DOI:10.7544/issn1000-1239.2016.20150614
53(12):2858-2866,2016

MTruths:Web 信息多真值发现方法

马如霞[1,2] 孟小峰[1] 王 璐[1] 史英杰[3]
[1](中国人民大学信息学院 北京 100872)
[2](首都师范大学教育技术系 北京 100048)
[3](北京服装学院信息工程学院 北京 100029)
(maruxia@126.com)

MTruths:An Approach of Multiple Truths Finding from Web Information

Ma Ruxia[1,2], Meng Xiaofeng[1], Wang Lu[1], and Shi Yingjie[3]
[1](*School of Information*, *Renmin University of China*, *Beijing* 100872)
[2](*Department of Education Technology*, *Capital Normal University*, *Beijing* 100048)
[3](*School of Information Engineering*, *Beijing Institute of Fashion Technology*, *Beijing* 100029)

Abstract Web has been a massive information repository on which information is scattered in different data sources. It is common that different data sources provide conflicting information for the same entity. It is called the truth finding problem that how to find the truths from conflicting information. According to the number of attribute values, object attributes can be divided into two categories: single-valued attributes and multiple-valued attributes. Most of existing truth finding work is designed for truth finding on single-valued attributes. In this paper, a method called MTruths is proposed to resolve truth finding problem for multiple-valued attributes. We model the problem using an optimization problem. The objective is to maximize the total weight similarity between the truths and observations provided by data sources. In truth finding process, two methods are proposed to find the optimal solution: an enumeration algorithm and a greedy algorithm. Experiments on two real data sets show that the correctness of our approache and the efficiency of the greedy algorithm outperform the existing state-of-the-art techniques.

Key words truth finding; data conflicting; single-valued attributes; multi-valued attributes; quality of data sources

摘 要 Web 已成为一个浩瀚的信息海洋,其信息分散在不同的数据源中.不同数据源常常为同一对象实体提供冲突的属性值.如何从这些冲突属性值中找到真值被称为真值发现问题.根据属性值数量可将对象属性分为单值属性和多值属性,现有的多数真值发现算法对单值属性的真值发现比较有效.针对多值属性的真值发现问题,提出了一个多真值发现方法 MTruths,该方法将多真值发现问题转化为一个

收稿日期:2015-06-30;**修回日期**:2015-10-13
基金项目:国家自然科学基金项目(61379050,91224008,61502279);国家"八六三"高技术研究发展计划基金项目(2013AA013204);高等学校博士学科点专项科研基金项目(20130004130001);中国人民大学科学研究基金项目(11XNL010)
This work was supported by the National Natural Science Foundation of China (61379050, 91224008, 61502279), the National High Technology Research and Development Program of China (863 Program) (2013AA013204), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20130004130001), and the Research Funds of Renmin University of China (11XNL010).
通信作者:孟小峰(xfmeng@ruc.edu.cn)

最优化问题,其目标是:各对象的真值与各数据源提供的观察值之间的相似性加权和达到最大.对象真值求解过程中,提出 2 种方法求真值列表的最优解:基于枚举的方法和贪心算法.与已有方法不同的是 MTruths 可以直接得到对象的多个真值.最后,通过图书和电影 2 个真实数据集上的实验表明,MTruths 的 2 种实现方法的准确性以及贪心算法的效率优于现有真值发现方法.

关键词 真值发现;数据冲突;单值属性;多值属性;数据源质量

中图法分类号 TP311

互联网信息量正以惊人的速度急剧增长,俨然成为一个巨大的信息库.Web 已经渗透到人们日常生产、生活的方方面面,逐渐成为人们获取信息的重要来源.人们在享受来自 Web 丰富信息的同时,也受到信息质量问题的困扰,大量错误、过时、不完整、虚假信息充斥于网络.其中,信息冲突问题尤为突出,不同数据源为相同对象同一属性提供冲突的值.例如,不同图书网站为同一本书提供了不同的作者信息,如表 1 所示;各航空网站为同一航班提供不同的登机时间、在线零售商为同一商品提供了不一致的产品规格说明等.这些冲突信息可能由于输入错误、信息过期、语义理解不一致、抽取程序错误等各种原因造成,给用户带来误导甚至造成巨大损失.如何从不同数据源提供的冲突信息中找到正确信息是提高 Web 信息质量亟待解决的问题,该问题也被称为真值发现问题[1].

Table 1 Conflicting Information of Book Authors[1]

表 1 Rapid Contextual Design 作者信息[1]

Data Sources	Authors
Powell's Books	Karen Holtzblatt
Barnes & Noble	Karen Holtzblatt, Jessamyn Wendell, Shelley Wood
A1 Books	Karen Holtzblatt, Jessamyn Burns Wendell, Shelley Wood
Cornwall Books	Karen Holtzblatt, Jessamyn Burns Wendell, Wood
Mellon's Books	Jessamyn Wendell
Blackwell Online	Jessamyn Wendell, Karen Holtzblatt, Shelley Wood

