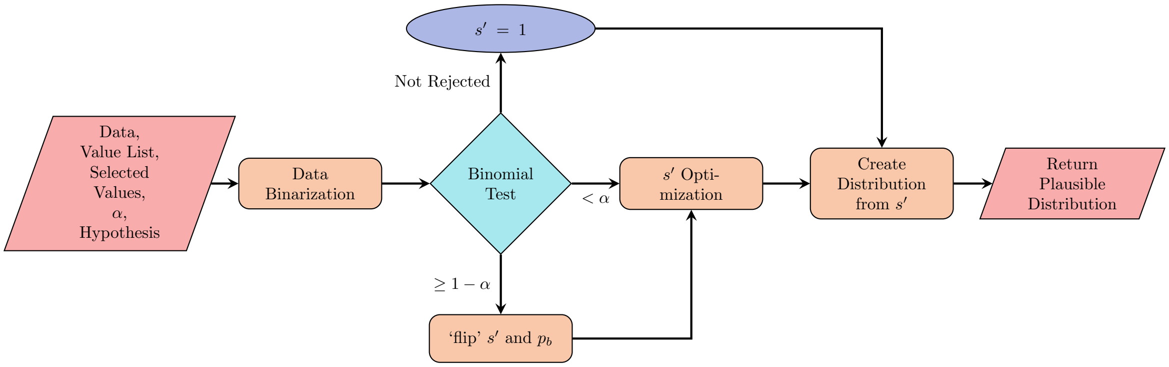

Figure 3: Exact Binomial Test Workflow.

Theorem 2. *For a random variable $X \sim Bin(n, p)$ and number of successes k_x, a BSC hypothesis test will return a smaller tail probability bound than an FSC hypothesis test for all*

$$np - c < k_x < np + c$$

where

$$c = \frac{n}{2} - \frac{n+1}{2 + 2\ln(n+1))} + 1.$$

These "flipping points" for when the FSC test becomes stronger than the BSC test are typically quite close to 0 and n when $p = 0.5$. For example, if $n = 100$, Theorem 2 gives the bound

$$7.99 < k_x < 92.01,$$

and when $n = 1000$ the bound is

$$62.28 < k_x < 937.72.$$

Another important distinction to make between the three tests presented is the k_x values for a given n at which they begin to reject a proposed hypothesis. These values for $n = 100$ and $n = 1000$ are shown in Table 1. Lastly, we compare the closest plausible distributions that each test returns for an observed percentage of successes. The returned probabilities of success are shown in Table 2.

Table 1: Minimum k_x successes out of $n = 100$ trials and $n = 1000$ trials required for rejection of fair hypothesis ($p = 0.5$) at various α levels.

n	α	Exact Binomial	BSC	FSC
100	0.1	57	65	67
	0.05	59	66	68
	0.01	63	69	70
	0.001	66	71	73
	0.0001	69	74	75
1000	0.1	521	553	562
	0.05	527	556	564
	0.01	538	563	570
	0.001	550	572	578
	0.0001	560	579	585

Table 2: The returned closest plausible probability of success for different observed success rates with $\alpha = 0.05$ and original hypothesis $p = 0.5$.

% Successes	Exact Binomial	BSC	FSC
60	0.574	0.545	0.537
65	0.625	0.595	0.588
70	0.675	0.647	0.641
75	0.727	0.699	0.694
80	0.778	0.752	0.748

6 EXPERIMENTAL SETUP

To validate the usability of our method, we applied our FSC hypothesis test to two real-world datasets. The first was the UCI Adult dataset [2], which is used to train algorithms that predict whether a U.S. adult will make more than $50,000 in annual income. For 48,842 adults, it contains a variety of information such as gender, race, education level, and marital status, as well as a binary label ($m = 2$) of "≤ 50K" or "> 50K". It is known that algorithms trained on this dataset exhibit bias against women compared to men [6], disproportionately assigning women to the lower income bracket. As such, we sought to test the process that generated the dataset for bias as defined in Section 1. Does this process disproportionately assign labels in a way that deviates from a fair hypothesis? To implement this, we used the proportions of men assigned to each of the two labels as our null hypothesis and used all female entries (n = 16,192) as our data for our FSC test. This gave us a null hypothesis of $P = [0.696, 0.304]$ and a count vector of $X = [14423, 1769]$. The chosen significance level was $\alpha = 0.05$. Since we are testing a proposed explanation, our null hypothesis tests whether the same process that generated the male income labels (as represented by the sample) could plausibly explain the female income labels. In the real world, this process includes a multitude of societal factors.

The second dataset we tested was the well-known COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) dataset [1]. This dataset was produced by the COMPAS algorithm, which attempts to accurately estimate potential risks associated with an offender: the risk of violence, risk of recidivism, and risk of failure to appear in court. For each risk, an offender was given a low, medium, or high rating. While the dataset was produced by an algorithm itself, it is often used to train other models to highlight the original algorithm's unfair behavior [6]. Most analyses in the literature use a filtered version of the dataset with 6,172 entries that contains true labels for whether a given person recidivated or not [6, 11, 23]. However, this filtered dataset is known to contain errors accrued during preprocessing by its original publishers, ProPublica [5]. Furthermore, we are interested in testing for bias in the dataset without training a model and determining whether a proposed explanation (i.e., that the original algorithm and everything else upstream was fair) could plausibly explain those data. As such, we use the original, unfiltered dataset, which contains 11,757 entries, since we have no need for the true labels of whether a person recidivated or not.

