AUDITING FAIRNESS ONLINE THROUGH INTERACTIVE REFINEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine learning algorithms are increasingly being deployed for high-stakes scenarios. A sizeable proportion of currently deployed models make their decisions in a black box manner. Such decision-making procedures are susceptible to intrinsic biases, which has led to a call for accountability in deployed decision systems. In this work, we focus on user-specified accountability of decision-making processes of black box systems. Previous work has formulated this problem as run time fairness monitoring over decision functions. However, formulating appropriate specifications for situation-appropriate fairness metrics is challenging. We construct AVOIR, an automated inference-based optimization system that improves bounds for and generalizes prior work across a wide range of fairness metrics. AVOIR offers an interactive and iterative process for exploring fairness violations aligned with governance and regulatory requirements. Our bounds improve over previous probabilistic guarantees for such fairness grammars in online settings. We also construct a novel visualization mechanism that can be used to investigate the context of reported fairness violations and guide users towards meaningful and compliant fairness specifications. We then conduct case studies with fairness metrics on three different datasets and demonstrate how the visualization and improved optimization can detect fairness violations more efficiently and ameliorate the issues with faulty fairness metric design.

1 INTRODUCTION

The use of advanced analytics and artificial intelligence (AI), along with its many benefits, poses important threats to individuals and broader society at large. Hirsch et al. (2020) identify: invasion of privacy; manipulation of vulnerabilities; bias against protected classes; increased power imbalances; error; opacity and procedural unfairness; displacement of labor; pressure to conform, and intentional and harmful use as some of the key areas of concern. A core part of the solution to mitigate such risks is the need to make organizations accountable and ensure that the data they leverage and the models they build and use are both inclusive of marginalized groups and resilient against societal bias. Deployed AI and analytic systems are complex multi-step processes that can produce several sources of risk at each step. To mitigate such risks, organizations need to be able to audit the algorithms to ensure that they are consistent with stated data ethics policies and regulatory requirements.

Governments across the world are wrestling with the implementation of auditing regulation and practices for increasing the accountability of decision processes. Recent examples include the New York City auditing requirements for AI hiring tools (Vanderford, 2022), European data regulation (GDPR 2018), accountability bills 2019; 2021 and judicial reports 2018. In an evolving environment, organizations also aim to augment their understanding of algorithmic audits. These societal forces have led to the emergence of checklists (Mitchell et al., 2019; Sokol & Flach, 2020), metrics of fairness (Verma & Rubin, 2018), and recently, algorithms and systems that observe and audits the behavior of AI algorithms. Such ideas date back to the 1950s (Moore, 1956) but research has largely been sporadic until very recently with the widespread use of AI-based decision making giving rise to the vision of algorithmic auditing (Galdon Clavell et al., 2020).

Machine learning testing (Zhang et al., 2020) is an avenue that can be used to expose undesired behavior and improve the trustworthiness of machine learning systems. Compared with traditional testing systems, machine learning testing faces fundamental differences. First, the behavior of machine

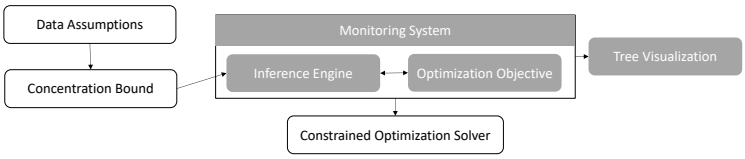


Figure 1: Shaded nodes describe our contributions to create the AVOIR framework.

Tool	Flexible Fairness Metrics	Adaptive Optimization	Online Monitoring	Visual Refinement	Black-Box Model
Albarghouthi & Vinitsky (2019)	✓	✗	✓	✗	✓
Bellamy et al. (2019)	✓	✗	✗	✗	✓
Bastani et al. (2019)	✓	✓	✓	✗	✓
Ghosh et al. (2021b)	✗	✗	✓	✗	✓
Ginart et al. (2022)	✗	✓	✓	✗	✓
Ghosh et al. (2021a)	✓	✓	✗	✗	✗
AVOIR (Our work)	✓	✓	✓	✓	✓

Table 1: Comparison of AVOIR with prior work.

learning systems may change based on the domain and distribution shift of the input training data. Second, machine learning is trained with the intention of generalizing to unobserved data, which is often at odds with general-purpose testing Zhang et al. (2020). Several works aim to verify the fairness of machine learning systems. Given a specific definition of fairness, Fairtest (Tramèr et al., 2017) and Verifair (Bastani et al., 2019) build a comprehensive framework for investigating fairness in data-driven pipelines. Fairness Testing provides a notion of causal fairness and generates tests to check the fairness of a given decision-making procedure (Galhotra et al., 2017). Fairness-aware programming Albarghouthi & Vinitsky (2019) combined the two demands of machine learning testing and fairness auditing to make fairness a first-class concern in programming. Fairness-aware programming applies a runtime monitoring system for a decision-making procedure with respect to an initially stated fairness specification.

Alternative frameworks such as the AI Fairness 360 (Bellamy et al., 2019) provide mechanisms to quantify fairness uncertainty for pre-supported metrics. Uncertainty quantification (Ghosh et al., 2021b; Ginart et al., 2022) provides adaptive guarantees, however, existing work is designed for commonly used outcome metrics such as accuracy, F1-score, etc., rather than for fairness metrics. Justicia (Ghosh et al., 2021a) optimizes uncertainity for fairness metrics estimates using stochaistic SAT solvers but can only be applied to a limited class of tree-based classification algorithms.

We present a framework for *Auditing and Verifying fairness Online through Interactive Refinement* (AVOIR) [1]. AVOIR builds upon the ideas on distributional probabilistic fairness guarantees (Albarghouthi & Vinitsky, 2019; Bastani et al., 2019), generalizing them to real-world data. Prior work has mainly focused on data generated from known distributions. The extensions that we propose in AVOIR allow for the generation of probabilistic guarantees for arbitrary, possibly unknown, decision functions. Further, AVOIR leverages a novel tree-based optimization and visualization that enables a user to choose an a suitable fairness specification that can be guaranteed to make more efficient estimates than prior work. This visualization and optimization is derived using the intrinsic structure of the fairness specification grammar supported by AVOIR. Figure 1 shows a summary of the framework, and Table 1 provides a summary of the comparison of AVOIR to related prior work.

2 Overview

2.1 Fairness Criteria

Decision making/scoring functions aim to optimize for a specific metric such as accuracy, precision, F1 score etc. Fairness criteria, on the other hand, quantify the relationship between the outcome metric across multiple subgroups or similar individuals among the population. Formal definitions of

[1] AVOIR in French means "to have" and this acronym reflects both our aspirational goal to achieve fairness in advanced analytics and AI but also reflects what is currently verifiable given a dataset, a model and a fairness specification.

fairness focus on observational criteria, i.e., those that can be written down as a probability statement involving the joint distribution of the features, sensitive attributes, decision making function, and actual outcome. For example, a decision making function that selects amongst applicants to hire for a specific position may have different hiring rates for majority and minority population groups. Suppose r denotes the return value of the decision function, and s is an indicator denoting whether a candidate belongs to a minority population. An example fairness criterion is the 80%-rule (Zafar et al., 2017) states that

$$\min\left\{\frac{\Pr[r|s]}{\Pr[r|\neg s]}, \frac{\Pr[r|\neg s]}{\Pr[r|s]}\right\} \geq 0.8$$

In our work, we focus on fairness criteria that can be expressed using Bernoulli random variables. This covers a majority of popular fairness criteria (Verma & Rubin, 2018). Section 4 contains descriptive analyses and case studies of real world applications of AVOIR on commonly used fairness criteria.

2.2 Fairness Specifications

We aim to derive statistical guarantees about fairness criteria based on observed outputs. These guarantees are derived in the form of assertions over specifications. For a given specification ψ, we denote the claim that $P[\psi = F] \geq 1 - \delta$ as $\psi : (F, \delta)$, where δ denotes the failure probability of a guarantee. Similar assertions can be made about observed and tracked variables. Specifically, let X be an observed Bernoulli r.v[2]

$$\phi_X := X : (\overline{\mathbb{E}}[X], \epsilon, \delta) \equiv \Pr[|\mathbb{E}[X] - \overline{\mathbb{E}}[X]| \geq \epsilon] \leq \delta \tag{1}$$

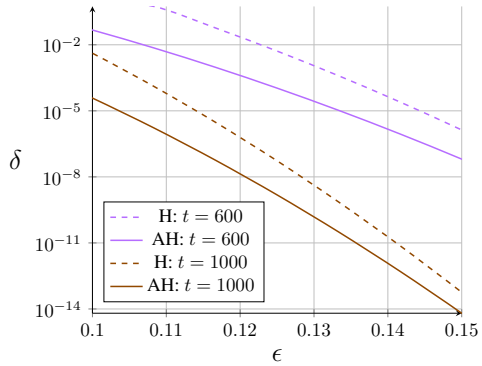
where $\overline{\mathbb{E}}[X]$ denotes an empirical estimate of $E[X]$. Given a stream of (observations, outcomes from the decision functions), and a specified threshold probability δ, AVOIR will continue to refine the estimate for a given specification until reaching the failure threshold.

2.3 Contributions

2.3.1 Probabilistic Guarantees: Optimization for Concentration Bounds

When monitoring the value of a fairness metric in real-world scenarios, a system makes sequential decisions based on observed outcomes. Previous work on verifying fairness improve bounds using better concentration from union bounding Hoeffding inequality (Albarghouthi et al., 2017; Albarghouthi & Vinitsky, 2019) to the Adaptive Hoeffding inequality (Zhao et al., 2016) used by Bastani et al. (2019)) to reduce the number of samples required to generate low failure probability estimates

[2]random variable



(a) At the same concentration ϵ, lower failure probability δ for the majority class.

$\langle$*spec*$\rangle$ ::= $\langle$*ETerm*$\rangle$ $\langle$*comp-op*$\rangle$ c
| $\langle$*spec*$\rangle \wedge \langle$*spec*$\rangle$
| $\langle$*spec*$\rangle \vee \langle$*spec*$\rangle$

$\langle$*ETerm*$\rangle$ ::= $\mathbb{E}[\langle E\rangle]$
| $\mathbb{E}[\langle E\rangle|\langle E\rangle]$
| c $\in \mathbb{R}$
| $\langle$*ETerm*$\rangle$ $\{+, -, \times, \div\}$ $\langle$*ETerm*$\rangle$

(b) $\langle E\rangle$ refers to pure expressions and $\langle comp-op\rangle$ is any arbitrary comparison operator $\in \{>, <, =, \neq\}$.

