arXiv:2302.05270v1 [cs.LG] 10 Feb 2023

树集成分类器的概念视图

Tom Hanika[1,2] 和 Johannes Hirth[1,2]

[1] 知识与数据工程组，卡塞尔大学，德国[2] 信息系统设计跨学科研究中心，卡塞尔大学，德国 tom.hanika@cs.uni-kassel.de, hirth@cs.uni-kassel.de

摘要 随机森林及相关的基于树的方法广泛应用于表格数据的监督学习。除了易于并行化之外，它们的分类性能也更为出色。然而，这种性能优势，尤其是可并行化的优势，却因可解释性的缺失而大打折扣。人们常借助统计方法来弥补这一不足，但这类方法提供局部解释、尤其是全局解释的能力有限。本文提出一种植根于格论的代数方法，用于解释树集成（全局解释）。具体而言，我们介绍了树集成分类器的两种新颖概念视图，并展示了它们在采用标准参数训练的随机森林上的解释能力。

关键词： 决策树、随机森林、集成分类、可解释性、形式概念分析、可解释人工智能

1 引言

决策树是最受欢迎的可解释机器学习模型之一。因此，它们常被用作其他透明度较低的机器学习模型的替代模型。然而，其缺点之一是，决策树的表现往往不如更现代的分类程序。此外，在没有进一步协助或数据准备的情况下，决策树无法自然地处理缺失数据。

一种广受欢迎的分类器类别——树集成，确实弥补了这些缺点。它们同时采用多种树结构（提升法），并可能利用数据的不同装袋法，例如随机森林或梯度提升树 [7, 14]。尽管这些方法能够实现很高的分类性能，但其可解释性不及决策树。这是因为信息分散在大量来自根节点各异的树的、彼此不可比较的并行分支中。人们已进行不少尝试，以解决这一可解释性问题。一种解释树集成的方法是将所有树合并为一棵决策树 [34]。尽管由此得到的树

作者按字母顺序排列。作者署名不表示优先顺序。

结构可以称为（更）“可解释”（因为它是一棵树），但往往会变得大到难以理解，也会失去处理缺失值的能力。在本研究中，我们提出一种新方法，将树集成转换为一种在设计上具有可解释性、同时能够通过并行处理缺失信息的数据结构。与树 [36]不同，我们采用“格”（即具有上确界和下确界的有序集）。在树中，从根节点到某个节点始终只有唯一一条路径，而格允许单个格中存在多条（并行）路径，并最终通向同一个结论元素。因此，这种方法能更好地处理缺失值。

然而，我们不应期待存在一种能够表示集成分类器且便于使用、规模适中的数据结构。因此，生成的格规模很大。幸运的是，格有一套严谨的数学理论，可以生成复杂结构的多个规模适中的投影，而这些投影合在一起包含完整信息。我们采用形式概念分析（FCA） [15, 37],，它从设计上就能够处理此类结构，并提供了丰富的方法框架，以实现规模适中、便于人类理解的表示。更重要的是，FCA 提供了大量工具，可用于解释这些表示并进一步对其进行后处理。

我们的工作并非首次尝试从树集成迈向格 [11],，但它从两个方面推动了这项研究：首先，所提出的格表示在表达能力方面前所未有，从而提升了树集成分类器的可解释性。其次，我们提供了一个形式化的解释框架，说明如何从概念（格）视角理解一个分类器。正如我们将展示的，这些视角能够提供细节程度各异的局部和全局解释。

在我们的实验研究中，我们通过一个来自 OpenML CC18 [5]分类基准数据集的真实示例，即car [10] （二分类版本[3]），展示了我们解释方法的有效性及其适用性。分析所用的随机森林是在现实的参数假设下训练的，也就是说，我们使用了 100 棵树，且没有限制每棵树的深度。

2 动机与形式化定义

形式概念分析（FCA）是一门研究概念及概念层次结构的数学理论，应用于数据科学和知识处理。本文旨在探究形式概念分析能否为基于树的分类器提供有用的视图，因此我们希望对相关概念和符号作出形式化介绍。

2.1 形式概念与概念视图

乍一看，形式概念分析（FCA）只关注一种基本数据类型——两个集合之间的二元关系（用 FCA 的术语来说，即一种 **形式背景** ），似乎限制颇多，

[3] https://www.openml.org/d/991

因为数据有许多不同的格式。但这种限制是有意为之的。它使客观的形式数据分析与主观解释得以更清晰地分离，也使数学理论拥有统一而清晰的结构。

形式概念分析以两步流程处理各种不同的数据格式。首先，将数据转换为标准形式——也就是形式背景——然后在第二步中对该背景进行概念分析。第一步称为**概念尺度化**，[4] 可视为一种解释行为，取决于分析者的主观判断；分析者必须揭示手头数据的含义。因此，这一过程既非明确无歧义，也非自动完成，而且通常无法完整呈现数据的复杂性。不过，多个这样的**概念视图** 合在一起，完全有可能反映出数据的全貌。

树分类器的分类数据通常是由 n 元组构成的列表，形式类似于关系数据库或所谓的数据表。在形式概念分析的术语中，这称为**多值背景**。在这种多值背景中，各行有不同的名称（由此构成一个键），各列也有不同的名称，这些名称称为**（多值）属性**。表格中的条目就是取值。分类时通常假设每个属性的取值都按线性有序排列。然后利用这种顺序来划分数据。这并不意味着取值必须是数值。低 $<$ 中等 $<$ 高以及真 $<$ 假，也都是这类**区间序数型** 数据的例子。对于数值数据，除了 $a < b$之外，还可以例如使用 $a^2 <$cos(b) 作为划分标准。不这样做，正是上述意义上的一种解释性决定。

为说明下文将使用的符号，图1展示了一个用于网球比赛 [27] 数据集的决策树小例子。

2.2 基于树的分类器

通常，决策树 $\mathcal{T}$ 是一棵正规二叉树，即一棵有根树，其中每个非终端节点[5] 恰有两个子节点。除根节点 r之外，每个节点 n 都由一个谓词 $\phi(n)$注释，该谓词划分 数据，使得每个数据对象要么满足 $\phi(n)$，要么满足其否定 $\neg\phi(n)$。注释方式使得内部节点的两个子节点互为否定：如果一个子节点的注释为 P，另一个则为 $\neg P$。根据这些定义，用于 $\mathcal{T}$的谓词集合 $\mathcal{P}(\mathcal{T}) := \{\phi(n) \mid n \in \mathcal{T} \setminus \{r\}\}$ 包含每个 $P \in \mathcal{P}(\mathcal{T})$的否定 $\neg P$。

每棵这样的树都具有一种自然的顺序，对于节点 m, n 我们有

$$m \leq n :\iff n \text{ 位于从 } m \text{ 到根节点的唯一路径上。}$$

图1的示例上半部分展示了一个包含十四个数据对象的数据表，编号为 0、.....、13。表中有四个属性：“天气状况”、“温度”、“湿度”和“有风”，最后一列“打球”包含分类结果

[4] 此处的“尺度化”一词指测量理论，并非指算法能否应用于大型数据集。[5] 终端节点也称为叶节点，非终端节点则称为内部节点

𝔻	天气状况	温度	湿度	有风	打球	
0	晴天	hot	high	假		no
1	晴天	hot	high	True		no
2	阴天	hot	high	假		yes
3	下雨	mild	high	假		yes
4	下雨	cool	正常	假		yes
5	下雨	cool	正常	True		no
6	阴天	cool	正常	True		yes
7	晴天	mild	high	假		no
8	晴天	cool	正常	假		yes
9	下雨	mild	正常	假		yes
10	晴天	mild	正常	真		yes
11	阴天	mild	high	True		yes
12	阴天	hot	正常	假		yes
13	下雨	mild	high	True		no

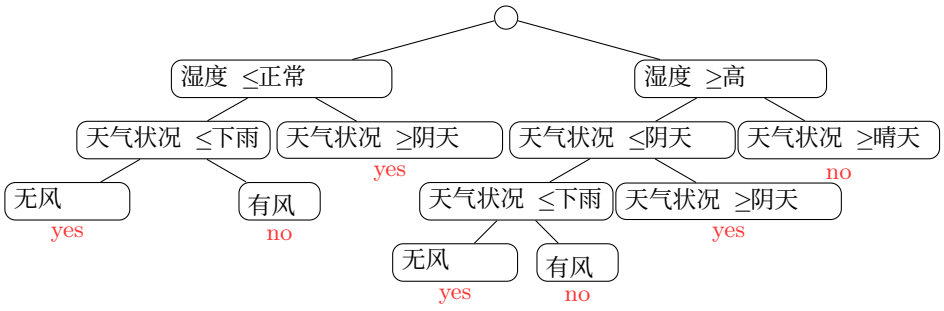


图1. 网球数据集的决策树。每个数据对象沿着从根节点（顶部节点）出发的路径，根据其满足的谓词前进，直到到达叶节点。以红色标注的“打球”决策。

结果。每个属性都采用其取值的序数解释，例如

$$\begin{array}{l}\text{下雨} < \text{阴天} < \text{晴天，凉} \\ \text{爽} < \text{温和} < \text{炎热，正常} \\ < \text{高，以及无风} < \text{有风。}\end{array}$$

此外，我们使用 $\geq$ 和 $\leq$ 关系，推导出这些取值的双序解释。进行分类时，数据对象 g 沿着树从根节点遍历到叶节点 b，并满足路径 r、……、b上的所有谓词。我们将 b称为分类/决策叶节点，将 r、……、b称为 g的决策路径。最后，还有第二种映射，它将分类结果（通常为“是”或“否”）关联到每个叶节点。

2.3 通过概念尺度化构建概念视图

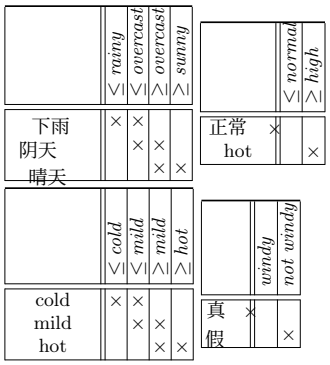
形式背景是一个三元组 $\mathbb{K} := (G, M, I)$，其中 G和 M是集合，$I \subseteq G \times M$是二元关系。如上所述，这是形式概念分析所使用的基本数据结构。G 中的元素称为形式背景的**对象**，M 中的元素称为其**属性** $\mathbb{K}$。人们将 $(g, m) \in I$ 读作“对象 g 具有属性 m”。

一个**多值背景** $\mathbb{D} := (G, M, W, I)$同样包含一个对象集合 G、一个**多值属性**集合 M，以及一个**属性值**集合 W。这里的关系 $I \subseteq G \times M \times W$是三元关系，其中$(g, m, w) \in I$表示对象 g 对于属性 m取值为 w。假设每个对象对于每个属性至多有一个取值，因此

$$(g, m, w_1) \in I, (g, m, w_2) \in I \implies w_1 = w_2.$$

此外，如果对于每个 $g \in G$，$m \in M$ 都存在一个 $w \in W$，使得$(g, m, w) \in I$，则称 $\mathbb{D}$ 完备。我们用 $m^{\mathbb{D}} := \{w \in W \mid \exists g \in G (g, m, w) \in I\}$ 表示所有满足以下条件的取值 $w \in W$ 的集合：存在一个对象 $g \in G$，它与属性 m 上的 w 相关联。从多值背景导出形式背景的过程称为**概念尺度化**。这个过程有多种变体，其中最简单的是**普通尺度化**，即只需指定如何理解每个多值属性的取值集合。为此，需要为每个多值属性指定一个形式背景（即**“尺度”**），使该尺度的对象集合包含此属性的取值。然后，将每个取值替换为该尺度中相应的一行。图2展示了网球数据集中的这一过程。此处的多值背景由图1中的数据表给出（忽略最后一列）。图2左半部分为四个多值属性中的每一个都给出了一个尺度。右侧的形式背景是通过普通尺度化从图1的数据表导出的，即将各个取值替换为相应尺度中的行。

