软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(Suppl.(2)):147−156
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种基于不确定规则的数据时效性判定方法*

李默涵, 李建中, 程思瑶

(哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

通讯作者: 李默涵, E-mail: limohan.hit@gmail.com

摘 要: 数据过时是影响数据质量的重要因素,因此判定数据时效性对于提高数据质量至关重要.当前判定数据时效性的方法可分为两类:基于时间戳的方法和基于规则的方法.基于时间戳的方法要求精确完整的时间戳,但这样的时间戳在很多应用中不存在.基于规则的方法不要求时间戳,但现有方法均依赖于冗余元组,且不能对数据时效性做出定量判定.同时,这些方法均基于确定规则,无法表达不确定的领域知识.针对上述问题,提出不确定时效规则及相应的数据时效性模型.基于该模型,进一步给出了两个可定量地判定数据时效性的算法.同时,还给出了时效规则的学习算法.真实数据上的实验结果验证了算法的有效性.

关键词: 数据质量;数据时效性;不确定规则;判定算法;规则学习

中文引用格式: 李默涵,李建中,程思瑶.一种基于不确定规则的数据时效性判定方法.软件学报,2014,25(Suppl.(2)):147−156. http://www.jos.org.cn/1000-9825/14033.htm

英文引用格式: Li MH, Li JZ, Cheng SY. Uncertain rule based method for evaluating data currency. Ruan Jian Xue Bao/Journal of Software, 2014,25(Suppl.(2)):147−156 (in Chinese). http://www.jos.org.cn/1000-9825/14033.htm

Uncertain Rule Based Method for Evaluating Data Currency

LI Mo-Han, LI Jian-Zhong, CHENG Si-Yao

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

Corresponding author: LI Mo-Han, E-mail: limohan.hit@gmail.com

Abstract: Data staleness is one of the most important factors leading to low data quality. It highlights the needs of determining the currency of data to identify whether a database is up-to-date. There are some works on determining data currency, but all these methods have their limitations. Some works require timestamps which are always invalid, and others are based on certain currency rules which can only decide relevant currency and cannot express uncertain semantics. To overcome the limitations of existing methods, this paper introduces a new approach for determining data currency based on uncertain rules. A new class of uncertain currency rule is first introduced. Based on the uncertain rules, mathematical models of data currency are proposed. Two algorithms to determine data currency are developed. A method of automatically learning the uncertain currency rules is also provided. Using real-life data, the effectiveness and efficiency of our methods are experimentally verified.

Key words: data quality; data currency; uncertain rules; evaluation algorithm; rules learning

当前,数据质量问题在许多领域得到了极大关注.有统计指出当前商用数据库中错误率通常在 1%~5%之间,在一些情况下甚至能够达到 30%[1].在美国,劣质数据每年在各个领域共造成 6 000 亿美元的经济损失[2],并且有高达 98 000 人因劣质医疗数据死亡[3].因此,提高数据质量的需求非常迫切.为了提高数据质量,首先需要对数据集合的质量进行判定,以决定是否需要进行进一步修复.随着时间的流逝数据质量会快速下降,例如,由于客户信息变动,每个月约有 2%的商业数据过时失效[2],如果这些数据没能被及时修复,则最坏情况下两年内将有约 50%的数据因过时失效而无法使用.因此,对时效性的判定非常必要.

* 基金项目: 国家重点基础研究发展计划(973)(2012CB316202); 国家自然科学基金(61133002)
收稿时间: 2014-05-07; 定稿时间: 2014-08-19

当前已经有一些工作研究如何判定数据的时效性,这些工作可以分为两类.第一类时效性判定方法基于精确的时间戳[4−6].这些方法在数据的时间戳精确可用的情况下判定较为准确,但是在实际应用中,时间戳往往不存在或不精确[7],因此很难完全依赖时间戳进行时效性判定.

第 2 类方法基于时效规则判定时效性[8−10].这类方法克服了第 1 类方法依赖时间戳的弱点,使得在不需要时间戳的情况下也能判定时效性并找到数据的最新值.然而,此类方法存在 3 点不足.首先,此类方法假设数据库中有多条元组描述同一实体,并依赖这些冗余元组找到数据最新值,但在实际应用中冗余元组并不总是存在.其次,此类方法无法给出数据集合的时效性的定量分析结果,也不能判断当前值相对于某个给定时刻是否过时.例如,如果两条元组分别指出 Alice 在不同时刻的工资是 3000$和 5000$,根据规则“工资只升不降”可知,5000$是较新的工资值,但如果 Alice 的工资上涨至 6000$而数据库又没有及时更新,则 3000$和 5000$相对当前时刻来说,均是过时的值.再次,此类方法假设时效规则都是确定的,即只要条件满足结论必然成立,但用户受其知识所限,往往只能根据一些不确定的知识给出不确定规则,例如用户可能会认为“工资在 90%的情况下是只升不降的”.而且,如果我们定义规则的覆盖范围为满足其条件的元组数,那么确定规则的覆盖范围通常较窄,如果待判定的数据项较多,则用确定规则判定数据集合的时效性虽然能保证判定结果的精确率,但是需要非常多的规则才能保证召回率.而事实上,只需要引入少量的不确定规则就可以在略微牺牲精确率的情况下大幅提升召回率.

为了克服第 2 类方法的 3 个不足,本文提出不确定时效规则及相应的时效性判定方法,贡献如下:

(1) 提出了基于不确定性的时效规则.该规则支持不确定语义,不依赖于冗余元组,不仅可以判定数据的时序关系,还能发现数据在给定时刻是否过时失效.

(2) 建立了数据时效性模型,分别定义了数据项,元组,数据集合的时效性.进一步地,将数据项间的时序关系构建成时效图,将数据与时刻的关系表示为结点标签,并给出了基于时效图的多项式时间的时效性判定算法.

(3) 分两步给出了不确定规则的学习算法.首先,给出了基于数据集合的熵值的训练集抽取算法;其次,给出了基于序列覆盖的规则学习策略.

(4) 在真实的数据集上进行了实验,验证了本文方法的准确性和有效性.

1 不确定时效规则

1.1 语法和语义

关系模式 $R=(A_1,\ldots,A_m)$,其中,属性 A_i 的值域为 $dom(A_i)$.R 可以划分为不相交的子集 R_{ver} 和 R_{unv},其中 R_{ver} 中属性是经用户确认的非过时属性,称为可信属性,而 R_{unv} 中属性称为待判属性,其时效性未知,可能存在过时值.数据集合 $D=\{e_1,\ldots,e_n\}$ 是 R 的实例,其中 e_i 是第 i 个元组,D 的所有 A 属性值称为 A 列,某元组的某属性值(即 $e_i[A_j]$)称为一个数据项.$e_i[A_j]$对应时刻 $t_{i,j}$,表示其有效时间的上限,换言之,$e_i[A_j]$的当前值在 $t_{i,j}$ 之前有效,在 $t_{i,j}$ 之后过时失效.例如,Alice(元组 e_1)的 *City*(属性 A_2)为 *Beijing*,即 $e_1[A_2]$=*Beijing*,若 $t_{1,2}$ 为 2011 年,则说明在 2011 年以前 Alice 居于 *Beijing*,在 2011 年之后,其城市改变,因此 *Beijing* 过时失效.符号 $\prec_A$ 表示数据项的新旧关系,如 $e_i \prec_{A_k} e_j$ 表示 $e_i[A_k]$ 的有效时间早于 $e_j[A_k]$ 的有效时间.D 上的不确定时效规则的语法定义如下.