真值发现问题已有一系列研究工作.其最简单直观的方法是采用投票的方法,当获得的票数占总票数比例达到某个阈值时认为该属性值为真.由于数据源质量存在差异,因此在投票时需要考虑数据源的质量因素,可将数据源质量作为先验知识,采用加权投票的方法得到真值.但当大多数数据源都发生错误时,投票方法很难得到正确的结果.并且在实际中,往往没有数据源质量的先验知识,为此文献[1-5]采用无监督的方法迭代地计算各属性值可信性以及数据源质量.为了简化问题很多方法提出"对象属性真值唯一性"假设,并且最终选择可信性分值最大或者为真概率最大的属性值作为真值,此类方法适用于单值属性的真值求解问题.然而,实际生活中的多值属性比比皆是,如一本图书可以有多个作者、一部电影可以有多个主演、一个人可以有多个电话号码等.针对这些多值属性,不但要确保所找到真值的正确性,而且要尽可能找到所有真值,我们称该问题为多真值发现问题.与投票方法类似,解决此问题最直观的方法是设置阈值:选择 Top N 个可信性分值的属性值作为真值,或者选择为真概率大于 K 的属性值作为真值.但是,阈值 K 的选择是一个挑战性的问题,其直接影响算法的查准率和查全率,例如:属性值为真概率的阈值选择越大查准率越高,但查全率随之降低.

本文目标是解决多值属性的真值发现问题,主要贡献有 3 点:

1) 将多真值发现问题转化为一个最优化问题.根据 2 个观察:对象的真值情况应该尽可能与各数据源提供的观察值接近;数据源的质量评估至关重要,其影响了真值发现的准确率,数据源的质量越高则其提供的对象属性值列表与真值列表越相似.该优化问题的目标函数是:各对象的真值取值与数据源提供的该对象观察值之间相似度权重之和达到最大,其中权重是数据源质量.通过该方法直接得到对象的真值列表,避免通过阈值的设置选择对象真值.

2) 真值计算过程中,我们首先提出了一个枚举的方法求最优解,但该方法的时间复杂度为对象可能值集大小的指数量级.为了提高算法的执行效率,我们提出了一个贪心算法求近似解,将时间复杂度从可能值集长度的指数量级降到线性量级.

3) 通过 2 个真实数据集上的实验表明:在准确性方面,基于枚举的方法和贪心算法均优于现有真值发现算法;在效率方面,贪心算法优于现有的真值发现方法.

1 相关工作

Yin 等人[1]首次提出真值发现问题,并提出一种迭代机制联合推导数据源质量和对象真值.该方法基于启发式:高质量数据源提供的对象值更可能为真,提供越多真值的数据源其质量越高.后续一系列相关研究工作都是在此工作基础之上考虑不同场景、不同影响因素、不同的对象真值和数据源可信性计算方法对基本算法进行了各种扩展:1)考虑数据源之间相互依赖关系提高真值发现的准确率,如拷贝关系[2-3,6-7]、隐含分组结构关系[4]和关联关系[8];2)考虑对象难易程度对真值发现的影响[5],通过估计每个对象真值判断的难易程度,避免数据源从相对容易的事实那里获得过高的可信性分值;3)真值发现的在线处理[9-10],解决很多真值发现方法由于其时间和空间复杂度只适合一次计算的问题;4)考虑属性值之间相互关系,如属性值之间的相似性[1-2]等;5)考虑数据源不同的质量评估方法[3,5,8],一个好的数据源质量模型是解决真值发现的关键.

上述真值发现方法中,文献[1-6,9-10]的真值计算模型均基于单真值假设,部分计算模型不适用于对象的多真值计算.另外由于这些算法只是返回各属性值的可信性分值,如何根据可信性分值选择多个真值仍是一个挑战.而本文提出的多真值发现算法,可以直接返回对象的真值列表.

文献[8,11]可以处理多真值发现问题.文献[11]将数据源质量和对象属性值正确性作为隐含变量构建一个概率图模型(LTM)自动推导对象属性值可信性和数据源质量,它是第1个可以处理多值属性真值发现的方法.LTM 方法假设数据源的准确率和召回率服从 Beta 分布,据此推导属性值为真的概率.如果真实数据集不满足假设的分布,则 LTM 算法的效率则受到很大影响.与 LTM 方法不同,本文将多真值问题转化为最优化问题,因此对真实数据集的分布没有限制.文献[8]基于贝叶斯方法对数据源之间的关系进行建模,从而提高真值发现算法的准确率,该模型中考虑了多真值发现问题.与本文方法不同的是:该方法采用监督学习的方法,通过训练数据直接计算数据源的质量,进而推导属性值为真的概率;而本文采用无监督的迭代机制联合推导数据源质量和对象真值列表.另外,文献[8,11]均返回属性值为真的概率,因此选择真值列表时同样需要确定选择概率值大于阈值 K 的属性值为真值,而本文提出的方法可直接返回对象真值列表避免阈值的选择问题.

2 问题定义

下面我们首先介绍问题相关定义.

定义 1. 多值属性.表示对象在某一特定属性上可以有多个值.

例 1. 如表1所示,图书作者属性就是一个多值属性.一本书可以同时有多个作者.每个数据源可以为同一对象的某个属性提供多个属性值.例如 Barnes & Noble 数据源为 Rapid Contextual Design 图书提供了3个作者.

定义 2. 对象可能值集.所有数据源为该对象提供属性值的集合,即各数据源为该对象提供的属性值集合的并集.

例 2. 如表1所示,Rapid Contextual Design 的可能值集是所有数据源为其提供的作者集合{Karen Holtzblatt, Jessamyn Wendell, Shelley Wood, Jessamyn Burns Wendell, Wood}.