使用形式概念分析的一个重要步骤是将形式背景展开为**概念格**。这一代数结构的元素称为**形式概念**。



$\mathbb{I}(\mathbb{D})$	$Overlook \leq$ *rainy*	$Overlook \leq$ *overcast*	$Overlook \geq$ *overcast*	$Overlook \geq$ *sunny*	$Temperature \leq$ *cool*	$Temperature \leq$ *mild*	$Temperature \geq$ *mild*	$Temperature \geq$ *hot*	$Humidity \leq$ *normal*	$Humidity \geq$ *high*	*windy*	*not windy*
0			×	×			×	×		×		×
1			×	×			×	×		×	×	
2		×	×				×	×		×		×
3	×	×				×	×			×		×
4	×	×			×	×			×			×
5	×	×			×	×			×		×	
6		×	×		×	×			×		×	
7			×	×		×	×			×		×
8			×	×	×	×			×			×
9	×	×				×	×		×			×
10			×	×		×	×		×		×	
11		×	×			×	×			×	×	
12		×	×			×	×	×	×			×
13	×	×				×	×			×	×	

图2. 网球数据集的概念尺度化（图1）。交叉表中的每个关联都用 × 表示。导出背景的概念格包含108个概念。

它们定义为集合对 $(A,\ B)$，其中 $A \subseteq G$、 $B \subseteq M$、 $A' = B$和$A = B'$ 成立，且对于所有 $g \in A\}$， $A' := \{m \in M \mid (g,\ m) \in I$；对于所有 $m \in B\}$, $B' := \{g \in G \mid (g,\ m) \in I$。 A 称为形式概念 $(A,\ B)$ 的**外延**，而 B 称为其**内涵**。图2中的导出背景有108个形式概念，远多于图1中决策树的规模。

这个例子已经表明，从单纯的概念尺度化导出的形式背景，其完整概念格过于复杂，难以用于说明问题。它确实展开了尺度化数据表中的全部概念结构。相较之下，只研究经过仔细筛选的导出属性子集及其诱导的子背景，可能更有用。这就是 Wille 在 Wille [38] 中所称的概念视图。下文中，我们将多值背景 $\mathbb{D}$ 经由概念尺度化 $\mathbb{D}$ 导出的形式背景称为背景视图。与之对应的概念格或其部分则称为概念视图。由于概念视图与背景视图一一对应，我们通常统称为视图。

逻辑尺度化 [29] 比普通尺度化更进一步。逻辑尺度化允许将尺度属性的（命题）逻辑组合用作导出背景的属性。这些尺度属性可能源自不同的多值属性。通过一个示例更容易理解这一简单技术。在网球数据集中，天气状况是通过指定湿度、风等的取值来描述的。你可能希望使用由给定属性组合而成的其他属性，例如

$$nice := (temperature = mild) \wedge \neg windy.$$

逻辑尺度化允许这样做。同样，我们只选取其中一小部分属性。因此，我们在这里也称之为**概念视图**。

2.4 从多值背景中学习决策树

为了在数据表上训练决策树（该数据表被形式化为一个多值背景 $\mathbb{D} = (G, M, W, I)$），我们需要知道所有 $g \in G$ 的分类标签。图1中的网球数据集展示了一个示例。最后一列包含分类标签，而表格的其余部分表示一个多值背景。然而，与现实生活中的实际数据集相比，这个示例非常小。对于此类数据集，需要使用像 C4.5算法 [32] 这样的快速实现。该算法在每一步都会选择 n 最有效地将其样本集 G 划分为多个子集的谓词 $\phi(n)$。这些谓词通常是属性-值对；如果对象 g 在属性上具有该值，则记作 $g \models \phi(n)$，否则记作 $g \models \neg\phi(n)$。术语最有效地 是相对于某种信息熵度量而言的。

结果是一棵决策树 $\mathcal{T}$，如第2.2节所述。在第一步中，根节点的划分决策被确定，所有样本的集合 G 被划分为两个集合，并分别传递给子节点。用于此次划分的谓词以注释形式写在子节点上，分别采用正类或否定形式。除根节点外的任何节点 $n \in \mathcal{T}$ 都会因此接收一条注释（为保证完备性，根节点的注释为 $\top$），并且当且仅当数据对象 $g \in G$ 是标注从根节点到该叶节点的唯一路径上各节点的每个谓词的模型时，它才会到达树的一个叶节点。

对于给定的决策树 $\mathcal{T}$ 及其训练数据集 $\mathbb{D}$，可以提出两项主要的解释任务。第一项关注的问题是： $\mathcal{T}$ 在多大程度上能够表示训练数据，尤其是对象 $g \in G$；以及如何利用 G 从 $\mathcal{T}$ 中推断出一般性解释。这些解释既包括局部解释，例如，为什么对象 g 被划分为 $\mathcal{C}$ 中的某个特定类别；也包括全局解释，例如，哪些谓词组合描述某个类别。第二项任务是理解 $\mathcal{T}$ 对一组迄今未知的对象 G 的视图；其中每个对象 $g \in G$ 都使用 $\mathbb{D}$ 的属性和值域来表示，甚至可能缺少某些取值。同样，这些视图既可以是局部的，也可以是全局的。

3 基于树分类器的概念格

下面我们将介绍树分类器的不同概念视图。首先回顾文献中的三种方法，并用形式概念分析的语言为它们推导出一种统一的表示。下一节中，我们将把这些方法作为两种新方法的基准。

3.1 文献中的方法

首先要考察的方法是 sklearn[6] 中的 RandomTreesEmbedding，它仅反映由决策树 [6, 26, 28] 的叶节点所诱导的数据对象聚类。

定义 1（关于 $\mathcal{T}$ 的概念叶视图）。给定一个多值背景$\mathbb{D} := (G, M, W, I)$，以及一棵在 $\mathbb{D}$ 上训练的决策树 $\mathcal{T}$，我们将关于 $\mathcal{T}$ 的上下文叶视图定义为

$$\mathbb{N}(G, \mathcal{T}) := (G, \mathcal{L}(\mathcal{T}), J), \textit{ where } (g, l) \in J \textit{ iff } g \models \phi(n) \textit{ for all } n \geq_{\mathcal{T}} l.$$

相应的概念格 $\mathfrak{B}(\mathbb{N}(G, \mathcal{T}))$ 称为关于 $\mathcal{T}$ 的概念叶视图。稍作推广后，可将具有相同属性和属性值的其他数据对象也纳入其中，而不仅限于训练数据。G 因此可以替换为$\check{}G$，正如第*2*节末尾所介绍的那样。这里用作属性集的集合 $\mathcal{L}(\mathcal{T})$，就是决策树的叶节点集。

与此类叶节点属性的关联由从根节点到该叶节点的唯一路径上所有谓词的合取定义。由于这是普通尺度化属性的逻辑组合，因此可以看出，该背景源自逻辑尺度化。具体的视图，即选取了哪些逻辑组合，取决于决策树。

$\check{}$ 基于特定对象集合 G 的概念叶视图能够通过 $\mathcal{T}$ 的分类叶节点来查看这些对象。这意味着，对于任意两个由同一叶节点 l 分类的对象g_1，$g_2 \in \check{}G$，我们有 $\{g_1\}^J = \{g_2\}^J$，因此对象概念（$\{g_1\}^{JJ}$，$\{g_1\}^J$）和（$\{g_2\}^{JJ}$，$\{g_2\}^J$）相等。通俗地说，g_1 和 g_2 被聚类到相同的概念中。此处可以注意到，这种视图仅限于这一事实。因此，由不同叶节点分类的对象不共享任何属性，因而彼此不可比较。图3展示了我们示例中的导出背景。

更一般地说，概念叶视图简单得令人失望，它不过是一个反链，加上一个顶元素和一个底元素。这是因为 $\mathfrak{B}$（$\mathbb{N}$（$\check{}G$，$\mathcal{T}$））具有名义标度，也就是说，每个叶节点恰好对应一个概念，且这些概念两两不可比较。总而言之，这种视图较为粗略，也无法呈现层级信息；换言之，除了涉及顶元素（$\check{}G$，$\check{}G^J$）或底元素（$\mathcal{L}$（T）^{J}，$\mathcal{L}$（T））的概念外，不存在处于子概念关系中的概念。
第二种基线视图考虑了决策树 $\mathcal{T}$ [6] 的整个序结构。相应的概念格与先前工作提出的概念格同构 [11]。

定义 2（关于 $\mathcal{T}$ 的概念树视图）。给定一个多值背景$\mathbb{D} := (G, M, W, I)$ 以及一个基于 $\mathbb{D}$ 训练的决策树 $\mathcal{T}$，我们通过将树节点视为属性来定义关于 $\mathcal{T}$ 的上下文树视图。当且仅当某个节点曾用于对某个数据对象进行分类时，该节点才与该数据对象关联。

$$\mathbb{T}(G, \mathcal{T}) := (G, \mathcal{T}, J), \textit{ where } (g, t) \in J \textit{ iff } t \textit{ is on the decision path of } g \textit{ in } \mathcal{T}.$$

[6] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomTreesEmbedding.html

相应的概念格 $\mathfrak{B}(\mathbb{T}(G,\mathcal{T}))$ 称为关于 $\mathcal{T}$ 的概念树视图˘。如上文定义 *1* 所示，G 可以替换为一个更一般的集合 G。

与叶节点划分背景类似，上下文树视图也可以理解为逻辑尺度化的结果。随后，每个节点都会被其路径上标注的谓词的合取式所替代。

与 $\mathcal{T}$ 上的上下文叶视图不同，上下文树视图涵盖了 $\mathcal{T}$的所有节点。因此，由不同叶节点分类的对象，其决策路径中可能包含共同的节点。它们各自的决策路径重叠越多，共同拥有的属性就越多。

概念格 $\mathfrak{B}(\mathbb{T}(G,\mathcal{T}))$与添加了一个最小元素的决策树序同构，即与$(\mathcal{T}\cup\{\bot\},\leq)$序同构，其中对所有 $n\in\mathcal{T}$，均有$\bot\leq n$。因此，$\mathcal{T}$上的概念树视图可以视为将决策树几乎一对一地转换到形式概念分析领域。

对于给定的任意对象集˘G，只能到达序结构$(\mathcal{T}\cup\{\bot\},\leq)$˘中的部分结构。更准确地说，由于 G 中的每个元素都有 G中的一个元素与之具有相同的决策路径，我们可以得出结论：存在一个从 $\mathbb{T}($˘$G,\mathcal{T})$到 $\mathbb{T}(G,\mathcal{T})$的某个子背景的同构（在背景澄清意义下唯一）。因此，$\mathrm{Int}(\mathbb{T}($˘$G,\mathcal{T}))\subseteq\mathrm{Int}(\mathbb{T}(G,\mathcal{T}))$。不过，在本文其余部分，我们不会进一步探讨这种关系。尽管如此，我们仍希望读者参阅我们关于概念测量 [17], 的研究，尤其是推论 22。

定义 3. 对于线性有序集 $(V,\leq)$，V 上的（一维）内序尺度 是形式背景

$$(V,\{\leq,\geq\}\times V,\models),$$

其中关联关系 $\models$ 以显而易见的方式定义：

$$v\models(\leq,w):\Longleftrightarrow\ v\leq w\ \textit{and}\ v\models(\geq,w):\Longleftrightarrow\ v\geq w.$$

用于决策树分类的数据集具有[7] 线性有序的值集，因此所有属性都有相应的内序尺度。这些尺度可用于普通尺度化。

我们要介绍的第三个也是最后一个基准，在形式概念方面表达能力最强。与另外两种视角不同，它完全取决于多值数据集 $\mathbb{D}$。虽然目前可能还看不出它与本工作的范围有何关联，但下一节将详细阐述其重要性。

定义 4（ $\mathbb{D}$的内序缩放）。当 $\mathbb{D}:=(G,M,W,I)$ 是一个多值背景，且对于所有多值属性集 $W(m)$，其值集均为线性有序的 $(m^{\mathbb{D}},\leq_m)$，那么由内序缩放导出的形式背景 以 G 为对象集，其属性形式为

$$(m,\leq_m,v)\ \textit{or}\ (m,\geq_m,v),$$

[7] 如果不是这样，则可以按线性顺序排列。

$\mathbb{N}(G,\ \mathcal{T})$	l_0	l_1	l_2	l_3	l_4	l_5	l_6
0	×						
1	×						
2		×					
3			×				
4						×	
5							×
6					×		
7	×						
8					×		
9						×	
10					×		
11		×					
12					×		
13				×			

$\mathbb{P}(G,\ \mathcal{T})$	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}	n_{12}
0		×				×						
1		×				×						
2					×					×		
3		×			×				×		×	
4	×		×				×					
5	×		×					×				
6	×			×								
7		×				×						
8	×			×								
9	×		×				×					
10	×			×								
11					×					×		
12					×				×			×
13		×			×							

