

文章编号:1001-9081(2010)02-0423-04

## 最小闭树特征集的聚类与分类方法

郭鑫<sup>1,2</sup>, 李云<sup>1</sup>, 黄云<sup>2</sup>, 周清平<sup>2</sup>

(1. 扬州大学 信息工程学院, 江苏 扬州 225009; 2. 吉首大学 信息管理与工程学院, 湖南 张家界 427000)

(jianghai079@126.com)

**摘要:**提出一种基于最小闭树特征集的聚类与分类方法,有效地解决了在实际应用中因数据量大而无法聚类与分类的问题。其基本思想为:以最小闭树特征集作为候选聚类与分类特征,采用动态阈值按相似度聚类,使得树聚类快速而精确;提出树分类规则等级概念,并应用于树分类方法中,能迅速预测未知的树结构。实验结果表明,在树节点数较多或数据量大时,新方法有效可行,且与类其他方法相比效率有显著提高。

**关键词:**数据挖掘;频繁子树;闭树模式;树聚类;树分类

**中图分类号:** TP311.13 **文献标志码:** A

## Novel tree cluster and classification approach based on least closed tree

GUO Xin<sup>1,2</sup>, LI Yun<sup>1</sup>, HUANG Yun<sup>2</sup>, ZHOU Qing-ping<sup>2</sup>

(1. School of Information Engineering, Yangzhou University, Yangzhou Jiangsu 225009, China;

2. School of Information Management and Engineering, Jishou University, Zhangjiajie Hunan 427000, China)

**Abstract:** A tree clustering and classification algorithm was proposed based on least closed tree, which effectively solved the problem that the clustering and classification can not be completed when data amount is very large in practical application. Least closed tree was regarded as the candidate cluster and classification features. The dynamic threshold was used for similarity cluster to make tree cluster operation rapid and accurate. Meanwhile the concept of tree classification rule grade was proposed and applied in tree classification algorithm, so that the unknown tree structure could be predicted promptly. Experimental results show that the method has higher speed and efficiency than that of other similar ones especially with large number of tree nodes.

**Key words:** data mining; frequent subtree; closed tree pattern; tree clustering; tree classification

## 0 引言

由于树模型能够准确地表示科学与工程领域中数据的关键特征,所以针对树挖掘<sup>[1]</sup>的研究也越来越引起人们的重视。很多领域内的数据都可抽象为树,如XML结构能抽象为有序标号树,生物学、计算机网络、WWW中的数据也可抽象为树,用于进一步的分析研究。

近年来,研究人员在树挖掘方面已经取得了不少成果,树聚类与树分类作为树挖掘的一个重要研究分支,在生物学、互联网分析等领域中有着广泛的应用。很多基于树的聚类与分类算法已经提出。在树聚类方面,文献[2-3]分别提出了XML文档树的聚类方法,将XML文档表示成有序标号树<sup>[4]</sup>,然后基于树挖掘进行聚类。这些算法存在的主要问题是当处理的数据量较小(树节点数较少)时,算法能很好运行,一旦树节点数目较多、数据库较大时,算法效率明显下降,甚至无法运行。特别是文献[3]提出的XML树相似性度量算法,节点数目已成为该算法的瓶颈。在树分类方面,文献[5]提出了基于频繁模式的分类方法,此算法存在的主要问题是挖掘频繁模式时如何设置最小支持度:设置过高,将产生少量树结构,影响分类器的性能;设置过低,当数据量较大时,算法效率较低以至于无法完成挖掘任务。文献[6]提出了基于图核函数的分类方法,算法利用支持向量机进行分类,然而这种分类

方法只在化合物结构中才具有较高的分类性能,在别的应用领域中该方法并不可行。

本文给出了一种在大型数据库中通用的树聚类与分类方法,提出基于最小闭树特征集的聚类与分类方法。算法首先调用最小闭树特征集挖掘算法挖掘特征集,然后根据本文提出的相似性度量方法进行树聚类,得到所有类集以及对应的类别,也可以对新的树结构调用树分类方法来预测树的类别。本文的主要贡献如下:

