arXiv:2603.01870v1 [cs.LO] 2 Mar 2026

通过优化推广机器学习分类器的基于逻辑的解释

Francisco Mateus Rocha Filho, Ajalmar Rˆego da Rocha Neto, Thiago Alves Rocha*

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Fortaleza, Brazil

摘要

机器学习模型支持决策，但其预测背后的原因并不透明。清晰可靠的解释有助于用户做出明智的决策，避免盲目信任模型输出。然而，许多现有的解释方法无法保证正确性。基于逻辑的方法能够确保正确性，但往往会提供过度受限的解释，从而限制覆盖率。近期研究通过在保持正确性的同时逐步扩展解释来解决这一问题。该过程针对每个特征分别进行，并调整其上下界。然而，这种方法面临权衡：增量较小时计算成本较高，而增量较大时，解释可能覆盖较少的实例。为此，我们提出了两种新方法。Onestep 在这项先前工作的基础上，为每个特征及其每个边界一次性生成解释，消除了迭代过程的开销。*Twostep* 采用渐进式方法，以提高覆盖率。实验结果表明，与 Onestep 以及由此与先前工作相比，Twostep 显著提升了解释覆盖率——在各数据集上的平均提升幅度最高达 72.60%。

关键词：可解释人工智能。机器学习。基于逻辑的人工智能。

*通讯作者。*Email addresses:* francisco.mateus.rocha06@aluno.ifce.edu.br (Francisco Mateus Rocha Filho), ajalmar@ifce.edu.br (Ajalmar Rˆego da Rocha Neto),thiago.alves@ifce.edu.br (Thiago Alves Rocha)

1. 引言

人工智能（AI）日益融入日常生活，促使机器学习（ML）模型得到更广泛的应用（LeCun 等人，2015；Biswas，2023）。其中一些模型，例如支持向量机（SVM）和多层感知机（MLP），用于分类任务，能够根据输入数据做出决策，并在复杂的决策场景中为人类专家提供重要信息（Ramkumar 等人，2021；Yang 等人，2018）。因此，确保这些模型生成的输出可信且有正确性保证至关重要。可解释人工智能（XAI）的理念有助于更好地理解此类人工智能系统给出的答案；可解释性是构建可信模型的基础（Ignatiev 等人，2019年b；Ignatiev，2020年；Marques-Silva 与 Ignatiev，2022年；Samek 与 M¨uller，2019）。这有助于生成解释，从而更好地理解模型的决策过程。

在本研究中，我们探讨基于实例的解释（Ignatiev，2020年）。这些解释能够提供有关模型用于确定答案的决策逻辑的可解释洞见，突出显示给定实例中影响输出的相关特征。将输出与特定实例及其特征联系起来，有助于更好地理解模型的运行方式，并可能促成对决策过程更全面的分析。

可解释人工智能领域的大多数工作都缺乏正确性的形式化保证，因此很难判断模型是否可以完全被信任。用于生成解释的模型无关方法，例如 LIME（Ribeiro 等人，2016）、SHAP（Lundberg & Lee，2017）和 Anchors（Ribeiro 等人，2018），往往依赖启发式技术，这可能导致错误的模型解释。这些方法通常通过探索给定数据的局部实例空间来生成解释，而这可能不够充分，无法体现模型的真实行为（Ignatiev 等人，2020年）。此外，它们常常可以通过反例被证明是错误的，从而暴露其矛盾之处。已有进一步证据支持如下论断：这些方法缺乏正确性保

证（Dimanov 等人，2020年；Ignatiev 等人，2019年b；Ignatiev，2020年）。这使得此类解释在高风险应用（例如医疗保健和金融）中难以信任，因为错误结论会导致严重后果。因此，需要能够确保正确性的替代方法。

近年来，针对神经网络、朴素贝叶斯、随机森林、决策树和提升树等机器学习模型的基于逻辑的可解释人工智能研究激增（Audemard 等人，2022年；Ignatiev 等人，2022年、2019年a；Izza 等人，2020年；Marques-Silva 等人，2020年；Rocha Filho 等人，2024）。与启发式方法不同，基于逻辑的方法能够确保所得解释经过证明是正确的，从而提升可信度（Ignatiev，2020年）。此外，基于逻辑的方法还能提供非冗余的解释。这一点尤为重要，因为简洁的解释似乎更容易被人理解（Miller，1956），有助于人们聚焦于最重要的特征。通过确保解释既正确又非冗余，基于逻辑的解释对于那些可解释性不仅有用而且必不可少的应用至关重要，例如法律决策、医学诊断和贷款审批等高风险场景（Ignatiev，2020年；Ignatiev 等人，2020年）。

这类基于逻辑的解释通常称为溯因解释（Ignatiev et al., 2019a）或充分理由（Shih 等人，2018）。溯因解释可以看作由一组特征构成的子集，这些特征共同形成一条规则。每当应用这条规则时，模型的预测都保持不变（Ignatiev et al., 2019a；Shih 等人，2018）。然而，近期研究得出的大多数规则都由一组受限取值组成，形式为 $feature = value$，因此只适用于实例空间中范围狭窄的一组点。这类解释通常提供的信息非常有限，可能无法传达模型的整体洞见。例如，指出某项医学分析适用于身高 180 厘米、年龄 56 岁的患者，不如说明该分析也适用于身高在 175 至 185 厘米之间、年龄在 50 至 60 岁之间的患者来得有信息量。因此，包含

与取值范围相关的特征的解释更具洞察力，因为单个解释就能应用于各种情况。这类更具一般性的解释在现实世界应用中大有裨益。例如，在金融领域，这些解释可用于风险评估或贷款审批的预测模型。通过针对收入、信用记录或消费行为等金融特征提供更具一般性的解释，金融机构能够更深入地了解 AI驱动决策背后的依据。此外，若能确保解释既正确又非冗余，金融机构便可提供更易于理解的洞见，最终有助于构建更可信的金融体系。

尽管近年来基于逻辑的可解释人工智能方法有所发展，但针对正确性和极小性提供保证、同时侧重于更一般性解释的研究仍然不足。虽然 Anchors（Ribeiro 等人，2018）能够给出具有一定泛化能力的解释，但它是一种生成解释的启发式方法，因此在给出正确答案方面存在一些问题。近期的一项研究（Izza 等人，2023）将膨胀的溯因解释 定义为一组特征，其中每个特征都与一组取值相关联，且该组取值始终包含被解释实例的取值。此外，对于特征所关联的任何允许取值，预测都必须与被解释实例保持一致。Izza 等人（2023）提出的方法侧重于扩展溯因解释中的特征范围，并通过逐步调整每个特征的上下界来实现这一点，同时确保整个过程的正确性。对于每个特征，该算法将溯因解释中该特征可取值的初始下界和上界设为其原始取值。然后，算法尝试通过迭代添加一个较小的值来扩大上界，并在每次迭代中检查更新后的取值是否仍能保证预测成立。如果不能，则舍弃最后一个取值，并将前一个取值设为解释的上界。下界也以类似方式重复这一过程。不过，这种方法在计算解释的耗时方面可能面临挑战。较小的增量会导致大量迭代，从而增加

计算解释的成本。此外，较大的增量可能会使算法过早地判定预测无法保持一致。换言之，这可能导致解释中的特征范围较窄，进而造成覆盖率损失，限制解释成立的实例空间。

鉴于基于逻辑的可解释性和泛化对于构建可信的机器学习模型至关重要，本文在先前工作的基础上（Izza等人，2023），提出了两种用于扩展基于逻辑的解释的新方法。第一种方法 Onestep 通过优化，为每个特征的每个边界（上界和下界）确定解释范围，并在一步内完成，从而避免先前工作中逐步扩展所带来的低效。第二种方法 Twostep 针对 Onestep 的一项关键局限：扩展某些特征时，可能会过度缩小其他特征的取值范围，导致解释限制过多、泛化能力较弱。Twostep 先控制特定特征的扩展，从而使范围扩展更加均衡灵活，提升解释的泛化能力。

为了评估这些方法，我们在分类问题场景中采用支持向量机和多层感知机作为机器学习模型。我们通过求解一组逻辑约束来计算溯因解释，具体而言，这组约束是线性约束的布尔组合。我们借助一阶逻辑来实现这一点，因为由此得到的一阶公式可以表示为混合整数线性规划模型，并可由混合整数线性规划求解器求解。最后，我们为每个特征赋予一个取值范围，以此扩展解释。虽然我们的实验聚焦于支持向量机和多层感知机，但所提出的方法也可应用于其他分类器。

我们在 12 个不同的数据集上比较 Twostep 和 Onestep，并根据四项关键指标进行评估：计算时间、在原始数据集上的覆盖率、在合成数据上的覆盖率以及范围宽度。这些指标对于了解每种方法的效率和泛化能力至关重要。尤其是，在合成数据上的覆盖率很重要，因为真实数据集可能没有足够的实例来恰当地评估每种方法的泛化能

力。在比较中，Onestep 作为参照点，相较于早期工作提出的方法（Izza等人，2023）有所改进，效率更高，也避免了范围越界。通过比较 Twostep 和 Onestep，我们不仅评估了 Twostep 带来的改进，也考察了它在解释泛化方面与先前工作的表现。实验结果表明，Twostep 能生成更加一般化的解释，覆盖的实例最多增加 72.60%，而计算时间增加 55.65%。

本文的组织结构如下。第 2 节介绍本文使用的符号和定义。第 3 节给出一种用于计算溯因解释的算法。第 4 节介绍本文的核心贡献，重点阐述我们提出的两种扩展溯因解释的新方法：Onestep 和 Twostep。此外，第 5 节展示了针对所提出算法的实验、结果和讨论，目标分类器为支持向量机和多层感知机。最后，第 6 节介绍结论和未来工作。

2. 背景

分类问题。在机器学习中，分类问题定义在一组 n 特征 $\mathcal{F} = \{f_1, ..., f_n\}$ 和一组 $\mathcal{N}$类别 $\mathcal{K}$ =$\{c_1, c_2, ..., c_{\mathcal{N}}\}$之上。在本文中，我们假设每个特征 $f_i \in \mathcal{F}$的取值 x_i 均来自实数域。此外，每个特征 f_i 都有一个上界 u_i 和一个下界 l_i ，满足 $l_i \leq x_i \leq u_i$，其定义域为闭区间 $[l_i, u_i]$。这表示为一组域约束 $D = \{l_1 \leq f_1 \leq u_1,\ l_2 \leq f_2 \leq u_2, ..., l_n \leq f_n \leq u_n\}$。此外，符号$\mathbf{x} = \{f_1 = x_1, f_2 = x_2, ..., f_n = x_n\}$ 表示一个特定点或实例，满足每个 x_i 都属于 f_i的定义域。

分类器 $\mathcal{C}$ 是一个将特征空间中的元素映射到类别集合 $\mathcal{K}$中的函数。例如，$\mathcal{C}$可以将实例 $\{f_1 = x_1, f_2 = x_2, ..., f_n = x_n\}$映射到类别 c_1。分类器通常通过训练过程获得，训练过程的输入为训练集 $\{\mathbf{x}_i,\ y_i\}_{i=1}^{\mathcal{L}}$ 其中 $\mathbf{x}_i \in \mathbb{R}^n$ 是一个输入向量或实例，$y_i \in \mathcal{K}$ 是相应的类别标签，$\mathcal{L}$ 是训练样本的数量。

接着，对于每个输入向量 $\mathbf{x}_i$，其输入值 $x_{i,\ 1}$、 $x_{i,\ 2}$、...、$x_{i,\ n}$ 分别属于相应特征 f_1、...、 f_n的定义域。在本文中，我们考虑以下知名分类器：支持向量机，以及采用修正线性单元 （$ReLU$）（Nair & Hinton，2010）作为激活函数的多层感知机。接下来，我们将介绍这些分类器的一些细节。

支持向量机。支持向量机（支持向量机）（Cortes & Vapnik，1995年）是一种监督式机器学习模型，常用于分类问题，即作为支持向量分类器（支持向量分类器）。它利用（Cortes & Vapnik，1995年）所定义的最优分离超平面概念来分离数据。在IR^n 空间中，超平面由一组点$\mathbf{x}$ 定义，这些点满足$\mathbf{w} \cdot \mathbf{x} + b = 0$，其中$\mathbf{w} \in$ IR^n 为最优权重向量，$\mathbf{x} \in$ IR^n 为包含 n 个特征及截距（偏置） $b \in$ IR的特征向量。因此，用于对输入实例$\mathbf{x}$ 进行分类的决策函数定义为 $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$。

在本研究中，我们考虑二分类场景下的线性支持向量分类器。因此，假设类别集合为 $\mathcal{K} = \{c_1, c_2\}$，且 $c_1 = +1$、 $c_2 = -1$，则输入实例$\mathbf{x}$ 的预测结果$\hat{} y \in \mathcal{K}$ 计算如下：

$$\hat{y} = \begin{cases} +1, & \text{if } f(\mathbf{x}) \geq 0, \\ -1, & \text{if } f(\mathbf{x}) < 0. \end{cases} \tag{1}$$

多层感知机。多层感知机（MLP）（Bishop，1995年）是一种可用于处理分类问题的前馈神经网络（NN）。神经网络由 $K+1$ 层神经元组成。每一层 $k \in \{0, 1, ..., K\}$ 由 n_k 个神经元组成，编号从 1 到 n_k。令 $\mathbf{x}^k \in \mathbb{R}^{n_k}$ 表示第 k层的输出向量，并令 x_j^k 表示该层第 j个神经元的输出，其中 $j \in \{1, ..., n_k\}$。给定层中的神经元的输出值$\mathbf{x}^k = [x_1^k, x_2^k, ..., x_{n_k}^k]$ 通过前一层的输出值$\mathbf{x}^{k-1}$ 计算得到。假设第 0 层为输入层，且 $\mathbf{x}^0$ 为输入向量，则对其后的每一层重复这一过程，直至

