Proof-checking bias in labeling methods

Giuseppe Primiero[1,*], Fabio D'Asaro[2]

[1] Logic, Uncertanty, Computation and Information Group, Department of Philosophy, University of Milan, Via Festa del Perdono, 7, Milano, 20122, Italy

[2] Ethos Group, Department of Human Sciences, University of Verona, Via S. Francesco, 22, Verona, 37129, Italy

Abstract

We introduce a typed natural deduction system designed to formally verify the presence of bias in automatic labeling methods. The system relies on a "data-as-terms" and "labels-as-types" interpretation of formulae, with derivability contexts encoding probability distributions on training data. Bias is understood as the divergence that expected probabilistic labeling by a classifier trained on opaque data displays from the fairness constraints set by a transparent dataset.

Keywords

Bias, Classifiers, Formal Verification

1. Introduction

The formal verification of AI systems is a growing field of research, and its applications are essential to the deployment of safe systems in several domains with high societal impact. This theme is attracting more and more attention from the community at large, not just the formal verification specialists, see e.g. the recent [1]. Essential to this task is to reason about probabilistic processes, in order to accommodate the uncertainty and non-deterministic behavior characteristic of AI systems, and to model desirable forms of accuracy and trustworthiness.

This objective can be formulated in terms of formal methods to prove trustworthy properties of interest, see e.g. [2, 3, 4]; or with linear temporal logic properties defined over Markov decision processes, e.g. with reinforcement learning methods [5] or with imprecise probabilities [6]. A different approach consists in exploiting a proof-theoretic semantics and the tradition of lambda-calculi and the Curry-Howard isomorphism to interpret computational processes as proof terms and output values as their types. Under the general heading of proof-checking techniques, we can identify untyped λ-calculi [7, 8, 9], probabilistic event λ-calculi [10], calculi for Bayesian learning [11], and calculi with types or natural deduction systems [12, 13, 14]. In this latter tradition, the calculus TPTND [15, 16] offers a novel interpretation of a probabilistic typed natural deduction system specifically designed to formally check the post-hoc validity of a computational process executed under a possibly opaque distribution of outputs against a

BEWARE: Joint BRIO, ME&E and AWARE Workshop @ AIxIA 2022, November 28 - December 2, 2022, University of Udine, Udine, Italy

*Corresponding author.

giuseppe.primiero@unimi.it (G. Primiero); fabioaurelio.dasaro@univr.it (F. D'Asaro)
https://sites.unimi.it/gprimiero/ (G. Primiero); https://www.fabiodasaro.com (F. D'Asaro)
0000-0003-3264-7100 (G. Primiero); 0000-0002-2958-3874 (F. D'Asaro)

transparent and desirable model. Here trustworthiness is interpreted as an admissible distance in the probability assigned to an observed output when compared to the probability that the intended behavior should display.

The threats that AI systems pose to reliable and safe knowledge are varied and, among these, the potential bias of is a rising phenomenon. Dating back to the analysis of bias in deterministic systems see e.g. [17], several recent studies have shown that training and evaluation data for classification algorithms are often biased. [18] illustrates algorithmic discrimination based on gender and race; [19] lists different types of bias for data and algorithms. [20] shows that the most well-known AI datasets are full of labeling errors. From a formal viewpoint, the trustworthiness of an AI system may be considered also from the point of view of the absence of bias. Given an automated labeling method, one might ask whether it displays any bias, and if so, whether such level is below an admissible maximum potential threshold. Under this perspective, a new task is the design of systems that will make it possible to formally verify the absence of bias, or its presence within limits considered admissible in the application domain. Consider as an example the potential of automatic AI-based classification in recruiting, where the aim is to automatically select the best applicants out of huge pools of resumè. Notoriously this is not without risks, as shown by the infamous Amazon case, where the selection process was non gender-neutral due to the training of models based on patterns in resumes submitted over a 10-year period in which most applicants were men, thereby reflecting and consolidating male dominance in the tech industry, [21]. An important task is therefore to devise methods were gender-balance (or similar sensitive attributes) are treated as safety properties to be automatically checked.