图3. $\mathcal{T}$的上下文叶视图（图1右侧），以及运行示例（图1左侧）。其概念格包含55个概念。右侧显示了 $\mathcal{T}$的上下文树视图$\mathbb{P}(G,\ \mathcal{T})$。

其中，v 是多值属性 m的一个值。关联关系显而易见：对象 g具有属性 $(m, \leq_m, v)$，当且仅当对象 g在属性 m上的取值为 $\leq_m v$。用 $m:\leq v$和 $m:\geq v$分别代替 $(m, \leq_m, v)$和 $(m, \geq_m, v)$。形式上，$\mathbb{I}(\mathbb{D}) := (G, N, J)$，其中

$$N := \{m : \leq v \mid m \in M, v \in m^{\mathbb{D}}\} \cup \{m : \geq v \mid m \in M, v \in m^{\mathbb{D}}\}$$

and

$$(g, m : \leq v) \in J :\iff m(g) \leq v, \qquad (g, m : \geq v) \in J :\iff m(g) \geq v.$$

为简洁起见，通常省略适用于所有对象的属性。

通过简单的区间序尺度化，从我们的引入示例（图1）导出的背景如图2所示。值集的线性序已在第2.2节末尾列出。概念标度显示在左侧。例如，在这个导出背景中（右侧），对象5在导出属性“温度 $\leq$ 温和”上有一个叉号，因为原始数据中该对象的温度取值为“凉爽”，而凉爽 $\leq$ 温和。

所有刚刚介绍的尺度化方式有一个共同点：它们各自的概念格都是原子格。此外，在所有格中，原子都由所有对象概念构成。这一观察基于两个假设：a）尺度化所查看的任何对象都没有缺失值（参见完整多值背景）；b）每个叶节点都由一个对象 $g \in G$支持。

3.2 谓词视图

迄今为止所介绍的用于尺度化基于树的分类器的方法，并未将谓词用作属性集。然而，这些谓词对于人类理解至关重要。因此，接下来我们将介绍两种全新的概念视图，即，

概念尺度化，这些尺度化也可以从决策树中提取。然而，与定义1和定义2不同，属性集将由谓词 $\mathcal{P}(\mathcal{T})$ 构成，而非树节点。

为此，我们引入一种中间结构，具体而言是一个形式背景，用它取代 ϕ $\mathcal{T}$ 的标注函数。该形式背景定义为 $\mathbb{P}(\mathcal{T}) \coloneqq (\mathcal{T}, \mathcal{P}(\mathcal{T})\ J)$，其中 $(n, P) \in J$ 当且仅当存在一个 $n \leq_{\mathcal{T}} h$ 满足 $\phi(h) = P$。也就是说，当且仅当谓词 $P \in \mathcal{P}(\mathcal{T})$ 标注到 n 或 n 的某个前驱节点时，树 $\mathcal{T}$ 的节点 n 才与该谓词相关联。下文将此形式背景称为 $\mathcal{T}$ 的谓词视图。我们想说明这种结构为何能够借助谓词对 $\mathcal{T}$ 进行（形式化）解释。对于位于对象 g 的决策路径上的任意节点 n，都有 $g \models \{n\}^J$，即 g 是所有与 n 相关联的谓词的模型。此外，对于 g 的决策路径上的叶节点 l，集合 $\{l\}^J$ 恰好是 $\mathcal{T}$ 用于对对象 g 进行分类的谓词集合。因此，我们可以根据 $\mathcal{P}(\mathcal{T})$ 来解释任意对象的分类。

树谓词视图

定义 5（概念树谓词视图）。对于一个多值上下文 $\mathbb{D} : = (G、 M、 W、 I)$ 以及一个决策树 $\mathcal{T}$（在 $\mathbb{D}$ 上训练），我们将从 $\mathcal{T}$ 上的上下文树谓词视图导出的上下文定义为

$$\mathbb{T}_{\mathcal{P}}(G, \mathcal{T}) \coloneqq (G, \mathcal{P}(\mathcal{T}), I_{\mathbb{T}(G,\mathcal{T})} \circ I_{\mathbb{P}(\mathcal{T})}).$$

类似于定义 *1*，我们称 $\mathcal{T}$ 上的概念树谓词视图为 $\mathfrak{B}(\mathbb{T}_{\mathcal{P}}(G、\ \mathcal{T}))$。同样，该视图也可以应用于未知数据 $\check{}G$。

该视图的定义与上一节给出的定义略有不同。首先，树谓词视图的属性集由 $\mathcal{T}$ 的谓词组成。此外，$\mathbb{T}_{\mathcal{P}}(\check{}G、\ \mathcal{T})$ 的关联关系由树视图和谓词视图的关联关系的关系乘积 $\circ$ 隐式给出。因此得名树谓词视图。我们的思路是希望通过树节点将对象与谓词连接起来。例如，如果 $g \in G$ 与节点 $n \in \mathcal{T}$ 具有关联关系，而 n 又与某个谓词 $P \in \mathbb{P}(\mathcal{T})$ 具有关联关系，那么 g 就与 P 在 $I_{\mathbb{T}(\check{G},\mathcal{T})} \circ I_{\mathbb{P}(\mathcal{T})}$ 中具有关联关系。与树视图不同，具有不相交决策路径的对象仍可能拥有共同的谓词，因此在 $I_{\mathbb{T}_{\mathcal{P}}(\check{}G、\ \mathcal{T})}$ 中仍可能具有共同的关联关系。

树谓词视图 $\mathbb{T}_{\mathcal{P}}(\check{}G,\ \mathcal{T})$ 的概念格不一定呈树形，因为多次标注的谓词的交可能会产生额外的概念。特别是在 $G = G$ 的情况下，我们发现

$$\mathrm{Ext}(\mathbb{T}(G, \mathcal{T})) \subseteq \mathrm{Ext}(\mathbb{T}_{\mathcal{P}}(G, \mathcal{T})), \tag{1}$$

因为对于所有 $n \in \mathcal{T}$，我们都有

$$n^{I_{\mathbb{T}(G,\mathcal{T})}} = \{\phi(m) \mid m \geq n\}^{I_{\mathbb{T}_{\mathcal{P}}(G,\mathcal{T})}}.$$

$\mathbb{T}_{\mathcal{P}}(G,\mathcal{T})$	*humidity*≤*normal*	*overlook*≤*overcast*	*overlook*≤*rainy*	*windy*	*humidity*≥*high*	*overlook*≥*sunny*	*overlook*≥*overcast*	*not windy*
0					×	×		
1					×	×		
2		×			×		×	
3		×	×		×			×
4	×		×					×
5	×		×	×				
6	×						×	
7					×	×		
8	×						×	
9	×		×					×
10	×						×	
11		×			×		×	
12	×						×	
13		×	×	×	×			

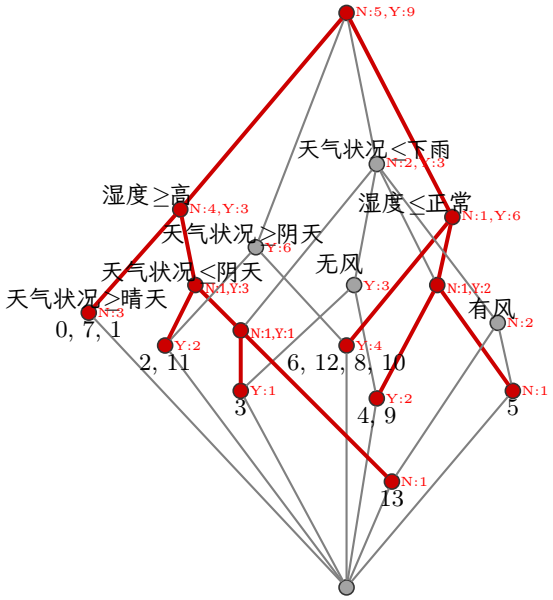


图4. 运行示例的上下文树谓词视图和概念树谓词视图（图1右侧）。原始决策树 $\mathcal{T}$ 在概念格图中以红色突出显示。

要了解其原因，请参阅命题1。我们在命题1中证明的关于 $\mathbb{T}_{\mathcal{P}}(\mathcal{T})$ 的另一个有用性质是，它的概念格是原子性的，且其对象外延与 $\mathbb{T}(\mathcal{T})$ 的对象外延相等。由此自然可知，概念视图的概念格能够像树分类器一样对每个对象 $g \in G$进行分类。也就是说，对于任意 $g \in G$，树谓词视图中 g 的闭包等于叶视图中的闭包，即由同一个叶节点分类的对象集合。对于任何没有缺失值的 $\check{g} \in G$，这一结论也成立。更重要的是，树谓词视图呈现了一些与 $\mathcal{T}$的任何节点都无关的概念；然而，这些概念解释了 $\mathcal{T}$的不同节点如何通过共同谓词相互关联。这就是公式（1）中的超集关系的原因。

如果树中的所有谓词都恰好标注一次，那么标注函数就是单射；也就是说，对于任意两个 $n,\ m \in \mathcal{T}$，都有 $\phi(n) = \phi(m) \implies n = m$。在这里，我们发现 $\mathrm{Ext}(\mathbb{T}(\check{G},\ \mathcal{T})) = \mathrm{Ext}(\mathbb{T}_{\mathcal{P}}(\check{G},\ \mathcal{T}))$。

互序谓词视图 刚刚介绍的树谓词视图能够反映对对象 g进行特定分类时所依据的重要谓词。然而， g 的关联关系仅限于标注在 g决策路径上的那些谓词。在即将介绍的视图中

我们希望通过将关联关系扩展到所有谓词 $P \in \mathcal{P}(\mathcal{T})$（其中 g 是$g \models P$的模型）来解除这一限制。

定义 6（概念性交错谓词视图）。对于一个多值背景$\mathbb{D} := (G、 M、 W、 I)$以及一个决策树 $\mathcal{T}$（在 $\mathbb{D}$上训练），我们将 $\mathcal{T}$上的概念性交错谓词视图定义为

$$\mathbb{I}_{\mathcal{P}}(G, \mathcal{T}) := (G, \mathcal{P}(\mathcal{T}), J),\ where\ (g, P) \in J\ iff\ g \models P.$$

类似于之前的所有定义，我们称 $\mathcal{T}$到 $\mathfrak{B}$（$\mathbb{I}_{\mathcal{P}}$（G、 $\mathcal{T}$）为概念性交错谓词视图。同样，该视图也可以应用于之前未˙知的数据 G。

在我们的设定中，有两种主要方法可以推导出互序谓词视图。自然的方法是应用普通尺度化，这会得到 $\mathbb{D}$的内序缩放的一个子背景，即 $\mathbb{I}_{\mathcal{P}}(G、 \mathcal{T}) \leq \mathbb{I}(\mathbb{D})$。这一点很容易看出，因为 $\mathcal{P}(\mathcal{T}) \subseteq N$，其中 N的定义与定义 4 中相同。注意，该定义要求值域是线性有序的。第二种方法是采用逻辑尺度化，其中每个逻辑表达式都恰好由涉及一个多值属性的条件组成。我们在图 5 中描绘了运行示例的互序谓词视图。

由于互序谓词视图与树谓词视图定义在相同的对象集和属性集上，因此这两种视图彼此相关。更确切地说，我们发现树谓词视图的关联关系是互序谓词视图的子集，即 $I_{\mathbb{T}_{\mathcal{P}}(\dot{G},\ \mathcal{T})} \subseteq I_{\mathbb{I}_{\mathcal{P}}(\dot{G},\ \mathcal{T})}$。

通过以下命题，我们希望说明经典视图与我们提出的新视图在训练数据集方面的关系，即 $G = G$。由此，我们可以推断这些视图能否以及在多大程度上用于解释基于树的分类器。