到 K 层。

在每一对相邻层之间，都有一个权重矩阵 $W^{k-1}=[\mathbf{w}_1^{k-1},\ \mathbf{w}_2^{k-1},\ .....,\ \mathbf{w}_{n_k}^{k-1}]$，用于连接层 $k-1$ 和 k。此外，层 k中的神经元通过输出值 $\mathbf{x}^{k-1}$、权重矩阵 W^{k-1} 以及相应的偏置向量 $\mathbf{b}^{k-1}$计算中间值向量 $\mathbf{y}^k=[y_1^k,\ y_2^k,\ ...,\ y_j^k]$。每个神经元的输出值通过对 $\mathbf{y}^k$逐元素应用非线性激活函数来计算，例如修正线性单元($ReLU$)（Nair & Hinton，2010）。此外，最后一层 K 由 $\mathcal{N}$ 个神经元组成，每个类别对应一个神经元。因此，可得：

$$\begin{aligned}\mathbf{y}^k &= W^{k-1}\mathbf{x}^{k-1}+\mathbf{b}^{k-1}\\ \mathbf{x}^k &= ReLU(\mathbf{y^k}).\end{aligned} \tag{2}$$

其中 $x_j^k=max(0,\ \ y_j^k)$，对于 $j\in\{1,...,n_k\}$。因此，每个标量元素 x_j^k可通过以下方式计算：

$$\begin{aligned}y_j^k &= \mathbf{w}_j^{k-1\intercal}\mathbf{x}^{k-1}+b_j^{k-1}\\ x_j^k &= ReLU(y_j^k),\end{aligned} \tag{3}$$

其中，$^\intercal$ 是转置操作。

预测输出 $\hat{}y$ 是对最后一层 K中的神经元值进行 $argmax$ 计算所得的结果，计算如下：

$$\hat{\mathbf{y}}=argmax(\sigma(W^{K-1}\mathbf{x}^{K-1}+\mathbf{b}^{K-1})), \tag{4}$$

其中，σ 是Softmax函数。

可解释人工智能的启发式方法。LIME、SHAP 和 Anchors 是最知名的几种用于生成机器学习模型解释的启发式方法。这些模型无关方法能够生成局部解释，但不会将整个实例空间纳入考量（Lundberg & Lee, 2017; Ribeiro 等人，2016, 2018）。因此，解释可能失效：应用解释时，可能无法得到预期结果，并导致预测结果属于不同类别。此外，这些方法也容易包含无关元素

而这些元素本可以在不影响答案正确性的情况下移除（Ignatievetal., 2020），从而增加冗余。

Anchors 已被证明是优于 LIME 的版本，其所得解释具有更高的准确率（Ribeiro 等人，2018）。所获得的解释是决策规则，旨在突出给定实例的哪些特征足以使分类器做出某一预测，同时直观且易于理解。例如，考虑著名 IRIS 数据集中的一个实例$\mathbf{x}$ $=$ {萼片长度 $=4.0$，萼片宽度 $=1.8$，花瓣长度 $=3.7$，花瓣宽度$=1.2$}，该实例被分类器C预测为类别变色鸢尾。Anchors 获得的解释示例如以下决策规则：

- - - -

如果 萼片宽度 ≤ 3.0 **并且** 花瓣长度 >1.5 **则** 变色鸢尾。

这种解释具有一定程度的泛化，因为它可以应用于某个取值范围，而非实例空间中的一个特定点。在此情况下，它适用于所有遵循所述规则的实例。然而，这种启发式方法缺乏对正确性和极小性的保证，降低了生成的解释的可靠性。

一阶逻辑。我们使用一阶逻辑（FOL）（Kroening 和 Strichman，2016），通过线性实数算术理论上的无量词一阶公式来生成具有正确性保证的解释。一阶变量可以取实数值。这样便能够构造以下公式：

$$\begin{aligned} \varphi, \psi &:= s \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi) \mid (\neg\varphi) \mid (\varphi \rightarrow \psi), \\ s &:= \sum_{i=1}^{n} a_i z_i \leq e \mid \sum_{i=1}^{n} a_i z_i < e, \end{aligned} \tag{5}$$

其中 φ 和 ψ 是无量词一阶公式，且 s 表示满足 $n \geq 1$的原子公式，每个 z_i 是一阶变量，每个 e 和 a_i 是具体的实数。根据这一定义，$(2.5z_1 + 3.1z_2 \geq 6) \wedge ($ $z_1 = 1 \vee z_1 = 2) \wedge (z_1 = 2 \rightarrow z_2 \leq 1.1)$被视为一个公式。请注意，我们允许标准

缩写，例如用 $\neg(2.5z_1 + 3.1z_2 < 6)$表示 $2.5z_1 + 3.1z_2 \geq 6$。另请注意，我们允许使用 z_i以外的其他字母作为变量，例如 f_i、 s_i、 x_i^k、 o_i和 r_i。

假设公式的语义定义在实数域上，公式 φ 的赋值 $\mathcal{A}$ 是一个映射，它将 φ 的一阶变量映射到实数域中的元素。 $\mathcal{A}$ 若在 $\mathcal{A}$下公式 φ 为真，则满足该公式。 $\{z_1 \mapsto 2,\ z_2 \mapsto 1.05\}$ 例如，$2.5z_1 + 3.1z_2 \geq 6(\wedge)$ $z_1 = 1 \vee z_1 = 2(\wedge) z_1 = 2 \rightarrow z_2 \leq 1.1$满足该公式，而 $\{z_1 \mapsto 2.3,\ z_2 \mapsto 1\}$ 则不满足。

如果存在一个满足公式 φ的 $\mathcal{A}$，那么 φ是可满足的。因此，上例中的公式是可满足的，因为 $\{z_1 \mapsto 2, z_2 \mapsto 1.05\}$ 满足该公式。可满足性的概念也可以扩展到公式集Γ。如果存在一个 $\mathcal{A}$ ，使得 Γ 中的所有公式同时为真，那么一阶公式集就是可满足的。例如， $\{(2.5z_1 + 3.1z_2 \geq 6),\ (z_1 = 1 \vee z_1 = 2),\ (z_1 = 2 \rightarrow z_2 \leq 1.1)\}$ 是可满足的，因为 $\{z_1 \mapsto 2, z_2 \mapsto 1.05\}$同时满足集合中的每个公式。众所周知，对于所有公式集 Γ 以及所有公式 φ 和 ψ,

$$\begin{aligned}\Gamma \cup \{\varphi \vee \psi\} \text{ is unsatisfiable iff } \Gamma \cup \{\varphi\} \text{ is unsatisfiable and} \\ \Gamma \cup \{\psi\} \text{ is unsatisfiable.}\end{aligned} \tag{6}$$

对于给定的 φ和公式集 Γ，可以使用记号 $\Gamma \models \varphi$表示逻辑后承。这意味着，满足 Γ 中所有公式的每个赋值 $\mathcal{A}$也满足 φ。举例来说，设 Γ为 $\{z_1 = 2, z_2 \geq 1\}$，且 φ 为 $(2.5z_1 + z_2 \geq 5) \wedge (z_1 = 1 \vee z_1 = 2)$。那么，$\Gamma \models \varphi$，因为满足 Γ 中所有公式的每个满足赋值也满足 φ。此外，众所周知，对于所有公式集 Γ 和所有公式 φ,

$$\Gamma \models \varphi \text{ iff } \Gamma \cup \{\neg\varphi\} \text{ is unsatisfiable.} \tag{7}$$

例如，不存在能满足 $\{z_1 = 2, z_2 \geq 1, \neg((2.5z_1 + z_2 \geq 5) \wedge (z_1 = 1 \vee z_1 = 2))\}$ 的赋值 $\mathcal{A}$，因为满足 $(z_1 = 2 \wedge z_2 \geq 1)$ 的 $\mathcal{A}$ 也满足 $(2.5z_1 + z_2 \geq 5) \wedge (z_1 = 1 \vee z_1 = 2)$，因此不满足$\neg((2.5z_1 + z_2 \geq 5) \wedge (z_1 = 1 \vee z_1 = 2))$。

如果对于每个赋值 $\mathcal{A}$，φ 和 ψ 在 $\mathcal{A}$ 下都为真，或者在 $\mathcal{A}$ 下都为假，我们就称两个一阶公式 φ 和 ψ 等价。我们用记号 $\varphi \equiv \psi$ 表示 φ 和 ψ 等价。例如，$\neg((z_1 + z_2 \leq 2) \wedge z_1 \geq 1) \equiv (\neg(z_1 + z_2 \leq 2) \vee \neg(z_1 \geq 1))$。此外，这两个公式也等价于 $((z_1 + z_2 > 2) \vee z_1 < 1)$。

3. 计算基于逻辑的解释

如前所述，大多数用于解释机器学习模型的方法都是启发式的，这意味着由于缺乏正确性保证，这些方法给出的解释无法得到充分信任。本节介绍一种通过将分类器编码为一阶公式来解决这一问题的方法，正如（Ignatiev et al., 2019a）所展示的那样。在这项工作中，给定一个待解释的实例及其对应的预测，溯因解释是由特征及其对应取值构成的子集极小集。此外，溯因解释保证：如果这些特征取指定值，就始终会产生相同的预测。为了说明（Ignatiev et al., 2019a）中介绍的方法，我们以线性支持向量分类器和多层感知机作为分类器的示例。

线性支持向量分类器。在二分类场景中，给定一个由权重向量$\mathbf{w}$ 和偏置 b定义的训练后线性支持向量分类器，我们用以下一阶公式表示该分类器：

$$P = (\sum_{i=1}^{n} w_i f_i + b \geq 0). \tag{8}$$

该公式对所有由支持向量分类器分类为 $c = +1$ 的点 $\mathbf{x}$ 均成立。对于实例 $\mathbf{x} = \{f_1 = x_1, f_2 = x_2, ..., f_n = x_n\}$，若特征的定义域由 $D = \{l_1 \leq f_1 \leq u_1, l_2 \leq f_2 \leq u_2, ..., l_n \leq f_n \leq u_n\}$ 定义，则解释是一个子集 $E \subseteq \mathbf{x}$，满足 $E \cup D \models P$。满足 $E \cup D$ 中所有公式的每个赋值也必须满足 P，从而确保预测保持不变。由于我们的目标是获得极小解释，E 必须满足：对于所有真子集 $E' \subset E$，均有 $E' \cup D \not\models P$。需要注意的是，E 中的值对应于实例 $\mathbf{x}$ 的原始值，而

$\mathbf{x} - E$ 中特征的值可以在定义域约束 D 内取任意值，而不影响预测。

对于预测为其他类别的实例，该方法仍然相似。例如，针对被分类为 $c = -1$ 的实例，一阶公式 P 定义为

$$P = (\sum_{i=1}^{n} w_i f_i + b < 0). \tag{9}$$

遵循 Ignatiev 等人（2019年a）提出的思路，我们使用算法 1 为给定的 $\mathbf{x}$、D 和 P 计算溯因解释。对于分类器预测的类别 c，相应地定义 P。算法从将 E 初始化为 $\mathbf{x}$ 开始，并迭代地移除特征。如果对于某个特征 $f_i = x_i \in E$，有 $(E \setminus \{f_i = x_i\}) \cup D \models P$，那么特征 f_i 的值 x_i 对于确保预测是不必要的，因此将其从 $f_i = x_i$ 中移除。否则，它会保留在E中。该过程会针对所有特征执行。算法结束时，可确保解释是正确的，并且不受已移除特征的取值影响。

根据（7）的结果，算法 1 中形式为 $(E \setminus \{f_i = x_i\}) \cup D \models P$ 的蕴涵关系，可以通过检验一阶公式集合 $(E \setminus \{f_i = x_i\}) \cup D \cup \{\neg P\}$ 是否不可满足来验证。此外，由于 $(E \setminus \{f_i = x_i\}) \cup D$ 是线性约束集合，且 P 是线性约束，因此可以使用线性规划（LP）求解器高效地检查其不可满足性。如果线性约束集合 $(E \setminus \{f_i = x_i\}) \cup D \cup \{\neg P\}$ 有解，则 $(E \setminus \{f_i = x_i\}) \cup D \not\models P$。从计算复杂度的角度来看，线性规划问题可在多项式时间内求解。因此，计算线性支持向量分类器的溯因解释也可在多项式时间内完成，所需的 LP 求解器调用次数与特征数量成线性关系。

算法 1 计算溯因解释**输入：** 预测为 c 的实例 $\mathbf{x}$，域约束 D，根据预测 c 得到的公式 P **输出：** 溯因解释 $EE \leftarrow$**x 对于** $f_i = x_i \in E$ **执行 如果** $(E \backslash \{f_i = x_i\}) \cup D \models P$ **则** $E \leftarrow E \backslash \{f_i = x_i\}$ **返回** E

多层感知机。我们遵循将神经网络模型表示为混合整数线性规划（MILP）模型的概念，如（Fischetti & Jo, 2018）所示。对于每一层 $1 \leq k \leq K-1$，第 j 个神经元的 $ReLU$ 输出 x_j^k 通过以下线性约束j得到：

$$\sum_{i=1}^{n_{k-1}} w_{ij}^{k-1} x_i^{k-1} + b_j^{k-1} = x_j^k - s_j^k, \quad x_j^k \geq 0, \quad s_j^k \geq 0. \tag{10}$$

我们还必须施加约束，使两个项 x_j^k 和 s_j^k 中至少有一个为零。然后，我们引入一个二元变量 $z_j^k \in \{0, 1\}$，并施加以下逻辑蕴含

$$z_j^k = 1 \rightarrow x_j^k \leq 0$$
$$z_j^k = 0 \rightarrow s_j^k \leq 0.$$

如果 z_j^k 等于 1，则 ReLU 输出 x_j^k 为 0，且 $-s_j^k$ 等于 ReLU 输入。否则，输出 x_j^k 等于 ReLU 输入，且 s_j^k 等于

0。MLP 的最终公式见约束 (11)-(13)：

$$\left.\begin{aligned} \sum_{i=1}^{n_{k-1}} w_{ij}^{k-1} x_i^{k-1} + b_j^{k-1} &= x_j^k - s_j^k \\ x_j^k &\geq 0 \\ s_j^k &\geq 0 \\ z_j^k &\in \{0,1\} \\ z_j^k = 1 &\rightarrow x_j^k \leq 0 \\ z_j^k = 0 &\rightarrow s_j^k \leq 0 \end{aligned}\right\} k = 1,...,K, \quad j = 1,...,n_k \tag{11}$$

$$l_i \leq x_i^0 \leq u_i, \quad i = 1,...,n \tag{12}$$

$$o_i = \sum_{j=1}^{n_{K-1}} w_{i,j}^K x_i^{K-1} + b_i^K, \quad i = 1,...,\mathcal{N} \tag{13}$$