In this paper we present TPTND-KL, a system re-interpreting TPTND to reason about labeling of random variables and datapoints, with the task of inferring the presence of bias in classification methods. The underlying interpretation of formulas can be dubbed ”datapoints-as-terms” and ”labels-as-types”. In this context, we understand bias as a measure of the entropy between a given opaque probabilistic assignment of labels to a datapoint and the results given by a distribution of labels assumed to be correct, or at least desirable. This interpretation is inspired by [22]. The rest of the paper is structured as follows. In Section 2 we introduce the language, stressing the novel interpretation with respect to the previous version of the system and the introduction of the new *Bias* operator. In Section 3 we reformulate the proof system under the novel isomorphism between syntactic terms and semantic interpretation. In Section 4 we extend the proof system with rules for checking the presence of bias, with particular attention to the novel distance measure chosen for this application. In Section 5 we provide a detailed example to show the checking method at work. We conclude with limitations of this approach and further topics of research.

2. Formal Preliminaries

The logic TPTND-KL is a typed natural deduction system built from the following syntax:

Definition 1 (Syntax).

$$\begin{array}{l}
\texttt{Variable Set} := \mathrm{x} \mid \mathrm{x}_{\mathrm{T}} \mid \tilde{\mathrm{x}}_{\mathrm{T}} \mid \mathrm{x}^{*}_{\mathrm{T}} \mid \langle \mathrm{X}, \mathrm{X} \rangle \mid \mathit{fst}(\mathrm{X}) \mid \mathit{snd}(\mathrm{X}) \mid [\mathrm{X}]\mathrm{X} \mid \mathrm{X.X} \mid \mathrm{X.T} \\
\texttt{Dataset} := \mathrm{t} \mid \langle \mathrm{T}, \mathrm{T} \rangle \mid \mathit{fst}(\mathrm{T}) \mid \mathit{snd}(\mathrm{T}) \mid [\mathrm{X}]\mathrm{T} \mid \mathrm{T.T}' \mid \mathit{Bias}(\mathrm{T}) \\
\texttt{Label Set} := \alpha \mid \alpha_r \mid \bot \mid (\alpha \times \beta)_r \mid (\alpha + \beta)_r \mid (\alpha \to \beta)_{[r']r} \\
\texttt{Assignments} := \{\} \mid \mathrm{C}, x : \alpha \mid \mathrm{C}, x : \alpha_a
\end{array}$$

It includes: (random) variables of the form x_t with an associated datapoint t and annotated with indicators for being the correct labeled variable $\tilde{x}$ or the wrongly labeled variable x^*, and composed as pairs, first and second projection, abstraction and application (also with terms); terms t standing for datapoints, composed as pairs, first and second projection, abstraction (with respect to a variable of interest) and application, and taken as argument of a *Bias* operator; a labelset with labels α decorating variables and terms and indexed by a real value r, composed as conjunction, disjunction and implication; finally, an assignment function which lists a probability distribution C of labels to variables.

Example formulas of TPTND-KL have the following form:

- $x : \alpha \vdash y : \beta_{0.1}$: the random variable y has probability 0.1 of having label β, provided x has label α, i.e. $P(y = \beta \mid x = \alpha) = 0.1$.
- $\Gamma \vdash p_{100} : F_{\widetilde{0.35}}$: under distribution Γ (e.g. of a given population), for 100 people one expects label $female$ to be assigned 35% of the times.
- $\Gamma \vdash p_{100} : F_{30}$: for 100 people in our dataset the label $female$ was assigned 30 times.
- $\tilde{y}_t : M_{0.5}, y^*_t : F_{0.35} \vdash t : M_{0.3}$: a datapoint t is labeled F with probability 0.3 provided that t is wrongly labeled M with probability 0.5, and correctly F with probability 0.35, i.e. $P(t = M) = 0.3$, $P(y_t = M \mid \tilde{y}_t \neq M) = 0.5$ and $P(y_t = F \mid \tilde{y}_t = F) = 0.35$.

The main novelty of TPTND-KL (and what differentiates it from its ancestor system TPTND) is the interpretation of its formulae in the style of ”datapoints-as-terms” and ”labels-as-types”, and the ability to express the probability of label assignment under a given distribution of labels for random variables, including true and false positive rates of labeling.

3. Proof system

Judgements are defined through rules for the associated logical operator.