1) 提出最小闭树特征集作为候选聚类与分类特征,同时提出最小闭树特征集挖掘算法,有效地解决了在实际应用中因数据量大而无法进行聚类与分类的问题。

2) 提出树相似性度量方法,解决了文献[6]中的瓶颈问题,并提出了动态阈值选择的树聚类方法。

3) 提出树分类规则等级和树分类方法,能准确有效地预测新树类别。

4) 实验结果表明,在对生物学中的Ribonuclease P<sup>[7]</sup>等大型数据聚类与分类时,本文提供的方法明显优于其他算法。

## 1 基本概念

本文将问题集中在有序标号树。在详细介绍聚类与分类方法之前,首先给出有关基本概念和问题的定义(相关概念与思想引用自作者的硕士论文<sup>[8]</sup>)。

收稿日期:2009-08-07。

**作者简介:**郭鑫(1984-),男,湖南常德人,助教,硕士,主要研究方向:数据挖掘、并行计算; 李云(1965-),男,安徽合肥人,教授,主要研究方向:概念格; 黄云(1976-),男,湖南张家界人,讲师,主要研究方向:数据挖掘; 周清平(1966-),男,湖南张家界人,教授,主要研究方向:量子计算。

**定义1** 有序标号树<sup>[1,9]</sup>。一棵有序标号树  $T = (V, E)$  是一个具有标号和根节点的有向无环图,其中  $V$  表示树中的节点集合,  $E = \{(x, y) \mid x, y \in V\}$  表示树中的边集合,用  $L = \{l_0, l_1, \dots, l_n\}$  表示一个标号集,节点  $V$  与  $L$  存在一个对应关系  $l: V \rightarrow L, l_i$  称为节点标号。有序是指对于树中每个节点,其所有子节点按照从左到右的顺序形成兄弟关系。

**定义2** 频繁子树。给定有序树数据库  $TDB = \{T_i \mid i = 1, 2, \dots, n\}$  以及最小相对支持度阈值  $min\_sup$ , 如果子树  $T$  满足  $rel\_sup(T, TDB) \geq min\_sup$ , 则称子树  $T$  为  $TDB$  中的频繁树模式或频繁子树,  $TDB$  中所有的频繁树模式集合记作:

$$F(TDB) = \{T \mid rel\_sup(T, TDB) \geq min\_sup\}$$

**定义3** 闭树模式。给定有序树数据库  $TDB = \{T_i \mid i = 1, 2, \dots, n\}$ , 设  $T$  是一个树模式(子树), 如果在有序树数据库  $TDB$  中不存在与  $T$  支持度相同的真超树, 则称  $T$  为  $TDB$  中的闭树模式,  $TDB$  中所有的频繁闭树模式(频繁子树)集合记作:

$$CF(TDB) = \{T \mid T \in F(TDB) \exists Q \in F(TDB),$$

$$T \subset Q, rel\_sup(T, TDB) = rel\_sup(Q, TDB)\}$$

树的聚类问题可描述为:给定一个树数据库  $TDB = \{T_i \mid i = 1, 2, \dots, n\}$ , 采用一定的方法得到  $r$  个互不相交的聚类  $\{\beta_1, \beta_2, \dots, \beta_r\}$ , 可从每一个聚类当中选取最具代表性树例, 将此树例称为树类别, 从而可以得到一个树类别数据集, 记为  $\{t_1^*, t_2^*, \dots, t_r^*\}$ 。

**定义4** 最小闭树。设  $T = \{t_1, t_2, \dots, t_r\}$  为某闭树集,  $t \in T$ , 如果  $\forall t' \in T, t'$  都不是  $t$  的子树结构, 则称  $t$  为最小闭树, 选取所有最小闭树中节点数最小的闭树  $t^*$  作为树类别。

树的分类问题可描述为:输入  $r$  个聚类  $\{(\beta_i, t_i^*) \mid i = 1, 2, \dots, r\}$  和某未知类别的树  $T$ , 其中  $\beta_i$  是树集合,  $t_i^*$  是树类别。输出其中某个聚类  $(\beta_i, t_i^*)$ , 且树  $T$  属于此类别。

## 2 最小闭树的聚类与分类

### 2.1 最小闭树特征集挖掘算法

在本文中只关注最小闭树特征集的挖掘,与挖掘所有频繁子树和频繁闭树相比,挖掘最小闭树特征集有如下几个优点:首先,频繁闭树的数量要小于频繁子树的数量。例如在生物学中的 Ribonuclease P 数据库应用中,数据库文件约 2 MB, 当最小支持度为 1% 时,将产生至少 8 GB 大小的频繁子树库文件,频繁闭树的库文件也接近 200 MB。而在这些海量数据中,大量的数据存在高度相似,有些子树之间仅仅只有某条边不同,或者某个节点标识不同。其次,支持度越小,所得到的聚类与分类信息将越准确,如果只使用闭树模式特征集来分析,就可以使用更小的支持度阈值,产生更多对聚类与分类信息相关的子树信息。更为重要的是,挖掘闭树模式不会丢失任何对聚类与分类有用的信息,因为可以从闭树集合中恢复出频繁子树。