在该表述中，权重 w_{ij}^{k-1} 和偏置 b_j^k 是给定的固定参数。此外，每个变量 x_i^0 表示一个输入特征 f_i，其上下界分别为 l_i和 u_i，由特征的定义域确定。我们假设约束（11）—（13）表示一个公式 D。请注意，在这种情况下，对于MLP分类器而言， D 由域约束以及神经网络的表述组成。给定一个在多类分类情境下被分类为类别 $c_{j'}$ 的实例 $\mathbf{x}$，我们将一阶公式 P 定义为：

$$P = \bigwedge_{j \neq j'} o_{j'} > o_j. \tag{14}$$

该公式 P 表示，与类别$c_{j'}$ 相关的输出神经元$o_{j'}$ 的值高于其他输出神经元的值。请注意，我们无需考虑 Softmax函数，因为它不会改变最后一层的最大值。与 (Ignatiev et al., 2019a) 中提出的方法类似，$\mathbf{x}$ 的生成的解释是一个子集 $E \subseteq \mathbf{x}$，满足 $E \cup D \models P$；任何满足 $E \cup D$ 中所有公式的赋值也必须满足P。同样， E 必须是极小的，即对于所有子集 $E' \subset E$，都必须满足 $E' \cup D \not\models P$。因此，我们必须在考虑这些要求的同时，进一步阐述 P。

根据 (7)， $E \cup D \models P$当且仅当 $E \cup D \cup \{\neg P\}$ 不可满足时成立。现代混合整数线性规划求解器能够处理 $E \cup D$ 中的约束。我们将 $\neg P$ 表示为

$$\begin{aligned} &\neg(o_{j'} > o_1 \wedge ... \wedge o_{j'} > o_{\mathcal{N}}), \quad \text{which is equivalent to} \\ &(o_{j'} \leq o_1 \vee ... \vee o_{j'} \leq o_{\mathcal{N}}). \end{aligned} \tag{15}$$

因此，为了将这一析取表示为现代混合整数线性规划求解器可接受的约束，我们为每个输出神经元 o_j 添加一个二元变量 r_j，使得 $j \neq j'$，定义如下：

$$\sum_{j \neq j'} r_j \geq 1 \tag{16}$$

$$\left.\begin{aligned} r_j = 1 \rightarrow o_{j'} \leq o_j \\ r_j \in \{0,1\} \end{aligned}\right\} j \in \{1,...,\mathcal{N}\}, \quad j \neq j'. \tag{17}$$

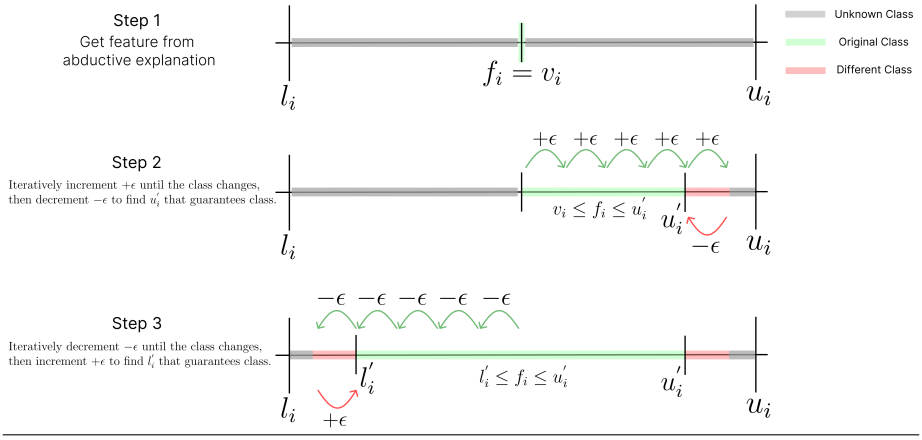
因此，我们可以使用现代混合整数线性规划求解器检查 $E \cup D \cup \{\neg P\}$ 是否不可满足。此外，我们还可以使用算法 1为 MLP分类器计算溯因解释。

4. 通过优化膨胀基于逻辑的解释

如第 3 节所述，得到的溯因解释 $E \subseteq \mathbf{x}$包含特征及其对应的取值；无论$\mathbf{x} - E$中的其他特征如何，这些取值都会导向同一个类别。然而，某些特征的取值即使发生变化$\mathbf{x}$ E，也仍有可能得到相同的决策。因此，$\mathbf{x}$ E 中包含的特征是扩展的目标。膨胀溯因解释 由一组特征构成，每个特征都对应一个取值范围，并且在允许的取值范围内，无论取何值，预测结果都保持不变。

早期研究（Izza等人，2023）提出了一种方法：从溯因解释出发，通过迭代调整每个特征的上界和下界来扩展特征范围，同时确保预测保持不变。图 1 展示了这一策略。然而，较小的增量可能导致迭代次数过多，从而增加计算成本。另一方面，较大的增量可能导致过冲，使人过早得出预测不成立的结论，进而得到更窄的特征范围。

图 1 示例：如何使用增量方法计算特征 f_i 的取值范围。



在此基础上，我们提出了一种更高效的计算膨胀溯因解释的方法。我们首个新方法名为 **Onestep**，其核心区别在于使用优化，在一步之内找到取值范围的最大值和最小值，而不是采用增量方法。我们的方法能够最大限度地减少早期研究（Izza等人，2023）中增量大小所带来的问题。

我们将解释扩展任务定义为一组优化问题。在初始步骤中，给定溯因解释 E中的 $f_i = x_i$ ，需要找到 u_i'，使得 $x_i \leq u_i' \leq u_i$且 $(E \setminus \{f_i = x_i\}) \cup \{f_i \leq u_i'\} \cup D \models P$，即用 $f_i \leq u_i'$ 替换 $f_i = x_i$ 不会改变预测，从而确保它们仍属于同一类别。我们通过寻找高于原始值 x_i 、且会使预测发生变化的最低值t_i^* 来确定 u_i' 。以下是用于寻找 t_i^*的优化模型描述：

$$
\begin{aligned}
min \quad & f_i \\
s.t. \quad & x_i \leq f_i \leq u_i \\
& f_j = x_j, \quad \text{for all } f_j = x_j \in E \text{ with } j \neq i \\
& l_g \leq f_g \leq u_g, \quad \text{for all } f_g = x_g \in \mathbf{x} \setminus E \\
& \mathcal{R}
\end{aligned}
\tag{18}
$$

其中 x_i 和 x_j 分别是特征 f_i, f_j的值， $f_j = x_j \in E$ 是溯因解释中尚未找到范围的特征，而 $f_g = x_g \in \mathbf{x} \backslash E$ 是溯因解释之外的特征 E，并且$\mathcal{R}$是一个要求预测不成立的约束。该约束取决于分类器。对于支持向量分类器，如果预测结果为类别 $c = +1$，则$\mathcal{R}$ 为（9）中的 P，即 $P =$ （$\sum_{i=1}^{n} w_i f_i + b < 0$）。否则，如果 $c = -1$，则 $\mathcal{R}$ 为（8）中的P。对多层感知机也采用类似过程。

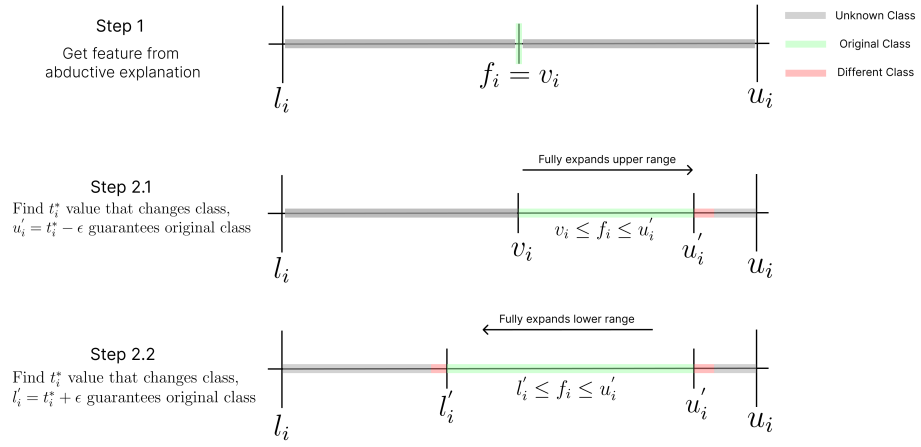
如果问题中的约束集可满足，则可以找到特征 f_i 的最优解值 t_i^*。因此，它是高于原始值且能改变预测结果的最小值，即 $x_i < t_i^* \leq u_i$。所以，我们设定 $u_i' = t_i^* - \varepsilon$，并假设 ε 是一个很小的正实数值。此外，如果 $t_i^* - \varepsilon < x_i$，则我们设定 $u_i' = x_i$，以保证 $x_i \leq u_i'$。从 $t_i^* - \varepsilon$ 得到的值是 f_i 在确保对 $[x_i, u_i']$ 中的所有值，预测结果都成立的情况下可取的最大值（精度误差不超过 ε）。反之，如果约束集不可满足，则不存在任何 f_i 的值能满足 $x_i < f_i \leq u_i$ 并改变预测。因此，我们设定 $u_i' = u_i$。下界 l_i' 也采用类似的处理过程：

$$
\begin{aligned}
max \quad & f_i \\
s.t. \quad & l_i \leq f_i \leq x_i \\
& f_j = x_j, \quad \text{for all } f_j = x_j \in E \text{ with } j \neq i \\
& l_g \leq f_g \leq u_g, \quad \text{for all } f_g = x_g \in \mathbf{x} - E \\
& \mathcal{R}
\end{aligned}
\tag{19}
$$

如果该问题可满足，则求得 t_i^* 对于 f_i 的最优值。因此，t_i^*是低于原始值的最高值，即会使预测发生改变的 $l_i \leq t_i^* < x_i$。在此情况下，我们设定 $l_i' = t_i^* + \varepsilon$。于是， $l_i^{'}$ 是使预测保持成立的最低值。此外，若 $t_i^* + \varepsilon > x_i$，则我们设定 $l_i' = x_i$，从而保证 $l_i' \leq x_i$。否则，若该问题不可满足，我们设定 $l_i' = l_i$。因此，当该特征在取值范围 $[l_i', u_i']$内取任意值，且其余每个特征 f_j 满足 $f_j = x_j \in E$时，预测结果成立

仍被设定为它们在溯因解释中的值。图2展示了如何用Onestep计算特征 f_i 的取值范围的简化示意图。

图2 示例：如何使用 Onestep 计算特征 f_i 的取值范围。



接下来，对于每个其他特征，我们通过以下方式找出取值范围的新最大值和最小值

$$
\begin{aligned}
min \quad & f_i \\
s.t. \quad & l_h' \le f_h \le u_h', \quad \text{for all } f_h \text{ such that } f_h = x_j \in E \text{ and } h < i \\
& x_i \le f_i \le u_i, \\
& f_j = x_j, \quad \text{for all } f_j \text{ such that } f_j = x_j \in E \text{ and } i < j \le n \\
& l_g \le f_g \le u_g, \quad \text{for all } f_g \text{ such that } f_g = x_g \in \mathbf{x} - E \\
& \mathcal{R}
\end{aligned}
\tag{20}
$$

and

$$
\begin{aligned}
& max \quad f_i s.t. \quad l_h' \le f_h \le u_h', \text{ 对于所有满足 } f_h = x_j \in E \text{ 且 } h < i \\
& l_i \le f_i \le x_i, \ f_j = x_j, \text{ 对于所有满足 } f_j = x_j \in E \text{ 且 } i < j \le n \\
& l_g \le f_g \le u_g, \text{ 对于所有满足 } f_g = x_g \in \mathbf{x} - E \mathcal{R}
\end{aligned}
\tag{21}
$$

分别。请注意，对于所有满足 $h < i$ 的 $f_h \in E$ ， l'_h 和 u'_h 是先前找到的、能够使预测成立的取值范围。我们的方法 Onestep 如算法 2 所示。该算法中的方法求解 会直接计算 u'_i 的值，如前所述。

算法 2 Onestep

输入： 预测为 c的实例$\mathbf{x}$，域约束 D，根据 c的约束 $\mathcal{R}$ ，溯因解释 E

输出： 膨胀溯因解释 E_{xp}

$h \leftarrow \{\}$

$E_{xp} \leftarrow \{\}$

令 $f_i = x_i \in E$

最小优化模型 $\leftarrow$ 创建第一个最小模型($\mathbf{x}$, D, $\mathcal{R}$, E, $f_i = x_i$) ▷ 18

最大优化模型 $\leftarrow$ 创建第一个最大模型($\mathbf{x}$, D, $\mathcal{R}$, E, $f_i = x_i$) ▷ 19

$u'_i \leftarrow$求解(最小优化模型)

$l'_i \leftarrow$求解(最大优化模型)

$h \leftarrow h \cup \{f_i = x_i\}$

$E_{xp} \leftarrow E_{xp} \cup \{l'_i \leq f_i \leq u'_i\}$

对于 $f_i = x_i \in E$ 使得 $f_i = x_i \notin h$ **执行**

- 最小优化模型 $\leftarrow$ 创建最小模型($\mathbf{x}$, D, $\mathcal{R}$, E_{xp}, $f_i = x_i$, E) ▷ 20
- 最大优化模型 $\leftarrow$ 创建最大模型($\mathbf{x}$, D, $\mathcal{R}$, E_{xp}, $f_i = x_i$, E) ▷ 21
- $u'_i \leftarrow$求解(min 优化模型)
- $l'_i \leftarrow$求解(max 优化模型)
- $h \leftarrow h \cup \{f_i = x_i\}$
- $E_{xp} \leftarrow E_{xp} \cup \{l'_i \leq f_i \leq u'_i\}$

