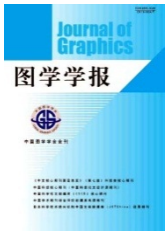


图学学报
Journal of Graphics
ISSN 2095-302X,CN 10-1034/T

《图学学报》网络首发论文

题目：基于属性相似性度量的 BIM 构件聚类
作者：王万齐，马宝睿，李倩，卢文龙，刘玉身
收稿日期：2019-09-10
网络首发日期：2020-04-02
引用格式：王万齐，马宝睿，李倩，卢文龙，刘玉身．基于属性相似性度量的 BIM 构件聚类．图学学报．
http://kns.cnki.net/kcms/detail/10.1034.T.20200401.1647.034.html

图 学 学 报
JOURNAL OF GRAPHICS
建筑与城市信息模型
2020-2

基于属性相似性度量的 BIM 构件聚类

王万齐[1]， 马宝睿[2]， 李 倩[2]， 卢文龙[1]， 刘玉身[2]

(1. 中国铁道科学研究院集团有限公司电子计算技术研究所，北京 100081；
2. 清华大学软件学院，北京 100084)

摘 要：近年来，随着建筑信息模型(BIM)构件库资源在互联网上迅猛增长，对大量 BIM 构件资源的聚类和检索应用变得日益迫切。现有方法还缺乏对 BIM 构件所承载的领域信息提取，基于 BIM 构件所承载的领域信息，对 BIM 构件库资源开展聚类研究：①针对 BIM 构件，提出了一种基于属性信息量的 BIM 构件相似性度量算法，以充分利用 BIM 构件属性信息。通过与传统的 Tversky 相似性度量算法以及几何形状相似匹配算法相比，其在相似性度量上效果更好。②基于 BIM 构件间的相似性度量算法，提出了一种 BIM 构件库聚类方法。并在 BIMSeek 检索引擎中集成了 BIM 构件的关键字检索功能以及分类器查看功能，为用户提供更丰富的检索和查看方式。通过与传统的 K-medoids 和 AP 聚类算法相比，其聚类方法效果更好。

关 键 词：建筑信息模型；工业基础类；信息检索；相似性度量；聚类
中图分类号：TU 17 **DOI**：10.11996/JG.j.2095-302X.2020020000
文献标识码：A 文 章 编 号：2095-302X(2020)02-0000-00

Clustering of BIM components based on similarity measurement of attributes

WANG Wan-qi[1], MA Bao-rui[2], LI Qian[2], LU Wen-long[1], LIU Yu-shen[2]

(1. Institute of Computing Technology, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China;
2. School of Software, Tsinghua University, Beijing 100084, China)

Abstract: In recent years, resources in the Building Information Modeling (BIM) components library are expanding rapidly on the Internet. There is an increasing demand for ways to cluster and retrieve appropriate BIM components among countless resources. However, the way to extract domain information of BIM components still can not be found in existing methods. This paper studies the clustering of BIM components based on the domain information of BIM components: ①For BIM components, tan algorithm measuring similarity is proposed based on the attribute information. Compared with the traditional Tversky similarity measure algorithm and geometry similarity matching algorithm, the newly proposed one the present study has produced a better result. ②A clustering method of BIM component library is proposed based on the similarity measure algorithm of BIM components. Users are provided with diverse ways to retrieve and check information thanks to the search engine of BIMSeek integrated with functions of keyword-based retrieval and classifier view. Compared with the K-medoids algorithm and AP algorithm, the results of ours are more desirable.

Keywords: building information modeling; industry foundation class; information retrieval; similarity measure; clustering

收稿日期：2019-09-10；**定稿日期**：2019-10-14
基金项目：国家重点研发计划资助项目(2018YFB0505400)；国铁集团科技研究开发计划项目(K2018G055)；高速铁路基础设施BIM核心平台研发可行性方案研究项目(2018YJ121)
第一作者：王万齐(1978–)，男，甘肃环县人，副研究员，博士。主要研究方向为建筑信息模型与应用等。E-mail：13701314627@163.com
通信作者：刘玉身(1976–)，男，辽宁瓦房店人，副教授，博士。主要研究方向为计算机图形学与建筑信息模型。E-mail：liuyushen@tsinghua.edu.cn

近几十年来，继声音、图像、视频之后，三维模型作为第四代多媒体资源，已被广泛地应用于机器学习、虚拟现实等领域，大量可共享的三维模型在互联网上迅猛增加[1]。由于采用多媒体检索技术可以提高开发效率、缩短开发周期、节省开发成本，因此得到了众多研究人员的重视，特别是在CAD 工程制图设计领域。

随着 BIM 在 AEC 领域的迅猛崛起，互联网涌现出大量的 BIM 资源库，目前比较主流的有 Autodesk Seek，BIM Object，National BIM Library，Modlar，SmartBIM，Arcat，RevitCity 网站等。这些网站中少则拥有几千个多则拥有几万个 BIM 构件，面对如此日益庞大的三维模型库，设计人员需要将主要精力从如何构建三维模型转变为如何基于已有的模型构建出符合需求的新模型的问题上。GUNN[2]在美国科学杂志上发表文章表示，40%的构件可以在已有的模型之上重新设计，40%的构件可以修改已有的模型，仅有 20%的构件需要重新设计。ULLMAN[3]认为超过 75%的设计可以复用以前的设计来满足新的需求。由此可见，构件复用的需求量相当大。如何快速准确地查找到满足设计人员需求的构件，实现设计资源的重复利用，成为当前的热点研究问题[4]。