ITMSV<sup>[10]</sup> 算法是一个有序树挖掘算法,该算法采用最右边扩展生成候选子树,并提出两个连接策略来生成子树。本文基于 ITMSV 算法提出一个最小闭树特征集挖掘算法,算法根据闭树的性质,定义一个剪枝程序,删除所有的非闭树结构,程序对所有的子树检查是否存在与其支持度相同的子树,因为所有的频繁子树信息已保存在哈希表中,所以我们只需根据节点信息查找哈希表值即可。算法又定义一个随机选择器来随机选择频繁闭树,并定义两个特征精度  $\beta = [m, n]$  (其中  $m, n$  为非负小数)和  $\lambda > 1$  来控制频繁树生成的数量,例如取  $\beta = [0.1, 0.2]$  和  $\lambda = 10000$  时,当随机选择器生成

的数在  $\beta$  范围之内时,则保留此子树,否则删除,并且判断子树总数是否超过  $\lambda$  值,若已生成 10000 棵特征树,则算法停止生成。最后算法根据得到的所有的闭树特征集,将所有的小闭树提取出来。

算法首先扫描数据库得到频繁一子树,将所有的频繁一子树构建成一个哈希表,并将此哈希表放到一个线性表结构中,对哈希表中的值按照候选生成策略进行两两合并得到频繁二子树,此时算法调用剪枝程序,对所有的频繁二子树检查是否存在与其支持度相同的子树,如果存在,则删除此二子树,否则调用随机选择器,根据系统当前时间随机选择一个非负小数,若生成数在  $\beta$  范围之内,则保留此子树,否则删除子树,若保留则判断生成树总数是否超过  $\lambda$  值,如果超过,则算法停止。最后构建一个新的包含频繁二子树的哈希表,同时放入表结构中,其中哈希表的键为树节点的值,哈希表的值存储有树前缀和子树向量等信息。依此类推得到余下的频繁闭树结构,最后统一生成所有的最小闭树集。

**算法1** Procedure LeastClosedTreeMine

输入: the database of tree:  $TDB$ , support,  $\alpha, \beta, \gamma$ ;

输出: all least closed trees.

```

1) LeastClosedTreeMine (  $TDB, min\_usp$  ):
2)  $F_1 = \{\text{frequent 1-subtrees}\}$ ; Hash  $\leftarrow F_1$ ;
3) Line[1]  $\leftarrow$  Hash and Node = 1 and TotalNum = 0;
4) While( Line[Node]! = null) do
5)   Node = Node + 1; new_hash = null;
6)   For all [  $T$  ] in Hash do
7)     For each element  $(x, i) \in [T]$  do
8)       For each element  $(y, j) \in [T]$  do
9)          $R = \{(x, i) \oplus (y, j)\}$ ;
10)        If Node <  $\alpha$  then
11)          If all subtrees of  $R$  is frequent then
12)            Delete  $R$ ; continue;
13)           $L(R) = \{L(X, i) \oplus L(Y, j)\}$ ;
14)          If support ( $R$ ) > min_usp AND Check ( $R$ ) AND
              Rand()  $\in \beta$  then
15)            TotalNum ++; New_hash  $\leftarrow R$ ;
16)            If TotalNum >  $\gamma$  then return;
17)          End if
18)        End for all;
19)      If new_hash! = null then
20)        Line[Node]  $\leftarrow$  new_hash;
21)      End while
22) End LeastClosedTreeMine

```

### 2.2 树相似性度量

相似性度量方法包括特征比较和转换代价计算等,也可以将多种方法结合起来计算。两个树结构  $T_a$  和  $T_b$  的相似度可用表示  $T_a$  和  $T_b$  相同部分的信息量在  $T_a$  和  $T_b$  的信息问题中所点比例求得,给定两个树结构  $T_a$  和  $T_b$ , 那么相似性的计算公式可定义为:

$$sim(T_a, T_b) = \frac{|t(T_a) \cap t(T_b)|}{|t(T_a) \cup t(T_b)|} \quad (1)$$

本文采用特征比较的方法来计算相似性,基于文献[3]的 XML 文档的相似性计算方法,考虑到树的结构化特性,树的相似性度量不仅仅要计算节点语义之间的相似性,更应该考虑到结构上的相似性。因此本文同时计算节点语义的相似性和树结构的相似性,然后通过对两者加权的方法来计算总的相似性。