定义 1. **不确定时效规则(uncertain currency rule,简称 UCR)**. $r:\forall e_1,\ldots,e_k(\psi\rightarrow\omega,\xi(r))$,其中 $e_1,\ldots,e_k$ 是 k 条元组;ψ 称为 r 的左部,其为下述两种谓词的合取:(a) $e_l[A]$ *op* a (*op* 为>,<,或=),其中 $A\in R$,a 是一个常量,$1\leqslant l\leqslant k$,(b) $e_i[A]$ *op* $e_j[A]$;ω 称为 r 的右部,其为下述 3 种谓词中的一个:(a) $e_i\prec_{A'}e_j$,(b) $lb(e_l[A'],\tau)$,表示 τ 是 $e_l[A']$ 有效时间的下界,(c) $ub(e_l[A'],\tau)$,表示 τ 是 $e_l[A']$ 有效时间的上界;$\xi(r)\in[-1,1]$ 为 r 的确定度,其属于(0,1]时,ψ 为真会提升 ω 为真的信度,属于[−1,0)时,ψ 为真会提升 ω 为假的信度,等于 0 时,ψ 与 ω 无关.

为了保证推理的可信度,我们只依赖可信属性来判断待判属性的时效性,因此规定,在 r 的左部可能出现的两种谓词中,属性 A 要么是 R_{ver} 中的属性,要么恰好是 r 的右部中出现的属性.

规则 $r:\forall e_1,\ldots,e_k(\psi\rightarrow\omega,\xi(r))$ 的语义如下:对数据集合 D 中的任意 $e_1,\ldots,e_k$,若 ω 的初始确定度(即信度)为 0,则 ψ 为真会使得 ω 的确定度提升为 $\xi(r)$.

图 1 给出一个 D 及 UCR 的例子.这些 UCR 来自用户的知识,例如,根据“员工入职之后才会有工资”可以写

出 r_1,即“工资的有效时间必然晚于入职时间”,因此,入职时间是工资有效时间的下界;根据“同时入职的员工中,*R&D* 的工资普遍高于 *QA* 的工资”可以写出确定度为 0.6 的规则 r_2,即“对任意 *R&D* 员工 e_i 和 *QA* 员工 e_j,如果入职时间相同且 e_i 工资低于 e_j,那么 e_i 的工资可能未被及时更新,即 $e_j[salary]$的有效时间可能更晚”.

EID	Name	City	Position	Level	EntryYear	Salary
e_1	Alice	Beijing	R&D	T4	2009	3000$
e_2	Bob	Beijing	QA	T5	2009	5000$
e_3	Cindy	Beijing	QA	T4	2008	2000$
e_4	Divide	Shanghai	QA	T3	2013	2500$
e_5	Emily	Shanghai	R&D	T3	2013	3000$

(a) 数据集合 *D*

r_1	$\forall e(e[EntryTime]\neq Null \rightarrow lb(e[Salary], e[EntryTime]), 1)$
r_2	$\forall e_i,e_j(e_i[Position]=R\&D \wedge e_j[Position]=QA \wedge e_i[EntryTime]=e_j[EntryTime] \wedge e_i[Salary]<e_j[Salary] \rightarrow e_i \prec_{Salary} e_j, 0.6)$
r_3	$\forall e_i,e_j(e_i[Level]=e_j[Level] \wedge e_i[Salary]<e_j[Salary] \rightarrow e_i \prec_{Salary} e_j, 0.7)$
r_4	$\forall e_i,e_j(e_i[EntryTime]<e_j[EntryTime] \wedge e_i[Salary]<e_j[Salary] \rightarrow e_i \prec_{Salary} e_j, 0.4)$
r_5	$\forall e(e[Level]=T5 \wedge e[Salary]<6000 \rightarrow ub(e[Salary],2013), 0.9)$

(b) 不确定时效规则

图 1 数据集合 *D* 及不确定时效规则示例

1.2 确定度的计算

贝叶斯理论是处理不确定性最常用的技术,但是其往往需要大量难以获得的统计信息,因此,本文选择贝叶斯理论的一个实用替代方法——确定度因子[11]来表示和计算不确定度.令 $\xi(\psi)$ 和 $\xi(\omega)$ 分别表示 ψ 和 ω 为真的确定度,$\xi(r)$ 表示当 r 的左部为真时,右部也为真的确定度.给定 φ(φ 是条件或结论),其确定度 $\xi(\varphi)$ 如下计算:(a)若 $\xi(\varphi)$ 已知或可以直接根据其他规则推导得到,则直接令 $\xi(\varphi)$ 等于 φ 的确定度(φ 为真则 $\xi(\varphi)=1$,为假则 $\xi(\varphi)=-1$,否则 $\xi(\varphi)$ 是 $(-1,1)$ 之间的实数);(b) 若 $\varphi=\varphi_1\wedge\ldots\wedge\varphi_k$,则 $\xi(\varphi)=\min\{\xi(\varphi_1),\ldots,\xi(\varphi_n)\}$;(c) 若 $\varphi=\varphi_1\vee\ldots\vee\varphi_k$,则 $\xi(\varphi)=\max\{\xi(\varphi_1),\ldots,\xi(\varphi_n)\}$.给定 r,若其左右部分别为 ψ 和 ω,则 $\xi(\omega)=\xi(r)\times\max\{0,\xi(\psi)\}$.