聚类的最初目的是将具有相似特征的物体放在一起[5]。聚类分析有 4 个功能：①对数据分类进行进一步扩展；②对归类进行概念性探索；③通过探索数据形成假说；④对实际的数据集归类假说的测试方法。一般而言，聚类是对数据集分成若干个簇的过程。所以对 BIM 构建进行聚类有利于生成更好的检索结果。

基于上述分析，本文针对如何快速准确查找符合设计需求的三维模型的问题，提出了一种 BIM 构件库聚类方法。并在 BIMSeek 检索引擎中集成了 BIM 构件的关键字检索功能以及分类器查看功能，为用户提供更丰富的检索和查看方式。

1 相关工作

由 IAI(International Alliance for Interoperability)组织定义的 IFC(industry foundation classes)国际标准是 BIM 的最主要数据交换标准[6]。因此，本文使用 IFC 文件表示 BIM 构件，展开对 BIM 构件的聚类研究。

聚类研究方法包括：基于划分的方法，将每个样本划分为一个归属，例如 K-means[7]，EM[8]，K-medoids[9]；基于层次的方法，创建层次，递归将样本合并或拆除，例如 BIRCH[10]，CUBE[11]，ROCK[12]；基于密度的方法，区域中点的密度大于阈值时，将其加入到最近的类簇中，例如 DBSCAN[13]，OPTICS[14]；基于网格的方法，将数据空间量化为网格单元，将样本点分配到相应网格中，例如 WaveCluster[15]；基于模型的方法，为每个类簇定义一个模型，根据给定模型为每个样本点选择合适模型，例如 SOM[16]。

对 BIM 构件的聚类研究有很多应用，例如将 BIM 构件聚类应用到对 BIM 信息的挖掘和噪声数据的检测[17-18]；将 BIM 聚类应用到对缺少标注的模型提取有用信息；本文将 BIM 构件的聚类算法应用到检索，集成到 BIMSeek 检索引擎中完成检索和分类器查看功能。

之前部分工作是在 BIM 领域做检索的研究[19-21]，而本文则是应用于 BIM 构件自身上。其结合复杂的语义信息减少数据集成的不一致性，是结合语义构建领域知识[22-24]，本文工作是结合语义信息进行聚类和检索。

在传统的三维模型检索领域中，主要通过提取模型的几何特征来构建向量，但是对于工程设计领域的三维模型，不仅包括几何特征，还包含语义属性，因此，仅通过提取几何特征是不足以描述整个模型。而基于模型本身内容的三维模型检索可以更好地支持针对 BIM 构件展开聚类的研究。

本文从 Arcat、Autodesk Seek 和 BIM Object 网站上提取了一万个 BIM 构件，对其开展检索与聚类的研究，首先提出了一种基于属性信息量的 BIM 构件相似性度量方法。基于 BIM 构件间的相似性度量算法，本文提出了一种 BIM 构件库聚类方法，并将聚类结果应用于检索结果分类展示中，从而生成更好的检索聚类效果。同时，为了给用户提供更丰富的检索和查看方式，本文在 BIMSeek 检索引擎中集成了 BIM 构件的关键字检索功能以及分类器查看功能。

2 方法

针对BIM构件的相似性度量方法，提出了一种BIM构件库的聚类算法，首先使用近邻传播(affinity propagation, AP)算法[25]对初始种子进行选取，然后使用K-medoids算法进行聚类，在进行相

似性度量时使用本文提出的基于属性信息量的BIM构件相似性度量算法。将从多个BIM资源库中提取的构件进行聚类，并将聚类应用于检索中，实现了检索结果的分类展示以及分类器查看功能。由于使用基于属性信息量的聚类结果类别比较精细，类别比较多，需要给其聚类结果打标签作为二级聚类标签。而类别太多不易于浏览，因此，需要将聚类结果合并，并将其结果再次打标签作为一级标签。

BIM 构件库聚类算法的流程如图 1 所示。

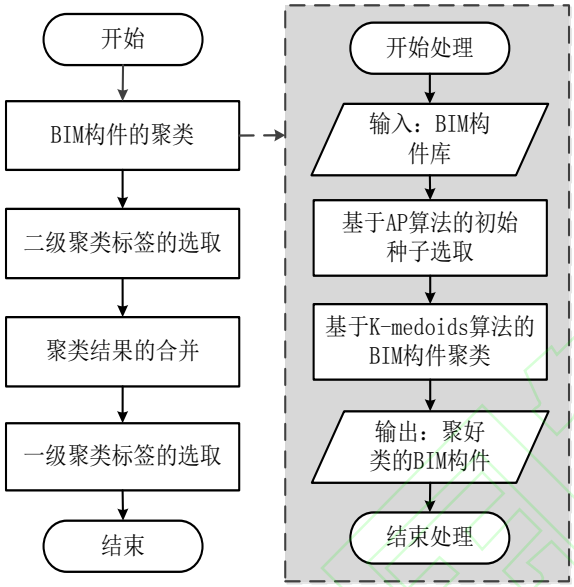


图 1　BIM 构件库聚类算法流程图

2.1　基于属性信息量的构件相似性度量

由于 IFC 文件中包含了该 BIM 构件的所有几何属性和语义属性，因此每一个 BIM 构件均需一个相应的属性向量表示，从而 BIM 构件的相似性度量即转换为构件属性向量的相似性度量。在此提出了一种基于 RESNIK[26]提出的信息量计算和 TVERSKY 和 GATI[27]相似度模型的 BIM 构件属性相似性度量算法。

