# 基于随机森林的实体识别方法

杨 萌 聂铁铮 申德荣 寇 月 于 戈

（东北大学计算机科学与工程学院 沈阳 110819）

**摘 要** 实体识别是将一个或多个数据源中描述同一现实世界实体的数据对象分到同一组的过程，它在数据清洗、数据集成、数据挖掘中起着至关重要的作用。然而，实体的特征具有随时间演化的特性，这使得实体识别面临巨大的挑战。传统的实体识别方法解决了特征随着时间规律性的改变问题，但没有考虑到数据的不规律变化。该文提出了基于分类的方法解决特征不规律演化的实体识别问题。该方法首先利用机器学习中改进的随机森林的方法计算记录的相似性，接着提出了一个新型的两阶段聚类算法完成记录聚类过程，最后通过在真实数据集上的对比试验证明了该算法的有效性。通过在真实数据集上的实验，证明了该方法能够有效提高演化实体的识别准确性。

**关键词** 实体识别；聚类；随机森林；记录相似度

**中图分类号** TP 315 **文献标志码** A

# An Entity Resolution Approach Based on Random Forest

YANG Meng NIE Tiezheng SHEN Derong KOU Yue YU Ge

( *School of Computer Science and Engineering*, *Northeastern University*, *Shenyang* 110819, *China* )

**Abstract** Entity Resolution assigns data objects corresponding to the same real world entity described in one or more data sources into the same group, which plays an important role in data cleaning, data integration, and data mining. However, the features of the entity may evolve over time irregularly, which makes the entity resolution significantly challenging. Traditional approaches can only tackle the issue that the feature of an entity changes regularly with time but can not deal with the case that the feature changes irregularly over time. An approach based on classification was proposed to solve this problem. Firstly, the random forest, a machine learning algorithm, was used to calculate the similarity of records. Consequently, new two-stage clustering algorithm was employed to perform the record clustering. Finally, the evaluation on real data sets shows that the approach can effectively improve the resolution accuracy of the evolutionary entity.

**Keywords** entity resolution; clustering; random forest; record similarity

**收稿日期**：2017-11-14 **修回日期**：2017-12-24

**基金项目**：国家自然科学基金项目(61672142)；中央高校基本科研业务费项目(N150408001-3、N150404013)

**作者简介**：杨萌，硕士，研究方向为实体识别；聂铁铮(通讯作者)，博士，副教授，研究方向为数据集成，E-mail：nietiezheng@cse.neu.edu.cn；申德荣，博士，教授，研究方向为数据库；寇月，博士，副教授，研究方向为数据库；于戈，教授，研究方向为数据库。

## 1 引 言

随着信息领域各项技术的飞速发展，数据呈爆炸式增长，以数据为中心的系统得到了广泛的应用。虽然各类应用系统中存储了大量数据，但这些信息并非总是正确无误的，即可能存在各种问题。一个典型的问题就是不同的数据提供方对同一个事物及实体可能会有不同的描述(包括数据格式、表示方法等)，为此需要实体识别技术进行数据清洗。实体识别也称作实体解析，是从“引用集合”中解析并映射到现实世界中“实体”的过程。记录链接则是一种面向结构化数据的实体识别技术，目的是从数据集中识别和聚类表示同一实体的记录。

现有研究工作中，有基于记录的组链接[1]和基于记录的实体链接。其中，组链接是指把表示同一类的实体放到相同的聚簇中；实体链接是把表示同一个实体的记录方法放到相同的聚簇中。本文主要研究实体链接。基于所处理的数据对象，实体识别技术可以分为两类：面向静态数据的实体识别和面向演化数据的实体识别。

面向静态数据的实体识别方法：从数据集中识别和聚类表示同一实体的记录，对相似度达到一定阈值的记录做聚类操作，从而获得表示同一个实体的记录簇，而不同簇中的记录认为表示不同的实体。实体间相似性一般根据领域知识设定匹配规则度量标准[2]，可通过编辑距离和欧氏距离计算[3]，也可以用机器学习训练分类器的方法实现[4]。基于相似性对实体进行聚类的方法有邻接性聚类[2]、相关性聚类[5,6]、密度聚类[7-9]。

面向静态数据的实体识别方法在很多情况下并不适用。在现实应用中，实体记录的某些属性值通常会随时间或解释的变化而发生演化，而面向静态数据的实体识别方法无法根据属性值的演化调整相似性的计算结果。

面向演化数据的实体识别，是考虑数据随时间的变化而变化的特性即考虑时间特征，体现了数据的动态性和演化性。Li 等[10]在计算记录的相似性时考虑了记录的时间特征：考虑时间的流逝对记录改变的影响，基于延迟提出了 early binding、late binding、adjusted binding 三个聚类算法。之后 Hu 等[11]提出了基于时序特征的记录链接的改进方法；Chiang 等[12]提出了两阶段聚类的方法；Chiang 等[13]提出了 mutation 模型，用来检测一个给定属性的值经过一段时间之后该值重复出现的概率；Li 等[14]提出了 source-aware temporal matching 算法，整合不同的数据源，丰富实体的信息。除此以外，还有基于增量的实体识别方法，从匹配规则的演化[11]及数据的演化[15-17]两方面为依据探讨记录链接的增量问题。

对于演化数据，实体记录的某些属性值通常会发生变化。其中有些实体属性会随着时间的变化而发生规律性演化，但也有实体属性的演化不具有规律性，因此很难抽取出其演化的规律。对于不规律演化的数据，基于演化的实体识别方法聚类的结果准确度并不高。这是因为对于这种不规律的变化也使用规律变化的准则结果会产生偏差。为此，本文的工作将解决不规律演化实体的识别问题。

本文提出一个基于随机森林的两阶段聚类实体识别模型：利用已有的数据集，训练出随机森林。其中，随机森林是由很多棵决策树组成的，每棵树的输入是任意两条记录，输出是两个记录的相似度结果。在最后的聚类结果，利用前面记录的相似度结果，进行两阶段聚类，其中保证簇内的记录尽可能完整，同时尽可能多的簇被发现，提高聚类结果的准确性。主要贡献如下：