Figure 2: *(Left)* Failure probability of Bernoulli r.v. being concentrated around its mean for different n. H = (online) Hoeffding, AH = Adaptive Hoeffding. *(Right)* Modified Grammar for Specification.

for a specification. The overall failure probability of an assertion is computed as the sum of failure probabilities of each constituting sub-expression (using the union bound). Proving guarantees for overall uncertainty across multiple groups involves balancing it across groups with differences in the number of observed samples. We derive a method for computing improved probabilistic bounds on point estimates of a model's adherence to a specification, in *provably fewer* iterations in the *online setting*. Prior work does not adequately optimize the union-bounding procedure, leading to weaker bounds. Specifically, they allocate equal failure likelihoods among all sub-expressions of a given expression. This is inefficient as fairness criteria typically involve estimating sub-expressions for terms having different numbers of observations (minority and majority groups). Such an estimator can tolerate a higher failure probability with lower change in the confidence interval. For example, consider Bernoulli r.vs $X_{1,2}$ for which we derive concentration guarantees $\Pr[|\mathbb{E}[X_i] - \overline{\mathbb{E}}[X_i]| \geq \epsilon_i] \leq \delta_i$ after t_i observations. From the Hoeffding inequality, $\delta = 2e^{-2t\epsilon^2}$. We can claim stronger guarantees for X_2 if $t_2 > t_1$ as the failure probability is lower at the same concentration. This observation enables us optimize over sub-expressions to provide stronger overall concentration for compound expressions. Adaptive versions of these inequalities also have similar patterns (see Figure 2a). Section 3 provides further details on how this allocation can be optimized.

2.3.2 Inference Engine for Arbitrary Metrics

One mechanism to support monitoring is to construct a custom domain specific language (DSL, see Figure 2b) parser and perform edit operations on the extracted syntax tree. Our implementation takes a different route - we implemented the entire DSL using python. This provides an easy mechanism to integrate AVOIR into existing python ML pipelines. We build our framework as a library for specifying fairness criteria as decorators over python functions. That is, each non-terminal expressible in the grammar corresponds to a unique python object. We construct classes to represent each of these objects and overload the appropriate operators such that any statement in the python version of the DSL corresponds to a unique statement in the modified grammar. The classes themselves contain code to carry out the inference necessary to compute the failure probabilities and optimization setup required for any fairness metric implemented in the grammar. Assuming that some concentration bound on the sum/mean of the decision functions is available, AVOIR supports tracking for any group fairness metric. We demonstrate AVOIR using the Adaptive Hoeffding bound (Zhao et al., 2016) which applies to sub-gaussian random variables. Further, we will open-source our implementation. Prior work either does not have open-source implementations available (Albarghouthi & Vinitsky, 2019) or the available implementations only provide examples for specific metrics (Bastani et al., 2019) and require tedious manual optimization and implementation efforts to generalize.

2.3.3 Visualization as a Tool for interactive refinement

There are a plethora of fairness criteria and subtle changes in their definition can change the implications on decision making (Castelnovo et al., 2021). Thus, practitioners need support when selecting, designing, and guaranteeing fairness for deployed machine learning algorithms. We develop an application for visually analyzing adherence to a specification for any data/model. The application uses a novel visualization to allow users to understand how specification violations arise by using the context from surrounding violations.

3 AVOIR Framework

3.1 Language Specification

We describe AVOIR's Domain Specific Language (DSL) used for specifying fairness metrics. Concrete examples of implemented specifications in AVOIR's DSL are provided in Section 4. We focus on binary decision making functions; their outputs can be characterized by Bernoulli r.v.s. Note that for such a Bernoulli r.v. X, $\mathbb{E}[X] = \Pr[X = 1]$ and hereafter, these are used interchangeably. We start with the grammar used by prior work and enhance the grammar to simplify the expressions used for common fairness specifications. Figure 2b describes the full grammar. We modified the grammar to include two additional operations. First, we added a `given` argument to the expectation term, which allows a user to specify conditional probabilities directly, in contrast to specifying it as

a ratio of joint/marginal probabilities.

$$\frac{\mathbb{E}(A \vee (B = b))}{\mathbb{E}(B = b)} \rightarrow \mathbb{E}(A, \texttt{given} = (B = b))$$

which is used to represent $\mathbb{E}[A|B = b]$, simplifying expressions used for group fairness specification. Additionally, we add binary comparison operators $<, >, ==, ! =$, which further simplifies the process of writing specifications.

3.2 Propagating Bounds

Generating the bounds for a specification requires propagating guarantees from elementary subexpressions. Assuming that observed values for each `ETerm` correspond to an underlying random variable X, A probabilistic guarantee ϕ_X consists of an empirical estimate $\overline{\mathbb{E}}[X]$, a concentration bound ϵ_X, and a failure probability δ_X, such that $\Pr[|\mathbb{E}[X] - \overline{\mathbb{E}}[X]| \geq \epsilon_X] \leq \delta_X$. We refer to expressions of this form as *elementary* subexpressions. A fairness specification will typically consist of multiple such elementary expressions, denoted as *compound* expressions. For compound expressions, we must infer the implied guarantees that can be provided, with corresponding constraints. Each inference rule corresponds to a derivation in the DSL grammar. Inference rules have preconditions and postconditions that follow the general expression

$$\frac{\bigcup\{r|r \in \{\phi, \psi, C\}\}}{\bigcup\{s|s \in \{\phi, \psi, C\}\}}$$

where ϕ denotes a claim for a subexpression, ψ for a `<spec>`, $\overline{\mathbb{E}}$ and ϵ are the mean and concentration terms associated with a subexpression claim, C denotes a constraint. For example, consider starting with $X : (\overline{\mathbb{E}}[X], \epsilon_X, \delta_X)$, $Y : (\overline{\mathbb{E}}[Y], \epsilon_Y, \delta_Y)$

$$\begin{aligned}|\mathbb{E}[X] \pm \mathbb{E}[Y] - (\overline{\mathbb{E}}[X] \pm \overline{\mathbb{E}}[Y])| &= |(\mathbb{E}[X] - \overline{\mathbb{E}}[X]) \pm (\mathbb{E}[Y] - \overline{\mathbb{E}}[Y])| \\ &\leq |\mathbb{E}[X] - \overline{\mathbb{E}}[X]| + |\mathbb{E}[Y] - \overline{\mathbb{E}}[Y]| \\ &\leq \epsilon_X + \epsilon_Y\end{aligned}$$

i.e., $X \pm Y : \big(\overline{\mathbb{E}}[X] \pm \overline{\mathbb{E}}[Y], \epsilon_X + \epsilon_Y, \delta_X + \delta_Y\big)$. Some derivations also lead to rules that require constraints. For instance, assume $X : (\overline{\mathbb{E}}[X], \epsilon_X, \delta_X), \overline{\mathbb{E}}[X] > c$. Then we have $\Pr[X < \overline{\mathbb{E}}[X] - \epsilon_X] > 1 - \delta$ If we add the constraint that $\overline{\mathbb{E}}[X] - \epsilon_X \geq c$, we have $\Pr[X < c] > 1 - \delta$, thus,

$$X : (\overline{\mathbb{E}}[X], \epsilon_X, \delta_X) \implies \psi \equiv X > c : (T, \delta_X)$$
$$\text{under the constraint } \{\overline{\mathbb{E}}[X] - \epsilon_X \geq c\}$$

The full set of inference rules required for the DSL is provided in the appendix (Figure 5). The implementation in AVOIR follows these rules but can be extended to other rule inference templates that support the DSL. We note that these rules extend the ones implemented by VeriFair (VF)[3] (Bastani et al., 2019) with constraints that enable the optimizations required in AVOIR (see Appendix A).

3.3 Optimizing Bounds

3.3.1 AVOIR Algorithm

The pseudocode for the optimization procedure in AVOIR is described in the appendix (Algorithm 1). The input to the algorithm is the reporting threshold probability Δ and a specification ψ. We then infer a symbolic optimization problem is inferred corresponding to the failure probabilities and constraints derived from concentration bounds. At each step, the `OBSERVE(X)` function is called with new observation of every *elementary* subexpression and observed output. The running mean and counts of observations are updated. The final optimization problem `OPT` corresponding to each specification is a nonlinear constrained optimization problem. We use the COIN-OR implementation of IPOPT (Wächter & Biegler, 2006), accessed though the Pyomo (Hart et al., 2011) interface to solve this problem at each step. If a solution is successfully found for `OPT`, the algorithm terminates, with the estimate for the specification having reached the required threshold. If no

[3] Verifair

solution is found, the estimates continue to be updated with $\delta_i = \Delta$ for each *elementary* subexpression. The main intuition behind the algorithm is to create a confidence sequence corresponding to the estimates at each time step. The OPT corresponding to a specification:

$$\begin{aligned} &\min_{\delta_i} \sum_{i=1}^{n} \delta_i \\ &\text{s.t. } g_k(\delta_{1,\ldots,n}, \overline{\mathbb{E}}[X_1], \ldots, \overline{\mathbb{E}}[X_n]) \leq \epsilon_k \\ &\quad 0 \leq \delta_i \leq 1 \end{aligned} \tag{2}$$

where g_k and ϵ_k are the functions/bounds derived using the transformations carried out through the DSL inference rules (further details in Appendix A.2).

Definition 1. *For $\delta \in (0,1)$, a $(1-\delta)$ confidence sequence is a sequence of confidence sets, usually intervals $(\mathrm{CI_t})_{\mathrm{t=1}}^{\infty}$,, say $\mathrm{CI_t} := (\mathrm{L_t}, \mathrm{R_t}) \subseteq \mathbb{R}$ satisfying a uniform convergence guarantee. After observing the tth unit, we calculate an updated confidence set $\mathrm{CI_t}$ for an unknown quantity of interest θ_t with the coverage property $\Pr(\forall t \geq 1, \theta_t : \theta_t \in \mathrm{CI_t}) \geq 1 - \delta$ (Howard et al., 2021).*

In this paper, we focus on the mean of r.v.s $\mathbb{E}[X]$ that constitute estimates for *elementary* subexpressions as the quantities of interest. We use adaptive concentration inequalities to construct these confidence sequences. Any adaptive concentration inequality that can be applied to a r.v. $X \in \{0,1\}$ such that

$$\Pr[|\overline{\mathbb{E}}_t[X] - \mathbb{E}[X]| \geq \epsilon(t,\delta)] \leq \delta \tag{3}$$

can be used in AVOIR. Here, $\overline{\mathbb{E}}_t[X]$ denotes the empirical estimate of $\mathbb{E}[X]$ after the tth observation. For the purpose of comparison with previous work (eg., VF), we use the Adaptive Hoeffding Inequality (Zhao et al., 2016), which will be referred to as AIN hereafter.