如果由 r_1 和 r_2 得到结论 ω 的确定度分别为 $\xi_{r_1}(\omega)$ 和 $\xi_{r_2}(\omega)$,则 $\xi(\omega)$ 可以由公式(1)合成得到.公式(1)保证了当有不同来源的两个“证据”(即规则 r_1 和 r_2 的左部)都支持ω时,ω的确定度应该比只有一个证据时更高.当有多条规则可以推出ω时,结论的确定度可以反复使用公式(1)合成得到,即先将 $\xi_{r_1}(\omega)$ 和 $\xi_{r_2}(\omega)$ 合成得到 $\xi_{r_1,r_2}(\omega)$,再用 $\xi_{r_1,r_2}(\omega)$ 和 $\xi_{r_3}(\omega)$ 合成得到 $\xi_{r_1,r_2,r_3}(\omega)$.容易证明合成的最后结果与规则的使用顺序无关,但须注意,存在 r_1 和 r_2 左部同时为真且 $\xi_{r_1}(\omega)\times\xi_{r_2}(\omega)=-1$ 是不合理的,因为这意味着存在两条矛盾的规则,一条规则推理出 ω 必然为真,而另一条却推出 ω 必然为假,本文假设规则集合中不存在互相矛盾的规则,因此 $-1<\xi_{r_1}(\omega)\times\xi_{r_2}(\omega)\leqslant 1$.

$$\xi(\omega)=\begin{cases}\xi_{r_1}(\omega)+\xi_{r_2}(\omega)-\xi_{r_1}(\omega)\times\xi_{r_2}(\omega) & \text{if } \xi_{r_1}(\omega)\geq 0 \text{ and } \xi_{r_2}(\omega)\geq 0\\ \xi_{r_1}(\omega)+\xi_{r_2}(\omega)+\xi_{r_1}(\omega)\times\xi_{r_2}(\omega) & \text{if } \xi_{r_1}(\omega)<0 \text{ and } \xi_{r_2}(\omega)<0\\ (\xi_{r_1}(\omega)+\xi_{r_2}(\omega))/(1-\min\{|\xi_{r_1}(\omega)|,|\xi_{r_2}(\omega)|\}) & \text{if } -1<\xi_{r_1}(\omega)\times\xi_{r_2}(\omega)<0\end{cases} \quad (1)$$

2 数据时效性模型

D 与一个表示时刻的常量 θ 关联,称为 *D* 的有效时间下限,即是说,*D* 中的所有数据项都应该在 θ 时刻是有效的.如果 *D* 的时效性没有问题,则任意数据项 $e_i[A_j]$的有效时间晚于 θ,即 $lb(e_i[A_j],\theta)$为真,若存在 $e_i[A_j]$使得 $ub(e_i[A_j],\theta)$为真,则 *D* 中出现了过时数据.

给定数据项 $e_i[A_j]$,判定其时效性即是要判定其有效时间是否晚于 θ.如果规则都是确定的,则可以确定地给出 $e_i[A_j]$的有效时间晚于或早于 θ.但不确定的规则会使得判定结果具有不确定性,故 $e_i[A_j]$的时效性如下定义:将 $lb(e_i[A_j],\theta)$ 为真的确定度记为 $\xi(lb(e_i[A_j],\theta))$;$ub(e_i[A_j],\theta)$ 为真的确定度记为 $\xi(ub(e_i[A_j],\theta))$;$e_i[A_j]$ 的时效性记为 $cur(e_i[A_j])$,其等于 $\xi(lb(e_i[A_j],\theta))$ 和 $\xi(ub(e_i[A_j],\theta))$ 使用公式(1)合并的结果.直观地,$cur(e_i[A_j])\in[-1,1]$ 表示 $e_i[A_j]$过时的确定度,其值大于 0 表示 $e_i[A_j]$更可能是一个非过时值,小于 0 则表示 $e_i[A_j]$更可能是一个过时值,等

于 0 则表示根据现有规则推断不出 $e_i[A_j]$的时效性.

元组和数据集合的时效性基于数据项定义.元组 e 的时效性记为 $cur(e)$, $cur(e)=\left(\sum_{A_j\in R_{unv}} cur(e_i[A_j])\right)/|R_{unv}|$,即 e 的所有待判数据项的时效性的平均.数据集合 D 的时效性记为 $cur(D)$, $cur(D)=\left(\sum_{e\in D} cur(e)\right)/|D|$,即 D 中所有元组的时效性平均.

3 数据时效性判定算法

由时效性定义得知,一旦所有数据项的时效性被计算出来,元组和数据集合的时效性就能够由定义直接得到,因此,本节主要讨论对给定的 $A\in R$,如何判定 A 列的数据项的时效性.首先,我们将数据项之间的时序关系以及其有效时间的上下界建模成时效图,接着根据时效图判定每个数据项的时效性.

3.1 时效图

给定属性 A,可以推理出两种结论:(a) $e_u \prec_A e_v$,即数据项之间的时序关系;(b) $lb(e_i[A],\tau)$ 或 $ub(e_i[A],\tau)$,即数据项有效时间的上下界.为表示结论之间的关系,我们定义时效图的概念.

定义 2. 给定 $R=R_{ver}\cup R_{unv}$,数据库实例 D, UCR 集合 Σ,属性 $A\in R_{unv}$,有向图 $G_A=(V,E)$ 称为 A 的时效图,包含如下 3 部分:(1) $|V|=|D|$,V 中结点 v_i 对应数据项 $e_i[A]$;(2)有向边 $(v_j,v_i)\in E$ 当且仅当存在 $r\in\Sigma$ 使得 e_i 和 e_j 能令 r 左部为真,且右部恰为 $e_i\prec_A e_j$ (称为根据 r 能够直接推出 $e_i\prec_A e_j$),其权值 $w(v_j,v_i)$ 等于所有能直接推出 $e_i\prec_A e_j$ 的规则确定度合并的结果;(3) $\forall v_i\in V$ 对应一个标签集合 $tag(v_i)$,标签 ρ 是下界标签 $\langle\tau,lb,\xi(\rho)\rangle$ (表示 $\xi(lb(e_i[A],\tau))=\xi(\rho)$)或上界标签 $\langle\tau,ub,\xi(\rho)\rangle$ (表示 $\xi(ub(e_i[A],\tau))=\xi(\rho)$).

标签描述了数据项的有效时间可能的区间,故只需对每个节点计算其完整的标签集合即可计算出对应数据项的时效性.因此,数据项的时效性判定可以等价地转化为结点的标签集合计算问题,该问题可分 4 步解决.

Step 1:为 A 列的每个数据项构建一个结点.

Step 2:对任意结点 v_i 和 v_j,扫描一遍 Σ,如果 r 能直接推出 $e_i\prec_A e_j$,就更新 (v_j,v_i) :(a)如果 $(v_j,v_i)\notin E$,则添加 (v_j,v_i) 且令 $w(v_j,v_i)=\xi(r)$;(b)如果 $(v_j,v_i)\in E$,则用 $w(v_j,v_i)$ 和 $\xi(r)$ 合并(式(1))的结果更新 $w(v_j,v_i)$.

Step 3:对任意结点 v_i,扫描一遍 Σ,每当遇到规则能够直接推出 $lb(e_i[A],\tau)$ 或 $ub(e_i[A],\tau)$,就为 v_i 添加标签 $\rho=\langle\tau,lb,\xi(\rho)\rangle$ 或 $\rho=\langle\tau,ub,\xi(\rho)\rangle$.

Step 4:根据图中有向路表示的时序关系,扩展各结点标签集合,使其包含所有可能的上下界标签.

