Modeling and Simulation 建模与仿真, 2023, 12(2), 1469-1480
Published Online March 2023 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2023.122137

Hans汉斯

基于HDBSCAN聚类算法的实例推理与规则提取

亓凯航，仲梁维

上海理工大学机械工程学院，上海

收稿日期：2023年2月14日；录用日期：2023年3月20日；发布日期：2023年3月27日

摘　要

针对复杂装配对象具有结构复杂、开发周期长、装配成本高等特点导致的装配工艺编制较慢、效率低的问题，为实现装配工艺重用，在规则提取过程中，利用Apriori关联规则算法提取出满足约束参数的强关联规则，作为知识检索的条件与结论放入规则库中；在实例推理过程中，提出基于DBSCAN聚类算法快速定位与目标装配对象相似的子实例集，即与目标对象最相似的簇，缩小实例检索的范围以提高匹配的效率。结果表明，该方法使检索范围缩小了50倍，实例匹配速度明显加快。

关键词

Apriori，HDBSCAN，规则提取，实例推理

Case Reasoning and Rule Extraction Based on HDBSCAN Clustering Algorithm

Kaihang Qi, Liangwei Zhong

School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Feb. 14th, 2023; accepted: Mar. 20th, 2023; published: Mar. 27th, 2023

Abstract

Due to the complex assembly object's complex structure, long development cycle and high assembly cost, the assembly process is slow and the efficiency is low. In order to realize assembly process reuse, in the process of rule extraction, the Apriori association rule algorithm is used to extract the strong association rules meeting the constraint parameters and put into the rule base as the conditions and conclusions of knowledge retrieval. In the process of case reasoning, the DBSCAN clustering algorithm is proposed to quickly locate the sub-instance set similar to the target assembly object, that is, the cluster most similar to the target object, and narrow the scope of in-

文章引用: 亓凯航, 仲梁维. 基于 HDBSCAN 聚类算法的实例推理与规则提取[J]. 建模与仿真, 2023, 12(2): 1469-1480.
DOI: 10.12677/mos.2023.122137

stance retrieval to improve the matching efficiency. The results show that the retrieval range is reduced by 50 times and the case matching speed is greatly accelerated.

Keywords

Apriori, HDBSCAN, Rule Extraction, Case Reasoning

1. 引言

在工艺设计阶段，工程师往往在编写复杂装配体的工艺文件与设备工艺卡时会耗费大量的时间，通常会借鉴以往工艺文件中相似的工艺内容，查询以往成熟的工艺方法，以及检索相应的装配工艺知识。这样人工的编写方式都会降低工艺工程师的工作效率，使得新产品的设计成本增高。而工艺重用可以通过在设计过程中调用以往的成熟工艺知识，在此基础上进行创新，以提高工程师设计效率。

装配工艺知识主要来自于企业制定的工艺标准、行业内部的规范以及相关专家的设计经验等等。大量的装配实例，都存储在企业的 CAPP 系统之中[1]，其中不仅含有规范化的工艺知识，而且还是广大工程师智慧与经验的结晶。但是这些实例并不是有序的规则，也不是结构化的信息，许多有价值的信息也隐匿在其中，设计人员在对这些知识的利用上有着较大的困难。

在对于型号多样的同类产品而言，装配件的结构大多相似。对这类产品进行装配工艺知识的重用[2]，可以避免许多无效的劳动，提高工程师的设计效率，使工程师把更多的精力花在对新装配件的工艺设计之上。实例推理技术可以借助装配对象内部的信息，使用相似度检索方法将与目标装配对象相似的装配对象实例匹配出来，复用相似对象的工艺知识，达到辅助工艺人员有效地进行产品工艺设计工作的作用。

2. Apriori 算法基本原理

关联规则挖掘通常是指挖掘两个对象之间的关联信息，采用统计学习的方法来从大量数据之中提取出频繁项集，作为专家经验来二次利用。关联规则的基本概念主要有以下几个：

项集：指的是由多个元素组成的一个集合，其中元素之间应当是一个独立的个体，k 个元素组成的项集即为 k 阶项集，k 项集如式(1) [3]：

$$I_k = \{i_1, i_2, i_3 \cdots, i_k\} \tag{1}$$

对象：包含同一个事务内，任意一个对象中可能包含一个或多个元素，对象之间互不相交，A 对象，B 对象的关系如式(2)：

$$A \subseteq I, B \subseteq I, A \cap B = \varnothing, A \cup B = I \tag{2}$$

事务集：在关联规则中，每个项集都有其从属于的事务，而特定的事务一起组成了事务集，包含 n 个事务的事务集表示如式(3)：

$$T_n = \{t_1, t_2, t_3, \cdots, t_n\} \tag{3}$$

支持度：指的是某一项集的事务个数在事务集中所占的比例，如式(4)：

$$Support(I) = \frac{count(I \subseteq T)}{|T|} \tag{4}$$

置信度：指的是同时包含项集 I_1 和 I_2 这两个对象的事务个数在只包含 I_1 的事务个数中所占的比例，如式(5)：

$$\text{Confidence}\left(I_1 \Rightarrow I_2\right)=\frac{\text{Support}\left(I_1 \cap I_2\right)}{\text{Support}\left(I_1\right)} \tag{5}$$

