软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(7):2482−2498 [doi: 10.13328/j.cnki.jos.006584]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于概率模型检查的树模型公平性验证方法*

王 艳[1,2], 侯 哲[3], 黄滢鸿[1,2], 史建琦[1,2], 张格林[1,2]

[1](华东师范大学 软件工程学院, 上海 200062)

[2](国家可信嵌入式软件工程技术研究中心(华东师范大学), 上海 200062)

[3](School of Information and Communication Technology, Griffith University, Brisbane 4111, Australia)

通信作者: 黄滢鸿, E-mail: yhhuang@sei.ecnu.edu.cn; 史建琦, E-mail: jqshi@sei.ecnu.edu.cn

摘 要: 如今, 越来越多的社会决策借助机器学习模型给出, 包括法律决策、财政决策等等. 对于这些决策, 算法的公平性是极为重要的. 事实上, 在这些环境中引入机器学习的目的之一, 就是为了规避或减少人类在决策过程中存在的偏见. 然而, 数据集常常包含敏感特征, 或可能存在历史性偏差, 会使得机器学习算法产生带有偏见的模型. 由于特征选择对基于树的模型具有重要性, 它们容易受到敏感属性的影响. 提出一种基于概率模型检查的方法, 以形式化验证决策树和树集成模型的公平性. 将公平性问题转换为概率验证问题, 为算法模型构建 PCSP# 模型, 并使用 PAT 模型检查工具求解, 以不同定义的公平性度量衡量模型公平性. 基于该方法开发了 FairVerify 工具, 并在多个基于不同数据集和复合敏感属性的分类器上验证了不同的公平性度量, 展现了较好的性能. 与现有的基于分布的验证器相比, 该方法具有更高的可扩展性和鲁棒性.

关键词: 公平性验证; 决策树集成模型; 概率模型检查; 可信机器学习

中图法分类号: TP311

中文引用格式: 王艳, 侯哲, 黄滢鸿, 史建琦, 张格林. 基于概率模型检查的树模型公平性验证方法. 软件学报, 2022, 33(7): 2482–2498. http://www.jos.org.cn/1000-9825/6584.htm

英文引用格式: Wang Y, Hou Z, Huang YH, Shi JQ, Zhang GL. Fairness Verification Method of Tree-based Model Based on Probabilistic Model Checking. Ruan Jian Xue Bao/Journal of Software, 2022, 33(7): 2482–2498 (in Chinese). http://www.jos.org.cn/1000-9825/6584.htm

Fairness Verification Method of Tree-based Model Based on Probabilistic Model Checking

WANG Yan[1,2], HOU Zhe[3], HUANG Yan-Hong[1,2], SHI Jian-Qi[1,2], ZHANG Ge-Lin[1,2]

[1](Software Engineering Institute, East China Normal University, Shanghai 200062, China)

[2](National Trusted Embedded Software Engineering Technology Research Center (East China Normal University), Shanghai 200062, China)

[3](School of Information and Communication Technology, Griffith University, Brisbane 4111, Australia)

Abstract: More and more social decisions are made using machine learning models, including legal decisions, financial decisions, and so on. For these decisions, the fairness of algorithms is very important. In fact, one of the goals of introducing machine learning into these environments is to avoid or reduce human bias in decision-making. However, datasets often contain sensitive attributes that can cause machine learning algorithms to generate biased models. Since the importance of feature selection for tree-based models, they are susceptible to sensitive attributes. This study proposes a probabilistic model checking solution to formally verify fairness metrics of the decision tree and tree ensemble model for underlying data distribution and given compound sensitive attributes. The fairness problem is transformed into the probabilistic verification problem and different fairness metrics are measured. The tool called FairVerify is developed based on the proposed approach and it is validated on multiple classifiers based on different datasets and compound sensitive attributes,

* 基金项目: 国家重点研发计划(2019YFB2102602)

本文由“智能系统的分析和验证”专题特约编辑明仲教授、张立军教授和秦胜潮教授推荐.

收稿时间: 2021-09-05; 修改时间: 2021-10-14; 采用时间: 2022-01-10; jos 在线出版时间: 2021-01-28

showing sound performance. Compared with the existing distribution-based verifiers, the method has higher scalability and robustness.
Key words: fairness verification; decision tree ensemble model; probabilistic model checking; trustworthy machine learning

机器学习模型如今被广泛应用于现代决策系统, 并被应用于司法、医疗、金融等公共领域. 这些系统与人们的生活和利益息息相关, 引起了人们对机器学习可信性研究的关注. 对于机器学习算法的鲁棒性和可解释性以及性质的验证, 都是其中的研究热点. 另一个重要的关注点就是公平性, 所谓公平性, 就是机器学习算法在决策过程中对个体或群体不存在偏见或歧视[1]. 在许多应用场景中, 算法是否公平, 或者说是否存在偏见是极为重要的. 欧洲联盟委员会于 2019 年 4 月发布的《可信人工智能的伦理指南》、美国国家科学技术委员会于 2019 年 6 月更新的《国家人工智能研究与发展战略规划》以及国家新一代人工智能治理专业委员会于 2020 年发布的《新一代人工智能治理原则——发展负责任的人工智能》也都强调了算法公平的重要性[2,3]. 然而机器学习系统容易受到历史数据的影响, 极有可能在决策过程中引入历史过程中的歧视, 或因为数据集的统计偏差等对少数群体产生偏见.

对于公平性的定义与量化, 在很多研究与工作中都能找到[4,5]. 一般的公平性要求分为 3 方面的指标来描述算法“不存在歧视”, 分别为独立性(independence)、分离性(separation)以及充分性(sufficiency)[6]. 独立性要求预测结果独立于个体所在的群体, 其典型的度量标准有统计/人口均等(statistical/demographic parity). 分离性是独立性的扩展, 考察预测结果在目标变量的条件之上和群体之间的关系, 典型的度量标准有几率均等(equalized odds)和机会均等(equal opportunity). 充分性限制了在给定预测结果的条件之上, 目标变量和敏感变量的独立性, 指标如预测均等(predictive parity). 对于前两种标准, 适合从形式化角度验证与分析算法模型.

在机器学习过程中, 不同的情境可以通过不同的方式来解决公平性问题, 往往可以从 3 个层面出发: 数据处理(data processing)、合成(synthesis)以及验证(verification)[7]. 数据处理方法通过调整数据集来消除可能导致学习不公平的偏见或歧视[8−10]. 合成包括通过在算法训练过程进行干预[11,12]以及后处理[13,14]生成更公平的算法, 往往可以通过把公平性也纳入训练目标, 来产生更公平的学习特征, 或是对已有的模型和预测进行修改和标签调整. 现有的基于这3个层面的工作大多是启发性的, 缺少形式化保证. 针对不同场景和定义的公平性形式化验证方法能够提高解决算法公平性的可靠性和可解释性, 是我们关注的重点.

在本文中, 我们关注树模型的公平性问题, 包括单棵决策树(decision tree)和树集成(tree ensemble)模型, 它们常常被应用于各领域的软件系统来获取决策[15,16]. 由于特征选择对基于树的模型的影响, 它们很容易因为敏感属性引入不公平的决策. 目前的一些公平性验证方法, 一部分工作如 FairSquare[17], VeriFair[18]等, 能够处理决策树模型, 但在性能与扩展方面仍有很大的优化空间. 本文提出了一种适用于决策树和树集成模型的多种公平性度量指标的形式化验证框架, 该方法将公平性转化为概率验证问题, 基于概率模型检查对不同数据集和分类器验证独立性和分离性公平指标. 和大多数只针对布尔敏感属性和单一敏感属性的方法不同, 我们支持数值和复合敏感属性,能检测到更多歧视现象. 此外, 我们实现了相应的验证工具 FairVerify, 并在实际应用的数据集和公平性提高算法之上进行了验证.

本文第 1 节介绍方法的一些背景知识, 包括公平性的定义准则和采用的概率建模语言 PCSP#及验证工具 PAT. 第 2 节描述方法的细节并进行举例说明. 第 3 节实现该方法, 并在不同数据集得到的算法上进行验证与分析. 第 4 节从树模型的可信性研究、机器学习的公平性研究、概率模型检查应用这 3 个方面对本文相关工作加以介绍. 第 5 节对全文进行总结, 并对未来值得关注的研究方向进行讨论.

1 背景知识

1.1 公平定义及指标

在机器学习算法的公平性讨论中, 主要考虑二分类任务. 考虑一个数据分布 $D=(X,A,Y)$, 其中, $X=\{X_1,\ldots,X_m\}\in\mathbb{R}^m$ 表示非敏感属性集合; $A=\{A_1,\ldots,A_m\}$表示预定义的敏感属性集合, 如种族、性别、婚姻状况等等; $Y=\{0,1\}$表示一个二值标签, 其中, $Y=1$ 表示该样本为正类, $Y=0$ 表示负类. $a=\{a_1,\ldots,a_n\}$是对于 A 的一组赋值, 表

示一组复合敏感属性. 例如定义敏感属性 A={*sex*,*race*}, 其中, *sex*∈{*male*,*female*}, *race*∈{*White*,*Asian*,*Black*}, 那么 a={*female*,*White*}就是其中的一个复合敏感属性. 通过模型$\mathcal{M}$:(X,A)↦Y 对测试集进行预测, 得到预测标签 $\hat{Y}$. 对于模型的公平性往往通过不同的敏感群体, 尤其是弱势群体和非弱势群体之间获得正类预测的概率是否差距悬殊来进行判断.

