软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(9):2303−2319 [doi: 10.13328/j.cnki.jos.005043]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向实体识别的聚类算法*

孙琛琛, 申德荣, 寇 月, 聂铁铮, 于 戈

(东北大学 计算机科学与工程学院,辽宁 沈阳 110819)

通讯作者: 孙琛琛, E-mail: dustinchenchen_sun@163.com

摘 要: 实体识别是数据质量的一个重要方面,对于大数据处理不可或缺.已有的实体识别研究工作聚焦于数据对象相似度算法、分块技术和监督的实体识别技术,而非监督的实体识别中匹配决定的问题很少被涉及.提出一种面向实体识别的聚类算法来弥补这个缺失.利用数据对象及其相似度构建带权重的数据对象相似图.聚类过程中,利用相似图上重启式随机游走来动态地计算类簇与结点的相似度.聚类的基本逻辑是,类簇迭代地吸收离它最近的结点.提出数据对象排序方法来优化聚类的顺序,提高聚类精确性;提出了优化的随机游走平稳概率分布计算方法,降低聚类算法开销.通过在真实数据集和生成数据集上的对比实验,验证了该算法的有效性.

关键词: 实体识别;聚类;随机游走模型;簇点相似度;数据对象排序

中图法分类号: TP311

中文引用格式: 孙琛琛,申德荣,寇月,聂铁铮,于戈.面向实体识别的聚类算法.软件学报,2016,27(9):2303−2319. http://www.jos.org.cn/1000-9825/5043.htm

英文引用格式: Sun CC, Shen DR, Kou Y, Nie TZ, Yu G. Entity resolution oriented clustering algorithm. Ruan Jian Xue Bao/ Journal of Software, 2016,27(9):2303−2319 (in Chinese). http://www.jos.org.cn/1000-9825/5043.htm

Entity Resolution Oriented Clustering Algorithm

SUN Chen-Chen, SHEN De-Rong, KOU Yue, NIE Tie-Zheng, YU Ge

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

Abstract: Entity resolution (ER) is a key aspect of data quality and is necessary for big data processing. Existing ER research focuses on data object similarity algorithms, blocking and supervised ER technologies, but pays little attention to matching decision problems in unsupervised ER. This paper proposes a clustering algorithm for ER to complement existing work. The algorithm builds a weighted similarity graph with data objects and their pairwise similarities. During clustering, the similarity between a cluster and a vertex is dynamically computed via random walk with restarts on the similarity graph. The basic logic behind clustering is that a cluster absorbs the nearest neighbor vertex iteratively. A data object ordering method is also proposed to optimize clustering order, promoting clustering accuracy. Further, an improved computation method of random walk's stationary probability distribution is proposed to reduce cost of the clustering algorithm. The evaluation on real datasets and synthetic datasets validates effectiveness of the proposed algorithm.

Key words: entity resolution; clustering; random walk model; cluster-vertex similarity; data object ordering

大数据时代,数据的一个重要特点是多样性(variety)[1],描述现实世界同一实体的数据对象在单个或多个数据源中可能以不同的形式重复地出现,由此导致了数据质量的低质化,成为大数据集成、处理、分析和挖掘的瓶颈.实体识别(entity resolution,简称 ER)作为数据质量的一个重要方面,通过分析脏数据集,将描述同一实体的

* 基金项目: 国家自然科学基金(61472070, 61402213); 国家重点基础研究发展计划(973)(2012CB316201); 教育部基本科研业务费项目(N110404010)

Foundation item: National Natural Science Foundation of China (61472070, 61402213); National Basic Research Program of China (973) (2012CB316201); Fundamental Research Funds for the Central Universities (N110404010)

收稿时间: 2015-09-24; 修改时间: 2016-01-12; 采用时间: 2016-02-22

重复数据对象分到同一个组,从而达到提高数据质量的目的[2−11].

实体识别由相似度计算和匹配决定两个必要部分组成,另外还可以包括一个可选部分:分块(blocking).很多实体识别研究工作聚焦于设计更加精确、高效的数据对象相似度比较算法,如各种文本相似度算法、基于实体关联关系的相似度算法及将不同相似度组合的算法等,以便应用于不同类型的数据对象(如关系型数据、XML数据和关联数据等)和不同的实体识别场景[8,9,12−15].另外,为了避免对整个脏数据集进行笛卡尔集级别的计算,提高实体识别效率,提出了实体识别的分块技术[16−20].分块技术的基本思想是:通过较小代价的预计算,将可能匹配的数据对象分在同一块中,只比较块内的数据对象,节省了大量的无用开销.分块技术对于大数据实体识别尤为不可或缺.

匹配决定的方法按照是否需要训练过程可分为两类:监督的方法和非监督的方法.监督的方法要依赖用户提供准确的训练数据来训练规则或数据模型,来对测试数据进行识别,主要以分类算法为主[2,3,5].训练数据需要领域内专家来标注,然而算法的使用者并不总是领域专家,导致训练数据不易获得,因此造成了监督类方法的局限性.本文的研究重点是非监督的实体识别方法.传统的实体识别通过测定固定阈值来决定两个实体数据对象(简称数据对象或对象)是否匹配[2,3,5],然而此类方法已被证明为不能产生较高精确性的结果[4].Hassanzadeh 等人[4]在匹配阶段利用已有的聚类算法来决定匹配结果,通过对比实验证明,基于聚类算法的方法比基于固定阈值的方法要更精确.Hassanzadeh 等人[4]采用的聚类算法是通用算法,虽然可以完成匹配决定的任务,但并没有考虑实体识别的具体特点.据本文作者所知,目前尚没有专门针对实体识别的聚类算法.本文的工作将弥补这一空白.

本文提出一个基于随机游走模型的图聚类算法 ERC(entity resolution oriented clustering)来解决实体识别中匹配决定的问题.本文的匹配决定流程如图 1 所示:首先,利用相似对集合构建对象相似图,它是一个带权、无向图(如图 1(II)所示);然后,利用 ERC 算法对数据对象进行图聚类,得到实体识别结果(如图 1(III)所示).相似对是三元组,包括两个数据对象和它们的相似度,通过相似度计算可得到相似对集合(如图 1(I)所示).ERC 算法的核心思想是:从一个单例的类簇开始,不断吸收最近的结点,直到类簇和结点距离不满足约束条件;迭代上述聚类步骤至所有结点都归入某个类簇.聚类过程中,簇点之间的相似度通过从类簇到结点的随机游走的平稳概率计算.定义结点的信用度来估计结点在候选队列中的优先级,将结点按照信用度降序排序,保证大的类簇先被发现以及每个类簇被尽量完整地发现,提高聚类结果的精确性.通过实验对比证明,ERC 算法在两个真实数据集上的识别效果优于已有工作.可见,该算法能更准确地发现相同实体对应的数据对象.

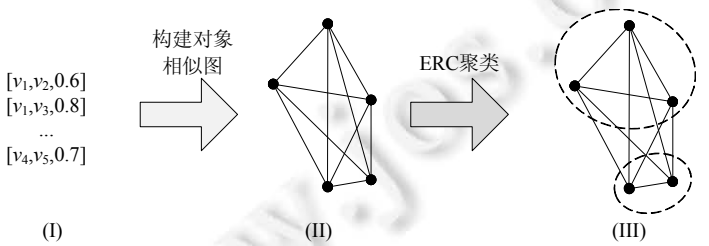


Fig.1 Workflow of match decision

图 1 匹配决定流程

本文的主要贡献如下:

- 提出基于图上随机游走模型的簇点相似度:基本的簇点相似度和双向的簇点相似度.这两个簇点相似度基于以类簇为初始组的随机游走到达目标结点的平稳概率,这样充分考虑了类簇到结点的直接关联和间接关联,从而能动态地、准确地衡量聚类过程中类簇与结点的相似性;
- 提出面向实体识别的聚类算法 ERC.每次选择一个结点作为单例的类簇,让它迭代地吸收距离最近的新结点,直到新结点不满足距离约束;还提出了基于信用度的结点排序来提高聚类结果的精确性;提出一个优化的以多结点类簇为初始组的平稳分布的计算方式,来降低开销,提高聚类速度;

- 在两个真实的数据集和一组生成的数据集上对提出的 ERC 算法进行充分地实验评价,验证了算法的有效性.ERC 算法与已有的聚类算法相比,实体识别结果的精确性更高.

本文第 1 节是准备工作,包括实体识别模型和问题描述.第 2 节定义基本的簇点相似度和双向的簇点相似度.第 3 节首先介绍基本的 ERC 算法框架,然后提出两个优化:数据对象排序和优化的平稳分布计算方式.第 4 节是实验与分析,通过与已有工作对比及自身对比评价 ERC 算法,证明其有效性.第 5 节介绍相关工作.最后,第 6 节总结全文.

1 准备工作

1.1 实体识别模型

如图 2 所示:实体识别模型包括 3 个模块——分块模块、相似度计算模块和匹配决定模块;整个模型的输入是待识别的脏数据集,输出是识别的结果.

- 分块模块

对于大数据背景下的实体识别,如果进行笛卡尔集数量级的处理,那么开销非常大,包含大量无用开销.为了在不影响识别质量的前提下降低开销和提升识别效率,实体识别模型利用分块技术[16−20]对脏数据进行分块处理来缩小搜索空间.分块技术只将有可能匹配的数据对象分在同一对象块中,而不可能匹配的数据对象则被分在不同对象块中,这样可以减少大量无用的对象比较和计算.每个数据对象至少隶属于一个对象块,所有对象块的并集是整个脏数据集;同一对象块中的任何两个对象都是一个候选匹配对(简称候选对),每个候选对都将被比较.分块模块的输入是整个脏数据集,输出是对象块的集合.