提升度：指的是项集和的置信度和的支持度之比，如式(6)：

$$\text{Lift}\left(I_1 \Rightarrow I_2\right)=\frac{\text{Support}\left(I_1 \cap I_2\right)}{\text{Support}\left(I_1\right)\cdot\text{Support}\left(I_2\right)}=\frac{\text{Confidence}\left(I_1 \Rightarrow I_2\right)}{\text{Support}\left(I_2\right)} \tag{6}$$

Agrawal 提出了关联规则算法中最为经典的算法 Apriori 算法，该算法的核心思想为所有频繁项集的非空子集必定是频繁项集，所有非频繁项集的超集必定是非频繁项集[4]。Apriori 算法主要有四大步骤：

1) 确定最大项集阶数 k，最小支持度 min_support，最小置信度 min_confidence，最小提升度 min_lift。

2) 从 1 到 k 地逐层遍历，获取高于最小支持度的频繁项集并利用先验性质进行剪枝。

3) 从每一个频繁项集中提取出高于最小置信度的两个对象，分别作为条件对象和结果对象，形成关联规则。

4) 利用最小提升度来验证关联规则是否有效，若高于最小提升度，则判定为有效强关联规则。

其中，计算成本最大的是第二步频繁项集的获取，Apriori 算法主要流程如图 1 所示。

首先对一阶候选项集进行计数，获取所有的一阶候选项集，之后判断并统计所有大于最小支持度的项集作为频繁项集。对于 K 阶候选项集的提出，主要是利用 Apriori-Gen 运算实现，其中包含了连接步和剪枝步，连接所有频繁项集生成 K 阶项集，然后利用先验定理排除含有 $K-1$ 阶非频繁项集的 K 阶项集[5]。之后判断并统计所有大于最小支持度的候选项集作为 K 阶频繁项集。

3. 装配工艺规则提取

3.1. 工艺规则库设计

规则库的结构不需要太复杂，主要按照条件表达式的结构，即“IF……THEN”的模式来构造，设计者只需按照输入条件进行模糊查询，系统会自动输出相关结论并给出相应概率，设计者就可以通过这种方式有选择地参考，从而优化新实例的装配工艺文件。这里选择关联规则算法 Apriori 是因为算法输出的是规则前项和规则后项，对应的就是规则库中的条件和结果，这也满足了人机交互问答系统的基本功能。

3.2. 具体方案设计与应用

由于提取的装配工艺信息主要是由工艺信息库，相关零件库与装配辅材库构成，由于字符串表示的信息很多，如果想要构建事务集则必须要对字符串进行分词处理，会出现信息冗余的情况，通过这种方法利用 Apriori 关联规则算法会提取出许多无用的规则项集，这不利于工艺知识的检索，因此，需要对工艺指令通过相关零件库与装配辅材库的帮助进行拆分。根据装配工艺信息的自身特点，可以将工艺指令拆分成五大部分，装配信息事务集[6]如下所示：

$$TS_n=\begin{pmatrix} P_1 & C_1 & F_1 & E_1 & M_1 \\ P_2 & C_2 & F_2 & E_2 & M_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{n-1} & C_{n-1} & F_{n-1} & E_{n-1} & M_{n-1} \\ P_n & C_n & F_n & E_n & M_n \end{pmatrix} \tag{7}$$

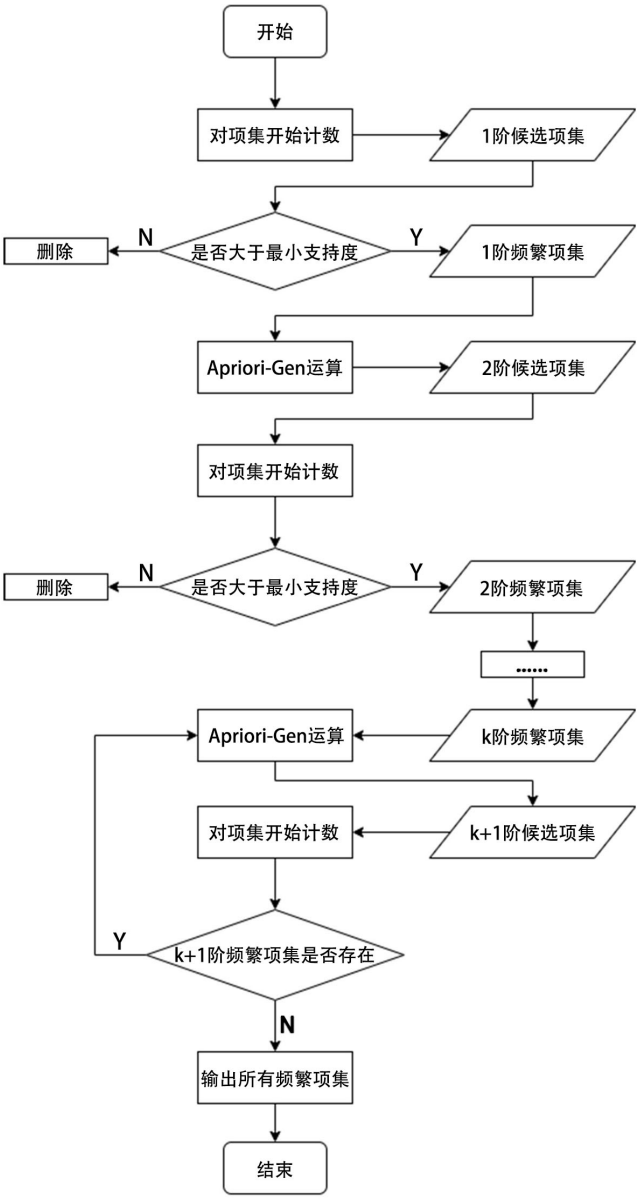

Figure 1. Flow chart of frequent item set fetching
图 1. 频繁项集获取的流程图

对于工艺事务集 TS_n，集合 P_n 代表了装配的基准集合，C_n 为相关组件的集合，F_n 为相关装配辅材的集合，E_n 为工艺方法的类型，M_n 为装配工艺指令对应的装配动作。对于对应工艺指令的各部分信息获取，主要提取流程如图 2 所示。