(1)提出基于随机森林的记录相似度计算模型。该相似度模型充分考虑实体记录变化的不规律性，从而动态地、准确地衡量记录的相似性。

(2)提出一个新型的两阶段聚类算法。簇的聚类过程分为两个阶段：第一阶段进行核聚类，

把尽可能多的已经确定的记录放在相同的簇中；第二阶段进行端点聚类，能够将剩余的记录和已知的簇合并或者合并已有的簇，保证簇内记录尽可能完整，尽可能多的簇被发现。

(3)在真实的数据集上对提出的算法进行充分的实验评价，验证算法的有效性。与已有的聚类算法对比，该算法能够有效提高演化实体的识别准确性。

本文组织结构如下：第 2 节介绍准备工作，包括实体识别模型和问题描述；第 3 节对连续值的决策树和随机森林进行了定义；第 4 节介绍了基本的随机森林计算相似度算法框架，并提出防止过拟合的优化算法；第 5 节通过实验与分析将本文工作与已有工作进行对比，证明其有效性；第 6 节总结全文。

## 2　准备工作

### 2.1　实体识别模型

基于聚类方法的实体识别模型包括相似度计算模块和聚类决定模块，具体如图 1 所示。整个模型输入待判断数据集，输出识别结果。

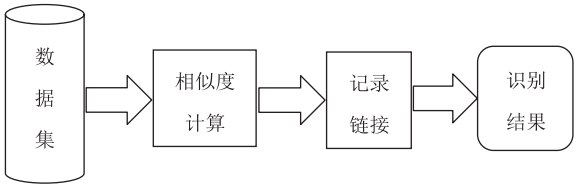



**图 1　实体识别模型**

**Fig. 1　The similarity algorithm based random forest**

2.1.1 相似度计算模块

该模块调用匹配函数得到候选记录对的相似度[18]，得到的相似性结果介于[0，1]。相似度的值越大，表示两个数据对象越有可能表示同一个实体。其中，最大值 1 代表两个记录表示同一个实体，最小值 0 代表两条记录表示不同实体。每个记录包含多个属性，不同属性可能是不同类型的数据。在确定记录对相似度之前，针对每个属性调用特定的相似度函数来计算其相似度，确定记录对的对应属性的相似性。在此基础上需要设计恰当的组合函数来将这些相似度合理地融合成一个综合相似度。组合函数可以是线性函数、非线性函数或者其他类型的函数[3-5]，如加权求和就是线性函数。在考虑时间属性的实体识别方法中[10-13]，根据时间为每个属性分配一个权值，综合相似性则为所有属性的加权和。综合相似度能有效地估计一个候选记录对是否对应同一实体。除了使用上述综合相似度方法来计算相似度外，也可以用机器学习中的监督方法(如支持向量机、决策树、EM 算法[3,19]和主动学习[20])计算记录对的相似性。相似度计算模块的输入是候选数据对象的集合，输出是每个候选对与其相似度组成的三元组。

2.1.2 聚类决定模块

该模块基于候选记录对的相似度，把表示同一个实体的记录放到一个簇中。在前一阶段作出表象局部相似性判断后，可以对实体进行邻接性聚类、相关性聚类或密度聚类，利用相似度阈值以及传递闭包确定记录是否属于同一个簇。Li 等[10]逐个判断每一个簇和记录的相似度阈值，选择相似度最大的记录和簇：如果这个相似度的值大于预先设定好的阈值，则记录和簇中表示同一个实体，从而把记录和簇合并，否则为记录新建一个簇。如果记录与多个簇的相似度都大于阈值，则判断是否将两个簇合并。文献[21-26]使用多个已有的聚类算法对数据集合进行聚类来得到匹配结果，获得了比基于阈值的匹配方法更好的识别结果。聚类决定模块的输入是相似对集合，输出是识别结果。

本文重点研究相似度匹配问题以及匹配后的聚类问题，针对监督的实体识别提出基于随机森林的实体识别方法。

### 2.2　问题描述

真实世界的实体用 $E$ 表示，一个实体可

能被多个数据记录(用 $r$ 表示)所描述，每一个记录都包括 $k$ 个特征，属性的特征集记作 $A=\{a_1,a_2,\cdots,a_k\}$，数据对象集合记 $R=(r_1,r_2,\cdots,r_n)$。对于任何一个记录 $r_i\in\mathbf{R}$，$r_i\cdot A$ 表示记录 $r_i$ 的特征 $a_i\in A$ 的值。任何一对数据记录都是候选匹配对 $[r_i, r_j]$。相似对是三元组，包括一个候选匹配对和它们的相似度 $Sim_{ij}$，记作 $\psi=[r_i,r_j,Sim_{ij}]$，相似对构成一个集合 $\Psi=\{\psi\}$。

本文提出采用随机森林方法计算记录的相似性，根据相似性把表示同一个实体的记录放到相同的簇中。因此，同一个簇中的记录表示同一个实体，不同簇中的记录表示不同的实体。

## 3 随机森林

本文模型中采用余弦相似度来度量记录属性的相似度，在计算记录的相似性时采用了随机森林的方法。这是因为随机森林对有偏差的数据有很好的泛化能力。它是以决策树为基学习器，可构建多个基学习器，且每个基学习器都能得到记录的相似性结果，综合所有基学习器的结果，得到最终的相似性结果。

### 3.1 决策树

决策树是一个树结构，每个非叶节点表示一个特征属性上的判断，每个分支代表这个特征属性在某个值域上的输出，而每个叶节点存放一个类别。使用决策树进行决策的过程就是从根节点开始，测试待分类项中相应的特征属性，并按照其值选择输出分支，直到到达叶子节点，将叶子节点存放的类别作为决策结果。在这个问题中，根节点的输入是两个记录的属性集合，中间节点表示对某个属性的决策，叶节点的输出结果表示两条记录是否对应于同一个实体。

本文采用 ID3 方法。该方法以信息增益和信息增益率两种方法选择分裂特征。首先计算信息熵，用来衡量样本集合纯度的指标，其中 $p_k\left(k=1,2,\cdots,|y|\right)$ 表示在集合 $D$ 中第 $k$ 类样本所占的比例。则信息增益的计算公式为：

$$\mathrm{Ent}(D)=-\sum_{k=1}^{|y|}p_k\log_2 p_k \tag{1}$$

其次，给定样本集 $D$ 和连续属性 $a$，将属性 $a$ 上出现的 $n$ 个值按从大到小排序，记为 $\{a^1,a^2,\cdots,a^n\}$，基于划分点 $t$ 将 $D$ 分为子集 $D_t^-$ 和 $D_t^+$。其中，$D_t^-$ 包含那些在属性 $a$ 上取值小于 $t$ 的样本，而 $D_t^+$ 则包含那些在属性 $a$ 上取值大于 $t$ 的样本，即把 $\left[a^i,a^{i+1}\right)$ 的中位点作为候选划分点。其中，划分点集合为：

$$T_a=\left\{\frac{a^i+a^{i+1}}{2}\middle|1\leqslant i\leqslant n-1\right\} \tag{2}$$