在实际的应用中,发现树的结构特性往往要比语义特性

重要,例如生物学者往往会更注重某个特殊结构的分子结构,优秀的网站结构能够让用户轻松快捷地浏览网页,因此在树相似性比较中,设置两个权值将会存在一个较大的差值。

**定义5** 扩展向量<sup>[3]</sup>。设  $na$  为树节点标识,  $Ex(na) = (na, na^1, na^2, \dots, na^m)$  称为名称  $na$  的扩展向量,其中  $na^i = (1, 2, \dots, m)$  为  $na$  的同义词、合成词或缩写形式。

**定义6** 设  $t_1, t_2$  为树节点标识,则  $t_1$  和  $t_2$  的相似度<sup>[3]</sup>评分定义为:

6分:  $t_1$  和  $t_2$  完全匹配;

5分:  $t_1$  与  $Ex(t_2)$  中除  $t_2$  外的某元素完全匹配或  $t_2$  与  $Ex(t_1)$  中除  $t_1$  外的某元素完全匹配;

4分:  $Ex(t_1)$  和  $Ex(t_2)$  中除  $t_1$  和  $t_2$  外的某两个元素完全匹配;

3分:  $t_1$  和  $t_2$  部分匹配;

2分:  $t_1$  与  $Ex(t_2)$  中除  $t_2$  外的某元素部分匹配或  $t_2$  与  $Ex(t_1)$  中除  $t_1$  外的某元素部分匹配;

1分:  $Ex(t_1)$  和  $Ex(t_2)$  中除  $t_1$  和  $t_2$  外的某两个元素部分匹配;

0分: 不存在任何匹配。

对任意两个树结构  $t_1, t_2$ , 计算  $t_1$  和  $t_2$  的相似度公式定义为:

$$Sim(t_1, t_2) = \lambda_1 SemSim(t_1, t_2) + \lambda_2 StruSim(t_1, t_2) \quad (2)$$

其中:  $SemSim(t_1, t_2)$  为语义相似度;  $StruSim(t_1, t_2)$  为结构相似度;  $\lambda_1, \lambda_2$  分别为语义相似度与结构相似度权值。在实际计算时,我们设定  $\lambda_1$  为 0.2,  $\lambda_2$  为 0.8。

在计算语义相似度时,首先根据树节点的数目计算如下两个向量的其中一个的值:

$$\begin{aligned} t_1 &= (\langle Ex(l_1^1), score_1^1 \rangle, \langle Ex(l_1^2), score_2^1 \rangle, \dots, \\ &\quad \langle Ex(l_1^m), score_m^1 \rangle) \\ t_2 &= (\langle Ex(l_2^1), score_1^2 \rangle, \langle Ex(l_2^2), score_2^2 \rangle, \dots, \\ &\quad \langle Ex(l_2^n), score_n^2 \rangle) \end{aligned}$$

根据树结构重要性要远大于语义重要性的特性,本文只需计算两个向量中其中一个的值来作为语义相似度的值:若  $t_1$  的节点个数要小于  $t_2$  的节点个数,则计算  $t_1$  的向量值;否则计算  $t_2$  的向量值。其中  $Ex(l_i^j)$  是树  $j$  的第  $i$  个节点标识的扩展向量,  $score_i^j$  是节点  $l_i^j$  的相似度评分值,取  $l_i^j$  与另一树中最相似的节点评分,对所有节点只计算一次相似性评分,然后按如下公式计算:

$$SemSim(t_1, t_2) = \sum_{i=1}^m score_i^j / (6k) \quad (3)$$

在式(3)中,  $k$  为节点数较小的树的节点数目,  $score_i^j$  之和为对应的树的相似性评分。当树节点数较多时,算法能大大节省评分函数计算的时间,提高算法效率。

在计算结构相似度时,首先对树节点进行深度优先编号,并且保持相似的节点编号相同,这样,节点树的每一条路径可用一个数字序列表示,通过挖掘同时出现在两子树中的频繁序列找出相似路径集,然后按如下公式进行计算:

$$StruSim(t_1, t_2) = \frac{1}{N+1} \left[ \left( \sum_{i=1}^N \frac{1}{L(R_i)} \times V(R_i) \right) + MR \right] \quad (4)$$