本文提出 BIM 构件的语义信息量为

$$IC_{bim}(pname\_pvalue)=-\log_2\frac{count(pname\_pvalue)}{total\_bims} \tag{1}$$

其中，$count(pname\_pvalue)$ 为属性 $pname\_pvalue$ 在多少个构件中出现过；$total\_bims$ 指共有多少个 BIM 构件；$count(pname\_pvalue)/total\_bims$ 为某一属性在所有的 BIM 构件中出现的概率，取其以 2 为底的对数代表信息量。

将所有 BIM 构件的属性信息量保存到计算机中，便于后续读取使用。

由于每个 BIM 构件均被处理成一个属性向量，其既包含了几何属性(长度、宽度等)，又包含了语义属性(材质、厂商等)，本文中默认的属性权重值设置为 1，当属性名称相同时，为了保证在相似度的计算中更加精确，需要在以下 2 种情况下修改属性的权重值：①对于几何属性，设定了一个阈值为 5%，当相差比例大于 5%时为不相同属性，其权重值为 0；相差比例小于 5%的属性设定为相同属性，但其权重值按比例缩小。②对于语义属性，如果描述 2 个部件的描述词有部分匹配也认为其属性是一样的，只不过其权重相应缩小，但若 2 个属性值完全不同，那么权重值则为 0。此外，对于自定义属性，由于不同的人可能会使用不同的单词来表达同一个意思，本文使用 WordNet 来解决这种相同属性的不同表达问题，即通过同义关系得到相应的同义词列表。

本节提出基于属性信息量的 BIM 构件相似度计算公式，通过集合运算计算出任意 2 个构件之间的相似度，即

$$\begin{aligned}&\mathrm{sim}(bim_i,bim_j)\\&=\frac{\alpha f\left(I_{bim_i}\cap I_{bim_j}\right)}{\alpha f\left(I_{bim_i}\cap I_{bim_j}\right)+\beta f\left(I_{bim_i}/I_{bim_j}\right)+\lambda f\left(I_{bim_j}/I_{bim_i}\right)}\end{aligned} \tag{2}$$

其中，

$$\begin{cases}\alpha=\dfrac{\sum\limits_{k\in\left(I_{bim_i}\cap I_{bim_j}\right)}IC_k}{M}\\ \beta=\dfrac{\sum\limits_{k\in\left(I_{bim_i}/I_{bim_j}\right)}IC_k}{M}\\ \lambda=\dfrac{\sum\limits_{k\in\left(I_{bim_j}/I_{bim_i}\right)}IC_k}{M}\end{cases},M=\sum_{k\in I_{bim_i}\cup I_{bim_j}}IC_k$$

$f(I)$ 为该集合中所有属性的信息量与权重值相乘之和，即

$$f(I)=\sum_{i=1}^{n}IC_i\times W_i \tag{3}$$

其方法可读取保存在属性信息量的中间文件，找到 I 所表示的所有属性，假设 I 中属性个数为 n，将这 n 个属性的信息量和权重值相乘之后再求和；IC_i 为第 i 个属性的信息量；W_i 为第 i 个属性的权

重值。

2.2 基于相似性传播算法的初始种子选取

本文在 AP 算法的基础上，融入了对 BIM 构件的语义相似性度量。在 AP 算法运行过程中，不断地从 BIM 构件预存好的相似度矩阵中提取数据，其算法称为 Tversky-AP 算法，具体如下：

算法 1. Tversky-AP 算法

输入：BIM 构件语义相似度矩阵 simiMatrix，该矩阵为二维矩阵，simiMatrix[i][j]代表 BIM 构件 i 与 BIM 构件 j 的相似度。

输出：初步聚类的 BIM 构件 clusters。

```
1. function tverskyAP(simiMatrix):
2.   simiMatrix←rebuildSimiMatrix(simiMatrix)
3.     while 聚类结果没有稳定 do
4.       for i from 0 to N-1
5.         for j from 0 to N-1
6.           r[i][j]←updateR(r[i][j])
7.           a[i][j]←updateA(a[i][j])
8.           r[i][j]←eliminateShockR(r[i][j])
9.           a[i][j]←eliminateShockA(a[i][j])
10.        end for
11.      k←chooseClusterCenter(i)
12.      end for
13.    end while
14.    clusters←buildClusters()
15.    return clusters
16. end function
```

rebuildSimiMatrix 对输入语义相似度矩阵的重建，即

$$\boldsymbol{s}(i,k)=\begin{cases} sim(i,k), i\neq k \\ p(k), i=k \end{cases} \tag{4}$$

其中，当 $i\neq k$，使用基于属性信息量的相似度表示 $sim(i,k)$；当 $i=k$，其值称为参考度，由于本文认为所有的构件均有可能成为聚类中心，因此该参考度的值需相同，其值取自相似度矩阵的中位数。

updateR 更新式见式(5)。当吸引度矩阵均有值后，需要根据吸引度的值更新归属度的值，updateA 在 $i\neq k$ 时，更新为式(6)，在 $i=k$ 时，更新为式(7)。

$$\boldsymbol{r}(i,k)=\boldsymbol{s}(i,k)-\max_{k'\in k'\neq k}\left\{\boldsymbol{a}(i,k')+\boldsymbol{s}(i,k')\right\} \tag{5}$$

$$\boldsymbol{a}(i,k)=\min\left\{0,\boldsymbol{r}(k,k)+\sum_{i'\notin\{i,k\}}\max\left(0,\boldsymbol{r}(i',k)\right)\right\} \tag{6}$$

$$\boldsymbol{a}(k,k)=\sum_{i'\notin\{i,k\}}\max\left\{0,\boldsymbol{r}(i',k)\right\} \tag{7}$$

