序数因子的贪心发现

Dominik D¨urrschnabel知识与数据工程组及信息系统设计跨学科研究中心，卡塞尔大学，德国卡塞尔
duerrschnabel@cs.uni-kassel.de
ORCID：0000-0002-0855-4185

Gerd Stumme知识与数据工程组及信息系统设计跨学科研究中心，卡塞尔大学，德国卡塞尔 stumme@cs.uni-kassel.de
ORCID：0000-0002-0570-7908

arXiv:2302.11554v1 [cs.LG] 19 Feb 2023

摘要——在大型数据集中，发现和分析结构十分困难。因此，为条目引入标签或关键词已十分常见。在应用中，人们会根据这些标签筛选此类数据集。即便如此，即使是只有少量标签的中型数据集，也会形成复杂且难以供人浏览的系统。本文采用序数因子分析方法来解决这一问题。序数因子依据标签的内在结构，将其中一个子集按线性序排列。由此类序数因子构成的完全序数因子分解能够精确表示原始数据集。基于这种序数因子分解，我们提出一种方法，用于发现并解释数据集中不同条目与属性之间的关系。然而，即使只计算一个基数较高的序数因子，计算复杂度也很高。因此，本文提出贪心算法 ORDIFIND。该算法利用形式概念分析中已有的快速算法提取序数因子。随后，我们借助 ORDIFIND 提出一种全面发现数据集内关系的方法。此外，我们还引入一种基于序数因子分解所得表示的距离度量，用于发现相似条目。为评估该方法，我们针对不同数据集开展了一项案例研究。

索引词—序数因子分析、序数数据科学、概念格

I. 引言

二元数据集由对象集和属性集以及关联关系构成。带有标签、关键词或类别的项目就是这类二元数据集的一个例子，其中项目是对象，标签是属性。典型的带标签数据包括带有类别或类型标签的电影或歌曲。此外，博客文章或科学文献等文章中的关键词也构成带标签的数据集。研究人员在不同会议上发表论文也是此类数据的一个例子。要基于这些标签或关键词理解此类数据集中的关系并从中提取知识，很快就会变得不可能；然而，这又是必不可少的。因此，提供能够从此类数据中提取解释并发现关系的系统，是一项重要的研究任务。

解决这一任务的一种常见方法是将二元属性视为数值型变量：若某个对象具有某项属性，则赋值为1，否则赋值为0。然后，采用经典降维工具中的方法，例如主成分分析。这些方法将不同的标签合并到少数几个轴中，同时对每个轴上的属性进行加权。因此，新形成的轴能够揭示原始数据集中相关特征的存在或缺失情况。接着，为每个对象在每个轴上赋予一个实数值，以表示该对象是否具有相应的属性。由于单个轴代表多个合并后的属性，因此，只具有部分属性的对象在最终布局中的位置会产生模糊的表示。由此，该方法的主要问题便显现出来。为对象赋予实数值，与底层二元数据的测量层次并不一致。所赋的数值会让人觉得，与其他对象相比，某个元素与某些属性的关联更强，而这种情况是不可能的。因此，在我们看来，鼓励此类比较并导致原始信息表示失真的方法并不可取。图1右侧展示了这种主成分分析投影的一个示例。

为了解决这一问题，Ganter 和 Glodeanu 在 [1]中提出了序数因子分析，这是一种不使用二元属性实数值测量的方法。其主要思想是将多个属性归纳为一个因子，与主成分分析类似。投影由属性的线性序构成，即所谓的序数因子。然后，该方法根据每个对象的属性，为其在每个因子中指定一个位置。与 PCA 方法相比，该过程中指定的位置是自然数，而非实数值。因此，所得投影不会表达不准确甚至错误的信息。此外，完全序数因子分解还允许推导出所有原始信息。与主成分分析投影类似，对于较小的数据集，也有一种可视化方法。具体而言，将每个对象放置在二维坐标系中各轴上满足其拥有截至该位置的所有属性的最末位置。图1将这种可视化与 PCA 投影进行了比较。与主成分分析相比，序数因子图的主要优点是其所有信息都能保证正确。例如，Facebook 和 Instagram 都有广告，且总部位于美国，这一点在纵轴上有所体现。另一方面，主成分分析投影给人的印象是

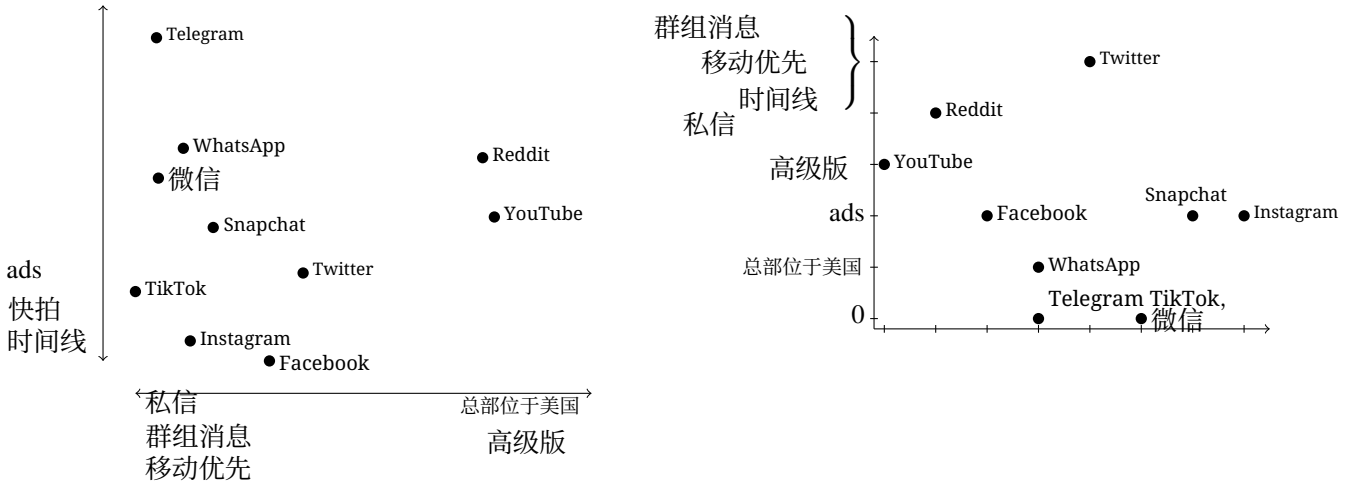

图 1。左：使用主成分分析对对象进行二维嵌入。右：为提高可读性，仅显示两个最大因子所对应的序数因子分解。除以下关联外，所有关联均可从右侧投影中推导出来：(TikTok，时间线)、(Whatsapp，快拍)、(Facebook，时间线)、(YouTube，快拍)、(Facebook，快拍)。序数投影不包含错误数据。

Instagram 的广告比 Snapchat 多，而且两者都不位于美国；这两种说法都是错误的。

然而，Ganter 和 Glodeanu 既没有提供计算序数因子分解的方法，其可视化方法也无法推广到更大数据集。不过，在我们看来，这种方法尤其适合处理更大数据集。尽管如此，迄今为止，针对这类数据集，既没有计算因子分解的方法，也没有可视化方法。此外，要应用该方法，仍有必要展示如何利用它来发现并解释这类大型数据集中的结构。我们目前的工作正是从这里切入。我们提出了 ORDIFIND，这是一种利用形式概念分析中的快速算法来计算数据集完全序数因子分解的贪心算法。此外，我们将现有的序数因子分析理论拓展为一种工具，用于解释大型二元数据集中的关系。随后，我们通过研究不同的数据集，展示如何利用序数因子分析从中发现知识。因此，我们的研究问题可归纳为：能否计算大型数据集的序数因子分解，并用它们来发现数据中的关系？

II. 相关工作

我们从三个不同领域讨论与本主题相关的工作。首先，我们讨论生成二元数据的一般标注系统。然后，我们考察以低维度表示属性的方法。在最后一节中，我们讨论形式概念分析方面的工作；我们的工作正是定位于这一领域。