返回 E_{xp}

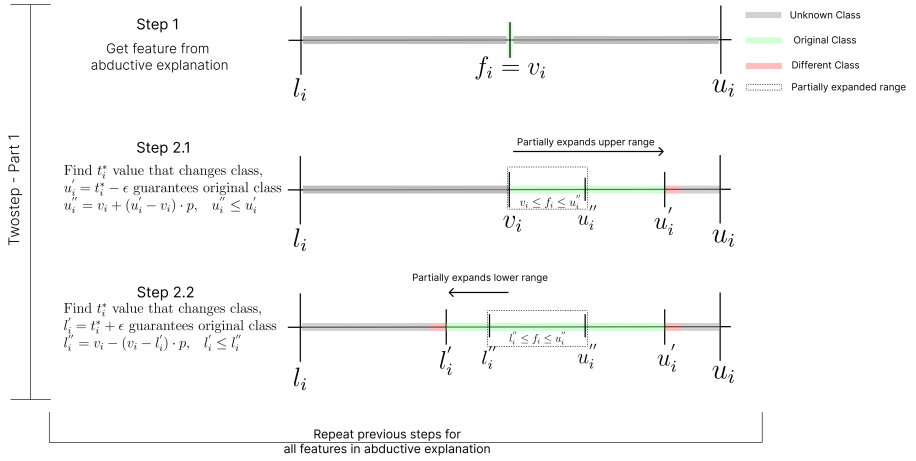
我们的第二种方法名为**Twostep**，其思路与 Onestep 类似，主要区别在于如何为每个特征f_i设定取值范围。虽然 Onestep 能够找到保证该预测的范围，但某些特征可能被扩展得过多，从而限制了其他特征的可能范围。因此，这些其他特征被限制在非常小的范围内。此外，有可能出现 $l'_i = u'_i = x_i$的情况，也就是说，由于任何其他取值都会使预测发生变化，该范围无法被扩展。

我们的方法 Twostep 是一种降低此类情况发生率的策略。具体做法是缩小某些特征的取值范围，以便扩大其他特征的取值范围。我们将 l''_i和 u''_i定义为 $[l'_i,u'_i]$内某个子范围的下限和上限，即 $l'_i \leq l''_i$、 $u''_i \leq u'_i$和 $l''_i \leq u''_i$。此外，我们设定 $l''_i \leq f_i \leq u''_i$，指的是

该特征 f_i 在此子范围内可以取的所有值。

在 Twostep 的初始阶段，我们按照 Onestep 中所示确定 l_i' 和 u_i' 。然后，设置 $l_i'' = x_i - (x_i - l_i') \cdot p$ 和 $u_i'' = x_i + (u_i' - x_i) \cdot p$，其中 p 是一个给定参数，其值为 $0 < p \leq 1$ ，用于控制子范围的大小。请注意， $p = 1$ 等同于 Onestep。例如，如果 $p = 0.5$，那么 $l_i'' = x_i - (x_i - l_i') \cdot 0.5$ 和 $u_i'' = x_i + (u_i' - x_i) \cdot 0.5$，也就是说，子范围的下限和上限分别等于相应原始范围界限的 50%。图3 展示了特征 f_i的 Twostep 初始阶段。

图3 展示如何使用 Twostep 计算 f_i 的部分取值范围。



我们算法的第二步包括在确保预测仍然成立的同时扩展已找到的范围。因此，我们可以采用以下优化问题来扩展这些范围，同时考虑在我们方法的第一步中为 u_i''和 l_i''找到的值:

$$min \quad f_i s.t. \quad l_h' \leq f_h \leq u_h', \text{ 对所有满足 } f_h = x_j \in E \text{ 且 } h < i \text{ 的 } f_h \, u_i'' \leq f_i \leq u_i, \; l_j'' \leq f_j \leq u_j'', \text{ 对所有满足 } f_j = x_j \in E \text{ 且 } i < j \leq n \text{ 的 } f_j \, l_g \leq f_g \leq u_g, \text{ 对所有满足 } f_g = x_g \in \mathbf{x} \text{ 的 } f_g - E \; \mathcal{R} \tag{22}$$

$max \quad f_i s.t. \quad l'_h \leq f_h \leq u'_h$，对所有满足 $f_h = x_j \in E$ 且 $h < i$ 的 $f_h\, l_i \leq f_i \leq l''_i$，$l''_j \leq f_j \leq u''_j$，对所有满足 $f_j = x_j \in E$ 且 $i < j \leq n$ 的 $f_j\, l_g \leq f_g \leq u_g$，对所有满足 $f_g = x_g \in \mathbf{x}$ 的 $f_g - E \mathcal{R}$.

(23)

算法3描述了 Twostep 方法。该算法的第一步是通过式 (18) 和 (19) 为第一个特征求出 u'_i 和 l'_i。然后，根据 p缩小 u'_i 和 l'_i，得到中间界 u''_i和 l''_i。对于后续每个特征，我们使用式 (20) 和 (21) 求出 u'_i 和 l'_i，进而求出 u''_i和 l''_i。该算法的第二步是使用式 (22-23) 对每个特征尽可能扩大已求出的范围，同时确保预测仍然成立。算法输出一个膨胀溯因解释 E_{xp}，其中包含每个特征在不改变预测的情况下可以取到的更新后取值范围。

算法3 Twostep

输入： 预测为 c 的实例 $\mathbf{x}$，域约束 D，根据 c的限制 $\mathcal{R}$ ，溯因解释 E，参数 p

输出： 膨胀溯因解释 E_{xp}

$h \leftarrow \{\}$ ▷ 步骤 1

$E_{in} \leftarrow \{\}$

令 $f_i = x_i \in E$

最小优化模型 $\leftarrow$ 创建第一个最小模型($\mathbf{x}$, D, $\mathcal{R}$, E, $f_i = x_i$) ▷ 18

最大优化模型 $\leftarrow$ 创建第一个最大模型($\mathbf{x}$, D, $\mathcal{R}$, E, $f_i = x_i$) ▷ 19

u_i', $l_i' \leftarrow$求解(最小优化模型), 求解(最大优化模型)

u_i'', $l_i'' \leftarrow x_i + (u_i' - x_i) \cdot p$, $x_i - (x_i - l_i') \cdot p$

$h \leftarrow h \cup \{f_i = x_i\}$

$E_{in} \leftarrow E_{in} \cup \{l_i'' \leq f_i \leq u_i''\}$

对于 $f_i = x_i \in E$ 使得 $f_i = x_i \notin h$ **执行**

最小化优化模型 $\leftarrow$ 创建最小模型($\mathbf{x}$, D, $\mathcal{R}$, E_{in}, $f_i = x_i$) ▷ 20

最大化优化模型 $\leftarrow$ 创建最太模型($\mathbf{x}$, D, $\mathcal{R}$, E_{in}, $f_i = x_i$) ▷ 21

u_i', $l_i' \leftarrow$求解(最小化优化模型), 求解(最大化优化模型)

u_i'', $l_i'' \leftarrow x_i + (u_i' - x_i) \cdot p$, $x_i - (x_i - l_i') \cdot p$

$h \leftarrow h \cup \{f_i = x_i\}$

$E_{in} \leftarrow E_{in} \cup \{l_i'' \leq f_i \leq u_i''\}$

$h \leftarrow \{\}$ ▷ 步骤 2

$E_{xp} \leftarrow \{\}$

对于 $l_i'' \leq f_i \leq u_i'' \in E_{in}$ 使得 $l_i'' \leq f_i \leq u_i'' \notin h$ **执行**

最小优化模型 $\leftarrow$ 创建最小模型($\mathbf{x}$, D, $\mathcal{R}$, E_{xp}, $l_i'' \leq f_i \leq u_i''$, E_{in}) ▷ 22

最大优化模型 $\leftarrow$ 创建最大模型($\mathbf{x}$, D, $\mathcal{R}$, E_{xp}, $l_i'' \leq f_i \leq u_i''$, E_{in}) ▷ 23

u_i', $l_i' \leftarrow$求解(最小优化模型), 求解(max 优化模型)

$h \leftarrow h \cup \{l_i'' \leq f_i \leq u_i''\}$

$E_{xp} \leftarrow E_{xp} \cup \{l_i' \leq f_i \leq u_i'\}$

返回 E_{xp}

5. 实验

在这项工作中，我们在实验[1]中共使用了 12 个数据集：输血、脊柱、帕金森病、电离层、Banknote Authentica- tion、气候模型模拟崩溃、玻璃和用户知识建模，这些数据集可从 UCI机器学习资料库[2]获取；Pima Indians Dia- betes 可从 Kaggle[3]获取；IRIS、威斯康星乳腺癌和 WINE 可从

[1]代码可在此处获取：https://github.com/franciscomateus0119/GExp[2] https://archive.ics.uci.edu/ml/datasets[3] https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

通过 scikit-learn 软件包[4]。按照标准流程，所有特征均已缩放至取值范围 $[0, 1]$,。

为确保在线性支持向量分类器的情境下进行二分类，Iris、Wine、User Knowledge Modeling 和 Glass 数据集中的类别均被二值化。对于多层感知机，使用这些数据集的原始版本，即 Iris 和 Wine 各有三个类别，User Knowledge Modeling 有四个类别，Glass 有六个类别。其余数据集本身已包含两个类别。由于我们假设特征取值为连续值，因此已移除所有类别数据。经过上述修改后的数据集概要见表1。

表1 数据集详情。

数据集	缩写	特征数量	实例
Iris	IRIS	4	150
输血	BLDT	4	748
纸币鉴别	BANK	4	1372
用户知识建模	UKMO	5	403
脊柱	VRTC	6	310
Pima	PIMA	8	768
玻璃	GLAS	9	214
Wine	WINE	13	178
气候模型模拟崩溃	CLIM	18	540
帕金森病	PARK	22	195
威斯康星乳腺癌	BRCW	30	569
电离层	IONS	34	351

支持向量机分类器。对于每个数据集，使用原始数据的 25% 训练了一个线性支持向量分类器。使用了正则化参数 $C = 1$ ，并采用分层抽样。其他参数均设为 scikit-learn 软件包中规定的默认值。在各数据集上训练的线性支持向量机分类器的平均准确率为 87%。

[4]https://github.com/scikit-learn/scikit-learn/tree/main/sklearn/datasets/data

MLP分类器。对于每个数据集，使用原始数据的 25% 训练了一个多层感知机。模型使用单个隐藏层，其神经元数量与输入层大小相同，并采用 ReLU 作为激活函数。在输出层使用了 softmax 激活。模型使用 Adam优化器进行编译。每个模型训练 400 轮，批大小为 32，早停耐心值设为 40，并且仅保存最佳模型。训练采用分层抽样。在各数据集上训练的多层感知器分类器的平均准确率为 83%。

需要指出的是，在我们的实验中，用于训练的实例比例并非主要关注点，因为我们的解释方法旨在解释训练模型，而不是数据本身。在评估中，我们会为测试实例生成解释，这意味着测试实例数量越多，就越能全面地评估这些解释方法。

我们的方法。 Onestep 和 Twostep 的原型实现均采用 Python 编写，并遵循算法 2 和 3。在比较中，Onestep 作为参照，因为它改进了早期工作中提出的方法（Izza 等人，2023），效率更高且避免了取值范围过冲。因此，将 Twostep 与 Onestep 进行比较，不仅能评估 Twostep 带来的改进，还能评估其在解释泛化方面与早期工作（Izza 等人，2023）的表现差异。我们选择 CPLEX MILP求解器来检查一阶语句集合的不可满足性，并求解式（18）—（23）中的优化问题。我们使用 DOcplex Python建模API 对式（18）—（23）中的优化问题进行建模。我们使用参数值 $p \in \{0.25, 0.50, 0.75\}$ 评估 Twostep 并计算结果。

评估。由于我们使用每个数据集的 25% 进行训练，其余 75% 用作生成解释的目标。为了比较各方法，我们评估了以下指标：

- **计算时间**：计算解释的平均耗时。

- **解释范围宽度**：解释的平均范围宽度，其中解释的范围宽度计算如下：

$$\frac{\sum_{l_i' \le f_i \le u_i' \in E_{xp}} (u_i' - l_i')}{|E_{xp}|} \tag{24}$$

- **解释覆盖率**：为了评估生成的解释的泛化能力，我们分析了两种覆盖率：

1. **数据集覆盖率** – 每个解释所覆盖的原始数据集实例的比例，即特征值落在计算出的解释范围内的实例。我们计算所有生成的解释的平均覆盖率，以衡量每种方法在整个数据集上的泛化能力。我们还进行了逐实例比较，计算 Twostep 相较于 Onestep 在每个测试实例上的覆盖率改进百分比。

2. **合成数据覆盖率**– 为了进一步评估 Onestep 和 Twostep 方法的性能，我们还关注解释所覆盖的人工生成实例的比例。这第二项指标尤为重要，因为真实数据集可能没有足够的实例来恰当地评估每种方法的泛化能力。与数据集覆盖率分析类似，我们按实例比较 Twostep 相较于 Onestep 的覆盖率改进百分比。对于每个原始实例$\mathbf{x} = \{f_1 = v_1, f_2 = v_2, \ldots, f_n = v_n\}$，我们分别使用 Onestep 和 Twostep 生成一个解释。然后，通过在受控取值范围内扰动每个特征值，生成 100 个人工实例。具体而言，每个人工实例的形式为$\mathbf{x}' = \{f_1 = v_1', f_2 = v_2', ..., f_n = v_n'\}$，其中 $v_i' \in [v_i - d, v_i + d]$，而$d$定义扰动幅度。dd 的取值至关重要。如果 d过大，尤其是在高维数据集上，两种方法都可能无法覆盖任何人工实例。反之，

如果 d 过小，且数据集的维度不多，那么在许多情况下，两种方法都可能覆盖所有人工实例，这可能无法清楚地区分它们的泛化性能。大多数数据集采用 $d = 0.1$ 。然而，对于维数较高的 BRCW 和 IONS，则采用较小的 $d = 0.001$ 。

需要指出的是，Onestep 和 Twostep 方法能够保证生成的解释具备正确性和极小性。因此，无需根据训练模型评估解释的保真度或准确率。

5.1. 实验结果

本小节介绍用于比较 Onestep 和 Twostep 方法的实验结果。支持向量机和 MLP 模型的计算时间结果分别见表2和表3。支持向量机和 MLP 的数据集覆盖率结果分别见表4和表5。支持向量机和 MLP分类器的数据集覆盖率逐实例比较分别见图4和图5。这些直方图显示了 Twostep 的覆盖率高于 Onestep 的案例数量。此外，我们还分析了不同百分比区间内覆盖率提升的分布，以更好地了解提升幅度。合成数据覆盖率结果以直方图形式展示在图6和图7中。解释范围宽度结果见表6和表7。