命题1。设 $\mathbb{D}$ 为一个完全多值背景， $\mathcal{T}$ 为一棵在 $\mathbb{D}$上训练的决策树，且 $\mathcal{T}$ 的每个叶节点都具有支持度。则以下陈述成立：

i) 对于每个视图 $\mathbb{N}(G, \mathcal{T})$ $\mathbb{T}(G, \mathcal{T})$ $\mathbb{T}_{\mathcal{P}}(G, \mathcal{T})$ 和 $\mathbb{I}_{\mathcal{P}}(G, \mathcal{T})$，我们发现对象概念是各自概念格中的原子；*ii)* $Ext(\mathbb{N}(G, \mathcal{T})) \subseteq Ext(\mathbb{T}(G, \mathcal{T})) \subseteq Ext(\mathbb{T}_{\mathcal{P}}(G, \mathcal{T}))$，且 $Ext(\mathbb{T}(G, \mathcal{T})) \subseteq Ext(\mathbb{I}_{\mathcal{P}}(G, \mathcal{T}))$；*iii)* $I_{\mathbb{T}_{\mathcal{P}}(G,\mathcal{T})} \subseteq I_{\mathbb{I}_{\mathcal{P}}(G,\mathcal{T})}$；*iv)* $\mathbb{N}(G, \mathcal{T})$ $\mathbb{T}(G, \mathcal{T})$ 和 $\mathbb{T}_{\mathcal{P}}(G, \mathcal{T})$ 的对象外延相等；*v)* 并且，如果属性 M 的值域线性有序，我们还发现 $\mathbb{I}(\mathbb{D})$ 的对象概念是其相应概念格中的原子，且 $Ext(\mathbb{I}_{\mathcal{P}}(G, \mathcal{T})) \subseteq Ext(\mathbb{I}(\mathbb{D}))$ 成立。

证明。 i）对于名义尺度，原子的集合由对象概念组成，这是一个已知事实。由于引入的叶视图采用名义尺度，我们可以推断该陈述为真。

$\mathbb{I}_{\mathcal{P}}(\mathcal{T})$	$humidity \leq normal$	$overlook \leq overcast$	$overlook \leq rainy$	$windy$	$humidity \geq high$	$overlook \geq sunny$	$overlook \geq overcast$	$not\ windy$
0					×	×	×	×
1				×	×	×	×	
2		×			×		×	×
3		×	×		×			×
4	×	×	×					×
5	×	×	×	×				
6	×	×		×			×	
7					×	×	×	×
8	×					×	×	×
9	×	×	×					×
10	×			×		×	×	
11		×		×	×		×	
12	×	×					×	×
13		×	×	×	×			

图 5. 运行示例（图1左侧）的上下文互序谓词视图（ $\mathcal{T}$，图1右侧）。其概念格包含55个概念。

对于树视图 $\mathbb{T}(G,\ \mathcal{T})$，我们知道 $\{g\}^J$ 等于 $\mathcal{T}$ 中位于 g的（完整）决策路径上的节点集合。假设在（$\{g\}^{JJ}$，$\{g\}^J$）下方存在一个形式概念，那么必然存在一个 $h \in G$，使得$\{g\}^J \subseteq \{h\}^J$ 和 $\{g\}^J \neq \{h\}^J$。然而，这意味着 g 的决策路径可以延伸，从而产生矛盾。需要注意的是，对于不完备的多值背景 $\mathbb{D}$，这一论证并不成立。

对于树谓词视图，我们可以运用相同的论证。

对于互序谓词视图，假设有两个人物 $g,\ h \in G$ 满足$\{g\}' \subseteq \{h\}'$和 $\{g\}' \neq \{h\}'$，那么 $\{h\}'$不是 $\mathfrak{B}(\mathbb{I}_{\mathcal{P}}(G,\ \mathcal{T}))$中的一个原子。注意，导出 $(\cdot)'$是相对于 $\mathbb{I}_{\mathcal{P}}(G,\ \mathcal{T})$进行的。因此，存在一个谓词$P \in \{h\}'$，满足 $h \models P$和 $g \not\models P$。由于完全多值背景带来的完备性，我们可以推断出 $g \models \neg P$。因此，$\neg P \in \{g\}'$，这与假设相矛盾。

ii) 叶节点集合 $\mathcal{L}(\mathcal{T})$ 是 $\mathcal{T}$ 中所有节点的子集。由此可推知，$\mathbb{N}(G,\ \mathcal{T})$ 是 $\mathbb{T}(G,\ \mathcal{T})$ 的属性诱导子背景，因此 $\mathrm{Ext}(\mathbb{N}(G,\ \mathcal{T})) \subseteq \mathrm{Ext}(\mathbb{T}(G,\ \mathcal{T}))$，参见形式背景并置 [15]。

接下来我们要证明 $\mathrm{Ext}(\mathbb{T}(G,\mathcal{T})) \subseteq \mathrm{Ext}(\mathbb{T}_{\mathcal{P}}(G,\mathcal{T}))$。对于 $A \subseteq G$ 的非空外延 $A \subseteq G$ of $\mathbb{T}(G,\mathcal{T})$，我们知道，其在上述视图中的导出是从根节点到某个节点 n 的决策树路径。对于决策树而言，这条路径由标注谓词唯一确定，因为根节点中划分的谓词 $\mathcal{Q} \subseteq \mathcal{P}(\mathcal{T})$ 不能被标注两次。否则将导致一个不受支持的叶节点，这与命题的要求相矛盾。 $\mathcal{Q}$ 在 $\mathbb{T}_{\mathcal{P}}(G,\mathcal{T})$ 中的导出等于 A，因为经过节点 n 的所有对象恰好都是 Q 的模型。接下来还需证明

空集在两个视图中都是外延，这由 i) 以及决策树 $\mathcal{T}$ 至少包含一个划分这一事实可知。[8] 因此，$\mathsf{Ext}(\mathbb{T}(G,\mathcal{T})) \subseteq \mathsf{Ext}(\mathbb{T}_{\mathcal{P}}(G,\mathcal{T}))$。同样的论证也可应用于 $\mathbb{I}_{\mathcal{P}}(G,\mathcal{T})$。

对于从根节点到任意其他节点的路径 $n \in \mathcal{T}$，令 $\mathcal{Q} \subseteq \mathcal{P}(\mathcal{T})$ 表示标注谓词的集合。那么，$\mathcal{Q}$ 在 $\mathbb{T}_{\mathcal{P}}(G, \mathcal{T})$ 中的导出等于 $\mathcal{Q}$ 在 $\mathbb{I}_{\mathcal{P}}(G, \mathcal{T})$ 中的导出。这是因为经过节点 n 的对象集合，即 $\mathcal{Q}^{I_{\mathbb{T}_{\mathcal{P}}(G,\mathcal{T})}}$，由满足 $\mathcal{Q}$ 的对象集合给出，即 $\mathcal{Q}^{I_{\mathbb{I}_{\mathcal{P}}(G,\mathcal{T})}}$。因此，$\mathsf{Ext}(\mathbb{T}(G, \mathcal{T})) \subseteq \mathsf{Ext}(\mathbb{I}_{\mathcal{P}}(G, \mathcal{T}))$。
iii) 这直接由它们的定义得出。
iv) $A \subseteq G$ 属于 $\mathbb{N}(G, \mathcal{T})$ 的任意对象外延都具有这样的性质：存在唯一的叶节点 $l \in \mathcal{T}$，使得 A 是由 l 分类的对象集合。
由 i) 可推知，$\mathbb{T}_{\mathcal{P}}(G, \mathcal{T})$ 的对象外延是概念格 $\mathfrak{B}(\mathbb{T}_{\mathcal{P}}(G, \mathcal{T}))$ 中的原子。对于 $\mathbb{N}(G, \mathcal{T})$ 的任意对象外延 A，令 $g \in G$ 为 A 的一个生成元，即 $\{g\}^{I_{\mathbb{N}(G,\mathcal{T})}I_{\mathbb{N}(G,\mathcal{T})}} = A$。

我们可以通过计算 g 在 $\mathbb{T}_{\mathcal{P}}(G, \mathcal{T})$ 中的导出，找到树谓词视图中与之关联的谓词集合。具体而言，我们可以通过投影来计算 $\{g\}^{I_{\mathbb{T}_{\mathcal{P}}(G,\mathcal{T})}}$

$$\{(g, n) \mid n \text{ is on the dec. path of } g\} \circ \{(n, P) \mid \exists m \in \mathcal{T} : n \leq_{\mathcal{T}} m \text{ and } \phi(m) = P\}$$

到第二个元素上。该集合等于标注在 g 决策路径上的谓词集合。在 $\mathbb{T}_{\mathcal{P}}(G, \mathcal{T})$ 中第二次应用导出运算，得到的是作为 $\{g\}^{I_{\mathbb{T}_{\mathcal{P}}(G,\mathcal{T})}}$ 的模型的对象集合。因此，$\{g\}^{I_{\mathbb{T}_{\mathcal{P}}(G,\mathcal{T})}I_{\mathbb{T}_{\mathcal{P}}(G,\mathcal{T})}}$ 的所有元素都被归入与 g 相同的叶节点。由此，$\mathbb{N}(G, \mathcal{T})$ 和 $\mathbb{T}_{\mathcal{P}}(G, \mathcal{T})$ 的对象外延相等。其余结论可直接由 ii) 得出。

v) 众所周知，对于间序标度完备多值背景，对象概念是相应概念格的原子。

此外，对于线性有序值域，我们发现 $\mathcal{P}(\mathcal{T}) \subseteq N$ 其中$N := \{m :\leq v \mid m \in M, v \in m^{\mathbb{D}}\} \cup \{m :\geq v \mid m \in M, v \in m^{\mathbb{D}}\}$，由此可推知 $\mathbb{I}_{\mathcal{P}}(G, \mathcal{T})$ 是 $\mathbb{I}(\mathbb{D})$ 的属性诱导子背景。因此，该结论成立。

□

根据这一命题，我们可以得出关于树分类器概念性诠释（或视图）的重要结论。根据 ii)，我们可以推断，树谓词视图和互序谓词视图的外延结构都蕴含了该外延结构。因此，这两种视图都涵盖了整个决策树结构。为了在我们的示例中展示这一点，我们在图4（右）中将决策树描绘在训练数据的树谓词视图中 G。除了 $\mathcal{T}$ 的决策树结构之外，我们还可以在 $\mathbb{T}_{\mathcal{P}}(G, \mathcal{T})$ 中观察到多个跨越不同树分支的谓词组合。这一理论发现为决策树的诠释带来了若干种方法。我们将结合示例介绍并讨论这些方法。

[8] 我们在第2节中对本研究所考虑的所有决策树都提出了这一要求。

替代叶描述：我们的方法能够以 $\mathcal{T}$ 的谓词语言为叶节点生成替代描述。例如，图4中用于对对象 13 分类的叶节点 l包含谓词 湿度≥高，天气状况≤阴天，天气状况≤下雨，有风。该叶节点在树谓词视图的概念格中有一个上邻节点，其属性为有风。 $\mathcal{T}$ 中没有表示这一概念的节点。不过，我们可以利用这一概念构造一个替代谓词组合，生成 l 的概念。该叶节点可以表示为交 $\bigwedge${有风，湿度≥高}。事实上，对于与 l 相关联的概念的内涵而言，这是一个极小生成元。

> 为了理解给定的决策树 $\mathcal{T}$，可以为所有叶概念生成全部极小生成元，并将其作为更简短的描述，以理解树的分类结构。由此获得的较短解释可能更易于理解。尤其是在决策树规模较大时，这种方法很有用，例如，当决策树是在包含许多属性的大型数据集上训练时。

解释叶集合：对于任意一组叶节点 $L \subseteq \mathcal{T}$，我们可以在树谓词视图中计算它们的（概念）并（$A,\ B$）：$= \bigvee L$。形式概念（$A,\ B$） $\in \mathfrak{B}(\mathbb{T}_{\mathcal{P}}(G,\ \mathcal{T}))$ 不一定与 $\mathcal{T}$ 中的某个节点相关联，例如，求图4中包含对象 13 和 5 的叶节点的并。从中我们了解到，这两个叶节点共享谓词有风。沿着格向顶概念移动时，我们会发现，叶节点 13 和 5 还共享谓词 天气状况≤下雨。相比之下，在 $\mathcal{T}$ 中，这两个叶节点的决策路径只有根节点是共同的。因此，我们的方法能够表达一组叶节点 L 的共性，而这些共性无法在 $\mathcal{T}$ 的结构中表达。

缺失数据处理：前两个条目中的替代描述有助于应对缺失属性。例如，在对一个属性 天气状况没有取值的对象 $g \in G$ 进行分类时，谓词树视图可以利用属性 有风和湿度，将 g 映射到包含对象 13 的叶节点，如前面的条目所述。