对于装配工艺信息库中的每一条装配工艺指令而言，其内部都蕴含着许多有价值的信息，因此需要使用不同的技术来提取每一条工艺指令中的不同类型的知识，将其组成一个完整的工艺事务集。其中，对于相关组件的提取，主要是通过相关部件库中组件信息与工艺指令中的组件名进行匹配得出。同样的，工具辅材库与工艺指令的辅材名称进行匹配，得出所需要的装配工具辅材信息。对于工艺方法的分类，这里使用一种分级加权的词袋模型构建方法[7]，结合 K-means 聚类方法获取。关于装配动作与装配基准做，则是通过和事先构建的词典进行匹配，若无法匹配得到，则取空值。

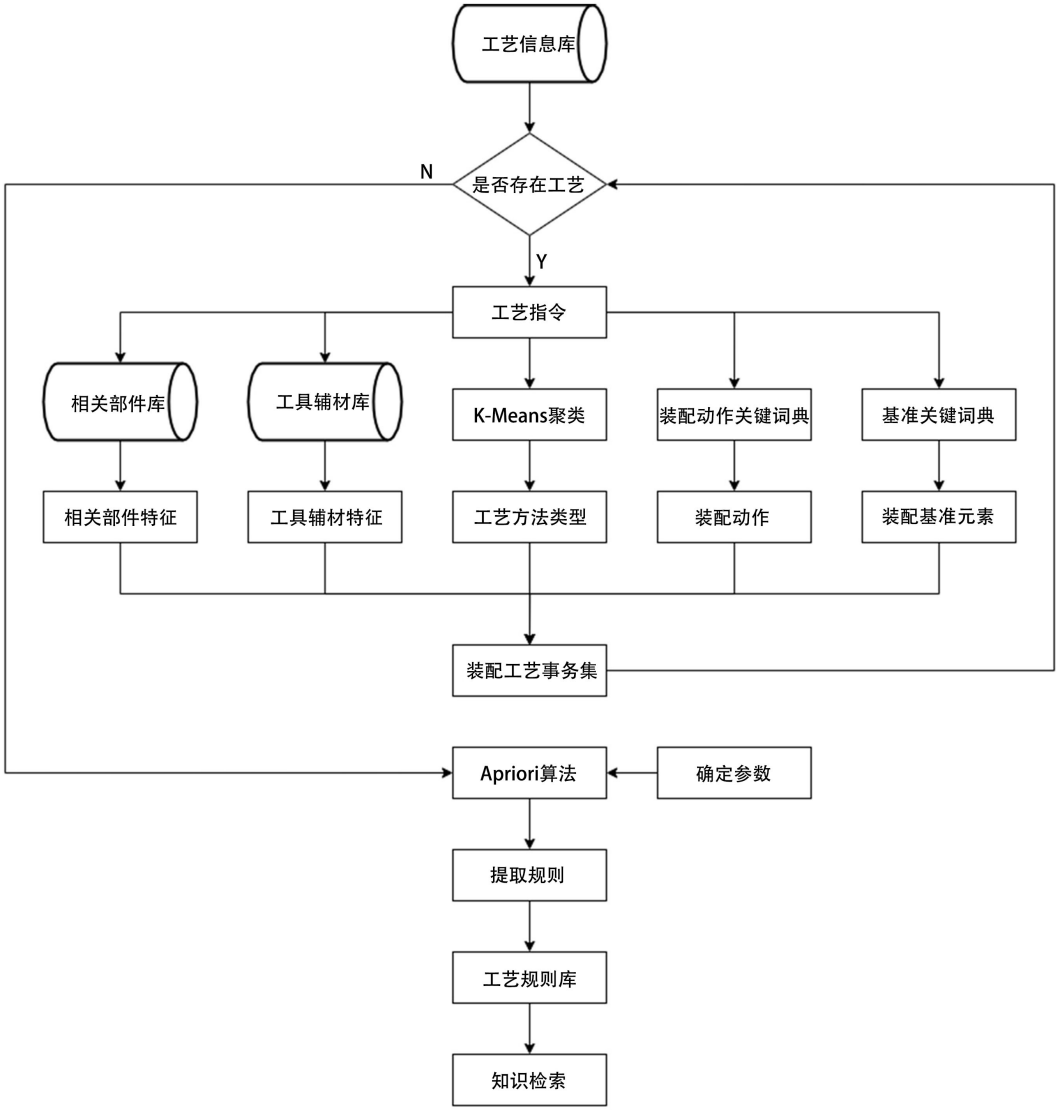

Figure 2. Flow chart of assembly process transaction set information acquisition
图 2. 装配工艺事务集信息的获取流程图