The construction rules for probability distributions of the label set mimic those of TPTND [15], see 1. Contexts are generated by judgements of the form $\vdash x : \alpha_r$ and $\vdash x_t : \alpha_r$, expressing assignments of probabilities to labeled variables for given terms. An empty set {} represents the base case (rule ”base”) to define distributions. Random variables with given theoretical probability values assigned to labels can be added to a distribution as long as they respect the standard additivity requirement on probabilities (”extend”). Labels are intended as exclusive (a single label for a datapoint) and categorical (at least a label for each datapoint). To denote an unknown distribution (”unknown”) we use the (transfinite) construction assigning to all values of interest the interval of all possible probability values, provided the additivity condition of ”extend” is preserved. Such a construction expresses formally the probability distribution underlying an opaque classifier.

$$\frac{}{\{\} :: \mathit{distribution}}\ \text{base}$$

$$\frac{\Gamma :: \mathit{distribution} \qquad x : \alpha_a \notin \Gamma \qquad \sum_p x : \rho_p \in \Gamma \leq 1}{\Gamma, x : \alpha_a :: \mathit{distribution}}\ \text{extend}$$

$$\frac{\alpha :: \mathit{label} \qquad \ldots \qquad \omega :: \mathit{label}}{\{x : \alpha_{[0,1]}, \ldots, x : \omega_{[0,1]}\} :: \mathit{distribution}}\ \text{unknown}$$

Figure 1: Distribution Construction

$$\frac{}{\Gamma, x_t^* : \alpha_a, \tilde{x}_t : \beta_b \vdash t : \alpha}\ \text{label assignment}$$

$$\frac{\Gamma, x_t : \alpha_a, x_t : \beta_b \vdash t : \alpha}{\Gamma \vdash t : (\alpha + \beta)}\ \text{I+}$$

$$\frac{\Gamma \vdash t : (\alpha + \beta) \qquad \Gamma \vdash t : (\alpha \to \bot)}{\Gamma \vdash t : \beta}\ \text{E+}_R$$

$$\frac{\Gamma \vdash t : (\alpha + \beta) \qquad \Gamma \vdash t : (\beta \to \bot)}{\Gamma \vdash t : \alpha}\ \text{E+}_L$$

$$\frac{\Gamma, x_t^* : \alpha_a \vdash t : \beta}{\Gamma \vdash [x^*]t : (\alpha \to \beta)_a} \qquad \frac{\Gamma \vdash [x]t : (\alpha \to \beta)_a \qquad \vdash t : \alpha}{\Gamma \vdash t.[t : \alpha] : \beta}$$

Figure 2: Categorical labeling rules.

Under the interpretation of TPTND-KL, categorical label assignment for each datapoint is made under a probability distribution with true and false positive rates for exclusive labels, see Figure 2. The ”label assignment” rule, axiomatically declares that the datapoint t is classified as α, under a distribution of labels which assigns true positive rate a for t to be α and false positive rate b for t to be β. This rule states the observed behavior of a classifier under a transparent confusion matrix for the available label set. In this version of our calculus, we do not include the joint probability of distinct labels under independent distributions. $I+$ allows to express that a single datapoint is assigned exclusively one of two (or more) labels included in the same distribution; correspondingly $E+$ extracts the unique label from such a disjoint set, when the other options have non-positive probability ($\bot$). Finally, $\to$ expresses the dependency of label assignment from feature satisfaction: if the label β for term t depends from the label α being correctly assigned to t with a certain probability a, then t has label $(\alpha \to \beta)$ with probability a; from this, provided t is labeled α, then term t is labeled β. Consider, for example, the probability of being assigned a label $female$ from a classifier depending on the probability that the feature *long_hair* be greater than some positive value (and in a bad classifier that probability might be 1).

