大数据集成中确定数据准确属性值的 WR 方法

周宁南[2]　张孝[1,2]　王珊[1,2]

[1](数据工程与知识工程教育部重点实验室(中国人民大学)　北京　100872)

[2](中国人民大学信息学院　北京　100872)

(zhangxiao@ruc.edu.cn)

WR Approach: Determining Accurate Attribute Values in Big Data Integration

Ningnan Zhou[2],　Xiao Zhang[1,2],　Shan Wang[1,2]

[1] (Key Laboratory of Data Engineering and Knowledge Engineering (Renmin University of China) of Ministry of Education　Beijing　100872)

[2] (School of Information, Renmin University of China　Beijing　100872)

Abstract Big data integration lays the foundation for high quality data-driven decision. One critical section thereof is to determine the accurate attribute values from records in data pertaining to a given entity. The state-of-the-art approach R-topK argues to design rules to decide relative accuracy among attribute values and thus obtain accurate values. Unfortunately, in cases where multiple true values or conflicted rules exist, it requires rounds of human intervention. In this paper, we propose weighted rule (WR) approach for determining accurate attribute values in big data integration. Each rule is augmented with weight and thus avoid human intervention when conflicts occur. This paper designs a chase procedure-based inference algorithm, and proof that it can figure out weighted constraints over relative accuracy among attribute values in $O(n^2)$, which introduces constraints for finding accurate data values. Take conflicts among constraints into consideration, this paper proposes an O(n) algorithm to discover accurate attribute values among combination of data values. We conduct extensive experiments under real world and synthetic datasets and the results demonstrate the effectiveness and efficiency of WR approach. WR approach boosts performance by factor of 3-15x and improves effectiveness by 7%-80%.

Keywords　Big Data Integration; Data Quality; Data Accuracy

摘要 大数据集成是提供高质量数据进行决策的基础。集成的一个关键环节是根据实体在数据库中的不同元组确定其准确属性值。最新的 R-topK 方法在数据上实施人工设计的规则确定属性值间的准确程度，得到了相对准确的属性值。然而这种方法在处理多个可能的准确值或设计的规则存在冲突等情况下需要较多人工交互。为此本文提出基于权重规则的 WR(Weighted-Rule)方法确定大数据集成中数据的准确属性值。该方法每对属性值间准确程度的判断规则上扩充了权重，在准确值发生冲突时避免了 R-topK 方法中人工交互干预。本文基于追逐过程设计了约束条件推理算法，并证明它能够在 $O(n^2)$内推导出每对属性值间的带权重的准确程度，形成推导准确属性值的约束条件。面对约束条件中可能的冲突，本文提出了目标求解算法，在 O(n)时间内从所有属性值组合中搜索最可能的准确属性值。我们在真实和合成数据集中进行了充分的实验，验证了 WR 方法的效果和效率。WR 方法较 R-topK 方法在性能上提高了 3-15 倍，在效果上提升 7%-80%。

关键词 大数据集成；数据质量；数据准确性

随着大数据时代的到来，大数据集成[1]为政府、　企业、个人根据数据进行决策提供了数据基础[2][3]。

收稿日期：2006-05-25 Date 六号

基金项目：中国人民大学研究生科学研究基金项目资助(项目名称:基于时空数据语义推断社交关系的研究　项目编号:14XNH115)；国家 973 计划项目“网络大数据感知融合与表示”(2014CB340403);国家 863 计划项目“开放环境下海量 web 数据提取、集成、分析和管理系统平台与应用” (2012AA011001)；中国人民大学科学研究基金(中央高校基本科研业务费专项资金资助)“高性能复杂数据管理新技术研究”(10XNI018)

大数据集成由三个环节组成[1][2]。不同数据源中的数据经过第一个环节模式匹配[10]被收集起来，并在第二个环节记录连接[11][12]结束后按照相关的实体分类。面对前两个环节收集到的数据库中同一实体 e 的记录集合$\{t_1, \ldots, t_k\}$，大数据集成的第三个环节提出了数据的准确性问题[2]，即构造目标元组 t_e，使得它对于每个属性 A，属性值 t_e[A]都最接近真实值。

一方面，数据的准确性问题在生产中具有迫切需求。据统计[7]，不准确的数据通常占企业数据的 1%-5%，对于有些企业，不准确率甚至高达 30%。据报道[8]，在美国，不准确的数据每年造成千亿美元的损失。为了减少不准确数据带来的危害，数据仓库项目通常需要花费 30%-80%的经费和时间用于数据清洗[9]。

另一方面，数据的准确性问题仍然是目前数据库中最具有挑战性的问题之一[4][5][13]。早期的工作主要集中在数据一致性等方面[5][6]。最新的方法 R-topK[14]通过定义准确性规则确定最准确的属性值。但 R-topK 既不允许规则中存在冲突，也不允许某个实体存在超过一个的目标元组。这导致它在运行过程中需要不断人工调整规则以及为规则不能覆盖的属性设计得分函数来计算 top-k 准确的属性值。

R-topK 方法存在上述缺陷的根本原因在于：多源连接后的数据集中包含庞大的可能准确的属性值组合，而 R-topK 方法试图利用规则推导出其中最准确的属性值。但是，当推导出的准确属性值存在多个时，很难判断这是因为给定实体本身存在多个准确属性值还是因为规则间存在冲突。因此，遇到这种情况时 R-topK 方法会重新设计规则或 top-k 得分函数，其中较多的人工交互阻碍了其在大规模数据上的应用。