最后分别利用信息增益(Gain)和信息增益率(Gain_ratio)来考察这些划分点，选择使 Gain($D$, $a$, $t$) 最大的划分点 $t$ 和对应的特征 $a$ 作为切分点；以及使 Gain_ ratio($D$, $a$, $t$) 最大的划分点 $t$ 和对应的特征 $a$ 作为切分点。

### 3.2 随机森林

到目前为止，基于随机森林的方法解决了很多实际问题[27]。本文采用随机森林的方法计算记录的相似度。随机森林是用随机的方式建立一个森林，森林由很多决策树组成，但决策树之间不具有关联性。当有一个新的输入样本进入时，用森林中的每一棵决策树分别进行判断，得到该样本应该属于哪一类(对于分类算法)；之后判断哪一类被选择得最多，据此预测该样本为选择最多的那一类。而本文提出的随机森林的方法，输入的是一个记录对，通过决策树判断该记录对是否表示同一个实体，整合所有决策树的结果，得到记录对最终的相似性。

建立决策树的过程中，需要注意两点：采样和完全分裂。首先是两个随机采样的过程，随机森林对输入的数据要进行行、列采样。对于行采样，采用有放回的方式，也就是在采样得到的样本集合中，可能有重复的样本。假设输入样本

为 $N$ 个，那么采样的样本为 $n$($n<N$)个。这样使得在训练的时候，每一棵树的输入样本都不是全部的样本，从而不容易出现过拟合。然后进行列采样，从 $M$ 个特征中，选择 $m$ 个($m\ll M$)，一般地，令 $m=\log_2 M$。之后对采样的数据使用完全分裂的方式建立决策树，这样决策树的叶子节点要么是无法继续分裂的，要么里面的所有样本都是指向同一个分类。

## 4　算法模型

### 4.1　基于随机森林的相似性算法

计算两条记录的相似性是实体识别过程中的基础。针对这个问题，本文提出了一个随机森林算法来计算记录的相似性。该算法的思想是，从样本集 $N$ 中随机选择 $n$ 个样本和 $m$ 个属性，利用所选的样本和属性构建一棵决策树，重复这个过程。决策树的输出结果用来计算两个记录是否表示同一个实体，构建随机森林的详细算法见表 1。

如算法 1 所示，先把记录集中所有记录两两配对，组成一个三元组 $\Psi=\left[r_i,r_j,Sim_{ij}\right]$。其中，$r_i$、$r_j$ 表示两条记录；$Sim_{ij}$ 表示匹配结果。如果两个记录对应同一个实体，则 $Sim_{ij}=1$；如果两个记录对应不同实体，则 $Sim_{ij}=0$。然后把这个三元组添加到集合 $N$ 中(第 2 行)，有放回地从训练集 $R$ 中随机选择 $n$ 个样本(每次随机选择一个样本，然后返回继续选择)。将选择好的 $n$ 个样本作为决策树根节点处的样本(第 4 行)，一般选择 $N$ 中的 80%，即 $n=N\times 80\%$。接着，进行列采样，每次随机地从所有特征中选择 $m$ 个特征满足条件 $m\ll M$(第 5 行)。通过算法 1，已选择每棵决策树的数据集和特征，接下来就是利用算法 2(如表 2)对 $n$ 个样本记录和 $m$ 个特征构建决策树，此时的决策树是多变量决策树。

多变量决策树是指每次选择完一个切分点 $a_j$ 时，并不将 $a_j$ 从原始的特征集 $A$ 中剔除，下次选择的时候仍然是从全部选中的特征集中选择特征和特征值中选择切分点，即每次都对所有的特征计算信息增益或信息增益率的值。

**表 1　随机森林的算法**

**Table 1　The algorithm of random forest**

| **算法 1**　随机森林的算法 | |
|---|---|
| 输入：所有的训练记录集 $R$，记录集对应的实体集 $E$，产生决策树的数量 $k$，每颗决策树的属性个数 $m(m<n)$ | |
| 输出：$k$ 棵决策树 | |
| (1) | 初始化决策树数目 $n=0$，初始化记录集 $N=\{\}$ |
| (2) | 将训练集 $R$ 中的任意两个记录转化为三元组并添加到 $N$ 中 |
| (3) | repeat |
| (4) | 随机从 $N$ 中选择 $n$ 个样本 |
| (5) | 随机从这 $M$ 个特征中选取出 $m$ 个特征 |
| (6) | 利用(4)(5)中选择的 $n$ 个样本和 $m$ 个特征通过算法 2 来构建决策树 |
| (7) | 终止本层循环 |
| (8) | Until 构建完 $k$ 棵决策树 |
| (9) | 输出最终的 $k$ 棵决策树 |

**表 2　连续值的多变量决策树的训练算法**

**Table 2　The training algorithm of continuous value and multivariable decision tree**

| **算法 2**　连续值的多变量决策树的训练算法 | |
|---|---|
| 输入：特征集 $A=(a_1,\cdots,a_m)$；信息增益的阈值 $\theta$ | |
| 训练集 $D=\{[r_1,r_2,w_{12}],\cdots,[r_1,r_n,w_{1n}],\cdots,[r_{n-1},r_n,w_{n-1n}]\}$ | |
| 输出：决策树 $T$ | |
| (1) | 如果 $D$ 中所有实例属于同一类 $C_k$，置 $T$ 为单结点树； |
| (2) | 计算 $A$ 中每个特征 $a_i$ 在划分点 $mid$ 处的信息增益值 |
| (3) | 选择信息增益最大的特征 $a_{\max}$ 以及其对应的特征值 $t$ |
| (4) | 如果最大的信息增益$<\theta$ |
| (5) | 该子树不能再分裂，选择类别数目最多的节点类别作为该子树的标签，返回 $T$ |
| (6) | 否则 |
| (7) | 依 $a_{\max}>t$ 和 $a_{\max}<t$ 将 $D$ 分割为两个非空子集 $D_i^-$ 和 $D_i^+$ |
| (8) | 在 $D_i^-$ 和 $D_i^+$ 上构建子节点，由节点及其子节点构成树 $T$，返回 $T$ |
| (9) | 在两个子树上重复 1～9 步，直到子树不能再分裂，返回 $T$ |
| (10) | 输出最后的决策树 $T$ |

