

基于权限聚类的属性值优化

毋文超^{1,2}, 任志宇¹, 杜学绘¹

(1. 信息工程大学, 河南 郑州 450001;
2. 中国人民解放军 31668 部队, 青海 西宁 810000)

摘要: 在新型大规模计算环境下应用 ABAC (基于属性的访问控制) 面临着属性数量多、来源复杂、质量参差不齐、难以人工修正、难以直接应用于访问控制的问题。针对属性标称值的优化问题, 设计了一种基于权限聚类的属性值优化算法, 通过将实体表示成对应的权限集合, 对实体进行基于密度的聚类, 为实体赋予权限对应的类别标签, 而后基于粗糙集理论对属性值进行化简与修正。最后在 UCI 公开数据集上对算法进行了验证, 证明应用该算法后, ABAC 策略挖掘在真阳性率和 $F1$ 得分上均具有较大的提升。

关键词: 属性值优化; 粗糙集理论; 基于属性的访问控制; 访问控制

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2021077

Permission clustering-based attribute value optimization

WU Wenchao^{1,2}, REN Zhiyu¹, DU Xuehui¹

1. Information Engineering University, Zhengzhou 450001, China
2. 31668 Unit PLA, Xining 810000, China

Abstract: In new large-scale computing environment, the attributes of entities were massive and they had complex sources and uneven quality, which were great obstacles to the application of ABAC (attribute-based access control). The attributes were also hard to be corrected manually, making it difficult to be applied in access control system straightly. To solve the optimization problem of nominal attributes, a novel algorithm of attribute value optimization based on permission clustering was designed, in which entities were presented by the privilege set related to them. So that the entities were tagged by density-based clustering method with distances of their privilege set presentations. Then the attribute values were reduced and corrected based on rough set theory. Finally, the algorithm was verified on UCI data sets, which proved that after applying it, ABAC policy mining was improved in the evaluation criteria, such as the true positive rate and $F1$ -score.

Keywords: attribute value optimization, rough set theory, ABAC, access control

收稿日期: 2020-04-08; 修回日期: 2020-08-07

通信作者: 任志宇, ren_ktzy@163.com

基金项目: 国家重点研发计划 (2018YFB0803603); 国家自然科学基金 (61702550, 61802436)

Foundation Items: The National Key R&D Program of China (2018YFB0803603), The National Natural Science Foundation of China (61702550, 61802436)

论文引用格式: 毋文超, 任志宇, 杜学绘. 基于权限聚类的属性值优化[J]. 网络与信息安全学报, 2021, 7(4): 175-182.

WU W C, REN Z Y, DU X H. Permission clustering-based attribute value optimization[J]. Chinese Journal of Network and Information Security, 2021, 7(4): 175-182.

1 引言

在云计算、大数据等新型计算环境下, 实体数量繁多、安全需求复杂、权限动态变化, 以及传统的访问控制模型难以直接适用于新型计算环境, 基于属性的访问控制 (ABAC, attribute-based access control) 机制以属性作为权限判决的依据, 可以较好地满足大规模环境下复杂的访问控制需求^[1]。作为权限判决的基本依据, 属性质量的高低直接关系到 ABAC 能否实施及其实施的效率^[2]。然而在新型计算环境中, 实体的属性来源复杂、质量参差不齐, 需要对属性进行预处理以获取优质决策属性集^[1]。

属性约简可从属性集中选出完备的较小的属性子集, 目前已经有较多关于属性约简的方法, 但这些方法无法获取比原属性集更加完备的属性子集, 约简后的属性集质量依赖于原属性集中的属性质量, 无法修正标定错误的属性。在云计算、大数据环境中, 属性数量多、来源复杂、质量参差不齐, 人工修正属性标定的难度极大, 亟须研究自动化的属性标定质量优化技术。