谓词的全局影响：解释和说明决策树的一种常见方法，是找出树中最先或第二个使用的属性。然而，正如树谓词视图所揭示的，还有其他在结构上很重要的谓词，例如天气状况≤下雨 和天气状况≥阴天，因为除了根节点外，与这些谓词对应的概念没有上邻居。若形式概念 ($A,\ B$) 被另一个概念 ($C,\ D$) 支配，当且仅当 ($A,\ B$) $\leq$ ($C,\ D$)。基于这一概念，我们可以说，谓词 P被另一个谓词 Q 支配，当且仅当 P 的属性概念是 Q的属性概念的下邻居，即 $P^J \subseteq Q^J$。

谓词叶覆盖率：衡量决策树 $\mathcal{T}$中谓词 P重要性的另一项指标，是该谓词涉及的叶节点数量。决策树中最先使用的谓词会涉及许多叶节点，这并不令人意外。然而，正如我们从图4的格图中可以推断出的那样，谓词 天气状况≤下雨 涉及四个叶节点，而它在树中的前驱只涉及三个叶节点。因此，

关于 $\mathcal{T}$ 的概念视图有助于从结构上识别重要谓词。此外，还可以轻松选出一个谓词子集，使其覆盖树 $\mathcal{T}$ 的所有叶节点。

刚才介绍的决策树分析与解释方法，可以用类似的方式应用于序际树视图，正如命题 1 ii) 所指出的。一般而言，给定外延集上的集合包含关系，可以将视图 $\mathbb{N}(G,\mathcal{T})$ $\mathbb{T}(G,\mathcal{T})$ 视为粗粒度视图，将视图 $\mathbb{T}_{\mathcal{P}}(G,\mathcal{T})$ $\mathbb{I}_{\mathcal{P}}(G,\mathcal{T})$ 视为中间粒度视图，并将视图 $\mathbb{I}(\mathbb{D})$ 视为 $\mathcal{T}$ 的细粒度缩放。

3.3 树集成视图

上一节中，我们用一种统一的语言介绍了所有树视图。正如我们所见，这使我们能够比较它们对决策树的不同视角。这种可比性为全面解释树集成奠定了基石。具体而言，我们提出了一种原则性方法，用于解释树的集合；这些树常用于监督式机器学习流程，例如随机森林或 AdaBoost。下文将使用记法 $\mathbb{S}(\check{}G,\mathcal{T})$，作为上一节介绍的任意一种视图的统称。

定义 7（森林视图）。对于多值背景 $\mathbb{D} := (G, M, W, I)$、在 $\mathbb{D}$ 上训练的一组决策树 $\mathfrak{T} = (T_i)_{i\in F}$，以及每棵树对应的一个概念视图 $\mathbb{S}(\check{}G, T_i)$，我们将它们的森林视图 定义为

$$\mathbb{S}(G,\mathfrak{T}) := \bigcup_{i\in F} \mathbb{S}(G,T_i), where\ \mathbb{S}_1 \cup \mathbb{S}_2 := (G_1 \cup G_2, M_1 \cup M_2, I_1 \cup I_2).$$

$\check{}$同样，此视图也可应用于此前未知的数据 G。

我们想进一步阐释此定义背后的推理。首先可以注意到，对于我们的视图，集合 $G_1 = G_2 = G$，因此，它们的并集为 G。这是有意为之，因为我们希望我们的方法能够像决策树视图一样，针对特定对象集 G 来解释森林。相比之下，属性集和关联关系可能有所不同。因此，森林视图提供解释的能力直接取决于树视图集合中所有属性的并集。

随机森林通过两种关键技术构建：装袋法和提升法的一种经验变体。装袋法也称为自助聚合，指的是从训练数据集中均匀、有放回地抽取样本的过程。常见做法是为集成中的每棵树抽取各自的训练数据集 [8]。用形式背景的语言来说，每棵树都是通过一个随机诱导子背景 $\mathbb{H} \leq \mathbb{D}$构建的。不过，这种建模没有考虑同一个对象可能被抽取两次的情况；显然，可以通过创建对象副本来解决这一问题。接下来，我们将讨论不同树视图对森林视图的影响。

叶视图：在此视图中，属性集由集成 $\mathfrak{T}$中任意一棵树 $\mathcal{T}$的叶节点 $\mathcal{L}(\mathcal{T})$ 构成。对于任意两棵树 $\mathcal{T}_1$ 和 $\mathcal{T}_2$，我们将它们的叶节点集 $\mathcal{L}((\mathcal{T}_1))$ 和 $\mathcal{L}(\mathcal{T}_2)$ 视为不相交集合，即 $\mathcal{L}(\mathcal{T}_1) \cap \mathcal{L}(\mathcal{T}_2) = \emptyset$。这意味着，树以叶尺度化的任何森林视图，都等于一组叶视图的形式背景并置 $|_{i\in F}\, \mathbb{S}(\check{}G,\ \mathcal{T}_i) = \bigcup_{i\in F} \mathbb{S}(\check{}G,\ \mathcal{T}_i)$。由形式背景并置可推得，其外延集，即 $\mathrm{Ext}(|_{i\in F}\, \mathbb{S}(\check{}G,\ \mathcal{T}_i))$，等于所有树视图的外延子集的交集集，即 $\{\bigcap \mathcal{A} \mid \mathcal{A} \subseteq \bigcup_{i\in F} \mathrm{Ext}(\mathbb{S}(\check{}G,\ \mathcal{T}_i))\}$。

树视图：与叶视图的建模方式类似，对于任意两棵树 $\mathcal{T}_1, \mathcal{T}_2$，我们认为它们的节点互不相交，即 $\mathcal{T}_1 \cap \mathcal{T}_2 = \emptyset$。因此，森林视图等于所有树视图的并置，其外延也会受到与上一项所述相同的影响。

树谓词视图：基于树谓词视图的森林视图比前两种更为复杂。例如，给定两棵树$\mathcal{T}_1, \mathcal{T}_2$、一个对象 $g \in \check{}G$ 和一个谓词 $P \in \mathcal{P}(\mathcal{T}_1) \cap \mathcal{P}(\mathcal{T}_2)$，可能出现这样的情况：$(g, P) \in I_{\mathbb{T}_{\mathcal{P}}(\check{}G, \mathcal{T}_1)}$，但 $(g, P) \notin I_{\mathbb{T}_{\mathcal{P}}(\check{}G, \mathcal{T}_2)}$。因此，基于树谓词视图的森林视图并非简单地将各个单棵树谓词视图并置而成。由此，森林视图可能产生并非已知外延的简单交集的外延。使用树谓词视图构建森林视图的一个优势是，所得表示更小。这是因为任意两棵树可能共享谓词，即 $\mathcal{P}(\mathcal{T}_1) \cap \mathcal{P}(\mathcal{T}_2) \neq \emptyset$，而且这些谓词尤其可能数量很大。

互序谓词视图：与上一个视图相比，对于随机森林的互序谓词视图，我们可以这样表述：给定两棵树$\mathcal{T}_1, \mathcal{T}_2$、一个对象 $g \in \check{}G$ 和一个谓词 $P \in \mathcal{P}(\mathcal{T}_1) \cap \mathcal{P}(\mathcal{T}_2)$，当且仅当 $(g, P) \in I_{\mathbb{I}_{\mathcal{P}}(\check{}G, \mathcal{T}_1)}$ 时，$(g, P) \in I_{\mathbb{I}_{\mathcal{P}}(\check{}G, \mathcal{T}_2)}$。由此我们可以推断，森林视图几乎就是互序谓词视图集合 $\mathbb{I}_{\mathcal{P}}(\check{}G, \mathcal{T}_i)$的并置，唯一的例外是：通过着色，不会重复显示在多个视图中出现的谓词 P。由于属性澄清（即移除形式背景中的重复属性）不会影响外延集合，因此我们可以采用与叶视图相同的推理方式。

刚刚介绍的这些视图为树集成的可解释性开辟了多种实际应用，我们将在第 4 节中对此进行更详细的研究。

4 处理大型概念视图

我们在第3.2节和第3.3节讨论了不同概念视图的实用性和适用性。然而，对于许多真实规模数据示例，这些视图可能大到难以理解。因此，为了从概念视图中提取人类可理解的选择结果和聚合结果，我们介绍以下方法。这些方法基于形式背景常用的数据缩减流程，但针对概念视图进行了调整。

对象或属性选择 第一类方法是选择方法，用于计算概念视图的诱导子背景。选择对象集的一个子集，将在属性集上产生一个更粗的闭包系统 [17] 。选择背景视图的属性，对对象 [17]也会产生相同的效果。从形式背景中选择相关属性的方法有很多 [12, 19]。在我们的实验（第5节）中，我们使用随机森林模型提供的特征重要性分数。此外，还可以应用 KMedoid 聚类来识别代表性对象，我们称之为中心对象。与 kmeans 等其他常用方法相比，KMedoid 的优势在于，聚类中心是数据集中的现有对象。因此，可以将聚类解释为：从原始对象集的一个子集中计算诱导子背景。

基于结构的对象选择 选择对象的一种特定方法，是依据对象在概念格中的结构位置。对于给定对象 $g \in G$，一种自然的方法是针对背景视图 $\mathbb{S}$计算 $\mathbb{S}$ 的序滤子 $\uparrow\{g\}^{I_{\mathbb{S}}I_{\mathbb{S}}} \subseteq \mathfrak{B}(\mathbb{S})$。这样便可得到一个局部概念视图 ，从而为单个对象推导解释。第二种方法进一步纳入 $\uparrow\{g\}^{I_{\mathbb{S}}I_{\mathbb{S}}}$ 的邻近概念。由此得到的局部概念视图能够更全面地解释 g 在 $\mathcal{T}$中的结构位置，以及其对不同属性值的依赖关系。具体而言，这使得我们能够开展与偏依赖图 [20], 相当的研究，即研究属性值扰动。可以利用集合 $\uparrow\{g\}^{I_{\mathbb{S}}I_{\mathbb{S}}}$ 的覆盖元素添加邻近概念，即对于 $B \in \uparrow\{g\}^{I_{\mathbb{S}}I_{\mathbb{S}}}\}$，$\{A \in \mathfrak{B}(\mathbb{S}) \mid A \prec B \vee B \prec A$ 。这些元素可以使用下一个_邻居算法 [25]递归枚举。

—

概念选择方法 为了减少形式概念的数量，通常会采用不同的标准来衡量其重要性。形式概念分析文献提供了大量度量指标 [23]。在我们的实验工作中，我们根据支持度 [35] （TITANIC）选择概念，即外延中所包含的对象数量除以对象总数。该过程得到概念视图中概念的一个子集。该集合构成一个并半格，即冰山概念格。

组合方法 另一种方法是根据对象集或属性集的给定划分，将概念视图分成多个部分。通过对各部分进行交与并运算，可以从中还原概念视图的原始概念。利用对象集的类别标签，可以得出合理的划分。因此，可以针对每个类别标签分别绘制图形。对于属性的划分，一种有意义的选择是依据其语义进行划分，例如利用本体背景知识。此外，还可以将视图限制为阈值的特定顺序方向 ，即 $\leq$ 和 $\geq$。这一过程可以

视为相对于形式背景并置运算 [15] 的序数因子。

属性聚合 导致概念视图中的概念数量增多的一个原因，是训练过程中推导出的不同谓词的数量。对不同谓词进行聚类并加以聚合，可能会显著减少概念数量。为此，应考虑这些谓词所依据的不同属性值分布。将等级 作为聚合值进行聚类，例如使用低 $\leq$中等 $\leq$高，尤其便于人类读者理解。

5 实验研究

以下实验旨在从两个实际研究问题的角度，为我们的理论发现提供支持。首先，概念视图的规模是否便于人类理解？其规模在多大程度上取决于树训练算法超参数的选择？其次，基于概念视图得出的解释对人类理解是否有意义？为回答这些问题，我们使用随机森林进行了两项实验。我们选择了知名的汽车数据集 [13,ID:991], 作为数据集，其中包含 1728 个对象，涉及七个（多值）属性。该数据集呈现一个二分类问题，类别标签为正类和负类。

5.1 概念视图的大小：参数研究

我们通过参数研究来探讨第一个研究问题。随机森林流程中最重要的两个超参数是树的数量 (nt)及其最大深度 (md)。每棵树的属性数量、纯度、划分准则等其他参数也会产生显著影响，但不会影响最终概念视图的大小。