5.1.1. 计算时间结果

正如预期，Twostep 通常比 Onestep 慢。如表3所示，在 IONS 数据集中使用多层感知机作为分类器时，耗时最多增加约 155.72%（从 1.941 秒增至 4.949 秒）。这种速度变慢是因为 Twostep 需要额外的计算步骤来生成解释。因此，在计算资源有限的情况下，

Twostep 的性能可能会受到限制，尤其是在数据集较大或需要更快响应时。不过，在许多实际应用中，覆盖率提升带来的益处可能足以抵消额外的计算成本，尤其是在优先考虑解释质量而非计算效率的定义域中。

对于表2中给出的支持向量机结果，Twostep 相对于 Onestep 的执行时间增幅在所有数据集上都不那么明显。差异范围从 BLDT 数据集上的 13.29%（0.013 秒对 0.015 秒）到 BRCW 数据集上的 56.40%（0.086 秒对 0.135 秒）。这很可能是因为底层线性规划问题更为简单，从而使两种方法都能更快地计算解释。

表2 支持向量机的平均执行时间结果。#M 表示平均执行时间及其标准差。%M 表示与 Onestep 相比，Twostep 的平均执行时间百分比增幅。

数据集	Onestep	Twostep					
		T步 0.25		T步 0.50		T步 0.75	
	$\#M$	$\#M$	$\%M$	$\#M$	$\%M$	$\#M$	$\%M$
IRIS	0.017 ± 0.003	0.022 ± 0.004	29.41	0.022 ± 0.004	29.41	0.022 ± 0.004	29.41
BLDT	0.014 ± 0.004	0.015 ± 0.004	7.14	0.015 ± 0.003	7.14	0.015 ± 0.004	7.14
BANK	0.017 ± 0.004	0.023 ± 0.004	35.29	0.023 ± 0.004	35.29	0.023 ± 0.004	35.29
UKMO	0.018 ± 0.005	0.023 ± 0.006	27.78	0.023 ± 0.006	27.78	0.023 ± 0.006	27.78
VRTC	0.021 ± 0.005	0.028 ± 0.007	33.33	0.028 ± 0.007	33.33	0.028 ± 0.007	33.33
PIMA	0.026 ± 0.005	0.036 ± 0.006	38.46	0.036 ± 0.007	33.33	0.037 ± 0.007	42.31
GLAS	0.025 ± 0.004	0.034 ± 0.007	36.00	0.033 ± 0.006	32.00	0.033 ± 0.005	32.00
WINE	0.037 ± 0.005	0.053 ± 0.009	43.24	0.054 ± 0.010	42.11	0.052 ± 0.009	40.54
CLIM	0.050 ± 0.007	0.072 ± 0.011	44.00	0.074 ± 0.011	45.10	0.072 ± 0.011	44.00
PARK	0.052 ± 0.011	0.072 ± 0.019	38.46	0.074 ± 0.020	42.31	0.073 ± 0.020	40.38
BRCW	0.087 ± 0.008	0.136 ± 0.016	56.32	0.133 ± 0.014	54.65	0.134 ± 0.014	54.02
IONS	0.099 ± 0.012	0.149 ± 0.019	50.51	0.148 ± 0.019	54.17	0.148 ± 0.019	54.17

表3 MLP 的平均执行时间结果。#M 表示平均执行时间及其标准差。%M 表示 Twostep 相较于 Onestep 的平均执行时间百分比增幅。

数据集	Onestep	Twostep					
		T步 0.25		T步 0.50		T步 0.75	
	$\#M$	$\#M$	$\%M$	$\#M$	$\%M$	$\#M$	$\%M$
IRIS	0.16 ± 0.02	0.21 ± 0.04	28.83	0.20 ± 0.03	21.30	0.21 ± 0.04	26.51
BLDT	0.14 ± 0.01	0.16 ± 0.02	14.71	0.16 ± 0.02	13.87	0.16 ± 0.02	14.49
BANK	0.16 ± 0.02	0.20 ± 0.03	24.84	0.20 ± 0.03	26.58	0.20 ± 0.03	25.48
UKMO	0.18 ± 0.02	0.23 ± 0.04	28.25	0.23 ± 0.04	32.39	0.24 ± 0.05	33.33
VRTC	0.18 ± 0.02	0.24 ± 0.05	32.07	0.24 ± 0.04	26.98	0.24 ± 0.04	28.42
PIMA	0.26 ± 0.07	0.38 ± 0.17	43.56	0.39 ± 0.08	48.29	0.39 ± 0.07	47.33
GLAS	0.34 ± 0.07	0.52 ± 0.12	55.65	0.50 ± 0.10	54.15	0.51 ± 0.15	50.29
WINE	0.45 ± 0.06	0.71 ± 0.09	55.85	0.71 ± 0.06	62.90	0.71 ± 0.07	63.16
CLIM	0.53 ± 0.12	0.81 ± 0.18	51.69	0.84 ± 0.27	58.52	0.80 ± 0.13	58.58
PARK	0.80 ± 0.14	1.31 ± 0.28	62.64	1.39 ± 0.33	71.06	1.40 ± 0.35	74.62
BRCW	1.70 ± 0.34	3.02 ± 0.56	78.36	3.20 ± 0.62	93.52	3.42 ± 0.73	79.77
IONS	1.94 ± 0.75	4.02 ± 1.22	107.32	4.55 ± 1.64	141.01	4.95 ± 1.71	155.76

5.1.2. 数据集覆盖率结果

关于表4中支持向量机的数据集覆盖率结果，Twostep 在大多数数据集上展现出达到与 Onestep 持平或更高覆盖率的能力。然而，UKMO 数据集是一个例外，Twostep 的平均覆盖率最多低 1.38%。对于 BANK 数据集，Twostep 覆盖的实例最多比 Onestep 多 41.55%。此外，PIMA 数据集的结果表明，Twostep 覆盖的实例最多可比 Onestep 多 28.58%。然而，对于 BLDT、VRTC、CLIM、PARK、BRCW 和 IONS 数据集，Twostep 相对于 Onestep 在覆盖率上仅取得极小改进或没有改进。

同样，表5中的多层感知机数据集覆盖率结果表明，Twostep 的覆盖率不低于 Onestep。同样，对于 PIMA 和 BANK 数据集，Twostep 的数据集覆盖率显著提升，覆盖的实例分别增加了 19.95% 和 34.59%。尤其是，对于

表4 支持向量机的平均覆盖率结果。$\#M$ 表示平均数据集覆盖率及其标准差。$\%M$ 表示 Twostep 相对于 Onestep 的平均数据集覆盖率改进百分比。

数据集	Onestep	Twostep					
		T步 0.25		T步 0.50		T步 0.75	
	$\#M$	$\#M$	$\%M$	$\#M$	$\%M$	$\#M$	$\%M$
IRIS	30.41 ± 19.51	35.04 ± 15.35	15.23	35.56 ± 14.62	16.94	34.44 ± 15.87	13.25
BLDT	519.92 ± 94.91	520.85 ± 90.60	0.18	521.02 ± 89.80	0.21	520.89 ± 90.41	0.19
BANK	36.03 ± 36.37	49.60 ± 36.08	37.66	51.00 ± 36.77	41.55	46.57 ± 36.92	29.25
UKMO	38.29 ± 28.95	37.81 ± 27.69	-1.25	37.76 ± 27.57	-1.38	38.10 ± 28.12	-0.50
VRTC	27.51 ± 22.43	27.89 ± 22.51	1.38	27.83 ± 22.71	1.16	27.75 ± 22.68	0.87
PIMA	12.07 ± 13.60	15.52 ± 15.19	28.58	14.35 ± 14.39	18.89	13.02 ± 13.93	7.87
GLAS	24.66 ± 22.73	26.19 ± 23.25	6.20	26.45 ± 23.36	7.26	25.98 ± 23.23	5.35
WINE	1.19 ± 0.59	1.22 ± 0.64	2.52	1.23 ± 0.66	3.36	1.22 ± 0.62	2.52
CLIM	1.01 ± 0.10	1.01 ± 0.11	0.00	1.01 ± 0.11	0.00	1.01 ± 0.10	0.00
PARK	2.20 ± 2.54	2.21 ± 2.47	0.45	2.22 ± 2.47	0.91	2.20 ± 2.50	0.00
BRCW	1.04 ± 0.28	1.06 ± 0.40	1.92	1.06 ± 0.40	1.92	1.06 ± 0.38	1.92
IONS	1.02 ± 0.15	1.02 ± 0.15	0.00	1.02 ± 0.15	0.00	1.02 ± 0.1	0.00

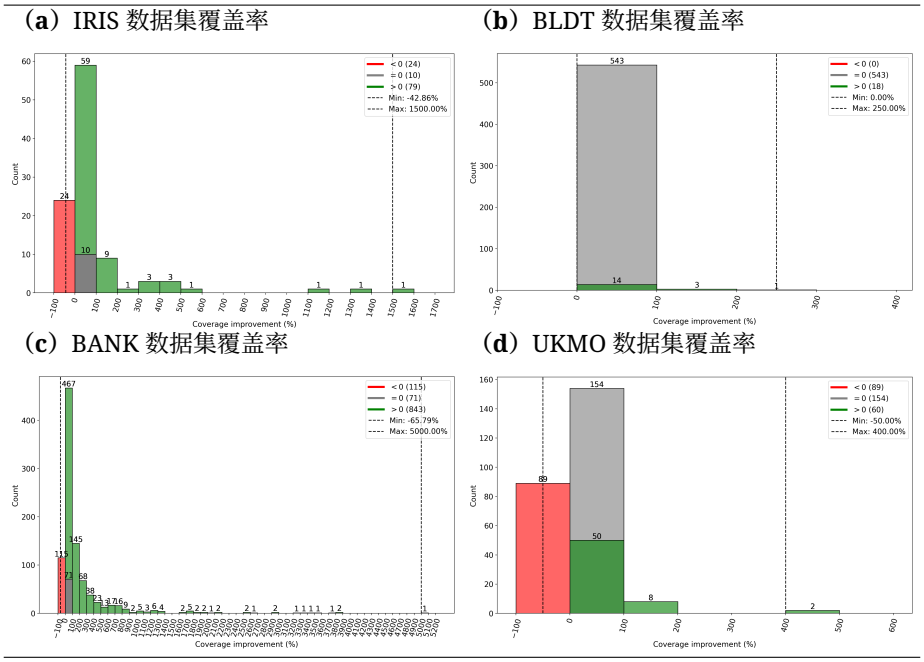
表5 多层感知机的平均覆盖率结果。$\#M$ 表示平均数据集覆盖率及其标准差。$\%M$ 表示Twostep相较于Onestep的平均数据集覆盖率改进百分比。

数据集	Onestep	Twostep					
		T步 0.25		T步 0.50		T步 0.75	
	$\#M$	$\#M$	$\%M$	$\#M$	$\%M$	$\#M$	$\%M$
IRIS	19.57 ± 14.48	21.81 ± 12.21	11.45	22.76 ± 11.81	16.30	22.65 ± 12.95	15.74
BLDT	149.55 ± 113.63	157.78 ± 122.77	5.50	162.46 ± 127.68	8.63	157.19 ± 122.89	5.11
BANK	25.15 ± 29.50	33.34 ± 31.56	32.56	33.85 ± 32.21	34.59	31.05 ± 31.59	23.46
UKMO	12.85 ± 10.93	13.91 ± 10.70	8.25	14.46 ± 10.99	12.53	14.14 ± 10.88	10.04
VRTC	12.94 ± 13.24	14.34 ± 13.61	10.82	14.41 ± 13.73	11.36	13.74 ± 13.57	6.18
PIMA	7.42 ± 8.65	8.90 ± 9.41	19.95	8.77 ± 9.63	18.19	8.10 ± 9.20	9.16
GLAS	1.46 ± 1.43	2.52 ± 2.34	72.60	2.27 ± 2.09	55.48	1.75 ± 1.69	19.86
WINE	1.00 ± 0.00	1.00 ± 0.00	0.00	1.01 ± 0.09	1.00	1.00 ± 0.00	0.00
CLIM	1.05 ± 0.22	1.05 ± 0.23	0.00	1.05 ± 0.23	0.00	1.05 ± 0.23	0.00
PARK	1.03 ± 0.22	1.03 ± 0.22	0.00	1.03 ± 0.20	0.00	1.03 ± 0.20	0.00
BRCW	1.06 ± 0.32	1.07 ± 0.34	0.94	1.07 ± 0.33	0.94	1.07 ± 0.33	0.94
IONS	1.01 ± 0.09	1.01 ± 0.09	0.00	1.01 ± 0.09	0.00	1.01 ± 0.09	0.00

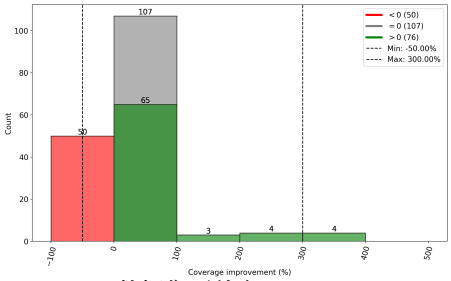
在 GLAS 数据集上，Twostep 比 Onestep 多覆盖了令人瞩目的 73.71% 的实例，展现出其提供更一般性解释的潜力。然而，对于 WINE、CLIM、PARK、BRCW 和 IONS 数据集，Twostep 的提升幅度较小，甚至毫无提升，这表明为这些数据集提供更广泛的覆盖更具挑战性。

现在，我们从逐个解释的角度分析覆盖率结果。具体而言，对于每个测试实例，两种方法都会生成一个解释，我们比较 Twostep 生成的解释相较于 Onestep 生成的解释在数据集覆盖率方面的提升。在这项分析中，我们将 Twostep 参数固定为 $p = 0.25$。