属性按其性质可以分为标称属性、序数属性、区间属性和比率属性。其中标称属性只对相等和不等操作有意义, 性质简单、易于分析, 其他类型属性的约束都可以用等价的标称属性约束代替, 因此标称属性被广泛应用于 ABAC 策略挖掘中^[3-5]。但真实环境中的属性类型不全为标称属性, 属性标定质量良莠不齐, 现有的相关研究集中在将其他类型属性转化为标称属性上, 缺乏对标称属性值本身进行优化的研究。如果属性标定质量过低, 会导致策略臃肿、策略执行效率低, 甚至无法构建正确完备的策略。针对该问题, 本文提出了一种基于主客体权限关系聚类的属性值优化方案, 通过权限信息对实体进行聚类, 而后基于聚类中实体属性的分布信息, 为低质量的标称属性调优。

2 相关工作

属性预处理是 ABAC 策略挖掘的重要前置工作, 其目标是获取用以访问控制决策的最优属性集。它可以抽象为决策表中属性的约简优化, 主

要分为两方面: 一方面是属性数量约简, 目标是从蕴含冗余信息的原始属性集中挑选出最小的属性子集, 同时保证该属性集维持原始属性集的权限区分能力^[6]; 另一方面是属性值优化, 包括将属性转化为易于处理的类型、合并冗余属性值以及修正错误的属性标定, 使属性能够清晰地表达权限关系。

邓大勇等^[7]对属性约简准则进行了归纳, 证明了以条件属性信息熵为约简准则的约简与绝对约简等价、在一致决策表中以联合熵为约简准则的约简与绝对约简等价; 指出了其他类型的约简仅仅不损失本约简准则下的信息, 但可能损失条件属性信息熵下的信息。属性约简结果往往不是唯一的, 研究者通常使用启发式算法获取其中的一个, 而难以从多个约简中选出最优。邓大勇等^[8]提出了属性约简重心的概念, 从而选择距离重心最近的属性约简作为最优的属性约简, 并通过实验进行了证实。张维等^[9]提出了一种处理部分标记数据的粗糙集属性约简算法, 针对有标记数据获取困难的问题, 基于半监督协同学习理论为半监督数据设计了属性约简算法, 该算法利用大量无标记数据提高了有限有标记数据属性约简的质量, 但损失了约简的性能。曾孝文等^[10]提出了一种基于粗糙集理论的属性值约简改进算法, 将不相容的决策表划分为若干个相容的子决策表, 而后在子决策表中尝试删除属性值来消除不相容, 该算法会产生属性语义上的损失。

本文针对传统的属性预处理方法在属性值优化方面存在的问题, 提出了一种基于权限聚类的属性值优化算法, 对实体按照权限关系进行聚类, 而后通过分析聚类内属性值的分布, 对属性值进行优化, 去除冗余值、修正错误值, 使属性能够更加有效地应用于访问控制。

3 问题描述

3.1 属性

属性被用来描述实体的性质, 如教师的任教科目、学生的班级、试卷的科目等。研究者通常使用如下性质区分属性的类型。

(1) 相异性: = 和 $\neq$ 。

(2) 有序性: <、 $\leq$ 、> 和 $\geq$ 。

(3) 可加性：+ 和 -。

(4) 可乘性：× 和 ÷。

属性按以上性质可以分为标称属性、序数属性、区间属性和比率属性^[11]，它们的特点和常见的示例如表 1 所示。

表 1 属性分类
Table 1 Classification of attributes

属性类型	性质	示例
标称属性	比较相等和不相等是有意义的	名字、性别
序数属性	比较大小是有意义的	{优、良、中、差}, {大、中、小}
区间属性	差是有意义的	日期、温度
比率属性	差和比率都是有意义的	质量、计数

标称属性仅对相等和不相等操作有意义，基于标称属性构造的策略形式最为简单直观，现有的策略构建与验证方法大多依赖于高质量标称属性，在属性质量较低时表现不佳。因此本文针对标称属性值的优化进行研究，从而为策略构建与维护提供有力保障。

3.2 基于粗糙集的属性优化问题

属性优化可以抽象为决策表的属性值优化问题。粗糙集理论是处理不完整、不确定知识的数学工具，该理论认为知识的本质是分辨能力，它也是决策表属性约简的重要工具。本文根据粗糙集理论将属性优化问题定义为信息系统中属性的优化问题。