A. 标注、类别与关键词

在现代世界中，数据往往以大量、无序的形式出现。因此，人们会为项目分配标签、类别或关键词，以应对数据量庞大的问题。这些标记使项目能够被筛选。这些概念之间的界限十分模糊，因为它们常常可以互换使用。社区通常会为项目定义标签，这一过程通常称为社交标注。 [2]中发表了一篇关于社交标签技术的全面综述。 [3] 是一项探究标注背后原因的研究。在[4], 中，研究者分析并比较了用户创建的各种类型的标签。 [5] 介绍了一种从论文语料库中自动提取主题的方法。 [6]探讨了计算机视觉领域中的图像自动标注。

B. 二元数据的嵌入方法

将高维数据嵌入较低维度的研究主要有以下两个原因。首先，嵌入可用于摆脱所谓的维数灾难 [7], ，这种现象会导致距离在更高维的空间中变得不那么有意义。其次，这为更好地可视化 [8] 复杂数据提供了一种方式，因为人类更适合理解低维空间中的关系。关于降维方法的全面综述，请参阅 [9]。一种常用于将高维数据嵌入低维空间的方法是主成分分析 [10], ，这是一种将数据点到一条直线的平均平方距离最小化的方法。它常与探索性因子分析 [12], 混淆 [11] ；后者是一种通过重构潜在因子来探索数据集的技术。

C. 形式概念分析中的因子分析

形式概念分析是由 Wille 于 1982 [13] 提出的一门数学数据分析研究领域。关于形式概念分析理论及其基础的全面概述，请参阅 [14]。形式概念分析中描述信息的规范方式是概念格。不过，有些方法会以不同方式组织数据，使其更易于理解。其中一种是布尔因子分析，于 [15]–[19] 年发展起来。在 [1] 中介绍了

序数因子分析。 [20] 展示了该方法的应用，并将其用于一些较小的医学数据集。 [21] 将该理论推广到三元关联关系的情形， [22] 则将其推广到模糊形式背景的情形。

III. 序数因子分析基础

形式概念分析是序数因子分析的基础，因此本节先简要介绍形式概念分析。接着，我们介绍布尔因子分析，它是序数因子分析的基础。本节用于阐释理论的贯穿示例，是图2中的数据集，内容涉及各种社交媒体平台的不同特征和属性。

A. 形式概念分析

本文使用的符号是根据 [14] 改编的标准符号。形式背景是一个三元组 (G, M, I)，由对象集 G 和属性集 M 以及关联关系 $I \subseteq G \times M$ 构成。对于任意子集 $A \subseteq G$ 和 $B \subseteq M$，定义两个导出算子如下：

$$A' := \{m \in M \mid \forall g \in A : (g, m) \in I\},$$
$$B' := \{g \in G \mid \forall m \in B : (g, m) \in I\}.$$

对于单个属性 m 或对象 g，简记为 $m' := \{m\}'$ 和 $g' := \{g\}'$。满足 $A' = B$ 和 $B' = A$，且 $A \subseteq G$ 和 $B \subseteq M$ 的一对 (A, B) 称为 (G, M, I) 的形式概念。偏序关系

$$(A_1, B_1) \leq (A_2, B_2) :\iff A_1 \subseteq A_2 \quad (\iff B_1 \supseteq B_2).$$

对语境中的所有形式概念进行排序。所有概念及其序关系所构成的集合记为 $(\mathfrak{B}, \leq)$。其中，任意两个概念都可比较的概念子集称为链。该序具有格结构，即对于每个

	USA-based	premium	ads	private messages	group messages	mobile first	stories	timeline
Facebook	×		×	×	×		×	×
Instagram	×		×	×	×	×	×	×
Reddit	×	×	×	×				
Snapchat	×		×	×	×	×	×	
Telegram				×	×	×		
TikTok			×	×	×	×	×	×
Twitter	×	×	×	×	×	×		×
微信			×	×	×	×	×	
WhatsApp	×			×	×	×	×	
YouTube	×	×	×				×	

图2。运行示例：此数据集比较了不同社交媒体平台的属性。

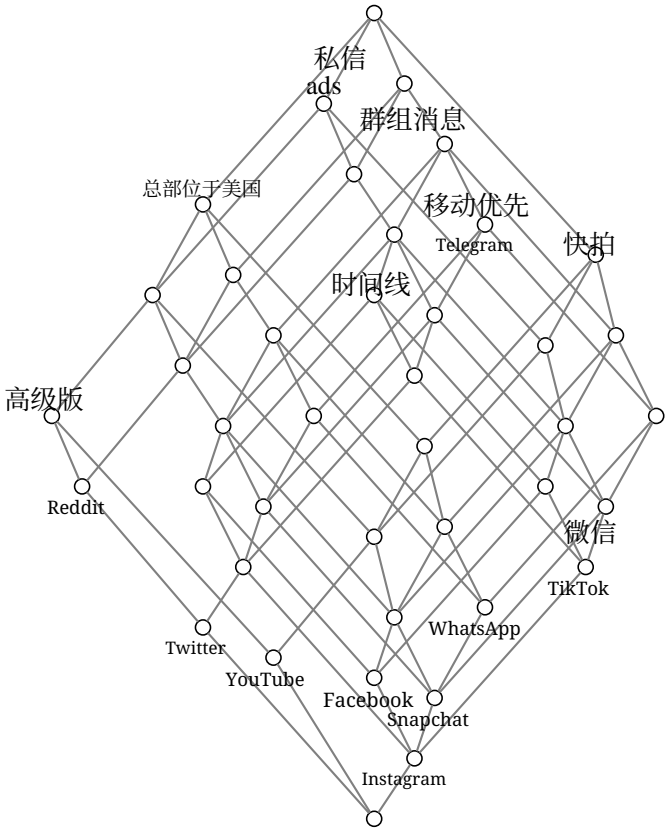

图 3. 运行示例的概念格序图展示了其中的 41 个概念。

每一对概念都存在唯一的上确界和唯一的下确界。由此形成的所谓概念格被用于形式概念分析，以分析形式背景中的结构和关系。若两个不同的概念（A_1, B_1）和（A_2, B_2）满足覆盖关系，即（A_1, B_1） $\leq$ （A_2, B_2），且不存在其他概念（A_3, B_3）使得（A_1, B_1） $\leq$ （A_3, B_3） $\leq$（A_2, B_2），则称它们满足覆盖关系。此时记作（A_1, B_1） $\prec$ （A_2, B_2）。序图以图形方式展示概念格。在该图中，平面上的节点表示概念，较小的概念节点位于较大的概念节点下方。满足覆盖关系的概念对由连续直线连接。包含某一特定属性的最大概念以该属性名称标注；与之对称，包含某一对象的最小概念以该对象名称标注。属性标注在概念节点上方，对象标签则标注在节点下方。通过这种构造，可以从概念格图中重建形式背景的全部信息。图 3 展示了本例中的概念格。

B. 布尔因子分析

布尔因子分析的研究既在形式概念分析的范围内进行，也在其范围之外进行。本文采用形式概念分析中使用的记号。一个形式背景 $\mathbb{K} = (G, M, I)$

	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15	f16	f17
1				×			×		×		×	×		×	×	×	×
2	×		×	×			×	×	×	×	×	×	×	×	×	×	×
3					×	×						×			×		×
4			×					×	×	×	×	×	×	×	×	×	×
5													×			×	×
6							×	×	×	×	×		×	×		×	×
7		×			×	×				×		×	×		×	×	×
8								×	×	×	×		×	×		×	×
9													×	×	×	×	×
10						×					×	×		×	×		

	a	b	c	d	e	f	g	h
f1	×		×	×	×	×	×	×
f2	×	×	×	×	×	×		×
f3	×		×	×	×	×	×	
f4	×		×	×	×		×	×
f5	×	×	×	×				
f6	×	×	×					
f7			×	×	×		×	×
f8			×	×	×	×	×	
f9			×	×	×		×	
f10			×	×	×	×		
f11			×				×	
f12	×		×					
f13				×	×	×		
f14							×	
f15	×							
f16				×	×			
f17				×				