Theorem 1. *The sequence of estimates generated by AVOIR form a confidence set.*

The proof follows from the fact that AVOIR always estimates using a failure probability higher than that which is provided by AIN, and hence applying a union bound ensures that the estimates are a confidence set. The full proof is provided in Appendix C.

Corollary 1.1. *The estimates for the overall specification ψ form a confidence sequence converging to $\psi : (b, \Delta), b \in \{T, F\}$.*

Proof. We initialize the main specification with the required failure probability Δ. The termination condition requires $\sum \delta_i \leq \Delta$. From Theorem 1 we can infer that the confidence sequence corresponding to the termination achieves the required threshold Δ, and therefore, is valid. □

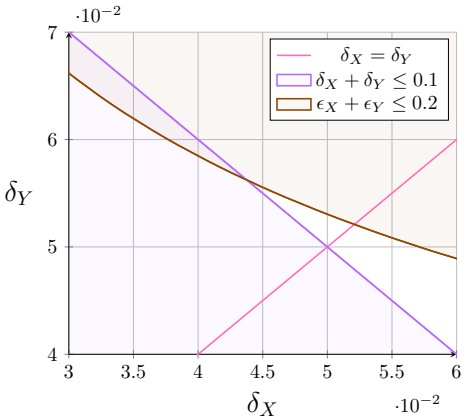
3.3.2 Improvements over Baseline

Concrete Example Consider a Bernoulli r.v R corresponding to the output of a binary decision function, with s being an indicator of class membership. Let $X = r \vee s$ and $Y = r \vee \neg s$ be r.vs corresponding to a positive decision for the majority and minority classes, respectively. Suppose we aim to estimate $\psi := X - Y < \epsilon_T$

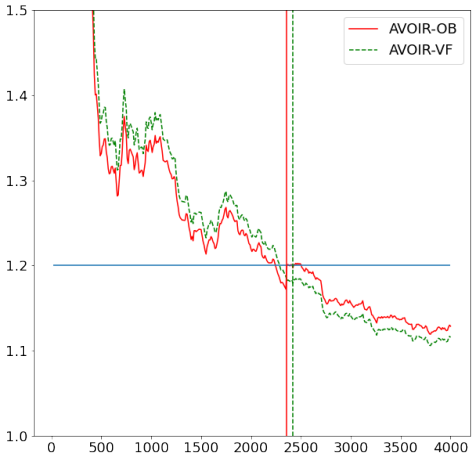
We demonstrate the improvements possible using our approach by instantiating this example with data. Suppose we want the upper bound of the failure probability $\Delta = 0.1$ for the specification. Consider a set of observations such that $\overline{\mathbb{E}}[X] = 0.8, n_X = 1550$ and $\overline{\mathbb{E}}[Y] = 0.5, n_Y = 310$. Figure 3a shows that no solution is feasible for the optimization problem with A_δ. However, AVOIR can find a solution. For the optimal solution, $\delta_2 \approx 2.35\delta_1$, which aligns with our intuition from section 2.3.1 about allocating higher failure probability to terms with the majority of observations. The optimization problem inferred by AVOIR:

$$\begin{aligned} &\min_{\delta_X, \delta_Y} \delta_X + \delta_Y \\ &\text{s.t. } \epsilon_X + \epsilon_Y \leq \overline{\mathbb{E}}[X] - \overline{\mathbb{E}}[Y] - \epsilon_T \\ &\quad 0 \leq \delta_{X,Y} \leq 1 \end{aligned} \tag{4}$$

Definition 2. *We define the specification stopping time $\mathcal{T}$ for a confidence sequence as the smallest time t such that, given a threshold Δ and a specification ψ, we can terminate any inference algorithm to claim that $\Pr[\forall t \geq 1, \psi_t = \widehat{\psi}_{\mathcal{T}}] \geq 1 - \Delta$, where $\widehat{\psi}_{\mathcal{T}}$ is the estimate of ψ at time $\mathcal{T}$.*



(a) No solution exists with additional constraint $A_\delta : \delta_X = \delta_Y = \Delta/2$ - common assumption in prior work.



(b) Bounds for first half of a gender-fairness specification generated by AVOIR-OB and AVOIR-VF.

Figure 3: *(Left)* AVOIR finds a solution for a *theoretical* scenario with $\delta_X + \delta_Y \leq \Delta$ under constraint $\epsilon_X + \epsilon_Y \leq \epsilon_T$ *(Right)* For *RateMyProfs*, a real-world dataset, the vertical lines show the step at which the methods can provide a guarantee of failure for the upper bounds with $\Delta <= 0.05$.

In all prior work (Albarghouthi et al., 2017; Albarghouthi & Vinitsky, 2019; Bastani et al., 2019), δ_i for each *elementary* subexpressions is set to Δ/n, where n is the number such term in the specification. This simplification is carried out using the assumption $A_\delta := \delta_i = \delta_j$ for all *elementary* subexpressions. As we do not make this assumption, we can prove the following key theorem.

Theorem 2. *Given a threshold probability Δ for a specification ψ, let the stopping time for AVOIR be $\mathcal{T}$ and stopping time with the A_δ assumption be $\mathcal{T}^+$. Then $\mathcal{T} \leq \mathcal{T}^+$*

See Appendix D for the proof.

3.4 Visualization for Interactive Refinement

Using our specification framework as a backend, we built an interactive application for analysis and refinement of specifications provided in our grammar. Specifically, fairness specifications can be naturally parsed into a tree because of the structure of the grammar. Each node of the tree represents some sub-expression in the syntax tree of the overall specification. These nodes allow a user of AVOIR to interactively audit and tune the specification definition. To create the visualization, we use Vega (Satyanarayan et al., 2015), a declarative JSON-based visualization grammar. We log the estimates during runs of AVOIR and then output the grammar in a tabular JSON-format that contains a row for each grammar element and its associated evaluations. This tabular data is used by our Vega specification to produce the visualizations. By selecting one of the nodes in the syntax tree, a user can see a plot of the evaluation values associated with the selected grammar element. This allows for comparison of multiple grammar elements. The ability to analyze and compare these evaluation values provides context surrounding specification violations, and assists the user in interactiving with and deciding how to refine a specification We provide a detailed example of how these interactions can help AVOIR users choose an appropriate fairness metric in Section 4.

4 Case Studies

The following text describes two real-world scenarios for AVOIR. We implement both Verifair and our optimized bounds within the AVOIR framework, denoted as AVOIR-VF and AVOIR-OB, respectively. An important case study on the COMPAS dataset can be found in Appendix G.

4.1 RATE MY PROFS

In this section, we provide a detailed black-box machine learning model (ML) based case study on a real-world dataset. In this case study, we use the rate my professors (RMP) dataset released by Keymanesh et al. (2021). This dataset includes professor names and reviews for them written by students in their classes, ratings, and certain self-reported attributes of the reviewer. Ratings are provided on a five-point scale (1-5 stars). We use the preprocessing described in Keymanesh et al. (2021) to infer the gender attribute for the professors. This dataset is divided into an 80-20 split (train-test). We then train a BERT-based transformer model Devlin et al. (2019) on the training split. We use the implementation from the simpletransformers[4] package. The loss function chosen is the mean-squared error from the true ratings. On the test set, we track a gender-fairness specification in the model outputs:

(E[r $>$ 3 | gender = F] / E[r $>$ 3 | gender = M $<$ 1.2) & \
(E[r $>$ 3 | gender = M)] / E[r $>$ 3 | gender = F] $>$ 0.8)

We set the failure probability $\Delta = 0.05$. `OPT` is run after each batch (5 items/batch). Figure 3b shows that AVOIR-OB[5] can provide a guarantee in $\mathbf{2.5}\%$ fewer iterations than AVOIR-VF. Note also that the OB guarantee provided tries to optimize for the failure probability while staying under the required threshold, remaining closer to the required threshold in subsequent steps.

4.2 ADULT INCOME

In this case study we use the Adult income dataset (Kohavi, 1996) which has been used frequently in prior fairness-related work. The historical dataset labels individuals from the 1994 census as having a *high-income* ($> 50,000$ a year) or not ($\leq 50,000$ a year). In this case study, we look at a column of data as a black-box measurement (internally, we use a materialized view, details in Appendix G.1).

US Federal laws mandate against race and sex based discrimination. Thus, the specification we start our analysis with is a group fairness property that monitors the difference of the proportions of individuals with sex recorded as male that have a high income to females that have a high income should be less than 0.5. In addition, we ensure that the difference between individuals with race marked as white and those without should have a difference of less than 0.5. The associated specification is given below, where `h` is an indicator for whether an individual is *high-income* is the binary classification output of our model:

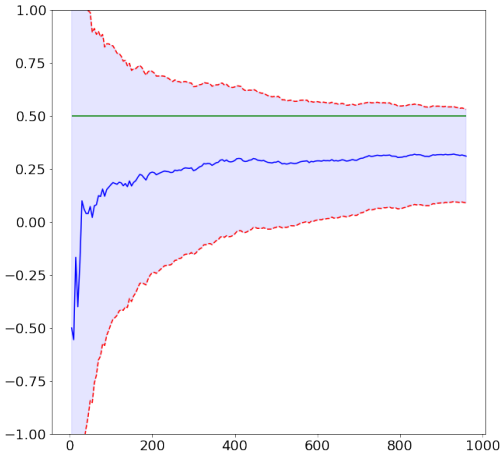
(E[h | sex=M] – E[h | sex=F] $<$ 0.5) & \
(E[h | race=W] – E[h | race!=W] $<$ 0.5)

In this example, we set the failure threshold probability $\Delta = 0.15$

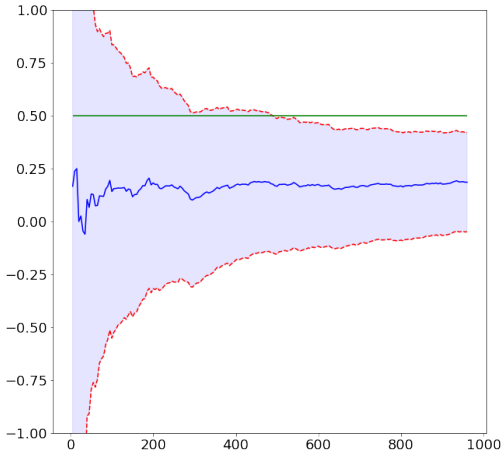
When run with this specification, the generated materialized view cannot achieve the required bound. We can then use our iterative refinement visualization tool to analyze different components of the specification. A developer would first interact with the left subtree of the specification. Due to paucity of space, this visualization is presented in Appendix G.2. The plot for the corresponding data is shown in Figure 4a shows that guarantees cannot converge under the threshold with the given number of data samples. The developer can now choose to either reduce the guarantee (i.e. reduce δ) or increase the threshold. Next, analyzing the right subtree, the race group fairness term can be guaranteed to be under the threshold (Figure 4b). Using this information, the developer can make an intelligent decision to increase the threshold on the group fairness for term for sex. Suppose they increase it to 0.55 and rerun the analysis. OB is able provide a guarantee at this threshold within 870 steps, whereas VF can provide it at 960 steps, demonstrating a relative improvement of about $\mathbf{10.35}\%$. Additionally, the optimal δ split across the terms are $\approx (0.135, 0.36 * 10^4)$ which is far from the equal split allocated by VF. The reason for this split is because increasing the threshold for the first time provides the optimizer with additional legroom to better distribute the failure probabilities between the two terms.