- 相似度计算模块

该模块利用对象相似度比较函数对每个候选对进行相似度比较,并得到介于[0,1]的相似度,相似度越大,表示两个对象越有可能对应同一实体,0 表示两对象完全不同,1 表示两对象完全相同.通常,数据对象包括多个属性,且不同属性可能是不同类型的数据,比如一条引文记录包括题目、合作者、期刊名和年份等,其中,题目是文本型数据而年份是数值型数据.对象属性以文本型数据居多,目前已有很多文本相似度函数[15],如 Jaccard, Cosine(TF/IDF),Levenstein distance,Winkler,Jaro,Q-gram 等;不同的文本相似度函数适用于不同类型的文本数据.数值型数据可由用户利用数学公式设计相应的相似度函数.对象相似度比较函数选择数据对象的某些属性(可能一个或多个),针对每个属性调用特定的相似度函数来计算其相似度,然后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[2,3,5],比如加权求和就是线性函数.综合相似度能有效地估计一个候选对是否对应同一实体.相似度计算模块的输入是候选数据对象的集合,输出是由每个候选对及其相似度组织成的集合,简称相似对集合.

- 匹配决定模块

该模块调用匹配函数分析候选对的相似度来决定一个候选对是否匹配.实体识别方法按是否需要训练集可以为:监督的实体识别方法和非监督的实体识别方法[2,3,5].监督的实体识别方法在匹配决定阶段使用分类算法(如支持向量机、决策树、EM 算法[2,3]和主动学习[21]).监督类方法要求用户提供一定规模高质量的训练数据,分类算法通过训练得到组合函数和相应参数;然后,利用组合函数和参数来对新的数据进行分类,即,决定候选对是否匹配.由于训练数据需要领域专家标注,监督类方法的应用范围受到局限.非监督的实体识别方法则不需要训练,直接分析候选对来决定匹配结果.基于阈值的匹配方法是最传统的非监督类方法,它将候选对的相似度与指定阈值进行比较:如果相似度大于等于阈值,则候选对匹配;否则,不匹配[2,3,12].另一种非监督类方法是基于聚类算法[22,23]的匹配方法,Hassanzadeh 等人使用多个已有的聚类算法对数据对象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[4,24−29].匹配决定模块的输入是相似对集合,输出是识别结果.

本文重点研究匹配决定的问题,将针对非监督的实体识别提出基于随机游走模型的 ERC 聚类算法.

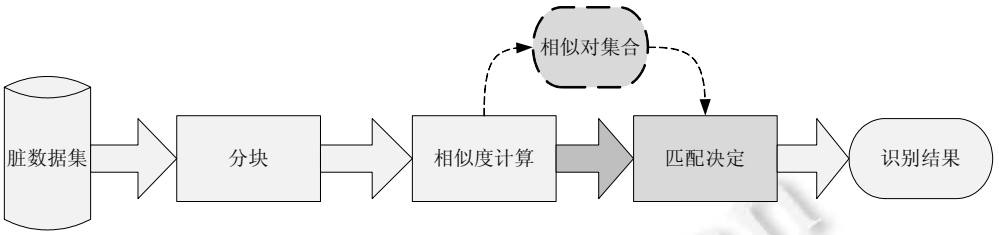


Fig.2 Entity resolution model

图 2 实体识别模型

1.2 问题描述

真实世界的一个实体用ε表示;一个实体可能被多个数据对象所描述,数据对象用v表示,数据对象集合记作V.同一分块中的任何一对数据对象都是候选匹配对,记作$[v_i,v_j]$;相似对是三元组,包括一个候选匹配对和它们的相似度w_{ij},记作$\psi=[v_i,v_j,w_{ij}]$,相似对集合$\Psi=\{\psi\}$.

定义 1. 给定一个数据对象集合 V 和对应的相似对集合 Ψ,面向实体识别的聚类算法利用一个映射函数$\varphi(v|\Psi)=\varepsilon$进行聚类,自主决定类簇数目,将对象集合 V 划分成若干个类簇$\{C\}$,每个类簇与一个特定实体一一对应,$C=\{v|\varphi(v|\Psi)=\varepsilon\}$,简称识别聚类算法.

识别聚类算法应该能自主地决定类簇的数目,聚类结果中的类簇数即实体数.识别聚类算法的输入是相似对集合(包含了数据对象集合),输出是识别结果.

本文将采用图聚类的方法来解决实体识别中匹配决定的问题.聚类前,利用给定的数据对象集合 V 和对应的相似对集合Ψ,按照定义 2 构建数据对象相似性图.

定义 2. 数据对象相似性图 $G=(V,E)$是带权重的无向图,结点 $v\in V$ 是一个数据对象,两结点之间的边 $e_{ij}\in E$ 表示数据对象 v_i 和 v_j 的相似性,相似性大小用边的权重 w_{ij} 表示,简称对象相似图.

本文中,结点和数据对象可以互相指代,结点也表示描述实体的数据对象.

本文提出一种基于图上随机游走模型的聚类算法——ERC 算法,来实现非监督实体识别中匹配决定.ERC 的主要步骤:

(1) 将数据对象按信用度降序排列,并依次插入到候选队列中;

(2) 每次从队首取一个对象,生成一个单例类簇.该类簇通过迭代地吸收距离最近的对象来不断扩大,直到下一个对象与该类簇的相似度不符合约束条件;

(3) 重复步骤(2),直到完成聚类.

步骤(1)中,为了提高聚类结果的精确性,应该优先发现大的类簇和尽量完整地发现每个类簇,因此,定义数据对象的信用度来估计数据对象在候选队列中的优先级,信用度越大,优先级越高(详见定理 3).步骤(2)中,定义基于重启式随机游走的簇点相似度,通过挖掘相似图结构来综合地衡量类簇与结点的相似性.

2 簇点相似度

在提出面向实体识别的聚类算法之前,定义类簇和结点间的相似度(简称簇点相似度):基本的簇点相似度和双向的簇点相似度.给定类簇 C 和结点 v,簇点相似度记作 $sim_{cv}(C,v)$.

本节提出的簇点相似度基于图上的重启式随机游走模型,因此先简述该模型.随机游走模型通过模拟随机游走行为来挖掘图的结构信息[30].给定一个图和随机游走者,游走者以一定概率同时从一组结点(可以是一个或多个结点)出发;每一步,游走者要么以一定概率$(1-\alpha)$到达邻居结点,要么以概率α跳回到初始结点组(即重启过程).游走过程一直迭代直到收敛,即游走者停留在图上每个结点的概率不再变化,达到平稳态.所有平稳概率组成的分布称为随机游走的平稳概率分布.参数α可以控制游走者离开初始结点组的范围.某结点的平稳概率可

以用来衡量该结点跟初始结点组的关联度,平稳概率越大,则关联度越大.本节将利用平稳概率来计算聚类过程中类簇和结点的相似性.

2.1 基本的簇点相似度

为了区别于簇点相似度,将前一步数据对象相似度计算得到的相似度称为数据对象(或结点)语义相似度,简称语义相似度.相似对集合提供了候选对的两两语义相似度,本文不直接利用这些语义相似度进行聚类,而是利用新提出的类簇与结点间的相似度来聚类.簇点相似度的基本思想是:不但考虑类簇到结点的直接关系,还要考虑通过其他结点关联的间接关系,综合地计算类簇与结点之间的关联程度.图上随机游走模型可以充分地挖掘这些信息,隶属于同一类簇的结点将彼此具有较高的平稳概率,从而更准确、全面地计算类簇和结点的相似性.本节定义基本的类簇与结点的相似度,见定义 3.

定义 3(基本的簇点相似度). 给定一个对象相似图 $G=(V,E)$,类簇 C 和结点 v 之间基本的相似度定义为图 G 上以类簇 C 为初始结点组的随机游走的平稳概率,即:

$$sim_{cv}(C\rightarrow v)=\Pr(v|C) \tag{1}$$

特别地,当类簇 C 是单例类簇时,簇点相似度可用来衡量结点与结点之间的相似度,此时,称之为基本的单例簇点相似度,记作 $sim_{cv}(\{v_1\}\rightarrow v_2)$.它将有特别的应用,详见第 2.2 节双向的簇点相似度中 k 近邻集合,第 3.1 节算法框架中算法 1 的初始化类簇的簇点相似度(第 5 行)和第 3.2 节数据对象排序中的信用度.需要指出的是,$sim_{cv}(\{v\}\rightarrow v)\neq 1$.

2.1.1 随机游走计算

连通图上的重启式随机游走是不可约、非周期、有限状态的马式链,因此平稳分布必然存在且唯一[30].给定图 G 和初始结点组 C,令向量$\boldsymbol{\pi}$为平稳分布,$\boldsymbol{P}$ 为规范化的马尔可夫转移矩阵,重启概率α=0.15[31],重启向量 $\boldsymbol{q}$ 中各初始结点对应的分量为 1/|C|,其他分量为 0,则平稳分布满足公式(2):

$$\boldsymbol{\pi}=(1-\alpha)\times\boldsymbol{P}^{\boldsymbol{T}}\times\boldsymbol{\pi}+\alpha\times\boldsymbol{q} \tag{2}$$

转移矩阵 $\boldsymbol{P}$ 可以由公式(3)求得,w 为图上边的权重:

$$\boldsymbol{P}=\{p_{ij}\mid p_{ij}=w_{ij}\Big/\sum w_{i^*}\} \tag{3}$$

平稳分布$\boldsymbol{\pi}$可以通过对公式(2)迭代计算获得.首先,将$\boldsymbol{\pi}$初始化为各初始结点对应的分量为 1/|C|,其他分量为 0,即,游走者以 1/|C|的概率停留在各初始结点;然后,利用公式(2)计算下一步游走者停留在各结点的概率分布;重复第二步的过程,直到相邻两次的概率分布的差值足够小.迭代停止后,最终得到平稳分布$\boldsymbol{\pi}$.