The rules can now be generalised for the expected assignment of labels for a given population of interest, see Figure 3. Again, these rules are adapted from their counterparts in TPTND. The "expected labeling" rule axiomatically declares that under a transparent distribution of labels which assigns true positive rate a for t to be α and false positive rate b for t to be β, in a population of n datapoints t the label α is expected to be assigned with probability a (the expected probability being denoted by the tilde). Note that this rule sets an expected assignment given a known probability distribution, which can be taken to be the one on which a classifier is trained: such expected probabilistic classification can be designed to embed specific fairness constraints, like demographic parity or equal opportunity. In the following, we will use this expected behavior as a constraint to check the fairness of a given unknown classifier. The "sampling" rule says that the frequency value f of label α appears for n datapoints t, where f is the number of cases in which α is actually assigned in those n cases by means of n occurrences of the "label assignment" rule. As a general case, the distribution under which t is labeled can be taken to be unknown, hence we consider the simple observation of a classifier at work without knowing its inner structure. This will turn out to be crucial in our bias checking rule. The "update" rule serves the purpose of considering multiple classifications to render the change in frequency of any given label with different sets of members of a given population of interest. Rule I+ introduces disjunction of possible labels: intuitively, if under a distribution Γ a population of n elements t has assigned label α with an expected probability $\tilde{a}$, and label β is assigned with expected probability $\tilde{b}$, then the expected probability of label α or label β is $\tilde{a} + \tilde{b}$. By E$+_R$ (respectively E$+_L$): if under a distribution Γ a population of n datapoints t is assigned label α or label β with expected probability $\tilde{c}$, and the former (respectively, the latter) label has probability $\tilde{a}$ (respectively, $\tilde{b}$), then in that population the latter (respectively, the former) label will be assigned with probability $\tilde{c} - \tilde{a}$ (respectively $\tilde{c} - \tilde{b}$). This formally expresses categoricity of labeling. The rule I$\to$ now says that: if label β is assigned with expected probability $\tilde{b}$ in a population of size n provided in that population label α is correctly assigned with probability a, then we express with such dependency with a term $[x]t_n$ which is assigned label $(\alpha \to \beta)$ with probability $\tilde{b}$ provided a. The corresponding elimination E$\to$ allows to verify the expected probability of label β: it considers a term $[x]t_n$ labeled $(\alpha \to \beta)$ with expected probability $\tilde{b}$ depending on the probability a of label α to be assigned to a given element x (e.g. a feature), and when such labeling occurs with probability $\tilde{a}$, it computes the expected probability $\tilde{a} \cdot \tilde{b}$ for n datapoints in the population to be labeled β. Notice that this rule might be made invalid if constrained in a manner that the expected frequency of α must be higher than a given threshold, in order for the probability of β to be assigned be positive.

4. Checking for Bias

We now provide a set of introduction and elimination rules for the *Bias* operator ranging over a label assignment. Let us consider a binary label set $L = \{\alpha, \beta\}$, a variable set $X : \{x_1, \dots, x_n\}$ and a data set $D := \{t, u, v, \dots, z\}$. Consider now the expression

$$\vdash \tilde{x}_t : \beta_b$$

saying that β is the wrong label for data point t with probability b. This expresses therefore

$$\frac{}{x_t^* : \alpha_a, \tilde{x}_t : \beta_b \vdash t_n : \alpha_{\tilde{a}}}\ \text{expected labeling}$$

$$\frac{\Gamma \vdash t^1 : \alpha^1 \qquad \dots \qquad \Gamma \vdash t^n : \alpha^n}{\Gamma \vdash t_n : \alpha_f}\ \text{sampling}$$

$$\text{where } f =\mid \{\alpha^1, \dots, \alpha^n \mid \alpha^i = \alpha\} \mid$$

$$\frac{\Gamma \vdash t_n : \alpha_f \qquad \Gamma \vdash t_m : \alpha_{f'}}{\Gamma \vdash t_{n+m} : \alpha_{f*(n/(n+m))+f'*(m/(n+m))}}\ \text{update}$$

$$\frac{\Gamma \vdash t_n : \alpha_{\tilde{a}} \qquad \Gamma \vdash t_n : \beta_{\tilde{b}}}{\Gamma \vdash t_n : (\alpha + \beta)_{\tilde{a}+\tilde{b}}}\ \mathsf{I+}$$

$$\frac{\Gamma \vdash t_n : (\alpha + \beta)_{\tilde{c}} \qquad \Gamma \vdash t_n : \alpha_{\tilde{a}}}{\Gamma \vdash t_n : \beta_{\tilde{c}-\tilde{a}}}\ \mathsf{E+}_L$$

$$\frac{\Gamma \vdash t_n : (\alpha + \beta)_{\tilde{c}} \qquad \Gamma \vdash t_n : \beta_{\tilde{b}}}{\Gamma \vdash t_n : \beta_{\tilde{c}-\tilde{b}}}\ \mathsf{E+}_R$$