对象: (1) Facebook (2) Instagram (3) Reddit (4) Snapchat (5) Telegram (6) TikTok (7) Twitter (8) 微信 (9) WhatsApp (10) YouTube

属性: (a) 总部位于美国 (b) 高级版 (c) 广告 (d) 私信 (e) 群组消息 (f) 移动优先 (g) 快拍 (h) 时间线

图4. 使用 17 个布尔因子对图2中的数据集进行概念布尔因子分解。

可以分解为两个因式分解形式背景 (G, F, I_{GF}) 和 (F, M, I_{FM})，使得

$$(g, m) \in I \iff (g, f) \in I_{GF} \text{ and } (f, m) \in I_{FM}$$

对于某个 $f \in F$。F中的元素称为布尔因子。对于每个因子 $f \in F$，令因子化族为有序对 (A, B)，其中 $A = \{g \mid (g, f) \in I_{GF}\}$且 $B = \{m \mid (f, m) \in I_{FM}\}$。那么，对于每个对象 $g \in A$ 和每个属性 $m \in B$，都有 $(g, m) \in I$。因此，扩展每个因子，使其因子化族成为一个形式概念，既合理也可行。由此得到的因子分解称为概念因式分解。由少量因子构成的布尔因子分解具有两项优势：它能够压缩数据集的规模，并提高数据集的可理解性。然而，形式背景并不一定存在规模较小的因子分解。相反，对于给定的 k，判断是否存在由 k个因子组成的因子分解是一项 NP-完全问题 [18]。

对于我们的贯穿示例，如图4所示，存在一种使用17个因子的概念分解。

C. 序数因子分析

为克服大型布尔因子分解的缺点，Ganter 和 Glodeanu 在 [1] 中提出如下方法：将多个布尔因子归并为一个因子。对于给定的形式背景 $\mathbb{K} = (G, M, I)$，若有一对因子化背景 (G, F, I_{GF}) 和 (F, M, I_{FM})，则集合 $E \subseteq F$称为 (G, M, I) 的一个多值因子。接着，将序数因子定义为一个多值序数因子 E，其中对于所有元素 $e_1, e_2 \in E$，若其因子化族为 (A_1, B_1) 和 (A_2, B_2)，则满足 $A_1 \subseteq A_2$ 或 $A_2 \subseteq A_1$。形式背景 $\mathbb{K} = (G, M, I)$ 的宽度为 k 的完全序数因子分解，是由序数因子 $F_1, \ldots F_k$ 组成的集合，且满足 $\bigcup_{i=1}^{k} F_i = I$。与布尔因子的情况类似，每个序数因子都可以扩展为一个形式概念链。这种序数因子分解称为概念序数因子分解。

序因子与费雷尔关系密切相关。在背景 (G, M, I) 中，费雷尔关系 F 是 $G \times M$ 的一个子集，其中对于所有 $g, h \in G$ 和 $m, n \in M$，均有

$$(g, m) \in I \wedge (h, n) \in I \Rightarrow (g, n) \in I \vee (h, m) \in I.$$

如果形式背景（G，M，I）中的 Ferrers 关系 F 满足 $F \subseteq I$，且不存在满足 $F \subsetneq F'$ 的 Ferrers 关系 $F' \subseteq I$，我们就称其为极大的。极大 Ferrers 关系与概念格的极大链一一对应，后者描述了极大的概念序数因子。基于这种对偶性，我们将极大 Ferrers 关系和形式概念的极大链都称为极大序数因子。序数因子的大小是指其 Ferrers 关系中所包含的关联数量。

再次考虑这个运行示例。由于这三个元素中任意两个都无法同时出现在同一个费雷尔关系中，因此无法用两个因子进行因子分解:

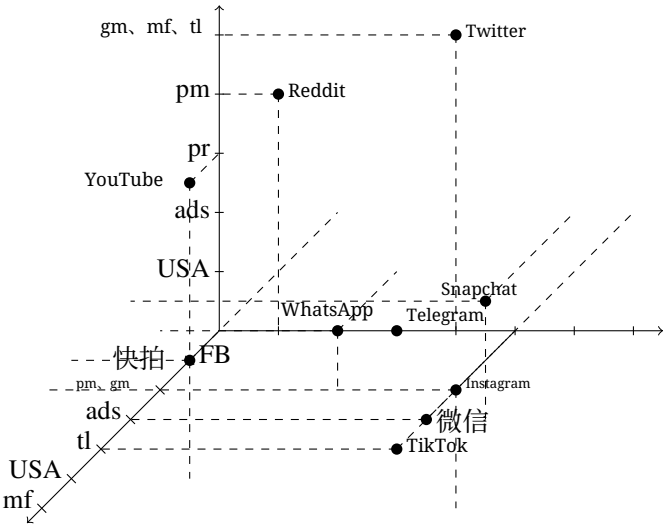

图5. 将图2中的数据集完全序数因子分解为三个因子。

（YouTube，高级版）、（WhatsApp，mobile- first）和（Facebook，时间线）。另一方面，可以按如下方式生成由三个因子组成的完全序数因子分解：

1) $f17 < f16 < f13 < f10 < f8 < f3 < f1$
2) $f15 < f12 < f6 < f5 < f2$
3) $f14 < f11 < f9 < f7 < f4 < f1$

请注意，图4中的概念布尔因子分解得到的每个因子都至少出现在一个序数因子中，因此这是一个完全序数因子分解。

Ganter 和 Glodeau 提出了一种坐标系，用于通过序数因子分解来展示数据集的信息。该图中的每条轴都表示一个序数因子，具体如下。每条轴上的刻度都对应概念链中的一个概念。刻度标签包含该概念相较于前一个刻度新增的属性。图中对象在每条轴上的位置，取为其不具备某个属性时，该属性首次出现的刻度之前一个刻度。对于当前示例中的给定因子分解，完全因子分解将产生一个三维坐标系。正如图1所示，在平面上展示包含三个序数因子的复杂数据集，会产生难以阅读的投影。因此，将此类图限制为两个最大的序数因子是合理的。然而，采用这种方法会丢失第三个因子的信息。图1比较了序数因子分析图与主成分分析图。

IV. 贪心序数因子分解是困难的

在接下来的两节中，我们将介绍并研究贪心序数因子分解。

定义 IV.1. 设 (G, M, I) 为一个形式背景。若对于每个 $i \in \{1, \ldots, k\}$，都不存在满足 $|F_i \setminus \{F_1 \cup \cdots \cup F_{i-1}\}| > |F_i \setminus \{F_1 \cup \cdots \cup F_{i-1}\}|$ 的因子 F_i，则称序数因子集合 $F_1, \ldots, F_k$ 为贪心的。

反复计算覆盖关联中尽可能大的未覆盖部分的 Ferrer 关系，会得到这样的因子分解。通过重复这一过程，直到关联关系的每个部分都被覆盖，即可得到完全序数因子分解。然而，事实证明，即使只是确定这一贪心过程中的第一个序数因子的大小，也是一个 NP-完全问题。为此，请看 Yannakakis [24] 的一个辅助引理。该引理将与 Ferrer 关系对应的二部图称为链图。本文改编后的版本如下。

引理 IV.1. 设 $\mathbb{K} = (G, M, I)$ 是一个形式背景，k 是一个自然数。判定是否存在一个大小为 k 的集合 $I \subseteq G \times M$，使得 $I \cup I$ 是 Ferrers 关系，是 NP-完全的，其中 ~ ~。