To illustrate the impact of choosing different null hypotheses, we conducted several tests on different subsets of the COMPAS dataset, all of which used the significance level $\alpha = 0.05$. First, we analyzed the recidivism risk scores in isolation, and tested the African-American subset for bias using two different null hypotheses:

(1) **The proportion of Caucasians assigned to each of the three risk categories.** This choice of hypothesis and data allows us to test if the process that produced the Caucasian recidivism scores could also be a reasonable explanation for the African-American scores.
(2) **A uniform distribution which gives 1/3 probability to each of the three risk categories.** This is equivalent to testing whether a completely random assignment process could have plausibly generated the African-American recidivism risk scores.

Secondly, to test for bias in more specific subsets of the dataset, we broke the data down based on charge type. We binarized the categories of charges for our testing into *Battery* and *Possession of Contraband*. *Possession of Contraband* included COMPAS charge descriptions that denoted possession of an illegal item or substance without the intent to sell (e.g., *Possession of Cocaine*). For each charge category, we tested the recidivism scores of the corresponding African-American subset using the score proportions of Caucasians charged with the same crime as the null hypothesis.

7 MAIN RESULTS

All of the following tests were conducted on a 16-core AMD Ryzen 9 5900HS CPU, with run times under 5 seconds.

7.1 UCI Adult

Our test returns a kardis value of 2.47×10^{-755} which is far less than our significance level of $\alpha = 0.05$. As such, we reject the null hypothesis that the same process that generated the male income labels could plausibly explain the female income labels, if the male sample is representative of its distribution. Furthermore, using Equation 7, we find that in order for an explanation to be considered plausible, it must boost the probability of the observing the data by at least $s_{min} = 2.02 \times 10^{753}$. The closest plausible distribution constructed from this s_{min}, as well as the hypothesis distribution are shown in Figure 4. The colors of the "lollipop" bars representing the closest plausible distribution are colored according to their proximity to the null hypothesis distribution, with a heatmap included at the right side of the figure. In the context of this dataset, the null hypothesis distribution shown in the Figure 4 represents the proportion of males assigned to each income label group and the closest plausible distribution represents the closest distribution of female income labels to the null that could explain the actual proportions of female income labels. These results are consistent with the existence of known biases in the dataset [24] while additionally providing an extra degree of interpretability with a numerical s_{min} value and a closest plausible distribution.

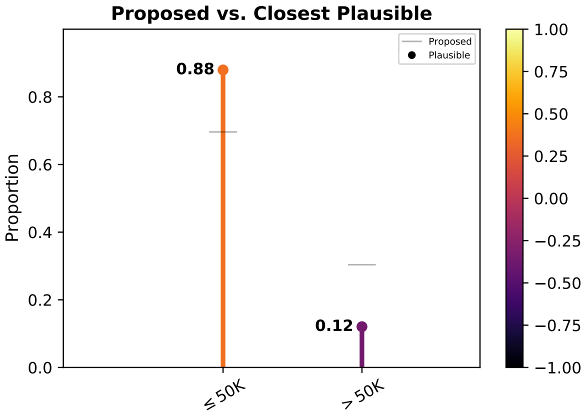

Figure 4: Closest plausible income label distribution for females in UCI Adult, compared to male proportion.

7.2 COMPAS

The first test for African-American recidivism scores using Caucasian proportions as the null returns a kardis value of 8.99×10^{-392}. This is far below our α level of 0.05, so we reject the null hypothesis. The s_{min} value associated with this kardis is 5.57×10^{389}, requiring any plausible explanation to boost probability by almost three hundred and ninety orders of magnitude. Shown in Figure 5 is the closest distribution that can plausibly explain the African-American data—could it also explain the Caucasian data, being closest to its proportions? Reversing directions, we test the Caucasian scores against the distribution from Figure 5, and find that it is also rejected with a kardis value of 9.19×10^{-184}. As such, we can conclude that the process that generated the Caucasian recidivism score distribution could not plausibly explain the African-American scores, and vice versa.

This test result is consistent with the existence of known biases in the COMPAS dataset [1] while also providing additional useful results with s_{min} and the closest plausible distribution. As seen in the graph, the closest plausible distribution requires lowering the proportion of Low scores while increasing those of Medium and High. This is consistent with the view that the data contains bias against African-Americans, as less favorable outcomes are associated to them with greater frequency than in our conjecture.

Somewhat less interestingly, the test for African-American recidivism scores using uniform proportions as the null returns a kardis value of 2.47×10^{-44} and an s_{min} value of 2.03×10^{42}. This implies that a completely random uniform process cannot plausibly

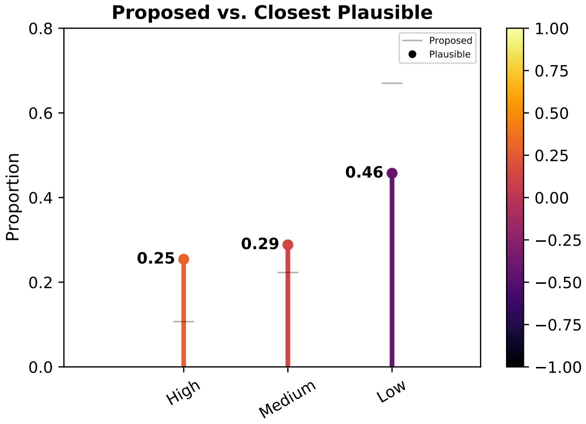

Figure 5: Closest plausible COMPAS recidivism score distribution for African-Americans, compared to Caucasian proportion.