定义 1 信息系统

$$IS = \langle S, A \cup \{d\}, V, f \rangle$$

其中， S 是主体集，也被称为论域， A 为主体属性集， d 为分析主体权限关系得到的主体的类标签， V 为属性的值域， f 为属性值分配关系。对于 $\forall a \in A \cup \{d\}$, $s \in S$, V_a 表示属性 a 的值域， $f(s, a)$ 表示主体 s 的 a 属性的值， $f(s, a)$ 为标称值，且 $f(s, a) \in V_a$ ，简写作 $s.a$ 。

属性优化就是通过分析信息系统中主体的属性关于主体类标签的分布，尝试对属性值进行合并、修正，得到新的属性值域 V' 和属性值分配关系 f' ，使属性能更加简单有力地区分属性，本文采用支持度来描述属性对于权限的区分能力。

定义 2 不可分辨关系

对于属性集合 $B \subseteq A$ ，关于 B 的不可分辨关系为

$$IND(B) = \{(s_1, s_2) \in S \times S \mid \forall a \in B, s_1.a = s_2.a\}$$

$IND(B)$ 产生的等价类集合记作 S/B 。

定义 3 下近似集

对于属性集合 $B \subseteq A$ ，主体集合 $X \subseteq S$ ， X 关于 B 的下近似集为

$$\underline{B}X = \{x \in S \mid [x]_B \subseteq X\}$$

其中， $[x]_B$ 表示 x 关于 $IND(B)$ 的等价类。

定义 4 主体属性集合 $B \subseteq A$ 对于类标签的支持度

$$K(B) = \frac{|\text{Pos}_B(\{d\})|}{|S|}$$

其中， $\text{Pos}_B(\{d\})$ 表示 $\{d\}$ 的 B 正域， $\text{Pos}_B(\{d\}) = \bigcup_{X \in S/\{d\}} \underline{B}X$ 。 $K(B) \in [0, 1]$ ， $K(B)$ 越大，表示属性集对权限的分辨能力越强。

4 基于权限聚类的属性值优化

在 ABAC 中，属性是 ABAC 进行权限判决的依据，权限判决可以看作以属性为特征，判决结果为标签的分类过程。但根据权限判决结果分类所得的类别只有两个，难以利用粗糙集理论对属性进行有效的优化。因此本文根据主客体的权限关系中主体对应的权限集合，为主体分类赋予足够的类别标签，以便通过粗糙集理论对主体属性进行优化，使其更加清晰准确地区分主体的权限（客体属性的优化与之相同，将客体按对应的权限集合进行分类，而后优化，下文均以主体属性的优化为例）。

4.1 构造主体的权限集合表示

针对属性优化而对主体进行的分类的最好依据是主体的权限。记主体权限关系为 SP ，任意主体、动作、客体三元组 $\langle s, a, o \rangle \in SP$ 表示主体 s 可以对客体执行动作 o ，从而定义主体的权限集合，其表示如定义 5。

定义 5 主体 s 的权限集合表示

$$SPSet(s) = \{\langle a, o \rangle \mid \langle s, a, o \rangle \in SP\}$$

例如, 在系统中, 有主体 s_1 、 s_2 、 s_3 , 有客体 o_1 、 o_2 、 o_3 , 假设在主体权限关系中有如下元组:

$$\{ \langle s_1, a_1, o_1 \rangle, \langle s_1, a_2, o_2 \rangle, \langle s_2, a_1, o_1 \rangle, \langle s_2, a_2, o_2 \rangle, \langle s_3, a_1, o_3 \rangle \}$$

可以得到 s_1 、 s_2 、 s_3 的权限集合表示如下:

$$\begin{aligned} s_1 &\leftarrow \{ \langle a_1, o_1 \rangle, \langle a_2, o_2 \rangle \} \\ s_2 &\leftarrow \{ \langle a_1, o_1 \rangle, \langle a_2, o_2 \rangle \} \\ s_3 &\leftarrow \{ \langle a_1, o_3 \rangle \} \end{aligned}$$