多变量决策树的构建由算法 2 实现。该算法基本思想是，利用公式(2)计算所有特征的划分点集合，并计算特征和对应划分点的信息增益的值，最终选择信息增益最大的值对应的切分点(特征、特征值)，一步步构建决策树。在这个算法中，先判断这些实例是否属于同一个类，如果 $D$ 中所有实例属于同一类 $C_k$，则置 $T$ 为单节点树，并将 $C_k$ 作为该节点的类，返回 $T$(第 1 行)；否则根据某特征的特征值对记录排序，选择两个连续的特征值的中值作为切分点的特征值，选择信息增益最大的切分 $(a_{\max},t)$(第 2～3 行)。如果切分点信息增益的值小于预先设定的 $\theta$，则此时树不再分裂，返回 $T$(第 4～6 行)；否则选择对应的特征和特征值作为切分点，根据切分点把数据集切分为两部分(第 7～8 行)，对应于一棵子树的两个分支，分别重复地在上面的分支上计算信息增益，选择切分点、切分子树，直到子树不能再分裂(第 9 行)。

以上选用的是计算信息增益的最大值，同时还可以计算信息增益率的最大值，其过程除了计算公式与前者不同外，其他步骤完全相同。

通过上述过程，已经构建完 $k$ 棵决策树。在计算两条记录的相似性时，需要用所有决策树的结果判断每棵决策树的结果是 1 或 0。因此，最后的相似性使用公式(3)计算。

$$Sim(r_1,r_2)=\frac{n_1}{k} \tag{3}$$

其中，$n_1$ 表示投票结果是 1 的决策树个数。通过上式可知，$Sim(r_1, r_2)$ 的值越大，则两个记录的相似性越高；该值越小，则两条记录的相似性越低。其中，最高相似性的值为 1，表示所有的决策树都认为这两个记录对应的是同一个实体；最低相似性的值是 0，表示所有的决策树都认为这两个记录不表示同一个实体。

### 4.2 记录的聚类模型

通过上文的计算，可以得到任何两条记录的相似性。把具有高度相似性的记录对合并成簇，即表示同一个实体的记录合并；使表示同一个实体的记录都放在相同的簇中，表示不同实体的记录放在不同的簇中。这个过程分为两个阶段：第一阶段是核聚类，该过程把能够确定的具有高度相似的记录放在同一个簇中；第二阶段是边缘聚类，或合并剩余的记录和已知的簇，或合并两个已知簇(如图 2)。核聚类主要是指利用传递闭包的思想，如果 $Sim(r_1, r_2)=1$ 且 $Sim(r_1, r_3)=1$，则可以判断 $r_1$、$r_2$、$r_3$ 表示的是同一个实体，即 $r_1$、$r_2$、$r_3$ 位于同一个簇中。所有的记录经过上述判断，把表示通过传递关系得到的相似度为 1 的记录对放在相同的簇中，可以得到几个核心簇，每一个核心簇对应着同一个实体，且每个记录仅属于一个实体。接着就是利用核心簇的结果进行边缘聚类。具体过程如算法 3(表 3)所示。

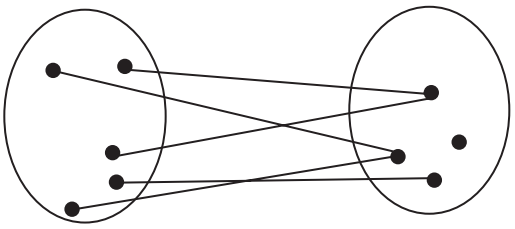


**图 2 边缘聚类中合并两个簇的情况**

**Fig. 2 The case of merging two clusters**

在以上的边缘聚类算法中，主要处理记录的相似性在 $e$～1 的记录对。将 $D$ 中的所有结果分类，把相似度范围为 $e<Sim(r_1,r_2)<1$ 的三元 $[r_1,r_2,Sim(r_1,r_2)]$ 添加到一个集合 $B$ 中(第 2 行)。对于三元组中的数，如果 $r_1$、$r_2$ 位于同一个簇中，则说明两个记录通过前面的传递闭包算法，已经合并，无需再考虑两个记录(第 5～6 行)。如果 $r_1$、$r_2$ 位于两个不同簇中，则先暂时不考虑这两个记录，为两个记录对应的簇新建一个三元组 $[c_i,c_k,1]$。如果三元组 $[c_i,c_k,m]$ 或 $[c_k,c_i,m]$ ($m>0$)在 $F$ 中已经存在，则更新三元组中 $m$ 值，令 $m=m+1$；否则将三元组 $[c_i,c_k,1]$ 添加到 $F$ 中(第 7～12 行)。如果两个记录 $r_1$、$r_2$ 有一个记录