$$\frac{\Gamma, x_t^* : \alpha_a \vdash t_n : \beta_{\tilde{b}}}{\Gamma \vdash [x^*]t_n : (\alpha \to \beta)_{[a]\tilde{b}}}\ \mathsf{I\to}$$

$$\frac{\Gamma \vdash [x^*]t_n : (\alpha \to \beta)_{[a]\tilde{b}} \qquad y_u : \alpha_a \vdash u_n : \alpha_{\tilde{a}}}{\Gamma \vdash t_n.[u_n : \alpha] : \beta_{\widetilde{a \cdot b}}}\ \mathsf{E\to}$$

Figure 3: Population Labeling Rules

the False Positive Rate for t having label β. And the expression

$$\vdash x_t^* : \alpha_a$$

saying that α is the correct label for data point t with probability a. This expresses therefore the True Positive Rate for any datapoint t with label α. Here $b = 1 - a$. Now let us assume we have a probability distribution for L in which such information is available over a population, i.e.

$$\Gamma, x^* : \alpha_a, \tilde{x} : \beta_b :: \mathit{distribution}$$

Assume the function h implemented by a classifier using this distribution is a desirable one, meaning that the expected classification by an application of the rule ”expected labeling” over the classes denoted by the available labels will behave following this distribution, which might reflect a fairness constraint we require on a protected group. For completeness, we denote the desirable Γ satisfying our fairness constraints as

$$\frac{\Gamma :: fair\_distribution \qquad \Delta \vdash t : \alpha_f \qquad D_{KL}(\Delta \parallel \Gamma) = \epsilon > \pi}{\Gamma, \Delta \vdash Bias(t : \alpha_f)}\ \mathsf{IB}$$

$$\frac{\Delta \vdash Bias(u_n : \alpha_f)}{\Gamma, x_u^* : \alpha_{[0,1]-[a-\epsilon(n),a+\epsilon(n)]} \vdash u_n : \alpha_f}\ \mathsf{EB}$$

Figure 4: Bias Checking Rules

$$\Gamma :: fair\_distribution$$

Consider now the observation for a given data point t assigning label α with frequency f:

$$\Delta \vdash t : \alpha_f$$

which we might consider biased if the divergence of f under Δ is significant from what is dictated by the constraint on the labels encoded in Γ:

Definition 2 (Bias labeling). *The assignment of label α shown with frequency f for a population of n datapoints t under the distribution generated by the possibly opaque training set Δ is biased if the divergence of f from the probability a generated by the correct label assigned by a transparent distribution Γ is greater than a given threshold considered appropriate for the labeling task at hand.*

We now use the Kullback-Leibler divergence, which notoriously is a measure of how the probability distribution Δ is different from Γ, or the expected excess surprise from using Γ as a model when the actual distribution is Δ. Fixing a constraint that defines the fairness of Γ, and for each term t and label α common to Δ, Γ, we obtain:

$$D_{\mathrm{KL}}(\Delta \parallel \Gamma) = \sum_{\alpha \in \Delta} (\Delta \vdash u : \alpha_f) \log \left(\frac{\Delta \vdash t : \alpha_f}{\Gamma, x_t^* : \alpha_a :: fair\_distribution} \right)$$

Using this measure, we want to check whether its value surpasses some reference threshold π which is considered appropriate for the classification task. This means considering sufficiently low values of π for sensitive classifications, e.g. gender or other socially and culturally relevant properties. Definition 2 is then expressed by Rule IB in Figure 4. In this rule ϵ is the sum of the divergence between Δ and Γ computed for each label α common to both. On the other hand, in the presence of a population of n datapoints where the frequency f of label α under distribution Δ is considered biased, we infer a family of probability values in the interval $[0, 1]$ excluding the interval $[a - \epsilon(n), a + \epsilon(n)]$ for output $\alpha \in \Gamma$ within which $u_n : \alpha_f$ can be correctly sampled from Δ. Note that this rule will not allow to derive a fair value for label α, but rather determine the conditions of the distribution under which the current frequency can be considered admissible.

5. Example

Consider context $\Gamma = \{x_t : M_{1/3}, x_t : F_{2/3}\}$, representing the reference distribution of gender in a population of interest in which 1/3 of the population is male and 2/3 is female.