因为使用场景和算法的关注点不同, 公平性一直没有一个通用的定义. 在本文中, 主要关注群体公平性, 即要求弱势群体和非弱势群体受到的待遇相似. 我们考虑两个经常用到的公平性定义: 统计均等(statistical parity)[19]和几率均等(equalized odds)[13].

定义 1(统计均等(statistical parity)). 统计均等要求模型预测结果独立于敏感属性群体, 表示为 $\hat{Y}\perp A$. 假设 A 是一个单一的敏感属性, 模型对于弱势群体和非弱势群体的阳性预测值 PPV (positive prective value)应当相等, 即

$$\Pr(\hat{Y}=1|A=0)=\Pr(\hat{Y}=1|A=1) \tag{1}$$

定义 2(几率均等(equalized odds)). 几率均等要求模型在给定实际结果条件下, 预测结果独立于敏感属性群体, 表示为 $\hat{Y}\perp A|Y$. 假设 A 是一个单一的敏感属性, 模型在给定实际标签为正类和负类的条件下, 对于弱势群体和非弱势群体的阳性预测值应当相等, 即

$$\Pr(\hat{Y}=1|Y=y,A=0)=\Pr(\hat{Y}=1|Y=y,A=1),\ 其中,\ y\in\{0,1\} \tag{2}$$

以上两个定义涵盖了对于群体公平性在实际应用中的不同焦点和概念[20]. 统计均等的目标是确保每个群体被分类为正类的概率相同, 而几率均等则考虑了两个群体之间的潜在差异. 然而, 实际上很难要求一个模型同时满足所有的公平性定义, 尤其是在针对复合敏感属性群体的情况下. 对于严格定义进行适当的放松, 可以改善大多数公平概念的适用性, 使评估更灵活[21]. 通过对上述两种公平定义的近似, 我们在本文中定义了以下几种度量标准, 并用于方法中.

度量标准 1(统计均等差异(statistical parity difference, SP)). 统计均等差异允许非弱势群体和弱势群体对于阳性预测值的差异在一个阈值ε以内, 形式化表述为

$$|\Pr(\hat{Y}=1|A=a)-\Pr(\hat{Y}=1|A=b)|\leqslant\varepsilon,\ 其中,\ a,b\in A,\ \varepsilon\in[0,1] \tag{3}$$

度量标准 2(不同影响(disparate impact, DI)). 不同影响允许弱势群体和非弱势群体对于阳性预测值的比值在一个阈值以内, 形式化表述为

$$\frac{\min\limits_{a\in A}\Pr(\hat{Y}=1|A=a)}{\max\limits_{b\in A}\Pr(\hat{Y}=1|A=b)}\geqslant 1-\varepsilon,\ 其中,\ a,b\in A,\ \varepsilon\in[0,1] \tag{4}$$

当ε=0.2, 则符合不同影响法则中的“80%法则”[22].

度量标准 3(机会均等差异(equal opportunity differences, E. Opp.)). 机会均等是指在实际标签为正类的条件下, 弱势群体和非弱势群体的阳性预测值, 即两者间的真阳率 TPR (true positive rates)的差异在一个阈值ε以内, 形式化表述为

$$|\Pr(\hat{Y}=1|A=a,Y=1)-\Pr(\hat{Y}=1|A=b,Y=1)|\leqslant\varepsilon,\ 其中,\ a,b\in A,\ \varepsilon\in[0,1] \tag{5}$$

度量标准 4(平均机会均等差异(average equalized odds difference, E. Odds)). 平均机会均等差异计算了在弱势群体和非弱势群体之间真阳率 TPR 和假阳率 FPR(false positive rates)差异的平均值, 将其限制在一个阈值ε以内, 形式化表述为

$$\frac{1}{2}\left|\Pr(\hat{Y}=1|A=a,Y=1)-\Pr(\hat{Y}=1|A=b,Y=1)\right|+\frac{1}{2}\left|\Pr(\hat{Y}=1|A=a,Y=0)-\Pr(\hat{Y}=1|A=b,Y=0)\right|\leqslant\varepsilon,\ \varepsilon\in[0,1] \tag{6}$$

对于以上 4 个指标, ε值越接近 0, 则说明该模型拥有越高的群体公平性. 在本文中, 我们将模型转换为 PCSP#概率模型, 并通过概率模型检查来验证以上 4 个公平性度量标准.

1.2 概率模型语言PCSP#

CSP#[23]是经典 CSP (communication sequential processes)[24]的一个扩展, 适用于自动化系统分析, 引入了

共享的数据结构和操作, 以及赋值、if-then-else 条件判断和 while 循环等编程结构.

而 PCSP#[25]则是在 CSP#上扩展了概率选择, 支持层次复杂系统的概率性质检查. 它的底层语义基于概率自动机(probabilistic automata).

一般来说, 一个 PCSP#模型包括 4 部分: 常量定义、变量声明、进程定义以及性质规范. 对于进程 P 的语义主要定义如下:

$$\begin{aligned} &P ::= Stop \mid Skip \mid e \to P \mid a\{program\} \to P \mid P \square Q \mid P \sqcap Q \mid \textbf{if } b \textbf{ then } P \textbf{ else } Q \\ &\mid \textbf{case}\{b_0 : P_0; b_1 : P_1; \ldots; b_k : P_k\} \mid P;Q \mid P \parallel Q \mid P \mid\mid\mid Q \mid P \setminus X \mid Q \mid \textbf{pcase}\{pr_0 : P_0; pr_1 : P_1; \ldots; pr_k : P_k\} \end{aligned} \tag{7}$$

其中, P, P_i, Q 都表示进程, e 表示简单事件, a 是顺序程序的名称, b 是布尔表达式, pr_i 是用来表示概率权重的值. *Stop* 表示不做任何操作, *Skip* 表示进程终止. $e \to P$ 表示进程首先执行 e 事件然后进入 P 进程. $a\{program\} \to P$ 表示同时生成事件 a 和执行程序 *program*, 然后进入进程 P. $P \square Q$, $P \sqcap Q$ 以及 **if** b **then** P **else** Q 都表示选择, $P \square Q$ 是外部选择, $P \sqcap Q$ 表示内部非确定性, **if-then-else** 是条件分支. ***case***$\{b_0:P_0;b_1:P_1;\ldots;b_k:P_k\}$是多重条件选择, 其中, b_i 是布尔变量. 在一个状态下, 有且仅有一个 b_i 为真. $P;Q$ 表示顺序执行, 即先执行进程 P, P 结束后再执行 Q. $P \parallel Q$ 表示两个进程并发执行, 而 $P \mid\mid\mid Q$ 表示两个进程交错执行. $P \setminus X$ 隐藏了任何 X 内事件的发生. Q 是对进程的调用, 并允许递归调用. 最后, 概率选择以 ***pcase***$\{pr_0:P_0;pr_1:P_1;\ldots;pr_k:P_k\}$形式表示, pr_0 可以用表示权重的整型表示, 即在概率 $pr_i/(pr_0+pr_1+\ldots+pr_k)$下, 进程将执行 P_i, 也可以直接使用概率, 此时 $\sum_{i=0}^{k} pr_i = 1$.

对于一个 PCSP#模型, 可以通过定义断言并进行概率模型检查来验证其可达性、死锁或 LTL 性质, 同时得到满足性质的最大和最小概率.

PAT (process analysis toolkit)[26]是一个功能强大的综合验证框架, 集建模、仿真和模型检查工具于一体, 适用于并发、实时或概率计算系统. 它支持多种建模语言, 包括 CSP 模型、概率 CSP 模型、实时系统模型等等. PAT 可以用来验证不同的系统性质, 如死锁、可达性、带有公平性假设的 LTL 性质、精化检查以及概率模型检查等. 本文中, 我们将使用 PAT 工具对经过转化的 PCSP#模型进行概率模型检查, 以获得相应的概率值来验证模型的公平性指标.

2 公平性验证框架

本文的目标是, 针对决策树及树集成模型提出一个基于概率模型检查的公平性验证框架. 方法的思路是: 通过将机器学习模型转化为能够进行概率模型检查的 PCSP#模型, 来获得针对不同复合敏感属性下的阳性预测值的条件概率, 以此验证前文所述的几种公平性指标, 衡量模型的公平性. 我们方法的核心在于: 计算在不同敏感属性群体之下, 模型的阳性预测概率 $\Pr(\hat{Y}=1 \mid A=a)$. 如图 1 所示, 方法的输入是待验证模型以及其数据分布. 首先, 如果模型为决策树算法, 不需要做更改, 如果模型是树集成模型, 需要先生成简化的解释模型; 其次, 对模型构建等价的 PCSP#模型, 为了获得概率 $\Pr(\hat{Y}=1 \mid A=a)$, 针对每组敏感属性, 对节点进行操作, 并设定合适的可达性性质, 并对所获得的模型进行概率模型检查得到概率值; 最后, 通过概率值计算不同公平性指标, 获得模型公平性程度. 为了更好地解释本文的方法, 我们在这里先假设保护属性和非保护属性之间是相互独立的.