定义 3. 对象值向量.该向量为二值向量,描述了给定数据源提供的对象观察值在对象可能值集上的分布情况.

对象值向量的获得方法是:向量长度为该对象可能值集的长度,如果数据源提供了对象可能值集上的第 i 个对象值,则其对象值向量的第 i 个元素设置为1,否则设置为0.

例 3. 如表1所示,根据例1中得到的对象可能值集,数据源 Barnes & Noble 的对象值向量为(1,1,1,0,0).

定义 4. 数据源质量.表示数据源提供对象值的准确性.这里我们采用数据源提供的对象值与对象真值之间总体的相似性来衡量,即如果数据源提供的对象值情况越接近于对象真值则其质量越高.

定义 5. 多真值发现问题.从多个数据源提供的冲突数据中找到对象多值属性的真值列表.

令 $O=\{o_1,o_2,\cdots,o_N\}$ 是对象的集合,每个对象代表真实世界中的一个实体,N 表示对象总数.$S=\{s_1,s_2,\cdots,s_M\}$ 为结构化数据源集合,M 表示数据源总数.对象多真值发现问题与以往单真值发现问题不同,每个对象 $o\in O$ 可以有多个真值,一个数据源 $s\in S$ 可以为其提供多个值(也可以不提供).令 $V_{i,j}$ 表示数据源 s_i 为对象 o_j 提供的值集.$V_{*,j}=\{v_1,v_2,$

…,v_{L_j}}表示对象 $o_j \in O$ 可能值集,即 $V_{*,j}=\bigcup_{i=1}^{M} V_{i,j}$,其中 L_j 表示对象 o_j 可能值集的长度.令 $Truth_i$ 表示对象 $o_i \in O$ 所有真值组成的集合.$D=\{(v,i,j)\mid 1\leqslant i\leqslant M \wedge 1\leqslant j\leqslant N\}$ 为数据源提供的冲突数据集,每个元组表示数据源 s_i 为对象 o_j 提供的属性值为 v.我们需要解决的问题是给定冲突数据集 D,为每个对象 $o_j \in O$ 从其可能值集 $V_{*,j}$ 中找到对象所有真值组成的集合 $Truth_j$.

3 MTruths:多真值发现方法

我们的目标是找到最可能正确的对象属性值集合$\{Truth_i \mid 1\leqslant i\leqslant N\}$,得到的结果应该尽可能与冲突集中数据源提供的对象值情况相似.但是不同的数据源其质量不同,高质量数据源提供的对象值与真值的相似度越高,而低质量的数据源其相似度越低.考虑到实际应用中数据源的质量一般没有先验知识提供,因此本文采用无监督的迭代方法计算,每次迭代过程分 2 步进行:1)通过上次迭代获得的数据源质量计算真值情况;2)通过本次迭代获得的真值情况计算数据源的质量.接下来我们首先介绍数据源质量的评估方法,然后介绍多真值发现方法.本节中所使用的所有变量如表 2 所示:

Table 2 Variables of MTruths

表 2 MTruths 使用的变量

Name	Description
M	Number of Data Sources
N	Number of Objects
O	Set of Objects
S	Set of Data Sources
$V_{i,j}$	Set of Values Provided by Source S_i for Object o_j
$V_{*,j}$	Set of Values Provided by all Sources for Object o_j
L_j	Length of $V_{*,j}$
$\boldsymbol{A}_{i,j}$	Vector of Values Provided by Source S_i for Object o_j
$\boldsymbol{A}_{*,j}$	Vector of Truth Values for Object o_j
$Truth_j$	Set of Truth Values for Object o_j
Q_i	Quality of Source S_i
$w_{i,j}$	Probability that the i'th Value of Object o_j is True
$\boldsymbol{W}_j$	Probability Vector of Values to be True for Object o_j

3.1 数据源质量评估

数据源质量越高则其提供的对象观察值与对象真值越相似,反之其质量越低则两者相似性越低.因此,本文通过数据源提供的对象观察值与对象真值之间的相似性来度量数据源质量.

针对多值属性,每个对象可能有多个真值,并且每个数据源可以为每个对象提供多个值,因此本文采用二值向量表示对象的取值情况.令向量 $\boldsymbol{A}_{i,j}$ 表示数据源 s_i 为对象 o_j 提供的值向量.向量 $\boldsymbol{A}_{i,j}$ 的长度为对象 o_j 可能值集 $V_{*,j}$ 的长度 L_j.向量 $\boldsymbol{A}_{i,j}$ 中第 l 个元素的取值为

$$A_{i,j}[l]=\begin{cases}1, & V_{*,j}[l]\in V_{i,j},\\ 0, & V_{*,j}[l]\notin V_{i,j},\end{cases} \tag{1}$$

其中,$V_{*,j}[l]$表示对象 o_j 可能值集 $V_{*,j}$ 中的第 l 个元素.令 $\boldsymbol{A}_{*,j}$ 表示对象 o_j 的真值向量,即对象 o_j 的真值集合 $Truth_j$ 的值向量.