Assume now a classifier trained on some data that distributes according to Δ, which is inaccessible, but which we know is – or can assume to be – similar to the one described by Γ. The classifier on Δ when used displays the following behavior:

$$\Delta \vdash t_{140} : M_{60}$$
$$\Delta \vdash t_{140} : F_{80}$$

that is, out of 140 datapoints, 60 are labelled as M and 80 are labelled as F. For readability we write $\Gamma(M) = 1/3$, $\Gamma(F) = 2/3$, $\Delta(M) = 60/140 = 3/7$ and $\Delta(F) = 80/140 = 4/7$. We want to check whether our classifier using Δ is fair with respect to the constraints set by Γ. We calculate the Kullback–Leibler divergence as follows:

$$\begin{aligned} D_{KL}(\Delta \parallel \Gamma) &= \sum_{x \in \{F,M\}} \Delta(x) \cdot \log_2 \left(\frac{\Delta(x)}{\Gamma(x)} \right) && (1) \\ &= \Delta(F) \cdot \log_2 \left(\frac{\Delta(F)}{\Gamma(F)} \right) + \Delta(M) \cdot \log_2 \left(\frac{\Delta(M)}{\Gamma(M)} \right) && (2) \\ &= \frac{4}{7} \cdot \log_2 \left(\frac{4/7}{2/3} \right) + \frac{3}{7} \cdot \log_2 \left(\frac{3/7}{1/3} \right) \approxeq 0.028 && (3) \end{aligned}$$

To determine when we would consider the classification produced over Δ biased, we need to set a threshold π. Let $\pi = 0.2$. Then, since $D_{KL}(\Delta \parallel \Gamma) < \pi$ we can conclude that $\Gamma, \Delta \vdash Fair(t_{140} : M_{60})$, i.e. that the classification of male inidivduals produces by a classifier under the distirbution Δ is fair when compared against Γ.

To further illustrate this example, let us re-consider Equation (2) but this time assume that we don't know how many data-points out of the 160 are labelled as M, i.e.

$$\Delta \vdash t_{140} : M_n$$
$$\Delta \vdash t_{140} : F_{140-n}$$

Recall that $\Gamma(M) = 1/3$ and $\Gamma(F) = 2/3$, and consider that since we are dealing with a binary classification task, $\Delta(F) = 1 - \Delta(M)$. Then we may rewrite Equation (2) as

$$D_{KL}(\Delta \parallel \Gamma) = (1 - \Delta(M)) \cdot \log_2 \left(\frac{3(1 - \Delta(M))}{2} \right) + \Delta(M) \cdot \log_2 (3 \cdot \Delta(M))$$

and if we let $\Delta(M)$ vary in the interval $[0, 1]$, we can plot $D_{KL}(\Delta \parallel \Gamma)$ as in Figure 5, which provides a more concrete illustration of which values are considered to be biased in our domain.

6. Conclusion

In this paper we have introduced TPTND-KL, a natural deduction system where formulas express the expected labeling of data, based on probability distributions of labels over training sets. We have shown how to model proof-theoretically a checking procedure to measure the

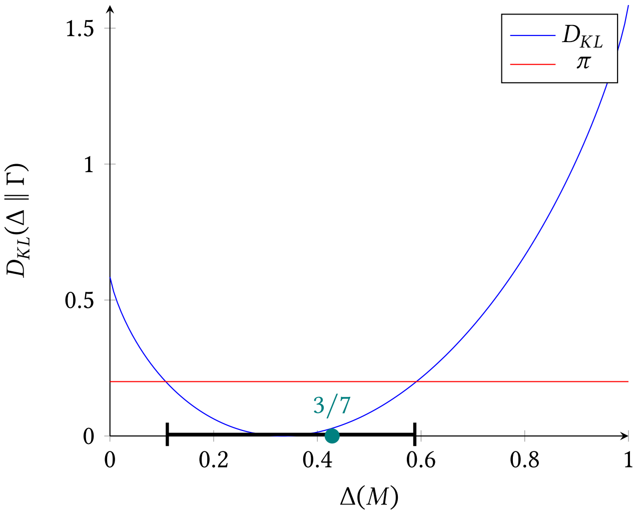

Figure 5: The Kullback-Leibler Divergence for a fixed reference distribution of $1/3$ males and $2/3$ females, and for a classifier that classifies a fraction $\Delta(M)$ of the population as male. If we set a threshold $\pi = 0.2$ (depicted as the red line in the figure) we get that $\Delta(M)$ should belong to the interval $[0.11, 0.59]$ in order to be considered unbiased (see the thick black line on the x axis). Note that, for the particular case where $\Delta(M) = 3/7$ as in our previous example, the bias is in fact acceptable (see the teal dot on the x axis).

distance that a given opaque classifier displays from the constraints set on the expected labeling by a transparent probabilistic distribution.