在下面的例 1 中，R-topK 方法不能处理两个可能的准确值。

例 1.表 1 以著名运动员 Michael Jordan 为实体，从包含大数据的互联网上收集了他在 1994-95 赛季的得分情况。表 2 是一些官方机构所维护的十分准确的主数据[15]。主数据表明 Michael Jordan 在 1994-95 赛季分别在篮球联盟 NBA 和棒球联盟 SL 打球。

通过对这部分数据进行观察，R-topK 认为存在规则“同一联盟的记录中，场次越大，得分越准确”。因此，考虑得分表 PT 中的第一条元组 t_1 和第三条元组 t_3，由于 t_1[场次]=16>t_3[场次]=1[1]，所以元组 t_1 的得分属性值 424 被认为比元组 t_3 的得分属性值 19 准确，即 t_3[得分]$\leq_{得分}$$t_1$[得分]。如果只考虑 PT 表中前三个元组，我们能得到得分属性上 3 个属性值之间的准确程度($19\leq_{得分} 424 \leq_{得分} 772$)。因此，如果 PT 表只包含表 1 中的前三个元组，R-topK 可以得到得分属性上最准确的值 772。

表 1　Michael Jordan 在 1994-95 赛季得分情况 PT 表

姓名	场次	联盟	得分
Michael Jordan	16	NBA	424
Michael Jordan	27	NBA	772
Michael Jordan	1	NBA	19
Michael Jordan	127	SL	51

但是，由于 Michael Jordan 在 1994-1995 赛季同时在两个联盟得分，因此联盟属性上目标元组 t_e 存在两个准确值：t_e[联盟]=NBA 和 t_e[联盟]=SL。R-topK 方法称这种情况不满足“丘奇-罗斯（Church-Rosser）”性质[2][16]，不能处理这种情况。不幸的是，判断数据与规则是否满足此性质需要在运行时判断[13]，这导致 R-topK 方法在实际数据中需要多次人工设计规则和运行；此外，对于目标元组取值不唯一的情况，R-topK 只能舍弃在相关属性上定义规则，改为人工另行设计得分函数求 top-k 准确的属性值，这严重伤害了方法的准确性，并带来了大量人工交互。

表 2　Michael Jordan 的主数据 MD 表

姓名	联盟	赛季
Michael Jordan	NBA	1994-95
Michael Jordan	SL	1994-95

为了克服 R-topK 方法的这些缺陷，我们给出的权重规则 WR 方法为每个准确性规则设计了权重属性，用来描述规则成立的可能性。利用这些权重，WR 方法可以推导出最准确的若干目标元组。例如，我们在 PT 表和 MD 表上定义表 3 所示的规则。其中，R1 表示在 MD 表中出现过的联盟名称也应该出现在目标元组中，这样，我们可以得到 t_e[联盟]=NBA 和 t_e[联盟]=SL。而 R2 表示在 PT 表的联盟属性值相同时，各元组的场次属性值越大，相应元组的得分属性值越准确。这样，我们可以得到 $19\leq_{得分} 424$和 $51\leq_{得分} 51$等等。注意，由于每条规则带有权重，它们推出的准确程度上的约束条件也带有权重。第 2.2 节详细介绍了约束条件上权重的设置。

这些规则的权重由人工指定，当人们对规则的正确性不肯定时，权重较小。例如，如果我们队另一条规则 R3“目标元组的得分属性值为 19”不太肯定时，我们可以把它的权重设置为 0.5。我们可以看到，R3 和我们从 R2 得到的 $19\leq_{得分} 424$和 $51\leq_{得分} 51$存在矛盾。这时，R-topK 方法不能得出 19 更准确还是 424 或 772 更准确。而 WR 方法利用约束条件上的权值求解目标元组。例如，以<Michael Jordal, 127, SL, 51>和<Michael Jordal, 27, NBA, 772>为目标元组遵循了权重较大的规则 R1 和 R2，而以<Michael Jordal, 1, NBA, 19>为目标元组虽然遵循了权重较小的规则 R3，却违

[1] 在本文中，我们使用 t[场次]表示元组 t 在“场次”属性上的值，用$\leq_{得分}$表示符号左边的属性值的准确程度弱于符号右边的属性值。

[2]这里，“丘奇-罗斯”性质指规则按任意顺序施加最终得到的目标元组唯一。

反了权重较大的 R1 和 R2。直觉上，遵循权重较大的规则的<Michael Jordal, 127, SL, 51>和<Michael Jordal, 27, NBA, 772>被判定为目标元组。

另一点和 R-TopK 方法不同的是，指定权重后，WR 方法就不再需要人工干预。

表 3　扩充权重后的规则

编号	规则	权重
R1	$\forall$ t $\in$ MD, $v = t$[联盟] $\rightarrow t_e$[联盟] $= v$	1.0
R2	$\forall$ t1, t2 $\in$ PT, t1[联盟] = t2[联盟], t1[场次] $\leq$ t2[场次] $\rightarrow$ t1 $\leq_{得分}$ t2	1.0

本文贡献如下：

(1)**设计了支持多真值发现和冲突容忍的基于规则的准确性推导方法。**特别地，我们提出的为准确性规则扩充权重的 WR 方法克服了现有方法的缺陷，允许在规则间存在冲突、准确属性值不唯一等情况下寻找目标元组。

(2)**给出了目标元组求解的高效方法。**我们解决了目标元组求解中的重要问题，包括(a)证明以不同顺序实施带权重的规则都能在 $O(n^2)$时间内停止并得到同样的结果;(b)将根据存在冲突的约束条件求解目标元组的问题表达成马尔科夫逻辑网络的推理问题，并利用约束条件的性质提出了 O(n)时间的高效求解方法。