为了计算数据源提供的对象观察值与真值之间的相似性,首先需要定义不同值向量之间相似性计算公式.我们计算值向量之间相似性时,不但考虑数据源对象值肯定的相似,还考虑其对象值否定的相似性.信息检索中常采用向量内积的方法计算 2 个文档向量相似性,但由于其只考虑对象值肯定的相似性,而忽略了否定相似性.例如 2 个向量(1,0,0,1,0)和(1,0,1,1,0)内积结果为 2,表示有 2 个同为 1 的元素,但是未考虑同为 0 的元素相似性.本文提出 2 种向量相似性计算方法考虑了属性值否定相似性因素.方法 1 可采用向量余弦相似性度量 2 向量之间相似性:

$$sim(\boldsymbol{A}_{i,j},\boldsymbol{A}_{*,j})=\frac{\boldsymbol{A}_{i,j}\cdot\boldsymbol{A}_{*,j}}{\|\boldsymbol{A}_{i,j}\|\|\boldsymbol{A}_{*,j}\|}. \tag{2}$$

第 2 种相似性计算方法为

$$sim(\boldsymbol{A}_{i,j},\boldsymbol{A}_{*,j})=\boldsymbol{D}_{i,j}\cdot\boldsymbol{D}_{i,j}^{\mathrm{T}}, \tag{3}$$

其中,向量 $\boldsymbol{D}_{i,j}$ 描述 $\boldsymbol{A}_{i,j}$ 与 $\boldsymbol{A}_{*,j}$ 中对应元素是否相同,计算方法为

$$D_{i,j}[l]=\begin{cases}1, & A_{i,j}[l]=A_{*,j}[l],\\ 0, & A_{i,j}[l]\neq A_{*,j}[l].\end{cases} \tag{4}$$

计算数据源质量我们使用数据源提供的所有对象的值与其真值之间的相似性度量:

$$Q_i=\sum_{1\leqslant j\leqslant N\wedge V_{ij}\neq\varnothing} sim(\boldsymbol{A}_{i,j},\boldsymbol{A}_{*,j}). \tag{5}$$

对 Q_i 进行标准化处理为

$$Q_i'=\frac{Q_i}{\sum_{k=1}^{M} Q_k}. \tag{6}$$

通过上述方法评估数据源质量.接下来,根据取得的数据源质量信息进一步计算对象的真值集合.

3.2 多真值发现

对象的真值选取结果应该最大程度地接近冲突

数据集 D 提供的对象取值情况,即得到的真值向量与冲突数据集提供的值向量相似度达到最大,其目标函数为

$$\max_{\{\boldsymbol{A}_{*,k}\mid \boldsymbol{A}_{*,k}[l]\in\{0,1\}\wedge 1\leqslant l\leqslant L_k\wedge 1\leqslant k\leqslant N\}}\left(\sum_{j=1}^{N}\sum_{i=1}^{M}Q_i\times sim(\boldsymbol{A}_{i,j},\boldsymbol{A}_{*,j})\right). \tag{7}$$

由于迭代过程中计算对象真值时数据源质量已经确定,且本文提出的 2 个相似性函数均为凸函数,所以目标函数是凸函数的线性组合,故该目标函数也为凸函数,定能找到一个最优解使得目标函数取最大值.

下面我们提出 2 种求最优解的方法:枚举法、贪心算法.

1) 枚举法

由式(7)可以看出,当每个对象的真值集合与冲突数据集提供的该对象值集相似性达到最大时,即可满足目标函数.因此,对每个对象 o_j 求满足 $\max_{\boldsymbol{A}_{*,j}}\sum_{i=1}^{M}Q_i\times sim(\boldsymbol{A}_{i,j},\boldsymbol{A}_{*,j})$ 的真值向量.对象 o_j 真值向量长度 L_j 为其可能值集的长度,因此如果使用枚举法需要列举 2^{L_j} 组不同的值向量,选择具有最大相似度的值向量作为真值向量 $\boldsymbol{A}_{*,j}$,具体算法如算法 1 所示.算法首先构造枚举向量 $\boldsymbol{B}$(行④~⑦);然后计算该值向量与各数据源为对象 o_j 提供的值向量的相似度之和(行⑧~⑩),并与当前最大相似度比较,保留相似度大的向量(行⑪~⑬);如此循环直到长度为 L_j 的所有二值向量枚举完毕.经过分析,该算法的时间复杂度为 $O(M2^{L_j})$.枚举算法的时间复杂度很高,但它可以找到准确的解.

算法 1. 基于枚举的方法(Enum_M).

输入:所有数据源为对象 o_j 提供的值集 $V_{*,j}$、各数据源质量$\{Q_i\mid 1\leqslant i\leqslant M\}$;

输出:对象 o_j 真值向量 $\boldsymbol{A}_{*,j}$.

① for $x=1$ to $2^{L_j}-1$
② 将 $\boldsymbol{B}$ 设置为长度是 L_j 的零向量;
③ $i=1$;
④ while $x!=0$ do
⑤ $B[i++]=x\%2$;
⑥ $x=x/2$;
⑦ end while
⑧ for $i=1$ to M
⑨ $temp+=Q_i\times sim(\boldsymbol{A}_{i,j},\boldsymbol{B})$;
⑩ end for
⑪ if $temp>temp\_max$ then
⑫ $\boldsymbol{A}_{*,j}=\boldsymbol{B}$, $temp\_max=temp$;
⑬ end if
⑭ end for
⑮ return $\boldsymbol{A}_{*,j}$.

2) 贪心算法

鉴于枚举方法时间复杂性过高,当对象可能值集太大时,其算法执行效率低.为了减少需要比较的值向量数目,本文设计了一个对象真值选择策略:以对象值为真的可能性高低先将对象进行排序,优先选择正确可能性高的对象值作为真值.