在本研究中，我们使用 $2 \leq \mathrm{nt} \leq 10$和 $2 \leq \mathrm{md} \leq 20$训练了不同的随机森林分类器。对于每种参数组合，我们使用十个不同的初始随机种子完成了十次运行。在图6（左）中，我们使用平均准确率报告分类性能；在图6（右）中，我们报告泛化误差。后者通过从训练数据集上取得的准确率中减去测试数据集上的准确率得到，即 $\mathrm{Error}_{\mathrm{Generalization}} \coloneqq \mathrm{ACC}_{\mathrm{Train}} - \mathrm{ACC}_{\mathrm{Test}}$。该值使我们能够估计训练所得的随机森林分类器出现过拟合的程度。在所有实验中，我们均进行了四折交叉验证，但观察到了更稳定的结果。

观察 从监督学习的角度来看，我们注意到，对于模型准确率，最大深度 md 的影响大于树的数量

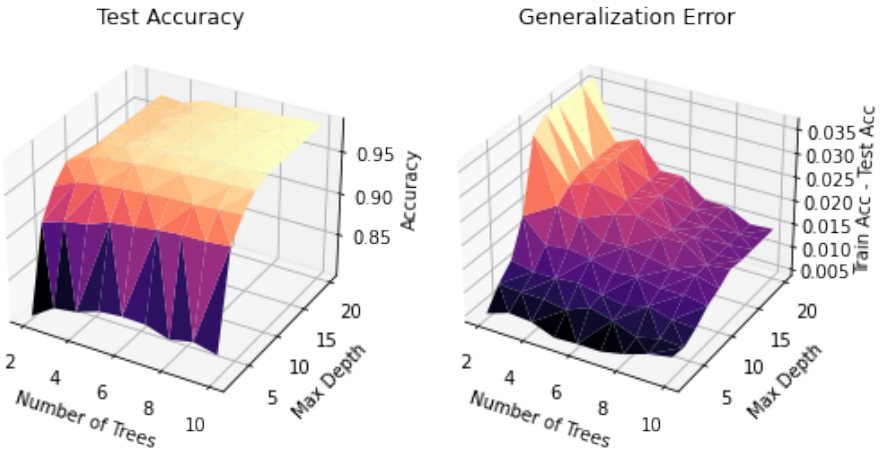


图6。 使用超参数 md 和 nt 训练的随机森林分类器的准确率（ACC）和泛化误差等性能指标分布的可视化结果。图中报告了平均值。

nt。不过，根据泛化误差（图6，右图），我们发现树的数量对于防止过拟合至关重要。此时，我们有充分理由认为，对于 $\mathrm{nt} \geq 8$ 和 $\mathrm{md} \geq 15$，存在非常出色的分类器。由此，我们可以转向概念视图，探讨它们解释随机森林分类器的能力。

为此，我们首先考察参数对每个相应概念视图中概念数量的影响。在本实验中，随机森林分类器在整个数据集上进行训练，未采用交叉验证。随后，我们同样使用整个数据集计算了不同的视图 $\mathbb{S}(G,\ \mathfrak{T})$，并在图7中展示了每个视图及参数组合下的形式概念数量。

观察结果 首先，我们注意到，三个不同概念视图的可视化形状在某种程度上相似。例外是互序谓词视图，它会随着 md 的增加而更快增长。这一现象是意料之中的，因为随着树深度的增加，划分谓词的出现次数会增加；并且与树谓词视图不同，对象-谓词关联与这些谓词在树中的位置无关。对于所有图，我们可以报告，将深度增加到十以上不会产生明显影响。图6中的性能测量结果表现出类似的趋势；然而，由于我们采用了不同的训练集，因此不应直接进行比较。

就形式概念的绝对数量而言，我们发现叶视图生成的数量最少（5,000），其次是树视图（15,000）、树谓词视图（80,000）和交序谓词视图（200,000）。根据我们在第3节中的理论发现，这一观察结果符合预期。显然，

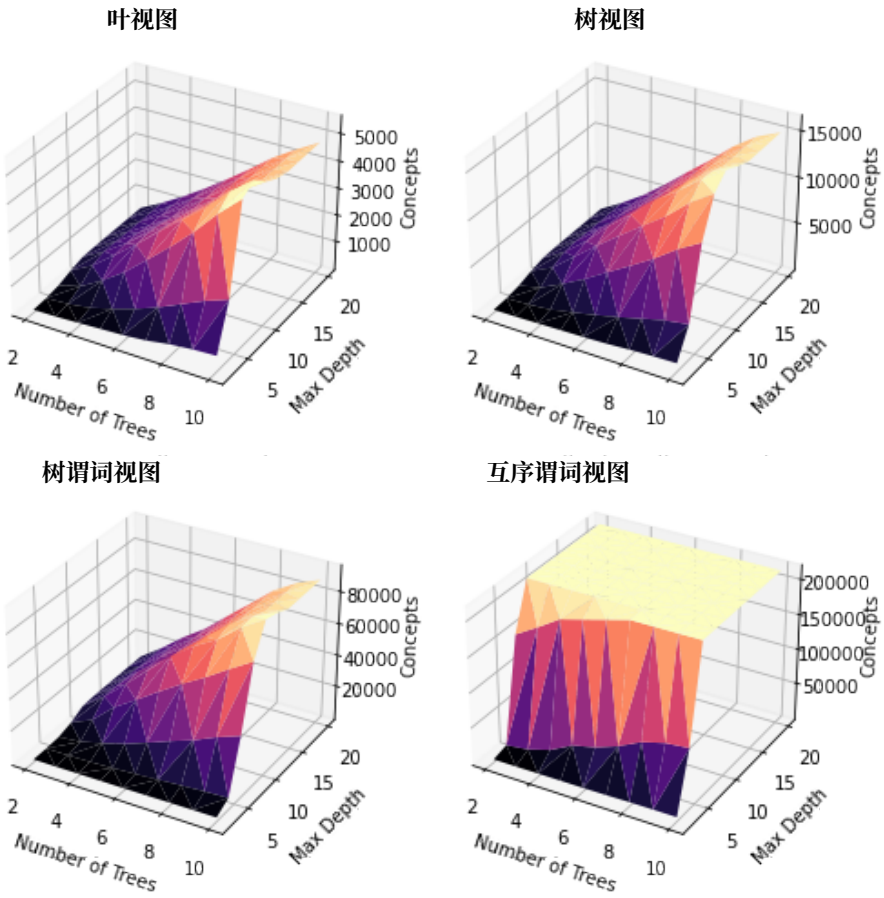


图7. 不同超参数组合和不同概念视图中形式概念数量的分布：叶视图（左上）、树视图（右上）、树谓词视图（左下）和互序谓词视图（右下）。

由于观察到的形式概念数量众多，所有视图都难以被人类直接理解。因此，需要采用第4节提出的连续数据缩减方法。

5.2 推导有意义的概念解释

为了最后展示如何使用概念视图解释随机森林分类器，我们选择了一组极端的超参数，以证明我们方法的可行性。具体而言，我们将树的数量设置为100，这是通常被接受的默认值 [30]。对于树的最大深度，我们不作限制，即训练算法会不断划分节点，直到达到类别纯度。由此产生的概念视图，更准确地说，其形式概念的数量，自然会增加到上一节所见的数量，甚至更多。尽管对于形式概念分析领域的算法而言，计算如此庞大甚至更大的形式概念集合并非难事，但为了实现人类可理解性，现在需要应用第4节介绍的选择与聚合方法。

我们希望先将组合方法与对象、属性和概念选择流程结合起来。为此，我们首先使用 sklearn 中的 KMedoids聚类，选出数量较少的代表性对象。通过反复试验，并评估范围 $2 \leq k \leq 50$ 内结果的轮廓系数，我们将参数 k（即中心对象的数量）确定为十九。第二步，我们按以下方式限制视图属性（即谓词）的集合。我们使用置换重要性 [2] 的概念，计算了汽车数据表中全部七个多值属性对分类的重要性。根据结果，我们可以选出最重要的属性。在本研究的其余部分，我们采用四个属性。根据这些数据表属性，我们可以推导出一个重要谓词子集，即视图属性。

以此为基础，我们还应用了其他各种选择方法。图8展示了树谓词视图的结果，其中还额外应用了 a）组合方法，更具体地说，我们使用相关类别标签对对象集进行划分；以及 b）TITANIC算法 [35]。

图8中的上方图由带有正类标签的对象构成，下方图则由带有负类标签的对象构成。正类和负类对应的最小支持度分别为五和三，即正类视图中的所有概念的外延大小至少为五，负类部分也同理，至少为三。选择这些取值是为了使生成的冰山概念格规模适中。三和五这两个取值似乎在一定程度上反映了类别标签的不平衡，不过这一观察并非必要。两个图都采用常规方式进行了标注。此外，我们还在每个概念节点右侧标注了该概念的类别纯度，即数据表中带正类标签和负类标签的对象数量。

首先，我们观察到正类中心对象的冰山概念格（PICL）与负类中心对象的冰山概念格（NICL）在结构上存在差异，尽管二者都包含二十五个形式概念。PICL 有十二个余原子，而 NICL 有四个余原子。NICL 的最长链包含五个元素，而 PICL 的最长链包含三个元素。

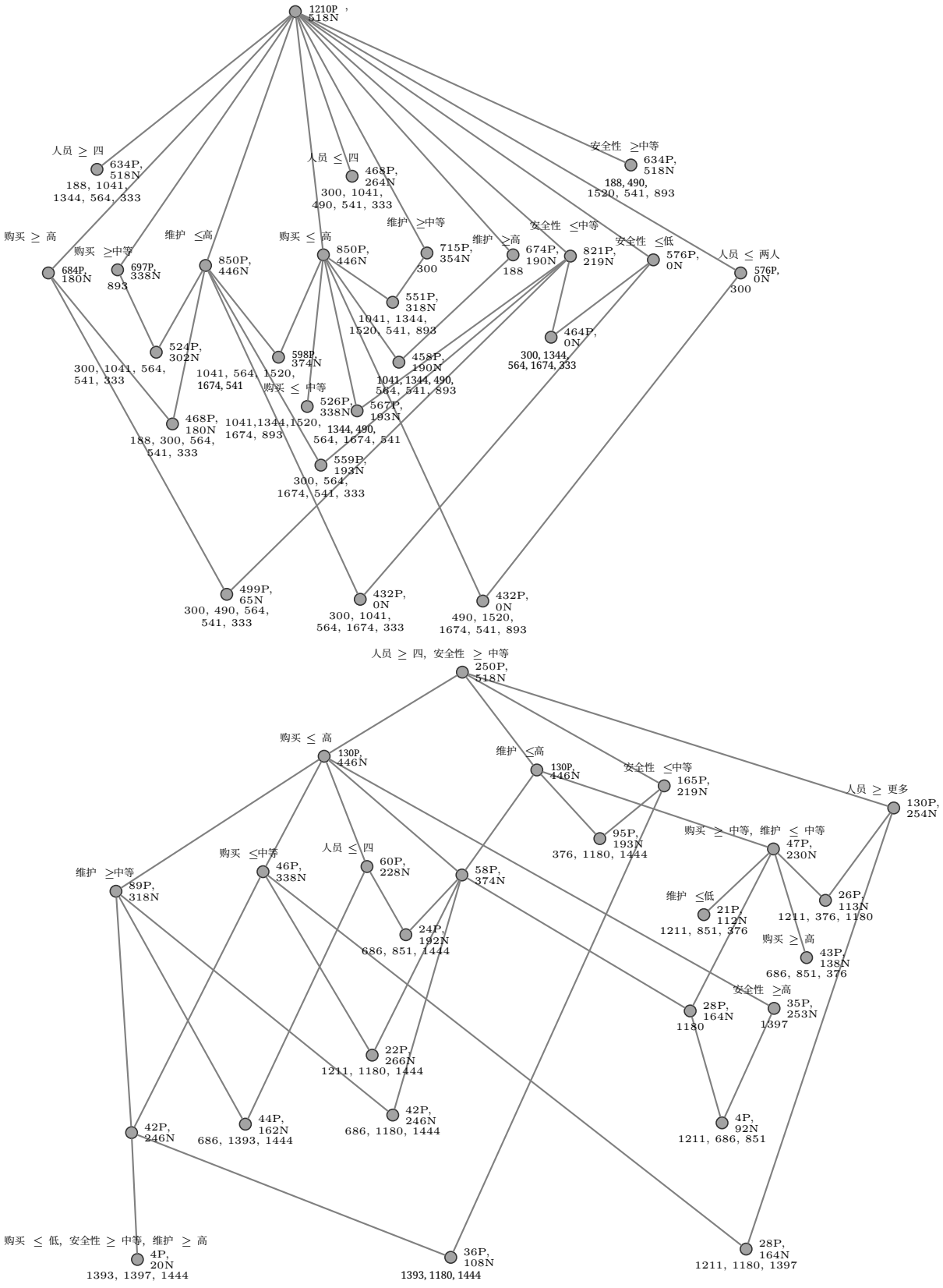


图8. 汽车数据集的树谓词视图尺度化。属于正类的质心元素显示在上方，负类的显示在下方。

我们认为，由于 PICL 的深度较小，因此比 NICL 更容易理解。与此同时，NICL 表明，负类的描述在属性数量（即意图大小）方面更为困难，也要求更高。更一般地说，在两张图中，我们都可以根据图中位置最低的概念，推断正类和负类的描述。尽管已有方法可以解释单个属性对分类的影响，但冰山概念格能让人轻松理解任意属性组合的影响。例如，外延标签为 300、490、564、541、333 的概念，是由属性组合购买 $\geq$高 和 安全性 $\geq$ 中等产生的。此外，这些概念结构有助于识别对分类具有较大整体影响的属性。例如，维护 $\leq$ 高 有五个直接下邻居，而人员 $\leq$ 两人 有一个直接下邻居。