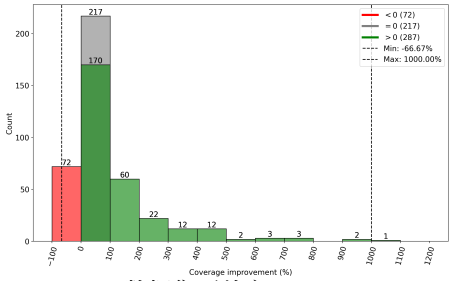
图4 支持向量机中 Twostep 相较于 Onestep 实现的数据集覆盖率提升（%）分布，其中 Twostep 参数固定为 $p = 0.25$。红色条形表示覆盖率变差的情况，灰色条形表示覆盖率不变的情况，绿色条形表示覆盖率提升的情况。最佳提升和最严重的下降已突出显示。



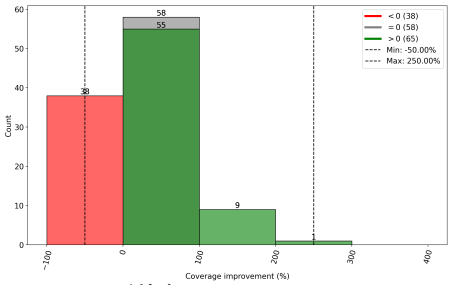
(e) VRTC 数据集覆盖率



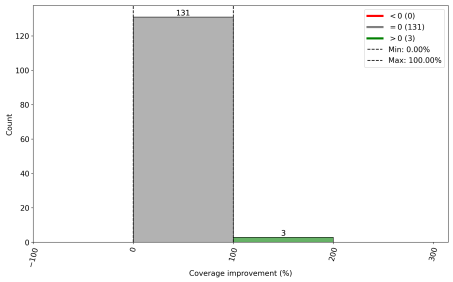
(f) PIMA 数据集覆盖率



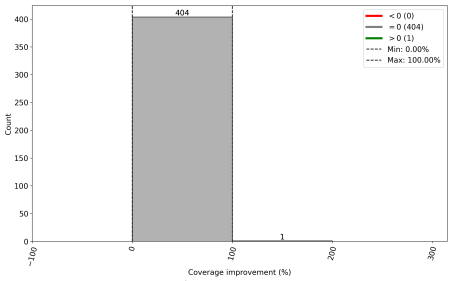
(g) GLAS 数据集覆盖率



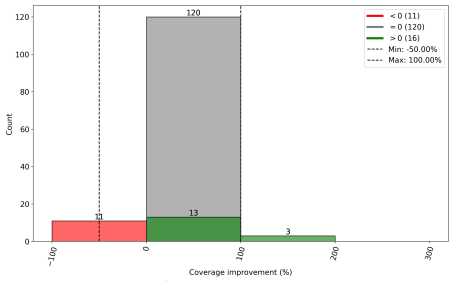
(h) WINE 数据集覆盖率



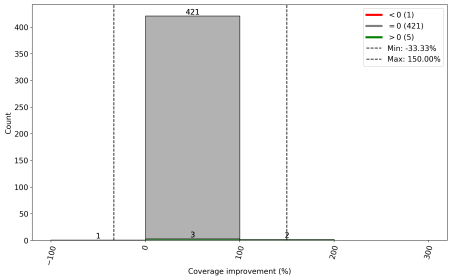
(i) CLIM 覆盖率



(j) PARK 覆盖率



(k) BRCW 覆盖率



(l) IONS 覆盖率

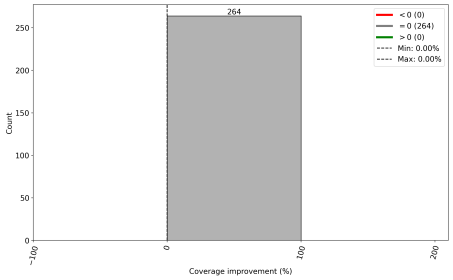
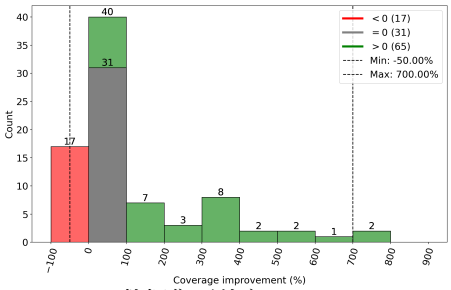


图4中支持向量机的结果突显了 Twostep 相较于 Onestep 所取得的显著提升。对于 BANK 数据集，Twostep 的覆盖率——

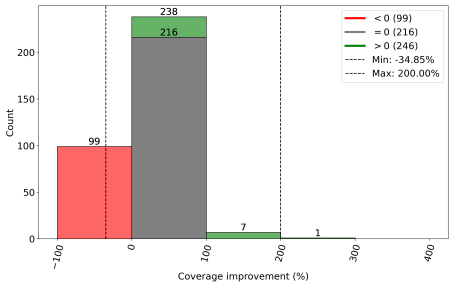
从减少 65.79% 到惊人的增加 5000% 不等。类似地，对于 IRIS 数据集，Twostep 也取得了显著提升，覆盖的实例数量从减少 42.86% 到增加 1500% 不等。在 UKMO 数据集中，与 Onestep 相比，Twostep 解释的覆盖率从减少 50% 到增加 400% 不等。尽管在该数据集中，Twostep 解释在 89 个案例中覆盖的实例更少，但在 60 个案例中覆盖的实例更多，且在 154 个案例中，两种方法取得了相同的覆盖率。此外，对于 BLDT、WINE 和 CLIM 数据集，Twostep 始终取得了等于或优于 Onestep 的覆盖率，没有出现覆盖率损失的案例。IONS 数据集是一个例外，在该数据集中，Twostep 和 Onestep 在每个解释中覆盖的实例数量都相同。

图 5 在 Twostep 参数固定为 $p = 0.25$ 时，Twostep 相较于 Onestep 在多层感知机上的数据集覆盖率提升（%）分布。红色柱表示覆盖率下降的情况，灰色柱表示覆盖率不变的情况，绿色柱表示覆盖率提升的情况。最佳提升和最严重的下降以高亮显示。

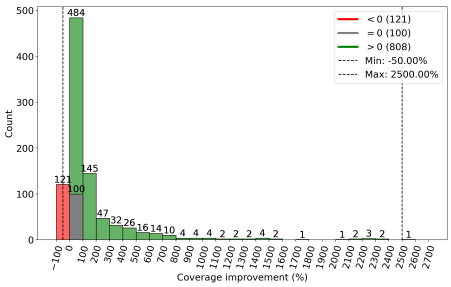
(a) IRIS 数据集覆盖率



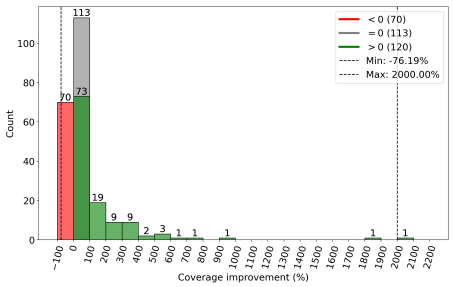
(b) BLDT 数据集覆盖率



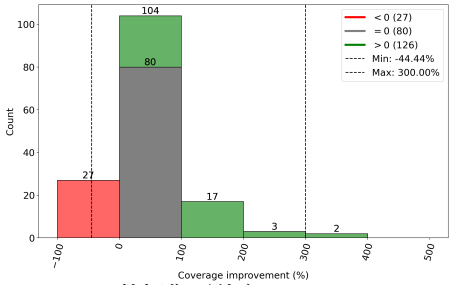
(c) BANK 数据集覆盖率



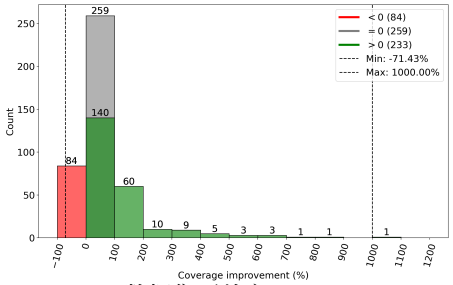
(d) UKMO 数据集覆盖率



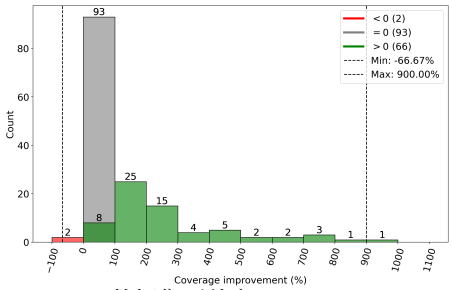
(e) VRTC 数据集覆盖率



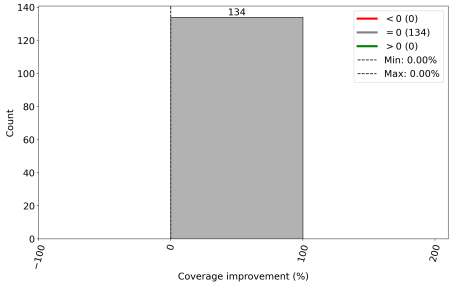
(f) PIMA 数据集覆盖率



(g) GLAS 数据集覆盖率



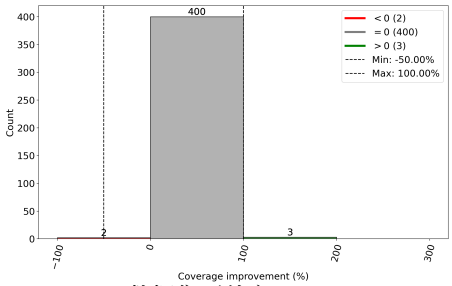
(h) WINE 数据集覆盖率



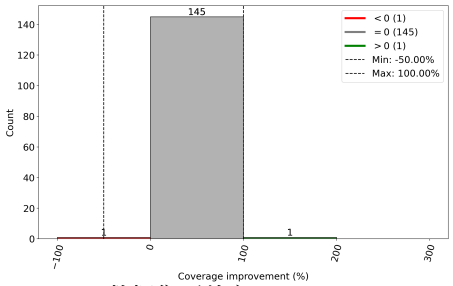
(i) CLIM 数据集覆盖率

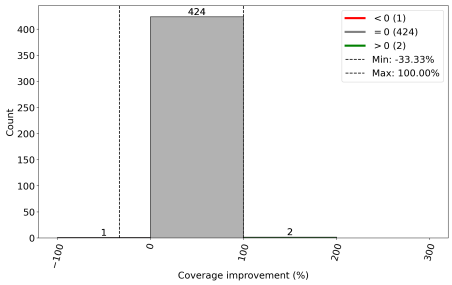
(j) PARK 数据集覆盖率

(k) BRCW 数据集覆盖率



(l) IONS 数据集覆盖率





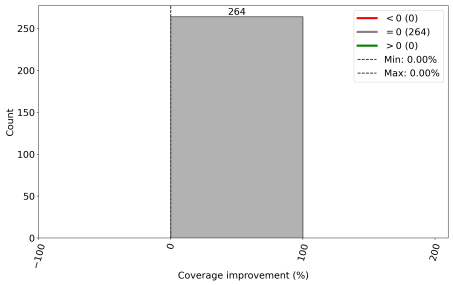


图 5 中多层感知机的结果进一步证明了 Twostep 生成更一般性解释的能力。在多个方面，Twostep 显著优于 Onestep——

eral 数据集，UKMO 数据集和 BANK 数据集的覆盖率提升分别高达 2000% 和 2500%，十分显著。同样，在 GLAS 和 BLDT 数据集中，Twostep 分别在 66 对 2 以及 246 对 99 个案例中提供了更具一般性的解释。尽管 Twostep 在 CLIM、PARK 和 BRCW 数据集上的提升较为温和，但其整体表现仍然更好。Twostep 并未为 WINE 和 IONS 数据集提供额外的覆盖率。

5.1.3. 合成数据覆盖率结果

在真实数据集上评估覆盖率可能具有挑战性，因为它们仅包含少量实例，这可能会限制对泛化的充分评估。为解决这一问题，我们使用人工生成的实例分析覆盖率，从而能够对 Twostep 与 Onestep 进行更全面的比较。

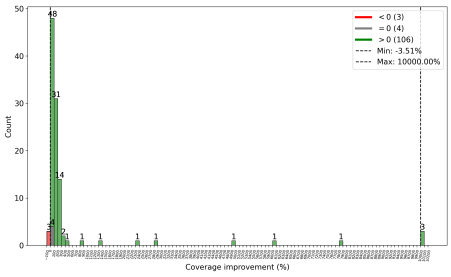
图 6 和图 7 分别展示了支持向量机和多层感知机的合成数据覆盖率结果。在所有数据集中，对于支持向量机和多层感知机，Twostep 在覆盖率提升的案例数量上都高于覆盖率下降的案例数量。此外，在一半的数据集中，对于支持向量机和多层感知机，Twostep 覆盖率提升的案例数量既多于覆盖率下降的案例，也多于覆盖率保持不变的案例。

对于特征数量较多的数据集 BRCW 和 IONS，与 MLP 的数据集覆盖率结果相比，Twostep 在合成数据评估中实现覆盖率提升的案例数量显著更多。具体而言，在数据集覆盖率分析中，IONS 的覆盖率没有提升，BRCW 也只有两个案例有所提升；而合成数据覆盖率结果显示，IONS 有 22 个案例有所提升，BRCW 则有 136 个案例。值得注意的是，在某些实例中，Twostep 的覆盖率提升幅度高达 10,000%。这些发现凸显了使用合成数据评估解释方法泛化特性的重要性。

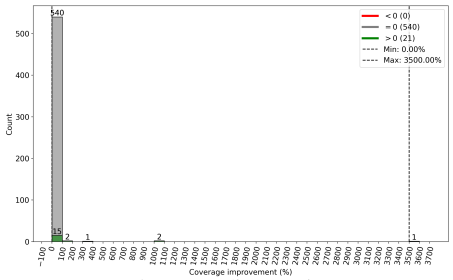
图 6 Twostep 相较于 Onestep 在

合成数据上的覆盖率提升（%）直方图，其中支持向量机模型和 Twostep 参数固定为 $p = 0.25$。红色柱表示覆盖率变差的情况，灰色柱表示覆盖率不变的情况，绿色柱表示覆盖率提升的情况。最佳提升和最严重的下降以高亮显示。