(3)**充分的实验验证和分析。**我们使用真实和合成数据验证了 WR 方法的效果和效率。特别地，WR 方法在真实数据集上比最新方法快 3-15 倍，其准确率和召回率比最新方法提高 7%-80%。

1　相关工作

目前数据准确性方面的的大量工作可以分成基于数据一致性的方法和基于真值发现的方法两类[17]。

1.1 基于数据一致性的方法

基于数据一致性的方法[14,18-22]假设数据应该满足一组约束。这些约束可以是对数据一致性的要求，比如数据应该满足数据的完整性约束等[18],也可以是从收集到的数据中挖掘出的条件函数依赖[19]等等。通过对收集到的数据进行最少的操作使得这些约束被满足后得到的数据被视为准确数据。

比如，在文献[22]中，条件函数依赖被视为数据应满足的约束，针对具体的函数依赖，元组可以修改函数依赖中被依赖部分的属性值，也可以修改依赖不分的属性值来满足条件函数依赖，由于元组需要进行尽量少的修改使它满足约束条件，但又需要避免与原来的元组过于相似，具体的修改方式的选择由对元组中属性值正确性的估计和修改幅度共同决定。类似地，文献[20]采用条件函数依赖作为数据需要满足的约束，并将准确值的寻找映射为超图优化问题，其中启发式的顶点覆盖算法可以确定最少的需要修改的属性值以满足定义的函数覆盖。最近，利用条件函数依赖推导准确程度的关系被用于寻找不准确的属性值[21]。

考虑到主数据的作用，R-topK 方法[14]首次引入了属性的准确值需要满足的规则，并将这些规则施加到数据中得到属性值间准确程度。它对不同属性上的属性值按照准确程度进行拓扑排序，当各个属性上都有某个属性值经过拓扑排序确认准确程度最高时，目标元组由这些准确程度最高的属性值确定。当存在某个属性上有多个最准确的值或规则间存在冲突时，此方法不能求出目标元组。

与这些方法不同，我们扩充了权重的规则不需要收集到的数据完全满足各种依赖关系；同时，本文求解最满足根据规则导出的准确程度约束条件的目标元组允许任意数量的目标元组存在。

1.2 基于真值发现的方法

基于真值发现的方法[17,23-26]利用对数据源和数据进行建模的方法判断数据的准确值。文献[24]通过数据源的更新历史和数据之间的重复程度利用隐马尔科夫模型判断数据之间的拷贝关系，并使用贝叶斯模型判断准确值。文献[25]通过冗余记录和时效约束在时间戳缺失的情况下辅助恢复数据的时序关系帮助改善数据质量。文献[23]通过对数据源的可信性和数据内容间的重复性对数据的准确性建模。文献[26]使用概率图模型描述数据源间的可信程度，并为导致错误属性值的原因进行建模。文献[17]使用统计学习方法对数据分布进行建模，并以此求出某些属性上最可能的值。

和基于数据一致性的方法相比，本文的方法为约束或规则附加了它们成立的权重，这允许实际数据违反某些规则，提高了方法的准确性。与基于真值发现的方法相比，我们的方法不需要指定数据源是否可靠和发现其中的时序关系。同时，本文直接求出目标元组，而不是根据一些已知正确的属性值求出另一些未知属性的值。

2 求解目标元组

求解目标元组的过程分为两个部分:(1)根据规则得到属性值间准确程度的约束条件及其权重；(2)基于约束条件求解最有可能的属性值组合作为目标元组。下面，我们首先在 2.1 节中介绍扩充权重后的规则，随后的 2.2 和 2.3 节分别介绍目标元组求解的两个过程。

2.1 带权重的规则

扩充了权重的准确规则和 R-topK 方法[14]中的准确规则类似，均通过元组间属性的值或已判断出准确程度的属性值判断其他属性上属性值的相对准确程度。

准确性规则分为两种：

- **第一规则(α-规则或赋值规则)**，直接指定目标元组中的属性值，形式定义如下：

$[\forall t(R(t) \wedge w \rightarrow t_e[A_i] = t[A_i]), p]$，

其中，R(t)表示元组t应被包含在某个关系中。w是下列三种谓词组成的合取公式:(1)$t_1[A_i]$ op $t_2[A_i]$，这里 op 是比较操作符≥, ≤等；(2)$t_i[A_j]$ op c，这里 c 是常数；(3)$t_1 \leq_{A_i} t_2$，这里$\leq_{A_i}$的含义是t_1在属性A_i上

的值$t_1[A_i]$的准确程度弱于t_2在A_i上的值$t_2[A_i]$。p($0 < p \leq 1$)是人们对这条准确规则成立的信心，p越大说明目标元组越应该满足该规则。

例2.以表3中的准确性规则R1为例，元组t被包含在表MD中。元组需满足的条件w为空，即直接指定目标元组的某个属性值。

- **第二规则(β-规则或支配规则)**，根据某些属性上的值的大小或准确性程度定义其他属性上值的准确性程度，形式定义如下：

$[\forall t_1, t_2(R(t_1) \wedge R(t_2) \wedge w \rightarrow t_1 \leq_{A_i} t_2), p]$。

其中，R、w和p的含义和α-规则中的相同。

例3.以表3中的准确性规则R2为例，w是上面提到的形如$t_1 \leqslant_{Ai} t_2$的第三种谓词 t1[场次] ⩽t2[场次]。

为了进行后面的约束条件和准确属性值的推导，我们把上述两种规则中符号→左边涉及到的属性集合记为LAS(R)，右边推断出的属性记为RA(R)。

2.2 推导约束条件

为规则扩充权重属性后，属性值应满足的约束条件包括两部分，即准确程度的约束和它对应的权重。下面，我们分别介绍如何(1)根据规则推断出准确程度的约束以及(2)求出这些约束条件相应的权重。