2.1 决策树模型公平性建模

一个分类决策树模型由内部节点和叶子节点构成, 每个内部节点 n_i 都关联一个特征阈值表达式 $e_i=(x_i,\theta_i)$, 并拥有两个后继节点, 当 $e_i=x_i \leqslant \theta_i$ 为真, 当前节点中满足表达式的样本将会转移到其左孩子节点, 不满足则转移到右孩子节点. 当节点特征为敏感属性, 该节点被视作为敏感节点. 每个叶子节点包含一组投票分布 $(v_0,\ldots,v_{m-1})$, 其中, m 为类别数目, $v_i(0 \leqslant i \leqslant m-1)$为类别的样本数, 即获得的票数. 在二分类任务中 $m=2$, 决策树的输出即预测结果是票数更多的那一类.

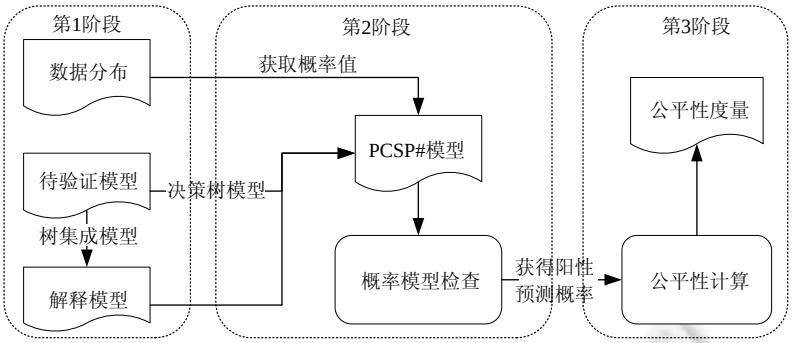


图 1 方法概述

为了获得概率 $\Pr(\hat{Y}=1|A=a)$, 我们根据算法 1 对决策树构建 PCSP#模型并求解, 算法的输入是决策树模型、数据分布以及预定义的复合敏感属性. 我们使用测试集来近似数据集的数据分布, 对于第 15 行中表达式概率值的计算, 使用测试集中满足节点逻辑表达式的分布来获取. 对于每一棵决策树, 在 PCSP#模型中对应一个主进程, 即需要进行验证的主体. 由于需要计算阳性预测概率, 我们定义了一个全局变量 *label* 来保存决策树的最终预测结果. 对于决策树中的每个内部节点, 构建相应进程, 其中, 主进程对表示决策树根节点的进程进行引用. 如算法第 5 行–第 17 行所示: 针对内部节点, 若节点为敏感节点, 则其后继根据预定义敏感属性赋值是确定的, 直接引用该分支所表示的进程即可; 若节点为一般节点, 则进程根据其左右孩子满足概率和表示相应后继节点的进程, 使用 *pcase* 算子构建. 对于每个叶子节点, 如第 19 行–第 24 行所描述, 根据叶子节点值判断节点输出, 即若正类数量更多则输出 1, 触发将节点输出赋给 *label* 变量的事件, 并结束进程. 对模型定义目标为 *label*==1 的可达性性质断言, 并对其进行概率模型检查, 就能获得概率 $\Pr(\hat{Y}=1|A=a)$.

算法 1. 决策树模型建模算法 *Verify_DT*.

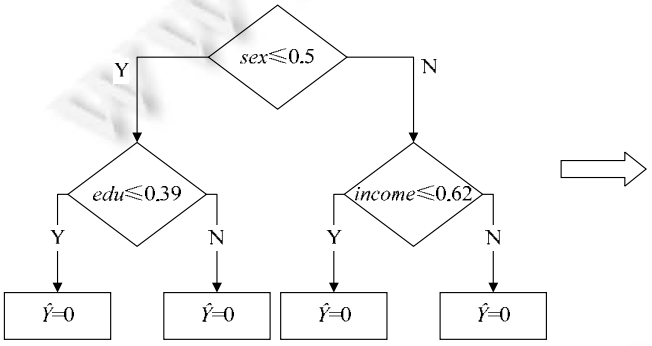
输入: 决策树模型$\mathcal{M}$, 数据分布 *D*, 敏感属性 *A*=*a*.

输出: 条件概率 $\Pr(\hat{Y}=1|A=a)$.

1: 定义全局变量并初始化 *label*=−1;
2: 定义主进程变量 *T*←根节点对应进程变量;
3: **for all** $n_i \in \mathcal{M}$ **do**
4: 定义进程变量 N_i;
5: **if** n_i is not leaf **then**
6: N_l←左孩子对应进程变量;
7: N_r←右孩子对应进程变量;
8: **if** n_i is *sensitive_node* **then**
9: **if** e_i *satisfy a* **then**
10: $N_i \leftarrow N_l$;
11: **else**
12: $N_i \leftarrow N_r$;
13: **end if**
14: **else**
15: $p_i \leftarrow CalculateProb(e_i)$;
16: $N_i \leftarrow pcase\{p_i, N_l; 1-p_i: N_r\}$;
17: **end if**
18: **else**

19: **if** $v_0 \leqslant v_1$ **then**
20: N_i←执行赋值操作 *label*=1 后停止进程;
21: **else**
22: N_i←执行赋值操作 *label*=0 后停止进程;
23: **end if**
24: **end if**
25: **end for**
26: *assetion*←*T* reaches *label*==1 with *prob*;
27: *Prob*←*Verification*();
28: **return** *Prob*;

我们将通过图 2(a)中所示决策树模型，说明如何使用我们的方法计算在 *sex*=1 的敏感属性条件下 $\Pr(\hat{Y}=1)$ 的值. 该决策树预测任务为某个个体能否申请到贷款，其中，$\hat{Y}=1$ 表示能够申请到贷款，该决策树中选择的特征为 *sex*, *edu* 以及 *income*. 我们将 3 个节点“*sex*≤0.5”“*edu*≤0.39”以及“*income*≤0.62”分别用“*N*1”, “*N*2”和“*N*3”来表示，其中，“*N*1”即节点内表达式特征为“*sex*”的节点，为敏感属性节点. 从数据分布中获得非敏感属性，各特征阈值表达式的概率分别为 Pr(*edu*≤0.39)=0.633 和 Pr(*income*≤0.62)=0.458. 根据上述建模方式，我们得到如图 2(b)所示的 PCSP#模型，其中，表示敏感属性节点的进程分支由相应的敏感属性赋值确定. 通过 PAT 计算，得到概率值为 0.633. 同样构建“*sex*=0”条件下等价模型进行验证，得到概率值为 0.458. 即在此模型中，对于敏感属性 *sex* 的两个群体，得到阳性预测结果的概率分别为 0.367 和 0.542. 然而实际上，对于敏感属性和非敏感属性之间往往具有相关性，因此，我们可以使用敏感属性条件下的概率分布 $p_i=\Pr(e_i|A=a)$来替代 $\Pr(e_i)$，以检查潜在的不公平性.



(a)

```
var label = -1;

Tree = N1;
//a:sex=1
N1 = N2;
N2 = pcase{
       [0.633]:L1
       default:L2
      };
N3 = pcase{
       [0.458]:L3
       default:L4
      };
L1 = e{label = 0}->Skip;
L2 = e{label = 1}->Skip;
L3 = e{label = 0}->Skip;
L4 = e{label = 1}->Skip;

#define goal label == 1;
#assert Tree reaches goal with prob;
```

(b)

图 2 决策树建模示例

2.2 树集成模型公平性建模

引用文献[27]中的定义，我们将树集成模型定义为 *T* 个决策树的加权和，即

$$\mathcal{M}(x)=\sum_{i=1}^{T}\omega_i\cdot t_i(x) \tag{8}$$

其中，$\mathcal{M}$表示树集成模型，t_i 和ω_i 表示决策树和它的权重. 综合所有决策树的票数，可得到树集成模型中对于每一类的投票结果.

由于 PCSP#对高级编程结构的支持，实际上也很容易构建对于树集成模型的 PCSP#模型. 然而，由于树集成模型需要通过每棵决策树的结果来得出最终的结果，计算量将会达到叶子节点数量的指数级别，对于大规模模型，很难有模型检查工具能很好地执行. 因此，考虑到方法的适用性，我们首先基于先前的工作[28]，将树集成模型转化为一个相对简单的替代模型，称为“解释模型”；然后再对解释模型构建等价的 PCSP#模型，来获得计算公平性所需概率.