例 1. D 和Σ如图 1 所示. R_{ver} ={*EID*, *Name*, *City*, *Position*, *TLevel*, *EntryTime*}, R_{unv} ={*Salary*}.标签集合如下计算.首先为每条元组构建一个结点;然后使用 r_2~r_4 向时效图中加入 3 条边,权值分别为 0.6,0.8,0.7;接着用 r_1 为每个结点添加标签 $\langle e_i[EntryTime],lb,1\rangle$,使用 r_5 为 v_2 添加标签 $\langle 2013,ub,0.9\rangle$.此时时效图如图 2(a)所示.根据边 (v_2,v_1) 可知 $e_1[Salary]$ 有效时间早于 $e_2[Salary]$ 的确定度为 0.6, v_2 的标签 $\langle 2013,ub,0.9\rangle$ 表示 $e_2[Salary]$ 有效时间早于 2013 的确定度为 0.9,故推断 $e_1[Salary]$ 有效时间也早于 2013 的确定度为 $\min\{0.9,0.6\}=0.6$,因此为 v_1 添加标签 $\langle 2013,ub,0.6\rangle$.按此原理继续扩展,最终时效图如图 2(b)所示.设 θ 为 2013,则 *Salary* 列各数据项时效性为 -0.6, -0.9, -0.7,1,1.D 的时效性为 -0.04. □

前 3 步非常直观,无需赘述,本节剩余部分主要研究 Step 4.标签集合扩展的策略有两种:(a) 如果根据 v_j 到 v_i 的有向路能得到 $\xi(e_i\prec_A e_j)>0$,则可将 $e_j[A]$ 的上界传播给 e_i,将 $e_i[A]$ 的下界传播给 e_j;(b) 如果 $e_i[A]$ 的上界恰是 $e_j[A]$ 的下界,则也可以将 $e_i[A]$ 的下界传播给 e_i.但是两种策略同时使用会导致某些证据被重复使用,进而令确定度计算不准,故本文仅使用策略(a),并将两种策略共同使用的问题作为未来工作.

3.2 基于最长路径的标签集合扩展

将 Step 3 完成之后标签集合不为空的结点称为 landmark 结点,将此时 landmark 结点的标签称为 landmark 标签.对于任意 landmark 结点 v_i,可以分两种情况将其 landmark 标签传播给其他结点:(1) 对结点 v_j,如果 v_i 到 v_j 存在有向路(可达),且路径上各边权值均大于 0,则 $\xi(e_j\prec_A e_i)>0$, v_i 的上界 $\langle\tau,ub,\xi(\rho)\rangle$ 可被传播给 v_j,即可以

为添加上界标签ρ',表示$e_j \prec_A e_i$和$ub(e_i[A],\tau)$的合取,根据第 1.2 节,$\xi(\rho')=\min\{\xi(e_j \prec_A e_i),\xi(\rho)\}$;(2) 同理,如果存在$v_j$,使得$v_j$可达$v_i$且路径上各边权值均大于 0,则有$\xi(e_i \prec_A e_j)>0$,故$v_i$的下界标签$\langle\tau,lb,\xi(\rho)\rangle$可被传播给$v_j$,即可以向$v_j$的标签集合中添加新标签$\rho'=\langle\tau,lb,\min\{\xi(e_i \prec_A e_j),\xi(\rho)\}\rangle$.对任意 landmark 结点$v_i$,其标签$\rho$对应的确定度$\xi(\rho)$都是已知的,那么剩余的问题便是如何计算上述两种情况下的$\xi(e_j \prec_A e_i)$和$\xi(e_i \prec_A e_j)$.

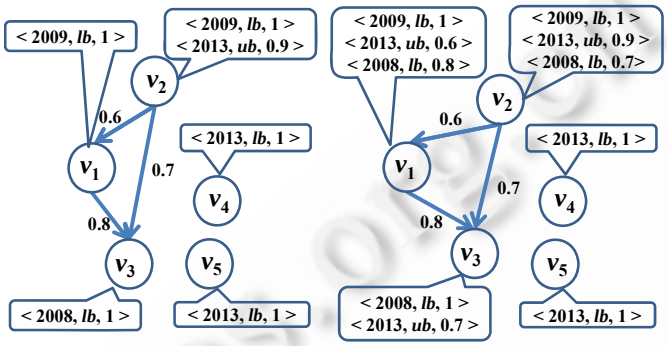


图 2 Salary 属性的时效图

$\xi(e_j \prec_A e_i)$如下计算.对任意e_i和e_j,如果v_i不可达v_j,则$\xi(e_j \prec_A e_i)$=0;设v_i到v_j有一条有向路$p=v_iv_{h_1}\ldots v_{h_k}v_j$,则$p$上的每条边都代表一个时序关系,如$(v_i,v_{h_1})$表示结论$e_i \prec_A e_{h_1}$,那么$e_j \prec_A e_i$相当于$p$上各边表示结论的合取,故有$\xi_p(e_j \prec_A e_i)=\min_{a\in p}\{w(a)\}$,其中$\xi_p$表示根据路径$p$计算得到的确定度,$a$是$p$上的边;如果$v_i$到$v_j$有多条有向路,将这些有向路构成的集合记为$P(v_i,v_j)$,那么由$P(v_i,v_j)$中任意一条路径都能够推理得到$e_j \prec_A e_i$,因此结论$e_j \prec_A e_i$可以看作是这些路径对应的结论的析取,根据第 1.2 节的析取式的确定度计算方法可得$\xi(e_j \prec_A e_i)$的最终结果为$\max_{p\in P(v_i,v_j)}\{\xi_p(e_j \prec_A e_i)\}$,即$\max_{p\in P(v_i,v_j)}\{\min_{a\in p}\{w(a)\}\}$.令时效图中有向路$p$的长度为$\min_{a\in p}\{w(a)\}$,则$\xi(e_j \prec_A e_i)$等于$v_i$到$v_j$的最长路径的长度.$\xi(e_i \prec_A e_j)$的计算方法同理.

对任意 landmark 结点v_i,其上界通过v_i到v_j的最长路径传播给v_j,使得v_j的标签集合得到扩展,而v_i的下界则通过v_j到v_i的最长路径传播给v_j.标签集合扩展算法见算法 1,其中$longest(v_j,v_i)$返回v_j到v_i的最长路长度,$\max\{0, longest(v_j,v_i)\}$保证了只考虑长度大于 0 的路径,$PropLandmarkTags(v_j,v_i,lb,len)$将$v_i$的 landmark 下界传播给$v_j$,$longest(v_j,v_i)$和$PropLandmarkTags(v_j,v_i,ub,len)$同理.对每个 landmark 结点,其也可能有来自其他结点的非 landmark 标签,在传播时,我们只需传播其 landmark 标签,因为传播非 landmark 标签会导致同一个标签被多次传播.Dijkstra 算法[12]只需略作修改就可以用来求$longest(v_j,v_i)$,因此$longest(v_j,v_i)$的时间复杂度与 Dijkstra 算法相同,为$O(|D|\log|D|)$.在最坏情况下需执行$|D|$次$longest(v_j,v_i)$,同时我们假设结点的标签数远小于$|D|$,则算法 1 时间复杂度为$O(|D|^2\log|D|)$.