2.2 双向的簇点相似度

基本的簇点相似度是对象相似图上从一个类簇出发后到达某结点的平稳概率,这是一个单向计算的量.然而,双向的相似性对于实体识别非常重要[12],如果一个类簇与一个结点对应同一实体,那么它们应该互相相似.接下来,将提出一个基于候选结点 k 近邻集合的、改进的簇点相似度.

定义 4. 给定对象相似图 G 上的结点 v,那么 v 的 k 个距离最近的邻居结点组成的集合记作 $nn(v,k)$,简称 k 近邻集合.

定义 4 中,结点间距离通过基本的单例簇点相似度来衡量.

定义 5. 给定对象相似图上类簇 C 和候选结点 v,簇点密度系数是类簇 C 中结点 v 的|C|近邻结点所占比例:

$$\rho=\frac{|C\cap nn(v,|C|)|}{|C|} \tag{4}$$

在基本的簇点相似度的基础上,如果类簇与结点的 k 近邻集合的交集越大,那么两者的相似度应该越高.这样不仅考虑了从类簇到结点的相似性,还考虑了从结点到类簇的相似性,可以更准确地衡量类簇与结点间相互的相似性.

定义 6. 给定对象相似图上类簇 C 和候选结点 v,C 与 v 的双向的簇点相似度是基本的相似度乘以簇点密度系数ρ:

$$sim_{cv}(C\leftrightarrow v)=\rho\times sim_{cv}(C\rightarrow v) \tag{5}$$

3 ERC 聚类算法

针对非监督的实体识别中的匹配决定,本节将提出基于随机游走模型的聚类算法——ERC(entity resolution oriented clustering)算法.给定数据对象集合和相应的相似对集合,ERC 算法计算哪些数据对象对应同一实体,将相同实体对应的数据对象放入同一类簇,不同实体对应的数据对象放入不同类簇.同时,ERC 算法能够自主决定类簇数目.第 3.1 节介绍 ERC 算法基本框架.第 3.2 节提出一个数据对象排序方法来提高 ERC 算法的精确性.第 3.3 节提出一个优化的以多结点为初始组的平稳分布计算方式,降低算法整体开销,提高聚类速度.

3.1 算法框架

ERC 聚类的基本思想是,让类簇通过迭代地吸收距离自己最近的结点来不断增大.类簇与其他结点的距离通过第 2 节中的簇点相似度(两个皆可)来计算.ERC 聚类算法的输入是一个对象相似图和一个阈值,该图可按照定义 2 构建;输出是聚类结果,即实体识别的结果.ERC 的详细算法流程见算法 1.

算法 1. ERC 算法框架.

输入:对象相似图 $G=(V,E)$,簇点相似度阈值ξ;

输出:聚类结果Λ.

1. 将 V 中所有结点加入到候选队列 Q 中
2. 初始化类簇集合$\Lambda=\{\}$
3. **repeat**
4. 从 Q 队首取出一个结点 v,并将 v 初始化成一个单例类簇 $C=\{v\}$
5. 初始化类簇 C 的簇点相似度,$C.cvSim=sim_{cv}(\{v\}\rightarrow v)$ //表示类簇 C 吸收上一个结点时的簇点相似度
6. **repeat**
7. 计算以 C 为初始结点组的平稳概率分布$\boldsymbol{\pi}_C$
8. 根据分布$\boldsymbol{\pi}_C$计算 C 和其他结点的簇点相似度
9. 找出候选队列 Q 中距离类簇 C 最近的结点 v^*
10. **if** $Sim_{cv}(C,v^*)\geqslant\xi^* C.cvSim$
11. 类簇 C 吸收结点 v^*
12. $C.cvSim$ 更新为 $Sim_{cv}(C,v^*)$
13. 将 v^*从队列 Q 中删除
14. **else**
15. 将 C 加入到类簇集合Λ
16. **中止本层循环** //对应第 6 行的 **repeat**
17. **until** Q 为∅
18. 输出聚类结果Λ

如算法 1 所示,将图上所有结点加入到一个候选队列(第 1 行);接着,进入聚类的主体迭代过程(第 3 行~第 17 行).从队列队首取结点 v 并生成单例类簇 C,并将其簇点相似度初始化为 $sim_{cv}(\{v\}\rightarrow v)$(第 4 行、第 5 行);然后进入下一层迭代,即,类簇扩大阶段(第 6 行~第 16 行).计算以当前类簇 C 为初始组的平稳概率分布$\boldsymbol{\pi}_C$(第 7 行),计算方式有两种:(1) 第 2.1.1 节中,基本的计算方式;(2) 第 3.3 节中,优化的计算方式.利用平稳概率分布$\boldsymbol{\pi}_C$来计算簇点相似度(基本的簇点相似度和双向的簇点相似度皆可),求得候选队列 Q 中距离当前类簇 C 最近的结点 v^*(第 8 行、第 9 行).如果结点 v^*满足约束条件(第 10 行),当前类簇 C 吸收结点 v^*并更新其簇点相似度(第 11 行、第 12 行),此外还要将 v^*从队列 Q 中删除(第 13 行).当前类簇 C 不断迭代地扩大,直到下一个距离 C 最近的结点不再满足约束条件(第 14 行),然后将当前类簇 C 加入到类簇集合Λ,并跳出内层迭代(第 15 行、第 16 行).当候选队列 Q 为空时,整个聚类算法终止(第 17 行).此时,类簇集合Λ即聚类结果,将其输出(第 18 行).

对 ERC 算法的聚类结果精确性有重要影响的两个因素有:候选队列的顺序和簇点相似度,详细分析如下:

1) 从聚类结果逆向分析可知,大的类簇对整个聚类结果的精确性的影响起主导作用.实体识别的脏数据集中的数据对象分布通常是不均匀的(即,不同实体对应的数据对象数目不相同)[4,6],因此,真实的识别结果中类簇是大小不一的.为了提高识别结果的精确性,应该尽量保证大的类簇被完整地发现.根据算法 1 可知:ERC 算法只对所有结点进行一次遍历,遍历顺序对聚类结果有直接的影响.第 3.2 节将提出一个数据对象排序策略来保证大类簇的优先级和每个类簇尽量被完整地发现,从而提高整个聚类结果的精确性;
2) 基本的簇点相似度是单向性计算,即,从类簇到结点.双向的簇点相似度能更好地衡量实体识别的聚类过程中类簇与结点之间相互的相似性.后文第 4.2.3 节簇点相似度测试中对比实验将验证,双向的簇点相似度比基本的簇点相似度更有助于 ERC 算法产生精确的结果.

对 ERC 算法速度起决定性影响的是随机游走的平稳概率的计算.第 2.2 节双向的簇点相似度中,k 近邻集合计算时和第 3.2 节中数据对象排序时,都要用到每个结点与其他结点的基本的单例簇点相似度(即,以单结点为初始组的平稳概率).考虑到单结点平稳概率必须计算,第 3.3 节提出了优化的簇点相似度计算方式,它将以单结点为初始组的平稳概率进行线性组合来求解以多结点类簇为初始组的平稳概率.这样可以很大地降低聚类算法的整体开销,提高聚类速度.

3.2 数据对象排序

候选队列的顺序从两方面影响聚类结果:(1) 全局来看,类簇的发现顺序;(2) 局部来看,单个类簇的初始结点的选择.为方便描述,将聚类算法求得的聚类结果称为算法类簇集合,而真实的聚类结果(即,完全正确的聚类结果)称为真实类簇集合.

从粗粒度来看,研究类簇集合,通过观察不同的类簇发现顺序产生的聚类结果,分析该如何对结点进行排序.每个实体对应数据对象的分布称为数据对象分布,现实世界中,待识别的数据集的数据对象分布通常是非均匀的,如幂率分布[32],因此,正确的聚类结果中的类簇规模是大小不一的.根据例 1 及图 3,顺序 I 和顺序 II 分别将不同类簇的成员排在候选队列高优先级位置,按照顺序 I,得到算法聚类结果$\{\{v_5,v_2,v_4,v_1,v_3\},\{v_7,v_6\}\}$,和真实类簇集合相同;按照顺序 II,得到结果$\{\{v_7,v_6,v_4\},\{v_5,v_2,v_1,v_3\}\}$,最大类簇 C_1 缺失一个成员.顺序 I 将大的真实类簇 C_1 的成员排在了候选队列高优先级位置,保证了大类簇 C_1 被完整发现;顺序 II 将小的真实类簇 C_2 的成员排在了候选队列高优先级位置,保证了小类簇 C_2 被完整发现,但大类簇 C_1 的发现受影响.根据定理 1 可知,大的类簇的完整性对聚类结果的影响要大于较小的类簇,从而聚类算法应该给予大的类簇较高的优先级以保证其被完整地发现.因此,应该将大的类簇的成员结点排在候选队列中高优先级的位置.