简单处理导轨和卫星部分模块装配工艺事务集，作为 Apriori 关联规则算法的输入，设置最小支持度为 0.04，最小置信度为 1，最小提升度为 2，同时筛选出规则后项为一个元素的规则，可以得到的如下所示的部分规则如表 1 所示。

支持度越高，证明关联元素在工艺信息库中的出现的频率越高，提升度越高，则证明规则前项与规则后项之间的关联性越大，置信度设置为 1，因此在规则前项确定的时候，结果是必定会出现的，通过与规则前项输出类似的条件来查询结果，从而实现对工艺知识的检索，通过已有知识来优化工艺文件。

4. 基于 HDBSCAN 聚类算法的实例推理

4.1. HDBSCAN 算法基本原理

HDBSCAN 是由 Campello，Moulavi 和 Sander 开发的聚类算法。它是通过将 DBSCAN 转换为分层聚类算法来扩展 DBSCAN，然后基于聚类稳定性[8]，使用了提取平面聚类的技术。

Table 1. The extracted association rules and indexes of assembly process
表 1. 提取的装配工艺关联规则和指标

规则前项	规则后项	支持度	置信度	提升度
无水乙醇	纱布	0.04	1	23
电动螺丝刀	批头	0.07	1	13.8
M5 螺丝，电动螺丝刀，套筒扳手	安装类	0.04	1	2.76
螺钉，紧固	限力扳手	0.04	1	23
安装类，防松螺母 M5，装配	套筒扳手	0.04	1	13.8
安装类，电动螺丝刀，装配	批头	0.07	1	13.8
喷涂类	导热硅脂	0.04	1	23
插座，插接，低频电缆网	插接类	0.12	1	5.75
螺钉，紧固，限力扳手	紧固类	0.04	1	23

传统的 DBSCAN 的密度可达距离是不可变的，而 HDBSCAN 可以处理密度不同的聚类问题，同时它使用了互相可达距离度量这一定义，大大提高了算法的鲁棒性，在生成最小树之后，采用分层聚类的方法聚合簇，在后期有利用压缩聚类树的方式来筛选离群值，这些过程都是通过算法自动处理，而人工只需要处理的参数是最小簇的样本个数，一般默认即可。因此在超参数的选择上即不需要指定簇的个数 K，也不需要指定密度可达距离以及密度可达数，这使得程序不需要通过贪婪搜索的方法来确定最优参数，大大减少计算成本。

HDBSCAN 的基本原理[9]包括了以下五个步骤：

1) 根据密度变换空间

通过将每个样本点作为核心点，两两构建核心距离，即相互可达距离来分散低密度点，这使得低密度点与高密度簇的距离更远，聚类算法对噪声的鲁棒性更强，点到点的相互可达距离定位为式(8)：

$$d_{mreach-k}(a,b)=\max\left\{core_k(a),core_k(b),d(a,b)\right\} \tag{8}$$

其中 $core_k(a)$ 是指在与核心点 a 距离最近的第 k 个样本点的彼此距离，$core_k(b)$ 指在与核心点 b 距离最近的第 k 个样本点的彼此距离，$d(a,b)$ 指的是 a 和 b 的距离，利用这种度量方法，可以让密集点之间保持相同的距离，而稀疏点可以被推开至远离核心点的地方，使得密度高的样本更容易聚在一起，而簇之间分隔的更远，同时这也让单链路聚类更加贴合，如图 3 所示。

2) 生成最小树

两个点可以确定一条边，而相互可达距离就是空间内样本点的边，这里也被定义为任意两个顶点间的权重。设定最小簇的样本点个数 k，利用 Prim 算法可以确定最小树，如图 4 所示。

3) 分层构建簇的层次结构

对于每个最小树，这里采用与最小距离分层聚类方法相同的策略，通过对权值递增排序，对于最近的最小树进行合并，不停地迭代这个过程，一直合并至权值最高为止，如图 5 所示。

利用分层聚类的思路，可以将这个簇的聚合过程清晰地展现出来，之后需要确定一条水平线来切割簇，而这水平线就相当于 DBSCAN 中用来分隔不同簇的密度可达距离 eps，eps 的值永远是固定不变的，而 HDBSCAN 的水平线却可以在不同的密度下进行分割来选择合适的簇，这就是 HDBSCAN 的强大之处。