在一定程度上, 可以认为 s_1 和 s_2 属于一类, 因此其用于进行权限判断的相关属性也应该具有相似性。

4.2 主体聚类

聚类的目的是为主体分配类别标签, 为属性优化提供依据。聚类的依据为主体的权限, 即该主体能访问的客体的集合, 因此采用 Jaccard 距离作为距离标准。

定义 6 Jaccard 距离

$$d_j(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

其中, $J(A, B)$ 表示 Jaccard 系数。

DBSCAN 是一种基于密度的聚类算法, 它基于中心来定义密度, 数据集中某个点的密度通过对该点长 ϵ 的半径之内的点计数 (包括本身) 来估计, 每个核心点 ϵ 邻域应该具有的最少数据点数由参数 $\min_pts$ 确定。相比于最常用的算法 k -means, 它不需要指定聚类的数目, 且对于噪声的容忍能力较强, 能够处理任意形状和大小的簇、高效地对主体按权限集合进行聚类。

4.3 属性值优化

属性值可能存在的问题分两方面。一方面是多个相同意义的值导致的冗余; 另一方面是属性值错误。因此属性优化值也包含两步: 合并冗余值和修正错误值 (错误标定的属性值)。

第一步是合并冗余值。实体的属性不可避免地存在相同意义的值, 以客体的属性为例, CPU 和中央处理器具有相同的含义, 但如果在进行策略制定时没有考虑到这一点, 就会导致策略臃肿。对于具有相同意义的属性值, 直接将其合并。在信息系统中, 判断属性值是否具有相同意义的方法是查看属性值区分类标签的能力是否相同, 即

属性值关于类标签的分布是否相同。

因此在合并冗余值阶段, 将在相同聚类集合内出现过的属性值进行合并。记 $ds(v) = \{s.cid \mid s.a = v\}$ 为有属性值 v 出现的类标签的集合, 对于 $v_1, v_2 \in V_a$, 若 $ds(v_1) = ds(v_2)$, 说明 v_1 和 v_2 关于类标签的分布相同, 其对应的权限相同, 在权限区分方面的语义也应该相同, 因此可以进行合并。

第二步是修正错误值。对于错误标定的属性值, 使用回溯法尝试对其进行修正。依据定义 4 中定义的条件属性集对类标签的支持度 (记 A 为待优化属性集, $K(A)$ 为属性集 A 对类标签的支持度) 来判断修正属性值的尝试是否会造成属性集对权限区分能力的损害。如果某个修正会导致支持度下降, 那么这个修正就是不可取的。

对 $\forall a \in A$, 计算 $K(A - \{a\})$, $K(A - \{a\})$ 越小, 表示失去属性 a 后, 属性集的权限区分能力越弱, 属性 a 对权限区分的贡献越大。本文按属性对权限区分贡献从大到小依次处理, 对权限区分没有贡献的属性为非授权属性, 不参与到后续的策略制定; 考虑到属性错误标定通常为偶发事件, 仅对低频属性值作修正尝试, 尝试将其合并到在相同聚类中出现过的其他值上, 并查看支持度是否下降。对于低频属性值 $v_1 \in V_a$, 若存在 $v_2 \in V_a$, 使 $ds(v_1) \cap ds(v_2) \neq \emptyset$, 且 v_1, v_2 合并后, $K(A)$ 不下降, 则将 v_1 合并到 v_2 。

5 算法实现

5.1 主体按权限聚类

将主体按其具有的权限聚类的过程如算法 1 所示, 首先根据主体权限关系, 统计出每个主体具有的权限, 保存入 `sor` 字典 (第 1 行), 而后初始化所有主体的聚类标识为 UNCLASSIFIED (第 3 行), 存入 `clusters` 字典, 继而对主体进行基于密度的聚类 (第 4~18 行)。

算法 1 主体聚类

输入 SP, ϵ , $\minPts$

输出 clusters

1) `sor` $\leftarrow$ `getSubObjRel(SP)`

2) `cid` $\leftarrow$ 0

3) `clusters` $\leftarrow$ `makeDict(sor.keys, UNCLASSIFIED)`

```

4) while numOfUnclassified(clusters) > 0 do
5)   cur ← popUnclassifiedSub(clusters)
6)   neighbors ← getNeighbors(cur, sor, eps)
7)   if len(neighbors) < minPts then
8)     clusters[cur] ← NOISE
9)     continue
10)  else
11)    clusters[cur] ← cid
12)    mark(clusters, neighbors, cid)
13)    while len(neighbors) > 0 do
14)      curNeighbor ←
        neighbors.dequeue()
15)      tmpNeighbors ←
        getNeighbors(curNeighbor, sor,
          eps)
16)      appendUnclassified(neighbors,
        tmpNeighbors, clusters)
17)      mark(clusters, neighbors, cid)
18)      cid ← cid + 1
19) return clusters