[4]https://simpletransformers.ai/

[5]OB = Optimized Bounds



(a) Group fairness for sex. Difference in ratio of high income earners in left subtree for initial specification.



(b) Group fairness for race (difference in ratio of high income earners) in right subtree for initial specification.

Figure 4: *(Left)* Red dotted lines the upper bound of the value cannot be guaranteed to be under the threshold at the specified failure probability. *(Right)* Guarantee possible with given data.

5 Discussion & Future Work

The case studies presented in the previous section demonstrate the ability of our tools to provide vital context when deciding how to refine a model or fairness specification. Although this contextual information makes decisions easier, it is not always clear how one should alter a specification in light of a violation and its relevant context. To assist in these decisions, we are currently examining ways work to suggest edits that are likely to achieve the desired intent of a developer. Using our visual analysis tool for refinement, we can gather edits from developers and then use that data to learn iterative changes to the syntax tree of the specification. In addition to improving the usability of our tools for making fairness specification refinements, we also envision a more scalable framework. Our case studies look at a single model with respect to a single dataset. However, real-world deployment of machine learning often contain many clients with models that may differ. We take it as future work to study the efficient monitoring of machine learning behavior with respect to a fairness specification in a distributed context, enabling horizontal scalability. We believe techniques such as decoupling the observation of data and the reporting results from the monitoring of the results are promising and can lead to the desired scalability.

6 Conclusion

We present the AVOIR framework for easily defining and monitoring fairness specifications online and aids in the interactive refinement of specifications. AVOIR is easy to integrate within modern database systems but can also serve as a standalone system evaluating whether blackbox machine learning models are meeting specific fairness criteria on specific datasets (including both structured and unstructured data) as described in our case studies. AVOIR extends the grammar from Fairness Aware Programming Albarghouthi & Vinitsky (2019) with operations that enhance expressiveness. In addition we derive probabilistic guarantees that improve the confidence with which specification violations are reported. To assist in refinement of specifications, we develop an interactive visual analysis application within AVOIR. Through case studies, we demonstrate that AVOIR can provide users with insights that contribute directly to refinement decisions. Our framework builds the foundation for further improvements to the fairness specification, auditing and verification workflow. We plan to extend this work to provide intelligent specification refinement suggestions and support distributed machine learning settings. We also plan to explore the use of AVOIR for fair ranking problems and tailored database integration.

7 REPRODUCIBILITY

To enhance the reproducibility of our work, on the theoretical side, all proofs (with the necessary assumptions) are provided in the appendix. Specifically, proofs for the inference engine are in Appendix A, and proofs for the correctness of bounds are provided in Appendix C. Theorem 2, which shows how AVOIR improves over prior work is proved in Appendix D. To reproduce the results of the case studies in the paper, each case study is encapsulated inside a Jupyter notebook. These notebooks are attached along with the source code for AVOIR. In addition, all datasets used for generating results for the case studies are also attached in the submitted supplementary documentation. Finally, the model weights used for the *RateMyProfs* study for exact reproduction are provided in a dropbox folder hosted at `https://www.dropbox.com/sh/n5o4vswnkxv34zr/AABthgLMaYL3MuA0KC39Z1G8a?dl=0`.

REFERENCES

Aws Albarghouthi and Samuel Vinitsky. Fairness-aware programming. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 211–219, 2019.

Aws Albarghouthi, Loris D'Antoni, Samuel Drews, and Aditya V Nori. Fairsquare: probabilistic verification of program fairness. *Proceedings of the ACM on Programming Languages*, 1 (OOPSLA):1–30, 2017.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of Data and Analytics*, pp. 254–264. Auerbach Publications, 2016.

Abolfazl Asudeh, H. V. Jagadish, Julia Stoyanovich, and Gautam Das. Designing fair ranking schemes. In Peter A. Boncz, Stefan Manegold, Anastasia Ailamaki, Amol Deshpande, and Tim Kraska (eds.), *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, pp. 1259–1276. ACM, 2019.

Osbert Bastani, Xin Zhang, and Armando Solar-Lezama. Probabilistic verification of fairness properties via concentration. *Proceedings of the ACM on Programming Languages*, 3(OOPSLA): 1–27, 2019.

R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilović, S. Nagar, K. Natesan Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4:1–4:15, 2019. doi: 10.1147/JRD.2019.2942287.

Alessandro Castelnovo, Riccardo Crupi, Greta Greco, and Daniele Regoli. The zoo of fairness metrics in machine learning. *arXiv preprint arXiv:2106.00467*, 2021.

US Congress. Hr 2231–algorithmic accountability act of 2019, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

William Dieterich, Christina Mendoza, and Tim Brennan. Compas risk scales: Demonstrating accuracy equity and predictive parity. *Northpointe Inc*, 7(4), 2016.

Avi Feller, Emma Pierson, Sam Corbett-Davies, and Sharad Goel. A computer program used for bail and sentencing decisions was labeled biased against blacks. it's actually not that clear. *The Washington Post*, 17, 2016.

Gemma Galdon Clavell, Mariano Martín Zamorano, Carlos Castillo, Oliver Smith, and Aleksandar Matic. *Auditing Algorithms: On Lessons Learned and the Risks of Data Minimization*, pp. 265–271. Association for Computing Machinery, 2020.

Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. Fairness testing: testing software for discrimination. In *Proceedings of the 2017 11th Joint meeting on foundations of software engineering*, pp. 498–510, 2017.

Bishwamittra Ghosh, Debabrota Basu, and Kuldeep S Meel. Justicia: a stochastic sat approach to formally verify fairness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7554–7563, 2021a.

Soumya Ghosh, Q Vera Liao, Karthikeyan Natesan Ramamurthy, Jiri Navratil, Prasanna Sattigeri, Kush R Varshney, and Yunfeng Zhang. Uncertainty quantification 360: A holistic toolkit for quantifying and communicating the uncertainty of ai. *arXiv preprint arXiv:2106.01410*, 2021b.

Tony Ginart, Martin Jinye Zhang, and James Zou. Mldemon: Deployment monitoring for machine learning systems. In *International Conference on Artificial Intelligence and Statistics*, pp. 3962–3997. PMLR, 2022.

William E Hart, Jean-Paul Watson, and David L Woodruff. Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3(3):219–260, 2011.

Dennis Hirsch, Tim Bartley, Arvind Chandrasekaran, Srinivasan Parthasarathy, Piers Turner, Devon Norris, Keir Lamont, and Christina Drummond. Corporate data ethics: Data governance transformations for the age of advanced analytics and ai (final report). In *Appeared at the Privacy Law Scholars Conference (also available as SSRN Abstract: 3828239)*, 2020.

Steven R Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021.

B Justice Srikrishna. A free and fair digital economy: Protecting privacy, empowering indians, 2018.

Moniba Keymanesh, Tanya Berger-Wolf, Micha Elsner, and Srinivasan Parthasarathy. Fairness-aware summarization for justified decision-making. *arXiv preprint arXiv:2107.06243*, 2021.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent Trade-Offs in the Fair Determination of Risk Scores. In Christos H. Papadimitriou (ed.), *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 43:1–43:23, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-029-3. doi: 10.4230/LIPIcs.ITCS.2017.43. URL http://drops.dagstuhl.de/opus/volltexte/2017/8156.

Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pp. 202–207, 1996.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pp. 220–229, 2019.

E.F. Moore. Gedanken-experiments on sequential machines. *Automata Studies, Princeton University Press*, 129-153, 1956.

Fatemeh Nargesian, Abolfazl Asudeh, and H. V. Jagadish. Tailoring data source distributions for fairness-aware data integration. *Proc. VLDB Endow.*, 14(11):2519–2532, 2021.

Central Digital Office and Data. Algorithmic transparency standard, Nov 2021. URL https://www.gov.uk/government/collections/algorithmic-transparency-standard.

European Parliament. 2018 reform of eu data protection rules. *European Commission*, 2018.

Arvind Satyanarayan, Ryan Russell, Jane Hoffswell, and Jeffrey Heer. Reactive vega: A streaming dataflow architecture for declarative interactive visualization. *IEEE transactions on visualization and computer graphics*, 22(1):659–668, 2015.

Kacper Sokol and Peter Flach. Explainability fact sheets: a framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 56–67, 2020.

Chenkai Sun, Abolfazl Asudeh, H. V. Jagadish, Bill Howe, and Julia Stoyanovich. Mithralabel: Flexible dataset nutritional labels for responsible data science. In Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (eds.), *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pp. 2893–2896. ACM, 2019.

Florian Tramèr, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, Jean-Pierre Hubaux, Mathias Humbert, Ari Juels, and Huang Lin. Fairtest: Discovering unwarranted associations in data-driven applications. In *2017 IEEE European Symposium on Security and Privacy (EuroS P)*, pp. 401–416, 2017. doi: 10.1109/EuroSP.2017.29.