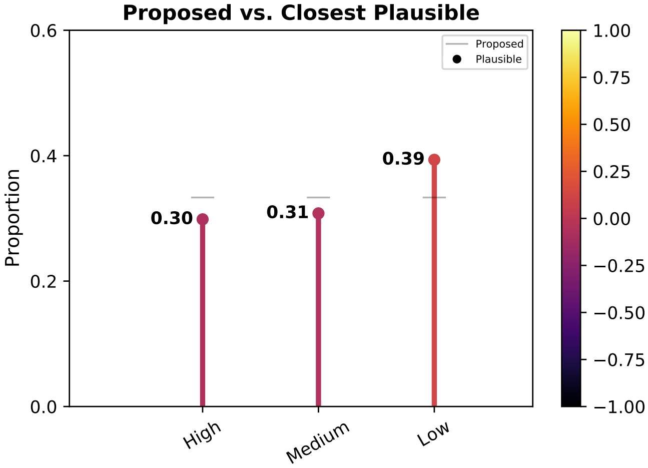

Figure 6: Closest plausible COMPAS recidivism score distribution for African-Americans, compared to a uniform proportion.

explain the African-American recidivism scores. A graph representing the closest plausible distribution in this case is shown in Figure 6, which, as a reminder, gives a distribution that would *not* be rejected under the same hypothesis test.

Upon seeing that the favorable outcome of 'Low' is given to African-Americans with greater frequency than in the null hypothesis, one might be tempted to conclude that the data are slightly biased *towards* African-Americans. This illustrates how bias is always assessed relative to one's original proposed explanation. Furthermore, these statistical tests can pinpoint the *location* and *direction* of biases in data, but further investigation is necessary to determine if the frequential biases identified are actually the result of prejudicial biases, or if they have some other explanation.

The test for recidivism scores of African-Americans charged with *Battery* returns a kardis value of 9.23×10^{-34}, and the test for recidivism scores of African-Americans charged with *Possession of Contraband* returns a kardis value of 4.91×10^{-44}. Both of these kardis values are far smaller than our α value of 0.05. As such, we reject both null hypotheses and conclude that the process that generated the recidivism scores for Caucasians charged with a given crime does not plausibly explain the recidivism scores for African-Americans charged with the same crime.

It is interesting to note that our tests found bias in the African-American recidivism scores both in general and when broken down by charge type. While we offer no conjecture regarding any underlying sociological or causal explanation for the results of our tests, a rejected hypothesis identifies a point in the machine learning workflow where a user should pause to further investigate the data and its generating process, taking greater caution when using the data to train any model.

7.3 Contributions and Comparison with Previous Work

The methods forwarded in this paper are primarily extensions of Montañez [20] but provide some significant, novel contributions. First, while Montañez formalized a specified complexity hypothesis test, we specifically implement this test so it can be applied to machine learning training data. We also define each abstract term used in the hypothesis test (e.g., $M_g(x)$) so that they can be used for actual calculations. Second, we uniquely forward a method to identify the closest plausible distribution beyond the trivial binary case. Lastly, we demonstrate how the tests of Montañez can be applied to real-world datasets such as the UCI Adult and COMPAS datasets.

Our work is most similar in nature to Taskesen et al. [23] in that we both forward statistical hypothesis tests for identifying bias or unfairness in classification tasks. We both compute some form of a closest plausible distribution (which they call the "most favorable distribution"), making both of our approaches distinct from other hypothesis testing frameworks such as those of DiCiccio et al. [11] and Tramer et al. [24]. However, one key difference between the two approaches is that their returned distribution is the closest distribution to the null hypothesis under which a classifier is considered fair, while ours is the closest distribution to the null which could plausibly explain the data. This gives the user a unique degree of interpretability as they are able to directly compare their null distribution, which could not plausibly explain the data, to the closest distribution which could. Furthermore, this ability to return a closest plausible distribution on top of providing an assessment of fairness within a dataset differentiates our method's from other hypothesis testing frameworks or simple correlation analyses which can only accomplish the latter.

In addition, we believe that our work provides some specific advantages over that of Taskesen et al.. First, their Wasserstein test analyzes the output of a trained model. This means that the test may not be able to isolate a specific source of bias, which is more often than not the training dataset itself [4]. In contrast, our hypothesis tests analyze the data directly, removing the need to train a classifier and eliminating some potential sources of bias in the ML workflow. This ensures that our methods avoid confounding bias in the training data and bias introduced by a specific learning algorithm. However, it should be noted that our test could still be applied to the output of a trained model, since we would still be able to set fairness standards for its behavior. This is a subject for future work. Second, since we are able to form rejection regions for possible hypotheses using the s lower bound condition of Equation 8, we allow for relaxed notions of fairness, or compound hypotheses, where a user might find a certain amount of bias acceptable, which the Wasserstein test is unable to do. To the authors' best knowledge, ours is the first hypothesis testing framework that can handle such hypotheses.