对象值为真的可能性通过各数据源加权投票的方法度量:

$$w_{j,l}=\frac{\sum_{1\leqslant i\leqslant M\wedge V_{*,j}[l]\in V_{i,j}}Q_j}{\sum_{1\leqslant i\leqslant M\wedge V_{i,j}\neq\varnothing}Q_i}, \tag{8}$$

根据式(8)生成对象 o_j 各值为真的可能性向量 $\boldsymbol{W}_j$.

根据对象值为真可能性的高低,选 Top N 个对象值作为对象真值集合,使得选出的真值集合与对象 o_j 的观察值集之间的相似度达到最高.具体实现方法如算法 2 所示,算法 2 分 2 步进行:1)计算每个对象值为真的可能性(行②~④);2)依次按照为真可能性从高到低的顺序选择对象值加入真值集合(行⑦~⑧),再计算该值集与各数据源提供的值集之间的相似性之和(行⑨~⑪),将得到的相似性与目前保留的最大相似性比较选择相似性大的值集作为候选真值集合(行⑫~⑮),如此循环直到候选值集不再发生变化.该算法的时间复杂度为 $O(ML_j)$,与枚举算法的 $O(M2^{L_j})$ 相比算法效率得到了很大的提升,但其得到的解为近似解.

算法 2. 多真值发现的贪心算法(Greedy_M).

输入:所有数据源为对象 o_j 提供的值集 $V_{*,j}$、各数据源质量信息$\{Q_i\mid 1\leqslant i\leqslant M\}$;

输出:对象 o_j 的真值向量 $\boldsymbol{A}_{*,j}$.

① 初始化 $temp\_max=0$, $\boldsymbol{A}_{*,j}$以及 $\boldsymbol{B}$ 为零向量;
② for $i=1$ to L_j
③ 根据式(8)计算对象 o_j 第 i 个值为真的概率 $w_{j,i}$;
④ end for
⑤ $i=1$;
⑥ do
⑦ $l=SelectTop(\boldsymbol{W}_j,i)$;
⑧ $change=$false, $temp=0$, $B[l]=1$;
⑨ for $k=1$ to M
⑩ $temp+=Q_k\times sim(\boldsymbol{A}_{k,j},\boldsymbol{B})$;

⑪ end for
⑫ if $temp > temp\_max$ then
⑬ $\boldsymbol{A}_{*,j}=\boldsymbol{B}$, $temp\_max=temp$;
⑭ $change=$true;
⑮ end if
⑯ $i++$;
⑰ while($change$)
⑱ return $\boldsymbol{A}_{*,j}$.

3.3 算法流程框架

到目前为止,我们已经讨论了数据源质量评估以及多真值计算方法.正如第 3 节所述,整个计算过程是数据源质量评估和多真值发现的一个迭代过程.我们下面给出 MTruths 算法的总流程.

如算法 3 所示,首先设置各数据源质量的初始值 $Q^{(0)}$,然后执行迭代过程.每次迭代主要分为 2 个步骤:步骤 1,计算对象真值集合,根据上一次迭代得到的数据源质量调用基于投票或贪心的多真值发现方法计算各对象的真值集;步骤 2,利用得到的真值集根据式(6)计算每个数据源质量 Q_i,直到算法达到收敛.

算法 3. 多真值发现算法总框架(MTruths).

输入:冲突集 D、数据源集合 S、对象集合 O;

输出:对象真值集$\{Truth_i \mid 1\leqslant i\leqslant N\}$;

① 初始化数据源质量$\{Q_i^{(0)} \mid 1\leqslant i\leqslant M\}$,$n=0$;
② do
③ $n=n+1$;
④ for $j=1$ to N
⑤ 调用基于枚举的方法或者贪心算法计算对象 o_j 的真值向量 $\boldsymbol{A}_{*,j}^{(n)}$;
⑥ end for
⑦ for $i=1$ to M
⑧ 根据式(6)计算每个数据源 s_i 的质量 $Q_i^{(n)}$;
⑨ end for
⑩ until(满足收敛条件)
⑪ 根据 o_j 的真值向量 $\boldsymbol{A}_{*,j}$ 得到真值集合 $Truth_j$;
⑫ return $\{Truth_j \mid 1\leqslant j\leqslant N\}$.

令迭代次数为 K 次,则采用枚举方法的迭代算法 MTruths_Enum 的时间复杂度为 $O(KNM2^{L_j})$,采用贪心算法的迭代算法 MTruths_Greedy 的时间复杂度为 $O(KNML_j)$.

4 实 验

在 2 个真实数据集上,将本文提出的方法与现有真值发现方法从查准率、查全率、收敛速度、运行时间等方面进行对比.

4.1 实验设计

4.1.1 对比算法

实验对比于 5 个算法:

1) Voting-K.采用投票机制计算真值.在所有为该对象提供属性值的数据源中,如果百分比超过 K 的数据源提供了该属性值,则该属性值为一个真值.

2) TruthFinder-K. TruthFinder[1]方法在计算属性值为真的概率时假设每个对象只有唯一真值,最终选取为真概率最高的属性值作为真值.为了选择多个真值,本文选择概率值大于 K 的所有属性值作为真值.

3) LTM-K. LTM 算法[11]对数据源的 2 类错误(错误肯定和错误否定)进行建模,提出一个概率图模型来自动推导属性值为真的概率. LTM-K 取属性值为真概率大于 K 的属性值作为真值.该算法的参数设置采用文献建议的默认参数.