对于树集成模型$\mathcal{M}$，它的解释模型$\mathcal{M}_e$的生成有 4 个决定性的参数: θ, ϕ, ψ和 k，即

$$\mathcal{M}_e=explain(\mathcal{M},\theta,\phi,\psi,k) \tag{9}$$

其中，θ决定决策规则的复杂程度，ψ决定每条决策规则的签名 s，ϕ和 k 两个参数决定决策规则的数量. 解释模型是一个包含若干加权决策规则的集合，图 3 包含了一个解释模型的示例.

决策规则	签名 (类别0、类别)	权重	对于一个样本x, 符合决策规则
$r_1: x_1 \leqslant \tau_1$	(0,2)	24	r_1,r_2,r_5
$r_2: (x_2 \leqslant \tau_2) \wedge (x_3 \geqslant \tau_3)$	(0,2)	20	(0,2)×24 + (0,2)×20 + (2,0)×35
$r_3: x_4 \geqslant \tau_4$	(2,0)	11	
$r_4: (x_5 \geqslant \tau_5) \wedge (x_6 \geqslant \tau_6)$	(0,2)	26	
$r_5: x_7 \leqslant \tau_7$	(2,0)	35	(70,88)
$r_6: x_8 \geqslant \tau_2$	(2,0)	18	预测结果: 类别1
...			

图 3 解释模型预测示例

使用解释模型的优势在于: 与原模型相比，解释模型的逻辑结构更加简单，能够大幅降低计算的复杂度；同时，解释模型具有与原模型高度相似的预测能力. 由于本文方法对于公平性的验证是基于统计的验证，一定程度的误差是允许的. 为了量化模型转换所带来的误差，我们引入保真度(fidelity)来近似评估这一误差. 保真度即解释模型的预测能力和原模型的相似程度，则解释集保真度越高，与原模型的误差越小. 考虑一个不带标签的测试集 $X=\{x_1,x_2,\ldots,x_i,\ldots,x_n\}$，我们将原模型对其进行预测的结果和解释模型的预测结果进行比较，定义保真度计算公式为

$$S_{opt}=\sum_{i}^{n}\Theta(\mathcal{M}(x_i),\mathcal{M}_e(x_i)) \tag{10}$$

其中，当$\mathcal{M}(x_i)=\mathcal{M}_e(x_i)$, $\Theta(\mathcal{M}(x_i),\mathcal{M}_e(x_i))=1$；否则，$\Theta(\mathcal{M}(x_i),\mathcal{M}_e(x_i))=0$.

我们理想的解释集是规则数量控制在不超过执行能力的条件下，使模型保真度尽可能高，并以此为目标，采用优化算法对解释集的生成进行优化. 粒子群优化算法(PSO)[29]是一种经典的仿生优化算法，被广泛引用于解决优化问题[30]. 并基于基本的 PSO 算法，衍生出各类提升算法[31]. 我们使用的线性递减惯性权重粒子群优化算法(LDIW-PSO)[32]与基本的 PSO 算法相比，平衡了全局搜索能力和局部搜索能力. 较大的惯性权重能在优化前期拥有比较好的全局搜索能力，较小的惯性权重能在优化后期拥有较好的局部搜索能力. 通过实验对比，对于解释集的优化工作，无需复杂的惯性权重策略，使用 LDIW-PSO 算法完全满足本文的优化需求. 我们以等式 S_{opt} 作为优化过程的拟合函数，来找出针对当前模型最优的一组参数；然后，使用最优参数对原模型进行简化得到解释模型.

使用该方法获得的解释模型$\mathcal{M}_e$包含一系列规则，每条规则 r_i 对应一个二元组性质(s_i,w_i). 其中，s 为签名，即类的标准化投票分布；w 为该决策规则权重，代表该规则的重要性. 规则由一系列特征逻辑表达式的并集组成，每个表达式 $e_i=(x_i,\tau_i^l,\tau_i^u)$ 通过上下界限定特征. 对于给定样本 x，凡是 x 所满足的决策规则，将这些规则的签名和权重相乘，累计相加，最终得到一个投票分布，分布中最大值所代表的类别就是解释模型得出的预测结果. 图 3 给出了一个解释模型$\mathcal{M}_e$的预测示例.

结合解释模型的预测方式, 我们使用算法 2 对树集成模型计算 $\Pr(\hat{Y}=1|A=a)$.

算法 2. 树集成模型建模算法 *Verify_TE*.

输入: 树集成模型$\mathcal{M}$, 数据分布 D, 敏感属性 A=a.

输出: 条件概率 $\Pr(\hat{Y}=1|A=a)$.

1: $\theta,\phi,\psi,k\leftarrow PSO(\mathcal{M})$;
2: $\mathcal{M}_e\leftarrow explain(\mathcal{M},\theta,\phi,\psi,k)$;
3: 定义全局变量并初始化 *label*=–1, c_0, c_1=0;
4: **for all** $r_i\in\mathcal{M}_e$ **do**
5: 定义规则进程变量 R_i;
6: **for all** $x_i\in r_i$ **do**
7: 定义表达式进程变量 N_i;
8: $P_{next}\leftarrow$后继表达式对应进程;
9: **if** e_i is not the last one **then**
10: $P\leftarrow P_{next}$;
11: **else**
12: $P\leftarrow$ 执行事件$c_0=c_0+s_i^0\times w_i$, $c_1=c_1+s_i^1\times w_i$ 后停止进程;
13: **end if**
14: **if** $x_i\in A$ **then**
15: **if** e_i satisfy a **then**
16: $N_i\leftarrow P$;
17: **else**
18: $N_i\leftarrow Skip$;
19: **end if**
20: **else**
21: $p_i\leftarrow CalculateProb(e_i)$;
22: $N_i\leftarrow pcase\{p_i,P;1-p_i:Skip\}$;
23: **end if**
24: **end for**
25: **end for**
26: 定义主进程变量 M;
27: $P_j\leftarrow$比较 c_0 和 c_1, 为 *label* 赋值相应的分类结果;
28: $M\leftarrow$所有 R_i 顺序执行后执行 P_j;
29: *assetion*$\leftarrow M$ reaches *label*==1 with *prob*;
30: *Prob*$\leftarrow$*Verification*();
31: **return** *Prob*;

在获得解释模型之后, 针对解释模型构建等价的 PCSP#模型. 除了定义结果标签变量外, 还需要分别定义两个全局变量来保存两个类别的累计权重. 以决策规则为单位分别构建进程, 并对每条规则中的表达式分别构建子进程并顺序调用, 每个代表规则的进程结束之后进行对标签变量的赋值操作. 其中, 和决策树模型建模一样, 如果表达式特征为敏感属性, 则相应的进程将直接调用确定的进程. 即如果表达式符合预定义的敏感属性群体, 则进入后继子进程; 否则, 进程结束. 最后, 表示模型的主进程顺序引用每个规则进程, 并在最后根据两个全局变量的比较对 *label* 变量赋值. 断言定义与决策树建模相同. 同样地, 每个进程分支的概率采用测试集中满足特征逻辑公式的概率分布来近似. 假设图 3 示例中, x_2 和 x_7 为敏感属性, 计算 $x_2\leqslant\tau_2$, $x_7\leqslant\tau_7$

敏感属性组合条件下阳性预测率,构建 PCSP#模型如图 4 所示.

```
var c0 = 0;
var c1 = 0;
var label = -1;

r1_n1 = pcase {[0.034] : e{c1 = c1 + 48} -> Skip default : Skip};
r2_n1 = r2_n2
r2_n2 = pcase {[0.001] : e{c1 = c1 + 40} -> Skip default : Skip};
r3_n1 = pcase {[0.168] : e{c0 = c0 + 22} -> Skip default : Skip};
r4_n1 = pcase {[0.586] : r4_n2 default : Skip};
r4_n2 = pcase {[0.004] : e{c1 = c1 + 52} -> Skip default : Skip};
r5_n1 = e{c0 = c0 + 70} -> Skip
r6_n1 = pcase {[0.001] : e{c0 = c0 + 36} -> Skip default : Skip};

M = r1_n1;r2_n1;r3_n1;r4_n1;r5_n1;r6_n1;if(c0 >= c1){
      {label = 0;} -> Skip
      }else{
      {label = 1;} -> Skip
      };

#define target label == 1;
#assert M reaches target with prob;
```

图 4　解释模型建模示例

2.3　公平性度量计算

在本文中，我们采用第 1.1 节中定义的 4 种度量标准衡量算法的群体公平性. 基于第 2.1 节和第 2.2 节描述的方式得到特定复合敏感属性下条件概率 $\Pr(\hat{Y}=1|A=a)$, 通过遍历每个复合敏感属性，我们可以得到最大概率和最小概率，即相对应最受保护群体和最不受保护群体. 通过两个概率值，可以计算统计均等差异(*SP*)和不同影响(*DI*)两个度量标准，如算法 3 所示. 对于机会均等度量(*E.Opp* 和 *E.Odds*)，我们基于相同的方式，只需要在建模过程中对概率分布 p_i 进行调整. 计算 *TPR*，使用条件概率 $p_i=\Pr(e_i|Y=1)$，计算 *FPR*，使用条件概率 $p_i=\Pr(e_i|Y=0)$.