Richard Vanderford. New york's landmark ai bias law prompts uncertainty. *WSJ*, 2022. URL `https://www.wsj.com/articles/new-yorks-landmark-ai-bias-law-prompts-uncertainty-11663752602`.

Sahil Verma and Julia Rubin. Fairness definitions explained. In *2018 ieee/acm international workshop on software fairness (fairware)*, pp. 1–7. IEEE, 2018.

Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1): 25–57, 2006.

Anne L Washington. How to argue with an algorithm: Lessons from the compas-propublica debate. *Colo. Tech. LJ*, 17:131, 2018.

Ke Yang, Julia Stoyanovich, Abolfazl Asudeh, Bill Howe, H. V. Jagadish, and Gerome Miklau. A nutritional label for rankings. In Gautam Das, Christopher M. Jermaine, and Philip A. Bernstein (eds.), *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pp. 1773–1776. ACM, 2018.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pp. 962–970. PMLR, 2017.

Jie M. Zhang, Mark Harman, Lei Ma, and Yang Liu. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, pp. 1–1, 2020. doi: 10.1109/TSE.2019.2962027.

Shengjia Zhao, Enze Zhou, Ashish Sabharwal, and Stefano Ermon. Adaptive concentration inequalities for sequential decision problems. *Advances in Neural Information Processing Systems*, 29, 2016.

A INFERENCE RULES

In Figure 5, we provide a set of rules that can be used to determining the constraints and guarantees associated with a specification. We represent

$$X \odot Y : (E, \epsilon, \delta) \equiv \Pr\left(|\mathbb{E}[X] \odot \mathbb{E}[Y] - E| \geq \epsilon\right) \leq \delta$$

where $\odot$ represents a binary operator. Constraints are represented in curly brackets $\{\}$.

The proof of correctness for each inference rule starts from the assumptions above the horizontal line and derives the assertions below. These proofs ise ideas similar to those in Bastani et al. (2019). We reproduce the proofs in Appendix A.1 here for completeness. Note that the assertions in the base case (elementary subexpressions) can be arrived at by applying AIN.

$$\frac{X : \left(\overline{\mathbb{E}}[X], \epsilon_X, \delta_X\right), Y : \left(\overline{\mathbb{E}}[Y], \epsilon_Y, \delta_Y\right)}{X \pm Y : \left(\overline{\mathbb{E}}[X] \pm \overline{\mathbb{E}}[Y], \epsilon_X + \epsilon_Y, \delta_X + \delta_Y\right)}$$

$$\frac{X : \left(\overline{\mathbb{E}}[X], \epsilon_X, \delta_X\right), Y : \left(\overline{\mathbb{E}}[Y], \epsilon_Y, \delta_Y\right)}{X \times Y : (\overline{\mathbb{E}}[X]\overline{\mathbb{E}}[Y], \epsilon_X\epsilon_Y + \overline{\mathbb{E}}[X]\epsilon_Y + \overline{\mathbb{E}}[Y]\epsilon_X, \delta_X + \delta_Y)}$$

$$\frac{X : \left(\overline{\mathbb{E}}, \epsilon, \delta\right), \overline{\mathbb{E}} - \epsilon > 0}{X^{-1} : \left(\overline{\mathbb{E}}^{-1}, \frac{\epsilon}{\overline{\mathbb{E}}(\overline{\mathbb{E}} - \epsilon)}, \delta\right)} \text{ (Inverse)} \qquad \frac{X : \left(\overline{\mathbb{E}}, \epsilon, \delta\right)}{X^{-1} : \left(\overline{\mathbb{E}}^{-1}, \frac{\epsilon}{\overline{\mathbb{E}}(\overline{\mathbb{E}} - \epsilon)}, \delta\right), \{\overline{\mathbb{E}} - \epsilon > 0\}} \text{ (Inverse Constr.)}$$

$$\frac{X : \left(\overline{\mathbb{E}}, \epsilon, \delta\right), \overline{\mathbb{E}} - \epsilon > c}{\psi \equiv X > c : (T, \delta)} \text{ (True)} \qquad \frac{X : \left(\overline{\mathbb{E}}, \epsilon, \delta\right), \overline{\mathbb{E}} + \epsilon < c}{\psi \equiv X < c : (F, \delta)} \text{ (False)}$$

$$\frac{X : \left(\overline{\mathbb{E}}, \epsilon, \delta\right)}{\psi \equiv X > c : (T, \delta), \{\overline{\mathbb{E}} - \epsilon > c\}} \text{ (True Constr.)}$$

$$\frac{X : \left(\overline{\mathbb{E}}, \epsilon, \delta\right)}{\psi \equiv X < c : (T, \delta), \{\overline{\mathbb{E}} + \epsilon < c\}} \text{ (False Constr.)}$$

$$\frac{\psi_1 : (\mathbb{B}_1, \delta_1), \psi_2 : (\mathbb{B}_2, \delta_2)}{\psi_1 \wedge \psi_2 : (\mathbb{B}_1 \wedge \mathbb{B}_2, \delta_1 + \delta_2)} \text{ (and)} \qquad \frac{\psi_1 : (\mathbb{B}_1, \delta_1), \psi_2 : (\mathbb{B}_2, \delta_2)}{\psi_1 \vee \psi_2 : (\mathbb{B}_1 \vee \mathbb{B}_2, \delta_1 + \delta_2)} \text{ (or)}$$

$$\frac{\psi_1 : (\mathbb{B}_1, \delta_1), \{C_{11,\ldots,1k}\}, \psi_2 : (\mathbb{B}_2, \delta_2), \{C_{21,\ldots,2m}\}}{\psi_1 \wedge \psi_2 : (\mathbb{B}_1 \wedge \mathbb{B}_2, \delta_1 + \delta_2), \{C_{11,\ldots,1k}, C_{21,\ldots,2m}\}} \text{ (and constr.)}$$

$$\frac{\psi_1 : (\mathbb{B}_1, \delta_1), \{C_{11,\ldots,1k}\}, \psi_2 : (\mathbb{B}_2, \delta_2)}{\psi_1 \vee \psi_2 : (\mathbb{B}_1 \vee \mathbb{B}_2, \delta_1 + \delta_2), \{C_{11,\ldots,1k}\} \vee \{C_{21,\ldots,2m}\}} \text{ (or constr.)}$$

Figure 5: Inference rules used to guarantees for expressions.The inference rules for each compound expression build on the union bound, triangle inequality, and structural induction approach described by Bastani et al. (2019).

A.1 INFERENCE RULES WITH CONSTRAINTS

In Section 3.2 we provided the proofs for $X \pm Y$, $X > c$. In the following text we provide the proofs for the remainder of the inference rules.

Product Starting with $\phi_X = X : (\overline{\mathbb{E}}[X], \epsilon_X, \delta_X)$, $\phi_Y = Y : (\overline{\mathbb{E}}[Y], \epsilon_Y, \delta_Y)$. First, from union bound, both of these hold true with probability at least $1 - \delta_X - \delta_Y$. Then,

$$\begin{aligned}
|\mathbb{E}[X]| &= |\overline{\mathbb{E}}[X] - \overline{\mathbb{E}}[X] + \mathbb{E}[X]| \\
&\leq ||\overline{\mathbb{E}}[X]| + |\overline{\mathbb{E}}[X] + \mathbb{E}[X]| \\
&\leq ||\overline{\mathbb{E}}[X]| + \epsilon_X
\end{aligned}$$

We have

$$
\begin{aligned}
|\overline{\mathbb{E}}[X]\overline{\mathbb{E}}[Y] - \mathbb{E}[XY]| &= |\overline{\mathbb{E}}[X]\overline{\mathbb{E}}[Y] - \mathbb{E}[X]\mathbb{E}[Y]| && \text{(as } X, Y \text{ bernoulli)} \\
&= |\overline{\mathbb{E}}[X]\overline{\mathbb{E}}[Y] - \overline{\mathbb{E}}[X]\mathbb{E}[Y] + \overline{\mathbb{E}}[X]\mathbb{E}[Y] - \mathbb{E}[X]\mathbb{E}[Y]| \\
&= |\overline{\mathbb{E}}[X](\overline{\mathbb{E}}[Y] - \mathbb{E}[Y]) + \mathbb{E}[Y](\overline{\mathbb{E}}[X] \ - \mathbb{E}[X])| \\
&\leq |\overline{\mathbb{E}}[X]||(\overline{\mathbb{E}}[Y] - \mathbb{E}[Y])| + |\mathbb{E}[Y]||(\overline{\mathbb{E}}[X] \ - \mathbb{E}[X])| \\
&\leq |\overline{\mathbb{E}}[X]|\epsilon_Y + |\mathbb{E}[Y]|\epsilon_X \\
&\leq |\overline{\mathbb{E}}[X]|\epsilon_Y + (|\overline{\mathbb{E}}[Y]| + \epsilon_Y)\epsilon_X \\
&= |\overline{\mathbb{E}}[X]|\epsilon_Y + |\overline{\mathbb{E}}[Y]|\epsilon_X + \epsilon_X\epsilon_Y
\end{aligned}
$$

Therefore, $X \times Y : (\overline{\mathbb{E}}[X]\overline{\mathbb{E}}[Y], \epsilon_X\epsilon_Y + \overline{\mathbb{E}}[X]\epsilon_Y + \overline{\mathbb{E}}[Y]\epsilon_X, \delta_X + \delta_Y)$

Inverse/Inverse constr. Assume $X : \left(\overline{\mathbb{E}}, \epsilon, \delta\right)$ and $\overline{\mathbb{E}} - \epsilon > 0$. Instead, in the constrained case, we start with only the prior assumption i.e., $X : \left(\overline{\mathbb{E}}, \epsilon, \delta\right)$ Then,

$$
\begin{aligned}
|\mathbb{E}[X]| &= |\mathbb{E}[X] - \overline{\mathbb{E}}[X] + \overline{\mathbb{E}}[X]| \\
&\leq |\mathbb{E}[X] - \overline{\mathbb{E}}[X]| + |\overline{\mathbb{E}}[X]| \\
&\leq \epsilon_X + |\overline{\mathbb{E}}[X]|
\end{aligned}
$$

i.e., $|\mathbb{E}[X]| \leq \epsilon_X + |\overline{\mathbb{E}}[X]|$. Also,

$$
\begin{aligned}
|\mathbb{E}[X]^{-1} - \overline{\mathbb{E}}[X]^{-1}| &= \left|\frac{\overline{\mathbb{E}}[X]^{-1} - \mathbb{E}[X]^{-1}}{\overline{\mathbb{E}}[X]\mathbb{E}[X]^{-1}}\right| \\
&\leq \frac{\epsilon}{|\mathbb{E}[X]||\overline{\mathbb{E}}[X]|} \\
&\leq \frac{\epsilon}{|\mathbb{E}[X]|(\mathbb{E}[X] - \epsilon_X)}
\end{aligned}
$$

where the last step follows from the previous derivation and if $\mathbb{E}[X] - \epsilon_X > 0$. The latter condition enforces that the sign of the inequality does not change. VF adds this as a precondition; we add it as a post-constraint.