在式(4)中,  $N$  为  $t_1$  和  $t_2$  中子树数较多的一级子树的数量,  $R_i$  为  $t_1$  和  $t_2$  中子树数较多的第  $i$  个一级子树的根节点,  $L(x)$  为树的层数函数,其计算公式为:

$$V(R_i) = \begin{cases} F(R_i), & R_i \text{ 为叶子节点} \\ F(R_i) + \frac{1}{N(C_i) \sum_{e \in C_i} \frac{1}{L(e)} V(e)}, & R_i \text{ 为非叶子节点} \end{cases}$$

$$F(R_i) = \begin{cases} 1, & \text{如果 } R_i \text{ 在相似路径中} \\ 0, & \text{如果 } R_i \text{ 不在相似路径中} \end{cases}$$

其中:  $C_i$  为  $R_i$  的儿子节点集合;  $N(C_i)$  为集合  $C_i$  的基数。

$$MR = \begin{cases} 1, & t_1, t_2 \text{ 的根节点相似} \\ 0, & t_1, t_2 \text{ 的根节点不相似} \end{cases}$$

### 2.3 树聚类算法

算法首先调用最小闭树特征集挖掘算法得到一个特征集,然后通过计算树的相似度来进行聚类分析,将所有相似性较近的树聚成一类。为了便于更加准确地聚类,本文将动态地选择阈值  $\varepsilon$ , 从特征集中随机选取一个子树  $t$ , 计算  $t$  与特征集中其他子树的相似度,并将所有相似度值按递减排序形成一条非增曲线,通过计算二阶导数来判断曲线的拐点,将离坐标原点最近的拐点所在相似度作为阈值  $\varepsilon$ , 将所有超过此阈值的子树构成一个候选集,如果候选集的数量达到用户设定的密度值  $d > 1$ , 则将这些子树构成一类集。然后在特征集中删除这些子树,余下的子树采用类似的方法进行计算得到其他类集,如果未达到聚类的密度,则重新随机选取一个子树按上述方法进行计算。

对得到的所有的类集计算树类别,将所有的树类别组成集合  $\{t_1^*, t_2^*, \dots, t_r^*\}$ , 其中  $t_i^*$  表示第  $i$  类中的类别,计算方法为:对所有类集分别选取节点数目最小的闭树  $t^*$ , 则  $t^*$  为此类集的类别树。

#### 算法2 Procedure TreeCluster

输入: least closed trees:  $T$ , density:  $d$ ;

输出: all class  $C$  and  $Tc$ 。

```

1) TreeCluster( $T, d$ ):
2)   While( $T! = \text{null}$ ) do
3)      $t = \text{rand}()$ ;
4)     For all  $temp$  in  $T$  do
5)        $Sim(t, temp)$ ;
6)     End for all;
7)      $threshold = \text{Sort}()$ ;
8)      $S = \{all \mid Sim(all, t) > threshold\}$ ;
9)     If  $|S| \geq threshold$  then
10)       $C \leftarrow S$ ;  $T = T - S$ ;
11)     End if
12)   End while
13)   For all  $c$  in  $C$  do
14)      $Tc \leftarrow \text{gen}(c)$ ;
15)   End TreeCluster

```

### 2.4 树分类算法

当算法聚类结束并得到  $r$  个类集  $\{\beta_1, \beta_2, \dots, \beta_r\}$  之后,就可以对一个新的树结构  $t$  进行分类。显然,对某类集  $\beta_i$  中的每个闭树  $p$  都将对应一个分类规则:  $p \rightarrow t_i^*$ ,  $t_i^*$  为类集  $\beta_i$  对应的树类别。尽管可以直接使用这些分类规则对一个新实例进行分类,然而这样做却存在一些问题。为了不丢失对分类有重要预测作用的闭树结构,支持度阈值通常取很小的数值,虽然算法只得到最小闭树特征集,但对某些类集来说依然数目很大,如果对所有闭树都进行树匹配操作,将涉及大量的计算,在实际应用中,树节点的数目往往很大,这样将会严重限制分类方法的实际应用。因此有必要从所有的分类规则中选择少量对分类起决定作用的分类规则。在介绍分类方法之前,先定义一个树分类规则等级的概念。

**定义7** 树分类规则等级。对任意两个分类规则  $t_1$  和  $t_2$ , 如果下面的条件之一成立,我们说  $t_1$  的等级高于  $t_2$ , 用

$t_1 > t_2$  表示。

- 1)  $t_1$  的支持度高于  $t_2$ 。
- 2)  $t_1$  和  $t_2$  支持度相同,  $t_1$  的节点数目大于  $t_2$ 。
- 3)  $t_1$  和  $t_2$  支持度和节点数目都相同,  $t_1$  较  $t_2$  先输出。