```

5.2 合并同分布属性值

这一步把每个属性同分布的值进行合并，如算法2所示，首先统计每个属性值在哪些聚类中出现过，以及出现的频率（第1行），分别存入ds和freq，将具有相同分布的属性值保存到一起（第3~9行），存入二维列表dealt，然后对同分布属性值进行合并（第10行）。为了加快聚类集合的比较，将ds按聚类集合的长度排序，只比较长度相同的集合。

算法2 同分布属性值合并

输入 clusters, sar
输出 newSar

```

1) ds, freq ← calcStatics(clusters, sar)
2) sortByLength(ds)
3) dealt ← emptyList()
4) while len(ds) > 0 do
5)   curVal, curDs ← ds.pop(0)
6)   line ← [curVal]
7)   line.append(sameDs(curVal, ds))
8)   ds.remove(line)
9)   dealt.append(line)

```

```

10) newSar ← updateSar(sar, dealt)
11) return newSar

```

5.3 低频属性优化

这一步尝试将频率低于阈值的属性值合并到在同一聚类内出现过的其他值上，如算法3所示，首先初始化A保存所有的属性名称（第1行），然后计算去掉某个属性后属性集对聚类标签的支持度，存入AttrKDict（第2行），接着将属性名按照AttrKDict中的支持度从低到高进行排序，存入队列B（第3行），进而依次从B中取出属性，找出该属性的低频属性值，尝试将其修改为该属性的其余值，若支持度不下降，则将值对存入pairs列表（第5~13行），最后依据pairs对newSar进行更新（第14行）。

算法3 低频属性优化

输入 clusters, newSar
输出 newSar

```

1) A ← getAllAttrNames(sar)
2) AttrKDict ← calcKForEachAttr(A, 1)
3) B ← getAttrQueueSortByK(AttrKDict)
4) while len(B) > 0 do
5)   curAttr ← B.dequeue()
6)   curK ← AttrKDict[curAttr]
7)   totalK ← calcK(newSar, clusters, A)
8)   if curK == totalK then
9)     A.pop(curAttr)
10)  else
11)    valFreqDict, ds ←
      calcStatics(clusters, newSar,
        curAttr)
12)    lowVals ← getLowfreqVals
      (valFreqDict)
13)    pairs ← verifyModify(newSar,
      lowVals, valFreqDict, ds, A, curK)
14)    update(newSar, pairs)
15) return newSar

```

6 实验验证

6.1 实验环境与数据集

实验环境为 Ubuntu 18.04 操作系统，配置为 Intel Core i7-8750H CPU，16 GB 内存；使

用 Python3.7 编写代码, 实现了基于 Jaccard 相似度的 DBSCAN 算法; 编写了基于权限聚类的属性优化算法。

实验数据集为 UCI 的 Amazon Access Samples 公开数据集^[12], 数据集信息如表 2 所示, 其中 HOST, PERMGROUP, SYSTEM_SUPPORT 为 3 种不同的客体, 值为 ID, PERSON_ID 为主体 ID。

6.2 实验结果

实验 1 聚类参数选取

为了保证属性优化的效果, 所得的聚类数目必须合适, 同时被标记为噪声点的主体不能太多。表 3 和表 4 分别为调整参数 ϵ 和 $\min_pts$ 对应的噪声点占比情况和聚类数目情况, 可以看出, 随着 ϵ 增大, 噪声点占比逐渐下降, 聚类数目逐渐减少; 随着 $\min_pts$ 增大, 噪声点占比逐渐增大, 聚类数目逐渐减少; 在 ϵ 较大时, $\min_pts$ 变化对噪声点占比和聚类数目的影响都会减小。