算法 3. 公平性验证遍历算法 *FairVerify*.

输入：模型$\mathcal{M}$，数据分布 $D=(X,A,Y)$;

输出：最不受保护群体 G_u 和最受保护群体 G_p、度量标准 *SP* 和 *DI*.

1: $G\leftarrow\varnothing$;
2: **for all** $a\in A$ **do**
3: $Pr\leftarrow Verify(\mathcal{M},D,a)$;
4: 将(Pr,a)加入到集合 G 中;
5: **end for**
6: $(max_{Pr},max_a)=MAX(G)$;
7: $(min_{Pr},min_a)=MIN(G)$;
8: **return** max_a, min_a, $max_{Pr}-min_{Pr}$, $1-min_{Pr}/max_{Pr}$;

对于某些数据集，可能存在多个敏感属性；同时，每个敏感属性又会有多个值. 这种情况下，对每一种复合敏感属性遍历将会造成不小的开销. 鉴于 PCSP#可以构建不确定性分支，且 PAT 验证工具能够同时计算验证性质所能达到的最大和最小概率，我们提出一种无需遍历的快速方式. 以决策树模型为例，如算法 4 所示，每个模型只需进行一次建模、一次求解，就能得到最大概率和最小概率. 其中，建模过程与遍历方式的区别在于敏感特征所在的进程的后继定义为非确定，如第 8 行所示. 得到最大概率和最小概率后，使用同样的方式计算度量标准.

对于显式的歧视，两个方法的计算结果是相同的. 而具体使用哪种方法，可以根据不同的需求做选择. 遍历方式的优势是：可以通过表达式对于敏感属性的条件概率来获取潜在的不公平性，同时还能得出最受保护

群体和最不受保护群体的具体值, 适合应用在复合敏感属性群体不多、但需要获得更具体的公平性的情况. 而快速算法最大的优势就是只需进行一次建模、一次求解, 适合应用在复合敏感属性较多的情况下.

算法 4. 公平性验证快速算法 *FairVerify_efficient*(*DT*).

输入: 模型$\mathcal{M}$, 数据分布 $D=(X,A,Y)$.

输出: 度量标准 *SP* 和 *DI*.

1: 定义全局变量并初始化 *label*=−1;
2: 定义主进程变量 T←根节点对应进程变量;
3: **for all** $n_i \in \mathcal{M}$ **do**
4: 定义进程变量 N_i;
5: **if** n_i is not leaf **then**
6: N_l←左孩子对应进程变量;
7: N_r←右孩子对应进程变量;
8: **if** n_i is *sensitive_node* **then**
9: $N_i \leftarrow N_l \sqcap N_r$;
10: **else**
11: $p_i \leftarrow CalculateProb(e_i)$;
12: $N_i \leftarrow pcase\{p_i,N_l;1-p_i:N_r\}$;
13: **end if**
14: **else**
15: **if** $v_0 \leqslant v_1$ **then**
16: N_i←执行赋值操作 *label*=1 后停止进程;
17: **else**
18: N_i←执行赋值操作 label=0 后停止进程;
19: **end if**
20: **end if**
21: **end for**
22: *assetion*1←T reaches label==1 with $p_{\max}$;
23: *assetion*2←T reaches *label*==1 with $p_{\min}$;
24: max_{Pr}, $min_{Pr} \leftarrow Verification()$;
25: **return** $max_{Pr}-min_{Pr}$, $1-min_{Pr}/max_{Pr}$;

3 实验与分析

本节检验第 2 节中所提出的公平性验证方法在真实数据集上的效果. 实验主要针对 3 个目标: (1) 表明方法能够验证决策树和树集成模型的公平性以及评估公平性提高算法的效率; (2) 表明方法能够分析针对多个复合敏感属性群体的公平性; (3) 分析方法在不同数据集上的性能.

3.1 实验建立

我们在 Python (3.8.5 版本)中实现了方法并开发了相应名为 FairVerify 的工具. FairVerify 中的关键验证步骤使用现有的模型检查工具 PAT 实现. 在实验中, 我们使用 scikit-learn[33]训练了不同规模的决策树和随机森林模型. 本文使用了在公平性环境中经常使用的 Adult Income, Bank Marketing 和 COMPAS 这 3 个数据集来进行实验, 前两个数据集可在 UCI 仓库下载. 其中, Adult Income 的任务是预测一个人年薪是否超过 50 k, 正类标签是年薪超过 50 k; Bank Marketing 分类目标是预测客户是否认购定期存款, 正类标签为是; COMPAS 包含

来自美国佛罗里达州布劳沃德县的被告记录, 分类目标是预测一个人是否再犯, 正类标签为是. 关于数据集的样本数量和特征数量信息见表 1. 对于每个数据集, 我们针对敏感属性进行了预处理, 对于给定的敏感属性组合, 对类别特征进行 one-hot 编码, 对于连续特征先进行离散化成不同的组别再进行 one-hot 编码. 实验在一台 CPU 为 Intel Core i9-7960X、RAM 为 32 GB 的主机上构建.

表 1 数据集信息

数据集	Adult	Bank	COMPAS
样本数量	32 561	41 188	7 214
初始特征数量	12	20	8
处理后特征数量	94	66	18
敏感属性	race (5), age (4), sex (2)	age (4), marital (4)	race (5), age (4), sex (2)

3.2 算法公平性验证

我们分别在 3 个数据集上针对不同的敏感属性实验了我们的方法. 同时, 我们挑选了两个提升公平性的算法, 分别为使类别阳性比率均等的数据集校准(equal ratio re-calibration)和反事实增强(counterfactual augmention), 两种算法都是基于数据预处理的公平性提升算法. 我们将原始模型和经过公平性提升算法处理过的模型在我们的方法之上应用的结果进行对比, 一方面体现我们的方法对于验证公平性的效果, 另一方面证明我们的方法还可以对公平性算法的效率进行评估. 实验结果见表 2, 其中加粗的数值为算法对比未处理的原始算法公平性存在提高, FD 数值表示随机森林模型生成的解释模型的保真度, 以衡量误差. 其中, 我们使用测试集数据来计算随机森林模型解释集的保真度.

表 2 对于不同数据集的公平性验证和公平性提升算法的性能评估

数据集	敏感属性	提升算法	度量指标								
			决策树				随机森林				
			SP	DI	E.Opp	E.Odds	SP	DI	E.Opp	E.Odds	FD
Adult	race	original	0.146	0.411	0.225	0.173	0.216	0.829	0.230	0.161	0.879
		re-calibration	0.178	**0.400**	**0.103**	0.194	0.625	0.960	0.296	0.473	0.846
		augmention	**0.012**	**0.046**	**0.001**	**0.004**	0.178	**0.365**	**0.216**	**0.119**	0.891
	sex	original	0.194	0.519	0.039	0.098	0.143	0.687	0.142	0.080	0.879
		re-calibration	**0.062**	**0.144**	0.170	0.185	**0.120**	**0.648**	**0.065**	0.107	0.846
		augmention	**0.004**	**0.014**	**0.002**	**0.003**	**0.018**	**0.295**	0.225	0.123	0.891
Bank	age	original	0.317	0.638	0.068	0.180	0.003	0.784	0.080	0.040	0.956
		re-calibration	**0.130**	**0.252**	**0.012**	**0.150**	0.315	0.902	0.278	0.310	0.793
		augmention	**0.012**	**0.044**	**0.003**	**0.006**	**0.001**	**0.124**	0.157	0.079	0.922
	marital	original	0.082	0.299	0.074	0.058	0.500	0.998	0.070	0.035	0.926
		re-calibration	0.312	0.623	0.249	0.286	**0.082**	**0.408**	0.199	0.164	0.832
		augmention	**0.003**	**0.012**	**0.003**	**0.003**	**0.004**	**0.145**	0.082	0.043	0.947
COMPAS	race	original	0.192	0.366	0.382	0.402	0.379	0.960	0.795	0.570	0.848
		re-calibration	**0.049**	**0.082**	**0.059**	**0.053**	**0.119**	**0.225**	**0.074**	**0.128**	0.804
		augmention	**0.062**	**0.158**	**0.074**	**0.057**	**0.167**	**0.167**	**0.163**	**0.136**	0.880
	sex	original	0.091	0.227	0.105	0.083	0.149	0.905	0.308	0.176	0.848
		re-calibration	**0.004**	**0.009**	**0.077**	**0.079**	0.335	**0.594**	**0.153**	0.306	0.804
		augmention	**0.006**	**0.015**	**0.002**	**0.004**	0.167	**0.490**	**0.215**	**0.138**	0.880

观察表中数据, 我们可以看到, 无论是决策树模型还是随机森林模型, 反事实增强算法比数据均衡算法对消除歧视和偏见更加稳定和有效, 在决策树模型中更为明显. 比如, 在 Bank 数据集中, 若预定义敏感属性为 marital, 使用调整数据集类别阳性比率的方法处理数据集, 可能非但无法提高公平性, 还会增加数据带来的偏见. 另外, 对于决策树模型, 可以看到, 对于同一个数据集, 这两种公平性提升算法对于预定义敏感属性群体只有两个(sex)的模型比敏感属性群体不止两个(race,age)的模型效果要好得多. 在随机森林模型上, 很难检测到这两种公平性算法对于 E.Opp 和 E.Odds 两个度量的公平性提升, 可以侧面反映这两种算法可能在消除群体之间潜在偏见方面存在不足.