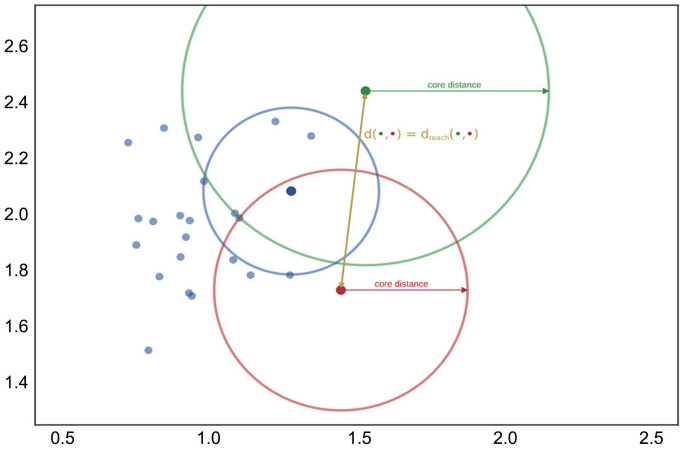

Figure 3. Mutually accessible distance
图 3. 相互可达距离

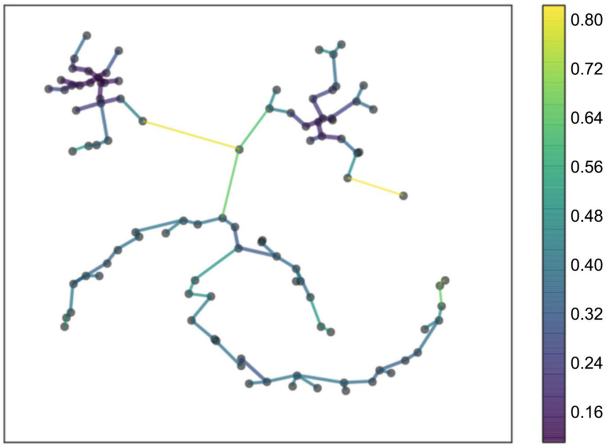

Figure 4. The minimum tree generated by Prim algorithm
图 4. Prim 算法生成最小树

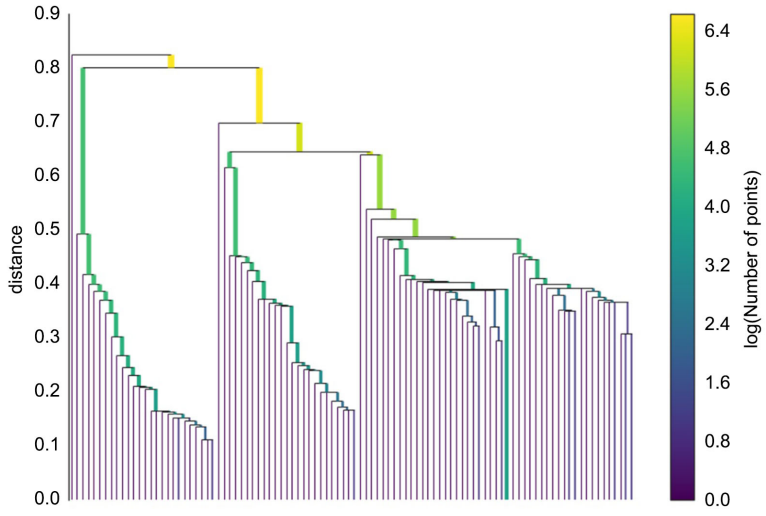

Figure 5. The minimum tree generated by hierarchical clustering
图 5. 分层聚类最小树

4) 压缩聚类树

利用水平线对上图所示的聚类集群进行拆分，这里又要确定最小簇大小 k，遍历层次结构并在每次拆分时判断：是否存在拆分出来的新簇的大小会小于预先定义的 k，如果存在，则将这些样本点标记为离群值点，这就是 HDBSCAN 的离群值处理方法。如果拆分的子集群不存在上述情况，则将该集群视为真正得集群进行拆分，遍历整个层次结构，可以得到如图 6 所示的树状图：

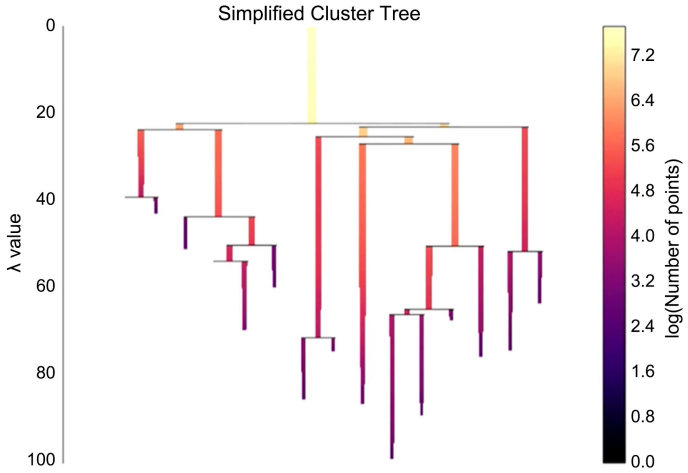

Figure 6. Hierarchical clustering minimum tree using plane segmentation clustering tree
图 6. 利用平面切分聚类树

5) 提取簇

很多情况下集群应当具备更长的生命周期，而不是随着数据量的不断增加而被取代或是被合并，因此一般会选择一些面积较大的集群。在选择之前，会先考量簇的持久性，这里使用相互可达距离的倒数来度量，即 $\lambda = 1/d_{\text{mreach}-k}$，$\lambda_{\text{birth}}$ 是当前集群拆分成功后并成为子集群的 λ 值，λ_{death} 是指当集群拆分成较小的集群时的 λ 值，λ_p 则是指该点被从集群中剔除时的 λ 值。对于每一个集群，稳定性都可以定义为：

$$\sum_{p\in\text{cluster}}\left(\lambda_p - \lambda_{\text{birth}}\right) \tag{9}$$

通过对树进行遍历，如果子集群的稳定性之和大于父集群，集群稳定性就等于其子集群稳定性之和，反之，则将集群以下的所有子类全部归并，不停迭代至根节点，当前选定的集群就是提取出来的平面簇，对于每个簇赋予其独一无二的标签，离群点则以标签-1 表示。