4) MTruths_Enum.本文提出的一种 MTruths 算法,其中对象真值发现时采用枚举方法判断真值列表.

5) MTruths_Greedy.在真值列表发现步骤中为了提高算法效率提出的贪心算法.

4.1.2 数据集

本文实验使用 2 个真实数据集:图书数据集,包含多个图书销售网站提供的图书作者信息;电影数据集,其包含来自多个电影视频网站的电影导演信息.

1) 图书数据集

第 1 个真实数据集采用文献[2]使用的数据集.该数据集爬取了图书网站 abebooks.com 的图书信息,包括书名、ISBN、作者列表和数据源(图书销售网站).我们对原始数据集进行了去重处理,并对来自不同数据源的作者列表中的分隔符进行了统一以便对作者列表进行分割.经过处理后的数据集包含 1 265 本图书、894 个数据源、5 741 个不同的作者名、119 579 条冲突记录.据统计,大量数据源仅提供很少的图书信息,平均每个数据源提供 28 本图书.图书的作者可能值集大小分布情况如图 1 所示,大部分图书的作者可能值集大小区间为[1,7],平均可能值集大小为 4.5.随机选择 100 本图书对作者进行手工标注,将其作为标准集.

2) 电影数据集

电影的导演属性是一个多值属性,一部电影可

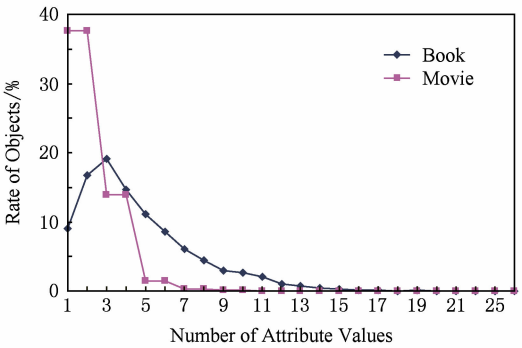

Fig. 1 Distribution of the number of attribute values for objects.

图 1 对象可能值集大小分布

以有多个导演.我们根据新浪娱乐的互动资料库中电影列表列出的电影,从 11 个视频和影评网站搜索这些电影的导演信息,采用电影名称和电影上映年代区分同名问题.针对一部电影有多个名称问题,根据网站提供的电影别名和年代信息判断是否为同一部电影.通过上述方法获得的数据集中包含:12 407 部电影实体、24 567 个不同的导演名以及包含来自 11 个数据源的 114 006 条冲突记录.平均每个数据源提供 4 980 个电影信息.电影导演属性可能值集大小的分布情况如图 1 所示,大部分电影的导演可能值集大小区间为[1, 5],可能值集最大为 25.为了评估真值发现方法的质量,我们随机选择了 100 部电影对其导演信息进行手工标注,对于进口电影导演同时提供中英文名 2 种形式,将其作为标准集.

4.1.3 度量指标

本文从 3 个方面评估真值发现方法准确性:1)查准率 *precision*,表示算法返回的真值中属于标准集的比例;2)查全率 *recall*,表示算法返回的真值中正确真值占标准集的比例;3)调和平均数 *F*-Score,表示查准率和查全率的调和平均数,即 $F1=\frac{2\times precision\times recall}{precision+recall}$.同时,通过算法的收敛速度和运行时间评估算法的效率.

4.1.4 实验环境

本节所有实验硬件环境为:Intel® Core™2 Quad 2.67 GHz 处理器、4 GB 内存、Windows 7 操作系统.本文所有算法包括对比算法均使用 Python 语言实现,软件开发环境为 Python2.7,数据库系统采用 mysql 5.6.17.

4.2 真值发现方法的准确性评估

原始 Voting、TruthFinder 和 LTM 算法只能返回每个属性值为真的可能性,并不能返回对象的真值列表,需要为它们设定一个阈值 K,当概率大于 K 时判定该属性值为真.实验中,我们讨论了 K 分别为 25%,50%,75%时真值发现方法的准确性差异.我们在图书数据集和电影数据集上分别比较了 Voting-K, TruthFinder-K, LTM-K, MTruths _Enum, MTruths_Greey 的查准率、查全率和 *F*-score,结果如图 2 和图 3 所示:

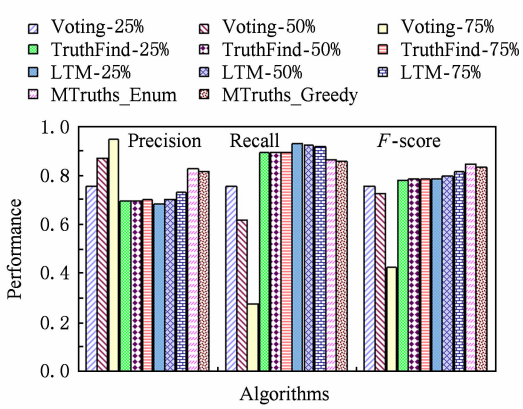

Fig. 2 Precision, recall and *F*-score for the book data set.