费雷尔关系的补关系仍然是费雷尔关系。因此，判断需要从二元关系中删除多少对才能使其成为费雷尔关系，在计算复杂性上也同样困难。

算法 1 极大 Ferrers 关系

输入：带有覆盖关系 $(\mathfrak{B}, \prec)$ 的概念格
已覆盖的 E
输出：极大 Ferrers 关系 F

$F_{(A,B)} := \emptyset \ \forall (A, B) \in \mathfrak{B}$
$L := linear_extension(\mathfrak{B}, \prec)$
对于 (A_1, B_1)，位于 L：
  对于 $(A_2, B_2) \in \mathfrak{B}$ 使用 $(A_2, B_2) \prec (A_1, B_1)$：
    $F_{(A_1,B_1)} = F_{(A_2,B_2)} \cup (A_1 \times B_1)$
    if $|F_{(A_1,B_1)} \setminus E| \geq |F_{(A_1,B_1)} \setminus E|$:
      $F_{(A_1,B_1)} = F_{(A_1,B_1)}$
return $F_{\max(\mathfrak{B}, \prec)}$

推论 IV.1.1. 设 $\mathbb{K} = (G, M, I)$ 是一个形式背景，且 k 是一个自然数。判断是否存在一个大小为 k 的集合 $I \subseteq I$，使得 $I \setminus I$ 是费雷尔关系，是 NP-完全的，其中 ~ ~。

证明。为此，只需证明一个费雷尔关系的补关系仍然是一个费雷尔关系。反设不然，即存在一个费雷尔关系 $F \subseteq G \times M$，其补关系 $(G \times M) \setminus F$ 不是费雷尔关系。那么必定存在 $g, h \in G$ 和 $m, n \in M$，使得 $(g, m) \notin F$ 且 $(h, n) \notin F$，但 $(g, n) \in F$ 且 $(h, m) \in F$，这与 F 是费雷尔关系矛盾。现在设 (G, M, I) 是一个上下文，且 ~ $I \subseteq G \times M$。那么 $(G \times M) \setminus (I \cup I)$ 是费雷尔关系，当且仅当 $I \setminus I$ 也是费雷尔关系，由此证明得证。

□

因此，即使只是判断第一个贪心因子的大小是否为 k，也是一项 NP-完全任务，这使得计算贪心序数因子分解的问题是 NP-困难的。

V. 计算贪心序数分解

本节介绍 ORDIFIND（二元数据序数因子算法），这是一种用于计算贪心序数因子分解的算法。如上一节所述，这个问题的计算复杂度很高。因此，我们所能找到的最佳算法仍然需要相对于输入规模的指数时间。不过，我们可以利用形式概念分析中的快速概念格算法。尤其是 Lindig 算法 [25]，它可以配合加速技术使用 [26]，非常有用。该算法能在合理的时间内计算形式背景的覆盖关系。这种方法的主要优势在于，计算概念格这一指数时间任务只需执行一次。此后，算法的时间复杂度相对于概念格覆盖关系的规模为线性。请注意，该覆盖关系的规模仍可能相对于输入数据规模呈指数增长。

算法 1 中描述的例程为 ORDIFIND 奠定了基础。它以概念格的覆盖关系和集合 E 作为输入。随后，该例程计算一个能够覆盖 $I \setminus E$ 中最多关联的序数因子。因此，该例程的输出是一个序数因子 F，

其费雷尔关系形式满足：不存在序数因子~ ~F使得 $|F \setminus E| > |F \setminus E|$。该例程的主要思想如下。形式背景的概念格按照线性扩张的顺序从底部迭代到顶部，即按照与概念格相容的线性序进行迭代。因此，在所有较小概念的相同处理过程完成之前，不会处理某个概念。每处理完一个概念，覆盖 $I \setminus E$ 中与该特定概念（A、B）相关的最多关联、并以其为顶概念的序数因子，就会存储在变量 $F_{(A,B)}$ 中。因此，在处理完每个元素后，该变量对应于顶概念 $F_{\max(\mathfrak{B},\prec)}$，其中包含的因子会从 $I \setminus E$ 中排除关联数量最多的部分。对于一组已经计算出的因子，可以将这些因子的并集选作 E，从而计算出包含最多未覆盖关联的序数因子。反复执行这一过程，最终便可得到完整的形式背景贪心因子分解，如算法 2 所述。

因此，可以得到以下关于该算法总体运行时间的引理。

引理 V.1。对于包含 k 个概念且给定覆盖关系的形式背景，可以在 $O(rk^2)$ 内计算出由 r 个序数因子组成的贪心序数因子分解。

证明。算法 1 的运行时间上界由覆盖关系的大小决定。覆盖关系的大小上界由概念对的数量决定。算法 2 中的循环重复执行 r 次。 □

上述算法计算出一个贪心序数因子分解。为了评估该算法的实用性，关键在于考察该因子分解的优劣，即将其性能与使因子数量最少的序数因子分解进行比较。我们将在下面的引理中对此进行研究。

引理 V.2。设 $\mathbb{K} = (G, M, I)$ 可被最优分解为 k 个序数因子。设 $F_1, \ldots, F_r$ 为一个贪心序数因子分解。则 $r \leq k \log_e |I|$。

证明。根据鸽巢原理，对于所有 $i \in \mathbb{N}$，都有 $|F_i| \geq (|I| - |F_1 \cup \ldots \cup F_{i-1}|)/k$。因此，对于 $r > k \log_e |I|$，

算法 2：朴素完全序数分解

输入：形式背景（G, M, I）
覆盖关系（$\mathfrak{B}, \prec$）
输出：贪心序数因子分解 $F_1, \ldots, F_k$

$E = \emptyset$
$i = 1$
当 $E \neq I$ 时：
  $F_i = max_ferrers((\mathfrak{B}, \prec)\ E)$
  $E = E \cup F_i$
  $i = i + 1$
返回 $F_1, \ldots, F_i$

有 $|I| - |F_1| - \cdots - |F_r| \leq |I|(1 - \frac{1}{k})^r \leq |I|e^{-r/k} < 1$，这意味着经过 r 次重复后，所有关联都已覆盖。 □

算法1和算法2中提出的朴素算法，在每一步计算新因子时都会重新处理每个概念。然而，尤其是在后期较小的因子中，该算法并不会修改格中的所有概念。识别出算法未涉及的概念后，便可以利用一种加速技巧：在各步骤之间保存这些概念的值。为此，我们必须修改朴素算法存储已计算信息的方式。我们不再为格中的每个概念保存序数因子的所有关联，而只保存其大小。此外，还要为因子中的每个概念保存其前驱。随后，从顶概念开始沿前驱回溯，即可提取该因子。现在，并非每个因子都包含所有属性。因此，在计算出一个缺少某些属性的因子后，只需重新计算那些具有包含属性的更大概念的概念。由此，加速效果便来自仅重新计算这些概念。为了确定需要重新计算哪些概念，我们将因子的属性存入某个集合R。此外，我们使用集合 R 来确定接下来要遍历哪些概念。我们不再按照预先固定的线性扩张枚举概念，而是始终选择 R 中属性集最小的概念作为下一个遍历对象，然后将其从 R中移除。为了支持这种提取方式，我们将 R 实现为一个堆。较小概念的属性集大小会增加，因此仍能遵循与概念格相容的线性序。本段提出的算法 ORDIFIND（算法 3）采用这些新思路来加速前两个算法。

VI. 因子与对象之间的关系

设（G，M，I）是一个形式背景，其序数因子分解为 F_1，$\ldots F_k$。对于一个固定因子 $F_i =$（A_1，B_1）.....，（A_j，B_j）和一个对象 g，称 g 在 F_i 中的位置为满足 $B_r \subseteq g'$ 的最大 r。由于 F_i 是一个概念链，因此概念 $B_1 \ldots B_r$ 的所有属性集都是 g' 的子集。若此条件成立，则称对象 g 支持因子 F_i，直至位置 r。若 $r = j$，则对象 g 支持整个因子 F_i。