Our research leaves room for further developments. To start with, the example used throughout this paper is somewhat simplistic. Instead, a more precise characterization of the types of biases and of fairness metrics that we can model within our language is required: for instance, one can take advantage of the use of the implication to build dependent variables to model more complex examples where multi-attribute biases and intersectionality occur. It is worth stressing here that currently our framework allows us to deal with *fairness-of-outcome*, as we can only compare the output distribution (i.e., the outcome of the classification process) with a reference distribution that we assume to be fair. It remains to be explored whether other definition of fairness can be modelled without substantially modify TPTND-KL. Other problems that we would like to tackle in a future version of this work are: adapting the framework to regression tasks, as at the moment it only works for classification, since the set of types is tacitly assumed to be finite; and investigate how to set the reference thresholds can be defined or extracted in less arbitrary ways. This latter point will require studying how the framework behaves with real data, and to this end an implementation of this work is very much needed. Another future step is the formulation of meta-threotical results, like progress and termination on the syntactic transformations determined by our proof systems, to prove under which conditions a given measure of bias will never exceed a certain threshold; finally, the integration of the bias checking procedure developed for our system with other non-automatic parameters and information quality criteria would be desirable.

Acknowledgments

This research has been partially funded by the Project PRIN2020 BRIO - Bias, Risk and Opacity in AI (2020SSKZ7R) awarded by the Italian Ministry of University and Research (MUR). The author Fabio Aurelio D'Asaro acknowledges the funding and support of PON "Ricerca e Innovazione" 2014-2020 (PON R&I FSE-REACT EU), Azione IV.6 "Contratti di ricerca su tematiche Green" in the context of the project titled "Il miglioramento dell'algorithmic fairness in ambito assicurativo: Un'analisi concettuale a partire da uno studio di caso".

References

[1] S. A. Seshia, D. Sadigh, S. S. Sastry, Toward verified artificial intelligence, Commun. ACM 65 (2022) 46–55. URL: https://doi.org/10.1145/3503914. doi:10.1145/3503914.

[2] R. Alur, T. A. Henzinger, P.-H. Ho, Automatic symbolic verification of embedded systems, IEEE Trans. Softw. Eng. 22 (1996) 181–201. URL: https://doi.org/10.1109/32.489079. doi:10.1109/32.489079.

[3] M. Z. Kwiatkowska, G. Norman, D. Parker, Prism: Probabilistic symbolic model checker, in: Proceedings of the 12th International Conference on Computer Performance Evaluation, Modelling Techniques and Tools, TOOLS '02, Springer-Verlag, Berlin, Heidelberg, 2002, pp. 200–204.

[4] A. Termine, G. Primiero, F. A. D'Asaro, Modelling accuracy and trustworthiness of explaining agents, in: S. Ghosh, T. Icard (Eds.), Logic, Rationality, and Interaction - 8th International Workshop, LORI 2021, Xi'ian, China, October 16-18, 2021, Proceedings, volume 13039 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 232–245. URL: https://doi.org/10.1007/978-3-030-88708-7_19. doi:10.1007/978-3-030-88708-7_19.

[5] Q. Gao, D. Hajinezhad, Y. Zhang, Y. Kantaros, M. M. Zavlanos, Reduced variance deep reinforcement learning with temporal logic specifications, in: Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 237–248. URL: https://doi.org/10.1145/3302509.3311053. doi:10.1145/3302509.3311053.

[6] A. Termine, A. Antonucci, G. Primiero, A. Facchini, Logic and model checking by imprecise probabilistic interpreted systems, in: A. Rosenfeld, N. Talmon (Eds.), Multi-Agent Systems - 18th European Conference, EUMAS 2021, Virtual Event, June 28-29, 2021, Revised Selected Papers, volume 12802 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 211–227. URL: https://doi.org/10.1007/978-3-030-82254-5_13. doi:10.1007/978-3-030-82254-5_13.