当 ϵ 取 0.6 和 0.7 时, 虽然噪声点占比不高, 但聚类数目只有 3 个, 对主体的区分能力太弱; 当 ϵ 取小于 0.5 的值时, 有超过 0.01 的主体无法被分类, 因此 ϵ 取 0.5。在 ϵ 取 0.5 的前提下, 当 $\min_pts$ 取大于 5 的值, 均有超过 0.01 的

主体无法被分类, 对后续优化影响较大, 因此 $\min_pts$ 取 5。

表 3 调整参数 ϵ 和 $\min_pts$ 对应的噪声点占比情况
Table 3 The corresponding proportion of noise points by adjusting parameters ϵ and $\min_pts$

ϵ	$\min_pts$				
	5	10	20	30	40
0.1	0.132 1	0.171 8	0.214 6	0.243 7	0.265 7
0.2	0.101 7	0.145 0	0.188 3	0.219 9	0.245 7
0.3	0.046 5	0.069 2	0.101 0	0.123 1	0.139 3
0.4	0.014 4	0.021 2	0.041 5	0.055 7	0.066 2
0.5	0.009 0	0.014 3	0.030 2	0.040 5	0.047 5
0.6	0.000 0	0.000 1	0.000 2	0.000 3	0.000 3
0.7	0.000 0	0.000 0	0.000 0	0.000 0	0.000 0

实验 2 属性优化结果

应用算法完成属性值优化后, 利用 FPGrowth 算法对主体权限关系进行策略挖掘, 并比较优化前后 FPGrowth 算法在真阳性率 (TPR, true positive rate) 和 F1 得分等评价标准上的表现。本文使用的数据集中权限关系相比于所有主客体的组合特别稀疏, 因此采用对稀疏访问支持更好的关联规则挖掘算法对算法进行验证^[4], 利用 pyfim 库^[13]中的 FPGrowth 算法来进行策略挖掘, 如图 1、

表 2 数据集信息
Table 2 Information of datasets

属性名	属性信息	属性值个数
HOST	客体属性, 一种客体的 ID	2 214
PERMGROUP	客体属性, 一种客体的 ID	9 117
SYSTEM_SUPPORT	客体属性, 一种客体的 ID	21 921
PERSON_BUSINESS_TITLE	主体属性, 主体职衔	4 979
PERSON_BUSINESS_TITLE_DETAIL	主体属性, 主体职衔详细内容	56
PERSON_COMPANY	主体属性, 主体所属公司	49
PERSON_DEPTNAME	主体属性, 主体所属部门	405
PERSON_ID	主体属性, 主体 ID	36 063
PERSON_JOB_CODE	主体属性, 主体职位码	13
PERSON_JOB_FAMILY	主体属性, 主体职位类别	70
PERSON_LOCATION	主体属性, 主体所处位置	1
PERSON_MGR_ID	主体属性, 主体所属经理 ID	3 207
PERSON_ROLLUP_1	主体属性, 主体分组 1	12
PERSON_ROLLUP_2	主体属性, 主体分组 2	111
PERSON_ROLLUP_3	主体属性, 主体分组 3	12

图2所示。本文提出的属性值优化算法对 FPGrowth 算法挖掘授权规则的真阳性率和 F1 得分有较明显的提升。

表4 调整参数 eps 和 min_pts 得到聚类的数量情况
Table 4 The corresponding number of clusters by adjusting parameters eps and min_pts

eps	min_pts				
	5	10	20	30	40
0.1	477	261	149	106	82
0.2	448	245	146	105	80
0.3	213	132	83	69	60
0.4	71	56	38	38	36
0.5	71	56	36	31	31
0.6	3	3	3	3	3
0.7	3	3	3	3	3

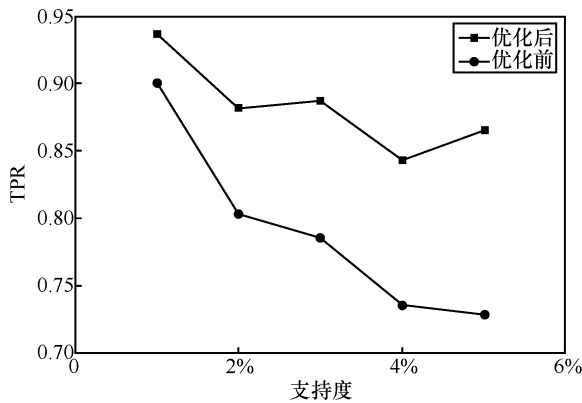


图1 优化前后 FPGrowth 算法取得的 TPR 随关联规则的支持度变化情况对比

Figure 1 Comparison of the TPR changes with the support of association rules before and after the optimization