用 eliminateShockR 和 eliminateShockA 保证结果准确度，防止数字震荡，阻尼系数(Damping Factor) λ 的取值为 0.5。

chooseClusterCenter 可对每一个 BIM 构件 i 确定其聚类中心。若 $i=k$，则构件 i 本身是聚类中心；若 $i\neq k$，则构件 k 是构件 i 的聚类中心。每次迭代选取 $a(i,k)+r(i,k)$ 最大值对应的 BIM 构件作为聚类中心。

2.3 基于 K-medoids 算法的 BIM 构件聚类

本文将 Tversky-AP 算法的结果作为 K-medoids 算法的初始聚类中心，因此称该算法为 AP-medoids 算法，具体如下：

算法 2. AP-medoids 算法

输入：Tversky-AP 算法的结果 clusters。

输出：聚类好的 BIM 构件 idResult。

```
1. function apMedoids():
2.     clusters←readInitialSeeds(clusters)
3.     while 聚类结果没有稳定 do
4.         for i from 0 to N-1
5.             k←chooseCenter(i)
6.             clusters[k].append(i)
7.         end for
8.         for cluster in clusters
9.             chooseClusterCenter(cluster)
10.        end for
11.        clusters←updateClusters()
12.    end while
13.    idResult←changeToIdResult()
14.    return idResult
15. end function
```

readInitialSeeds 读取初始的聚类中心，这也是本文对 K-medoids 算法的改进之一。读取本节中 Tversky-AP 算法的聚类结果作为初始聚类中心。并将其表示为 $clusters=\{bim_1,bim_2,\cdots,bim_k\}$。

chooseCenter 为每一个非初始聚类中心的 BIM 构件选取初始类别，读取在 2.1 节中保存的 BIM 构件的相似度矩阵，得到每一个 BIM 构件 i

与初始的 K 个聚类中心的语义相似度，选取语义相似度最大的聚类中心作为应该属于的类。

chooseClusterCenter 计算该构件与其余构件之间的语义相似度之和，将语义相似度的和最大的构件作为该类的聚类中心。updateClusters 更新所有的聚类中心供下一次迭代使用。

原始的 K-medoids 算法的时间复杂度主要浪费在计算彼此的距离，本文算法不需要实时地计算 BIM 构件之间的相似度，而是采取了预处理的方法，这也是本文对 K-medoids 算法的第二点改进。

2.4 二级聚类标签的统计和选取

经过聚类之后，每一类 BIM 构件需要一个标签来概括该类构件，便于用户浏览。并将小类别合并成为大类别，相当于大类别的标签为一级标题，而小类别的标签为二级标题，在分类器中显示 BIM 构件时，首先看到的是一级标签，点进之后分列表显示二级标签。在标签选取后根据 WordNet 将具有相似标签描述的小类别进行一次初始合并。二级聚类标签的选取算法如下：

算法 3. 二级聚类标签的选取算法

输入： AP-medoids 聚类算法的结果 idResult。

输出： 打过二级标签的聚类结果 labelResult。

```
1. function twoLevelLabel():
2.    despResult←changeToDespResult(idResult)
3.      for desp in despReuslt
4.          for word in desp
5.              fliter(word)
6.          end for
7.      end for
8.      despResult←updateDespResult()
9.      for desp in despResult
10.         for word in desp
11.             calculateTfidf(word)
12.         end for
13.         label←maxTfidfWord()
14.         idList←changeToIdList(desp)
15.         labelResult.append(label,idList)
16.     end for
17. labelResult←mergeWithWordnet(labelResult)
18.     return labelResult
19. end function
```

changeToDespResult 即为将 id 转换成相应的构件描述信息。fliter 为对描述信息的停用词处理。停用词列表中需要去除 6 类单词：①单词中含有数字；②单词长度为 1；③常用的一些介词；④无用的形容词；⑤含特殊字符的单词；⑥人名、地名、厂商名。

calculateTfidf 和 maxTfidfWord 基于权重值进行聚类标签的选取。本文使用 TFIDF 进行权重值的赋予。使用 WordNet 中的同义词组，在为每个类别描述信息的每个单词计算出权重值之后，选取权重值最大的那个单词作为该类的标签。

mergeWithWordnet 在给聚类结果打标签之后，由于某些类别的标签依据 WordNet 是相似的，因此，可以将具有相似标签的类别进行一次初始的简单合并。例如标签 "toilet"，"lavatory" 和 "bathroom"，而这 3 个标签在 WordNet 中是同义词，如图 2 所示，而这 3 个标签的词根是 toilet，因此合并成一个大类别，使用"toilet"作为标签。