例 1:一个待识别的数据对象集合如图 3 所示,真实聚类集合是$\{C_1=\{v_1,v_2,v_3,v_4,v_5\},C_2=\{v_6,v_7\}\}$,现有 3 个候选队列顺序,分别为顺序 I:$v_5\to v_2\to v_4\to v_3\to v_1\to v_7\to v_6$,顺序 II:$v_7\to v_6\to v_5\to v_4\to v_2\to v_3\to v_1$,顺序 III:$v_4\to v_5\to v_2\to v_3\to v_1\to v_6\to v_7$.根据算法 1,由顺序 I 得到算法类簇集合$\{\{v_5,v_2,v_4,v_1,v_3\},\{v_7,v_6\}\}$,由顺序 II 得到算法类簇集合$\{\{v_7,v_6,v_4\},\{v_5,v_2,v_1,v_3\}\}$,由顺序 III 得到算法类簇集合$\{\{v_4,v_6,v_7\},\{v_5,v_2,v_1,v_3\}\}$.

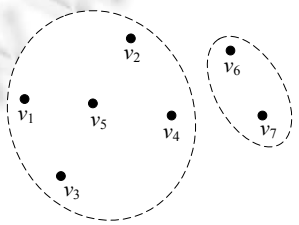


Fig.3 Different candidate queue orders' influences on clustering

图 3 候选队列的不同排序对聚类的影响

定理 1. 对于非均匀分布的脏数据集,大的类簇的完整性对聚类结果精确性影响比小的类簇更大.

证明:令 C 是真实的聚类结果中的一个类簇,现从 C 去除一部分数据对象,记作 ΔC,满足 $|\Delta C|<|C|$.那么,整个聚类结果缺少 $\binom{|C|}{2}-\binom{|C|-|\Delta C|}{2}=\frac{1}{2}(2|C|\times|\Delta C|+|C|-|\Delta C|^2)$ 个匹配对,当 $|\Delta C|$ 固定时,此式可看作是变量为 $|C|$ 的函数,且单调递增.假定缺少相同数目的成员,当 $|C|$ 越大时,对聚类结果的损失越大.因此,大的类簇的完整性对聚类结果精确性影响比小的类簇更大,定理 1 得证. □

从细粒度来看,研究单个类簇形成过程,通过观察选择不同的类簇成员作为初始结点产生的聚类结果,分析如何保证单个类簇被完整地发现.ERC 算法在每个类簇形成过程中,对类簇成员只做一次不可逆遍历.由于不存在回溯过程,初始结点的选择对于一个类簇的形成影响很大,同一类簇内的不同成员作为初始结点,可能会产生不同的聚类结果.根据例 1,顺序 I 和顺序 III 都将真实类簇 C_1 的成员排在了候选队列高优先级位置,按照顺序 I,将得到结果 $\{\{v_5,v_2,v_4,v_1,v_3\},\{v_7,v_6\}\}$,与真实类簇集合相同;按照顺序 III,将得到结果 $\{\{v_4,v_6,v_7\},\{v_5,v_2,v_1,v_3\}\}$,导致最大类簇 C_1 缺失一个成员.顺序 III 把最大类簇的成员点 v_4 当成初始结点,但由于点 v_4 距离 C_1 的中心结点较远,无法保证 C_1 被完整地发现;反观顺序 I,把点 v_5 作为初始结点,它距离 C_1 的中心结点最近,从而保证 C_1 被完整地发现.通过上述对比示例可以发现:就单个类簇而言,应该选择距离真实类簇的中心结点最近的结点作为初始结点,因此,应该把靠近类簇中心结点的结点排在候选队列中高优先级位置.

定义 7. 真实类簇的中心结点是虚拟结点,它距离簇内成员结点的平均距离最小.

定义 8. 给定数据对象集合 V,数据对象 $v\in V$ 的信用度定义为,其他数据对象对于 v 应当获得在候选队列中优先级的支持度的总和:

$$\theta(v)=\sum_{v'\neq v,v'\in V}\varsigma(v'\to v) \tag{6}$$

其中,支持度是以 $\{v\}$ 为起始组的基本的单例簇点相似度,$\varsigma(v'\to v)=Sim_{cv}(\{v\}\to v')$.

定理 2. 对于非均匀分布的脏数据集,对数据对象按照信用度进行降序排列的聚类结果,比随机排列的聚类结果更精确.

证明:要想证明按对象信用度降序排列的聚类结果比随机排列的更精确,先证明以信用度降序排列为条件的两个子结论:

① 全局地,大的类簇将被优先发现,即,越大的真实类簇的成员结点在候选队列中被赋予越高的优先级;

② 局部地,每个类簇被尽量完整地发现,即:类簇内部越靠近中心结点的结点,在候选队列中被赋予越高的优先级.

令非均匀分布的脏数据集 V 对应的真实聚类集合为 $C(V)$,$C\in C(V)$ 是其中任意一个真实类簇,$v\in C$ 是 C 中任意一个结点,那么 v 的信用度的估计值为 $\theta(v)\approx|C|\times s_{in}(v)+(|V|-|C|)\times s_{out}(v)$,$s_{in}(v)$ 是 v 与类簇内部其他结点之间的平均相似度,$s_{out}(v)$ 是 v 与其他类簇的成员结点之间的平均相似度.

(1) 从全局来分析类簇集合.

所有类簇内部结点之间的平均相似度记作 s_{avgi},不同类簇间结点之间的平均相似度记作 s_{avgo},据聚类的定义[26−28],应该满足不等式 $s_{avgi}>>s_{avgo}$.令 $s_{in}(v)=s_{avgi}$,$s_{out}(v)=s_{avgo}$,那么,

$$\theta(v)\approx|C|\times s_{avgi}+(|V|-|C|)\times s_{avgo}=|C|\times(s_{avgi}-s_{avgo})+|V|\times s_{avgo}=f(|C|).$$

该估计值是关于 $|C|$ 的单调递增函数.因此,越大的真实类簇的成员结点的平均信用度越大,在候选队列中被赋予的优先级越高,子结论①得证.

(2) 从局部来分析单个类簇内部.

$v_i,v_j\in C$,v_C 是 C 的中心结点,令 v_i 比 v_j 更靠近 v_C,根据中心结点的定义,越靠近中心结点位置的结点距离簇内成员的平均距离越小,即平均相似度越大,$s_{in}(v_i)>s_{in}(v_j)$;而 $s_{avgi}>>s_{avgo}$,可以认为 $s_{out}(v_i)=s_{out}(v_j)=s_{avgo}$,那么,

$$\theta(v)\approx|C|\times s_{in}(v)+(|V|-|C|)\times s_{avgo}=g(s_{in}(v)).$$

该估计值是关于 $s_{in}(v)$ 的单调递增函数.因此,类簇内部越靠近中心结点的结点的信用度估计值越大,那么在候选队列中,v_i 比 v_j 获得更高的优先级,子结论②得证.

根据子结论①和定理 1,从类簇集合级别分析,可知信用度降序排列的聚类结果比随机排列的要更优;子结

论②则从单个类簇级别进一步证明,信用度降序排列的聚类结果比随机排列的要更优.定理 2 得证. □

数据对象排序应该插入到算法 1 中的第 1 行,把候选队列 Q 按照数据对象的信用度降序排列.在以单结点为起始组的平稳分布的集合提前计算好的前提下,对结点按信用度进行排序的算法复杂度为 $O(n\times\log n)$,n 是结点数.

3.3 计算优化

第 2.2 节中,双向的簇点相似度中 k 近邻集合计算时,可能要用到每个结点与其他结点的基本的单例簇点相似度;第 3.2 节中数据对象排序时,必须用到每个结点与其他结点的基本的单例簇点相似度.鉴于这两点,本节提出优化的以多结点类簇为初始组的平稳概率分布的计算方式,可以有效地提高聚类速度.计算优化的基本思想是:提前计算好以单结点为初始组的平稳概率分布,当需要计算某个类簇和其他结点的平稳概率分布时,直接利用以单结点为初始组的平稳概率分布进行线性组合来求解,以达到从全局上降低开销的目的.这个计算方式的依据是定理 3.

定理 3. 给定对象相似图 G,C 是图上一个类簇,令 $\boldsymbol{\pi}_C$ 是以类簇 C 为初始结点组的随机游走的平稳概率分布,$\boldsymbol{\pi}_v$ 是以单个结点 v 为初始结点的随机游走的平稳概率分布,那么,

$$\boldsymbol{\pi}_C=\frac{1}{|C|}\times\sum_{v\in C}\boldsymbol{\pi}_v \tag{7}$$

证明:以结点 v 为初始结点的平稳概率分布 $\boldsymbol{\pi}_v$ 满足等式(8):

$$\boldsymbol{\pi}_v=(1-\alpha)\times\boldsymbol{P}^{\boldsymbol{T}}\times\boldsymbol{\pi}_v+\alpha\times\boldsymbol{q}_v \tag{8}$$

其中,重启向量 $\boldsymbol{q}_v$ 中,v 对应分量为 1,其他分量全为 0.