图 2 图书数据集算法结果的查准率、查全率和 *F*-Score

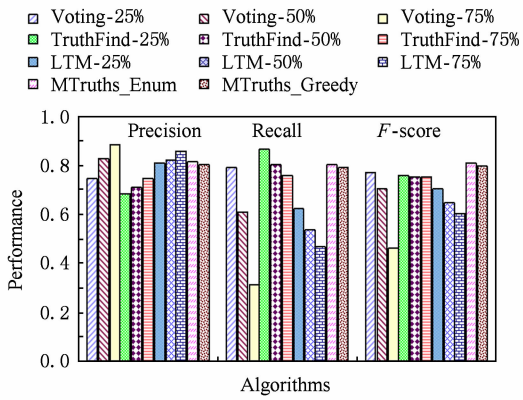

Fig. 3 Precision, recall and *F*-score for the movie data set.

图 3 电影数据集算法结果的查准率、查全率和 *F*-Score

总之,MTruths 算法的 *F*-score 优于其他算法.在图书数据集上,MTruths 算法的查准率较 TruthFinder 和 LTM 算法高出 17%; Voting-50% 和 Voting-75%的算法查准率虽然稍高于 MTruths,但其查全率却显著低于 MTruths.在电影数据集中,MTruths 算法同样获得了较好的 *F*-score.针对多值属性问题,MTruths 算法既考虑了查准率也兼顾到查全率,因此在 2 个数据集中 MTruths 算法的查准率和查全率比较均衡.而其他算法由于受到阈值或者单真值假设的影响,其算法的查准率和查全率

差异较大. MTruths_Enum 和 MTruths_Greedy 相比,前者的查准率、查全率和 F-score 均略高于后者,但总体差距很小.

Voting 算法根据提供属性值的数据源比例判断真值,比例越高的属性值则越可能为真,因此随着阈值 K 的增加其查准率逐渐增高,而查全率显著降低. 在 2 个数据集上,Voting 算法当 $K=75\%$ 时虽然有很高的查准率,但查全率均低于 35%,因此很难为对象找出完整的真值列表.

TruthFinder 方法的查全率低于文献[11]的实验结果,其原因是度量查全率的标准不同. 本文仅当作者名与标准集中的姓名完全相同则认为正确,而文献[11]考虑了姓名的部分正确问题,因此本文对正确性的评判标准更为严格. TruthFinder 方法假设了真值唯一,可以将最可能为真的属性值与其他属性值区分开来,但对需要找到多个真值时,则需要通过阈值的设定完成. 在电影数据集中,随着阈值 K 的变化,TruthFinder 方法的查准率和查全率发生了显著变化.

LTM 算法考虑了多真值问题,在图书数据集上算法准确性仅次于 MTruths. 但由于该算法假设了数据的服从某种概率分布,针对不同的数据集需要调整参数,因此算法的准确性将受到数据的分布以及参数的影响. 在电影数据集上,由于标准集同时提供了导演的中文名和英文名,但大多数数据源仅提供其中一种,虽然其他算法的查全率也有所降低但不显著,但 LTM 算法的查全率显著降低,从而影响了其算法的准确性.

总之,MTruths 算法在准确性方面总体优于其他算法,且不受阈值影响. MTruths_Enum 算法准确性略优于 MTruths_Greedy.

4.3 真值发现方法效率评估

由于阈值 K 对算法 Voting-K,TruthFinder-K,LTM-K 的影响仅仅体现在算法查准率、查全率和 F-score 的计算上,对算法执行时间影响很小. 因此在对比算法时间时,我们仅列出 $K=50\%$ 的执行时间,如表 3 所示. Voting 算法最为高效,其次是 MTruths_Greedy. 由于 MTruths_Enum 算法是枚举算法,图书数据集上算法运行时间达到 70.4 min. 但是,MTruths_Enum 算法的 F-Score 在所有算法中也是最高的,为了获得更好的结果在离线的环境下这样的时长也是可以接受的. TruthFinder-50% 方法在 2 个数据集上的运行时间差异很大,主要是因为运行时间不仅与数据量有关还与收敛速度相关,在图书数据集上 TruthFinder-50% 方法迭代了 40 次收敛,而在电影数据集上其迭代 7 次收敛,故电影数据集上的运行时间远远少于图书数据集.

Table 3 Comparison of Runtime on Two Data Sets

表 3 对比各算法在 2 个数据集上运行时间 s

Algorithm	Book	Movie
Voting-50%	1.31	1.69
TruthFinder-50%	71.13	17.31
LTM-50%(100iter)	34.43	40.68
MTruths_Enum	4 222.83	982.26
MTruths_Greedy	5.55	4.57

图 4 显示了 MTruths 的枚举算法和贪心算法在电影数据集上的收敛速度. 本文算法在迭代过程的收敛条件是:采用 2 次迭代得到的数据源质量向量余弦相似性来衡量 2 次迭代结果的变化情况,相似性越大则变化越小,当变化达到一定阈值则迭代停止. 从图 3 可以看出 2 个算法都可以快速收敛. 经过多次实验,我们的算法在 5 次迭代后即可满足收敛条件.

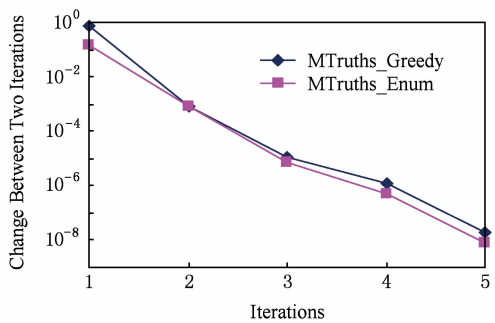

Fig. 4 Convergence rate of iterations on movie data set.