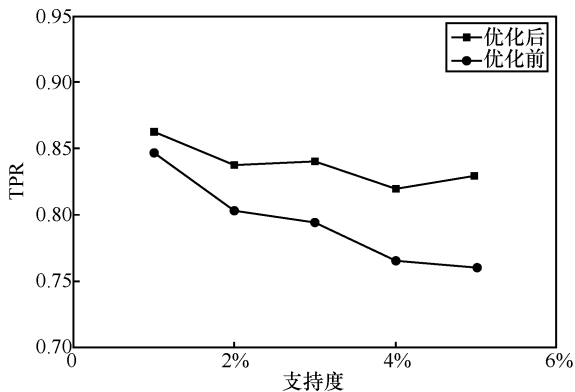


图2 优化前后 FPGrowth 算法取得的 F1 得分随关联规则的支持度变化情况对比

Figure 2 Comparison of the F1-score changes with the support of association rules before and after the optimization

实验3 性能分析

在进行实验之前, 首先将数据集中的主体权限关系作预处理, 得到所有主体的权限集合表示, 这一步的时间复杂度为 $O(N_{sp})$, 其中 N_{sp} 表示主体权限关系中主体权限元组的数量, N_{sp} 的最坏情况为 $N_s \cdot N_o \cdot N_{acts}$, 其中 N_s 为主体的数量, N_o 为客体的数量, N_{acts} 为动作的数量, 预处理的时间复杂度与读取所有主体权限元组的时间复杂度相同, 该开销是不可避免的。下面展示算法处理预处理后数据的性能分析。

图3为随主体数量增大算法所消耗时间的变化情况, 从图3中可以得到, 算法的时间消耗与主体数量的平方呈正相关。事实上, 主体按权限聚类的时间复杂度为 $O(N_s^2)$; 合并同分布属性值的时间复杂度为 $O(N_s \cdot N_{av} + N_{av}^2)$, 其中 N_{av} 为主体与授权相关的属性的值的总数量, 一般情况下, 该值小于 N_s 的常数倍, 因此这一步的时间复杂度可以化简为 $O(N_s^2)$; 低频值优化的时间复杂度为 $O(Q \cdot N_s^2)$, 其中 Q 为低频属性值的数量, 该值通常较小, 可以视为常数, 因此总的时间复杂度为 $O(N_s^2)$ 。虽然在实际的大数据环境中, 客体数量通常远大于主体数量, 算法的时间复杂度低于数据预处理的时间复杂度, 但总的时间复杂度较高, 在后续的研究中需要作进一步优化。

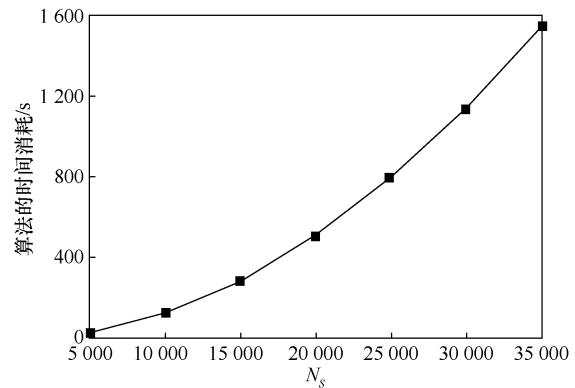


图3 主体属性优化消耗时间随主体数量变化情况
Figure 3 Time consumed by subject attribute optimization changes with the number of subjects