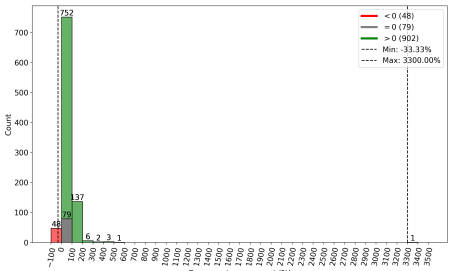
(a) IRIS 合成数据集覆盖率



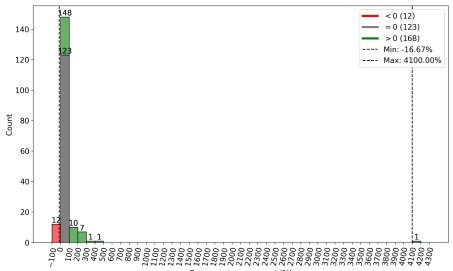
(b) BLDT 合成数据集覆盖率



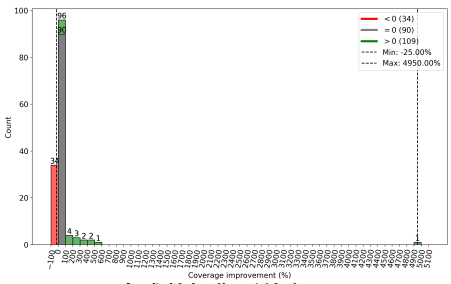
(c) BANK 合成数据集覆盖率



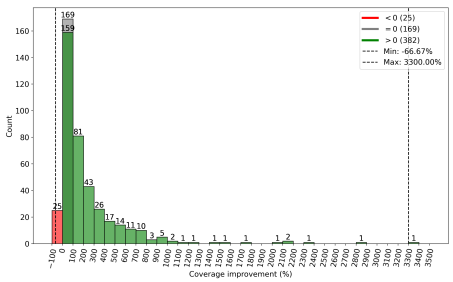
(d) UKMO 合成数据集覆盖率



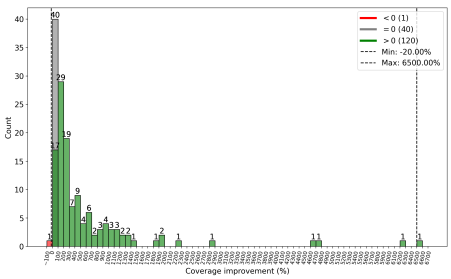
(e) VRTC 合成数据集覆盖率



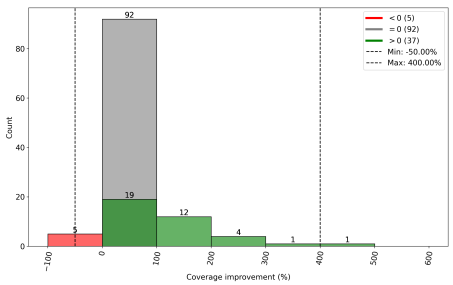
(f) PIMA 合成数据集覆盖率



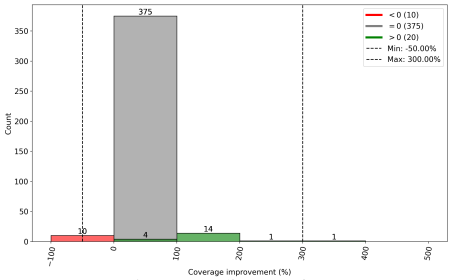
(g) GLAS 合成数据集覆盖率



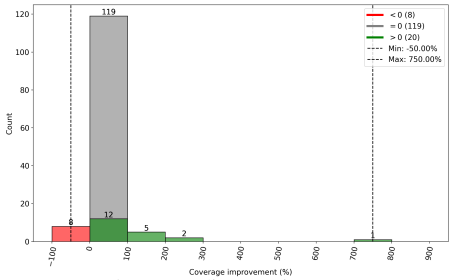
(h) WINE 合成数据集覆盖率



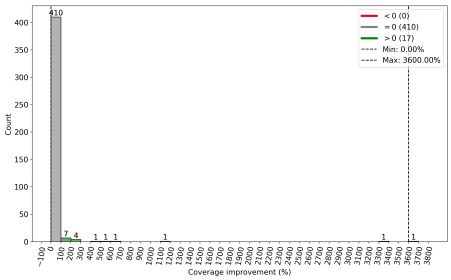
(i) CLIM 合成数据集覆盖率



(j) PARK 合成数据集覆盖率



(k) BRCW 合成数据集覆盖率



(l) IONS 合成数据集覆盖率

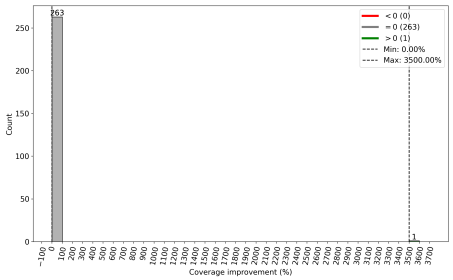


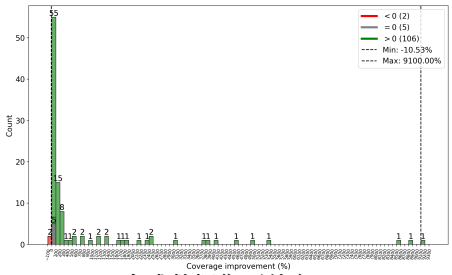
图7 Twostep 相较于

Onestep 在合成数据上的覆盖率提升（%）直方图，采用多层感知机模型，且 Twostep 参数固定为 $p = 0.25$。红色条形表示覆盖率下降的情况，灰色条形表-

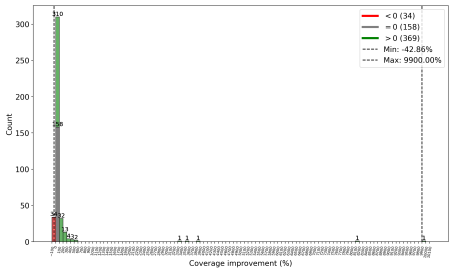
示覆盖率不变的情况，绿色条形表示覆盖率提升的情况

。图中突出显示了最大提升和最严重的下降。

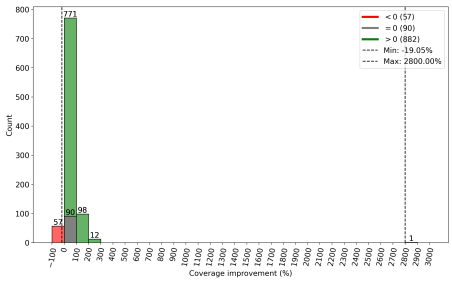
(a) IRIS 合成数据集覆盖率



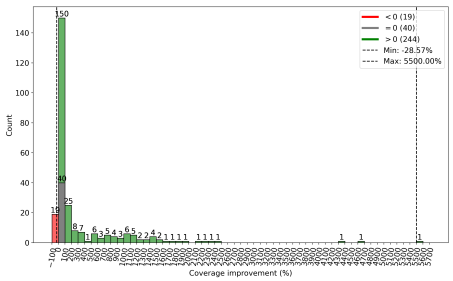
(b) BLDT 合成数据集覆盖率



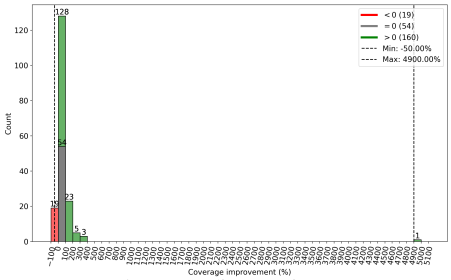
(c) BANK 合成数据集覆盖率



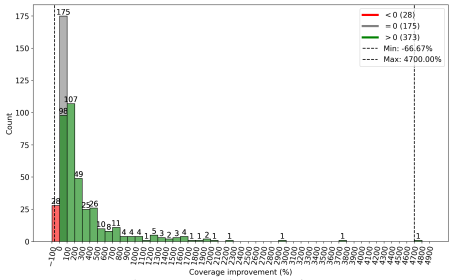
(d) UKMO 合成数据集覆盖率



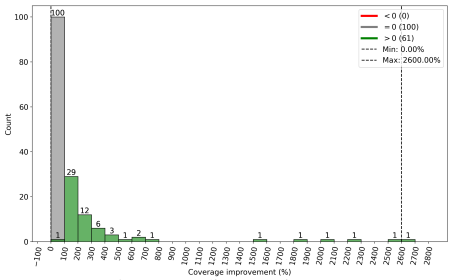
(e) VRTC 合成数据集覆盖率



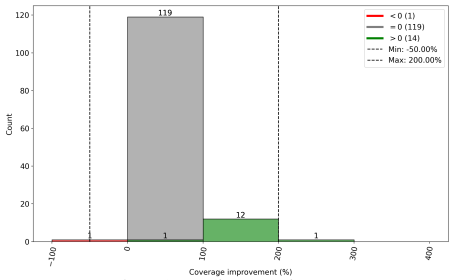
(f) PIMA 合成数据集覆盖率



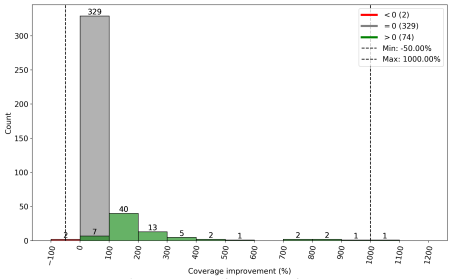
(g) GLAS 合成数据集覆盖率



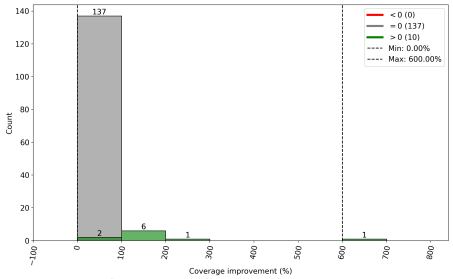
(h) WINE 合成数据集覆盖率



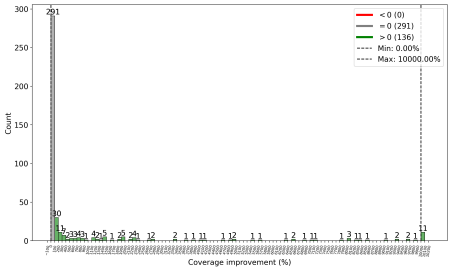
(i) CLIM 合成数据集覆盖率



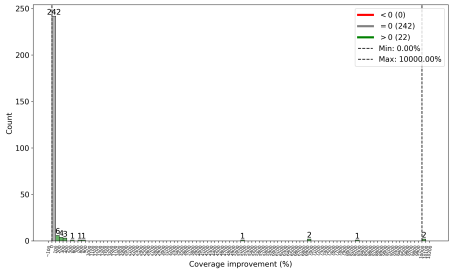
(j) PARK 合成数据集覆盖率



(k) BRCW合成数据集覆盖率



(l) IONS合成数据集覆盖率



5.1.4. 解释范围宽度的结果

最后，表6和表7展示了与解释范围宽度相关的结果。结果表明，在大多数情况下，Twostep 都能够增大范围宽度。对于大多数数据集，包括特征较多的数据集，MTwostep 的结果与 Onestep 相近或略好。不过，使用支持向量机的 Twostep 在 IRIS 和 UKMO 数据集上的表现较差，在 VRTC 和 BLDT 数据集上的结果则与 Onestep 相近。

表6 支持向量机的解释范围宽度结果。#M 表示平均范围宽度及其标准差。%M 表示 Twostep 相较于 Onestep 的平均范围宽度改进百分比。

数据集	Onestep	Twostep					
		T步 0.25		T步 0.50		T步 0.75	
	$\#M$	$\#M$	$\%M$	$\#M$	$\%M$	$\#M$	$\%M$
IRIS	2.55 ± 0.43	2.52 ± 0.44	-1.18	2.52 ± 0.45	-1.18	2.53 ± 0.44	-0.78
BLDT	3.29 ± 0.11	3.29 ± 0.11	0.00	3.29 ± 0.11	0.00	3.29 ± 0.11	0.00
BANK	2.47 ± 0.13	2.48 ± 0.13	0.40	2.48 ± 0.13	0.40	2.48 ± 0.13	0.40
UKMO	3.55 ± 0.63	3.53 ± 0.62	-0.56	3.53 ± 0.62	-0.56	3.54 ± 0.62	-0.28
VRTC	4.45 ± 1.21	4.45 ± 1.21	0.00	4.45 ± 1.21	0.00	4.45 ± 1.21	0.00
PIMA	4.75 ± 0.80	4.79 ± 0.80	0.84	4.78 ± 0.80	0.63	4.76 ± 0.80	0.21
GLAS	6.17 ± 0.60	6.18 ± 0.59	0.16	6.18 ± 0.60	0.16	6.17 ± 0.60	0.00
WINE	8.33 ± 0.76	8.35 ± 0.77	0.24	8.35 ± 0.76	0.24	8.34 ± 0.76	0.12
CLIM	11.75 ± 1.16	11.77 ± 1.16	0.17	11.76 ± 1.16	0.09	11.76 ± 1.16	0.09
PARK	16.07 ± 3.00	16.09 ± 3.00	0.12	16.09 ± 3.00	0.12	16.08 ± 3.00	0.06
BRCW	14.86 ± 4.07	14.88 ± 4.07	0.13	14.87 ± 4.07	0.07	14.87 ± 4.07	0.07
IONS	17.52 ± 2.08	17.54 ± 2.08	0.11	17.54 ± 2.08	0.11	17.53 ± 2.08	0.06

IRIS、UKMO 和 VRTC 的情况尤为有趣，因为表4显示，Twostep 和支持向量机使 IRIS 数据集的平均覆盖率大幅提高。此外，如图6所示，对于这三个数据集，Twostep 与支持向量机结合，在合成数据上的覆盖率也有显著提升。类似地，Twostep 与多层感知机结合时，UKMO 和 VRTC 数据集的解释范围宽度变差或没有变化。然而，对于这些数据集，Twostep 在合成数据上的覆盖率取得了显著提升