Boolean Operators Starting from $\psi_1 : (b_1, \delta_1)$, $\psi_2 : (b_2, \delta_2)$, we can apply the union bound for $\psi_1 \wedge \psi_2$, $\psi_1 \vee \psi_2$ to derive the rules for and/or. Similarly, constraints follow the semantics specified by the rules as they also follow from the union bound.

A.2 Inferred Optimization Problem

For a given overall specification ψ, suppose (ϵ_i, δ_i), $i \in \{1, \ldots, n\}$ represents the concentration bounds associated with each constituent elementary subexpression. Using the aforementioned inference rules, we can derive the overall $\delta_T = \sum_i \delta_i$, along with a set of (say) K constraints

$$
g_k(\epsilon_1, \ldots, \epsilon_n, \overline{\mathbb{E}}[X_1], \ldots, \overline{\mathbb{E}}[X_n]) \leq \epsilon_k
$$

where

$$
\epsilon_k = \left|c_k - \overline{\mathbb{E}}[f(\overline{\mathbb{E}}[X_1], \ldots, \overline{\mathbb{E}}[X_n])]\right|
$$

denotes the maximum allowed margin for the kth inequality subexpression (i.e. having form `<ETerm> <comp-op> c`). The objective is to minimize the overall failure probability δ_T. The overall optimization problem can then be formulated as shown in 2, having n optimizaiton variables δ_i and $2n + K$ constraints (bounds on δ_i provide the $2n$ constraints). A developer using AVOIR inputs a required acceptable upper bound of failure probability Δ. If the solution to the optimization problem $\delta_T^* = \sum_i \delta_i \leq \Delta$, then the optimization can conclude with the required confidence in the proved guarantee. At this point, the developer may choose to terminate AVOIR. However, using Corollary 4.1, they may continue to run and refine the estimates.

B CONCENTRATION BOUNDS

The adaptive Hoeffding inequality (Zhao et al., 2016; Bastani et al., 2019).

Theorem 3. *Given a Bernoulli random variable X with distribution P_X. Let $\{X_i \sim P_X\}, i \in \mathbb{N}$ be i.i.d samples of X. Let*

$$\overline{\mathbb{E}}_t[X] = \frac{1}{t}\sum_{i=1}^{t} X_i.$$

Let $\mathcal{T}$ be a random variable on $\mathbb{N} \cup \{\infty\}$ such that $\Pr[\mathcal{T} < \infty] = 1$, and let

$$\epsilon(\delta, t) = \sqrt{\frac{\frac{3}{5}\log\left(\log_{1.1} t + 1\right) + \frac{5}{9}\log\left(24/\delta\right)}{t}}$$

Then, for any $\delta \in \mathbb{R}_+$, we have

$$\Pr[|\overline{\mathbb{E}}_{\mathcal{T}}[X] - \mathbb{E}[X]| \leq \epsilon(\delta, \mathcal{T})|] \geq 1 - \delta$$

.

Theorem 3 provides a mechanism for choosing the stopping time using arbitrary methods for a fixed δ. Note that in general, any adaptive concentration inequality suffices; we use the Hoeffding inequality that does not depend on the empirical variance but is frequently used in scenarios dealing with bounded rvs. However, we use confidence intervals to visualize the evolution of sub-expressions (and overall specification) over the sequence of observations. For doing so, we require an additional result

Theorem 4. *(Zhao et al., 2016, Proposition 1, Lemma 1) Let $S_n = \sum_{i=1}^{n} X_i$ be a random walk from i.i.d. random variables $X_1, \ldots, X_t \sim D$. For any $\delta > 0$,*

$$\Pr[S_{\mathcal{T}} \geq f(\mathcal{T})] \leq \delta$$

for any stopping time $\mathcal{T}$ if and only if

$$\Pr\left[\exists n, S_t \geq f(t)\right] \leq \delta$$

Corollary 4.1. *For any $\delta > 0$,*

$$\Pr[|\overline{\mathbb{E}}_{\mathcal{T}}[X] - \mathbb{E}[X]| \leq \epsilon(\delta, \mathcal{T})|] \geq 1 - \delta$$

for any stopping time $\mathcal{T}$ if and only if

$$\Pr\left[\forall t, |\overline{\mathbb{E}}_t[X] - \mathbb{E}[X]| \leq \epsilon(\delta, t)|\right] \geq 1 - \delta$$

Proof. Follows directly from applying Theorem 4 to Theorem 3. □

Intuitively, Theorem 3 holds since we can choose an adversarial stopping rule for $\mathcal{T}$ that terminates as soon as the boundary for $\epsilon(\delta, t)$ is crossed (Zhao et al., 2016). Thus, when we establish a bound with a stopping rule, as long as the underlying distribution remains unchanged, the bound will hold prior to and after the stopping rule is enforced. Theorem 4.1 implies that once we choose an optimal bound for each subexpression, we can extend the bounds derived using Theorem 3 to following observations with the guarantees for the subexpressions still holding true.

C CONFIDENCE SEQUENCES

In this section, we show that the estimates generated from AVOIR form a confidence set (Theorem 1). First, we assume the existence of a concentration sequence for the mean of each elementary subexpression (eg., Theorem 3 can provide one). That is, we need a function $\epsilon(t, \delta)$ such that

$$\Pr[\forall t \geq 1, |\overline{\mathbb{E}}_t[X] - \mathbb{E}[X]| \leq \epsilon(t, \delta_X)] \geq 1 - \delta_X. \tag{5}$$

For convenience of exposition, we denote such adaptive inequality functions as AIN. For any AIN to be usable with AVOIR, we require $\epsilon(t, \delta)$ to be monotonically non-increasing in δ and n. We expect this to be the case for most AIN, since increasing the number of observations of increasing the failure threshold should allow for additional concentration around the mean. For example, the adaptive Hoeffding inequality (Theorem 3) follows this assumption. Second, we assume that, except in degenerate cases, AVOIR terminates (see corollary 4.2 for termination criteria). We now prove Theorem 1.

Proof. First, we will prove that the estimates for *elementary* subexpressions are a confidence sequence. Following this, using the inference rules from Appendix A, we will show that the estimates for every compound expression are also a confidence sequance.

Elementary subexpressions Consider a specification ψ consisting of *elementary* subexpressions $X_1, \ldots, X_n$. At stopping time $\mathcal{T}$, let

$$\phi_{X_i}^{\mathcal{T}} := X_i : (\overline{\mathbb{E}}_{\mathcal{T}}[X_i], \epsilon(\mathcal{T}, \delta_{X_i}), \delta_{X_i}) \tag{6}$$

be the stopping time estimates. Then, from the termination criterion, a solution to the optimization problem OPT exists, i.e,

$$\Delta \geq \sum_i \delta_{X_i} \tag{7}$$

The sequence of bounds claimed by AVOIR are

$$\epsilon_{X_i}(t) = \begin{cases} \epsilon(\Delta, t), & t < \mathcal{T}, \\ \epsilon(\delta_{X_i}, t), & t \geq \mathcal{T} \end{cases} \tag{8}$$

From equation 7 and the optimization constraint $\delta_i \in [0, 1]$ we have $\Delta \geq \delta_{X_i}$. From the non-decreasing behavior of AIN

$$\epsilon(\Delta, t) \leq \epsilon(\delta_i, t) \tag{9}$$

Now

$$\begin{aligned}
&\Pr[\forall t \geq 1, |\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| \leq \epsilon_{X_i}(t)] \\
&\quad = 1 - \Pr[\exists t \geq 1, |\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| > \epsilon_{X_i}(t)] \\
&\quad = 1 - \Pr\left[\bigcup_{t \geq 1} \left\{|\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| > \epsilon_{X_i}(t)\right\}\right] \\
&\quad = 1 - \Pr\left[\bigcup_{t=1}^{\mathcal{T}-1} \left\{|\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| > \epsilon_{X_i}(t)\right\} \cup \bigcup_{t \geq \mathcal{T}} \left\{|\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| > \epsilon_{X_i}(t)\right\}\right] \\
&\quad \text{(associativity of } \cup) \\
&\quad = 1 - \Pr\left[\bigcup_{t=1}^{\mathcal{T}-1} \left\{|\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| > \epsilon(\Delta, t)\right\} \cup \bigcup_{t \geq \mathcal{T}} \left\{|\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| > \epsilon(\delta_{X_i}, t)\right\}\right] \\
&\quad \text{(From 8)} \\
&\quad = 1 - \Pr\left[\bigcup_{t=1}^{\mathcal{T}-1} \left\{|\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| > \epsilon(\delta_{X_i}, t) \cup |\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| \in (\epsilon(\Delta, t), \epsilon(\delta_{X_i}, t)]\right\} \cup \right. \\
&\quad \left. \bigcup_{t \geq \mathcal{T}} \left\{|\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| > \epsilon(\delta_{X_i}, t)\right\}\right] \quad \text{(Using 9)} \\
&\quad = 1 - \Pr\left[\bigcup_{t=1}^{\mathcal{T}-1} \left\{|\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| \in (\epsilon(\Delta, t), \epsilon(\delta_{X_i}, t)]\right\} \cup \bigcup_{t \geq 1} \left\{|\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| > \epsilon(\delta_{X_i}, t)\right\}\right] \\
&\quad \text{(Rearranging)} \\
&\quad \geq 1 - \Pr\left[\bigcup_{t \geq 1} \left\{|\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| > \epsilon(\delta_{X_i}, t)\right\}\right] \\
&\quad = 1 - \Pr\left[\exists t \geq 1, |\overline{\mathbb{E}}_t[X_i] - \mathbb{E}[X_i]| > \epsilon(\delta_{X_i}, t)\right] \\
&\quad \geq 1 - \delta_{X_i}
\end{aligned}$$

where the last step follows from the definition of the adaptive concentration bound used. Thus, $\epsilon_{X_i}(t)$ defines a $1 - \delta_{X_i}$ confidence sequence for $\mathbb{E}[X_i]$.