2.2.1 推导准确程度的约束

α-规则和β-规则分别可以推断出两类约束条件：(1) **α-约束条件，**形如$[t_e[A_i] = v, p]$，以权重p确定了属性A_i上某个值v与其他值间的准确程度（v比其他值都更准确）；(2) **β-约束条件,**形如$[t_m \leq_{A_i} t_n, p]$,以权重p确定了属性A_i上两个属性值$t_m[A_i]$和$t_n[A_i]$之间的准确程度。我们定义每个约束条件c涉及到的属性为CA(c)。本节求出约束条件中属性的准确程度。

准确程度的获得是一个迭代的过程，其获得的方式分为两种，**一种是通过将规则作用于数据，直接得到某个约束条件。**例如，通过α-规则$[\forall t(R(t) \wedge w \rightarrow t_e[A_i] = t[A_i]), p]$，直接得出α-约束条件。**另一种是通过已经得到的约束条件推断出新的约束条件。**比如，根据β-规则$[\forall t_1, t_2(R(t_1) \wedge R(t_2) \wedge w \rightarrow t_1 \leq_{A_i} t_2), p]$，如果已经得到准确程度$t_1 \leq_{A_i} t_2$和$t_2 \leq_{A_i} t_3$，那么就可以推断出新的准确程度$t_1 \leq_{A_i} t_3$。这样通过对数据以及已经得到的准确程度施加规则得到新的准确程度的过程称为追踪过程。

与以往追踪方法面临规则冲突就停止不同，本文的算法会继续实施追踪，因此追踪过程能否结束是一个重要问题。定理1表明无论规则的实施顺序，迭代过程可以在$O(|A|n^2)$时间内终止，并得到相同的准确程度。其中|A|是属性的个数,n是相关实体的记录数。

定理 1. 追踪过程最多经过$O(|A|n^2)$次迭代收敛。

证明.若规则R两次实施得到的结果不同，则存在涉及LAS(R)中属性的新增属性值间准确程度的偏序约束。因为无论新增的偏序对与已有的偏序对是否冲突，在|A|个属性上，最多有$2|A|n^2$个偏序对，因此最多经过$2|A|n^2$次迭代即可求出全部约束条件。■

有了定理1的保证，我们设计了CC算法构造找出属性值间准确程度上的约束条件。

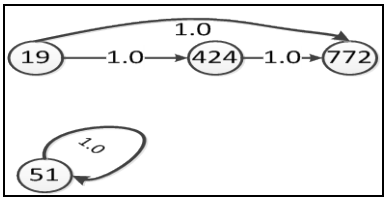


图 1 由准确规则 R2 推断出的 4 对准确程度约束构成的图

算法 1　CC(Construct Constraints)算法

Input : A set of weighted rules *rs*
Output : Accuracy contraints *cs* partitioned by CA
1: *gs* := partition(*rs*); 2: **while (***acs* does not change**)**
3: **for each** rule *r* in **each** *g* in *gs*:
4: **if** *r*∈ β-rule /**r* brings $[t_1 \leq_{A_i} t_2, p]$*/
5: *cs[i]* := *cs[i]*∪{$[t_1 \leq_{A_i} t_2, p]$};
6: **for each** $[t_2 \leq_{A_i} t_3, p] \in cs[i]$:
7: **if** $\nexists [t_1 \leq_{A_i} t_3, p] \in cs[i]$:
8: *cs[i]*:=*cs[i]*∪{$[t_1 \leq_{A_i} t_3, p]$};
9: **for each** $[t_3 \leq_{A_i} t_1, p] \in cs[i]$:
10: **if** $\nexists [t_3 \leq_{A_i} t_2, p] \in cs[i]$:
11: *cs[i]*:=*cs[i]*∪{$[t_3 \leq_{A_i} t_2, p]$};
12: updateWeight(*cs[i]*);
13: **else** /**r* = ($[t_e[A_i] = t[A_i], p]$),*r*∈ α-rule*/
14: *cs[i]* := *cs[i]*∪{$[t_e[A_i] = t[A_i], p]$};

算法CC首先根据给定的规则集合RS确定规则间的依赖程度(第1行)。我们约定两个规则R和R'之间存在依赖当且仅当|LAS(R)∩RA(R')|或|LAS(R')∩RA(R)|不为0。由于规则间有可能存在若干组无关的规则，我们每次迭代一起实施存在依赖的规则(第3-14行)。在对规则追踪过程时，α-规则为目标元组在属性A_i上增加了一个可能取值$t_e[A_i]$(第13-14行)；β-规则在增加了一对属性值上准确程度的同时，还需要计算这时属性A_i上准确程度的传递闭包(第6-13行)。由于定理1已经证明了追踪过程最多经过$2|A|n^2$次迭代，上述过程直至没有心的准确程度约束出现时终止(第2行)。注意，updateWeight使用2.2.2节中介绍的方法计算经过传递闭包产生的约束条件的权重。

例4.以表3中的规则R2为例，假设它的某次实施增加了“424比773的准确程度弱”这样的约束条件；如果之前的迭代已经如“19比424的准确程度弱”的约束条件，则根据传递性我们得到“19比773的准确程度弱”这样的准确程度约束。

2.2.2 计算约束条件的权重

上一节中，规则不同的实施顺序得到相同的准确程度；本节中，我们来确定这些准确程度的权重。

属性值和它们之间的准确程度可以用图表示。例1中得分属性上的4对准确程度约束 $19 \leq_{得分} 424$, $19 \leq_{得分} 772$，$424 \leq_{得分} 772$和$51 \leq_{得分} 51$等如图1所示。我们把每个属性值的所有存在的取值视为节点，两个节点中有边当且仅当这两个点存在于某个准确程度约束中，