以上就是 HDBSCAN 的工作原理，由于 HDBSCAN 也具有密度聚类的原理，因此数据集不变以及顺序不变的情况下，聚类得到的簇也不会发生改变，因此具有稳定性，对于数据集分布不均匀的装配实例库，最小簇大小也可以默认，不需要人为确定模型的任何参数，这可以大大节省技术人员的时间成本。

针对装配特征向量组成的数据集，各特征的数值分布并没有规则，其内部簇的密度也比较不均匀，使用 HDBSCAN 算法可以有效地解决簇密度不均和离群值问题，且在计算成本上由于不需要进行大量调参实验而变得较低。因此，这里选择 HDBSCAN 算法作为加速检索的核心算法。

4.2. HDBSCAN 融合相似度算法实际应用

联合零件信息库，装配体结构库，装配体配合库生成的实例装配特征矩阵就是 HDBSCAN 中的目标数据集，该数据集并没有明确的类标签，因此符合无监督学习的应用场景，HDBSCAN 在相似度算法中的具体应用流程如图 7 所示：

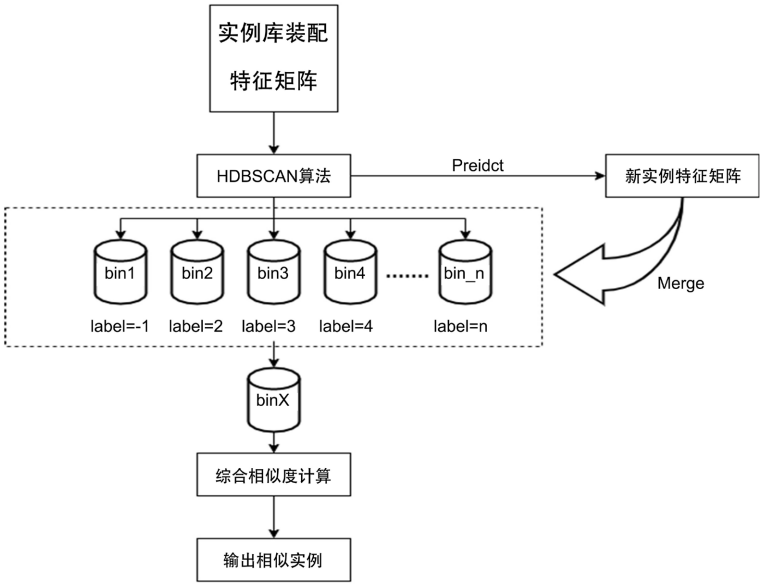

Figure 7. Flow chart of HDBSCAN accelerates search
图 7. HDBSCAN 加速检索流程图

HDBSCAN 算法会提前根据原有的实例库特征矩阵进行聚类，构建一个类似分桶的具体特征，将所有的特征向量分为 *n* 个类别，这与分箱操作相类似，其中离群值会被分配到-1 类，类别特征也会自动存储入实例库中，之后将获取到的新实例特征向量放入之前拟合完成的 HDBSCAN 算法中及进行预测，将其划分入相似度最高的类别之中，之后，只从实例库中取出该类别的所有实例进行相似度计算，这种方法大大地缩减了检索范围，除此之外，也让相似度匹配的对象更加具有针对性，提高匹配效率，减少冗余数据或无效计算。

这里将目标实例设置为 SES0487DT-B10-00A 前导轨组装，从实例库中取出 500 个实例样本，生成装配特征矩阵如图 8 所示：

	实例名	轴类	盘类	肋板件	箱盖类	传动件	紧固件	密封件	连接件	定位件	...	Screw	Linear_coupler	Universal_joint	Coordinate	Slot	Hinge	层数	零件总数	配合总数
0	SES0487DT-B10-001A 前导轨焊接	0	0	0	0	0	0	0	3	0	...	0	0	0	0	0	0	2	8	9
1	SES0487DT-B10-001B 后导轨焊接	0	0	0	0	0	0	0	3	0	...	0	0	0	0	0	0	2	8	9
2	SES0487DT-B10-005 气缸后固定座焊接	0	0	0	0	0	0	0	0	3	...	0	0	0	0	0	0	2	6	6
3	SES0487DT-B10-006 电机端608轴承固定座焊接	0	0	0	0	0	0	0	0	2	...	0	0	0	0	0	0	2	4	4
4	SES0487DT-B10-009 从动轮锁紧板焊接	0	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0	2	6	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
495	SES1169STR-B10-035 同步带轮16-5M-20-AF 组件	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	2	6	4
496	SES1169STR-B10-041 前玻璃门挂件组装	8	0	3	0	4	11	2	2	0	...	0	0	0	0	0	0	3	62	51
497	SES1169STR-B20-000 玻璃门焊接	0	0	2	0	0	0	0	56	0	...	0	0	0	0	0	0	3	158	52
498	SES1169STR-B20-001 玻璃外框焊接	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	2	8	9
499	SES1169STR-B20-002 玻璃中框焊接	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	2	8	9

500 rows × 39 columns

Figure 8. Assembly feature matrix of 500 assembly instances
图 8. 500 个装配实例的装配特征矩阵