算法 1. 基于最长路径的标签集合扩展算法.

Input: currency graph G'_A outputted by Step 3;

Output: currency graph G_A with completed tag sets.

1: **for all** $v_i \in Landmarks$ and $v_j \in V$ **do**

2: **if** v_j has at least a direct path to v_i **then**

3: $len=\max\{0,longest(v_j,v_i)\}$

4: $PropLandmarkTags(v_j,v_i,lb,len)$

5: **if** v_i has at least a direct path to v_j **then**

6: $len=\max\{0,longest(v_i,v_j)\}$

7: *PropLandmarkTags*(v_j,v_i,*ub*,*len*)

3.3 基于拓扑排序的标签集合扩展

算法 1 在最坏情况下要计算|*D*|次最长路径,但是当 G_A 是有向无环图(DAG)时,我们可以给出一个更快地基于拓扑排序的标签集合扩展方法.使用该算法,每个结点只需将自己的标签传给其孩子结点即可.

在 Step 3 结束之后,我们为 G_A 中每个标签 ρ 标记其来源 $s(\rho)$, $s(\rho)$ 等于 Step 3 中产生该标签的 landmark 结点的编号.上界标签传播方法如下.首先,在找到一个入度为 0 的结点 $v^{(0)}$,显然 $v^{(0)}$ 不会从其他结点得到上界标签,因此其上界标签集合已经计算完成.将 $v^{(0)}$ 记为第 0 层结点,并从 $v^{(0)}$ 开始传播上界.如果 $v^{(0)}$ 的标签集合不为空,那么对 $v^{(0)}$ 的每个上界 $\langle\tau,ub,\xi(\rho)\rangle$,向其每个孩子的标签集合添加 $\rho'=\langle\tau,ub,\min\{w(v^{(0)},v_c),\xi(\rho)\}\rangle$,其中 v_c 表示当前正在扩展的孩子结点.接着,将 $v^{(0)}$ 删除,在剩余的点中找一个入度为 0 的点 $v^{(1)}$,则 $v^{(1)}$ 的上界标签只可能是自身的 landmark 标签或从 $v^{(0)}$ 得到的上界标签,因此其上界标签集合已经计算完成,我们将 $v^{(1)}$ 标记为第 1 层.与 $v^{(0)}$ 类似,可以将 $v^{(1)}$ 的所有上界传给其孩子结点,并将 $v^{(1)}$ 删去.该过程反复进行,直到图中所有结点都被删掉为止.在标签传播的过程中,每个孩子结点在收到新的标签之后,都需要检查一遍自己的标签集合,如果发现有标签的来源 $s(\rho)$ 相同,则只保留确定度 $\xi(\rho)$ 最高的标签,因为来源相同的标签表示的结论之间是“或”的关系,所以应当取确定度最大值.下界的传播原理与上界标签相似,区别仅在于每次应选取出度为 0 的结点.

用拓扑排序[12]可以将各结点按照其被删除的顺序排成一个序列(拓扑序).基于拓扑排序的标签集合扩展算法见算法 2 所示,其中 *TopologicalSort*(G_A) 按照拓扑序为每个结点标记层数, *PropTage*(v_j,$v^{(lv)}$, *ub*,$w(v_i,v_j)$)将 $v^{(lv)}$的上界传播给 v_j, $ch(v^{(lv)})$ 表示 $v^{(lv)}$ 的孩子结点集合, $pa(v^{(lv)})$ 表示 $v^{(lv)}$ 的父结点集合.假设标签数远小于|*D*|,那么算法 2 的时间复杂度为 $O(|V|+|E|)$,其中 $|V|=|D|$ 是 G_A 的结点数,$|E|$ 是 G_A 的边数.

算法 2. 基于拓扑排序的标签集合扩展算法.

Input: currency graph G'_A outputted by Step 3;

Output: currency graph G_A with completed tag sets.

1: *TopologicalSort*(G_A)

2: **for** $lv=0$ **to** $|D|$ **do**

3: **for each** $v_j \in ch(v^{(lv)})$ **do**

4: *PropTags*(v_j,$v^{(lv)}$,*ub*,$w(v_i,v_j)$)

5: **for** $lv=|D|$ **to** 0 **do**

6: **for each** $v_j \in pa(v^{(lv)})$ **do**

7: *PropTags*(v_j,$v^{(lv)}$,*lb*,$w(v_j,v_i)$)

算法 2 只适用于 DAG,然而,如果 G_A 中有环,而用户对时效性判定结果的要求又不十分精确时,可以通过去除 G_A 中的一些边来将其转化为有向无环图 $\widehat{G_A}$,再用算法 2 完成标签集合扩展.从一个有向图中去掉最少的边使之成为一个 DAG 的问题被称为最小反馈弧集问题,该问题是 NP-难的[13]且已经有较为成熟的算法,因此不再赘述.如果 $|\Sigma|\ll|D|$ 时,则使用 $\widehat{G_A}$ 近似 G_A 仍能得到几乎正确的结果.

定理 1. 如果 $|\Sigma|\leqslant|D|-1$,且 Step 2 中每条规则生成的边数不超过|*D*|,则对任意 $e_i[A]$,设由 $\widehat{G_A}$ 判定得的时效性为 $\widehat{\xi(e_i[A])}$,由 G_A 得到的时效性为 $\xi(e_i[A])$,则 $|\widehat{\xi(e_v[A])}-\xi(e_v[A])|$ 的期望小于 $2|\Sigma|/(|D|-1)$.

证明:设 *D* 有 *n* 条元组,Σ 有 *k* 条规则.对于任意规则 r_i,设 r_i 能够产生 M_{r_i} 条边,那么任意结点 *v* 能得到 r_i 产生的边的概率为 $M_{r_i}/(n(n-1))$.*v* 能从 Σ 中得到边的概率 $P(d(v)>0)=1-\prod_{i=1}^{k}(1-M_{r_i}/(n(n-1)))$.令 $M_{\max}=\max_{i=1}^{k}\{M_{r_i}\}$,则 $P(d(v)>0)\leqslant 1-(1-M_{\max}/(n(n-1)))^k$.由假设 $M_{\max}\leqslant n$ 得 $P(d(v)>0)\leqslant 1-(1-1/(n-1))^k$.由伯努利不等式 $(1-p)^k\geqslant 1-pk$ 得 $P(d(v)>0)\leqslant k/(n-1)$.设 *v* 对应的元组为 e_v,那么 $\widehat{\xi(e_v[A])}\neq\xi(e_v[A])$ 当且仅当 *v* 受某个环的影响,而 *v* 受环影响的概率必然小于 *v* 有边的概率.又因为 $|\widehat{\xi(e_v[A])}-\xi(e_v[A])|\leqslant 2$ 且 $k\leqslant n-1$,所以 $|\widehat{\xi(e_v[A])}-\xi(e_v[A])|$ 的期望不超过 $2k/(n-1)$,即 $2|\Sigma|/(|D|-1)$. □

3.4 根据标签集合计算数据项的时效性

在对所有结点的标签集合都扩展完成之后,即可计算数据项时效性.该过程非常直观:对每个结点,扫描一遍标签集合,找出所有 τ 值早于 θ 的上界标签,将使用式(1)合并得到 $\xi(ub(e_i[A_j],\theta))$,再找出所有 τ 值不晚于 θ 的下界标签,将其确定度用式(1)合并得 $\xi(ub(e_i[A_j],\theta))$,再根据第 2 节数据项时效性定义,计算出其最终时效性.