对于数据集中的两个对象 g_1、$g_2 \in G$，现在可以将序数因子分解距离定义为

$$d(g_1, g_2) = |g_1' \setminus g_2'|.$$

因此，两个对象之间的这种距离统计了一个对象拥有而另一个对象缺少的属性数量。请注意，这种距离并不对称，并且 f $g_1' \subseteq g_2'$ 时它变为 0。汉明距离是上述距离的对称版本，常用于计算二元数据集中的距离，其定义如下：

$$d_h(g_1, g_2) = d(g_1, g_2) + d(g_2, g_1).$$

我们可以将序数因子分解距离的概念推广到因子中的位置。为此，我们根据这些位置计算因子分解中的属性数量

算法 3 ORDIFIND

输入：形式背景 (G, M, I)
覆盖关系 $(\mathfrak{B}, \prec)$
输出：贪心序数因子分解 $F_1, \ldots, F_k$

$E = \emptyset$
$i = 1$
$R =$最小值$(\mathfrak{B}\ \prec)$
当 $E \neq I$:
$b_{(A,B)} := \perp\ \forall (A, B) \in \mathfrak{B}$
$s_{(A,B)} := \emptyset\ \forall (A, B) \in \mathfrak{B}$
当...时 $R \neq \emptyset$:
$(A_1, B_1) = R.pop_concept_min_no_attr()$
对 $(A_2, B_2) \in \mathfrak{B}$ 使用 $(A_1, B_1) \prec (A_2, B_2)$:
$\widetilde{s_{(A_2,B_2)}} = s_{(A_1,B_1)} + |(A_2 \times B_2) \setminus E|$
if $\widetilde{s_{(A_2,B_2)}} > s_{(A_2,B_2)}$:
$s_{(A_2,B_2)} = \widetilde{s_{(A_2,B_2)}}$
$b_{(A_2,B_2)} = (A_1, B_1)$
$R = R \cup \{(A_2, B_2)\}$
$F_i := \emptyset$
$b = \max(\mathfrak{B}\ \prec)$
当 $b \neq \perp$:
$F_i = F_i \cup \{(A, B)\}$
$b = b_{(A,B)}$
$R = \{(g'', g') \mid g \in B, (A, B) \in F_i\}$
$i = i + 1$
return $F_1, \ldots\ldots, F_i$

然后计算对象缺失的属性数量。因此，设 (G, M, I) 为一个形式背景，其因子分解为 $F_1, \ldots, F_k$，且 $p_1, \ldots, p_k$ 为表示每个因子中位置的整数。将对象 g 到位置 $p_1, \ldots, p_k$ 的序数因子分解距离定义为

$$d_F(g, p_1, \ldots, p_k) = |\{m \mid m \in F_i \text{ before position } p_i\} \setminus g'|.$$

因此，该距离统计了一个项目缺少支持所有因子的属性数量，直到位置 F_i，直到位置 p_i。请注意，对象的序数因子分解距离可能为 0。当 $p_1 = \cdots = p_k = 0$ 时，这显然成立。

VII. 将序数因子分解用作发现结构的工具

我们将序数因子分析视为一种用于解释和发现无序数据集中关系的工具。计算因子分解这一计算开销较大的任务只需在预处理步骤中执行一次。为了让该工具易于使用，我们设想构建一个网络平台，只需相对轻量地计算对象在序因子中的位置。用户随后便可交互式浏览数据集，发现数据中已有的关系。为此，我们提出一种由滑动控件组成的导航系统。每个滑动控件对应一个因子。对于包含 k 个因子的序数因子分解，该系统设有 k 个滑动控件。对于一个因子中的每个概念，都有一个位置与相应的滑动控件相对应。由于一个因子的顶端和底端概念通常具有空对象集或空属性集，因此它们在滑动控件中没有对应位置。此外，每个滑动控件还设有一个 0 位置，用于表示未选择任何内容。每个位置都标注了相较于该因子中前一个概念，相应概念新增的属性。分析人员随后可以使用滑块为各个滑动控件选择属性位置。系统接着会按照对象到所选位置的序因子分解距离，对对象进行排序。该平台的第二项功能是，分析人员可以点击对象来选中它。随后，系统会自动将每个因子的滑块移至所选对象所支持的最高位置。剩余对象则根据所选位置的序因子分解距离重新排序。关于此类导航系统可能的运作方式，读者可参阅我们的演示平台[1]原型。我们使用本文提出的贪心因子分析计算了该工具的序数因子分解。该平台的源代码或可作为未来开发更复杂探索工具的基础。

VIII. 案例研究

现在，我们通过五个具有代表性的数据集说明，序数因子分析是一种可用于发现数据集中关系的方法。

A. 源代码

我们的算法使用 Python 3 实现。改进版 Lindig 算法 [25], [26] 能够直接快速地计算出我们的方法所需的概念格覆盖关系。因此，我们使用该方法计算概念格。尽管我们精心实现了算法代码，但其重点在于支持实验，而非追求速度。因此，我们认为，通过优化代码有可能提升运行速度。源代码[2] 可供审查并复现研究结果。

B. 数据集

我们在来自多个来源的多个数据集上进行实验，并在本节中加以介绍。

a) 德国人工智能和通用人工智能：我们基于 Koopmann 等人的一项研究生成了这份文献计量数据 [27]。它包含人工智能领域作者与会议之间的发表关系。他们根据 Kersting 的一篇文章选择了这份数据中的会议 [28]。我们从这些数据中生成了两个不同的数据集。第一个数据集包含仅限德国机构隶属关系作者的人工智能社群，另一个则不受此限制。我们将这两个数据集分别称为德国人工智能和通用人工智能。

[1] https://factoranalysis.github.io/ordinal/ [2] https://figshare.com/s/8edc72eaebc53e74f146

表 I 数据集描述

	德国人工智能	通用人工智能	IMDb	Wahl-O-Mat政党	Wahl-O-Mat陈述
属性	133	137	28	38	76
对象	2238	218308	3784644	76	38
概念	12709	1981691	6765616	24245	24245
密度	0.0258	0.0149	0.0680	0.429	0.429
每个概念的平均属性数	4.87	7.24	11.52	7.26	8.28
每个概念的平均对象数	7.35	13.54	792.78	8.28	7.26
覆盖关系大小	46776	11455411	63990855	24245	24245
贪心序数因子的数量	129	137	28	20	20

b) IMDb：我们从 IMDb 生成了这个数据集。IMDb 是一个包含电影和电视剧信息的数据库。电影和电视剧是该数据集中的对象，其类别则是属性。我们使用截至 10 月 6 日 2021 [29] 的数据，并将这个数据集称为 IMDb 数据集。

c) Wahl-O-Mat：Wahl-O-Mat 是由德国“联邦公民教育中心”¨提供的一个网站。 [30] 它是一种帮助德国公民决定在联邦选举中投票支持哪个政党的工具。网站会提供一组陈述，公民需要判断自己是否赞同这些陈述。随后，系统会将结果与不同政党的立场进行比较，以推荐可能符合该公民观点的政党。我们提取了2021年选举的陈述和政党信息，创建了两个数据集。第一个数据集称为 Wahl-O-Mat政党数据集，其中政党是对象，陈述是属性。Wahl-O-Mat陈述数据集则以陈述为对象、政党为属性。

C. 运行时间

所有计算均在配备 800 GB 内存的英特尔至强金牌 5122 处理器上完成。对于每个数据集，我们都使用本文的方法计算完全序数因子分解。表 II 对比了 ORDIFIND 与朴素算法的运行时间，同时列出了计算概念格所需的运行时间。可以看出，在几乎所有实例中，计算覆盖关系都耗费了相当多的时间。显然，ORDIFIND 的表现优于朴素算法。我们无法使用朴素算法计算 IMDb 数据集的因子分解。总而言之，我们认为，对于能够计算出概念格的所有数据集，都可以应用本文提出的方法。