边的权重为准确程度约束的权重。可以发现这 4 个约束可以根据图的连通性分成两组。

由于图中包含了得分属性上的所有取值，我们可以如下计算每个属性值的正确权重：

$$\text{node.w} = \begin{cases} 0, \text{如果 out(node)} > 0 \\ \dfrac{\sum_{\text{path}} \prod_{\text{e}\in\text{path}} \text{e.w}}{\text{in(node)}}, \text{其他} \end{cases}$$

其中，out(node)和 in(node)分别代表节点 node 的出度和入度。当某个节点出度不为 0 时，它对应的属性值在已有约束中已经比另一属性值准确程度弱,因此它们对应的权重为 0。注意，该节点对应的权重为 0 并不意味着该节点对应的属性值不可能出现在目标元组中，因为第一种约束条件可以直接定义这些属性值存在于目标元组的权重。我们会在直接加入这些约束条件(算法 1 第 18 行)，并在求解算法中对它们特殊考虑(算法 2 第 4 行)。

那些只有入度的节点，在包含它的连通分量做拓扑排序以后存在若干源点到它的简单路径。根据权重计算方法，属性值 772 对应的权重是每条路径对应的权重的平均值。而每条路径的权重为该路径上准确程度约束的权重的乘积。

例 5. 以图 1 中为约束“$19\leq_{\text{得分}} 772$”计算权重为例，与“得分”属性值 772 对应的节点具有两条从源点（“得分”属性值 1 对应的节点）到它的简单路径：19→424→772 和 19→772。因此，其权重是:$\frac{1.0+1.0\times1.0}{2}=1$。

2.3 推导目标元组

在本节中，我们将根据约束条件求解最可能的目标元组。这个问题可以形式化为马尔科夫逻辑网络(Markov Logic Network, MLN)的推理问题[27]。我们首先形式化目标元组的求解问题，然后针对应用一般 MLN 求解器求解目标元组的不足，基于约束条件的特点提出求解目标元组的优化算法。

2.3.1 基于约束条件求解目标元组

MLN 广泛用于根据一组变量间已有的规则和权重求解这组变量最可能的取值[28]。我们发现可以将本文求解目标元组的问题表达成 MLN 上的推理问题。利用 MLN 首先给出某个元组 t 成为目标元组的可能性，并由此搜索最可能的目标元组。

根据 MLN，某个元组 t 成为目标元组的可能性与该元组满足约束条件的个数和权重有关，其概率是 $\text{P}(\text{t}=t_e)=\frac{1}{z}\text{e}^{\sum p_i n_i(t)}$。其中，z 是归一化因子，$p_i$是某条约束条件对应的权重，$n_i(t)$为 1 时表示元组 t 满足这个约束条件，为 0 时表示 t 不满足。则求解目标元组的过程可以看作搜索某一使得$\text{P}(\text{t}=t_e|\text{rules})$最大的元组 t：

$\text{t}=argmax_t P(t=t_e|rules)\sim argmax_t \sum p_i n_i(t)$。

这已经被证明是一个 NP-hard 问题，但存在高效的求解器，如 MaxWalkSAT[29]。这些求解器求解目标元组的基本方法是对某元组各个属性的赋值检查各个约束条件的符合与违反程度，并根据违反的规则的权重修正元组属性的赋值。在修正时，这些求解器还以一定概率随机赋值。限于篇幅，我们这里仅给出应用马尔科夫逻辑网络求解目标元组的一般方法，下一节详细介绍针对约束条件进行优化后的求解算法。

2.3.2 目标元组求解优化

求解过程中贪心和随机策略结合使用避免了陷入局部最优解。但是，在求解目标元组时会导致可能属性值很多，算法运行时间长的问题。本节中我们介绍如何通过对属性的划分加速目标元组的求解。

我们的方法中一共有两类约束条件，它们都针对一个属性，因此优化问题可以重写为 $\text{t}=argmax_t P(t=t_e|rules)\sim argmax_t \sum p_i n_i(t)$ $\sim argmax_t \sum_{A_i} p_j^{A_i} n_j^{A_i}(\text{t})$。

其中，$p_j^{A_i}$表示 CA 为A_i的约束条件,$n_j^{A_i}(\text{t})$表示元组 t 是否满足这条约束条件[3]。

如算法 TS 所示，基于重写的优化问题，我们可以得到高效的目标元组求解算法。

在 CC 算法中，每个约束条件 c 已经根据 CA(c)进行了划分(算法 1,第 5、7 行)。在 TS 算法中，我们在属性A_i上只考虑图 1 中没有出度的节点所对应的属性值以及α-约束条件中明确规定的属性值(第 3-4 行)。这是因为如果只考虑β-约束，具有出度的节点不会比包含它的连通分量中出度为 0 的节点代表的属性值更准确。只有当α-约束条件直接要求目标元组取这些节点的属性值时，它们才可能成为准确的属性取值。对于这些可能的取值，我们计算它们对规则的违反程度(第 5-9 行)。最后，目标元组由各个属性上最可能的取值组成(第 11 行)。由于出度不为 0 的节点成为候选属性值的情况不超过α-约束的个数，本算法计算各属性上准确值的时间复杂度是 O(n)。