利用 HDBSCAN 算法聚类，对于每个样本的向量标签如图 9 所示：

```
[28 28 32 31 27  1  1 32 14 37 -1  8  8 -1 -1 10 -1 30 -1 -1 -1 -1 -1 -1
 5 10 32 36 12 12 13 19 10 -1 17 32 16 -1 26 16 -1 12 31 31 18 -1 10 22
-1 26 33 31 30 25 31 10 -1 10 32 27 -1 10 26 12 31 12 -1 -1  5 10 32 36
-1 -1 10 10 -1 10 34 33 -1 33  5 -1 10 10 -1 11 11 10 10  4 -1 11 10 -1
-1 26 -1 13 20  7  7 22 18 18 10 10 35 35 35 -1 -1 10 35 10  0 -1 -1 -1
-1 22 34  5  5 27 27  3  2 23 29 24  3  2 33 10  2 -1  2 13 21 32 32  9
10 10 18  3  2 -1 29 24  3  2 33  3  2 -1  2 10 15 15 33 25 25  5 23 23
29 24 34 10 15 15 32 33 31 25 25 13 20 28 28 -1 31 36  1  1 32 14 36 -1
-1  1 11 11 -1 -1  5 10 32 36  9 12 13 19 10  4 17 32 16 -1 26 -1  8  8
-1  7  7 22  4 17 32 16 -1 10 -1 19 12 31 31 31 30 25 31 10 -1 -1 32 27
-1 22 26  7  7 22  0 18 18 35 -1  0 -1 -1 22 34  5  5 26 26  3  2 23 29
24  3  2 33 13 20 13 21 13 21 10 32  9 21 13 29 24 34 -1 15 15 32 31 25
25 -1  8  8 -1 -1 10 -1 30 35 28 32 31 36  1 32 14 37 -1 -1 11 11 -1 10
-1 -1  5 10 32 36  9 12 13 19 -1  4 17 36 16 12 31 12 10 -1 10 30  7  7
22 -1 -1 35 -1  6  0 -1 -1  6 35  6 -1 35 -1 -1 14  9 12 13 19 10 35 10
-1 -1 10 18 -1 33 -1 -1 35 35 10 -1 -1 -1  6 35 35 35 33  0 -1 -1 -1 -1
22 34  5  5 26 26  3  2 23 29 24  3  2 33 13 20 13 21 13 21 32 32  9 21
13 29 24 34 10 15 15 32 31 25 13 -1  5 10 28 32 31 37  1 32 14 37 -1  1
11 11 -1 -1 -1 31 31 18 18  1 -1 -1 -1 31  3  2 33  3  2 23 29 -1 32 10
-1 10 30 35 35  0 -1 -1 22 34  5  5 26 26 13 20 13 21 13 21 10 32  9 21
-1 15 15 32 31 25 25 -1 10 32 31 37  1 32 14 37 -1 -1 11 11 -1 10 -1 -1
 5 10 32 36  9 12 13 19 -1  4 17 32 16 12 31 12  7  7 22 -1 10 -1  6 -1
-1 -1 -1 35 10  6 -1 10 -1 -1  4 -1 11 10 31 31 30 32 27 -1 22 26 18 -1
-1 32 10 -1]
```

Figure 9. Cluster labels obtained by HDBSCAN
图 9. 通过 HDBSCAN 分得的簇标签

统计标签个数输出标签对应标签个数的条形图如图 10 所示：

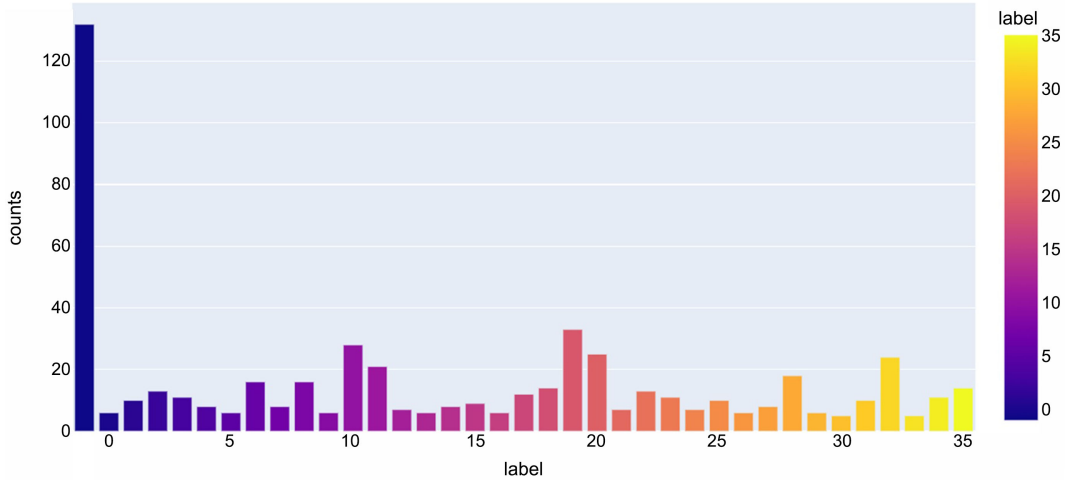

Figure 10. Statistics on the number of labels in each cluster
图 10. 各簇的标签个数统计

从中可以发现该数据集中存在着大于 120 个不能被聚类的数据集的离群样本，即 label = −1，占了非常大的比例，除此之外，其他簇的个数非常接近，说明聚类得到的簇有一定的持久性，之后就可以将新实例归并到其中的一个簇中，进行综合相似度的计算。