3.3 复合敏感属性分析

大多数公平性验证器或是公平性评估算法只考虑单一敏感属性，并且只考虑该属性的正负两个赋值. 然而，对于一个存在歧视或偏见的数据集，往往存在不止一个敏感属性，其中的一些属性可能有多个赋值. 我们的方法考虑多个复合敏感属性的多个赋值的情况，使评估的角度更全面. 当数据集可能存在多个敏感属性，FairVerify 可以验证不同组合敏感属性群体下的公平性. 对于验证是针对某个敏感属性的两个特定的群体还是整体，是可选的.

对于预定义一组敏感属性，我们可以通过可视化不同复合敏感属性组别的 PPV 即不同敏感属性组合 a 的条件下预测值为正的概率值来进一步分析公平性. 通过不同群体 PPV 之间的差异，我们能够直观地看到模型在不同群体上的偏见程度. 图 5 为 COMPAS 数据集上的一个随机决策树模型对于{age,sex}敏感属性的不同群体的 PPV 结果. 可以看到, “*age*_1, *sex*_F”显然在该分类任务中是一个弱势群体.

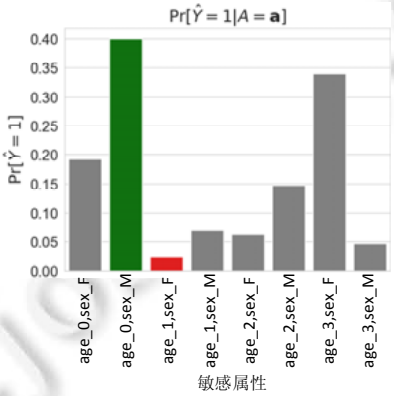


图 5 COMPAS 数据集中对复合敏感属性群体的 PPV 值差异

另外，我们观察到，无论是决策树模型还是随机森林模型，在大多数数据集上，被考虑的复合敏感属性越多，各个度量标准的值就越大，即存在越大的歧视与偏见. 我们在 Adult 数据集上针对决策树和随机森林模型分别测试了对各个不同的敏感属性组合下 4 个度量标准的值，其中，以同一个敏感属性 race 为标准，分别对比了只有 race 属性(5 个赋值)、race 与 sex 属性(10 个赋值)、race 与 age 属性(20 个赋值)以及 race, sex, age 属性(40 个赋值)，随机测试了 5 组模型，得到结果如图 6 所示. 可以看到，复合敏感属性的赋值越多，有可能检查到更多的歧视.

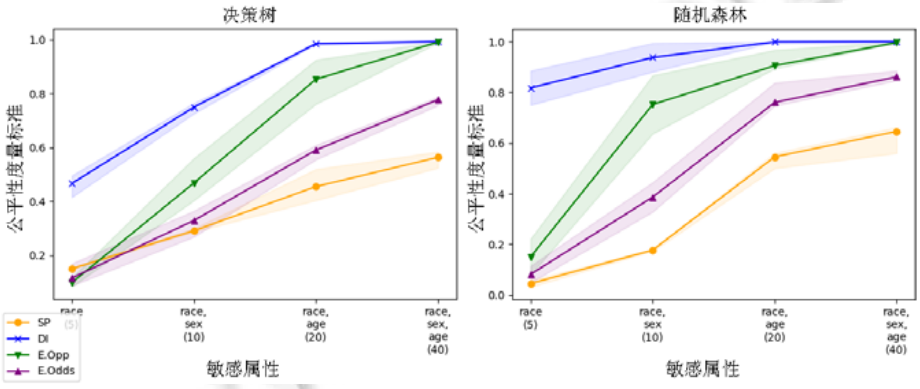


图 6 Adult 数据集中不同复合敏感属性组合的度量标准结果

3.4 验证结果与性能的比较分析

对于决策树模型的公平性验证，已经有很多工作提出了解决方案. Justicia[34]是与我们最相近的工作，通

过 SSAT 求解的方法，与现有的 FairSquare, VeriFair 等方法相比，它显著提高了在决策树和线性回归算法的公平性验证性能. 对于集成树模型与决策树模型公平性验证的时间开销差异主要在于对解释模型生成的过程，而该过程的性能受迭代参数的影响，和同类方法相比是可观的[28]. 我们基于前文所述 3 个数据集，将 FairVerify 在决策树模型上的验证结果与 Justicia 作对比. 表 3 展示了不同敏感属性设置下, FairVerify 和 Justicia 的遍历算法及快速算法的部分实验结果，差值大于 0.1 的数值加粗显示. 可以观察到，FairVerify 和 Justicia 的结果数值非常接近，产生微小差异的原因是，算法对于处理决策树模型的策略不同. 其中，根据公式(4)的定义，这样的差异会在 DI 中被放大.

表 3　对于决策树模型 FairVerify 与 Justicia 的公平性验证结果比较

数据集	敏感属性	验证算法	FairVerify				Justicia			
			SP	DI	E.Opp	E.Odds	SP	DI	E.Opp	E.Odds
Adult	race	遍历	0.124	0.383	0.131	0.114	0.138	0.344	0.099	0.103
		快速	0.135	0.357	0.185	0.142	0.066	0.295	0.096	0.078
	sex	遍历	0.196	0.576	0.031	0.095	0.194	0.547	0.034	0.094
		快速	0.027	0.088	0.022	0.022	0.006	0.019	0.006	0.006
	race, sex	遍历	0.537	0.745	0.503	0.360	0.59	0.81	0.468	0.377
		快速	**0.181**	**0.442**	**0.214**	0.176	**0.078**	**0.225**	**0.102**	0.087
Bank	age	遍历	0.344	0.647	0.118	0.219	0.359	0.596	0.094	0.229
		快速	0.109	0.354	0.138	0.116	0.082	0.276	0.063	0.070
	marital	遍历	0.471	0.578	0.069	0.064	0.531	0.678	0.069	0.061
		快速	0.056	**0.208**	0.118	0.080	0.014	**0.057**	0.041	0.027
	age, marital	遍历	0.39	0.712	0.545	0.461	0.46	0.709	0.470	0.490
		快速	0.156	**0.461**	**0.236**	0.183	0.105	**0.334**	**0.103**	0.100
COMPAS	race	遍历	0.286	**0.615**	0.401	0.369	0.192	**0.248**	0.307	0.366
		快速	0.494	**0.799**	**0.511**	**0.492**	0.203	**0.603**	**0.181**	**0.199**
	sex	遍历	0.023	0.058	0.004	0.033	0.001	0.001	0.013	0.023
		快速	0.164	**0.343**	0.162	0.165	0.082	**0.189**	0.065	0.079
	race, sex	遍历	0.403	0.438	**0.431**	**0.469**	0.391	0.392	**0.292**	**0.351**
		快速	0.415	**0.888**	0.679	0.653	0.32	**0.519**	0.305	0.315

此外，我们对比了 FairVerify 和 Justicia 的计算性能. 我们分别对 3 个数据集预定义全部的敏感属性，即 Adult 数据集具有 40 个敏感属性群体, Bank 数据集具有 16 个敏感属性群体，而 COMPAS 数据集具有 48 个敏感属性群体. 在此基础上，我们通过限定决策树的最大深度来生成不同规模大小的模型. 不失一般性，特征数多的数据集往往在最大深度限制大的情况下模型也会相对要大. 从图 7 所示结果可以观察到，对于 COMPAS 这类训练出的模型较小的模型, Justicia 的性能较好于 FairVerify; 但对于大规模的模型, FairVerify 的处理速度将会快很多. 可以说，FairVerify 具有更高的扩展性. 此外，对比实验是在 Linux 平台完成的，而 PAT 工具在 Windows 平台的性能会更加出色.