D. 有趣的发现

本节将展示如何运用该方法发现关系，并讨论我们在数据集中得到的发现。

a) 德国人工智能与通用人工智能：对德国人工智能和通用人工智能数据集进行贪心分解，分别得到 129 个和 137 个因子。对于德国人工智能数据集，我们展示了图 6 中的因子；第 VII 节介绍的原型网络平台则为通用人工智能社群提供了启示。将这些单独的因子解读为会议社群似乎是合理的，这些社群中的作者兴趣相近，并在其中发表论文。例如，两种情况下的第一个因子都由机器人学会议构成。因此，德国人工智能数据集的第二个因子对应计算机视觉社群，而第三个因子则包含通用人工智能会议。由于序数因子分解的特性，在此类因子中较早出现的会议更为综合，而深入因子之中的会议则更为专业化。因此，这种序数因子分析可以为作者提供选择会议的工具。值得注意的是，用这种方法发现的社群大多与 Kersting 在 [28] 中阐述的人工智能社群相吻合。

b) IMDb：IMDb 电影数据集的完整贪心分解包含 27 个因子。每个因子都对应一种类型的线性序。我们可以通过以下方式利用这些因子了解电影领域。第一个因子似乎侧重于轻娱乐。由于大多数电影制作都归属于这个因子，因此它对行业最为重要。第二个因子则更具教育导向。第三个因子再次涵盖了娱乐节目。不过，这个因子中的娱乐内容不如第一个因子那么轻松。这种因子分解很有价值，因为它可用于浏览电影数据库，或任何具有类似类型系统的数据库。此外，我们认为可以基于这些因子，将现有方法拓展为更复杂的推荐范式。届时，用户可以选择例如想看一部娱乐性很强、但也带有一些教育意义的电影。然后，系统便会根据序数因子距离推荐电影。

c) Wahl-O-Mat：我们的研究采用了两种不同的方式来解读 Wahl-O-Mat 数据集。原因在于，

表 II 算法运行时间

	概念	朴素算法	OrdiFIND
通用人工智能	4.63 小时	4.18小时	3.57小时
德国人工智能	3.76秒	33.54秒	12.97秒
IMDb	4.46天	-	14.42天
Wahl-O-Mat政党	4.34秒	16.92秒	11.96秒
Wahl-O-Mat陈述	8.18秒	14.23秒	13.36秒