Our methods have other general advantages as well. First, no assumptions concerning the distribution of the sample points within the dataset are required to use our tests. Furthermore, the test statistic, $\kappa(x)$, does not need to fit any specific distribution (e.g., Student's t or chi-square). Second, unlike other tests which assume that the individual observations within a dataset are independently and identically distributed (i.i.d.), our methods can be straightforwardly adapted to test the probabilistic feasibility of explanations for non-i.i.d. data. Third, our tests are fast and efficient enough to run on consumer-grade personal computers. To illustrate this, we extrapolated the COMPAS dataset to 100,000 entries based on proportions in the original dataset, then tested the run time of our whole workflow for various numbers of outcome labels for a protected group on a 16-core AMD Ryzen 9 5900HS CPU. The binary label scenario ran in just over one second, and the worst-case scenario of 10 possible categories took about 20 minutes. Furthermore, when we extrapolated the COMPAS dataset to one million records and conducted our test using three categories (as per Section 6), the resulting run time was less than three minutes. Additional details and results can be found in the Appendix. Lastly, our methods greatly reduce the burden placed on a practitioner seeking to uncover bias in data. A user only needs to specify a significance level α and a distribution they deem to be roughly fair. In practice, this distribution would likely be simple for someone to provide as they are naturally generated when asking questions about bias in datasets and may be already evident based on certain social and political circumstances. For example, in the COMPAS dataset discussed in Section 7.2, if a practitioner was analyzing for biases in recidivism scores along racial lines, then they would probably choose the Caucasian distribution as the null hypothesis, wary that discriminatory policing may be reflected in models trained on the dataset. While the need to specify a roughly fair distribution may present a tough choice for the user in some cases, in general it provides a degree of freedom and removes the need for everyday users to possess in-depth knowledge about various inference models or fairness notions.

8 CONCLUSIONS

We forward a set of statistical hypothesis tests that use the *specified complexity kardis* and its level-α property to identify bias in ML training data, extending the work of Montañez [20]. Our tests allow us to construct probabilistic lower bounds, rule out whole sets of hypotheses, and return a closest plausible distribution giving clarity for users analyzing test results. Lastly, we demonstrate potential real-world applications of our methods by analyzing the UCI Adult and COMPAS datasets.

The authors acknowledge that any user of our tests must have their data in a convenient format, where protected attribute(s) are available and every entry has discrete attributes and labels. The datasets tested in Section 7 fit this mold, but many others do not. For example, our method as presented would be poorly suited to handle the task of identifying bias in training data for image classification tasks. It should also be noted that our $\nu(x)$ in Definition 4 explores frequential structure, but it could be validly defined to account for other or multiple notions of structure (e.g., *ordinal*). Extending our tests to other types of structure (such as ordinal structure, or structure in image data) is the subject of future work.

Furthermore, the authors emphasize that our tests should not be used to draw any societal or casual conclusions based on their results. A rejected test does not mean the process that generated the data was malicious but rather that it should be further investigated before any models are trained.

Lastly, just as traditional hypothesis tests are subject to manipulation of results via "p-hacking" [14], our tests are not immune to malicious intent. One could "SC-hack" by defining $\nu(x)$ such that it imparts a high specification value to the given observations. As such, it is essential that $\nu(x)$ be defined prior to observing x or at least not be conditioned on the observation x [20]. With this in mind, we note that the provided tools are powerful and therefore must be used with caution, as should all power tools.

ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation under Grant No. 1950885. Any opinions, findings, or conclusions are those of the authors alone, and do not necessarily reflect the views of the National Science Foundation. The authors thank Xanda Schofield, Erin Talvitie, Melissa O'Neill, and Lucas Bang for helpful feedback and suggestions for this work, and thank their friends, family, and the members of the AMISTAD Lab for their continued support.

REFERENCES

[1] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine Bias. ProPublica. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing Accessed 07/05/2021.

[2] Arthur Asuncion and David Newman. 2007. UCI Machine Learning Repository.

[3] Onur Avci, Osama Abdeljaber, Serkan Kiranyaz, Mohammed Hussein, Moncef Gabbouj, and Daniel J Inman. 2021. A Review of Vibration-Based Damage Detection in Civil Structures: From Traditional Methods to Machine Learning and Deep Learning Applications. *Mechanical systems and signal processing* 147 (2021), 107077.

[4] Agathe Balayn, Christoph Lofi, and Geert-Jan Houben. 2021. Managing Bias and Unfairness in Data for Decision Support: A Survey of Machine Learning and Data Engineering Approaches to Identify and Mitigate Bias and Unfairness within Data Management and Analytics Systems. *The VLDB Journal* (2021), 1–30.

[5] Matias Barenstein. 2019. ProPublica's COMPAS Data Revisited. *arXiv preprint arXiv:1906.04711* (2019).

[6] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. 2019. AI Fairness 360: An Extensible Toolkit for Detecting and Mitigating Algorithmic Bias. *IBM Journal of Research and Development* 63, 4/5 (2019), 4–1.

[7] Simon Caton and Christian Haas. 2020. Fairness in Machine Learning: A Survey. *arXiv preprint arXiv:2010.04053* (2020).