将类簇 C 中所有结点对应的公式(8)进行累加,并两边同除以|C|,得到等式(9):

$$\frac{1}{|C|}\times\sum_{v\in C}\boldsymbol{\pi}_v=\frac{1}{|C|}\times(1-\alpha)\times\boldsymbol{P}^{\boldsymbol{T}}\times\sum_{v\in C}\boldsymbol{\pi}_v+\frac{1}{|C|}\times\alpha\times\sum_{v\in C}\boldsymbol{q}_v \tag{9}$$

而以类簇 C 为初始结点组的平稳概率分布 $\boldsymbol{\pi}_C$ 满足等式(10):

$$\boldsymbol{\pi}_C=(1-\alpha)\times\boldsymbol{P}^{\boldsymbol{T}}\times\boldsymbol{\pi}_C+\alpha\times\boldsymbol{q}_C \tag{10}$$

易知 $\boldsymbol{q}_C=\frac{1}{|C|}\times\sum_{v\in C}\boldsymbol{q}_v$,且平稳概率分布存在且唯一.对比等式(9)和等式(10),可得公式(7). □

按照定理 3 的计算方式,直接利用以单结点为初始组的平稳概率分布来计算以多结点类簇为初始组的平稳概率分布,避免了聚类过程中像第 2.1.1 节中那样每次迭代地计算,节省了开销.为此,在算法 1 的开始部分(第 1 行之前),应该先计算所有以单结点为初始组的平稳概率分布.

4 实验评价

4.1 准备工作

- 数据集

使用 3 个真实数据集和一组生成数据集来进行实验评价.真实数据集分别是引文数据集 Cora[16,20]、电商数据集 Amazon-GoogleProducts[3,20]和引文数据集 Citeseer.Cora 数据集包括 1 295 条引文记录,引用了 112 篇论文,其属性有标题、作者、作者单位、会议名称、地点、出版商、年份、页码、册和编辑.Amazon-GoogleProducts(简称 A-G)数据集包括 4 589 条商品记录,有 1 300 个匹配对,其属性有商品名、商品描述、制造商和定价.Citeseer 数据集包括 100 000 条引文记录,引用了 9 387 篇论文,其属性有标题、作者、作者单位、会议名称和年份.相比于 Cora 和 A-G 数据集,Citeseer 数据集是一个较大规模的数据集,本文将通过在此数据集上实验,证明 ERC 算法可以有效地处理大规模数据,具有良好可扩展性.生成数据集是基于真实的人口统计数据(美国)生成的,包括社保号、姓名、地址、城市名、州名和邮编.生成数据的工具是 UIS 数据生成器[4],它可以控制生成数据集的规模、字段错误类型和数据分布;生成重复记录的字段错误类型有字符插入、删除、替换和倒置等.本实验中用到的生成数据集的数据分布类型有均匀分布、zipf 分布和泊松分布.

- 评价指标.本文采用 F 指数评价实体识别结果的精确性.F 指数是准确率和召回率的调和平均数,准确率记作 P,召回率记作 R,那么, $F=\frac{2\times P\times R}{P+R}$;
- 实验环境.处理器:Intel(R) Core(TM) i7-2600,主频 3.4GHz,8 核;内存:8G;操作系统:Microsoft Windows 7 Ultimate,64 位;编程语言:Java;
- 实验说明.本实验测试目标是聚类算法,将数据对象相似度计算当作黑盒来用,因此,实验默认已经得到数据对象的相似对集合.每次进行对比时,不同类聚算法的相似对集合是相同的;
- 缩写解释.本实验将用到 4 个版本的 ERC 算法,分别记作 ERC-0,ERC-1,ERC-2 和 ERC-3,具体构成见表 1.其中,ERC-0 是完整的 ERC 算法,ERC-1,ERC-2 和 ERC-3 用于自身对比实验.

Table 1 Different versions of ERC algorithms

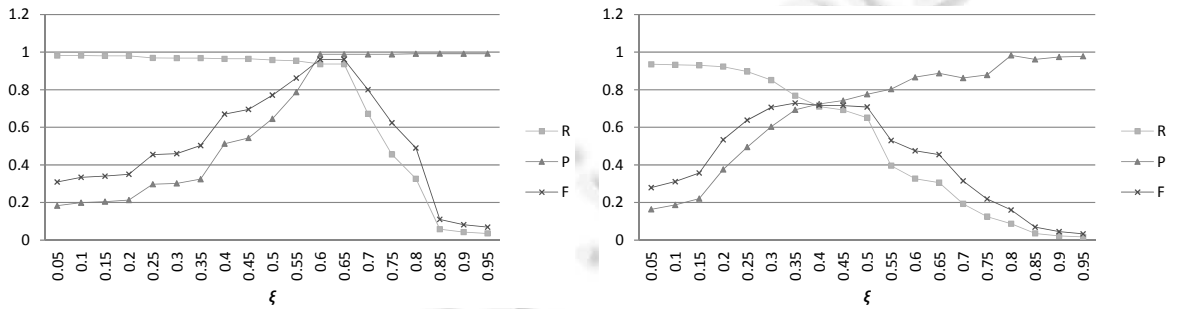
表 1 不同版本的 ERC 算法

算法	簇点相似度	是否对数据对象排序	是否使用计算优化
ERC-0	双向的簇点相似度	是	是
ERC-1	双向的簇点相似度	否	是
ERC-2	基本的簇点相似度	是	是
ERC-3	双向的簇点相似度	是	否

4.2 实验结果与分析

4.2.1 参数测试

在两个真实集上测试参数 ξ 对 ERC-0 算法效果的影响.图 4(a)中,Cora 数据集上,在 ξ 从 0.05 增长到 0.95 的过程中,准确率(P)从 0.183 开始,逐渐提高至 0.992;召回率(R)从 0.982 开始,逐渐降低到 0.036;精确性(F)先逐渐提高至最大值 0.961,此时 ξ=0.6,然后不断降低.图 4(b)中,A-G 数据集上,在 ξ 从 0.05 增长到 0.95 的过程中,准确率(P)从 0.164 开始,逐渐提高至 0.987;召回率(R)从 0.935 开始,逐渐降低到 0.017;精确性(F)先逐渐提高至最大值 0.729,此时 ξ=0.35,然后不断降低.ERC-0 算法在两个数据集上的变化趋势是一致的,随着 ξ 变大,即簇点相似度的约束条件变严格,准确率提高,召回率降低,精确性是前两者的调和平均数,因此经历先提高后降低的过程,并在极值点处取得最大值.本实验中,所有的 ERC 算法中,ξ 值都是通过此方法测定的.



(a) Cora (b) A-G

Fig.4 Tests of the parameter ξ's influence on clustering on real datasets

图 4 在真实集上测试参数 ξ 对聚类的影响

实际应用中,聚类算法的阈值 ξ 可以通过抽样测定.给定一个较大的数据集 D,通过抽样获得分布特征相同的较小样本集 d,并对 d 进行上述参数测试得到最优阈值 ξ',将 ξ'应用在原数据集 D 上.以 Citeseer 数据集为例,随机地从 Citeseer 数据集中抽取 3 000 条数据,用本文提出的 ERC 算法对样本数据进行处理,调节阈值来获得最优的结果.本文对 Citeseer 数据集进行了 3 次抽样,分别记作抽样 1、抽样 2 和抽样 3;Citeseer 数据集本身记作 Citeseer 整体集.利用 ERC 算法分别对这 4 个数据集进行阈值测试,结果如图 5 所示.Citeseer 整体集上,ERC 算

法在 ξ 取[0.4,0.5]时获得了最高的 F 值;而在抽样 1~抽样 3 这 3 个数据集上,ERC 算法依次在 0.4 左右、[0.4,0.5]和[0.45,0.5]等范围内取得最高 F 值.整体来讲,3 个样本数据集的最佳阈值范围与 Citeseer 整体集的最佳阈值范围要么有一定重合,要么非常接近.因此,通过抽样测定的方法可估计出 ERC 算法在某个数据集上的最佳阈值.

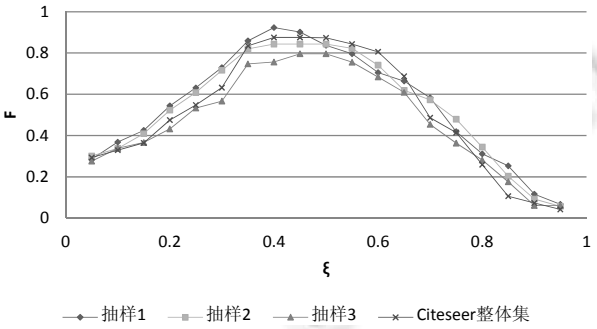


Fig.5 Comparisons of threshold ξ via sampling and via the whole dataset with Citeseer dataset

图 5 在 Citeseer 数据集上采用抽样方法测定的阈值 ξ 与在整体集上测定的阈值 ξ 对比

4.2.2 与已有工作对比

传统的、非监督的实体识别中,匹配决定采用基于阈值的的方法[2,4],实际是简单的划分式聚类,记作 Part.. Merge-Center 聚类算法通过排序选出中心点来聚类,并将足够近的类簇合并[4,24],记作 MC.Markov Clustering 算法通过模拟图上的马尔可夫随机流来进行聚类[25,26],记作 MCL.MinCut 聚类算法在相似图上发现边的最小切割来实现聚类[27].Articulation Point Clustering 算法在相似图上发现关节点和重连通分量来进行聚类[28,29],记作 ArtPt.