1) IEEE机器人与自动化国际会议 - IEEE/RSJ智能机器人与系统国际会议 - 《国际机器人研究杂志》 - 《IEEE机器人学汇刊》 - 《机器人与自主系统》 - 《自主机器人》 - 《IEEE机器人与自动化快报》 - 《IEEE机器人与自动化杂志》 - 《野外机器人学杂志》 - 机器人：科学与系统 - 野外与服务机器人会议 - 国际机器人研究研讨会 - 人工智能促进协会 - 国际人工智能联合会议 - 神经信息处理系统大会，国际机器学习大会 - 类人机器人国际会议，《人工智能》，《人工智能研究杂志》，《机器学习》 - 计算语言学协会，自主智能体与多智能体系统国际联合会议，机器人学习会议，《IEEE模式分析与机器智能汇刊》，国际计算机视觉大会，不确定性人工智能会议，《机器学习研究杂志》，国际自动规划与调度会议 2) IEEE计算机视觉与模式识别会议 - 国际计算机视觉大会 - 欧洲计算机视觉会议 - 《IEEE模式分析与机器智能汇刊》 - 《国际计算机视觉杂志》 - 《计算机视觉与图像理解》 - 亚洲计算机视觉会议 - 人工智能促进协会 - 国际人工智能联合会议 - 神经信息处理系统大会 - 国际机器学习大会 - 《IEEE神经网络汇刊》 - 神经信息处理系统大会 - 国际联合神经网络会议 - 《IEEE知识与数据工程汇刊》，IEEE国际数据挖掘会议 - 数据挖掘与知识发现会议，知识发现与数据挖掘会议，人工智能与统计学国际会议 - 《IEEE神经网络与学习系统汇刊》，《IEEE系统、人类与控制论汇刊C辑》，《神经计算》 - 《数据挖掘与知识发现》，北美计算语言学协会人类语言技术分会年会，《机器学习》，IEEE/RSJ智能机器人与系统国际会议，不确定性人工智能会议，《机器学习研究杂志》 3) 人工智能促进协会 - 国际人工智能联合会议 - 信息与知识管理国际会议 - 知识发现与数据挖掘会议 - IEEE国际数据挖掘会议 - 《IEEE知识与数据工程汇刊》 - 数据挖掘与知识发现会议 - 亚太知识发现与数据挖掘会议 - 万维网大会 - 网络搜索与数据挖掘会议 - 信息检索研究与发展国际会议 - 欧洲机器学习与知识发现联合会议 - 国际机器学习大会 - 计算语言学协会 - 神经信息处理系统大会 - IEEE计算机视觉与模式识别会议 - 不确定性人工智能会议，《IEEE模式分析与机器智能汇刊》 - 《人工智能》 - 《机器学习》 - 计算语言学协会／国际自然语言处理联合会议，国际语义网会议，自然语言处理实证方法会议／国际自然语言处理联合会议，ECML - 神经信息处理系统大会，人类语言技术—北美计算语言学协会年会，IEEE机器人与自动化国际会议，计算自然语言学习会议，Autonomous Agents and Multi-Agent Systems，北美计算语言学协会人类语言技术分会年会，PKDD，机器学习，《机器人与自主系统》，欧洲人工智能会议，学习理论会议，人工智能与统计学国际会议，《机器学习研究杂志》，自然语言处理实证方法会议，IEEE/RSJ智能机器人与系统国际会议，国际计算语言学会议，自然语言处理实证方法会议—计算自然语言学习会议 4) 国际联合神经网络会议 - Neural Networks - 《IEEE神经网络汇刊》 - 人工神经网络国际会议 - 《神经计算》 - 神经信息处理系统大会 - 《机器学习研究杂志》 - 《IEEE模式分析与机器智能汇刊》 - 《机器学习》 - 国际机器学习大会 - 欧洲机器学习与知识发现联合会议 - 不确定性人工智能会议 - 数据挖掘与知识发现会议 - 知识发现与数据挖掘会议 - 学习理论会议 - 自然语言处理实证方法会议／国际自然语言处理联合会议，人工智能促进协会，亚太知识发现与数据挖掘会议，人工智能与统计学国际会议，推荐系统会议 - 北美计算语言学协会人类语言技术分会年会，IEEE计算机视觉与模式识别会议，《国际计算机视觉杂志》，信息与知识管理国际会议，信息检索研究与发展国际会议，网络搜索与数据挖掘会议，欧洲计算语言学协会会议，国际计算机视觉大会，ECML，万维网大会，自然语言处理实证方法会议—计算自然语言学习会议 5) IEEE/RSJ智能机器人与系统国际会议 - 《机器人与自主系统》 - 《自主机器人》 - 《IEEE机器人与自动化杂志》 - IEEE机器人与自动化国际会议 - 类人机器人国际会议 - 国际机器人研究研讨会 - 机器人：科学与系统 - 《IEEE机器人与自动化快报》 - 《国际机器人研究杂志》 - 神经信息处理系统大会 - 国际人工智能联合会议 - 人工智能与统计学国际会议 - 《机器学习研究杂志》 - 国际机器学习大会，机器人学习会议，ECML，《IEEE机器人学汇刊》，Neural Networks，《神经计算》，国际联合神经网络会议 - 欧洲机器学习与知识发现联合会议，《IEEE神经网络与学习系统汇刊》，《人工智能》，人工智能促进协会，人工神经网络国际会议，自主智能体与多智能体系统国际联合会议，《机器学习》，欧洲人工智能会议，《IEEE模式分析与机器智能汇刊》，国际自动规划与调度会议 6) 神经信息处理系统大会 - 国际机器学习大会 - 人工智能与统计学国际会议 - 《机器学习研究杂志》 - 不确定性人工智能会议 - 《机器学习》 - 学习理论会议 - 神经信息处理系统大会 - 知识发现与数据挖掘会议 - IEEE国际数据挖掘会议 - 国际人工智能联合会议 - 《IEEE模式分析与机器智能汇刊》 - 人工智能促进协会 - 《神经计算》 - IEEE机器人与自动化国际会议，《IEEE神经网络汇刊》，数据挖掘与知识发现会议，Neural Networks，国际联合神经网络会议 - 类人机器人国际会议，人工神经网络国际会议，《自主机器人》，亚太知识发现与数据挖掘会议，IEEE/RSJ智能机器人与系统国际会议 - 《人工智能》，《IEEE机器人与自动化快报》，IEEE计算机视觉与模式识别会议，《国际计算机视觉杂志》，亚洲计算机视觉会议，PKDD，《国际机器人研究杂志》，网络搜索与数据挖掘会议，欧洲计算机视觉会议，国际计算机视觉大会，ECML，万维网大会，机器人：科学与系统，国际学习表征会议 7) CHI - 泛在计算国际联合会议 - 《ACM计算机与人类交互汇刊》 - 智能用户界面国际会议 - 国际人工智能联合会议 - 自主智能体与多智能体系统国际联合会议 - 《人工智能研究杂志》 - 人类计算与众包会议，《人工智能》，人工智能促进协会 - 万维网大会，不确定性人工智能会议 - 人类语言技术—北美计算语言学协会年会，计算语言学协会，国际机器学习大会，信息与知识管理国际会议，知识发现与数据挖掘会议，网络搜索与数据挖掘会议，自然语言处理实证方法会议 - 计算语言学协会／国际自然语言处理联合会议，《IEEE知识与数据工程汇刊》，知识表示与推理国际会议，自然语言处理实证方法会议／国际自然语言处理联合会议，北美计算语言学协会人类语言技术分会年会，《机器学习》，AIPS，国际自动规划与调度会议 8) 国际人工智能联合会议 - 《人工智能》 - 欧洲人工智能会议 - 《人工智能研究杂志》 - 知识表示与推理国际会议 - 欧洲逻辑与人工智能会议 - 《理论与实践逻辑编程》 - 国际逻辑编程会议 - 人工智能促进协会 - 逻辑编程与自动推理国际会议 - 规则与表示推理会议 - 欧洲语义网会议 - 《IEEE 知识与数据工程汇刊》，国际语义网会议，规则标记语言 - 规则标记语言+规则与表示推理会议 - 自动推理联合国际会议，归纳逻辑编程，约束规划与运筹研究国际会议，约束编程，万维网大会，理论与理性会议 9) 信息与知识管理国际会议 - 信息检索研究与发展国际会议 - 万维网大会 - 网络搜索与数据挖掘会议 - 知识发现与数据挖掘会议 - 计算语言学协会 - 自然语言处理实证方法会议 - 自然语言处理实证方法会议／国际自然语言处理联合会议 - 人工智能促进协会 - 北美计算语言学协会人类语言技术分会年会 - 神经信息处理系统大会 - 神经信息处理系统大会，国际机器学习大会 - 国际人工智能联合会议 - 欧洲机器学习与知识发现联合会议 - 《人工智能研究杂志》 - 学习理论会议，IEEE国际数据挖掘会议，《机器学习》 - 计算语言学协会／国际自然语言处理联合会议，人类语言技术—北美计算语言学协会年会，《人工智能》，人类语言技术／自然语言处理实证方法会议，机器学习，推荐系统会议，数据挖掘与知识发现会议，不确定性人工智能会议，自然语言处理实证方法会议—计算自然语言学习会议，国际学习表征会议 - 计算自然语言学习会议，亚太知识发现与数据挖掘会议，欧洲计算语言学协会会议，国际计算机视觉大会，国际计算语言学会议，《机器学习研究杂志》 10) 计算语言学协会 - 自然语言处理实证方法会议 - 国际计算语言学会议 - 人类语言技术—北美计算语言学协会年会 - 欧洲计算语言学协会会议 - 北美计算语言学协会人类语言技术分会年会 - 自然语言处理实证方法会议／国际自然语言处理联合会议 - 计算自然语言学习会议 - 自然语言处理实证方法会议—计算自然语言学习会议 - 人类语言技术／自然语言处理实证方法会议 - 《人工智能研究杂志》 - 人工智能促进协会 - 国际机器学习大会 - 国际人工智能联合会议，知识发现与数据挖掘会议，《机器学习》，神经信息处理系统大会，《机器学习研究杂志》 - 国际计算语言学会议—计算语言学协会 - 《IEEE知识与数据工程汇刊》，知识表示与推理国际会议，归纳逻辑编程，自主智能体与多智能体系统国际联合会议，PKDD，欧洲计算机视觉会议，万维网大会，欧洲机器学习与知识发现联合会议，IEEE计算机视觉与模式识别会议，信息与知识管理国际会议，《IEEE模式分析与机器智能汇刊》，神经信息处理系统大会，《人工智能》，ECML，《神经计算》，学习理论会议，人工智能与统计学国际会议，IEEE国际数据挖掘会议，《自动推理杂志》 11) 知识发现与数据挖掘会议 - IEEE国际数据挖掘会议 - 数据挖掘与知识发现会议 - 《IEEE知识与数据工程汇刊》 - 亚太知识发现与数据挖掘会议 - 《数据挖掘与知识发现》 - 欧洲机器学习与知识发现联合会议 - PKDD - 《机器学习》 - 国际机器学习大会 - 国际人工智能联合会议 - 信息与知识管理国际会议 - 人工智能促进协会 - 《机器学习研究杂志》 - 不确定性人工智能会议 - 神经信息处理系统大会，《人工智能研究杂志》，ECML - 知识表示与推理国际会议，《人工智能》，归纳逻辑编程，IEEE机器人与自动化国际会议，《国际近似推理杂志》，《自主机器人》，国际逻辑编程会议，《理论与实践逻辑编程》，欧洲机器学习研讨会，机器学习，欧洲人工智能会议，归纳逻辑编程研讨会，《计算机视觉与图像理解》，国际计算机视觉大会，IEEE/RSJ智能机器人与系统国际会议，自然语言处理实证方法会议—计算自然语言学习会议，概率图模型，国际联合神经网络会议 12) 《IEEE模式分析与机器智能汇刊》 - 欧洲计算机视觉会议 - 亚洲计算机视觉会议 - 《计算机视觉与图像理解》 - 《国际计算机视觉杂志》 - 国际计算机视觉大会 - IEEE计算机视觉与模式识别会议 - 《IEEE神经网络与学习系统汇刊》 - 人工智能促进协会 - 《IEEE知识与数据工程汇刊》 - IEEE国际数据挖掘会议 - 万维网大会 - 国际人工智能联合会议 - 信息检索研究与发展国际会议 - 亚太知识发现与数据挖掘会议 - CHI，自然语言处理实证方法会议，网络搜索与数据挖掘会议，知识发现与数据挖掘会议 - 自然语言处理实证方法会议／国际自然语言处理联合会议，欧洲机器学习与知识发现联合会议，人类语言技术—北美计算语言学协会年会，计算语言学协会，信息与知识管理国际会议，数据挖掘与知识发现会议

图6。德国人工智能数据集的贪心分解中前十二个因子的属性

我们认为，这两个数据集都适合使用序数因子分析进行考察。在第一个数据集中，因子包含政党；而在第二个数据集中，因子包含陈述。在讨论政治议题时，人们通常会将陈述称为进步派或保守派。这些陈述也常常会根据其进步程度或保守程度相互比较。因此，在日常讨论政治议题时，人们会认为各方立场之间存在线性序。政党也是如此，我们也以类似方式对它们进行了排序。然而，在多元的政治格局中，单一的排序很可能不足以用这种方式进行分析。在我们的数据集中，我们发现了一些通常被归类为右翼的政党，它们也持有一些进步立场，例如提高最低工资。序数因子分析能够更深入地考察这种情况。每个因子都会给出一个线性序，分别对政党或陈述进行排列。因此，每个因子都与某个政党的立场相一致。分析陈述数据集时，我们可以观察到，传统的进步派立场集中在第一个因子中，而第二个因子则包含保守派立场。其他因子则更为专门，因为立场的排序依据并非保守派—进步派这一维度，而可能是移民或经济学等议题。在每个因子中，排在前面的陈述都是许多政党所持有的。因此，这些陈述最有可能由未来的政府付诸实施。