[8] L. Elisa Celis, Vijay Keswani, and Nisheeth Vishnoi. 2020. Data Preprocessing to Mitigate Bias: A Maximum Entropy Based Approach. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 1349–1359.

[9] Andrew Cotter, Heinrich Jiang, Maya R Gupta, Serena Wang, Taman Narayan, Seungil You, and Karthik Sridharan. 2019. Optimization with Non-Differentiable

Constraints with Applications to Fairness, Recall, Churn, and Other Goals. *J. Mach. Learn. Res.* 20, 172 (2019), 1–59.

[10] Daniel Andrés Díaz-Pachón, Juan Pablo Sáenz, and J Sunil Rao. 2020. Hypothesis Testing with Active Information. *Statistics & Probability Letters* 161 (2020), 108742.

[11] Cyrus DiCiccio, Sriram Vasudevan, Kinjal Basu, Krishnaram Kenthapadi, and Deepak Agarwal. 2020. Evaluating Fairness Using Permutation Tests. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1467–1477.

[12] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 259–268.

[13] Robert M Hazen, Patrick L Griffin, James M Carothers, and Jack W Szostak. 2007. Functional Information and the Emergence of Biocomplexity. *Proceedings of the National Academy of Sciences* 104, suppl 1 (2007), 8574–8581.

[14] Megan L Head, Luke Holman, Rob Lanfear, Andrew T Kahn, and Michael D Jennions. 2015. The Extent and Consequences of P-Hacking in Science. *PLoS biology* 13, 3 (2015), e1002106.

[15] Cynthia Hom, Amani Maina-Kilaas, Kevin Ginta, Cindy Lay, and George D Montañez. 2021. The Gopher's Gambit: Survival Advantages of Artifact-Based Intention Perception. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART*, Ana Paula Rocha, Luc Steels, and H. Jaap van den Herik (Eds.). INSTICC, SciTePress, 205–215. https://doi.org/10.5220/0010207502050215

[16] Heinrich Jiang and Ofir Nachum. 2020. Identifying and Correcting Label Bias in Machine Learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 702–712.

[17] Konstantina Kourou, Themis P Exarchos, Konstantinos P Exarchos, Michalis V Karamouzis, and Dimitrios I Fotiadis. 2015. Machine Learning Applications in Cancer Prognosis and Prediction. *Computational and structural biotechnology journal* 13 (2015), 8–17.

[18] Dieter Kraft. 1988. *A Software Package for Sequential Quadratic Programming*. Technical Report DFVLR-FB 88-28. DLR German Aerospace Center – Institute for Flight Mechanics, Koln, Germany.

[19] 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.

[20] George D. Montañez. 2018. A Unified Model of Complex Specified Information. *BIO-Complexity* 2018, 4 (2018).

[21] Razieh Nabi-Abdolyousefi et al. 2021. *Causal Inference Methods for Bias Correction in Data Analyses*. Ph. D. Dissertation. Johns Hopkins University.

[22] Pedro Saleiro, Benedict Kuester, Loren Hinkson, Jesse London, Abby Stevens, Ari Anisfeld, Kit T Rodolfa, and Rayid Ghani. 2018. Aequitas: A Bias and Fairness Audit Toolkit. *arXiv preprint arXiv:1811.05577* (2018).

[23] Bahar Taskesen, Jose Blanchet, Daniel Kuhn, and Viet Anh Nguyen. 2021. A Statistical Test for Probabilistic Fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 648–665.

[24] Florian Tramer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, Jean-Pierre Hubaux, Mathias Humbert, Ari Juels, and Huang Lin. 2017. Fairtest: Discovering Unwarranted Associations in Data-Driven Applications. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 401–416.

[25] Masatoshi Tsuchiya. 2018. Performance Impact Caused by Hidden Bias of Training Data for Recognizing Textual Entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

[26] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M Shieber. 2020. Investigating Gender Bias in Language Models Using Causal Mediation Analysis.. In *NeurIPS*.

[27] Lu Zhang, Yongkai Wu, and Xintao Wu. 2017. A Causal Framework for Discovering and Removing Direct and Indirect Discrimination. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 3929–3935.

[28] Lu Zhang, Yongkai Wu, and Xintao Wu. 2017. Achieving Non-Discrimination in Data Release. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1335–1344.

A CALCULATING $M_g(x)$

Recall that by Equation 2, for any event x with count vector $\mathbf{X}$ and mean count vector $\mathbb{E}[\mathbf{X}]$, its corresponding distance vectors is

$$\mathbf{D} = [D_1, D_2, \dots, D_m]$$

where each $D_i = X_i - \mathbb{E}[X_i]$. Furthermore, by Equation 3, its extremity is

$$g(x) = \sum_{i=1}^{m} |X_i - \mathbb{E}[X_i]| = \sum_{i=1}^{m} |D_i|.$$

There are a few key points to note about $\mathbf{D}$ and g. The first is that the components of $\mathbf{D}$ must sum to 0 because any positive distance from the mean in one component must be "balanced out" by negative distance in another. A consequence of this is that g can only take on non-negative, even values, since it is the sum of absolute values of the components of $\mathbf{D}$. Furthermore, this means that the sum of the non-negative and strictly negative components of $\mathbf{D}$ are equal in magnitude. Specifically, their magnitudes are both half the value of g. With these points in mind, there is a systematic way to compute $M_g(x)$, the number of sequences in the space $\mathcal{X}$ which are at least as extreme as x. Since each count vector matches to a unique distance vector (and vice versa), we can do so by finding the number of distance vectors $\mathbf{D}$ with corresponding g values greater than or equal to some initial value $g(x)$, the extremity of the initial observation. Clearly, there are many possible values for $\mathbf{D}$. Its components can be positive or negative, and each component can take on a variety of values. For now, we will focus on all the count vectors with values of $g = g(x)$, or all the count vectors *equally* as surprising as the initial observation $\mathbf{X}$.