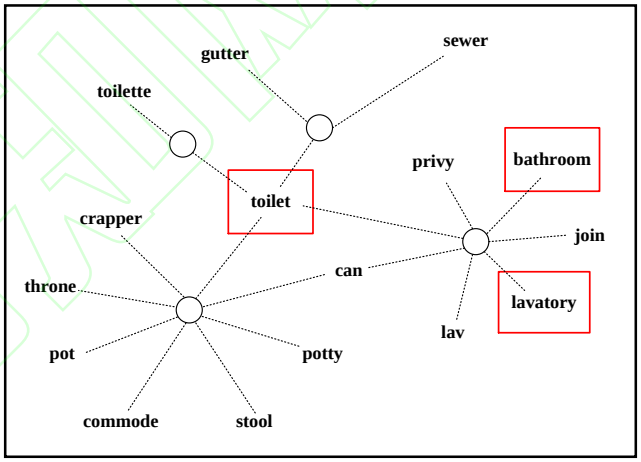


图 2 WordNet 中 toilet 的同义词示意图

2.5 聚类结果合并

由于基于属性信息量的相似度计算方法，使得聚类结果更加精细，导致聚类类别较多。例如，原本均是门，但是由于内部结构不同(双开门、单开门等)，被聚成了多个类别，而类别太多不易于浏览，因此有必要将原本相关的小类别合并成大类。

本文采用 VSM[28]向量空间模型(vector space model)进行构件描述信息相似度的比对，根据构件的描述信息的相似性进行类别的合并。基于构件描述信息的聚类合并算法如下：

算法 4. 基于聚类描述信息构件合并算法

输入： 打过二级标签的聚类结果 idResult。

输出： 经过合并的 BIM 构件聚类结果 mergedResult。

```
1. function mergeResult():
2.    despResult←changeToDespResult(idResult)
```

```
3.  despVector←buildDespVector(despResult)
4.  tfidfVector←changeToTfidfRes(despResult)
5.  n←len(tfidfVector)
6.  for i from 0 to n-2
7.      for j from i+1 to n-1
8.          simiMatrix[i][j]←calculateSimi(i,j)
9.      end for
10. end for
11. for i from 0 to n-2:
12.     for j from i to n-1:
13.         if simiMatrix[i][j] > threshod then
14.             mergedResult←merge(i,j)
15.         end if
16.     end for
17. end for
18. removeDuplicate(mergedResult)
19. return mergedResult
20. end function
```

changeToDespResult 是将打过二级聚类标签的聚类结果使用构件描述信息表示。buildDespVector 是使用向量空间模型表示构件描述信息集合。对于 BIM 构件的描述信息集合，将其进行分词，最终形成一个由"key=value"构成的描述文档向量。由于语言本身就客观存在着诸多的不确定性，本文仍使用 WordNet 表示，凡是在其中具有相同词根的单词均被认为是相同的单词。changeToTfidfRes 是通过计算向量空间模型中每个词项的权重值来构建描述信息集合的数值向量，便于相似度的计算。每个词项的权重值使用 TFIDF 来表示，其为 TF 值与 IDF 值的乘积。TF 为某一词项在文中出现的频率，IDF 为一个词项在多个文档中出现频率，代表词汇的普遍性。calculateSimi 计算 BIM 构建文档信息向量之间的相似度度量方法是余弦距离相似度。

时间复杂度分析：假设打过二级标签的聚类结果有 m 个类簇，将聚类结果转为其对应的描述信息的时间复杂度为 $O(m)$；将描述信息集合使用向量空间模型表示的时间复杂度为 $O(m)$；假设所有向量空间模型中不同的词项个数为 n，为每一个词项计算 TFIDF 的时间复杂度为 $O(m)$，那么转为 TFIDF 向量的时间复杂度为 $O(m\times m\times n)$；使用余弦相似度计算相似度的时间复杂度为 $O(n)$，因此计算任意 2 个向量之间相似度的时间复杂度为 $O(m\times m\times n)$；将相似向量合并的时间复杂度为 $O(m^2)$；去重的时间复杂度为 $O(m)$，因此总的时间复杂度为 $O(m\times m\times n)$。

算法在实现过程中的改进。对于每个向量而言，其中 0 占了绝大多数，而在计算 2 个向量的相似度时只有非 0 值才起作用，因此本文在保存 TFIDF 向量时仅仅保留非零部分，就能大大降低 n 的值，从而提高算法运行效率。

2.6 一级标签的选取

经过合并后即可得到一级聚类，且需要有一个标签来进行描述，称其为一级聚类标签，其是直接给用户进行浏览的，因此类别不能太多。由于本文的研究对象是使用 IFC 文件来表示的 BIM 构件，构件基本都隶属 IfcBuildingElement，含有 21 个子类别，可使用自然语言来表示 21 个子类别，使用 IfcBuildingElement 的子类别(以下简称 IFC 标签)来引导一级聚类标签的选取。使用 WordNet 的同义词功能，可以得到 IFC 标签的同义词列表，用该列表过滤 BIM 构件的描述信息，这样就能够起到引导聚类标签选取的效果。

一级聚类标签的选取算法如下：

算法 5. 基于聚类描述信息构件合并算法

输入：经过合并后聚类结果 mergedResult，IFC 标签列表 ifcList。

输出：打了一级标签的聚类结果 labelResult。

```
1. Fun oneLevelLab(mergedResult, ifcList):
2. despResult←changeDespResult(mergedResult)
3. ifcSynonymList←getSynonyms(ifcList)
4.     for desp in despResult
5.         for word in desp
6.             filter(word,ifcSynonymList)
7.         end for
8.     end for
9.     despReuslt←upDateDesp(despResult)
10.    for desp in despResult
11.        for word in desp
12.            calculateTfidf(word)
13.        end for
14.        label←maxTfidfWord()
15.        idList←changeToIdList(desp)
16.        labelResult.append(label, idList)
17.    end for
18.    return labelResult
19. end function
```

getSynonyms 为获取 IFC 标签的同义词列表，filter 为 BIM 构件描述信息的过滤。将描述信息进

行分词，对于每个单词使用 WordNet 计算其同义词列表，如果同义词列表中有一个单词与 IFC 标签的同义词列表中的单词相同，那么该单词保留，否则滤掉。calculateTfidf 和 maxTfidfWord 是基于权重值的聚类标签的选取。基于 WordNet 计算初始标签的同义词列表，看同义词列表中的单词与哪个 IFC 标签的同义词列表中的单词相同，就选取那个 IFC 标签作为一级聚类标签。