图 4 电影数据集上迭代的收敛速度

对比 MTruths_Enum 和 MTruths_Greedy 算法执行时间随对象可能值集大小变化的情况. 如图 5

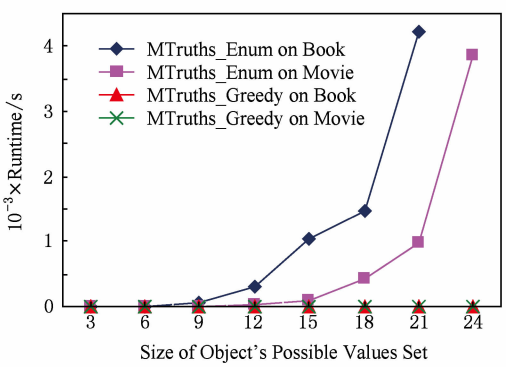

Fig. 5 Time with increasing the size of possible values sets of objects selected.

图 5 算法执行时间随对象可能值集大小变化情况

所示,在2个数据集上分别选择对象可能值集分别小于等于3,6,9,12,15,18,21,24的对象集合.算法MTruths_Enum的执行时间随对象可能值集大小呈指数级增长.MTruths_Greedy时间复杂度是对象可能值集大小的线性量级,因此其效率远远高于MTruths_Enum算法,但该算法的查全率、查准率和F-score略低于MTruths_Enum.因此,在实际应用中,当数据集的对象可能值集较大时,可以选择贪心算法.

5 结 论

Web中存在大量冲突信息,如何从冲突信息中找到正确信息是数据集成领域研究的一个重要问题.同时,多值属性普遍存在,很多对象的属性存在多个真值.然而,已有真值发现方法多数针对单值属性.针对多真值发现问题,本文提出一个MTruths方法将该问题转化成一个最优化问题.根据观察,对象真值应尽可能与冲突集提供的观察值相似.因此,所求的对象真值应该使其与各数据源提供的属性值相似度加权和达到最大.另外,计算真值过程中,我们考虑了数据源质量对真值发现的影响.通过迭代的方法,联合推导数据源质量和对象真值.本文分别提出枚举方法和贪心算法求真值集的最优解.通过2个真实数据集上的实验结果表明,这2种方法在准确性方面均优于已有真值发现方法,贪心算法在效率方面优于已有真值发现方法.

参 考 文 献

[1] Yin Xiaoxin, Han J, Yu P S. Truth discovery with multiple conflicting information providers on the Web [J]. IEEE Trans on Knowledge and Data Engineering, 2008, 20(6): 796-808

[2] Dong X L, Berti-Equille L, Srivastava D. Integrating conflicting data: The role of source dependence [J]. Proceedings of the VLDB Endowment, 2009, 2(1): 550-561

[3] Dong X L, Berti-Equille L, Srivastava D. Truth discovery and copying detection in a dynamic world [J]. Proceedings of the VLDB Endowment, 2009, 2(1): 562-573

[4] Qi Guojun, Aggarwal C C, Han J, et al. Mining collective intelligence in diverse groups [C] //Proc of the 22nd Int Conf on World Wide Web. New York: ACM, 2013: 1041-1052

[5] Galland A, Abiteboul S, Marian A, et al. Corroborating information from disagreeing views [C] //Proc of the 3rd ACM Int Conf on Web Search and Data Mining. New York: ACM, 2010: 131-140

[6] Blanco L, Crescenzi V, Merialdo P, et al. Probabilistic models to reconcile complex data from inaccurate data sources [C] //Proc of the 22nd Int Conf on Advanced Information Systems Engineering. Berlin: Springer, 2010: 83-97

[7] Li Xian, Dong X L, Kenneth L, et al. Scaling up copy detection [C] //Proc of the 31st Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2015: 89-100

[8] Pochampally R, Sarma A D, Dong X L, et al. Fusing data with correlations [C] //Proc of the 2014 Int Conf on Management of Data. New York: ACM, 2014: 433-444

[9] Liu Xuan, Dong X L, Ooi B C, et al. Online data fusion [J]. Proceedings of the VLDB Endowment, 2011, 4(11): 932-943

[10] Zhao Zhou, Cheng J, Ng W. Truth discovery in data streams: A single-pass probabilistic approach [C] //Proc of the 23rd ACM Int Conf on Information and Knowledge Management. New York: ACM, 2014: 1589-1598

[11] Zhao Bo, Rubinstein B I P, Gemmell J, et al. A Bayesian approach to discovering truth from conflicting sources for data integration [J]. Proceedings of the VLDB Endowment, 2012, 5(6): 550-561

Ma Ruxia, born in 1977. PhD candidate at Renmin University of China. Student member of China Computer Federation. Lecturer in Capital Normal University. Her main research interests include Web data management, the credibility of Web information etc.

Meng Xiaofeng, born in 1964. Professor and PhD supervisor at Renmin University of China. Executive member of China Computer Federation. His main research interests include cloud data management, Web data management, flash-based databases, privacy protection etc.

Wang Lu, born in 1986. PhD candidate at Renmin University of China. Her main research interests include spatial database and location privacy management (luwang@ruc.edu.cn).

Shi Yingjie, born in 1983. PhD. Her main research interests include Web data management, cloud data management, online aggregation techniques over big data.