As noted previously, the sum of the non-negative components of $\mathbf{D}$ must be $g(x)/2$, meaning that these components can range from 0 to $g(x)/2$. Similarly, the sum of the strictly negative components must be $-g(x)/2$. Unlike the positive components, however, the negative components have restrictions on their ranges. Since each $D_i = X_i - \mathbb{E}[X_i]$, the non-negativity of each X_i ensures that the largest negative value each D_i can take on is $-\mathbb{E}[X_i]$.

Treat the strictly negative distances in $\mathbf{D}$ as $g(x)/2$ indistinguishable red balls and the non-negative entries as $g(x)/2$ indistinguishable black balls. We need to place these balls into m distinct bins, corresponding to the m components of $\mathbf{D}$. We cannot have both red and black balls in the same bin. Note that any bin dedicated to red balls must have at least 1 ball while bins dedicated to black balls are allowed to be empty - zero is included in the set of non-negative options, and therefore is excluded as an option for the strictly negative bins.

First, we compute $\mathbf{S}$, the set of all possible bin combinations to allocate for red balls. $\mathbf{S}$ will be all combinations of bins where the number of bins can be 1 through $m - 1$ inclusive. We exclude bin combinations of all m bins because we must leave at least one bin for black balls. We then filter out the combinations in $\mathbf{S}$ whose total maximum capacity is less than $g(x)/2$, since any valid bin combinations for red balls must be able to hold the required number of balls. What remains is $\mathbf{S}'$, the set of all valid bin combinations to allocate for red balls. For each combination $s \in \mathbf{S}'$, we need to know how many ways there are to distribute the $g(x)/2$ red balls between them. The general formula for this is given by Theorem 3

Theorem 3. *Let $T := \mathcal{P}(\{1, \dots, n\})$ for some positive integer n, where $\mathcal{P}(\cdot)$ denotes the powerset operation. The number of combinations of n strictly positive integers $a_1, \dots, a_n$ such that $a_1 + \dots + a_n = N$*

and $a_i \leq r_i$ for each $i = 1, \ldots, n$ is given by

$$\sum_{k=0}^{n} \left(\sum_{S \in T : |S| = k} (-1)^k \binom{N - \sum_{i \in S} r_i - 1}{n-1} \right).$$

We make use of this theorem by letting N be the number of balls, n be the number of bins, and r_i be the maximum negative value for D_i, corresponding to the maximum capacity of the bin. For each combination of bins dedicated to red balls $s \in \mathbf{S}'$, there is a corresponding set of bins which we must distribute the $g(x)/2$ black balls between. The general formula for this, colloquially known as the "Stars and Bars" formula, is given by Lemma 2

Lemma 2. *The number of combinations of n non-negative integers $a_1, \ldots, a_n$ such that $a_1 + \ldots + a_n = N$ is given by*

$$\binom{N + n - 1}{n - 1}.$$

We make use of the lemma by letting N be the number of black balls and n be the number of bins for black balls. Multiplying the results of Theorem 3 and Lemma 2 for a given $s \in \mathbf{S}'$ and summing over $\mathbf{S}'$ gives us the number of count vectors with $g = g(x)$. Repeating this process and summing for all possible[2] $g \geq g(x)$ will give us $M_g(x)$, the number of count vectors which are at least as surprising as the observation $\mathbf{X}$. Note that the maximum g is obtained when the least likely possible outcome i is the only observed outcome. That is, $X_i = n$ and all other $X_j = 0$. The pseudocode for the entire $M_g(x)$ calculator is shown in Algorithm 1.

Algorithm 1 $M_g(x)$ Calculator

```
Initialize $M_g(x) \leftarrow 0$
for all $i = g(x), g(x) + 2, \ldots, \max(g)$ do
    if $i = 0$ then
        $M_g(x) \leftarrow M_g(x) + 1$
    else
        for all $s \in \mathrm{S}'$ do
            $r \leftarrow$ result from Theorem 3
            $b \leftarrow$ result from Lemma 2
            $M_g(x) \leftarrow M_g(x) + r \cdot b$
        end for
    end if
end for
```

B RUN TIME EXPERIMENTS

The COMPAS dataset contains decile risk scores for each entry, with higher scores indicating that a person is more likely to recidivate. We used different groupings of these scores to artificially create a desired amount of labels for our tests. For example, for the three label tests, we grouped scores 1-3, 4-6, and 7-10 together. Our first step was taking the proportion of each decile score for African-Americans in the original dataset and using those to generate a dataset with 100,000 entries by uniformly sampling with replacement. Then, for a desired number of labels k, we relabeled the decile scores based on which score group they belonged to. This transformed the 100,000 entry dataset with 10 labels, into one with k labels. Lastly, using the same k labels, we took the proportion of Caucasians in the original dataset belonging to each score group and used it to construct a null hypothesis and run our FSC test 10 times. We repeated this process for $k = 2, \ldots, 10$ labels and constructed 95% confidence intervals for run time, which are shown in Table 3.