3 实例验证与评估

3.1 BIM 构件聚类应用于检索系统的实现

本文将 BIM 构件的聚类应用于 BIMSeek[20-21] 构件检索系统和 3DSeek[29-35]三维模型检索中，实现了对于关键子检索结果的分类展示以及分类器查看 2 个功能。将关键字的检索结果进行分类展示，便于用户浏览。

图 3 为系统首页，用户可以通过 3 种方式进行检索：①输入关键词进行检索，②点击分类查看器中的一级聚类标签进行检索，③上传 BIM 构件进行属性检索。图 4 为当输入的关键词为"window"时的查询结果示意图(分类器查看页面与其类似)，在右侧可以选择"window"下面的任意一个二级聚类标签，左侧的结果会根据二级聚类标签而变化，结果列表展示了检索结果构件的名称、类别、厂家、简要描述、属性信息、三维模型的展示以及 IFC 文件和 RFA 文件的下载。

图 3 系统首页示意图

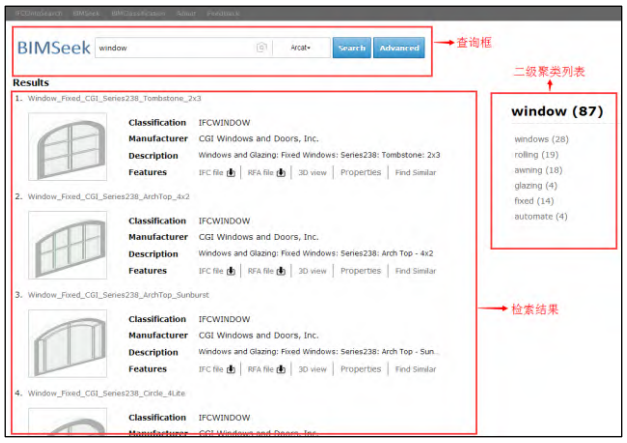
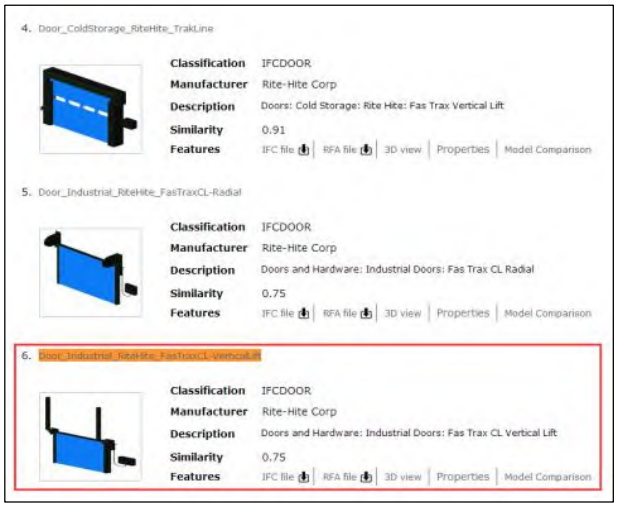


图 4 关键字检索结果示意图

针对上传 BIM 构件进行属性检索功能，例如上传一个门的 BIM 构件根据属性检索，Door_Industrial_RiteHite_FasTraxCL-VerticalLift 这个构件在使用基于信息量和 Tversky 的 BIM 构件属性相似性度量方法的结果列表中第 6 个出现，而在使用传统的 Tversky 相似性度量方法的结果列表中是第 12 个出现，如图 5 所示。由于该构件与上传构件的共同属性中包含的信息量更大，例如 Door Slab Material，Vision Panel Material 这些属性，因此该构件应当在检索列表的前面显示，此例子说明本文方法可以更好地根据属性检索到信息量更接近的模型。



(a) Door_Industrial_RiteHite_FasTraxCL-VerticalLift 构件在使用基于信息量和 Tversky 的 BIM 构件属性相似性度量方法时的出现位置

(b) Door_Industrial_RiteHite_FasTraxCL-VerticalLift 构件在使用 Tversky 相似性度量方法时的出现位置

图 5 Door_Industrial_RiteHite_FasTraxCL-VerticalLift 构件在 2 种相似度比较方法中的实例对比图

3.2 聚类结果比较

本文采用类内类外标准和 Purity 标准对聚类结果进行评判，且进行实验的数据是经过 AP-medoids 聚类之后的数据。

类内类外相似度：in_outSim 为每个类中每个 BIM 构件的类内相似度与类外相似度之差的平均值，即

$$in\_outSim=\frac{1}{n}\sum_{j=1}^{c}\sum_{i=1}^{n_j}inSim(j,i)-outSim(j,i) \quad (8)$$

其中，in_outSim 的值越大说明聚类结果越好。

Purity 标准：计算正确聚类的模型占总模型数的比例，即

$$purity(\Omega,M)=\frac{1}{N}\sum_{j=1}^{p}\max_{j}\left|w_i\cap m_j\right| \quad (9)$$