4 不确定时效规则学习算法

用户通常会因为领域知识不足而无法给出有效的UCR,因此需要有方法能自动发现规则.本节给出UCR的学习框架,并基于该框架,给出基于分层抽样和序列覆盖的两种策略来帮助规则学习.UCR 的学习框架如下.

Step 1:从待估计的数据集合 D 中抽样,并标记时间戳,从而得到一个训练集合 D_{tr}.

Step 2:在 D_{tr} 中挖掘频繁项集集合 $F_{D_{tr}}$,任意 $f\in F_{D_{tr}}$ 的形式为 $\{Attr_1=a_1,Attr_2=a_2,...,Attr_h=a_h\}$.

Step 3:用 $F_{D_{tr}}$ 中的频繁项集作为左部,生成两种规则:(a) 对 $\forall f_1,f_2$,令 f_1^A 和 f_2^A 分别为 f_1 和 f_2 对应的属性名,f_1^V 和 f_2^V 分别为 f_1^A 和 f_2^A 中属性值,生成规则 $\forall e_1,e_2(e_1[f_1^A]=f_1^V)\wedge(e_2[f_2^A]=f_2^V)\rightarrow e_1\prec_A e_2$;(b) 对 $\forall f\in F_{D_{tr}}$ 和时刻 τ,令 $x(e,\tau)$ 为 $lb(e,\tau)$ 或 $ub(e,\tau)$,生成规则 $\forall e((e[f^A]=f^V)\rightarrow x(e,\tau)$.

Step 4:对于 Step 3 中生成的所有约束,计算其成立的确定性因子,并保留满足要求的约束.

Step 2 可用现有方法计算[14],Step 3 的也非常直观,无需赘述.因此本节重点讨论 Step 1 和 Step 4.

4.1 训练集合抽取

手工标记时间戳的开销非常大,因此,要用尽量小的 D_{tr} 反映 D 中属性取值和时间戳的关联关系.得到 D_{tr} 的最直观的方法是使用简单随机抽样:给定样本集合大小 n_s,任何 D 的大小为 n_s 的子集被抽中的概率都相同.然而,这样会使得某些出现频率较低的值难以被抽到,进而导致一些覆盖面小但很重要的规则难以学习.例如,考虑图 5 中所示的数据集合 (U 为待判属性,T 是表示 U 的时间戳),若要抽取大小为 5 的 D_{tr},为了学习到较好的规则集合,e_5,e_6,e_7 应当出现在 D_{tr} 中,这样才能保证关于 t_2 和 t_3 的规则是准确的,但简单随机抽样的 21 种结果中只有 6 个满足要求,很难保证抽到较好的 D_{tr}.

TID	A	B	C	U	T
e_1	a_1	b_1	c_1	u_1	t_1
e_2	a_1	b_1	c_1	u_2	t_1
e_3	a_1	b_1	c_1	u_3	t_1
e_4	a_1	b_1	c_1	u_4	t_1
e_5	a_2	b_5	c_1	u_5	t_2
e_6	a_2	b_6	c_1	u_6	t_2
e_7	a_2	b_6	c_2	u_7	t_3

图 3 数据集合示例

为了克服简单随机抽样的弱点,我们使用基于分层抽样的策略,以使得抽到的样本集合尽量靠近最好的训练集合.该策略如下:(a) 将 D 划分为若干不相交的子集,每个子集作为一层,其中元组可以使用相同的规则来判定时效性.不失一般性地,可设子集的个数为 l,即 $D=\{D_1,D_2,...,D_l\}$. (b)令样本集合大小为 n_s,则从每层抽到的样本个数的期望 $E_i=\min\{n_s/l,N_i\}$,其中 $N_i=|D_i|$.(c) 对 D_i 层,以 E_i 为期望抽取样本.若 E_i 不是整数,那么在一次抽样后,D_{tr} 中的样本数有可能小于 n_s,此时将 $n_s-|D_{tr}|$ 个样本再分配给各层,使每层样本个数的期望

$$E_i=\min\{(n_s-|D_{tr}|)/l,N_i'\},$$

其中,N_i' 表示 D_i 中尚未被加入 D_{tr} 的元组数.重复(c)步,直到 $|D_{tr}|=n_s$.

图 3 中,如果将底色相同的记录划分在同一层,那么我们至少能保证 e_5,e_6,e_7 中有两个出现在 D_{tr} 中.

在时间戳缺失时,我们往往无法得知怎样划分才是最好的.为此,我们使用熵刻画数据集合的纯度.熵越低,则纯度越高,属性取值越一致.令 D 的总体信息熵 $H(D)=-(1/|R|)\sum_{i=1}^{|R|}\sum_{a_{ij}\in dom(A_i)}p(a_{ij})\log p(a_{ij})$,其中 $p(a_{ij})$ 表 A_i 的值域中的第 j 个值 a_{ij} 出现的概率.子集 D_i 的熵越小,则其属性取值越相近,而在这样的 D_i 中就越容易学习到有代表性的规则.进而,分层方法如下:令信息增益 $Gain(D,A)=H(D)-\sum_{v\in dom(A)}H(D_v)\times|D_v|/|D|$,每次选择信息增益最高的属性,将该属性上取值相同的元组划分至同一个子集合,划分次数超过预先设定的阈值后停止,将得到的每个子集作为 D 的一层.

4.2 计算确定度因子及规则选取

由确定度因子的定义[11],对 $r:\forall e_1,...,e_k\ (\psi \to \omega,\xi(r))$, $\xi(r)=MB(\omega,\psi)-MD(\omega,\psi)$.其中, $MB(\omega,\psi)$ 称为给定 ψ 时 ω 的可信度量, $MD(\omega,\psi)$ 称为给定 ψ 时 ω 的不可信度量.令 $P(\omega|\psi)$ 表示 ψ 为真时 ω 为真的概率, $P(\psi)$ 表示 ψ 为真的概率,如果 $P(\psi)$ <1,则 $MB(\omega,\psi)=\big(\max\{P(\omega|\psi),P(\psi)\}-P(\psi)\big)/\big(1-P(\psi)\big)$,否则 $MB(\omega,\psi)$ =1;如果 $P(\psi)$ >0,则 $MD(\omega,\psi)=(\min\{P(\omega|\psi),P(\psi)\}-P(\psi))/\big(-P(\psi)\big)$,否则 $MD(\omega,\psi)$ =1.