Compound subexpressions Consider a non-specification compound <ETerm> C_j consisting of *elementary* subexpressions with indices $\mathsf{C}_j = \{\{j_1, j_2, \ldots, j_M\}\}$ as the decision r.v.s, i.e, $X_{j_1} \ldots, X_{j_M}$. Note that C_j is a multiset as the same expression could occur multiple times within C_j. At stopping time $\mathcal{T}$,

$$\phi^{\mathcal{T}}_{C_j} : (\overline{\mathbb{E}}_{\mathcal{T}}[C_j], \delta_{C_j}, \epsilon_{C_j}) \tag{10}$$

where $\overline{\mathbb{E}}_{\mathcal{T}}[C_j], \delta_{C_j}, \epsilon_{C_j}$ are the corresponding values computed through the inference rules. In general, we denote by

$$\overline{\mathbb{E}}_t[C_j], \delta_{C_j}(t), \epsilon_{C_j}(t) = \mathrm{INFER}(\phi^{\mathrm{t}}_{\mathrm{X_{j_1}}}, \ldots, \phi^{\mathrm{t}}_{\mathrm{X_{j_M}}}) \tag{11}$$

the values inferred at time step t, where INFER denotes the inference rules. Now,

$$\begin{aligned}
&\Pr[\exists t \geq 1, |\mathbb{E}[C_j] - \overline{\mathbb{E}}[C_j]| > \epsilon_{C_j}(t)] \\
&\quad \leq \Pr\left[\bigcup_{i=1}^{M} \exists t \geq 1, \neg\phi^t_{X_{j_i}}\right] && \text{(From 11)} \\
&\quad \leq \sum_{i \in \mathsf{C}_j} \Pr\left[\exists t \geq 1, \neg\phi^t_{X_{j_i}}\right] && \text{(union bound)} \\
&\quad = \sum_{i \in \mathsf{C}_j} \Pr\left[\exists t \geq 1, |\overline{\mathbb{E}}_t[X_{j_i}] - \mathbb{E}_t[X_{j_i}| > \epsilon_{X_{j_i}}(t)\right] && \text{(definition of } \phi^t_{X_{j_i}}\text{)} \\
&\quad \leq \sum_{i \in \mathsf{C}_j} \delta_{X_{j_i}} && \text{(elementary subexpressions)} \\
&\quad \leq \delta_{C_j} && \text{(applying 11 for } t = \mathcal{T}\text{)}
\end{aligned}$$

Therefore $\epsilon_{C_j}(t)$ defines a $1 - \delta_{C_j}$ confidence sequence for $\mathbb{E}[C_j]$

A similar proof can be constructed for any <spec>. Consider any specification ψ_k. Let

$$\psi^t_k : (\hat{b}_{\psi_k}(t), \delta_{\psi_k}(t)) \tag{12}$$

where $\hat{b}_{\psi_k}(t) \subseteq \{T, F\}$ is the inferred value and $\delta_{\psi_k}(t)$ corresponds to the confidence for the assertion at time t. Let the *elementary* subexpressions involved be $X_{k_1}, \ldots, X_{k_D}$ corresponding to the index multiset $\mathsf{B}_k = \{\{k_1, \ldots, k_D\}\}$. Denote b_{ψ_k} as the true value of ψ_k, and δ_{ψ_k} as the inferred threshold at stopping time $\mathcal{T}$. From INFER, we have

$$\hat{b}_k(t), \delta_{\psi_k}(t) = \mathrm{INFER}(\phi^{\mathrm{t}}_{\mathrm{X_{k_1}}}, \ldots, \phi^{\mathrm{t}}_{\mathrm{X_{k_D}}}) \tag{13}$$

We have

$$\begin{aligned}
&\Pr[\exists t \geq 1, b_k \notin \hat{b}_k(T)] \\
&\quad \leq \Pr\left[\bigcup_{i=1}^{D} \exists t \geq 1, \neg\phi^t_{X_{k_i}}\right] && \text{(From 13)} \\
&\quad \leq \sum_{i \in \mathsf{B}_k} \Pr\left[\exists t \geq 1, \neg\phi^t_{X_{k_i}}\right] && \text{(union bound)} \\
&\quad = \sum_{i \in \mathsf{B}_j} \Pr\left[\exists t \geq 1, |\overline{\mathbb{E}}_t[X_{k_i}] - \mathbb{E}_t[X_{k_i}| > \epsilon_{X_{k_i}}(t)\right] && \text{(definition of } \phi^t_{X_{j_i}}\text{)} \\
&\quad \leq \sum_{i \in \mathsf{B}_j} \delta_{X_{k_i}} && \text{(elementary subexpressions)} \\
&\quad \leq \delta_{\psi_k} && \text{(applying 11 for } t = \mathcal{T}\text{)}
\end{aligned}$$

Thus, $b_{\psi_k}(t)$ is a $1 - \delta_{\psi_k}$ confidence sequence for b_{ψ_k} □

D OPTIMALITY

Proof. Under A_δ, at the stopping time $\mathcal{T}^+$, $\delta^+_i = \Delta/n$, with

$$\sum_{i=1}^{n} \delta^+_i = \Delta.$$

As δ_i^+ are propagated using INFER (without constraint rules), we know that they must satisfy the constraints of the optimization problem 2. At time $\mathcal{T}^+$ AVOIR would find solution δ_i^* such that minimizes $\sum_{i=1}^{n} \delta_i$.

$$\sum_{i=1}^{n} \delta_i^* \leq \sum_{i=1}^{n} \delta_i^+ = \Delta$$

Thus, AVOIR would terminate at step $\mathcal{T}^+$, but may find a feasible solution at an earlier step, i.e. $\mathcal{T} \leq \mathcal{T}^+$.

□

Corollary 4.2. *Under mild conditions, AVOIR terminates in finite steps with an assertion over the required specification.*

Proof. We know that the stopping time $\mathcal{T} \leq \mathcal{T}^+$, the stopping time for AVOIR. Thus, AVOIR would terminate whenever Verifiar can. For completeness, we provide the conditions under which Verifair terminates.

- For every subexpression C_k occurring in the specification such that it is involved in the inverse or inverse constr. rules (i.e., $\overline{\mathbb{E}}[C_k]^{-1}$), $\overline{\mathbb{E}}[C_k] \neq 0$, $C_k \neq 0$
- For every subexpression C_k such that it occurs a True/False type inequality (such as $C_k > c$), $\overline{\mathbb{E}}[C_k] \neq c$, $C_k \neq c$

□

E IMPLEMENTATION

Algorithm 1 AVOIR Algorithm

Input: Δ, ψ ▷ Δ, Specification
Output: T_s time step when the value of ψ can be guaranteed with probability $\geq 1 - \Delta$

```
for X_i ∈ ψ do
    δ_{X_i} = Δ                                  ▷ Set initial value ∀i
    S_{X_i} = 0                                  ▷ Sum of observations
    n_{X_i} = 0                                  ▷ Number of observations
end for
T = 0                                            ▷ Time step
Initialize OPT_ψ                                 ▷ Initialize Optimization Problem (Fig. 5)
procedure OBSERVE(X)
    for X_i ∈ X do
        S_{X_i} = S_{X_i} + X_i
        n_{X_i} = n_{X_i} + 1
        E̅[X_i] = S_{X_i} / n_{X_i}
        Initialize δ_{X_i} as a symbolic variable
        Assign ϵ(δ_{X_i}, n_{X_i}) symbolic variable
    end for
    Propagate δ_{X_i} using the inference rules
    Initialize constraints g_K in OPT_ψ using the computed values
    δ*_T = Solve(OPT_ψ)
    if δ*_T ≤ Δ then
        δ_{X_i} = δ*_T[X_i]
        return T_s = T
    end if
    T = T + 1
end procedure
```

We built a python library to create specifications that can be implemented as a decorator over decision functions. The front end interactive application was implemented using streamlit[6] and the visualizations were built in Vega Satyanarayan et al. (2015). Each term in the DSL is implemented through a corresponding python class. New input/output observations are monitored to update all the terms in a specification. Inference for evaluating the value and bounds is carried out via operator overloading in these classes. In line with previous work (Albarghouthi et al., 2017; Bastani et al., 2019; Albarghouthi & Vinitsky, 2019) on distributional verification, we use rejection sampling for conditional probability estimation.

E.1 Visual Analysis

Using our specification framework as a backend, we built an interactive application for analysis and refinement of specifications provided in our grammar. Given a user provided machine learning model, dataset, and specification the application simulates a stream of observations to the provided model. Following the simulation, a visualization is provided that represents the specification as a syntax tree where each node of the tree corresponds to an element of our grammar. Figure 6 shows the visualization.

Note that for each observation made by our machine learning model, the specification is evaluated to check for violations. Each grammar element that makes up the specification is evaluated as well, and thus each grammar element is associated with the value it evaluates to for a given observation. For specifications `<spec>`, there is a boolean value associated with each observation, whereas an expectation term, `<ETerm>`, is associated with a real value. By selecting one of the nodes in the syntax tree, a user can see a plot of the evaluation values associated with the selected grammar element. We call these plots evaluation plots and two can be observed at a time each with shared scales along the horizontal axis which denotes observations over time. This allows for comparison of multiple grammar elements. The ability to analyze and compare these evaluation values provides context surrounding specification violations, and assists the user in deciding how to refine a specification. The case studies in section 4 demonstrate the usefulness of the context provided by these visualizations.

F AVOIR in Database Setting

In the database literature researchers Nargesian et al. (2021), have explored an approach to tailoring data integration strategies to ensure that the data set used for analysis has an appropriate representation of relevant (demographic) groups and it meets desired distribution requirements. The authors describe how to acquire such data in an approximate cost-optimal manner for several realistic settings. This work is orthogonal to our work and yet AVOIR can potentially integrate with the authors approach to examine if fairness criteria are being met during the integration process. In other studies on fairness researchers Yang et al. (2018); Asudeh et al. (2019); Sun et al. (2019), have considered the problem of personalized fair ranking functions and discuss approaches to determine if a proposed ranking function satisfies a set of desired fairness criteria and, if it does not, to suggest modifications that do. AVOIR attempts to solve a more general purpose problem (not limited to any particular fairness criteria) and is agnostic to the specific model (treats it as a blackbox). While we have not examined the performance of AVOIR for fair ranking problems, it is something we plan to examine in the future.