[7] G. Bacci, R. Furber, D. Kozen, R. Mardare, P. Panangaden, D. S. Scott, Boolean-valued semantics for the stochastic λ-calculus, in: A. Dawar, E. Grädel (Eds.), Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018, ACM, 2018, pp. 669–678. URL: https://doi.org/10.1145/3209108.3209175. doi:10.1145/3209108.3209175.

[8] J. Borgström, U. D. Lago, A. D. Gordon, M. Szymczak, A lambda-calculus foundation for universal probabilistic programming, in: J. Garrigue, G. Keller, E. Sumii (Eds.), Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, ICFP

2016, Nara, Japan, September 18-22, 2016, ACM, 2016, pp. 33–46. URL: https://doi.org/10.1145/2951913.2951942. doi:10.1145/2951913.2951942.

[9] P. H. A. de Amorim, D. Kozen, R. Mardare, P. Panangaden, M. Roberts, Universal semantics for the stochastic λ-calculus, in: 36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021, IEEE, 2021, pp. 1–12. URL: https://doi.org/10.1109/LICS52264.2021.9470747. doi:10.1109/LICS52264.2021.9470747.

[10] M. Antonelli, U. D. Lago, P. Pistone, Curry and howard meet borel, in: C. Baier, D. Fisman (Eds.), LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022, ACM, 2022, pp. 45:1–45:13. URL: https://doi.org/10.1145/3531130.3533361. doi:10.1145/3531130.3533361.

[11] F. Dahlqvist, D. Kozen, Semantics of higher-order probabilistic programs with conditioning, Proc. ACM Program. Lang. 4 (2020) 57:1–57:29. URL: https://doi.org/10.1145/3371125. doi:10.1145/3371125.

[12] A. D. Pierro, A type theory for probabilistic λ-calculus, in: From Lambda Calculus to Cybersecurity Through Program Analysis, 2020.

[13] M. Boričić, Sequent calculus for classical logic probabilized, Archive for Mathematical Logic 58 (2019) 119–136.

[14] S. Ghilezan, J. Ivetić, S. Kašterović, Z. Ognjanović, N. Savić, Probabilistic reasoning about simply typed lambda terms, in: International Symposium on Logical Foundations of Computer Science, Springer, 2018, pp. 170–189.

[15] F. A. D'Asaro, G. Primiero, Probabilistic typed natural deduction for trustworthy computations, in: D. Wang, R. Falcone, J. Zhang (Eds.), Proceedings of the 22nd International Workshop on Trust in Agent Societies (TRUST 2021) Co-located with the 20th International Conferences on Autonomous Agents and Multiagent Systems (AAMAS 2021), London, UK, May 3-7, 2021, volume 3022 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021. URL: http://ceur-ws.org/Vol-3022/paper3.pdf.

[16] F. A. D'Asaro, G. Primiero, Checking trustworthiness of probabilistic computations in a typed natural deduction system, CoRR abs/2206.12934 (2022). URL: https://doi.org/10.48550/arXiv.2206.12934. doi:10.48550/arXiv.2206.12934. arXiv:2206.12934.

[17] B. Friedman, H. Nissenbaum, Bias in computer systems, ACM Trans. Inf. Syst. 14 (1996) 330–347.

[18] J. Buolamwini, T. Gebru, Gender shades: Intersectional accuracy disparities in commercial gender classification, in: Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA, volume 81, PMLR, 2018, pp. 77–91.

[19] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM Comput. Surv. 54 (2021) 115:1–115:35. URL: https://doi.org/10.1145/3457607. doi:10.1145/3457607.

[20] C. G. Northcutt, A. Athalye, J. Mueller, Pervasive label errors in test sets destabilize machine learning benchmarks, 2021. Preprint at https://arxiv.org/pdf/2103.14749.pdf.

[21] J. Dastin, Amazon scraps secret ai recruiting tool that showed bias against women, ????

[22] H. Jiang, O. Nachum, Identifying and correcting label bias in machine learning, CoRR abs/1901.04966 (2019). URL: http://arxiv.org/abs/1901.04966. arXiv:1901.04966.