在计算 $\xi(r)$ 时,最直观的方法是根据训练集 D_{tr} 生成所有可能的规则,并使用 D_{tr} 中的所有元组来计算 $P(\omega|\psi)$ 和 $P(\psi)$.但在判定时效性时,规则数太多会导致大量的相互矛盾或相互蕴含的规则,进而影响时判定精度,同时也会导致时效性判定耗时严重,因此,我们基于序列覆盖[15]给出了一种规则学习策略,在保证规则覆盖范围的基础上,尽量减少规则数目.该策略的基本原理如下:在学习约束时,首先在当前训练集合上学习出单条规则 r, r 根据当前最频繁的项集构造(以保证每次产生的约束的覆盖范围尽量广),且 $\xi(r)$ 满足预先设定的条件(如绝对值大于等于 0.5);移去被 r 覆盖的正例,然后在剩余的训练数据上执行上述过程来学习第 2 条约束.该过程一直重复,直到正例的覆盖程度达到所希望的比例.因为 UCR 有两种,所以学习分为 2 轮,第 1 轮仅学习结论为 $e_i \prec e_j$ 的规则,第 2 轮则针对时间戳 τ,学习结论为 $lb(e,\tau)$ 和 $ub(e,\tau)$ 的规则.

5 实 验

我们在真实数据上进行了本文实验.所有实验均在 4G 内存,i5 CPU,64 位 win7 系统的 PC 机上运行,代码使用 C++编写.实验数据使用 IMDB(http://www.imdb.com)电影数据,模式为(*ID*,*Title*,*Genres*,*Directors*,*Writers*,*Actors*,*Actresses*),其中 R_{ver} 为{*Genres*,*Directors*,*Writers*,*Actors*,*Actresses*}, R_{unv} 为{*Title*},使用上映年份作为 *Title* 属性的有效时间,并使用本文方法判定给定数据集合中的电影的上映年份是否晚于阈值 θ.

5.1 算法执行效率

时效图的构建共分 4 步,其中第 1 步的时间可以忽略不计,因此我们仅给出第 2~4 步的执行时间,如图 6~图 8 所示.从图 6 中可以看出,当数据集合和规则集合大小增大时,第 2 步耗时增长最快,这是因为在添加边时,需要每一对元组都扫描一遍规则集合才能决定相应的有向边是否存在及权值是多少.图 7 中第 3 步产生 landmark 标签集合的时间与剩余几步相比,几乎可以忽略不计,当 $|D|$=10000 而 $|\Sigma|$=100 时,也仅需 0.06s 就可完成.图 8 所示的第 4 步的时间与 landmark 集合的大小相关,在该组实验中令 $|D|$=10000,变化 landmark 结点的所占的百分比并测量算法执行时间,显然,算法 2 的执行效率要高于算法 1 的执行效率,而且随着 landmark 结点的增多,算法 2 的时间消耗增长的也明显慢于算法 1.

除此之外,我们还考察了规则学习所消耗的时间,结果如图 9 所示.分层抽取训练集合的时间开销最小,仅不到 0.1s.在得到训练集合之后,影响规则学习时间的最重要的两个参数为训练集大小($|D_{tr}|$)及频繁项集的支持度阈值(Support),我们分别改变这两个参数,并研究算法的执行时间.可以看出,规则学习时间随 $|D_{tr}|$ 的上升和 *Support* 降低而上升,在 $|D_{tr}|$=1000 且 *Support* =5时学习规则所需的时间最长,约 8min.

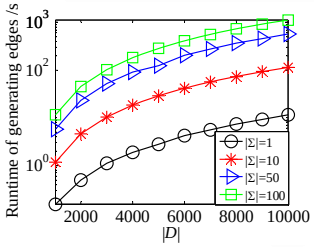


图 6 向时效图中添加边

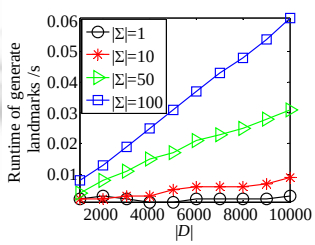


图 7 产生 landmark 标签

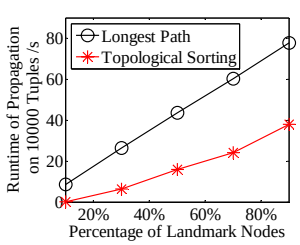


图 8 扩展标签集合

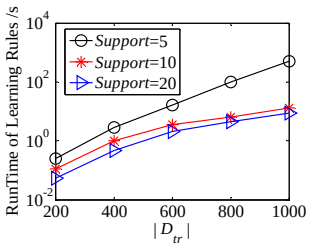


图 9 规则学习时间

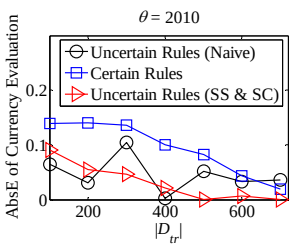


图 10 绝对误差

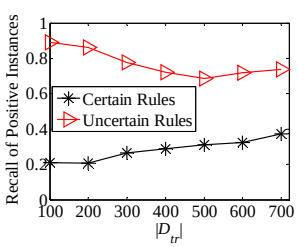


图 11 正召回率

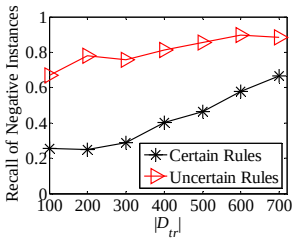


图 12 负召回率

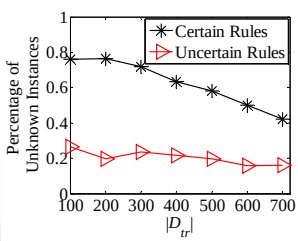


图 13 不可判定率

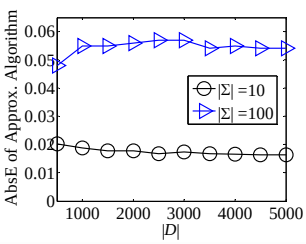


图 14 算法 2 对算法 1 结果的近似

5.2 算法准确度

时效图中无环时算法 2 和算法 1 的执行结果相同,因此,在本节仅讨论算法 1 的准确度,当时效图中有环时,算法 2 对算法 1 结果的近似的准确度的实验结果将在最后一组实验中给出.