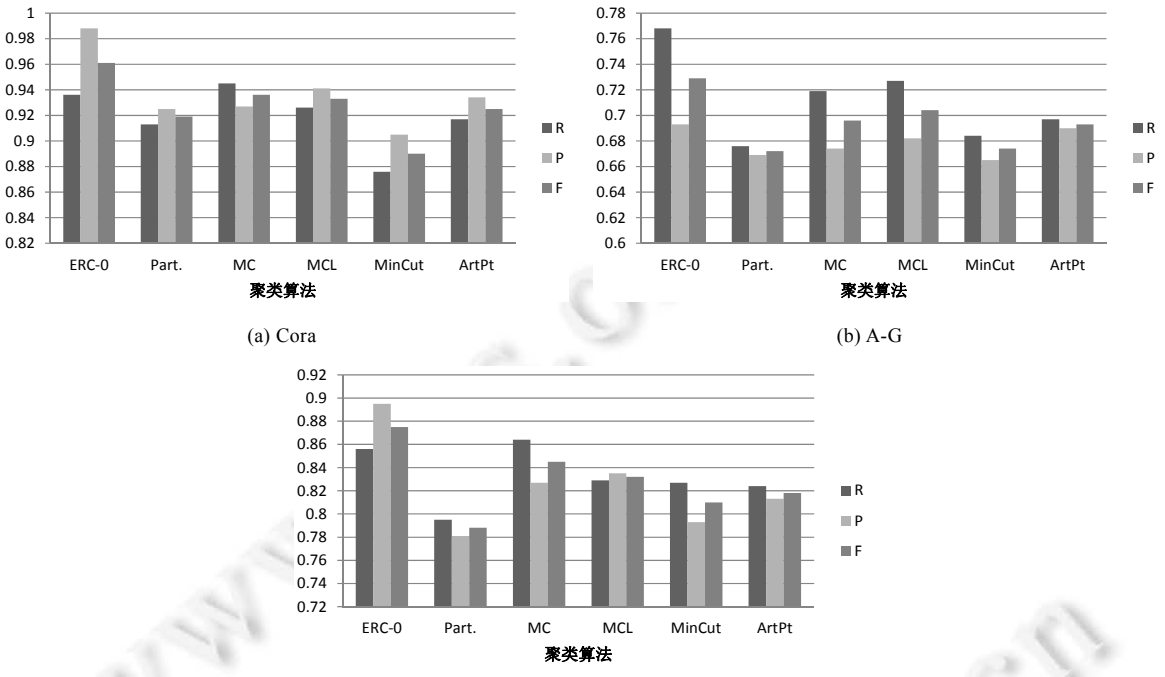
Hassanzadeh 等人[4]首次在匹配决定中使用 MC,MCL,MinCut 和 ArtPt 聚类算法,并与 Part.算法进行了比较.

将上述 5 种算法与本文提出的 ERC-0 算法基于 3 个真实数据集进行对比.图 6(a)中:在 Cora 数据集上,整体来讲(即 F 值),算法表现降序排列是 ERC-0>MC>MCL>ArtPt>Part.>MinCut,ERC-0 算法要明显优于其他 5 个聚类算法,以基于阈值的方法 Part.为基准,ERC-0 提高了 4.6%,而其他聚类算法中表现最好的 MC 提高了 1.8%,前者是后者的 2.6 倍;图 6(b)中:在 A-G 数据集上,整体来讲,算法表现降序排序是 ERC-0>MCL>MC>ArtPt>MinCut>Part.,ERC-0 算法在此数据集上表现依然明显优于其他 5 个聚类算法,以 Part.为基准,ERC-0 提高了 8.5%,而其他聚类算法中表现最好的 MCL 提高了 4.8%,前者是后者的 1.8 倍;图 6(c)中:在 Citeseer 数据集上,整体来讲,算法表现降序排序是 ERC-0>MC>MCL>ArtPt>MinCut>Part.,ERC-0 算法在此数据集上表现也明显优于其他 5 个聚类算法,以 Part.为基准,ERC-0 提高了 11%,而其他聚类算法中表现最好的 MC 提高了 7.2%,前者是后者的 1.5 倍.所有 6 个算法在 3 个数据集上的表现并不完全相同,但 ERC-0 算法在 3 个数据集上的表现都要明显优于其他 5 种方法,证明 ERC-0 算法是匹配决定的一个有效的解决方案.另外,在 3 个数据集上,ERC-0,MC,MCL,MinCut 和 ArtPt 这 5 种聚类算法的表现整体要优于基于阈值的 Part.算法,只有在 Cora 数据集上,MinCut 表现不如 Part..

ERC-0 算法与其他 5 种已有算法在 3 个真实数据集上的时间开销的对比见表 2.从一方面看,将算法按时间开销从大到小排列,依次为 MinCut,ERC-0,ArtPt,MCL,MC,Part.;ERC-0 的时间开销除了比 MinCut 的时间开销小很多外,比其他 4 种聚类算法的时间开销都要大很多,至少为 2 倍以上(ArtPt),最大达到 20 倍以上(Part.).这是因为 ERC-0 算法在计算基本的单例簇点相似度的时候要进行迭代计算,时间开销比较大.已有算法中,时间开销最小的是 Part.算法,该算法只需要将记录对的相似度与给定阈值比较就可以得到匹配结果;时间开销最大的是 ArtPt 算法,该算法要对相似图进行最小切割,具有非常高的时间复杂度.

从另一方面看,比较 ERC-0 算法在 Cora(1 295 条记录)和 Citeseer(100 000 条记录)两个数据集上的时间开销,分别为 0.837s 和 68.63s,并没有呈幂指级的增长,而是近似于线性的增长.这是因为这两个数据集在分块后得

到的块平均大小是差不多的,而迭代计算只发生在块内的记录之间,不同块之间的记录之间不存在迭代计算.与 Cora 数据集相比,Citeseer 数据集拥有更多的块.由此可见,ERC-0 算法具有良好的可扩展性,可应用于大规模数据.



(a) Cora (b) A-G

(c) Citeseer

Fig.6 Comparisons between ERC-0 algorithm and existing works on real datasets

图 6 ERC-0 算法与已有工作在真实数据集上的对比

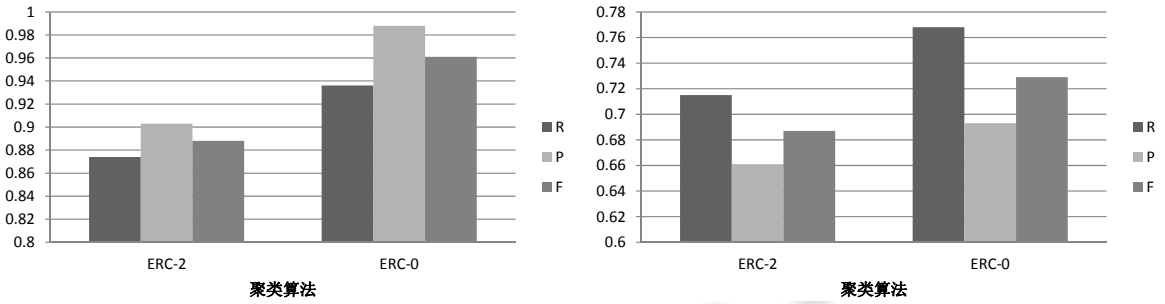
Table 2 Time cost comparisons of all clustering algorithms on three real datasets (s)

表 2 各个聚类算法在 3 个真实数据集上时间开销对比 (s)

	ERC-0	Part.	MC	MCL	MinCut	ArtPt
Cora	0.837	0.034	0.089	0.236	36.5	0.361
A-G	0.615	0.026	0.063	0.185	28.3	0.276
Citeseer	68.63	2.85	6.81	19.63	5361	31.74

4.2.3 簇点相似度测试

本节在两个真实数据集上,比较基本的簇点相似度和双向的簇点相似度在聚类过程中的表现.基于基本簇点相似度的 ERC 算法记做 ERC-2.图 7(a)中:在 Cora 数据集上,ERC-0 的 F 值比 ERC-2 的高 8.2%;图 7(b)中:在 A-G 数据集上,ERC-0 的 F 值比 ERC-2 的高 6.1%.可见:在 ERC 算法聚类过程中,双向的簇点相似度比基本的簇点相似度能更好地衡量类簇与结点之间的相似性.分析其原因,基本的簇点相似度考虑从类簇出发到达候选结点的随机游走的平稳概率,这是一个单向的量;双向的簇点相似度在基本的簇点相似度基础上,考虑了类簇中候选结点的 k 近邻结点所占的比例,这是一个双向的量,是一种相互的相似度.在实体识别中,相似度的相互性(即,非严格的对称性)非常重要[12].由此,在 ERC 算法聚类过程中,双向的簇点相似度比基本的簇点相似度更有效.



(a) Cora (b) A-G

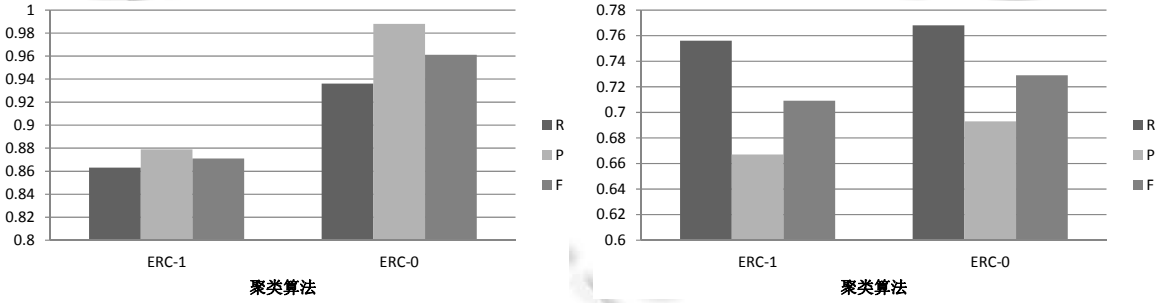
Fig.7 Comparisons of the two proposed cluster-vertex similarities on real datasets

图 7 两种簇点相似度在真实集上的对比

4.2.4 数据对象排序测试

本节测试数据对象排序在 ERC 算法中的作用.首先在两个真实数据集上进行对比实验,然后在不同分布类型的生成数据集上进行对比实验.