**表 3　边缘聚类算法**

**Table 3　The algorithm of edge clustering**

**算法 3**　边缘聚类算法

输入：随机森林计算的相似性结果 $D$，相似度阈值 $e$，核聚类结果 $C=\{c_1,c_2,\cdots,c_n\}$

输出：最终的聚类结果 $C$

(1)　初始化一个集合 $B$ 和一个集合 $F$

(2)　将 $D$ 中的所有结果分类，把相似度范围为 $[e,1]$ 的三元组添加到一个集合 $B$ 中

(3)　取集合 $B$ 中的三元组 $[r_1,r_2,Sim(r_1,r_2)]$

(4)　repeat

(5)　　如果三元组中的两个记录 $r_1\in c_i,r_2\in c_k$ 且 $i=k$

(6)　　将两个记录 $r_1$、$r_2$ 添加到同一个簇 $c_k$ 中

(7)　　如果三元组中两个记录 $r_1\in c_i$，$r_2\in c_k$，其中 $i,k=1,\cdots,n$ 且 $i\neq k$

(8)　　为记录 $F$ 中 $r_1$、$r_2$ 新建三元组 $[c_i,c_k,1]$

(9)　　若存在三元组 $[c_i,c_k,m]$ 或 $[c_k,c_i,m]$，其中 $m>0$

(10)　　更新三元组的值为 $[c_i,c_k,m+1]$

(11)　　否则

(12)　　把三元组 $[c_i,c_k,1]$ 插入到 $F$ 中

(13)　　如果三元组中的两个记录 $r_1\in c_i,\neg r_2\in C$，其中 $i=1,\cdots,n$

(14)　　计算 $r_2$ 与 $c_i$ 中相似度大于 $e$ 的个数 $p$

(15)　　如果 $p\geqslant\log(|c_i|)$，将两个记录 $r_1$、$r_2$ 添加到 $c_i$ 中

(16)　　否则为两个记录 $r_1$、$r_2$ 新建簇 $c_{n+1}=\{r_2\}$，把 $c_{n+1}$ 添加到 $C$ 中

(17)　　如果三元组中两个记录 $\neg r_1\in C,\neg r_2\in C$

(18)　　为两个记录 $r_1$、$r_2$ 新建簇 $c_{n+1}=\{r_1,r_2\}$，把 $c_{n+1}$ 添加到 $C$ 中

(19)　　将三元组 $[r_1,r_2,Sim(r_1,r_2)]$ 从 $B$ 中删除

(20)　　终止本层循环

(21)　Until 集合 $B$ 为空

(22)　取集合 $F$ 中的一个记录三元组 $[c_i,c_k,m]$

(23)　Repeat

(24)　　如果 $m\geqslant\log[\max(|c_i|,|c_k|)]$

(25)　　$c_i,c_k$ 中的记录合并到一个簇中

(26)　　将三元组 $[c_i,c_k,m]$ 从集合 $F$ 中删除

(27)　　终止本层循环

(28)　Until 集合 $F$ 为空

(29)　输出聚类结果 $C$

$r_1$ 位于已知的簇 $c_i$ 中，另一个 $r_2$ 没有位于任何一个已知的簇中，计算 $r_2$ 与 $c_i$ 中记录相似度大于 $e$ 的个数，记为 $p$。如果 $p\geqslant\log(|c_i|)$，则将两个记录都和这个已知的簇合并，否则为 $r_2$ 新建一个簇(第 13～18 行)。如果两个记录 $r_1$、$r_2$ 没有位于任何一个已知的簇中，则为 $r_1$、$r_2$ 新建一个簇(第 19～20 行)。接着对于 $F$ 中的三元组遍历，如图 3 所示，如果 $[c_i,c_k,m]$ 中 $m$ 的值为两个簇中连接线的个数，且 $m\geqslant\log[\max(|c_i|,|c_k|)]$ 的值，则将两个簇合并(第 25～28 行)并将三元组 $[c_i,c_k,m]$ 从集合 $F$ 中删除，直到 $F$ 为空。

### 4.3　基于随机森林的相似性改进算法

利用算法 2 构建决策树时，在一棵树的分支部分可能会出现如图 3(a)所示的情况。相似度小于 0.68 的记录对表示同一个实体，而相似度大于 0.68 的记录对反而表示不是同一个实体，这显然与实际是相悖的。

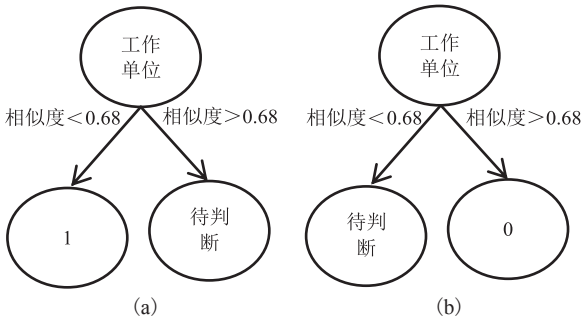



**图 3　决策树分裂后产生与事实相悖情况的子树图**

**Fig. 3　The subtree is contrary to the facts in the decision tree**

对于上图中的问题，有多种解决办法，本文采用了一个强制手段。如果按切分点划分时出现了图 3 中的这种情况，强制在该阶段不能以该特征和特征值作为切分点。因此进一步对算法 2 进行改进。在每一次选择完切分点的时候进行检查，检查该切分点中的 $D^-$ 部分会不会输出结果为 1(表示同一个实体)，如果是，则需要重新选择下一个信息增益最大的点，再判断是否会产生这种现象。除此之外，还有一种情况如图 3(b)，即 $D^+$ 部分的结果会输出 0(表示不同实体)，即当属性的相似度大于某一个值时表示不同的实

体，当属性的相似度小于某一个值时反而可能表示相同的实体。这种情况与前面采取的措施是一样的，即对每一个分割点增加一次判断。最终的算法如算法 4(表 4)所示，该算法与算法 2 基本相同，只是在第 7 行后增加了一次判断：对每一个 $D^-$ 部分判断其结果是不是全是 0(表示不同实体)；对每个 $D^+$ 部分判断其结果是不是全是 1(表示相同实体)，如果不是才可以选择该点作为切分点，否则重新选择新的切分点并判断。

**表 4 连续值的多变量决策树的改进算法**

**Table 4 The improved training algorithm of continuous value and multivariable decision tree**

**算法 4** 连续值的多变量决策树的改进算法

输入：特征集 $A=(a_1,\cdots,a_m)$；信息增益的阈值 $\theta$

训练集 $D=\{[r_1,r_2,w_{12}],\cdots,[r_1,r_n,w_{1n}],\cdots,[r_{n-1},r_n,w_{n-1n}]\}$

输出：决策树 $T$

(1) 如果 $D$ 中所有实例属同一类 $C_k$，置 $T$ 为单结点树

(2) 计算 $A$ 中每个特征 $a_i$ 在划分点 $mid$ 处信息增益值

(3) 选择信息增益最大的特征 $a_{\max}$ 及其对应的特征值 $t$

(4) 如果最大的信息增益$<\theta$

(5) 该子树不能再分裂，选择类别数目最多的节点类别作为为该子树的标签，返回 $T$

(6) 否则

(7) 依 $a_{\max}>t$ 和 $a_{\max}<t$ 将 $D$ 分割为两个非空子集 $D_i^-$ 和 $D_i^+$，判断 $D_i^-$ 中的所有结果是否输出都为 1 或 $D_i^+$ 中的所有结果是否输出为 0