其中，N 为模型总数；$\Omega=\{w_1,w_2,\cdots,w_p\}$ 为聚类的集合；w_i 为第 i 个聚类的模型集合；$M=\{m_1,m_2,\cdots,m_q\}$ 为标准分类的模型集合；m_j 为第 j 个标准分类的模型集合；$purity(\Omega,M)$ 的值越高，聚类结果越准确。

为了验证使用 AP-medoids 聚类算法的聚类效果，分别将其与单独使用 K-medoids 算法和单独使用 AP 算法进行对比，并分别将 3 个聚类算法应用于 Arcat,Autodesk Seek,BIM Object 资源库和混合资源库这 4 个 BIM 资源库中，并使用 2 种聚类评价标准来评判聚类结果。

由于 AP 算法和 AP-medoids 算法聚类结果均是稳定的，而 K-medoids 算法由于初始聚类中心的选取是随机的，在本实验中，将随机选取初始聚类种子的个数为 benchmark 中对应资源库的 BIM 构件的类别数，而表 1 中的实验数据对于 K-medoids 聚类算法的结果是采用 10 次实验结果的平均值。

表1 benchmark中BIM构件的个数及其分类数

BIM 资源库	BIM 构件个数	BIM 构件类别数
Arcat 资源库	2 000	92
Autodesk Seek 资源库	4 000	63
Arcat 资源库	4 000	70
混合资源库	10 000	225

表 2 展示了针对 4 个资源库，使用类内类外标准的对比结果。

表 2 3 种聚类算法针对 4 个资源库的类内类外标准评判结果

BIM 资源库	AP 算法	K-medoids 算法	AP-medoids 算法
Arcat 资源库	0.65	0.56	0.70
Autodesk Seek 资源库	0.55	0.26	0.58
BIM Objec 资源库	0.46	0.23	0.49
混合资源库	0.54	0.32	0.58

由表 2 可知，无论哪个资源库，AP-medoids 算法的类内类外相似度的值均大于单独使用 AP 算法的值；且单独使用 AP 算法的值均大于单独使用 K-medoids 的值。亦即使用 AP-medoids 聚类算法的效果要好于单独使用 AP 算法的效果，单独使用 AP 算法的效果要好于单独使用 K-medoids 算法。

表 3 展示了针对 4 个资源库，使用 Purity 标准的对比结果。

表 3 3 种聚类算法针对 4 个资源库的 Purity 标准评判结果(%)

BIM 资源库	AP 算法	K-medoids 算法	AP-medoids 算法
Arcat 资源库	93.2	76.8	94.9
Autodesk Seek 资源库	89.3	66.1	90.6
BIM Object 资源库	90.4	68.6	92.4
混合资源库	88.9	69.6	91.6

从表 3 可知，AP-medoids 聚类算法的准确度高于单独使用 AP 算法的准确度，且单独使用 AP 算法又高于单独使用 K-medoids 算法的准确度。亦即，AP-medoids 聚类算法的效果最好。

4 结 束 语

本文提出的基于 BIM 构件属性信息量的构件聚类算法，其对传统经典的 K-medoids 聚类算法进行了 2 点改进：①利用 AP 算法的结果作为 K-medoids 的初始聚类中心，使得聚类结果变得稳定；②提出的基于属性信息量的 BIM 构件相似性度量方法，由于构件之间的相似度是经过预处理的，结果保存到中间文件，大大提高了 K-medoids 算法的运行速度和降低了算法复杂度，充分结合了 BIM 构件本身的领域信息。

为了验证本文提出的聚类算法的效果，针对 Arcat，Autodesk Seek，BIM Object 资源库和混合资源库 4 个 BIM 构件资源库，利用类内类外标准和 purity 度量 2 种聚类评价手段，将 AP-medoids 聚类算法与单独使用 AP 聚类算法和单独使用 K-medoids 聚类算法进行聚类结果的评判，实验结果证明使用 AP-medoids 聚类效果更好。

本文还将该聚类结果应用于 BIMSeek 检索系统中，实现了对关键字检索结果的分类展示以及分类器查看功能。为用户在分类器查看时更加方便，还对聚类结果进行了二次聚类标签的选取，并通过 IFC 领域信息再次对结果进行合并以及一级聚类标签的选取。

参 考 文 献

[1] GAO Y, DAI Q H, WANG M, et al. 3D model retrieval using weighted bipartite graph matching[J]. Signal Processing: Image Communication, 2011, 26(1): 39-47.

[2] GUNN T G. The mechanization of design and manufacturing[J]. Scientific American, 1982, 247(3): 114-130.

[3] ULLMAN D G. The mechanical design process[M]. New York: McGraw-Hill, 1992: 47-51.

[4] 潘翔, 张三元, 叶修梓. 三维模型语义检索研究进展[J]. 计算机学报, 2009, 32(6): 1069-1079.

[5] ALDENDERFER M S, BLASHFIELD R K. Cluster analysis[M]. Los Angeles: Sage Publications, 1984: 2-12.

[6] YU K，FROESE T M，GROBLER F. International alliance for interoperability: industry foundation classes[EB/OL]. [2013-12-23]. https://www.researchgate.net/publication/246506361_International_alliance_for_interoperability_Industry_foundation_classes.

[7] CAO J, WU Z A, WU J J, et al. Towards information-theoretic K-means clustering for image indexing[J]. Signal Processing, 2013, 93(7): 2026-2037.