在两个真实数据集上,测试数据对象排序在 ERC 算法中的作用,将没有数据对象排序的 ERC 算法记作 ERC-1.图 8(a)中:在 Cora 数据集上,ERC-0 的 F 值比 ERC-1 高 10.3%;图 8(b)中:在 A-G 数据集上,ERC-0 的 F 值比 ERC-1 高 2.8%.在两个真实数据集上,ERC-0 的聚类效果都要比 ERC-1 好.可见,ERC 算法中数据对象排序有利于提高聚类结果的精确性.分析原因,两个真实数据集都是非均匀分布的,根据定理 3,将数据对象按照信用度降序排列的聚类结果比随机排列的更好.另外,比较不同数据集上的ΔF发现,在 Cora 数据集上的ΔF要比 A-G 数据集上的ΔF要大很多.从两者的数据分布来看:Cora 中真实类簇的大小差别很大,A-G 中真实类簇的规模以 2~3 为主.越不均匀分布的数据集上,数据对象排序带来的聚类顺序的变化越大,对聚类结果的影响也越大.后面将在生成数据集上专门针对数据分布的不均匀程度进行实验.



(a) Cora (b) A-G

Fig.8 Tests of data objects ordering's influence on clustering on real datasets

图 8 在真实集上测试数据对象排序对聚类的影响

在不同数据分布的生成数据集上,测试数据对象排序在聚类中的作用.用 UIS 数据生成器分别生成均匀分布的数据集、zipf 分布的数据集和泊松分布(λ=3)的数据集,规模都是 3 000 条.均匀分布记作 unf,泊松分布记作 psn.图 9 中:在均匀分布的数据集上,ERC-0 的 F 值与 ERC-1 几乎相当;在 zipf 分布的数据集上,ERC-0 的 F 值比 ERC-1 高 22.9%;在泊松分布的数据集上,ERC-0 的 F 值比 ERC-1 高 5.4%.首先,均匀分布的数据集上,ERC-1 和 ERC-0 的表现几乎相同,数据对象排序在均匀分布的数据上对聚类结果的影响几乎没有;其次,在 zipf 分布的和泊松分布的数据集上,ERC-0 的效果明显好于 ERC-1,数据对象排序在非均匀的数据分布上能够提高聚类结果的精确性;最后,在 zipf 分布数据集上的ΔF要远大于在泊松分布数据集上的ΔF.分析原因,zipf 分布的数据集上,真实类簇大小的不均匀程度远大于泊松分布数据集上的不均匀程度.

为了进一步测试数据分布的不均匀程度对数据对象排序在 ERC 算法中的影响,生成一组泊松分布的数据集,调节λ来控制分布变化,λ越大,分布悬殊越大,λ分别取 1,2,3,4 和 5,数据规模都是 3 000 条.图 10 中:整体来讲,ERC-0 的 F 值高于 ERC-1;具体来讲,随着泊松分布的参数λ增长(从 1~5),ERC-0 和 ERC-1 的ΔF 在不断增大,F 值提高率从 0.9%增长至 12.2%.综上所述,在越不均匀分布的数据集上,数据对象排序对 ERC 算法的聚类结果的精确性提高越大.

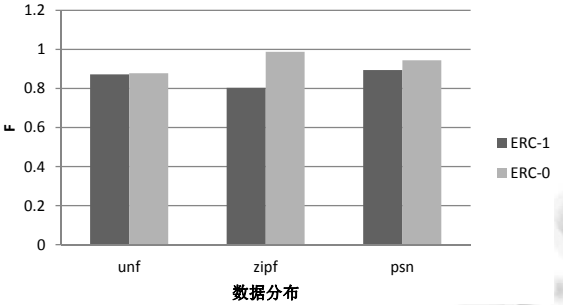


Fig.9 Data objects ordering's influences on clustering with different data distributions

图 9 不同数据分布下数据对象排序对聚类的影响

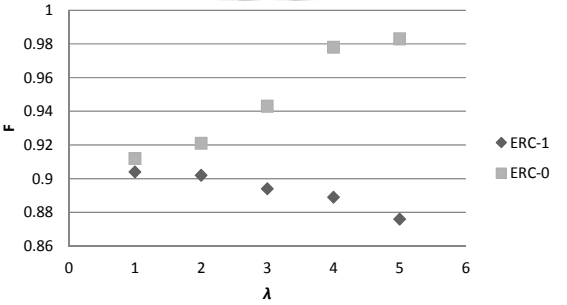


Fig.10 Data objects ordering's influence on clustering with Poisson distributions of different biases

图 10 数据对象排序在不均匀程度不同的泊松分布下对聚类结果的影响

4.2.5 计算优化测试

本节测试计算优化对 ERC 算法的作用.表 1 中,ERC-3 算法与 ERC-0 算法的不同之处在于,ERC-3 算法没有计算优化,聚类过程中需要迭代地计算簇点相似度.首先比较两个算法的计算结果的精确性,ERC-0 算法与 ERC-3 算法的计算结果是相同的,图 11 是两种算法在 3 个真实数据集上的结果对比.计算优化对 ERC 算法计算结果的精确性没有影响.

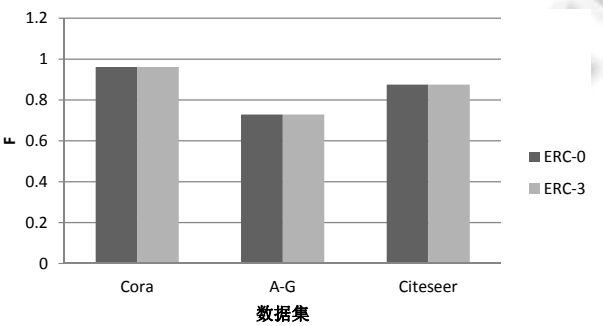


Fig.11 Tests of computation optimization's influence on accuracy with real datasets

图 11 在真实数据集上测试计算优化对精确性的影响

然后比较两者的时间开销,分别在 3 个真实的数据集上进行实验,得到结果见表 3.根据表 3,ERC-0 算法比 ERC-3 算法在 Cora,A-G 和 Citeseer 这 3 个数据集上分别少开销 38.5%,37.4%和 33.3%的时间,因此,计算优化可以为 ERC 算法带来较大的时间开销节省.基本的单例簇点相似度在对记录进行排序的时候已经计算得到;利用计算优化,ERC-0 算法在聚类过程只需要对基本的单例簇点相似度进行线性组合,即可计算出类簇与结点的相似度,因而节省了较大的开销.

Table 3 Tests of computation optimization's influence on time cost with real datasets (s)

表 3 在真实数据集上测试计算优化对时间开销的影响 (s)

	Cora	A-G	Citeseer
ERC-0	0.837	0.615	68.63
ERC-3	1.362	0.983	102.95

5 相关工作

实体识别是数据质量的一个重要方面,又称为实体解析、实体匹配、记录匹配、实体辨析、合并与清洗等[2−14].实体识别主要包括对象相似度计算和匹配决定两个必要组成部分和一个可选组成部分:分块.已有工作针对对象相似度计算提出很多相似度算法,以适应不同类型的数据对象[8,9,12−14],如文本的相似度算法、基于实体关系的相似度算法等.大数据实体识别中,分块技术变得不可或缺.分块技术通过减小搜索空间来降低开销[16−20],提高大数据实体识别效率.对象相似度计算和分块技术在引言和第 1.1 节中描述,此处不再赘述.

实体识别中的匹配决定方法按照是否需要训练过程可分为监督类方法和非监督类方法:监督类方法要求用户提供高质量的标注数据来训练分类器,然后利用分类算法来执行匹配决定[2,3,5](见第 1.1 节);监督类方法严重依赖领域知识,造成其应用的局限性.传统的非监督类方法通过测定相似度阈值来判定候选对是否匹配.Hassanzadeh 等人将已有的聚类算法应用在匹配决定中,并通过对比实验证明,基于聚类的方法比基于阈值的方法更有效[4].

Center 算法[24]首先将相似对降序排列依次加入队列;遍历队列,从第 1 个扫描到的候选对中选一个结点 v_i 作为下一个类簇的中心;将后续所有与 v_i 相似的结点加入到最新生成的类簇,并将包含这些结点的候选对从队列中删除;不断循环,直到生成聚类结果.即,所有的类簇都由一个中心和与它相近的结点组成.Center 算法在网络文档检索上非常高效.文献[4]在 Center 算法基础上提出 Merge-Center 算法,它允许足够相似的类簇合并.在实体识别的匹配决定中,Merge-Center 算法比 Center 算法更高效[4].Markov Clustering 算法[25]通过模拟图上的马尔可夫随机流来进行聚类,它的基本思想是:图上一块关联紧密的区域会形成一个类簇,类簇内部的流的总量会很强;相反,两个类簇之间的的关联较弱,类簇间的流的总量会较弱.Markov Clustering 算法在图上进行随机游走,加强原本就强的流(即簇内),减弱原本就弱的流(即簇间),不断迭代,直到类簇结构形成.Markov Clustering 算法被应用在生物信息领域中,并能快速地得到高质量的聚类结果[26].MinCut 聚类算法[27]在图上发现边的最小切割来实现聚类,该算法基本思想是:发现类簇间的最小切割,从而使得簇内的边的权重和最大化,这样得到的聚类结果将是簇内紧密耦合,簇间松散关联.MinCut 算法被应用在引文数据和网络数据聚类中[28].Articulation Point Clustering 算法在图上发现关节点和重连通分量来进行聚类[29],每个重连通分量就是一个类簇.该算法被应用于发现博客圈中热点话题[32].Hassanzadeh 等人首次将上述聚类算法应用于匹配决定中,并进行了实验对比[4].通过本文的实验对比可知:在 Cora 和 A-G 两个真实的数据集上,本文提出的 ERC 算法的识别效果,要比 Merge-Center,Markov Clustering,MinCut 和 Articulation Point Clustering 等算法的识别效果更好.