(8) 执行(4)，选择与该切分点不同的使信息增益最大的的切分点，终止本次循环

(9) 否则

(10) 在 $D_i^-$ 和 $D_i^+$ 上构建子节点，由节点及其子节点构成树 $T$，返回 $T$

(11) 在两个子树上重复(1)～(10)步骤，直到子树不能再分裂，返返回 $T$

(12) 输出最后的决策树 $T$

## 5 实验评价

### 5.1 实验设置

实验使用基于 DBLP 数据创建的数据集。该数据集包含了 12 401 条引文记录，对应于 1 239 个实体，其属性包含引文作者姓名、引文标题、引文作者单位、引文的其他作者、引文发表时间。属于同一个实体的某些属性可能不同，但属于不同实体的某些属性值也有可能相同。本文通过在此数据集上实验，对本文提出的基于随机森林的实体识别算法的性能进行测试。

对算法的性能采用准确率和召回率进行评价，采用准确率和召回率的调和平均数 $F$ 评价实体识别结果的精度。

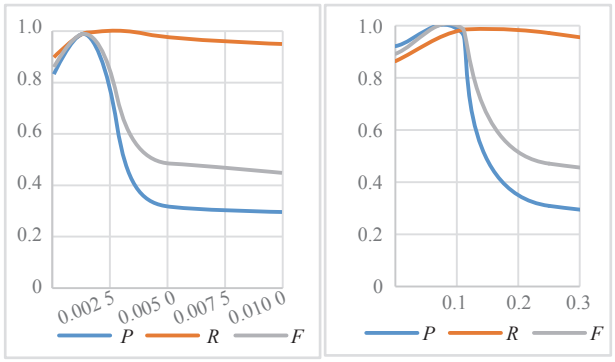
实验采用 Intel(R)Core(TM)i7-26003.4 GHz 处理器，8 G 内存，Microsoft Windows 864 位操作系统进行数据处理。

本实验旨在解决，变化频繁且变化与时间没有规律的记录的实体识别问题，从计算记录的相似度，到最终的聚类算法都提供了一个新型的解决方案。

### 5.2 实验结果与分析

5.2.1 参数测试

在 DBLP 数据集上测试决策树构建算法(算法 2)中的信息增益和信息增益率的阈值参数 $\theta$ 对最终结果的影响。由图 4 可以看出，随着 $\theta$ 增加，准确率($P$)变化趋势是先增高后下降，最终趋于稳定：在开始阶段逐渐增高，达到最大值后开始下降，最后基本保持不变。召回率($R$)的值随着 $\theta$ 的增大而增加，达到了最大值后随着 $\theta$ 的增加缓慢减少。精确性($F$)的走向基本和准确率的一样：在开始阶段逐渐增高，达到最大值后开始下降，最终趋于稳定。在信息增益的试验中，最终结果最好的 $\theta$ 取值为 0.001 6 时，准确率、召回率和精确性都很高；当 $\theta>0.005$ 时，准确率、精度值都很低。在信息增益率的试验中，当 $\theta$ 取值为 0.09 时，准确率、召回率和精确性都很高；当 $\theta>0.2$ 时，准确率、精度值都很低。阈值较低时造成了过拟合，阈值较高时造成了欠拟合，造成了最终结果出现偏差，准确率和 $F$ 精度值都很低。

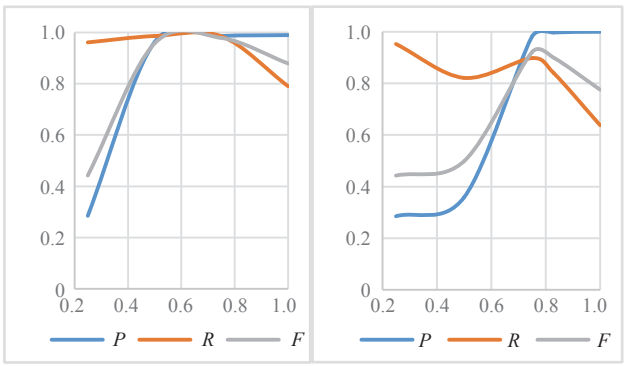



(a) 最终记录链接结果与信息增益阈值 $\theta$ 的关系　　(b) 最终记录链接结果与信息增益率阈值 $\theta$ 的关系

**图 4　在 DBLP 数据集上测试参数 $\theta$ 对聚类的影响**

**Fig. 4　Tests of the parameter $\theta$ influence on clustering on DBLP**

另外，还检测了 DBLP 数据集上聚类算法(算法 4)中的相似度阈值参数 $e$ 对算法效果的影响。从图 5(a)可以看出，准确率($P$)在开始阶段逐渐提高，当 $P$ 达到最大值后，随着 $e$ 的一直增大，$P$ 仍能保持较高的水平。召回率($R$)随着 $e$ 的增大一直维持在较高的水平，之后随着 $e$ 的增加开始下降。精确性($F$)的走向基本和准确率是一样的，随着 $e$ 的增大，精确性一直增大到最大值，之后随着 $e$ 的增大开始减小。从图 5(b)可以看到 3 条曲线的走势和图 5(a)是一致的。产生这种现象的原因是，相似度阈值越高，越能保证同一个簇中的记录越准确，即准确率越高，但是簇中发现的记录越不全，即召回率越低。




(a) 在使用信息增益方法下，最终记录链接结果与相似度阈值 $e$ 的关系　　(b) 在使用信息增益率方法下，最终记录链接结果与相似度阈值 $e$ 的关系

**图 5　在 DBLP 数据集上测试参数 $e$ 对聚类的影响**

**Fig. 5　Tests of the parameter $e$ influence on clustering on DBLP**

5.2.2 算法性能对比

本文提出了一个随机森林的相似度计算方法(RFBas)(见算法 2)以及基于该随机森林的相似度算法的改进算法(RF)。改进算法考虑了事实情况，构建决策树时，当按某个相似度分类时，相似度大于某一个值判断为不同实体，相似度小于某一个值判断为相同实体，把这些与事实相悖的分类情况剔除，得到最终的聚类结果。两个算法的对比结果如图 6 所示，可以看出改进之后的算法，准确率、召回率和精确性都有所提高。以信息增益为特征选择方法改进后提高了 0.6%，以信息增益率为特征选择方法改进后提高了 2.8%，因此改进后的决策树构建方法优于改进前的决策树构建方法。