我们使用 4 种度量来衡量算法准确度:(1) 绝对误差 *AbsE* ,即根据标准的有效时间得到的 D 的时效性 C_{real} 和使用算法 1 得到的 D 的时效性 C_{LPA} 的差的绝对值 $|C_{real}-C_{LPA}|$;(2) 正召回率 $Recall_{pos}$,即算法 1 能够判定正确的不过时数据项(时效性大于 0)的百分比;(3) 负召回率 $Recall_{neg}$,即算法 1 能够判定正确的过时数据项(时效性小于 0)的百分比;(4) 无法判定率 P_{unk} ,即算法 1 无法判定是否过时(时效性为 0)的数据项占的百分比.在实验中, $|D|=1000$, $\theta=2010$ (θ 取其他值的时候情况类似,不一一列出),用从训练集合中学习到的规则进行时效性判定(*Support*=2).我们将本文提出的不确定时效规则与确定规则进行对比,并分别研究上述 4 种度量的变化.

图 10 所示为 *AbsE* 的变化情况,横轴表示训练集合大小,纵轴表示 *AbsE* 值.标签为 Uncertain Rules(Naive)的黑线表示的实验中,在学习规则时没有使用分层抽样和序列覆盖的策略;标签为 Certain Rules 的线表示的实验中使用了上述两种策略,但只学习了文献[8−10]中使用的确定规则;标签为 Uncertain Rules(SS & SC)的线表示的实验中,使用上述两种策略学习不确定规则.将上述 3 个规则集合用来判定数据时效性,得到的结果如图所示.可以看出,当使用本文提出的不确定规则时,时效性判定结果和标准值之间的误差要小于只使用确定规则的判定误差,同时,在使用了第 4 节提出的两种策略之后,判定误差又有了进一步的缩小.同时,随着 $|D_{tr}|$ 的增大,判定误差逐渐减小,当 $|D_{tr}|=200$ 时,误差已经能够控制在 0.05 以内.图 11~图 13 所示为 $Recall_{pos}$, $Recall_{neg}$, 和 P_{unk} 的变化情况.标签为 Certain Rules 的黑线表示的是仅使用确定规则时的实验结果,标签为 Uncertain Rules 的红线表示的是使用不确定规则的实验结果.可以看出在使用不确定规则时,正负召回率的值高出很多,无法判定的数据项明显更少,这是因为训练集合相同时,不确定规则的覆盖范围更广,因此判定时效性的能力也更强.

接下来讨论算法 2 对算法 1 结果的近似情况.在去掉 G_A 的一些边得 $\widehat{G_A}$ 时,我们扫描一遍 G_A 的结点集合,并交替地删除结点的所有出边或所有入边,这样得到的图中结点要么只有出边,要么只有入边,因此 $\widehat{G_A}$ 必然无环.令 C_{LPA} 为算法 1 在 G_A 上得到的数据集合时效性, C_{TSA} 为算法 2 在 $\widehat{G_A}$ 上得到的数据集合时效性,图 14 展示了 $|C_{LPA}-C_{TSA}|$ 的变化情况,横轴为 $|D|$ 的变化,纵轴为 $|C_{LPA}-C_{TSA}|$ 的变化.可以看出当 $|\Sigma|/|D|$ 很小时, C_{LPA} 和 C_{TSA} 非常接近: $|D|=5000$ $|\Sigma|=10$ 时, $|C_{LPA}-C_{TSA}|$ 仅不到 0.02,误差几乎可以忽略不计.

6 结 论

本文提出了不确定时效规则,用以判定数据的时序关系及数据在给定时刻是否过时失效.基于不确定时效规则建立了数据的时效性模型,能够定量地描述数据项、元组和数据集合的时效性.进一步地,给出了基于时效图的多项式时间的时效性判定算法及基于分层抽样和序列覆盖两种策略的不确定规则的学习方法.真实的数据集上的实验验证了本文方法的准确性和有效性.在未来工作中,我们将从以下几个方面展开研究:(1) 研究同时支持多种策略的时效图标签传播算法;(2) 提高规则学习算法的效率,并研究更有效的抽样方法来进一步缩小规则学习所需的训练集大小;(3) 研究数据的时效性修复问题.

References:

[1] Rahm E, Do HH. Data cleaning: Problems and current approaches. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2000,23(4):3−13.

[2] Eckerson WW. Data quality and the bottom line. TDWI Report, The Data Warehouse Institute, 2002.

[3] Medical News Today. http://www.medicalnewstoday.com/releases/11856.php

[4] Heinrich B, Klier M. Assessing data currency—a probabilistic approach. Journal of Information Science, 2011,37(1):86−100.

[5] Cappiello C, Francalanci C, Pernici B. A model of data currency in multi-channel financial architectures. In: Proc. of the 7th Int'l Conf. on Information Quality (ICIQ-02). Cambridge, MA: 2002. 106−118.

[6] Heinrich B, Klier M, Kaiser M. A procedure to develop metrics for currency and its application in CRM. Journal of Data and Information Quality (JDIQ), 2009,1(1):5.

[7] Zhang H, Diao Y, Immerman N. Recognizing patterns in streams with imprecise timestamps. Information Systems, 2013,38(8): 1187−1211.

[8] Fan W, Geerts F, Wijsen J. Determining the currency of data. ACM Trans. on Database Systems (TODS), 2012,37(4):25.

[9] Fan W, Geerts F, Tang N, Yu W. Inferring data currency and consistency for conflict resolution. In: Proc. of the IEEE ICDE 2013. Brisbane: IEEE, 2013. 470−481.

[10] Li MH, Li JZ, Gao H. Evaluation of data currency. Chinese Journal of Computers, 2012,35(11):2348−2360 (in Chinese with English abstract).

[11] Shortliffe EH, Buchanan BG. A model of inexact reasoning in medicine. Mathematical Biosciences, 1975,23(3):351−379.

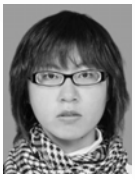
[12] Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to algorithms. 3rd ed., Cambridge: MIT Press, 2001.

[13] Karp RM. Reducibility among combinatorial problems. New York: Springer-Verlag, 1972.

[14] Han J, Kamber M, Pei J. Data mining: Concepts and techniques. 2nd ed., San Francisco: Morgan Kaufmann Publishers, 2006.

附中文参考文献:

[10] 李默涵,李建中,高宏.数据时效性判定问题的求解算法.计算机学报,2012,35(11):2348−2360.



李默涵(1987－),女,河南舞阳人,博士生,主要研究领域为数据质量.

E-mail: limohan.hit@gmail.com

程思瑶(1982－),女,博士,讲师,主要研究领域为传感器网络,CPS.

E-mail: csy@hit.edu.cn

李建中(1950－),男,教授,博士生导师,主要研究领域为海量数据管理与计算,无线传感器网络,CPS.

E-mail: lijzh@hit.edu.cn