大数据的一个特点是产生和更新速度快,Gruenheid 等人在已有算法的基础上提出了增量的实体识别算法,能增量、快速地处理逐步更新的数据[33].为满足实时的应用需要,即,在短时间内识别大部分的数据对象,提出了 Pay-as-you-go 实体识别算法[6,7].基于时间特征的实体识别算法通过分析数据对象随时间的演化信息来完成识别任务[8,9].

随着众包(crowdsourcing)的流行,一些研究工作借助大众的力量提出了基于众包的实体识别算法[10,11].

6 结束语

实体识别对于数据集成和数据挖掘都必不可少.本文针对非监督的实体识别中匹配决定问题,提出了一个基于随机游走模型的聚类算法——ERC 算法.该算法利用图上的随机游走,通过挖掘图结构来计算聚类过程中类簇和结点的相似度;为了优化聚类顺序、提高识别结果的精确性,该算法根据数据对象的信用度来进行降序排列.通过在两个真实数据集上和若干生成数据集上的实验对比和分析,验证了 ERC 算法的有效性以及其组成

部分的作用.在未来的工作中,作者将致力于提出更加理论化的阈值确定方法.另外,如何使本文提出的算法能够更高效、准确地处理增量实体识别中匹配决定,也将是进一步的研究工作.

References:

[1] Mayer-Schönberger V, Cukier K. Big Data: A Revolution That Will Transform How We Live, Work, and Think. Boston: Houghton Mifflin Harcourt, 2013.

[2] Elmagarmid AK, Ipeirotis PG, Verykios VS. Duplicate record detection: A survey. IEEE Trans. on Knowledge and Data Engineering, 2007,19(1):1−16. [doi: 10.1109/TKDE.2007.250581]

[3] Köpcke H, Thor A, Rahm E. Evaluation of entity resolution approaches on real-world match problems. Proc. of the VLDB Endowment, 2010,3(1-2):484−493. [doi: 10.14778/1920841.1920904]

[4] Hassanzadeh O, Chiang F, Lee HC, Miller RJ. Framework for evaluating clustering algorithms in duplicate detection. Proc. of the VLDB Endowment, 2009,2(1):1282−1293. [doi: 10.14778/1687627.1687771]

[5] Guo ZM, Zhou AY. Data quality and data cleaning: A survey. Ruan Jian Xue Bao/Journal of Software, 2002,13(11):2076−2028 (in Chinese with English abstract).

[6] Altowim Y, Kalashnikov DV, Mehrotra S. Progressive approach to relational entity resolution. Proc. of the VLDB Endowment, 2014,7(11):999−1010. [doi: 10.14778/2732967.2732975]

[7] Papenbrock T, Heise A, Naumann F. Progressive duplicate detection. IEEE Trans. on Knowledge and Data Engineering, 2015,27(5): 1316−1329. [doi: 10.1109/TKDE.2014.2359666]

[8] Chiang YH, Doan AH, Naughton JF. Tracking entities in the dynamic world: A fast algorithm for matching temporal records. Proc. of the VLDB Endowment, 2014,7(6):469−480. [doi: 10.14778/2732279.2732284]

[9] Li F, Lee ML, Hsu W, Tan WC. Linking temporal records for profiling entities. In: Proc. of the 2015 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2015. 593−605. [doi: 10.1145/2723372.2737789]

[10] Vesdapunt N, Bellare K, Dalvi N. Crowdsourcing algorithms for entity resolution. Proc. of the VLDB Endowment, 2014,7(12): 1071−1082. [doi: 10.14778/2732977.2732982]

[11] Wang S, Xiao X, Lee CH. Crowd-Based deduplication: An adaptive approach. In: Proc. of the 2015 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2015. 1263−1277. [doi: 10.1145/2723372.2723739]

[12] Benjelloun O, Garcia-Molina H, Menestrina D, Su Q, Whang SE, Widom J. Swoosh: A generic approach to entity resolution. The Int'l Journal on Very Large Data Bases, 2009,18(1):255−276. [doi: 10.1007/s00778-008-0098-x]

[13] Bhattacharya I, Getoor L. Collective entity resolution in relational data. ACM Trans. on Knowledge Discovery from Data, 2007, 1(1):5. [doi: 10.1145/1217299.1217304]

[14] Sun CC, Shen DR, Kou Y, Nie TZ, Yu G. A related data oriented joint entity resolution approach. Chinese Journal of Computers, 2015,38(9):1739−1754 (in Chinese with English abstract).

[15] Cohen W, Ravikumar P, Fienberg S. A comparison of string metrics for matching names and records. In: Getoor L, Senator TE, Domingos PM, Faloutsos C, eds. Proc. of the ACM KDD Workshop on Data Cleaning and Object Consolidation. New York: ACM Press, 2003. 73−78.

[16] Christen P. A survey of indexing techniques for scalable record linkage and deduplication. IEEE Trans. on Knowledge and Data Engineering, 2012,24(9):1537−1555. [doi: 10.1109/TKDE.2011.127]

[17] Papadakis G, Papastefanatos G, Koutrika G. Supervised meta-blocking. Proc. of the VLDB Endowment, 2014,7(14):1929−1940. [doi: 10.14778/2733085.2733098]

[18] Fisher J, Christen P, Wang Q, Wang Q, Rahm E. A clustering-based framework to control block sizes for entity resolution. In: Proc. of the 21th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2015. 279−288. [doi: 10.1145/2783258.2783396]

[19] Karakasidis A, Koloniari G, Verykios VS. Scalable blocking for privacy preserving record linkage. In: Proc. of the 21th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2015. 527−536. [doi: 10.1145/2783258.2783290]

[20] Kenig B, Gal A. Efficient entity resolution with mfiblocks. Proc. of the VLDB Endowment, 2009,4(1-2):484−493.

[21] Arasu A, Götz M, Kaushik R. On active learning of record matching packages. In: Elmagarmid AK, Agrawal D, eds. Proc. of the 2010 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2010. 783−794. [doi: 10.1145/1807167.1807252]

[22] Xu R, Wunsch I. Survey of clustering algorithms. IEEE Trans. on Neural Networks, 2005,16(3):645−678. [doi: 10.1109/TNN.2005.845141]

[23] Sun JG, Liu J, Zhao LY. Clustering algorithms research. Ruan Jian Xue Bao/Journal of Software, 2008,19(1):48−61 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/48.htm

[24] Haveliwala T, Gionis A, Indyk P. Scalable techniques for clustering the Web. In: Proc. of the 2000 Int'l Workshop on the Web and Databases. New York: ACM Press, 2000. 129−134.

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

[26] Brohee S, Van Helden J. Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinformatics, 2006, 7(1):488. [doi: 10.1186/1471-2105-7-488]

[27] Flake GW, Tarjan RE, Tsioutsiouliklis K. Graph clustering and minimum cut trees. Internet Mathematics, 2004,1(4):385−408. [doi: 10.1080/15427951.2004.10129093]

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

[29] Bansal N, Chiang F, Koudas N, Tompa FW. Seeking stable clusters in the blogosphere. Proc. of the VLDB Endowment, 2007: 806−817.

[30] Motwani R, Raghavan P. Randomized Algorithms. Cambridge: Cambridge University Press, 1995.

[31] Tong H, Faloutsos C, Pan JY. Fast random walk with restart and its applications. In: Proc. of the 6th IEEE Int'l Conf. on Data Mining. Piscataway: IEEE, 2006. 613−622. [doi: 10.1109/ICDM.2006.70]

[32] Clauset A, Shalizi CR, Newman ME. Powerlaw distributions in empirical data. SIAM Review, 2009,51(4):661−703. [doi: 10.1137/070710111]

[33] Gruenheid A, Dong XL, Srivastava D. Incremental record linkage. Proc. of the VLDB Endowment, 2014,7(9):697−708. [doi: 10.14778/2732939.2732943]

附中文参考文献:

[5] 郭志懋,周傲英.数据质量和数据清洗研究综述.软件学报,2002,13(11):2076−2028.

[14] 孙琛琛,申德荣,寇月,聂铁铮,于戈.面向关联数据的联合式实体识别方法.计算机学报,2015,38(9):1739−1754.

[23] 孙吉贵,刘杰,赵连宇.聚类算法研究.软件学报,2008,19(1):48−61. http://www.jos.org.cn/1000-9825/19/48.htm

孙琛琛(1987－),男,山西平遥人,博士生,CCF 学生会员,主要研究领域为实体识别.

聂铁铮(1980－),男,博士,副教授,CCF 会员,主要研究领域为数据质量,数据集成.

申德荣(1964－),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为分布式数据管理,数据集成.

于戈(1962－),男,博士,教授,博士生导师,CCF 会士,主要研究领域为数据库,大数据管理.

寇月(1980－),女,博士,副教授,CCF 会员,主要研究领域为实体搜索,数据挖掘.