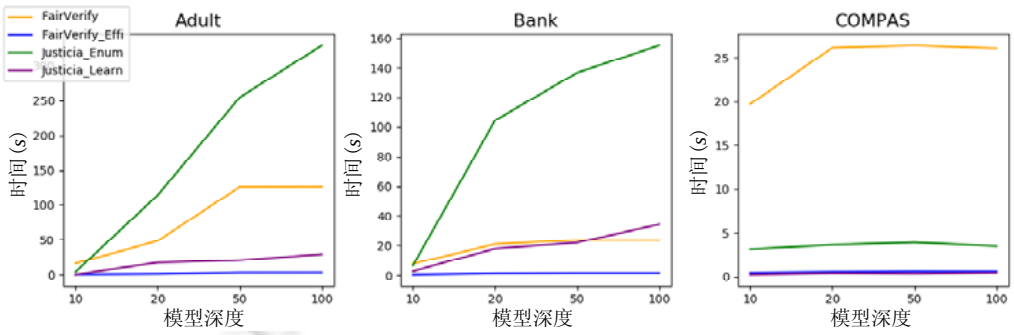


图 7　对于决策树模型 FairVerify 和 Justicia 性能比较

4 相关工作

对于树模型的可信性研究, 包括对于树模型可解释性、可靠性、公平性以及安全性质的研究, 与线性模型和神经网络模型相比, 近几年才刚刚兴起. Bride 等人提出了 Silas 机器学习框架[35], 将决策树学习和逻辑推理思想相联系, 使基于 Silas 框架训练得到的模型具有良好的可解释性和可靠性. 麻省理工学院的 Chen 等人[36]将鲁棒性验证问题转换为多个图的上的最大团问题, 给出了一个关于鲁棒性验证的精确描述. 此外, 他们进一步提出了一种有效的多级验证算法, 允许在迭代改进和任意时间终止的情况下, 为决策树集合的鲁棒性提供精确的下界. Sato 等人[37]实现了一种基于树模型验证结果的异常值过滤器, 他们通过提取会导致模型发生异常的特征值的范围之后, 基于该范围创建输入过滤器. Devos 等人[38]把树集成模型的验证任务转化为优化求解问题, 包括对抗样本生成、鲁棒性检查、公平性等等, 并提出了名为 VERITAS 的验证框架. John 等人[39−41]针对树模型范围的合理性和输入扰动的鲁棒性两种系统性质, 提出了一系列形式化验证技术, 包括通过抽象精化的方法, 来解决高维度验证案例中的组合爆炸问题以及提取等价类验证输入输出映射复合需求等工作, 并设计了名为 VoTE 的工具来实现.

公平性机器学习是公平性研究的一个重要方面. Kilbertus 等人[42]利用贝叶斯网络对敏感属性和非敏感属性之间的关联性进行解析, 提出了解析属性和代理属性的概念. 即使贝叶斯网络中的属性与敏感属性之间不存在直接可达的路径, 而该属性与代理属性之间存在路径相连, 则该属性存在代理歧视. 2019 年, Thomas 等人[43]总结了多个公平性定义, 说明常规机器学习算法所得模型都存在一定的不公平性, 并阐述了标准化学习的局限性. 石鑫盛等人[10]提出了一种基于分类间隔的加权方法, 用于处理二分类任务中的歧视现象, 并在 demographic parity 和 equalized odds 公平性判定准则上实现分类公平.

近些年来, 公平性的验证也逐渐成为一个重要的关注点. 一些工作关注算法个体公平性(individual fairness)的验证, 个体公平性考虑的是最坏情况, 往往还需要提供鲁棒性的保证. John 等人[44]提出了一种具有全局鲁棒性的个体公平性验证方法, 但对于最坏情况, 可能出现指数级的开销. Ranzato 等人[45]针对决策树算法提出了个体公平性验证和训练的方法, 他们将个体公平性等价于稳定性. 不同于个体公平性, 群体公平性(group fairness)考虑整体的平均情况, 现有的公平性定义与指标大多也针对群体公平性的概念. FairSquare 将公平性定义为约束求解问题, 使用数值积分方法来验证. VeriFair 通过对敏感变量进行抽样, 实现多种公平性度量的概率验证. Justicia 将公平性问题转化为随机布尔可满足性问题进行验证, 并扩展到了复合敏感属性的前提. 然而对于大规模的输入, 这些方法仍然具有局限性. Ignatiev 等人[7]提出了一个形式化的验证框架来评估数据集和算法的无意识公平(fairness through unawareness), 但没有考虑非敏感属性对公平性的影响.

概率模型检查是模型检查的一种扩展形式化验证技术, 常被用于一些概率系统或是概率相关性质的验证. 周女琪等人[46]将 Web 服务组合过程建立为定量多目标马尔科夫决策过程, 将不同的用户需求建模成多目标时序逻辑公式, 并使用 PRISM 概率模型检测器对其进行验证. 侯翌等人[47]定义了概率模型树行为到模型检测形式化模型的转换规则, 有效提高了机电系统可靠性评估的准确度与效率. 任胜兵等人[48]基于概率模型检查提出了一套软件缺陷定位方法, 设计了基于执行路径构建程序概率模型的学习算法. Gao 等人[49]基于概率模型检查方式对阿里巴巴余额宝数据集进行了分析, 以证明购买数量和赎回对用户行为的影响. 他们结合概率模型、离散时间马尔科夫决策过程以及概率计算树逻辑对整个金融过程以及用户行为进行形式化建模, 并使用 PRISM 工具进行验证与分析.

5 总结与讨论

本文提出了一种基于概率模型检查的方法来验证决策树和树集成模型的公平性问题. 基于统计均等和几率均等两种公平性, 定义了便于计算且可读性强的度量标准, 通过构建 PCSP#模型并进行概率模型检查, 得到在复合敏感属性群体条件下模型的阳性预测率, 从而衡量算法的群体公平性程度. 本文所提出的方法在多个数据集训练得到的分类器中都有很好的效果, 同时还能较好地处理大规模的决策树模型, 并提供了对于树集成模型的验证方法, 具有较好的可扩展性.

本文的方法给出了对于树集成模型公平性的统计性验证的解决办法，但仍然存在一定程度的误差，我们将在后续工作中关注这部分的内容. 此外，方法是基于预定义的敏感属性的，但在某些场景或应用中，敏感属性并不一定是明显的，且可能存在一些对公平性存在较大影响的“非敏感属性”，这也是一个值得研究的方向.

References:

[1] Saxena NA, Huang K, DeFilippis E, Radanovic G, Parkes DC, Liu Y. How do fairness definitions fare? Testing public attitudes towards three algorithmic definitions of fairness in loan allocations. Artificial Intelligence, 2020, 283: Article No.103238. [doi: 10.1016/j.artint.2020.103238]

[2] Liu WY, Shen CY, Wang XF, Jin B, Lu XJ, Wang XL, Zha HY, He JF. Survey on fairness in trustworthy machine learning. Ruan Jian Xue Bao/Journal of Software, 2021, 32(5): 1404−1426 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6214.htm [doi: 10.13328/j.cnki.jos.006214]

[3] Deng W, Xing YH, Li YF, Li ZH, Wang GY. Survey on fair machine learning. CAAI Trans. on Intelligent Systems, 2020, 15(3): 578−586 (in Chinese with English abstract).

[4] Verma S, Rubin J. Fairness definitions explained. In: Proc. of the 40th Int’l Conf. on Software Engineering. New York: Association for Computing Machinery, 2018. 1−7.

[5] Narayanan A. Tutorial: 21 fairness definitions and their politics. In: Proc. of the Conf. on Fairness, Accountability, and Transparency. New York: Association for Computing Machinery, 2018. 3−4.

[6] Caton S, Haas C. Fairness in machine learning: A survey. arXiv: 2010.04053, 2020.

[7] Ignatiev A, Cooper MC, Siala M, Hebrard E, Marques-Silva J. Towards formal fairness in machine learning. In: Proc. of the 26th Int’l Conf. on Principles and Practice of Constraint Programming. Online: Springer, 2020. 846−867.

[8] Calmon FP, Wei D, Vinzamuri B, Ramamurthy KN, Varshney KR. Optimized data pre-processing for discrimination prevention. In: Proc. of the 31st Int’l Conf. on Neural Information Processing Systems. 2017. 3995−4004.

[9] Xu D, Yuan S, Zhang L, Wu X. FairGAN: Fairness-aware generative adversarial networks. In: Proc. of the IEEE Int’l Conf. on Big Data. IEEE, 2018. 570−575.

[10] Shi XS, Li Y. Discriminatory sample identifying and removing algorithms based on margin in fairness machine learning. Scientia Sinica Informationis, 2020, 50(8): 1255−1266 (in Chinese with English abstract). [doi: 10.1360/SSI-2019-0112]

[11] Elisa Celis L, Huang L, Keswani V, Vishnoi NK. Classification with fairness constraints: A meta-algorithm with provable guarantees. In: Proc. of the Conf. on Fairness, Accountability, and Transparency. Atlanta: Association for Computing Machinery, 2019. 319−328.

[12] Zhang BH, Lemoine B, Mitchell M. Mitigating unwanted biases with adversarial learning. In: Proc. of the 2018 AAAI/ACM Conf. on AI, Ethics, and Society. 2018. 335−340.

[13] Hardt M, Price E, Srebro N. Equality of opportunity in supervised learning. In: Advances in Neural Information Processing Systems. 2016. 3323−3331.

[14] Wei D, Ramamurthy KN, Calmon F. Optimized score transformation for fair classification. In: Proc. of the Int’l Conf. on Artificial Intelligence and Statistics. 2020. 1673−1683.

[15] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: Proc. of the 22nd ACM SIGKDD Int’l Conf. on Knowledge Discovery and Data Mining. 2016. 785−794.