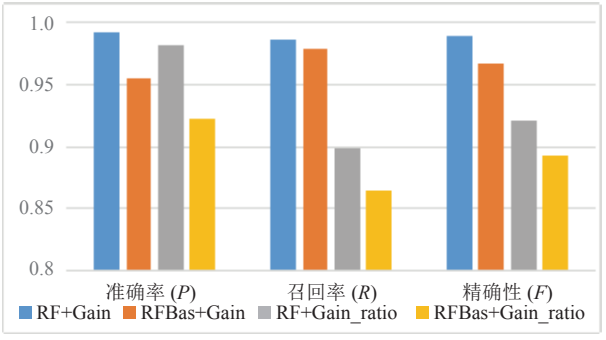



**图 6　基本随机森林算法与改进算法在 DBLP 数据集上对比**

**Fig. 6　Comparisons between basic RF and improved RF on DBLP**

5.2.3 随机森林与决策树方法对比

本文选择了两种划分决策树的方法：信息增益和信息增益率，即基于信息增益的决策树方法和基于信息增益率的决策树方法，与之对应的是基于信息增益的随机森林方法和基于信息增益率的随机森林方法。基于这 4 种方法来计算记录相似性，结果如图 7 所示。由图 7 可以看出，随机森林的方法在准确率、召回率、$F$ 精度值上都比相应的决策树方法要好。产生这种结果的原因是，随机森林的方法综合了所有结果的投票，防

止了数据倾斜。在这个实验中，利用信息增益的方法比利用信息增益率方法的 $F$ 值较高。并且使用信息增益的方法比使用信息增益率算法的 $P$、$R$、$F$ 值都较高，这是因为信息增益率更加适合于特征值种类比较多的情况。

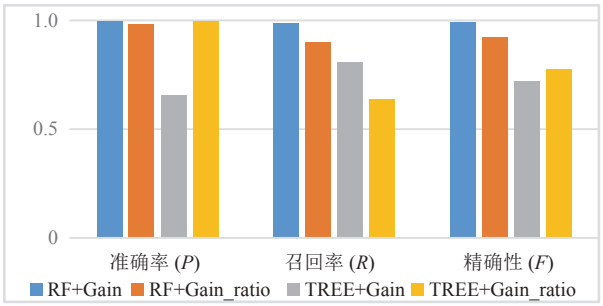



**图 7 基于信息增益和信息增益率的森林算法的对比**

**Fig. 7 Comparisons between RF+Gain and RF+Gain_ratio**

5.2.4 与已有工作对比

传统的相似性一般根据特定匹配规则度量或利用编辑距离、欧式距离计算，在做出表象局部相似性判断后，利用 Rodriguez 等[9]提出的密度聚类。该算法是经典的 $K$-mean 算法的改进算法，不需要指定聚类中心 $k$ 值，并且可以检测非球面类别，通过计算可以确定 $k$ 值，但该算法有一定的局限性。之后 Bie 等[28]又提出了该算法的改进算法，此时确定 $k$ 值不仅仅是依靠图形，还能依靠计算公式直接计算，避免 $k$ 值确定的偶然性。本文主要与 Bie 等[28]提出的算法(CFSFDP)进行对比。同时还和传统的 Partition(简称“Part”)方法对比，结果如图 8 所示。

由图 8 可以看出，在 DBLP 数据集上，RF＋Gain 算法的准确率、召回率和 $F$ 值都明显高于其他算法，$F$ 值顺序为：RF＋Gain＞RF＋Gain_ratio＞CFSFDP＞Part。本文提出的 RFES 算法明显优于其他 3 种算法，以 Part 为基础，在 $F$ 精度值上，CFSFDP 高出了 10%，RF＋Gain_ratio 高出了 32%，RF＋Gain 算法高出 38%。可见，本文提出的基于随机森林的相似度计算方法和聚类方法加一起比其他实体识别方法对于属性值在变化，且变化和时间关系不规律的数据上更加准确。分析其原因，本文提出的方法考虑到了属性的改变，把许多弱分类器的投票综合起来，使结果更加可靠可信，同时提出的聚类算法充分利用上一阶段的结果。因此，对于数据的属性一直在改变，并且这种改变和时间的相关性不大的问题，本文提出的 RF＋Gain 和 RF＋Gain_ratio 算法具有更大的优越性。

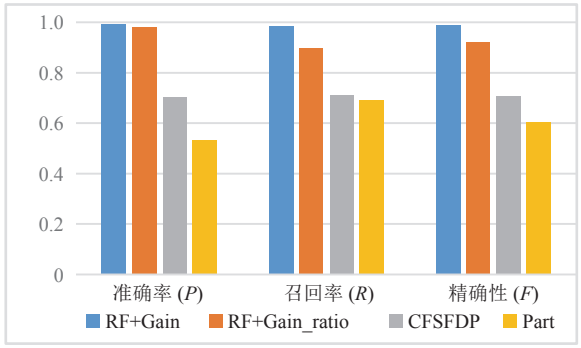



**图 8 改进的随机森林算法与已有聚类算法在 DBLP 上的对比**

**Fig. 8 Comparisons between improved RF algorithm and existing clustering algorithm on DBLP**

除此之外，本文还和 Li 等[10]提出的动态记录链接算法进行比较。该算法中属性的变化和时间是有规律的，某实体的某个属性值，经历的时间越久，变化为不同值的可能性越大；不同实体记录的属性值，经历时间越久，变化为相同值的可能性越大。根据时间为每一个属性分配一个权值，在此基础上提出了 EARLY、LATE、ADJUST 三种聚类算法，这三种算法和本文算法对比结果如图 9 所示。

从图 9 可以看出，RF＋Gain_ratio 算法在准确率、召回率和 $F$ 值都明显高于其他算法，$F$ 值的顺序为：RF＋Gain_ratio＞RF＋Gain＞ADJUST＞LATE＞EARLY，本文提出的 RF＋Gain 和 RF＋Gain_ratio 算法明显优于其他三种算法。其原因为考虑时间特征的记录链接算法，在计算属性的权重时，某属性在改变了一个值后，在很短一段时间内又变回原来值的概率是很低