Table 3: FSC hypothesis test 95% confidence intervals across 10 trials for extrapolated COMPAS dataset

Number of Labels	Run Time 95% CI
2	(1.31, 1.41) s
3	(1.39, 1.43) s
4	(2.00, 2.10) s
5	(4.34, 4.46) s
6	(12.20, 12.54) s
7	(37.94, 39.42) s
8	(2.05, 2.12) m
9	(6.09, 7.27) m
10	(19.69, 25.96) m

We also used the proportion of each score text category for African-Americans provided in the original COMPAS dataset (high, medium, and low) to similarly generate a dataset with 1,000,000 entries. Using the proportions of each score text category for Caucasians in the original dataset as our null hypothesis, we conducted a test on the extrapolated dataset. This is similar to the first COMPAS experiment described in Section 6. Recording the run time for this test and repeating for 10 trials resulted in a 95% confidence interval of (2.33, 3.27) minutes.

C PROOFS

Lemma 1. *If $\mathcal{X}$ is the space of all possible outcomes of $X \sim Bin(n, p)$, then*

$$\sum_{x \in \mathcal{X}} \binom{n}{k_x}^{-1} = n + 1$$

where k_x is the number of successes in event x.

Proof.

$$\sum_{x \in \mathcal{X}} \binom{n}{k_x}^{-1} = \sum_{k=0}^{n} \left(\sum_{x : k_x = k} \binom{n}{k}^{-1} \right)$$
$$= \sum_{k=0}^{n} \binom{n}{k}^{-1} \left(\sum_{x : k_x = k} 1 \right).$$

[2] Note that if $g(x) = 0$, there is only one count vector corresponding to it, the observation $\mathbf{X}$ itself. Thus, the process described above is not necessary for this singular g value.

For a given number of successes k, there are n choose k events x with $k_x = k$ successes. Thus,

$$\begin{aligned}\sum_{k=0}^{n} \binom{n}{k}^{-1} \left(\sum_{x:k_x=k} 1 \right) &= \sum_{k=0}^{n} \binom{n}{k}^{-1} \binom{n}{k} \\ &= \sum_{k=0}^{n} 1 \\ &= n + 1.\end{aligned}$$

It follows that,

$$\sum_{x \in \mathcal{X}} \binom{n}{k_x}^{-1} = n + 1,$$

as desired. □

Lemma 2. *The number of combinations of n non-negative integers $a_1, \ldots, a_n$ such that $a_1 + \ldots + a_n = N$ is given by*

$$\binom{N + n - 1}{n - 1}.$$

Proof. This problem is colloquially known as "Stars and Bars" because we imagine it as laying out N stars on a table in a line and using $n - 1$ bars to divide them into n groups. If two bars are placed next to each other, then the group represented by the area between them is considered to be empty. If a bar is placed on the very left edge of the line of stars, the empty area to its left represents and empty group, and similarly for the right edge. As such, there are $N + n - 1$ spots for us to place our $n - 1$ bars. The number of ways to place the bars is therefore

$$\binom{N + n - 1}{n - 1}.$$

□

Theorem 2. *For a random variable $X \sim Bin(n, p)$ and number of successes k_x, a BSC hypothesis test will return a smaller tail probability bound than an FSC hypothesis test for all*

$$np - c < k_x < np + c$$

where

$$c = \frac{n}{2} - \frac{n + 1}{2 + 2\ln(n + 1))} + 1.$$

Proof. From Definition 5, we have

$$\kappa_{BSC}(x) = (n + 1) \Pr(X = k_x),$$

and from Equation 14,

$$\kappa_{FSC}(x) = (n - 2|k_x - np| + 2)(1 + \ln(n + 1)) \Pr(X = k_x).$$

We can now construct the inequality $\kappa_{BSC}(x) < \kappa_{FSC}(x)$, fix n and p and solve for k_x:

$$\begin{aligned}\kappa_{BSC}(x) &< \kappa_{FSC}(x) \\ (n + 1) \Pr(X = k_x) &< (n - 2|k_x - np| + 2)(1 + \ln(n + 1)) \Pr(X = k_x) \\ n + 1 &< (n - 2|k_x - np| + 2)(1 + \ln(n + 1)) \\ \frac{n + 1}{1 + \ln(n + 1)} &< n - 2|k_x - np| + 2 \\ |k_x - np| &< \frac{n}{2} - \frac{n + 1}{2 + 2\ln(n + 1)} + 1.\end{aligned}$$

For succinctness, let

$$c := \frac{n}{2} - \frac{n + 1}{2 + 2\ln(n + 1)} + 1.$$

Then,

$$|k_x - np| < c \Rightarrow np - c < k_x < np + c.$$

□

Theorem 3. *Let $T := \mathcal{P}(\{1, \ldots, n\})$ for some positive integer n, where $\mathcal{P}(\cdot)$ denotes the powerset operation. The number of combinations of n strictly positive integers $a_1, \ldots, a_n$ such that $a_1 + \ldots + a_n = N$ and $a_i \leq r_i$ for each $i = 1, \ldots, n$ is given by*

$$\sum_{k=0}^{n} \left(\sum_{S \in T:|S|=k} (-1)^k \binom{N - \sum_{i \in S} r_i - 1}{n - 1} \right).$$