根据分类规则等级,所有类集中的分类规则可以按从大到小形成线性顺序,本文从中选取  $n$  个等级较高的规则,为了尽可能准确地分类,对  $r$  个类集分别计算与  $t$  的相似性评分均值,其计算公式为:

$$Equal(t, \beta_i) = \sum_{k=1}^n Sim(t, t_k) / n \quad (5)$$

其中:  $\beta_i$  表示第  $i$  个类集;  $t_k \in \beta_i$ ;  $n$  表示类集  $\beta_i$  中等级较高的规则数目,若所选  $n$  值大于所有规则数目,则  $n$  取所有规则。根据计算出的所有评分均值来预测  $t$  的类别,取其中最大值所有在类集作为新树类别。

算法3 Procedure TreeClassification

输入: class:  $C$ , trees:  $t$ , number:  $n$ ;

输出: category。

- 1) TreeClassification( $C, t, n$ );
- 2) Score[ ];
- 3) Sort( $C$ );
- 4) For all  $c$  in  $C$  do
- 5)     Score ← Equal( $t, c$ );
- 6) End for all;
- 7) max = Max(Score);
- 8) End TreeClassification

### 3 实验与分析

本文采用 C++ 实现了提出的所有算法,对这些算法在不同的参数下进行了大量实验。实验环境为 Pentium(R) 4 CPU,操作系统为 Red Hat Linux6.0。算法采用线程来计算用时。

本文用人工数据库来对算法进行测试,为了真实证实算法有效性,采用通用的人工数据库生成器,生成器采用文献[6]的方法,通过8个参数来调整产生数据的分布。这8个参数分别为标号集大小  $S$ 、节点生成子节点概率  $p$ 、基本树的个数  $L$ 、基本树的高度  $I$ 、基本树中每个节点的扇出  $C$ 、 $TDB$  的大小  $N$ 、 $TDB$  中每棵树的最大高度  $H$ 、 $TDB$  中每个节点的扇出  $F$ 、基本树和  $TDB$  中树的高度都遵循期望为  $I(H)$ 、标准差为1的高斯分布。生成数据的默认参数为  $S100 P0.5 L10 I4 C3 N100000 H8 F6$ ,默认的支持度阈值为1%。

本文将文献[2]中的 ATFC 算法和文献[3]中的 XProj 算法与本文的 TreeCluster 算法进行对比,在不同支持度阈值时实验结果如图1所示。

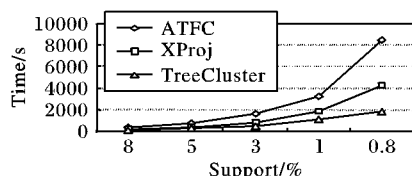


图1 运行时间与支持度的关系(人工集)

将得到的所有类集使用分类算法 TreeClassification 与文献[6]中的 XRules 算法进行对比,对包含10棵未知类别的树,按(扇出、树高)复杂程度递增进行实验,其实验结果如图2所示。

本文对生物学中的 Ribonuclease P 数据进行聚类与分类,因为数据量太大,我们较人工数据库提高支持度阈值,同样聚类算法在不同支持度情况下实验结果如图3所示。分类算法在树复杂程度不同情况下实验结果如图4所示。

本文将提出的聚类与分类方法来进行网络日志分析。从 <https://www.cs.washington.edu/research/adaptive/> (Adaptive Web Sites) 下载了网络日志,选择1999年9月20日至10月4日的网络日志,从海量数据中统计出访问 cs.washington.edu 的日志约 500 000 条,并转换为树结构。在实验中得到了如表1所示的类关系集合。

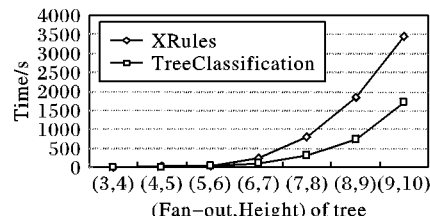


图2 运行时间与树复杂程度的关系(人工集)

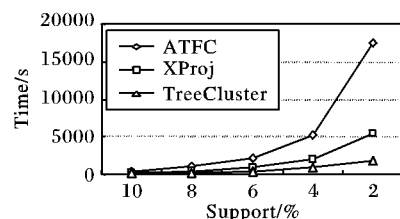


图3 运行时间与支持度的关系(真实集)

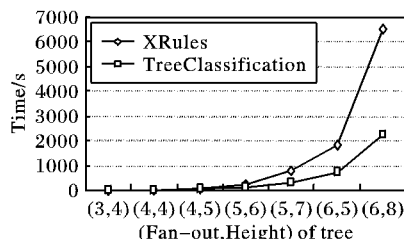


图4 运行时间与树复杂程度的关系(真实集)

表1 产生的类关系集合

Label1	Label2
research	projects
education	courses
people	faculty
...	...