NICL 图中一个特别有趣的观察是，存在支持所有对象的属性。因此，这些属性对于将所有对象分类为负类标签至关重要。与分析底层随机森林的全部一百棵树相比，从概念结构中尤其容易推断出这一结论。最后，NICL 的冰山概念格揭示了冗余属性。例如，标注了对象外延 1393、1397、1444 的概念带有三个标注属性，其中只有一个是识别该概念所必需的。

树谓词视图还支持对随机森林进行更一般的推断，即识别“代价高昂”的对象。这里指的是那些分类过程需要进行大量（可能存在冗余的）阈值测试的对象。例如，请读者参见 NICL 中包含对象 1393、1397、1444 的概念。一方面，这些对象都需要对属性进行冗余测试，即 $maint{\geq}med$ 和 $maint{\geq}high$。另一方面，内涵的组成包含四个数据表属性，即 $buying$,$safety$,$maint$和 $persons$。

序数因子 一般而言，内序标度对人类读者来说更加复杂。这是因为在内序尺度化中，必须考虑一个阈值区间，而非单个阈值。例如，在图8（底部）中，我们可以看到概念 c，其外延为 686、1180、1444，并且分别是包含属性 $maint{\geq}med$ 和 $maint{\leq}high$ 的两个概念的下邻居。因此，人类读者必须在线性序 $maint$ 的阈值中考虑区间 $[med,high]$，该线性序为 low,med,$high$,$vhigh$。此外，c 具有属性 $buying{\leq}high$、$persons{\geq}four$和 $safety{\geq}med$。因此，人类读者必须同时理解不同的方向 顺序，即 $\geq$ 和 $\leq$。基于这一点，我们聚焦于图9中的 $\geq$-排序。当然，这限制了基于这些视图所能给出的解释的表达能力。然而，如上所示，这提高了可理解性。此外，这种方法通常会减少概念的数量。因此，我们可以在冰山概念格中采用较低的支持度值。图9展示了负类支持度值为一时的冰山概念格

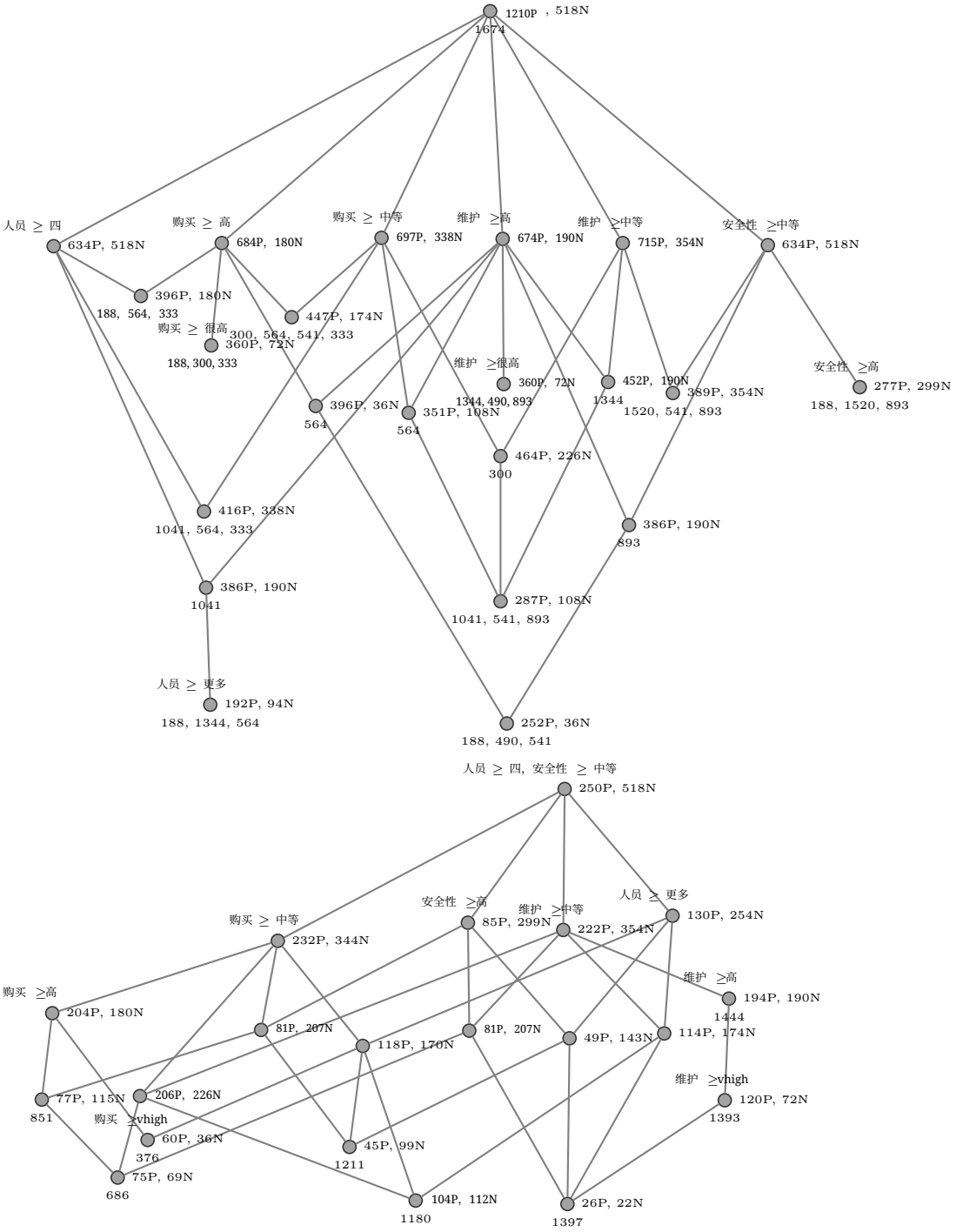


图9. 基于单棵树谓词视图的森林视图。我们将谓词限定为使用包含 ≥ 的表达式的谓词。上方的图展示了带有正类标签的中心对象对应的冰山概念格，下方的图则展示了负类标签对应的冰山概念格。

正类有三个。总体而言，这种方法可视为相对于背景运算并置 [15]的一种序数因子方法。除了上述优点外，还可以对图9中的图应用针对图8所说明的所有分析推导。

互序谓词视图 作为最后一个分析示例，我们展示随机森林的互序谓词视图的一个序数因子。如图9所示，我们选择 $\geq$ ，并采用相同的支持度参数。与基于树谓词视图的示例不同，图10所示的格编码了用于解释随机森林的另一类信息。更确切地说，互序谓词视图表示随机森林中对象与谓词之间的模型关系。

互序谓词视图的一个显著特点是，它反映了由属性阈值取值之间的序关系所蕴含的关系。例如，我们在图10中发现，对于数据表的$vhigh \to high \to$中等 购买属性，存在这种关系。此外，可以很容易地从图中读出相应的链。

此外，还可以从互序谓词视图中推断出数据表中不同属性的取值之间所有有效的属性蕴含。例如，在图10中可以发现，属性值 安全性$\geq$中等 蕴含维护$\geq$中等。虽然树谓词视图的冰山概念格中也存在一些蕴含，但需要注意的是，这些蕴含不一定是数据表中的蕴含。

在结束对互序谓词视图的分析时，我们想强调两个重要事实。第一，特定属性阈值是由训练过程（即随机森林）得出的，因此代表了训练后的分类器函数对数据的“视图”。因此，借助互序谓词视图揭示阈值蕴含时，我们实际上找到了通过随机森林的尺度化来观察数据表时成立的蕴含。第二，对于所有具有相同类别标签的数据对象而言，其所有有效蕴含的集合就是随机森林“所看到的”该类别的蕴含理论。因此，分别计算两个类别标签对应的蕴含理论后，就可以比较这两个理论的相似之处和差异。

讨论与展望 显然，我们的方法能够识别重要的属性阈值组合及其对分类结果的影响。当然，识别有趣且有意义的组合有多种组合方法。然而，所提出的概念方法的主要优势在于，它提供了一种结构化的数学方法，能够直接、高效地识别这些重要组合，同时给出其语义解释 [15]。

在此，我们想通过指出已开发的概念结构能够支持应用其他多种概念方法，来为本研究作结。例如，正如前文所述，

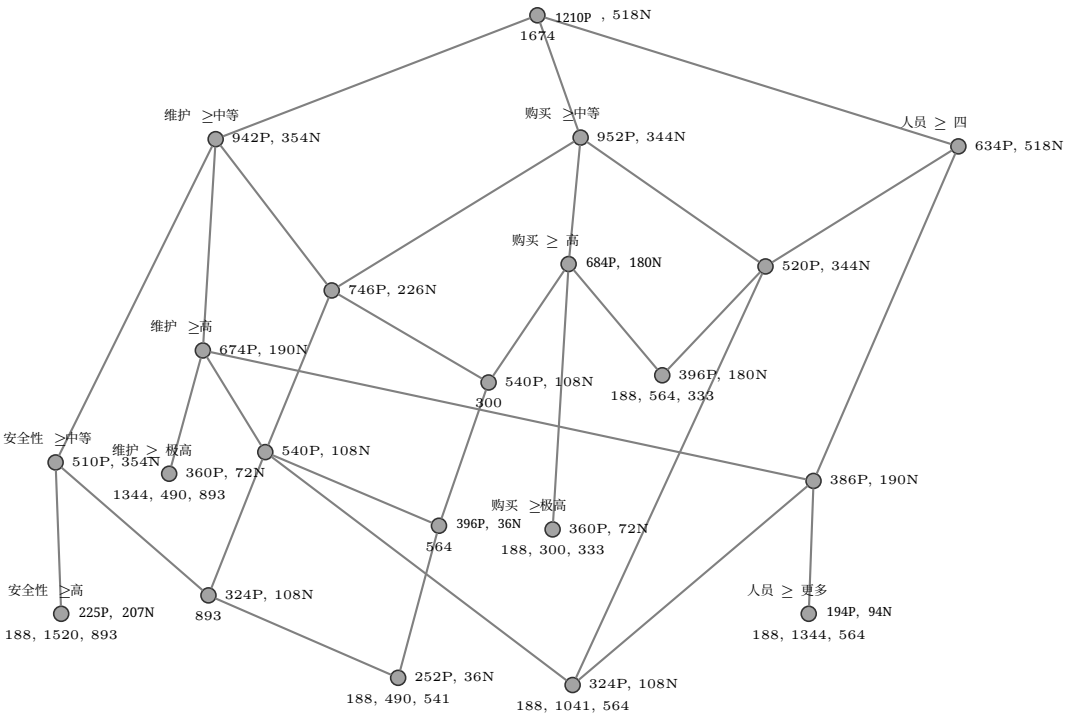


图10. 基于各个 interodinal谓词视图的森林视图。我们将谓词限制为使用包含 $\geq$ 的表达式的谓词。图中展示了与带有正类标签的中心对象相关的冰山概念格。