例 6.根据例 5 得到的场次属性上的准确程度约束，我们得到$A_{\text{场次}}.v$={127,27}，类似地，有$A_{\text{得分}}.v$={772,51}等等。最终，利用各个属性上的准确属性值的笛卡尔积和现有元组进行交集，我们得到例 1 中的两个目标元组。<Michael Jordal, 127, SL, 51>和<Michael Jordal, 27, NBA, 772>。

算法 2　TS(Target Solver)算法

Input :　A set of constraints *cs*，relation *r*
Output : Target tuple T_eS
1: **for each** *cs[i]*，*i*∈[1,|A|]:
2:　　*g* := buildGraph(*cs[i]*);
3:　　*candiVal* := {*n.val*|*n*∈*g*,out(*n*)=0};
4:　　*candiVal* := *candiVal*∪{*c*∈*cs*|*c*∈ α-constraints}
5:　　**for each** *val* in *candiVal*:
6:　　　*val.w* := 0;
7:　　　**for each** *valCmp* in *candiVal*:
8:　　　　**if** *val* != *valCmp*:
9:　　　　　*val.w* = *val.w* + *valCmp.w*;

[3]需要指出的是：由于满足权重较大的元组也许会违反很多权重较小的元组，因此该问题不是通常的 top-k 问题。

10: $A_i.v = \{val|val=argmin_{val}\ val.w\}$
11: $T_eS:= \{t|t\in r\ \cap (A_1.v\ \times ... \times A_{|A|}.v)\}$

3 实验结果与分析

本节中我们同时使用真实和合成数据验证本文方法的性能与效果。我们使用最新方法 R-topK[14]作为基线方法。特别地，实验说明了下述问题：(1)本文为规则扩充权重后的方法比只使用规则的推理方法更高效；(2)本文设计的存在冲突的规则提升了寻找准确值的效果；(3)本文中各规则权重的设置是健壮的。

3.1 实验环境与数据集描述

我们使用 C++实现本文方法 WR，实验环境是一台 Linux 服务器，英特尔至强 E5645 2.4Ghz 处理器，8G 内存，1T SATA 硬盘。除了 R-topK[14]方法用于进行性能和效果的比较外，我们还实现了基于数据一致性和真值发现中可以求出目标元组的具有代表性的若干方法：deduceOrder 方法[21]，对 copyCEF 方法[24]以及属性值进行计数的 voting 方法。

我们使用三个被广泛使用的公开数据集(Med, CFP 和 Rest)以及合成数据集(Syn)。公开数据集分别包含了不同公司的医药销售数据、论文征稿数据和餐厅营业数据。合成数据集由向例 1 中的 PT 和 MD 表添加随机选中的数据产生。它们的数据大小和使用规则的情况如表 3 所示。

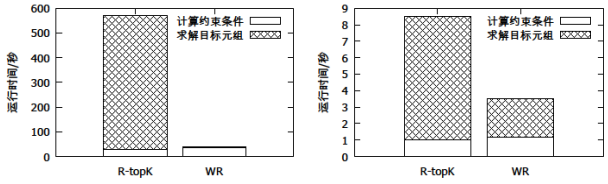
表 3　数据集及使用的规则

数据集	元组个数	属性个数	主数据元组个数	规则数量
Med	10K	30	2.4K	105
CFP	503	22	55	43
Rest	246K	4	0	131
Syn	10-60M	20	1-6M	1-10K

3.2 运行时间

我们使用 Med、CFP 和 Syn 三个数据集测试本文的方法和最新方法针对不同大小的元组个数和不同数量的规则情况下的运行时间。

图 2 使用 Med 和 CFP 数据集在规则数量大致相同时测试两种方法在不同元组个数情况下的运行时间。这两种方法的运行时间包括两部分即寻找约束条件及目标元组求解。



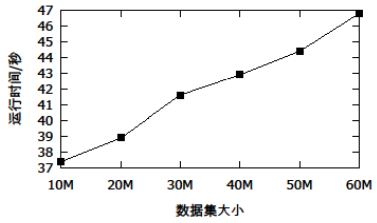
(a)Med 数据集　　(b)CFP 数据集

图 2 真实数据集上 WR 与 R-topK 方法的性能对比

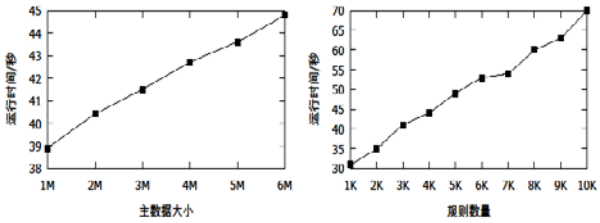
从图 2(a)中可以看出，两种方法计算约束条件的时间相当，其中 WR 方法由于还要计算约束条件的权重，用时比 R-topK 方法稍多不到 10%，但 R-topK 由于规则限制需要运行 top-k 算法，用时远远超过其寻找约束条件的时间及 WR 算法求解目标元组的时间。这是因为 R-topK 方法由于规则的限制，需要为很多属性值求可能性，但通过对遵循和违反约束条件的情况的分析，本文的目标元组求解算法过滤掉了大部分需要在最新的方法中考虑的属性值。

图 2(b)使用 CFP 数据集的结果也可以看出，在数据规模大幅增加的情况下，WR 比 R-topK 用时少的趋势也可以看出，数据集中元组数增多从而每个属性上可能的属性值增多时，WR 算法用了更少的时间。

图 3 使用 Syn 数据集测试元组数量、规则数量以及主数据规模对 WR 性能的影响，它们的初始值分别被设为 30M, 3M 和 3K。图 3(a)-(c)是依次修改这三个变量的实验结果。