（见图7），平均覆盖率也明显提高（见表5）。这些结果进一步支持了这样一种观点：更有策略地扩展取值范围，即使会缩小解释范围宽度，也能提高解释的泛化能力。

表7 多层感知机的解释范围宽度结果。#M 表示平均解释范围宽度及标准差。%M 是Twostep相对于Onestep在平均范围宽度上的改进百分比。

数据集	Onestep	Twostep					
		T步 0.25		T步 0.50		T步 0.75	
	$\#M$	$\#M$	$\%M$	$\#M$	$\%M$	$\#M$	$\%M$
IRIS	2.09 ± 0.69	2.12 ± 0.65	1.44	2.14 ± 0.63	2.39	2.13 ± 0.65	1.91
BLDT	2.71 ± 0.27	2.72 ± 0.27	0.37	2.73 ± 0.27	0.74	2.72 ± 0.27	0.37
BANK	2.31 ± 0.29	2.33 ± 0.29	0.87	2.33 ± 0.29	0.87	2.32 ± 0.29	0.43
UKMO	3.27 ± 0.65	3.23 ± 0.64	-1.22	3.23 ± 0.63	-1.22	3.25 ± 0.62	-0.61
VRTC	3.87 ± 1.37	3.87 ± 1.36	0.00	3.87 ± 1.36	0.00	3.87 ± 1.37	0.00
PIMA	4.40 ± 0.60	4.43 ± 0.59	0.68	4.44 ± 0.59	0.91	4.43 ± 0.59	0.68
GLAS	3.68 ± 0.78	3.72 ± 0.78	1.09	3.70 ± 0.80	0.54	3.68 ± 0.80	0.00
WINE	5.87 ± 1.13	5.98 ± 0.92	1.87	6.03 ± 0.79	2.73	6.06 ± 0.73	3.24
CLIM	12.28 ± 1.17	12.31 ± 1.11	0.24	12.32 ± 1.08	0.33	12.34 ± 1.02	0.49
PARK	13.70 ± 3.20	13.78 ± 3.17	0.58	13.82 ± 3.10	0.88	13.87 ± 3.06	1.24
BRCW	16.67 ± 2.76	16.81 ± 2.44	0.84	16.96 ± 2.17	1.74	17.07 ± 2.00	2.40
IONS	14.36 ± 3.43	14.93 ± 2.82	3.97	15.17 ± 2.51	5.64	15.43 ± 2.26	7.45

5.2. 简要案例研究

我们将根据 Onestep 和 Twostep 生成的解释，介绍一个简要案例研究。设 IRIS 数据集中的一个实例为 $\mathbf{x} = \{$萼片长度$= 4.6$、萼片宽度 $= 3.4$、花瓣长度 $= 1.4$、花瓣宽度 $= 0.3\}$，由支持向量分类器预测为类别 山鸢尾。Onestep 获得的、能够保证

预测的解释为:

如果 $2.51 \leq$ 萼片宽度 ≤ 4.40 **且** $1.00 \leq$ 花瓣长度 ≤ 1.40**且** $0.10 \leq$ 花瓣宽度 ≤ 0.30**那么** 山鸢尾。

相比之下，Twostep 得到的解释为:

如果 $2.84 \leq$ 萼片宽度 ≤ 4.40**且** $1.00 \leq$ 花瓣长度 ≤ 1.73 **且** $0.10 \leq$ 花瓣宽度 ≤ 0.38**那么** 山鸢尾。

在此情况下，Onestep 和 Twostep 得到的解释都省略了特征 萼片长度 , 因为该特征在相应定义域内可以取任意值，而不会改变分类结果。与 Onestep 得到的解释相比，Twostep 得到的解释对 萼片宽度的取值区间限制更窄，因此 花瓣长度 和 花瓣宽度的取值区间都更宽。结果，Twostep 得到的解释覆盖了 IRIS 数据集中的 13 个实例，而 Onestep 得到的解释覆盖了 8 个实例。因此，Twostep 比 Onestep 多覆盖了 62.5% 的实例。

此外，设 VRTC 数据集中的一个实例为 $\mathbf{x} = \{$骨盆入射角$= 63.03$，骨盆倾斜 $= 22.55$，腰椎前凸角 $= 39.61$，骶骨倾斜角 $=$40。48，骨盆半径 $= 98.67$，脊椎滑脱程度 $= -0.25\}$ ，由支持向量分类器预测为异常类别。Onestep 对该实例得出的解释

预测为

如果27.43 ≤ 腰椎前凸角 ≤ 125.74**且**
13.37 ≤ 骶骨倾斜角 ≤ 42.38**且**70.08 ≤ 骨盆半径
≤ 98.67**则**异常。

相比之下，Twostep 给出的解释为

如果 31.24 ≤ 腰椎前凸角 ≤ 125.74**且**
13.37 ≤ 骶骨倾斜角 ≤ 51.47**且** 70.08 ≤ 骨盆半径
≤ 98.79 **则** 异常。

对于此案例，Onestep 和 Twostep 得到的解释省略了特征 骨盆入射角、骨盆倾斜和脊椎滑脱程度，因为这些特征在各自的定义域内可以取任意值，而不会改变预测。Twostep 方法缩小了腰椎前凸角 的取值区间，以便扩大骶骨倾斜角 和骨盆半径的取值区间。这一调整使 Twostep 得到的解释覆盖了 VRTC 数据集中的 6 个实例，而 Onestep 得到的解释覆盖了 4 个实例，覆盖的实例数增加了 50%。

这些案例更清楚地展现了 Twostep 与 Onestep 的不同：在可能的情况下，Twostep 往往会缩小某些特征的取值范围，同时扩大其他特征的取值范围。这对 Twostep 也很重要，因为它凸显了 Twostep 相较于 Onestep 实现更高实例覆盖率的潜力。

5.3. 更多解释示例

在上一节中，我们展示了 IRIS 和 VRTC 数据集中的实例解释。在本小节中，我们将展示 Twostep 和 Onestep 为 PIMA 和 WINE 数据集生成的解释示例。

这些数据集用于可视化特征更多的数据集的解释。

对于 PIMA 数据集，原始实例具有以下取值：

原始实例：

妊娠次数 $= 7.00$，葡萄糖 $= 194.00$，血压 $= 68.00$，皮肤厚度 $= 28.00$，胰岛素 $= 0.00$，体质指数 $= 35.90$，糖尿病遗传函数 $= 0.745$，年龄 $= 41.00$

两种方法生成的解释如下：

单步解释：

IF $188.96 \leq$ 葡萄糖 ≤ 199.00

AND $35.66 \leq$ 体质指数 ≤ 67.10

THEN 阳性。

两步解释：

IF $190.56 \leq$ 葡萄糖 ≤ 199.00

AND $34.45 \leq$ 体质指数 ≤ 67.10

THEN 阳性。

对于 WINE 数据集，原始实例具有以下取值：

原始实例：

酒精 $= 13.88$，苹果酸 $= 5.04$，灰分 $= 2.23$，灰分碱度 $= 20.00$，镁 $= 80.00$，总酚 $= 0.98$，黄酮类化合物 $= 0.34$，非黄烷类酚 $= 0.40$，原花青素 $= 0.68$，色泽强度 $= 4.90$，色调 $= 0.58$，od280/od315 $= 1.33$，脯氨酸 $= 415.00$

两种方法生成的解释如下：

	单步解释：		**两步解释：**
IF	$0.98 \leq$ 总酚 ≤ 1.95	**IF**	$0.98 \leq$ 总酚 ≤ 1.57
AND	$0.34 \leq$ 黄酮类化合物 ≤ 0.34	**AND**	$0.34 \leq$ 黄酮类化合物 ≤ 0.48
AND	$1.27 \leq$ od280/od315 ≤ 1.33	**AND**	$1.27 \leq$od280/od315 ≤ 1.37
AND	$278.00 \leq$ 脯氨酸 ≤ 415.00	**AND**	$278.00 \leq$ 脯氨酸 ≤ 423.01
THEN	阳性。	**THEN**	阳性。

6. 结论

在这项工作中，我们提出了两种方法：Onestep 和 Twostep，旨在通过优化提高基于逻辑的解释的泛化能力。两种方法都能保证正确性和极小性，从而确保解释可靠且简洁。Onestep 可作为基准，它改进了早期工作中提出的方法（Izza等人，2023），通过消除增量过程提高效率，并得出更全面的解释。Twostep 则在此基础上引入了更灵活的扩展过程，旨在解决特征范围仍然过窄的情况，从而进一步提升泛化能力。

我们在 12 个不同的数据集上比较了 Onestep 和 Twostep。Onestep 在先前工作的基础上进行了改进（Izza等人，2023），可作为评估 Twostep 所带来改进的参照。通过将 Twostep 与 Onestep 进行比较，我们间接评估了其相对于以往方法的性能。两种方法均应用于以下分类器：多层感知机和线性支持向量分类器。结果表明，Twostep 采取了更具策略性的扩展方式，尽管速度略慢于 Onestep，但实现了更高的覆盖率，同时保持了相近甚至更大的解释范围宽度。

Onestep 和 Twostep 都存在明显的局限性。一个值得注意的问题是，随着数据维度的增加，覆盖率会下降。尽管我们的方法实现了较好的泛化程度，但对于更复杂的

数据集。此外，由于这两种方法都建立在最小解释之上，它们可能会遗漏特征更少的解释，而我们无法保证找到这些解释。这些更小的解释可能提供更高的覆盖率，尤其是考虑到我们仅通过数据集中存在的实例来评估这些解释。

未来的工作可以从多个方向着手，以改进这两种方法。一种可能的改进是预先分析实例分布，从而更深入地了解特征在实例空间中的组织方式。这种分析有助于通过优先考虑覆盖率更广的特征范围来提升泛化能力。另一个有前景的方向是将范围扩展任务表述为成本优化问题，其中成本函数用于评估每个特征的取值范围的质量。这样，求解器便能够系统地优化上下界，从而有望得到更高效的范围分布。此外，我们的技术还可以用于生成对比解释，通过强调为何作出了某项决策而非另一种选择，提供更清晰的见解。最终，这些改进有望带来更有启发性的解释，提升模型的可解释性并加深对模型的理解。

7. 致谢

作者感谢 FUNCAP、CNPq 和 CAPES 对我们的研究工作提供部分资助。

参考文献

Audemard、G.、Bellart、S.、Bounia、L.、Koriche、F.、Lagniez、J.-M. 和 Mar- quis、P.（2022年）。关于决策树和随机森林的优选溯因解释。载于第*31*届 *IJCAI*（第643–650页）。

Bishop, C. M.（1995）。用于模式识别的神经网络。牛津大学出版社。

Biswas, S. S.（2023）。ChatGPT在公共卫生中的作用。《生物医学工程年鉴》，*51*，868–869。Cortes, C., & Vapnik, V.（1995）。支持向量网络。机器学习，*20*，273–297。Dimanov, B., Bhatt, U., Jamnik, M., & Weller, A.（2020）。你不该信任我：会隐瞒不公平性的学习模型，及多种解释方法。IOS出版社。Fischetti, M., & Jo, J.（2018）。深度神经网络与混合整数线性优化。《约束》，*23*，296–309。Ignatiev, A.（2020）。迈向可信赖的可解释人工智能。载于*IJCAI*（第5154–5158页）。Ignatiev, A., Izza, Y., Stuckey, P. J., & Marques-Silva, J.（2022）。利用最大可满足性高效解释树集成。载于*AAAI*（第3776–3785页）。第36卷。Ignatiev, A., Narodytska, N., & Marques-Silva, J.（2019年a）。面向机器学习模型的基于溯因的解释。载于*AAAI*（第1511–1519页）。第33卷。Ignatiev, A., Narodytska, N., & Marques-Silva, J.（2019年b）。关于验证、修复和改进启发式机器学习解释。*CoRR*，*abs/1907.02509*。arXiv:1907.02509。Ignatiev, A., Narodytska, N., & Marques-Silva, J.（2020）。关于解释的形式化推理。载于*RCRA*。Izza, Y., Ignatiev, A., & Marques-Silva, J.（2020）。关于解释决策树。*arXiv*预印本 *arXiv:2010.11034*，。Izza, Y., Ignatiev, A., Stuckey, P., & Marques-Silva, J.（2023）。提供膨胀的解释。arXiv:2306.15272。

Kroening, D., & Strichman, O. (2016). 判定过程. Springer. LeCun, Y., Bengio, Y., & Hinton, G. (2015). 深度学习. *Nature*, *521*, 436–444. Lundberg, S. M., & Lee, S.-I. (2017). 解释模型预测的统一方法. 神经信息处理系统进展, *30*. Marques-Silva, J., Gerspacher, T., Cooper, M., Ignatiev, A., & Narodytska, N. (2020年). 以多项式时间和延迟解释朴素贝叶斯及其他线性分类器. *NeurIPS*, *33*, 20590–20600. Marques-Silva, J., & Ignatiev, A. (2022年). 通过形式化可解释人工智能实现可信人工智能 . 载于第 *36* 届 *AAAI* 人工智能会议论文集 （第12342–12350页）. Miller, G. A. (1956). 神奇的数字七，加减二：我们信息处理能力的若干限制. *Psychological Review*, *63*, 81. Nair, V., & Hinton, G. E. (2010). 修正线性单元改进受限玻尔兹曼机. 载于 第*27*届国际机器学习会议（*ICML-10*）论文集 （第807–814页）. Ramkumar, M., Babu, C. G., Kumar, K. V., Hepsiba, D., Manjunathan, A., & Kumar, R. S. (2021年). 使用 DWT 、 ICA 和多层感知机神经网络进行心电图心律失常分类 . *Journal of Physics: Conference Series*, *1831*, 012015. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “我为什么应该信任你？”解释任意分类器的预测. 载于 第*22*届*ACM SIGKDD* （第1135–1144页）. Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors：高精度模型无关解释. 载于 *AAAI*. 第32卷.

Rocha Filho, M., Alves Rocha, T., Pereira Fernandes Ribeiro, R., Rˆego da Rocha Neto, A. 等 (2024). 带拒绝选项的线性支持向量分类器的基于逻辑的解释. *arXiv*电子预印本, （页码 arXiv–2403）.

Samek, W., & M¨uller, K.-R. (2019). 迈向可解释人工智能。可解释人工智能：深度学习的解读、解释与可视化, （第 5–22 页）。Shih, A., Choi, A., & Darwiche, A. (2018). 解释贝叶斯网络分类器的一种符号方法。arXiv:1805.03364。

Yang, W., Si, Y., Wang, D., & Guo, B. (2018). 基于主成分分析网络和线性支持向量机的心律失常自动识别。《计算机生物学与医学》, *101*, 22–32。