该数据集就变成了相似度计算的数据集如图 11 所示，检索范围整整缩小了 50 倍，实例匹配的速度也大大加快了，最终的综合相似度输出结果为表 2。

可以发现簇聚集的质量很高，目标实例与实例库的相似度匹配获得的前 6 个实例的综合相似度较高，这也验证了即使在做了聚类操作之后，相似度匹配算法的精度并没有受到任何影响。

	实例名	轴类	盘类	肋板件	箱盖类	传动件	紧固件	密封件	连接件	定位件	...	Screw	Linear_coupler	Universal_joint	Coordinate	Slot	Hinge	层数	零件总数	配合总数	聚类
5	SES0487DT-B10-00A 前导轨组装	7	0	2	1	2	0	2	5	16	...	0	0	0	0	0	0	3	78	58	1
6	SES0487DT-B10-00B 后导轨组装	7	0	2	1	2	0	2	5	16	...	0	0	0	0	0	0	3	78	58	1
186	SES1259DTL-B10-00A 前导轨组装	7	0	2	1	2	0	1	7	10	...	0	0	0	0	0	0	3	78	56	1
187	SES1259DTL-B10-00B 后导轨组装	7	0	2	1	2	0	1	7	10	...	0	0	0	0	0	0	3	78	56	1
193	SES1259DTL-B20-000 玻璃门焊接	0	0	0	0	0	8	0	8	0	...	0	0	0	0	0	0	3	66	61	1
302	SES1369STL-B10-00A 前导轨组装	7	0	2	1	2	0	1	7	16	...	0	0	0	0	0	0	3	78	56	1
426	SES0679STL-B10-00A 前导轨组装	7	0	3	1	2	0	1	5	16	...	0	0	0	0	0	0	3	78	56	1
431	SES0679STL-B20-000 玻璃门焊接	0	0	2	0	0	6	0	1	0	...	0	0	0	0	0	0	3	62	61	1
441	Q183101-01-B09-000 小推车装配体	0	0	0	0	0	13	2	1	1	...	0	0	0	0	0	0	4	74	63	1
492	SES1169STR-B10-00A 前导轨组装	7	0	3	1	2	0	1	5	14	...	0	0	0	0	0	0	3	78	55	1

10 rows × 39 columns

Figure 11. The retrieved similarity of the front guide rail with a threshold of 10
图 11. 标签为“1”的簇内部装配特征集合

Table 2. System resulting data of standard experiment
表 2. 阈值为 10 的前导轨相似度检索报告

序号	实例名	欧式相似度	余弦相似度	综合相似度%
1	SES0487DT 后导轨组装	1	1	100
2	SES0679STL 前导轨组装	0.9812	0.999	98.02
3	SES1369STL 前导轨组装	0.9744	0.999	97.34
4	SES1169STR 前导轨组装	0.9744	0.999	97.34
5	SES1259DTL 前导轨组装	0.9009	0.995	89.64
6	SES1259DTL 后导轨组装	0.8821	0.994	87.68
7	Q183101 小推车装配体	0.3342	0.97	32.42
8	SES1259DTL 玻璃门焊接	0.159	0.959	15.25
9	SES0679STL 玻璃门焊接	0	0.951	0

5. 结论

本文分析了提高装配工艺重用效率的方法，首先介绍了 Apriori 关联规则算法的基本原理，并利用 Apriori 关联规则算法进行规则提取，作为知识检索的条件和结论放入规则库，以此提高知识检索的效率；针对工艺实例推理，提出了基于 HDBSCAN 聚类算法加速实例推理，通过缩小实例检索范围提高实例匹配的效率。通过验证表明，使用了 Apriori 关联规则算法和 HDBSCAN 聚类算法后，在不影响相似度匹配算法的精度的同时，数据集检索范围缩小了 50 倍，实例匹配的速度明显加快。

参考文献

[1] 曹勇. 基于数据挖掘的工艺知识发现与重用研究[D]: [硕士学位论文]. 济南: 山东大学, 2019.

[2] 吴锐. 面向工艺重用的装配关键技术研究[D]: [硕士学位论文]. 长沙: 国防科学技术大学, 2008.

[3] 何丽瑶. 基于 Apriori 及 XGBoost 算法的道路交通事故分析研究[D]: [硕士学位论文]. 苏州: 苏州大学, 2020.

[4] Yu, H.F. (2021) Apriori Algorithm Optimization Based on Spark Platform under Big Data. *Microprocessors and Microsystems*, **80**, Article ID: 103528. https://doi.org/10.1016/j.micpro.2020.103528

[5] 田春子, 张磊, 朱泽一, 邵晓康, 胡子悦. 基于 Apriori 和 FP-Growth 算法的学生行为分析研究[J]. 信息记录材料, 2020, 21(12): 156-159.

[6] 张胜文, 陆贤磊, 程德俊, 方喜峰, 官威, 李群. 基于改进 Apriori 算法的装配工艺规则挖掘技术[J]. 船舶工程, 2020, 42(5): 108-112.

[7] 崔晴洋, 梁小峰, 倪静, 李帅, 张生, 仲梁维. 基于卫星装配工艺的短文本聚类研究[J]. 软件工程, 2020, 23(4): 7-11.

[8] 王继业, 邓春宇, 郑亚芹, 张玉天, 刘凤魁. 基于 HDBSCAN 动态跟踪客户用电行为模式[J]. 供用电, 2019, 36(1): 10-16.

[9] 郗洋. 基于云计算的并行聚类算法研究[D]: [硕士学位论文]. 南京: 南京邮电大学, 2011.