E. 局限性

该方法的主要局限性在于计算可行性。该算法必须求解一个 NP-完全问题，这会导致相对于输入数据规模呈指数时间复杂度的算法。因此，对于大型数据集而言，该方法并不可行；目前尚无法计算其概念格。另一个局限性是，就目前而言，该算法只能处理二元数据集。包含数值数据或类别数据的数据集必须先转换为二元形式，之后才能应用该算法。

IX. 结论与展望

在本研究中，我们基于已有的序数因子分析概念，提出了算法 ORDIFIND。为此，我们利用了形式概念分析领域中开发的快速算法。由此得到的算法使我们能够为更大型的无序二元数据集计算贪心序数因子分解，这类数据集通常出现在数据挖掘任务中。借助这种因子分解，我们展示了如何发现原始数据中的关系。

数据集通常不仅包含二元数据，也包含已经具有序数特征的数据。目前形式的序数因子分析只能将这类数据视为二元数据来处理。尽管形式概念分析中的尺度化是处理这类数据的一种工具，但因子未必会遵循数据所包含的顺序。我们认为，下一步应当扩展这一方法，使其能够直接处理序数数据。此外，更快的概念格算法始终值得期待，尤其是在现代网络环境中，数据集的增长往往还在加速。对于规模大到我们的方法无法处理的数据集，可以开发用于近似序因子的启发式方法。最后，我们认为，探索其他非序数多值因子可能也很有意思。

参考文献

[1] B. Ganter 和 C. V. Glodeanu，“序数因子分析”，载于《形式概念分析——第10届国际会议，ICFCA》，《计算机科学讲义》系列，第7278卷。Springer，2012年，第128–139页。[2] M. Gupta、R. Li、Z. Yin 和 J. Han，“社交标注技术综述”，SIGKDD Explor.，第12卷，第1期，第58–72页，2010年。[3] O. Nov、M. Naaman 和 C. Ye，“是什么驱动内容标注：Flickr 上照片的案例”，载于《2008年计算系统中的人为因素大会论文集》，CHI。ACM，2008年，第1097–1100页。[4] M. Heckner、T. Neubauer 和 C. Wolff，“树、有趣、可读、谷歌：标签应该实现什么？不同数字资源类型的用户关键词比较分析”，载于《2008年ACM社交媒体搜索研讨会论文集》，SSM。ACM，2008年，第3–10页。[5] A. Onan，“基于词嵌入和聚类的文献计量数据分析两阶段主题提取模型”，IEEE Access，第7卷，第145 614–145 633页，2019年。[6] M. M. Kalayeh、H. Idrees 和 M. Shah，“NMF-KNN：使用加权多视图非负矩阵分解进行图像标注”，载于《2014年IEEE计算机视觉与模式识别会议》，CVPR 2014。IEEE 计算机学会，2014年，第184–191页。[7] R. Bellman，“动态规划”，Science，第153卷，第3731期，第34–37页，1966年。[8] P. E. Hoffman 和 G. G. Grinstein，“高维数据挖掘可视化综述”，载于《数据挖掘与知识发现中的信息可视化》，2001年，第47–82页。[9] M. Espadoto、R. M. Martins、A. Kerren、N. S. T. Hirata 和 A. C. Telea，“朝着定量评估降维技术的方向迈进”，IEEE视觉与计算机图形学汇刊，第27卷，第3期，第2153–2173页，2021年。[10] K. Pearson，“Liii. 关于空间点系的最佳拟合直线和平面”，《伦敦、爱丁堡和都柏林哲学杂志与科学期刊》，第2卷，第11期，第559–572页，1901年。[11] P. M. Jain 和 V. Shandliya，“主成分分析（PCA）与探索性因子分析（EFA）比较研究综述论文”，International Journal of Managment, IT and Engineering，第3卷，第6期，第415–424页，2013年。[12] D. Child，《因子分析基础》。Cassell Educational，1990年。[13] R. Wille，“重构格理论：一种基于概念层次结构的方法”，第445–470页，1982年。[14] B. Ganter 和 R. Wille，《形式概念分析——数学基础》。Springer，1999年。[15] A. Keprt 和 V. Sn´asel，“借助形式概念进行二元因子分析”，载于《CLA 2004年概念格及其应用国际研讨会论文集》，《CEUR研讨会论文集》系列，第110卷。CEUR-WS.org，2004年。[16] A. Keprt，“二元因子分析算法”，博士学位论文，博士论文，2006年。[17] R. Belohl´avek 和 V. Vychodil，“布尔因子分析中的形式概念作为最优因子：蕴涵与实验”，载于《第五届概念格及其应用国际会议论文集》，CLA 2007，《CEUR研讨会论文集》系列，第331卷。CEUR-WS.org，2007年。[18] ——，“通过一种新颖的矩阵分解方法发现二元数据中的最优因子”，计算机与系统科学杂志，第76卷，20，2010 1，3第。页–。[19] P. D. Boeck 和 S. Rosenberg，“层级类别：模型与数据分析”，Psychometrika，第53卷，第3期，第361–381页，1988年9月。[20] C. V. Glodeanu 和 B. Ganter，“序数因子分析的应用”，载于《形式概念分析，第11届国际会议，ICFCA 2013》，《计算机科学讲义》系列，第7880卷。Springer，2013年，第109–124页。[21] ——，“序数因子分析的应用”，载于《形式概念分析，第11届国际会议，ICFCA 2013》，德国德累斯顿，2013年5月21日至24日。论文集，《计算机科学讲义》系列，第7880卷。Springer，2013年，第109–124页。[22] C. V. Glodeanu 和 J. Konecny，“分级数据的序数因子分析”，载于《形式概念分析——第12届国际会议，ICFCA 2014》，《计算机科学讲义》系列，第8478卷。Springer，2014年，第128–140页。

[23] Z. Zhang、T. Li、C. H. Q. Ding 和 X. Zhang，“二元矩阵因子分解及其应用”，载于《第七届 IEEE 国际数据挖掘会议论文集（ICDM 2007）》。IEEE 计算机学会，2007年，第391–400页。[24] M. Yannakakis，“求解最小填充问题是 NP 完全的”，SIAM 代数离散方法杂志，第2卷，第1期，第77–79页，1981年。[25] C. Lindig，“快速概念分析”，《概念结构研究——ICCS论文集》，第2000卷，第152–161页，2000年。[26] P. Krajca，“利用空交集提升 Lindig 风格算法的性能”，载于《基于图的表示与推理——第26届国际概念结构会议，ICCS 2021》，《计算机科学讲义》系列，第12879卷。Springer，2021年，第91–104页。[27] T. Koopmann、M. Stubbemann、M. Kapa、M. Paris、G. Buenstorf、T. Hanika、A. Hotho、R. J¨aschke 和 G. Stumme，“邻近维度与合作的兴起：德国人工智能研究的 hyptrails 研究”，《科学计量学》，第1–22页，2021年。[28] K. Kersting、J. Peters 和 C. A. Rothkopf，“什么是人工智能教授职位？”，CoRR，第abs/1903.09516卷，2019年。 [在线]。网址：http://arxiv.org/abs/1903 09516 。IMDb.com，“IMDb 数据集”，在[29] 2004, [线；访问日期：06-October-2021]。[在线]。网址：https://www.imdb.com/interfaces/ [30] Bundeszentrale f ¨ 2021, ur [07 2022] [ ] politische Bildung，“Wahl-O-Mat”，在线；访问日期：-February-。在线。网址：https://www.bpb.de/politik/wahlen/wahl-o-mat/337541/download