### 4 结语

本文介绍了一种在大型数据库中的树聚类与分类方法,为许多应用领域给出了一个通用的解决方法。本文提出以最小闭树特征集作为候选聚类与分类特征,使得可以采用更小的支持度来挖掘更多的树集合,产生更多对聚类与分类有重要作用的候选特征。提出了最小闭树特征集挖掘算法,有效地解决了在实际应用中因数据量大而无法进行聚类与分类的问题。采用动态选择阈值并按相似度聚类方法,可以将包含不同相似度的树结构区分开来,从而能快速而精确进行树聚类。提出树分类规则等级概念,并将此方法应用于树分类方法中,能较准确预测求知的树结构。大量实验结果表明:在树节点数较多或数据量大时,本文介绍的方法有效可行;在 Ribonuclease P 数据库应用中,本文方法的效率明显好于其他方法。

本文的树聚类与分类算法是基于有序标号树,可以考虑将更多的特征形式,如无序树、无标号树等,引入到聚类与分类分析中,也可考虑扩展树挖掘,将已有的方法引入到图挖掘<sup>[11-12]</sup>、图的聚类与分类中,这些都是我们以后深入探讨的研究方向。

### 参考文献:

- [1] ZAKI M J. Efficiently mining frequent trees in a forest: Algorithms and applications [J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(8): 1021-1035. (下转第448页)

8.5%,本文方法的识别率略高于HMM方法,误识率稍低于HMM方法,得到了比较满意的结果,而且本文方法数据预处理方法简单,只对原始数据进行了归一化,没有任何去噪、平移、旋转等过程,最大限度地保留了原始签名数据的特性,这增加了算法的鲁棒性、抗噪性;其次对全局信息进行特征提取,对整个签名识别,无需对连笔现象普遍的签名分割,避免了复杂的分割;再次利用小波包特征提取,是完全且可逆过程,而一般方法会丢失细节且不可逆;最后用混合高斯模型拟合了聚类后的小波包特征提取值,并用状态转移概率描述了模式间的依赖性和相关性,使模型更好地描述了签名数据的特征信息。

### 5.3 算法时间复杂度分析

假设 $n$ 表示签名的种类, $m$ 表示每类中签名的个数, $S_f$ 表示小波包分解的频段数, $T_k^i$ 表示第 $i$ 样本在第 $k$ 个频段上子集数目,通过分析可得:

$$T(n, m) = C_1 nm + C_2 S_f nm + C_3 S_f n + C_4 S_f T_k^i n + C_5 n^2 m S_f T_k^i + C_6 nm S_f T_{k+1}^i T_k^i$$

式中: $C_1, C_2, \dots, C_6$ 为常数项;对于归一化后的签名图像 $S_f$ 为常数; $T_k^i$ 较小,大多在5以下,可看作常数。 $T(n, m)$ 可简化成:

$$T(n, m) = Q_1 nm + Q_2 n + Q_3 n^2 m$$

式中 $Q_1, Q_2, Q_3$ 为常数。所以时间复杂为 $T(n, m) = O(n^2 m)$ 。

## 6 结语

本文提出的手写体签名识别方法,数据预处理简单,小波包特征提取完全且可逆,转移概率表征了拟合了小波包分解值的混合高斯模型间的相似和变化。通过实验可知,算法的识别率较高、误识率低,具有较好的抗噪性、鲁棒性,它为含噪脱机手写体签名识别提供了一种可行的技术解决方案。

今后可从以下几个方面做重点研究:1)对小波包特征提取的特性进行分析,用更少的特征点来描述图像特征;2)对转移概率模型进一步改进,提高算法识别效果;3)从算法实现的角度进一步降低算法时间复杂性,提高运行效率。