7 结束语

针对新型大规模计算环境下属性繁多、质量差强人意、难以人工修正、难以直接用于 ABAC

策略构建的问题,本文设计了基于权限聚类的属性值优化算法。首先,基于实体的权限集合表示和 DBSCAN 聚类算法为实体分配与权限对应的类别标签;接着,基于粗糙集理论对属性值进行化简与修正,使其能够更加清晰有力地区分实体的权限、高效地用以 ABAC 的实施与维护;最后,在 UCI 公开数据集上对属性值优化的效果进行了验证,根据本文提出的属性值优化算法对数据集的主体属性进行优化后, FPGrowth 算法挖掘到的 ABAC 策略质量在真阳性率和 F1 得分上有较为明显的提升。但是,算法中属性集对类标签的支持度计算复杂度较高,在处理大规模数据集的性能有待进一步优化,下一步计划研究支持度的增量式更新。

参考文献:

- [1] 房梁,殷丽华,郭云川,等. 基于属性的访问控制关键技术研究综述[J]. 计算机学报, 2017, 40(7): 1680-1698.
FANG L, YIN L H, GUO Y C, et al. A survey of key technologies in attribute-based access control scheme[J]. Chinese Journal of Computers, 2017, 40(7): 1680-1698.
- [2] HU C T, FERRAILOLO D F, KUHN D R, et al. Guide to attribute based access control (ABAC) definition and considerations [includes updates as of 02-25-2019][R]. 2019.
- [3] XU Z, STOLLER S D. Mining attribute-based access control policies from logs[C]//IFIP Annual Conference on Data and Applications Security and Privacy. 2014: 276-291.
- [4] COTRINI C, WEGHORN T, BASIN D. Mining ABAC rules from sparse logs[C]//2018 IEEE European Symposium on Security and Privacy (EuroS&P). 2018: 31-46.
- [5] IYER P, MASOUMZADEH A. Mining positive and negative attribute-based access control policy rules[C]//Proceedings of the 23rd ACM on Symposium on Access Control Models and Technologies. 2018: 161-172.
- [6] 张任伟,白晓颖,郁莲,等. 决策表的属性约简算法综述[J]. 计算机科学, 2011, 38(11): 1-6.
ZHANG R W, BAI X Y, YU L, et al. Survey of decision table research of attribute reduction[J]. Computer Science, 2011, 38(11): 1-6.
- [7] 邓大勇,薛欢欢,苗夺谦,等. 属性约简准则与约简信息损失的研究[J]. 电子学报, 2017, 45(2): 401-407.
DENG D Y, XUE H H, MIAO D Q, et al. Study on criteria of attribute reduction and information loss of attribute reduction[J]. Acta Electronica Sinica, 2017, 45(2): 401-407.
- [8] 邓大勇,葛雅雯,黄厚宽. 属性约简簇的优化选择[J]. 电子学报, 2019, 47(5): 1111-1120.
DENG D Y, GE Y W, HUANG H K. An optimizing selection in a family of attribute reducts[J]. Acta Electronica Sinica, 2019, 47(5): 1111-1120.
- [9] 张维,苗夺谦,高灿,等. 一种处理部分标记数据的粗糙集属性约简算法[J]. 计算机科学, 2017, 44(1): 25-31.
ZHANG W, MIAO D Q, GAO C, et al. Rough set attribute reduction algorithm for partially labeled data[J]. Computer Science, 2017, 44(1): 25-31.
- [10] 曾孝文,胡虚怀,严权峰. 一种基于粗糙集理论的属性值约简改进算法[J]. 电子技术, 2017(1): 1.
ZENG X W, HU X H, YAN Q F. An improved algorithm for attribute value reduction base on rough set theory[J]. Electronics Technology, 2017(1): 1.
- [11] TAN P N, STEINBACH M, KUMAR V. Introduction to data mining[M]. Pearson Education India, 2016: 21.
- [12] LICHMAN M. UCI machine learning repository. amazon access samples data set[EB]. 2011.
- [13] BORGELT C. Frequent item set mining[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2012, 2(6): 437-456.

[作者简介]



毋文超(1995-),男,河南焦作人,中国人民解放军 31668 部队助理工程师,主要研究方向为授权管理与访问控制。



任志宇(1974-),女,河南汤阴人,博士,信息工程大学副教授,主要研究方向为网络与信息安全。



杜学绘(1968-),女,河南新乡人,博士,信息工程大学教授、博士生导师,主要研究方向为网络与信息安全、空间信息网络、云计算安全。