[8] LIU Z, SONG Y Q, XIE C H, et al. Clustering gene expression data analysis using an improved EM algorithm based on multivariate elliptical contoured mixture models[J]. Optik, 2014, 125(21): 6388-6394.

[9] PARK H S, JUN C H. A simple and fast algorithm for K-medoids clustering[J]. Expert Systems with Applications, 2009, 36(2): 3336-3341.

[10] ZHANG T, RAMAKRISHNAN R, LIVNY M. BIRCH: an efficient data clustering method for very large databases[C]//Proceedings of the ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 1996: 103-114.

[11] ZHANG Z J, SHU H, CHONG Z H, et al. C-Cube: Elastic continuous clustering in the cloud[C]//2013 IEEE 29th International Conference on Data Engineering (ICDE). New York: IEEE Press, 2013: 577-588.

[12] LI R, LIU L. A method for large scale ontology partitioning and block matching based on ROCK clustering[J]. Applied Mechanics and Materials, 2014, 536-537: 390-393.

[13] CHAKRABORTY S, NAGWANI N K. Analysis and study of incremental K-means clustering algorithm[M]//High Performance Architecture and Grid Computing. Heidelberg: Springer, 2011: 338-341.

[14] ANKERST M, BREUNIG M M, KRIEGEL H P, et al. OPTICS: ordering points to identify the clustering structure[C]//ACM Sigmod Record. New York: ACM Press, 1999: 49-60.

[15] ANGGRAINI E L, SUCIATI N, SUADI W. Parallel computing of WaveCluster algorithm for face recognition application[C]//2013 International Conference on QiR. New York: IEEE Press, 2013: 56-59.

[16] LIU Y C, WU C, LIU M. Research of fast SOM clustering for text information[J]. Expert Systems with Applications, 2011, 38(8): 9325-9333.

[17] PENG Y, LIN J R, ZHANG J P, et al. A hybrid data mining approach on BIM-based building operation and maintenance[J]. Building and Environment, 2017, 126: 483-495.

[18] ALI M, MOHAMED Y. A method for clustering unlabeled BIM objects using entropy and TF-IDF with RDF encoding[J]. Advanced Engineering Informatics, 2017, 33: 154-163.

[19] LIU H, LIU Y S, PAUWELS P, et al. Enhanced explicit semantic analysis for product model retrieval in construction industry[J]. IEEE Transactions on Industrial Informatics, 2017, 13(6): 3361-3369.

[20] GAO G, LIU Y S, LIN P P, et al. BIMTag: concept-based automatic semantic annotation of online BIM product resources[J]. Advanced Engineering Informatics, 2017, 31: 48-61.

[21] GAO G, LIU Y S, WANG M, et al. A query expansion

method for retrieving online BIM resources based on industry foundation classes[J]. Automation in Construction, 2015, 56: 14-25.

[22] EL-MEKAWY M. EL-MEKAWY M. Integrating BIM and GIS for 3D city modelling[J]. Licentiate Thesis Geoinformatics Division Department of Urban Planning and Environment Royal Institute of Technology (KTH), 2010, 25: 55-58,.

[23] KARAN E P, IRIZARRY J. Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services[J]. Automation in Construction, 2015, 53: 1-12.

[24] MIGNARD C, GESQUIERE G, NICOLLE C. SIGA3D: a semantic bim extension to represent urban environment[C]//Proceedings of the 5th International Conference on Advances Semantic Processing. Lisbon: IARIA XPS Press, 2011: 20-25.

[25] FREY B J, DUECK D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814): 972-976.

[26] RESNIK P. Using information content to evaluate semantic similarity in a taxonomy[J]. Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995, 1(2): 448-453.

[27] TVERSKY A, GATI I. Studies of similarity[J]. Cognition and Categorization, 1978, 1(1978): 79-98.

[28] SALTON G, WONG A, YANG C S. A vector space model for automatic indexing[J]. Communications of the ACM, 1975, 18(11): 613-620.

[29] LI N, LI Q, LIU Y S, et al. BIMSeek++: retrieving BIM components using similarity measurement of attributes[J]. Computers in Industry, 2020, 116: 103186, 1-12.

[30] HAN Z, SHANG M, LIU Z, et al. SeqViews2SeqLabels: learning 3D global features via aggregating sequential views by RNN with attention[J]. IEEE Transactions on Image Processing, 2019, 28(2): 658-672.

[31] HAN Z, LU H, LIU Z, et al. 3D2SeqViews: aggregating sequential views for 3D global feature learning by CNN with hierarchical attention aggregation[J]. IEEE Transactions on Image Processing, 2019, 28(8): 3986-3999.

[32] HAN Z, LIU Z, VONG C-M, et al. Deep spatiality: unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax[J]. IEEE Transactions on Image Processing, 2018, 27(6): 3049-3063.

[33] HAN Z, LIU Z, VONG C-M, et al. BoSCC: bag of spatial context correlations for spatially enhanced 3D shape representation[J]. IEEE Transactions on Image Processing, 2017, 26(8): 3707-3720.

[34] LIU X, HAN Z, LIU Y S, et al. Point2Sequence: learning the shape representation of 3D point clouds with an attention-based sequence to sequence network[J]. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2019, 33(1): 8778-8785.

[35] HAN Z, SHANG M, LIU Y S, et al. View inter-prediction GAN: unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions[J]. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2019, 33(1): 8376-8384.