### 参考文献:

- [1] 吴谨,邱亚.基于空间分布特征的手写体数字识别[J].武汉科技大学报,2004,27(2):176-178.
- [2] 孟明,吴仲城,余永,等.基于笔段特征和HMM的在线签名认证方法研究[J].模式识别与人工智能,2007,20(1):95-100.
- [3] PESSOA L F C, MARAGOS P. Neural networks with hybrid morphological/rank/linear nodes: A unifying framework with applications to handwritten character recognition[J]. Pattern Recognition, 2000, 33(6): 945-960.
- [4] 李媛,袁余良,沈峰,等.一个基于神经网络的动态手写签名验证模型[J].计算机科学,2005,32(5):181-184.
- [5] LV HAI-RONG, WANG WEN-YUAN, WANG CHONG, et al. Off-line Chinese signature verification based on support vector machines[J]. Pattern Recognition Letters, 2005, 26(15): 2390-2399.
- [6] 陈刚,李炳程,曹闻,等.一种有效的基于证据理论的离线手写签名识别方法[J].计算机工程与设计,2006,27(17):3256-3260.
- [7] KOERICH A L, SABOURIN R, SUEN C Y. Recognition and verification of unconstrained handwritten words[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(10): 1509-1522.
- [8] MOHAMED M A, GADER P. Generalized hidden Markov models - Part II: Application to handwritten word recognition[J]. IEEE Transactions on Fuzzy Systems, 2000, 8(1): 82-95.
- [9] 李辉,李峰,黄道昌.基于MHMM的脱机手写体字符识别[J].长沙理工大学学报,2007,4(2):63-67.
- [10] 刘刚,张洪刚,郭军.用于脱机手写体数字识别的隐马尔可夫模型[J].计算机研究与发展,2003,40(8):1252-1257.
- [11] EL-YACOUBI A, GILLOUX M, SABOURIN R, et al. An HMM-based approach for off-line unconstrained handwritten word modeling and recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999, 21(8): 752-760.
- [12] 汪雪林,赵书斌,彭思龙.基于小波域隐马尔可夫模型的图像复原[J].计算机学报,2005,28(6):1007-1012.
- [13] 汪雪林,赵书斌,彭思龙.基于小波域局部高斯模型的图像复原[J].软件学报,2004,15(3):443-450.
- [14] 汪雪林,文伟,彭思龙.基于小波域局部高斯模型的图像超分辨率[J].中国图象图形学报,2004,9(8):941-946.
- [15] 侯建华,熊承义,田金文,等.基于MATLAB的图像小波子带广义高斯模型的研究[J].计算机应用与软件,2006,23(1):131-132.
- [16] 马海豹,刘漫丹,张岑.基于小波包分析的在线手写签名认证方法[J].计算机工程与应用,2007,43(12):235-238.
- [1] 吴谨,邱亚.基于空间分布特征的手写体数字识别[J].武汉科技大学报,2004,27(2):176-178.
- [2] AGGARWAL C C, TA N, WANG J, et al. XProj: A framework for projected structural clustering of XML documents [C]// SIGKDD'07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2007: 46-55.
- [3] 吴扬扬,雷庆,陈锻生,等.一种从XML数据中发现关系信息的方法[J].软件学报,2008,19(6):1422-1427.
- [4] 赵传申,孙志挥,张净.基于投影分支的快速频繁子树挖掘算法[J].计算机研究与发展,2006,43(3):456-462.
- [5] DESHPANDE M, KURAMOCHI M, WALE N, et al. Frequent substructure-based approaches for classifying chemical compounds [J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(8): 1036-1050.
- [6] HORVATH T, GARTNER T, WROBEL S. Cyclic pattern kernels for predictive graph mining [C]// KDD 2004: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2004: 158-167.
- [7] BROWN J W. The ribonuclease P database [J]. Nucleic Acids Research, 1998, 26(1): 351-352.
- [8] 郭鑫.频繁子树挖掘及其相关技术的研究[D].扬州:扬州大学,2009.
- [9] 朱永泰,王晨,洪铭胜,等.ESPM——频繁子树挖掘算法[J].计算机研究与发展,2004,41(10):1720-1726.
- [10] LI YUN, GUO XIN, YUAN YUN-HAO. A fast algorithm of mining induced subtrees [C]// ICIA 2008: Proceedings of International Conference on Information and Automation. Washington, DC: IEEE Press, 2008: 195-199.
- [11] CHAOJI V, HASAN M A, SALEM S, et al. ORIGAMI: A novel and effective approach for mining representative orthogonal graph patterns [J]. Statistical Analysis and Data Mining, 2008, 1(2): 67-84.
- [12] CHAOJI V, HASAN M A, SALEM S, et al. An integrated, generic approach to pattern mining: Data mining template library [J]. Data Mining and Knowledge Discovery, 2008, 17(3): 457-495.

(上接第426页)