(a)改变 Syn 数据集的规模



(b)改变 Syn 中主数据规模　　(c)改变 Syn 中规则数量

图 3 Syn 数据集上 WR 方法性能实验

图 3(a)中，数据规模每增加 1 千万个元组，运行时间增加小于 2 秒，这部分增加的时间主要来自于约束条件的寻找；而图 3(b)显示主数据规模对运行时间影响不大；图 3(c)中，每增加 1 千个规则，运行时间增长 2 秒。图 3 中的数据说明 WR 方法对数据规模、规则个数是可扩展的。

3.3 准确性

我们使用 Med，CFP 和 Rest 三个真实数据比较 WR 方法和 4 种最新的已有方法(DeduceOrder、copyCEF、voting 以及 R-topK)的准确性。我们只考虑只具有一个目标元组的实体以有利于这些现有方法。

并非所有方法都在 3 个真实数据集上进行了实验：(1)DeduceOrder 方法需要数据集中存在 CFD，但 Med 数据集中不存在这种函数依赖;(2)copyCEF 方法需要在数据集中寻找重复的数据，它所需要的数据只有

Rest 数据集中存在;(3)在剩下的 voting、R-topK 和 WR 方法在三个数据集上都进行了实验，检测对于只具有一个目标元组的实体，这些方法对其目标元组检测的准确率。

Med 数据集上的结果：

在 Med 数据集中，我们比较了这些方法寻找到的目标元组有多少是正确的。Voting、R-topK 和我们的方法找到的目标元组中正确的分别占 45%、80%和 90%。其中，R-topK 方法也考虑过在 WR 算法中被判定为目标元组的若干元组，但由于这些元组和其他目标元组存在规则上的冲突，R-topK 方法为了让规则满足大多数目标元组，将它们舍弃。

CFP 数据集上的结果：

CFP 数据集的结果和 Med 数据集的结果类似，Voting、DeduceOrder、R-topK 和 WR 找到的目标元组中正确的分别占 37%，0%，70%和 89%。由于 CFP 数据从多个数据源爬取得到，为它设计规则很难避免冲突，这时 WR 算法比 R-topK 方法的效果更好。

Rest 数据集上的结果：

Rest 数据集只需要判断一个属性值是否准确，因此我们使用准确率、召回率和 F1-measure[30]来比较我们的方法和现有方法的效果。

如表 4 所示，DeduceOrder 实现了 100%的准确率，但只有 15%的召回率，因此 F1-measure 只有 0.26。voting 方法的召回率可以达到 74%，但其准确率只有 60%。copyCEF 在具有不同时期冗余数据的 Rest 数据集上可以实现准确率和召回率的平衡(分别是 76%和 85%)，因此收获了 0.83 的较高的 F1-measure。但基于规则的 R-topK 方法和 WR 算法更好。由与 WR 允许对若干规则的违反，其准确率 92%和召回率 95%都高于目前最新方法 R-topK 的 81%和 90%。

表 4　Rest 数据集上的准确性比较

方法	准确率	召回率	F1-measure
DeduceOrder	100%	15%	0.26
Voting	62%	92%	0.74
copyCEF	76%	85%	0.8
R-topK	81%	88%	0.85
WR	90%	95%	0.92

3.4 冲突与权重敏感程度

我们在三个真实数据集上测试了规则的冲突情况和我们的方法对规则的权重的敏感程度。图 4 显示了我们的方法在三个数据集上分别有 10%、13%和 17%的规则是存在冲突的。这些冲突的规则在 R-topK 算法中只能被去掉，当我们把这些规则去掉后，WR 的准确率和召回率有了 5-10%的下降。

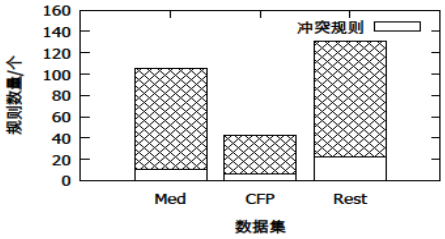


图 4 冲突规则比率

为了便于使用，每条规则的权重不应过于敏感。为了检查 WR 是否对规则的权重敏感，我们对各个规则的权重进行小范围的随机调整，然后检测方法的准确性。三个数据集的准确性随权重变化的情况类似，限于篇幅，我们只在图 5 中展示 Rest 数据集上的情况。从图 5 可以看出，当规则的权重在 10%的范围内改变时，Rest 数据集上准确率和召回率的影响不超过 5%。其中，准确率随着离最优的权重幅度越远，准确率均匀下降。但召回率在权重小幅度上升时下降较慢，而在权重小幅度下降时下降较快，这是因为大部分规则不存在冲突，但小部分存在冲突的规则会产生以下正确的目标元组。当全部规则的权重都小幅度下降时，这部分元组会被其他元组取代，导致召回率快速下降。

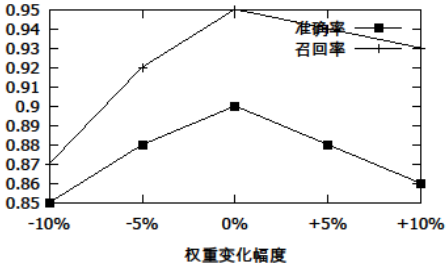


图 5 Rest 数据集对权重敏感程度

4　结束语

本文中，我们为寻找实体相对准确的属性值这一大数据集成中的关键步骤提出了为规则扩充权重的 WR 方法，它克服了现有方法在多准确值发现和不支持冲突规则的缺陷。特别地，我们为实施带权重规则推导约束条件的追踪过程提出了在 $O(n^2)$时间停止的算法，为基于约束条件搜索准确属性值的问题提出基于马尔科夫逻辑网络的线性高效求解算法。真实数据集与合成数据集上的充分实验证明 WR 方法在性能上有 3-15 倍的提升，在目标元组的准确率和召回率上提高了 7%-80%。数据准确性上的研究刚刚起步，如何自动发现规则及其权重是我们将要解决的问题。同时，如何提升规则的表达能力也是数据准确性研究中的重要问题。此外，如何将 top-k 算法应用在 WR 方法上求解近似目标元组也是未来有意义的工作。