To demonstrate how AVOIR can be integrated within a database system we use pandas[7] dataframes to simulate the application of AVOIR in the database setting. Specifically, we wrap pandas dataframes with a python 'Database' class, and provide a query mechanism to create materialized views. Queries are provided in the form of python functions that take a dataframe as input and output a corresponding dataframe. The corresponding view thus generated can be updated with insertion/update/deletion of data. The specification is added as a decorator inside the refresh function, allowing AVOIR to track specifications in a database setting. Note that this tie-in with pandas is only for ease of implementation; the inference engine and optimization can be extended to any database engine.

[6] https://streamlit.io/

[7] https://pandas.pydata.org/

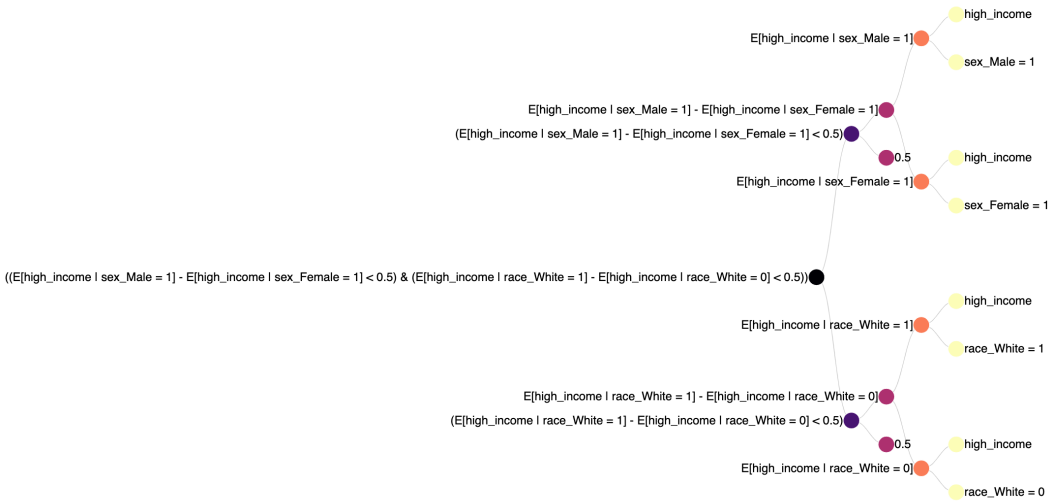


Figure 6: Tree of initial specification before refinement in the adult income dataset.

G Additional Case Studies

G.1 Materialized views

A materialized view is constructed by querying the dataset to select for employees of the federal government. We simulate the materialized view using a pandas[8] dataframe wrapped in a python class to monitor updates and run AVOIR for any monitored specification.

G.2 Interaction through Vega

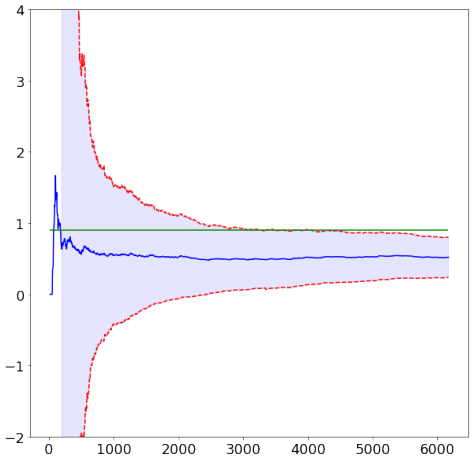
Figure 6 shows a subtree of the specification visualized through Vega. A developer analyzing this spec can click on the top pink node to see the evolution of the sex fairness part of the specification and superimpose the threshold. The threshold is set to be evaluated with every 5 new data points added to the materialized view. Clicking on the corresponding element in the right subtree, the developer can see Figure 4b.

G.3 COMPAS Risk Assessment via Materialized Views

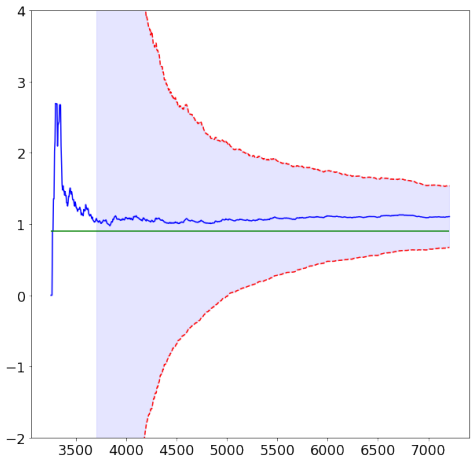
The Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) recidivism risk score data is a popular dataset for assessing machine bias of commercial tools used to assess a criminal defendant's likelihood to re-offend. The data is based on recidivism (re-offending) scores derived from a software released by Northpointe and widely used across the United States for making sentencing decisions. In 2016, Angwin et al. (2016) released an article and associated analysis code critiquing machine bias associated with race present in the COMPAS risk scores for a set of arrested individuals in Broward County, Florida over a period of two years. The analysis concluded that there were significant differences in the risk assessments of African-American and Caucasian individuals. Northpointe pushed back in a report (Dieterich et al., 2016) strongly rejecting the claims made by the ProPublica article; instead, they claimed that Angwin et al. (2016) made several statistical and technical errors in the report. In this case study, we use AVOIR to study the claims made by the two aforementioned reports. First, we start with the data released by ProPublica and load it into a pandas-simulated DB. We then create a materialized view that corresponds to the preprocessing steps used in the publicly available ProPublica analysis notebook[9]. We look at "Sample A" (Dieterich et al.,

[8] https://pandas.pydata.org/

[9] https://github.com/propublica/compas-analysis



(a) (ProPublica) COMPAS, "Sample A" False Positive Rate Bias specification required to *above* the $10\% \implies 0.9$ threshold converges to a value that can be guaranteed to be *under* the required threshold.



(b) (Northpointe) "Sample B" analysis done by Northpointe using False Discovery Rate that opposed the ProPublica reports.

Figure 7: COMPAS dataset case study.

2016), where the analysis of the "not low" risk assessments using a logistic regression model reveals a high coefficient associated with the factor associated with race being African-American. In terms of a fairness metric, this corresponds to false positive rate (FPR) balance (predictive equality) (Verma & Rubin, 2018) metrics. The associated specification in AVOIR grammar would be

$$\mathrm{E}[\,\mathrm{hrisk} \mid \mathrm{race{=}African{-}American}\ \&\ \mathrm{recid{=}0}]\ /\ \mathrm{E}[\,\mathrm{hrisk} \mid \mathrm{race{=}Caucasian}\ \&\ \mathrm{recid{=}0}] < 1.1$$

Where `hrisk` is an indicator for high risk assessments made by the *black-box* COMPAS tool as defined by Angwin et al. (2016), `recid` is an indicator for re-offending within 2 years of first arrest, and a 10%-rule is used as the threshold. We choose a failure threshold probability of $\Delta = 0.1$, with the optimization run after every batch of 5 samples. AVOIR finds that when the decisions are made in a sequential, online fashion, the assertion for violation of the specification cannot be made with the required failure guarantee.

By analyzing the components using the visualization tool, one can glean that AVOIR is unable to optimize since the lower FPR in the denominator (FPR for Caucasian individuals) increasing the overall variance and limiting the ability to optimize for guarantees. We follow this analysis by using the reciprocal specification, i.e.,

$$\mathrm{E}[\,\mathrm{hrisk} \mid \mathrm{race{=}Caucasian}\ \&\ \mathrm{recid{=}0}]\ /\ \mathrm{E}[\,\mathrm{hrisk} \mid \mathrm{race{=}African{-}American}\ \&\ \mathrm{recid{=}0}] > 0.9$$

we find that indeed, the specification is violated with a confidence of over $1 - \Delta = 0.9$ probability, and AVOIR can detect this violation within about half the number of available assessments (3350 steps) when run in an online setting. Figure 7a demonstrates the progression of the tracked expectation term. Thus, if deployed with the corrected specification, AVOIR would be able to alert Northpointe of a violation/potentially-biased decision making tool.

The Northpointe report (Dieterich et al., 2016) makes several claims about the shortcomings of this analysis, but one of the primary claims is that Angwin et al. (2016) used an analysis based on "Model Errors" rather than "Target Population Errors". In Fairness metric terms, this refers to the difference between a False Positive Rate (FPR) balance vs False Discovery Rate (FDR) balance, i.e. balancing for predictive parity over predictive equality. In probabilistic terms, the difference amounts to comparing $P(\hat{Y} = 1 | Y = 0, g = 1, 2)$ (FPR) vs $P(Y = 0 | \hat{Y} = 1, g = 1, 2)$ (FDR), where $\hat{Y}$ refers to the decision made by the algorithm, Y refers to the true value, and $g = 1, 2$ reflects group mem-

bership (Verma & Rubin, 2018). This analysis is run on the dataset subset dubbed "Sample B". To test their hypothesis, we run reproduce the corresponding preprocessing steps and run both versions (numerator and denominator being Caucasian) versions of the corresponding specification under the same setup as earlier. We find that despite the point estimate being within the required threshold, neither version can be guaranteed with the required confidence with the given data. Due to paucity of space, we describe only one of the two variants with the corresponding figure (Figure 7b).

$$\mathrm{E}[\,\mathrm{recid}{=}0 \mid \mathrm{race}{=}\mathrm{Caucasian}\ \&\ \mathrm{hrisk}\,]\ /\ \mathrm{E}[\,\mathrm{recid}{=}0 \mid \mathrm{race}{=}\mathrm{African{-}American}\ \&\ \mathrm{hrisk}] > 0.9$$

We note that the Northpointe report (Dieterich et al., 2016) does not provide confidence intervals for their claim either. Further, even though the report does not release associated code, the point estimates of the False Discovery Rates (FDRs) match those present in the report which increases our confidence in our AVOIR-based analysis.

The back and forth exchange has been the subject of much discussion in both academic and journalistic publications (Feller et al., 2016; Washington, 2018). Seminal work by Kleinberg et al. (2017) proved the impossibility of simultaneously guaranteeing certain combinations of fairness metrics. While AVOIR cannot solve this problem, its usage can help provide explicit guarantees on defined metrics. The specification grammar also provides a simple mechanism for independent replication of claims. We conclude this case study by noting that AVOIR lends itself to successful analysis that is not possible with the Verifair implementation available online.