在本节中，分析类别内部属性阈值之间的蕴含结构，可以揭示关于随机森林的新见解。同样，概念视图可以用于比较经过训练的不同随机森林分类器，探究它们在蕴含关系上的差异与相似之处。对这些问题的详细研究将作为未来工作开展。

6 相关工作

有许多使用树和树集成的分类方法，例如决策树 [9], 随机森林[7],或决策树桩 [21], ，这里只举几个例子。单个决策树分类器最重要的特性是易于人工理解。例如，这类树的可视化有助于深入了解分类过程，同时也呈现了数据集的尺度视图。遗憾的是，迄今为止，后一种方法鲜少受到研究关注。针对这些尺度化的方法包括随机树嵌入（RandomTreesEmbedding），该方法由$sklearn$实现; 以及树视图 [11]。第一种方法根据将数据集对象分类的树叶节点，提取数据集对象的划分。然而，这种划分视图对数据集的缩放非常粗略，也很少利用树的层次结构。第二种方法通过概念格分析树的序结构。尽管作者提出了一种将树结构转换到形式概念分析领域的新颖方法，但并未详细说明如何利用这些结构进行解释。此外，他们仅反映了数据集上诱导出的序结构及其层次。这样做并未考虑所使用的树谓词，而这些谓词对于人工理解至关重要。尽管如此，这种转换方法本身很有价值，因为它使形式概念分析相关的后处理方法得以应用，进而生成解释。例如，尺度度量 [17],TITANIC [35], 核心结构 [16] 或重要性度量 [23]。

其他试图统一呈现树集成的方法，会通过合并将所有树组合成一棵新树 [34]。然而，这些方法有两个主要缺点。首先，它们的输出仍然是一棵树，而与格序不同，树中的每个节点只能沿线性路径展开。因此，这些方法无法处理缺失信息，也无法体现树集成的并发性。第二个缺点涉及输出结果的解释：合并后输出的树唯一的目标，是在数据集上产生与树集成相同的划分。然而，这忽略了树的内部表示。

另一条不同但相关的研究路线，是从概念格构造分类器 [3, 31]。例如，可以自上而下计算概念格，并以类别纯度作为停止准则 [4], ，然后从生成的偏序集中选取一棵树。尽管这些方法在数学上很优雅，但其表现不如随机森林等方法。

7 结论

我们正式介绍了关于树和树集成分类器的概念视图，这是一种用于分析并全面解释基于树的机器学习模型的新方法。为此，我们将决策树的划分谓词解释为形式背景的属性，并通过概念尺度化将其转化为概念结构。借助这一方法，用户可以深入了解基于树的分类函数。至少同样重要的是，现在还可以分析分类器对训练数据以及先前未知数据的视图。

具体而言，我们介绍了两种用于树集成概念化的新方法，即概念视图，并将其与先前的工作进行了比较。为此，我们证明了所提出的方法具有更高的粒度，因此在解释基于树的分类器时表达能力更强。我们的主要理论结果是，任何树集成都嵌入在所引入的树概念视图中。为了支持我们的理论建模和结果，我们开展了一项参数研究，探讨不同视图如何受到分类模型参数的影响。在此，我们根据决策树集合比较了形式概念。由于由此得到的概念视图可能规模庞大，我们引入了不同的尺度化和后处理方法，以保留树集成中相关知识的重要部分。第一种方法是逻辑尺度化，它有助于理解单个对象的分类，以及该对象的属性值发生变化时，其分类将如何变化。第二种尺度化方法则针对节点谓词为每个属性引入的阈值数量。我们阐述了如何利用概念视图减少这些阈值。总体而言，未来工作应进一步研究树集成分类器概念视图的尺度化。最后，我们指出，我们展示了该方法在真实数据集以及采用标准参数训练的随机森林上的适用性，即使用 100 棵树且不限制深度。尤其是，评估中使用的树的数量使我们的方法有别于以往基于形式概念分析的工作；后者关注的是个位数数量的小规模树集成。

在我们的工作中，我们并未关注树集成的技术细节，尤其是用于计算树和随机森林的超参数研究，因为我们的方法着重于解释给定的 森林，以了解已知数据以及可能未知的数据。尽管如此，深入研究超参数与由此生成的森林及其不同概念视图之间的关系，或许能让我们更深入地了解随机森林的训练过程。另一个我们未深入探讨的密切相关主题是：将所介绍的视图与分类器结合起来，可用于自动对多值数据表进行尺度化。此外，我们还可以设想将其用于枚举不同的决策树，因为这些树具有格序 [33]。此外，用于解释黑箱分类器函数的代理模型方法 [22] 也可能受益于基于我们概念视图的深入解释。基于多臂老虎机的方法 [1]亦是如此。总而言之，我们的工作

为多种监督式机器学习模型的概念分析打开了一扇门；未来关于可解释人工智能的研究应当继续探索这一方向。

参考文献

[1] T. Aldinucci 等。上下文决策树。2022。doi: 10.48550/ARXIV.2207. 06355。[2] A. Altmann 等。“置换重要性：一种经过校正的特征重要性度量。”载于：《生物信息学》26.10（2010），第1340–1347页。[3] R. Belohlávek 等。“刻画概念格中的树。”载于：《国际不确定性、模糊性与知识基础系统杂志》16.Supplement-1（2008），第1–15页。doi: 10.1142/S0218488508005212。[4] R. Belohlávek 等。“通过概念格诱导决策树。”载于：CLA。P. W. Eklund、J. Diatta 和 M. Liquiere 编。第331卷。CEUR研讨会论文集。CEUR-WS.org，2007。[5] B. Bischl 等。“OpenML基准测试套件。”载于：arXiv:1708.03731v*2* /stat.ML/（2019）。[6] H. Blockeel、L. D. Raedt 和 J. Ramon。“聚类树的自顶向下归纳。”载于：ArXiv cs.LG/0011032（1998）。[7] L. Breiman。“随机森林。”英文。载于：《机器学习》45.1（2001），第5–32页。issn: 0885-6125。doi: 10.1023/A:1010933404324。[8] L. Breiman。“随机森林。”载于：《机器学习》45.1（2001），第5–32页。[9] L. Breiman 等。《分类与回归树》。Wadsworth，2002年1月3日。isbn: 0-534-98053-8。[10] D. Dua 和 C. Graff。UCI机器学习知识库。2017。[11] E. Dudyrev 和 S. O. Kuznetsov。“决策概念格与决策树和随机森林的比较。”载于：国际形式概念分析会议。施普林格。2021，第252–260页。[12] D. Dürrschnabel、M. Koyda 和 G. Stumme。“使用反名义尺度进行属性选择。”载于：《基于图的表示与推理——第26届概念结构国际会议，ICCS 2021，线上活动，2021年9月20日至22日，论文集》。T. Braun 等编。第12879卷。计算机科学讲义。施普林格，2021，第127–141页。doi: 10.1007/978- 3- 030- 86982-3_10。[13] M. Feurer 等。“OpenML-Python：用于OpenML的可扩展Python API。”载于：arXiv 1911.02490（2020）。[14] J. H. Friedman。“随机梯度提升。”载于：《计算统计与数据分析》38.4（2002年2月），第367–378页。[15] B. Ganter 和 R. Wille。《形式概念分析：数学基础》。施普林格出版社，柏林，1999，第x+284页。[16] T. Hanika 和 J. Hirth。“大型形式背景中的知识核。”载于：《人工智能数学年刊》90.6（2022），第537–567页。doi: 10.1007/s10472-022-09790- 6。[17] T. Hanika 和 J. Hirth。“概念测量格。”载于：《信息科学》613（2022），第453–468页。doi: 10.1016/j.ins.2022.09.005。[18] T. Hanika 和 J. Hirth。“概念测量格。”载于：《信息科学》613（2022），第453–468页。doi: 10.1016/j.ins.2022.09.005。

[19] T. Hanika、M. Koyda 和 G. Stumme。“形式背景中的相关属性。”载于：基于图的表示与推理——第*24*届概念结构国际会议，*ICCS 2019*，德国马尔堡，*2019*年*7*月*1*日至*4*日，论文集。D. Endres、M. Alam 和 D. Sotropa 编。第11530卷。计算机科学讲义。施普林格，2019，第102–116页。doi: 10.1007/978- 3- 030- 23182-8_8。[20] T. Hastie 等。统计学习要素：数据挖掘、推断与预测。第2卷。施普林格，2009。[21] W. Iba 和 P. Langley。“单层决策树的归纳。”载于：机器学习论文集*1992*。爱思唯尔，1992，第233–240页。[22] J. Kim 和 S. Choi。“通过基于树的代理模型进行序贯模型优化中的不确定性估计。”载于：第*25*届人工智能与统计国际会议论文集。G. Camps-Valls、F. J. R. Ruiz 和 I. Valera 编。第151卷。机器学习研究论文集。PMLR，2022年3月28日至30日，第4359–4375页。[23] S. O. Kuznetsov 和 T. P. Makhalova。“形式概念的有趣度度量。”载于：信息科学442-443（2018），第202–219页。doi: 10.1016/j.ins.2018. 02.032。[24] S. O. Kuznetsov 和 T. P. Makhalova。“形式概念的有趣度度量。”载于：信息科学442-443（2018），第202–219页。[25] C. Lindig。“快速概念分析。”载于：概念结构研究——*ICCS 2000*论文集。Shaker Verlag，2000，第152–161页。[26] B. Liu、Y. Xia 和 P. S. Yu。“通过决策树构建进行聚类。”载于：信息与知识管理国际会议。2000。[27] T. M. Mitchell。机器学习。纽约州纽约市：McGraw-Hill，2010。[28] F. Moosmann、B. Triggs 和 F. Jurie。“利用随机聚类森林构建快速判别式视觉码本。”载于：神经信息处理系统进展。B. Schölkopf、J. Platt 和 T. Hoffman 编。第19卷。麻省理工学院出版社，2006。[29] S. Prediger 和 G. Stumme。“理论驱动的逻辑尺度化。”载于：第*6*届国际知识表示与数据库相遇研讨会论文集（*KRDB’99*）。E. F. 等编。第CEUR研讨会论文集21卷。另见：P. Lambrix 等（编）：国际描述逻辑研讨会论文集（DL’99）。CEUR研讨会论文集22，1999 http://ceur-ws.org/Vol-21。1999。[30] P. Probst、M. N. Wright 和 A.-L. Boulesteix。“随机森林的超参数与调优策略。”载于：《*Wiley*跨学科评论：数据挖掘与知识发现》9.3（2019），e1301。[31] O. Prokasheva、A. Onishchenko 和 S. Gurov。“基于形式概念分析的分类方法。”载于：*FCAIR 2012*——形式概念分析与信息检索相遇（2013），第95页。[32] J. R. Quinlan。*C4.5*：机器学习程序。Morgan Kaufmann，1993。isbn: 1-55860-238-0。[33] S. Ruggieri。“枚举不同的决策树。”载于：第*34*届国际机器学习会议论文集。D. Precup 和 Y. W. Teh 编。第70卷。机器学习研究论文集。PMLR，2017年6月，第2960–2968页。[34] P. Strecht。“决策树合并数据挖掘方法综述。”载于：第*10*届信息工程博士生研讨会论文集。2015，第36–47页。

[35] G. Stumme 等。“使用 TITANIC 计算冰山概念格。”载于：数据与知识工程42.2（2002 年 8 月），第 189–222 页。issn：0169-023X。doi：10.1016/S0169-023X(02)00057-5。[36] T. Vidal 和 M. Schiffer。“再生树集成。”载于：第 *37* 届国际机器学习会议论文集。H. D. III 和 A. Singh 编。第 119 卷。机器学习研究论文集。PMLR，2020 年 7 月 13–18 日，第 9743–9753 页。[37] R. Wille。“重构格论：一种基于概念层级的方法。”载于：有序集：北约高级研究院会议论文集。I. Rival 编。多德雷赫特：施普林格，1982 年，第 445–470 页。isbn：978-94-009-7798-3。[38] R. Wille。“作为概念与概念层级数学理论的形式概念分析。”载于：形式概念分析。B. Ganter、G. Stumme 和 R. Wille 编。第 3626 卷。计算机科学讲义。施普林格，2005 年，第 1–33 页。isbn：3-540-27891-5。