参考文献

[1] Dong X. L., Srivastava D.:Big data integration [C] // Proc of ICDE'13. Piscataway, NJ: IEEE, 2013: 1245-1248

[2] Dong X. L., Srivastava D.: Big data integration [J]. The Proc of VLDB Endowment, 2013, 6(11): 1188-1189

[3] Gelman I.: Setting priorities for data accuracy improvements in satisficing decision-making scenarios: A guiding theory [J]. DSS, 2010, 48(1): 507-520

[4] Fan W.: Querying big social data [M]. Berlin: Springer, 2013: 14-28.

[5] Fan W. Geerts F.: Foundations of Data Quality Management [M]. San Rafael: Morgan Claypool Publishers, 2012: 1-20

[6] Batinl C. Scannapieco M.: Data Quality: Concepts, Methodologies and Techniques [M]. Berlin: Springer, 2006: 5-17

[7] Redman T.: The impact of poor data quality on the typical enterprise [J]. Communications of. ACM, 1998, 41(2):79-82

[8] Eckerson W.: Data quality and the bottom line: Achieving business success through a commitment to high quality data [R]. The Data Warahousing Institute, 2002

[9] Firestone J.: Enterprise information portals and knowledge management [M]. New York: Routledge, 2003

[10] Bellahsene Z., Bonifati A. and et al.: Schema Matching and Mapping [M]. Berlin: Springer, 2011.

[11] Getoor L., Machanavajjhala A.: Entity resolution: Theory, practice & open challenges [J]. The Proc of VLDB Endowment, 2012, 5(12): 2018–2019

[12] Koudas N., Sarawagi S. and et al.: Record linkage: similaritymeasures and algorithms [C] // Proc of SIGMOD'06. New York: ACM, 2006: 802-803

[13] Fan W., Geerts F. and et al.: Inferring data currency and consistency for conflict resolution [C] // Proc of ICDE'13. Piscataway, NJ: IEEE, 2013: 470-481

[14] Cao Y., Fan W. and et al. Determining the Relative Accuracy of Attributes [C] // Proc of SIGMOD'13. New York: ACM, 2013: 565-576

[15] Radcliffe J., White A.: Key issues for master data management [R]. Stanford: Gartner, 2008

[16] Abiteboul S., Hull R. and et al.: Foundations of Databases [M]. Indiana: Addison-Wesley, 1995

[17] Yakout M., Berti-Equille L. and et al.: Don't be SCAREd: use SCalable Automatic REpairing with maximal likelihood and bounded changes [C] // Proc of SIGMOD'13. New York: ACM, 2013: 553-564

[18] Arenas M., Bertossi L. E. and et al.: Consistent query answers in inconsistent databases [C]// Proc of PODS'99. New York: ACM, 1999: 68-79

[19] Fan W., Geerts F. and et al.: Conditional functional dependencies for capturing data inconsistencies [J]. IEEE Trans on Database System, 2008 33(1): 1-48

[20] Kolahi S., Lakshmanan L.: On approximating optimum repairs for functional dependency violation [C] //Proc of ICDT'09. New York: ACM, 2009: 53-62

[21] Cong G., Fan W. and et al,: Improving data quality: consistency and accuracy [J]. The Proc of VLDB Endowment, 2007

[22] Blanco L., Crescenzi V. and et al.: Probabilistic models to reconcile complex data from inaccurate data sources [C] // Proc of AISE'10. Berlin: Springer, 2010: 83-97

[23] Dong X., Berti-Equille L. and et al.: Truth discovery and copying detection in a dynamic world [J]. The Proc of VLDB Endowment, 2009

[24] Li, M., Li J. and et al.: Evaluation of Data Currency [J]. Journal of Computer, 2012 35(11): 2348-2360 (in Chinese)
李默涵, 李建中, 高宏.: 数据时效性判定问题的求解算法[J]. 计算机学报, 2012, 35(11): 2348-2360

[25] Galland A. Abiteboul S. and et al.: Corroborating information from disagreeing views [C] // Proc of WSDM. New York: ACM, 2010: 131-140

[26] Richardson M., Domingos P.: Markov logic networks [J]. Machine learning, 2006, 62(1): 107-136.

[27] Domingos P., Lowd D.: Markov Logic: An Interface Layer for Artificial Intelligence [M]. San Rafael: Morgan Claypool Publishers, 2009

[28] Argelich J., F.Manyà: Exact Max-SAT solvers for over-constrained problems [J]. Journal of Heuristics, 2006, 12(4): 375-392

[29] Baeza-Yates R., Ribeiro-Neto B.. Modern information retrieval [M]. New York: ACM press, 1999.

Zhou Ningnan, born in 1990. Currently PhD Candidate at Renmin University of China. His main research interests include high performance database.

Zhang Xiao, born in 1972. Received his PhD degree from Institute of Computing Technology, Chinese Academy of Sciences. Associated professor at Renmin University of China. His main research interests include database architecture, meta data management and big data management.

Wang Shan, born in 1944. Received her master degree from Renmin University of China. She is a professor and PhD supervisor at Renmin University of China. Her main research interests include high performance database, main memory database, video database, DB-IR techniques and data warehouse.