Proof. This problem is the same as asking how many ways there are to sort N indistinguishable balls into n distinct bins, where each bin $i = 1, \ldots, n$ can hold at most r_i balls and no empty bins are allowed. To ensure no empty bins, we put 1 ball into each bin, leaving us with $N - n$ balls to sort into n bins. We will now use the Inclusion-Exclusion principle to count our integer partitions. First, we will count how many total ways there are to sort $N - n$ balls into n bins, with no regards to the bins' limits, which is given by Lemma 2

$$\binom{(N - n) + n - 1}{n - 1} = \binom{N - 1}{n - 1}.$$

Next, we must subtract the cases where there is a size violation in at least one bin. That is, we must subtract the cases where bin i is designated more balls than its specified capacity r_i. To guarantee a violation in bin i, we will put r_i balls in it, remembering that we already put 1 ball in bin i to guarantee it was not empty. Then, we sort the remaining $N - n - r_i$ balls into n bins. The number of ways to do so is given again by Lemma 2

$$\binom{(N - n - r_i) + n - 1}{n - 1} = \binom{N - r_i - 1}{n - 1}.$$

For all possible choices of one bin i, the total number of cases where there is a violation in at least one bin is therefore

$$\sum_{i=1}^{n} \binom{N - r_i - 1}{n - 1}.$$

Which can be conveniently rewritten as

$$\sum_{S \in T:|S|=1} \binom{N - \sum_{i \in S} r_i - 1}{n - 1}$$

where T is the powerset of $\{1, \ldots, n\}$. This notation change is valid since each element $S \in T$ whose cardinality is 1 represents a unique choice of one bin i. While this notation may seem unnecessary, it will become useful in constructing our final formula. At this point, the total number of "good" cases is

$$\binom{N - 1}{n - 1} - \sum_{S \in T:|S|=1} \binom{N - \sum_{i \in S} r_i - 1}{n - 1}.$$

However, since we have subtracted the number of cases where there is a violation in *at least one* bin, a case where there was a violation in bin i *and* bin j would have subtracted out twice by the above summation. Thus, we must add back all the cases where there is a violation in *at least* two bins. For each pair of bins $i, j = 1, \ldots, n$ where $i \neq j$, we place r_i balls in i and r_j balls in j to guarantee violations in both. Then, we sort the remaining $N - n - r_i - r_j$ balls into n bins using Lemma 2

$$\binom{(N - n - r_i - r_j) + n - 1}{n - 1} = \binom{N - r_i - r_j - 1}{n - 1}.$$

For all choices of i and j, the total number of cases where there is a violation in at least two bins is

$$\sum_{S \in T : |S| = 2} \binom{N - \sum_{i \in S} r_i - 1}{n - 1}$$

which we must add back to our previous value. Thus, the new number of good cases is

$$\binom{N - 1}{n - 1} - \sum_{S \in T : |S| = 1} \binom{N - \sum_{i \in S} r_i - 1}{n - 1} + \sum_{S \in T : |S| = 2} \binom{N - \sum_{i \in S} r_i - 1}{n - 1}.$$

A pattern is beginning to take shape, but for the sake of demonstration, we will continue for one more iteration. Consider the cases where there is a size violation in 3 bins, i, j, and k. These cases would have been subtracted out 3 times when we considered all cases with violations in i, j, and k individually, but also added back in 3 times when we considered all the cases with violations in i and j, i and k, and j and k. These cancel out, and we have therefore not taken into account any of the cases where there are violations in all three bins i, j, and k. In order to count these, we will proceed as before. To guarantee violations in all 3 bins, we place r_i balls in i, r_j balls in j, and r_k balls in k, leaving us $N - n - r_i - r_j - r_k$ balls to sort between n bins. Lemma 2 yields

$$\binom{(N - n - r_i - r_j - r_k) + n - 1}{n - 1} = \binom{N - r_i - r_j - r_k - 1}{n - 1}.$$

For all choices of i, j, and k, the total number of cases where there is a violation in at least three bins is

$$\sum_{S \in T : |S| = 3} \binom{N - \sum_{i \in S} r_i - 1}{n - 1}$$

which we subtract from the total number of possible cases so far

$$\binom{N - 1}{n - 1} - \sum_{S \in T : |S| = 1} \binom{N - \sum_{i \in S} r_i - 1}{n - 1} + \sum_{S \in T : |S| = 2} \binom{N - \sum_{i \in S} r_i - 1}{n - 1} - \sum_{S \in T : |S| = 3} \binom{N - \sum_{i \in S} r_i - 1}{n - 1}.$$

We continue this pattern of adding and subtracting bad cases where there are violations in x bins for $x = 1, \ldots, n$, which will yield us the final formula

$$\sum_{k=0}^{n} \left(\sum_{S \in T : |S| = k} (-1)^k \binom{N - \sum_{i \in S} r_i - 1}{n - 1} \right).$$

□