[16] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY. LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 2017, 30: 3146−3154.

[17] Albarghouthi A, D’Antoni L, Drews S, Nori AV. FairSquare: Probabilistic verification of program fairness. In: Proc. of the ACM on Programming Languages. 2017. 1−30.

[18] Bastani O, Zhang X, Solar-Lezama A. Probabilistic verification of fairness properties via concentration. In: Proc. of the ACM on Programming Languages. 2019. 1−27.

[19] Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R. Fairness through awareness. In: Proc. of the Innovations in Theoretical Computer Science Conf. 2012. 214−226.

[20] Yan S, Kao H Te, Ferrara E. Fair class balancing: enhancing model fairness without observing sensitive attributes. In: Proc. of the Int'l Conf. on Information and Knowledge Management. 2020. 1715−1724.

[21] Makhlouf K, Zhioua S, Palamidessi C. On the applicability of machine learning fairness notions. ACM SIGKDD Explorations Newsletter, 2021, 23(1): 14−23.

[22] Feldman M, Friedler SA, Moeller J, Scheidegger C, Venkatasubramanian S. Certifying and removing disparate impact. In: Proc. of the 21th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. 2015. 259−268.

[23] Sun J, Liu Y, Jin SD, Chen C. Integrating specification and programs for system modeling and verification. In: Proc. of the 3rd IEEE Int'l Symp. on Theoretical Aspects of Software Engineering. 2009. 127−135.

[24] Hoare C. Communicating sequential processes. Communications of the ACM, 1978, 21(8).

[25] Song SZ. Model checking stochastic systems in PAT [Ph.D. Thesis]. Singapore: National University of Singapore, 2013.

[26] Sun J, Liu Y, Dong JS, Pang J. PAT: Towards flexible verification under fairness. In: Proc. of the Int'l Conf. on Computer Aided Verification. Berlin: Springer, 2009. 709−714.

[27] Cui Z, Chen W, He Y, Chen Y. Optimal action extraction for random forests and boosted trees. In: Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. 2015. 179−188.

[28] Zhang G, Hou Z, Huang Y, Shi J, Bride H, Dong JS, Gao Y. Extracting optimal explanations for ensemble trees via logical reasoning. arXiv: 2103.02191, 2021.

[29] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: Proc. of the 6th Int'l Symp. on Micro Machine and Human Science (MHS'95). 1995. 39−43.

[30] Wang D, Tan D, Liu L. Particle swarm optimization algorithm: An overview. Soft Computing, 2018, 22(2): 387−408.

[31] Yang BW, Qian WY. Summary on improved inertia weight strategies for particle swarm optimization algorithm. Journal of Bohai University (Natural Science Edition), 2019, 40(3): 274−288 (in Chinese with English abstract). [doi: 10.13831/j.cnki.issn.1673-0569.2019.03.015]

[32] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. on Evolutionary Computation, 2002, 6(1): 58−73.

[33] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay É. Scikit-learn: Machine learning in python. Journal of Machine Learning Research, 2011, 12: 2825−2830. [doi: 10.5555/1953048.2078195]

[34] Ghosh B, Basu D, Meel KS. Justicia: A stochastic sat approach to formally verify fairness. In: Proc. of the AAAI Conf. on Artificial Intelligence. 2021. 7554−7563.

[35] Bride H, Cai CH, Dong J, Dong JS, Hou Z, Mirjalili S, Sun J. Silas: A high-performance machine learning foundation for logical reasoning and verification. Expert Systems with Applications, 2021, 176(1): Article No.114806.

[36] Chen H, Zhang H, Si S, Li Y, Boning D, Hsieh CJ. Robustness verification of tree-based models. Advances in Neural Information Processing Systems, 2019, 32: 1−12.

[37] Sato N, Kuruma H, Nakagawa Y, Ogawa H. Formal verification of a decision-tree ensemble model and detection of its violation ranges. IEICE Trans. on Information and Systems, 2020, E103D(2): 363−378.

[38] Devos L, Meert W, Davis J. Versatile verification of tree ensembles. In: Meila M, Zhang T, eds. Proc. of the 38th Int'l Conf. on Machine Learning, Vol.139. 2021. 2654−2664.

[39] Törnblom J, Nadjm-Tehrani S. Formal verification of random forests in safety-critical applications. In: Proc. of the Int'l Workshop on Formal Techniques for Safety-critical Systems. 2018. 55−71.

[40] Törnblom J, Nadjm-Tehrani S. An abstraction-refinement approach to formal verification of tree ensembles. In: Proc. of the Int'l Conf. on Computer Safety, Reliability, and Security. 2019. 301−313.

[41] Törnblom J, Nadjm-Tehrani S. Formal verification of input-output mappings of tree ensembles. Science of Computer Programming, 2020, 194: Article No.102450. [doi: 10.1016/j.scico.2020.102450]

[42] Kilbertus N, Rojas-Carulla M, Parascandolo G, Hardt M, Janzing D, Schölkopf B. Avoiding discrimination through causal reasoning. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. 2017. 656−666.

[43] Thomas PS, Da Silva BC, Barto AG, Giguere S, Brun Y, Brunskill E. Preventing undesirable behavior of intelligent machines. Science, 2019, 366(6468): 999−1004. [doi: 10.1126/science.aag3311]

[44] John PG, Vijaykeerthy D, Saha D. Verifying individual fairness in machine learning models. In: Proc. of the 36th Conf. on Uncertainty in Artificial Intelligence. 2020. 769−778.

[45] Ranzato F, Urban C, Zanella M. Fair training of decision tree classifiers. arXiv: 2101.00909, 2021.

[46] Zhou NQ, Zhou Y. Multi-objective verification of Web service composition based on probabilistic model checking. Computer Science, 2018, 45(8): 288−294 (in Chinese with English abstract). [doi: CNKI:SUN:JSJA.0.2018-08-054]

[47] Hou Y, Yang PL, Xu K. Research on conversion from probabilistic behavior tree model to formal model of model checking. Machinery Design and Manufacture, 2020, (8): 94−98 (in Chinese with English abstract). [doi: 10.19356/j.cnki.1001-3997.2020.08.022]

[48] Ren SB, Chen J, Tan WZ, Zuo X. Probabilistic model checking method for software fault location. Application Research of Computers, 2021, 38(11): 3387−3392, 3397 (in Chinese with English abstract). [doi: 10.19734/j.issn.1001-3695.2021.04.0132]

[49] Gao H, Mao S, Huang W, Yang X. Applying probabilistic model checking to financial production risk evaluation and control: A case study of Alibaba's Yu'e Bao. IEEE Trans. on Computational Social Systems, 2018, 5(3): 785−795.

附中文参考文献:

[2] 刘文炎, 沈楚云, 王祥丰, 金博, 卢兴见, 王晓玲, 查宏远, 何积丰. 可信机器学习的公平性综述. 软件学报, 2021, 32(5): 1404−1426. http://www.jos.org.cn/1000-9825/6214.htm [doi: 10.13328/j.cnki.jos.006214]

[3] 邓蔚, 邢钰晗, 李逸凡, 李振华, 王国胤. 公平性机器学习研究综述. 智能系统学报, 2020, 15(3): 578−586.

[10] 石鑫盛, 李云. 公平性机器学习中基于分类间隔的歧视样本发现和消除算法. 中国科学: 信息科学, 2020, 50(8): 1255−1266. [doi: 10.1360/SSI-2019-0112]

[31] 杨博雯, 钱伟懿. 粒子群优化算法中惯性权重改进策略综述. 渤海大学学报(自然科学版), 2019, 40(3): 274−288. [doi: 10.13831/j.cnki.issn.1673-0569.2019.03.015]

[46] 周女琪, 周宇. 基于概率模型检测的 Web 服务组合多目标验证. 计算机科学, 2018, 45(8): 288−294. [doi: CNKI:SUN:JSJA.0.2018-08-054]

[47] 侯翌, 杨培林, 徐凯. 概率行为树模型转化为模型检测模型方法研究. 机械设计与制造, 2020, (8): 94−98. [doi: 10.19356/j.cnki.1001-3997.2020.08.022]

[48] 任胜兵, 陈军, 谭文钊, 左兴. 基于概率模型检测的软件缺陷定位方法. 计算机应用研究, 2021, 38(11): 3387−3392, 3397. [doi: 10.19734/j.issn.1001-3695.2021.04.0132]

王艳(1995－), 女, 硕士生, CCF 学生会员, 主要研究领域为形式化验证, 可信人工智能.

史建琦(1984－), 男, 博士, 副研究员, 博士生导师, 主要研究领域为工业软件, 可信人工智能, 嵌入式控制系统.

侯哲(1988－), 男, 博士, 讲师, 博士生导师, 主要研究领域为自动推理, 形式化验证, 机器学习, 区块链.

张格林(1994－), 男, 硕士生, 主要研究领域为形式化验证, 可信人工智能.

黄滟鸿(1986－), 女, 博士, 副研究员, 主要研究领域为可信计算, 形式化建模与验证, 高可信嵌入式控制软件.