的，即对属性在一段时间内反复变化或者属性的变化跟时间关系不大的问题建模时是有问题的，它会为这种频繁改变的属性值分配一个很低的权重值，影响最后的聚类结果。因此在属性变化和时间变化关联性不大的时候，本文提出的算法更具有优越性。

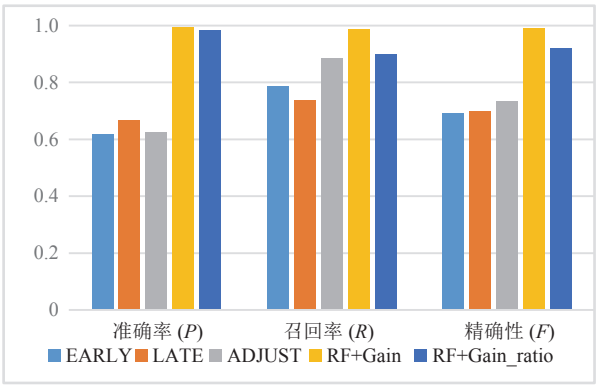



**图 9　改进的随机森林算法与考虑时间特征的算法在 DBLP 上的对比**

**Fig. 9　Comparisons between improved RF algorithm and temporal records linking algorithm on DBLP**

## 6　总　结

实体识别对于数据集成和数据分析是非常重要的。本文针对演化数据的实体识别问题，提出了一个基于随机森林与两阶段聚类的实体识别模型。在该问题中，通过训练几棵决策树构成随机森林计算记录对的相似性，并提出了一个两阶段的聚类算法。最后通过在 DBLP 数据集上的实验对比和分析，验证了该算法的有效性。

## 参 考 文 献

[1] Li P, Dong XL, Guo ST, et al. Robust group linkage [C] // International Conference on World Wide Web, 2015: 647-657.

[2] Hernández MA, Stolfo SJ. Real-world data is dirty: data cleansing and the merge/purge problem [J]. Data Mining and Knowledge Discovery, 1998, 2(1): 9-37.

[3] Elmagarmid AK, Ipeirotis PG, Verykios VS. Duplicate record detection: a survey [J]. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(1): 1-16.

[4] Sarawagi S, Bhamidipaty A. Interactive deduplication using active learning [C] // Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002: 269-278.

[5] Charikar M, Guruswami V, Wirth A. Clustering with qualitative information [J]. Journal of Computer and System Sciences, 2005, 71(3): 360-383.

[6] Bansal N, Blum A, Chawla S. Correlation clustering [J]. Machine Learning, 2004, 56(1-3): 89-113.

[7] Davies DL, Bouldin DW. A cluster separation measure [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1979, 1(2): 224-227.

[8] Ester M, Kriegel HP, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise [C] // Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, 1996: 226-231.

[9] Rodriguez A, Laio A. Clustering by fast search and find of density peaks [J]. Science, 2014, 344(6191): 1492-1496.

[10] Li P, Dong XL, Maurino A, et al. Linking temporal records [J]. Frontiers of Computer Science, 2012, 6(3): 293-312.

[11] Hu YC, Wang Q, Vatsalan D, et al. Improving temporal record linkage using regression classification [C] // Pacific-Asia Conference on Knowledge Discovery & Data Mining, 2017: 561-573.

[12] Chiang YH, Doan AH, Naughton JF. Tracking entities in the dynamic world: a fast algorithm for matching temporal records [J]. Proceedings of the VLDB Endowment, 2014, 7(6): 469-480.

[13] Chiang YH, Doan AH, Naughton JF. Modeling entity evolution for temporal record matching [C] // ACM SIGMOD International Conference on Management of Data, 2014: 1175-1186.

[14] Li F, Lee ML, Hsu W, et al. Linking temporal

records for profiling entities [C] // SIGMOD'15 Proceedings of the ACM SIGMOD International Conference on Management of Data, 2015: 593-605.

[15] Whang SE, Garcia-Molina H. Entity resolution with evolving rules [J]. Proceedings of the VLDB Endowment, 2010, 3(1-2): 1326-1337.

[16] Whang SE, Garcia-Molina H. Incremental entity resolution on rules and data [J]. The VLDB Journal-The International Journal on Very Large Data Bases, 2014, 23(1): 77-102.

[17] Gruenheid A, Dong XL, Srivastava D. Incremental record linkage [J]. Proceedings of the VLDB Endowment, 2014, 7(9): 697-708.

[18] Cohen W, Ravikumar P, Fienberg S. A comparison of string metrics for matching names and records [C] // KDD Workshop on Data Cleaning & Object Consolidation, 2003: 73-78.

[19] Köpcke H, Thor A, Rahm E. Evaluation of entity resolution approaches on real-world match problems [J]. Proceedings of the VLDB Endowment, 2010, 3(1-2): 484-493.

[20] Arasu A, Götz M, Kaushik R. On active learning of record matching packages [C] // ACM Sigmod International Conference on Management of Data, 2010: 783-794.

[21] Haveliwala T, Gionis A, Indyk P. Scalable techniques for clustering the Web [C] // Third International Workshop on the Web and Databases, 2000: 129-134.

[22] Dongen S. Graph clustering by flow simulation [D]. Utrecht: University of Utrecht, 2000.

[23] Brohée S, Van Helden J. Evaluation of clustering algorithms for protein-protein interaction networks [J]. BMC Bioinformatics, 2006, 7(1): 488.

[24] Flake GW, Tarjan RE, Tsioutsiouliklis K. Graph clustering and minimum cut trees [J]. Internet Mathematics, 2004, 1(4): 385-408.

[25] Cormen TH, Leiserson CE, Rivest RL. Introduction to Algorithms [M]. Cambridge: MIT Press, 1990.

[26] Bansal N, Chiang F, Koudas N, et al. Seeking stable clusters in the blogosphere [C] // VLDB'07 Proceedings of the 33rd International Conference on Very Large Data Bases, 2007: 806-817.

[27] Kim K, Giles CL. Financial entity record linkage with random forests [C] // International Workshop on Data Science for Macro-Modeling, 2016.

[28] Bie RF, Mehmood R, Ruan S. Adaptive fuzzy clustering by fast search and find of density peaks [J]. Personal and Ubiquitous Computing, 2016, 20(5): 785-793.