软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(2):383−395 [doi: 10.13328/j.cnki.jos.005388]

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

一种基于近似因子的在线概率知识库推理方法*

王艳艳[1,2], 陈 群[1,2], 钟 评[1,2], 李战怀[1,2]

[1](西北工业大学 计算机学院,陕西 西安 710129)

[2](大数据存储与管理工业和信息化部重点实验室(西北工业大学),陕西 西安 710129)

通讯作者: 王艳艳, E-mail: wangyanyan@mail.nwpu.edu.cn

摘 要: 概率知识库中的推理技术是近年来的研究热点.目前,大多数系统的推理主要基于批处理的方式实现,并不适用于在线查询场景.对此,提出了一种基于近似因子的在线概率知识库推理方法.它可以重复利用已推断结果计算查询变量的边缘概率.该算法首先提取查询变量的子图(含已推断变量);然后,在此子图上添加近似因子,以模拟子图外其余变量的影响;最后,采用团树算法推断查询变量的边缘概率.实验结果表明:相对于已有算法,该算法可在时间和精度上取得较好的权衡.

关键词: 概率知识库;在线推理;近似因子;马尔可夫逻辑网

中图法分类号: TP181

中文引用格式: 王艳艳,陈群,钟评,李战怀.一种基于近似因子的在线概率知识库推理方法.软件学报,2018,29(2):383−395. http://www.jos.org.cn/1000-9825/5388.htm

英文引用格式: Wang YY, Chen Q, Zhong P, Li ZH. Online inference based on approximate factors for probabilistic knowledge bases. Ruan Jian Xue Bao/Journal of Software, 2018,29(2):383−395 (in Chinese). http://www.jos.org.cn/1000-9825/5388.htm

Online Inference Based on Approximate Factors for Probabilistic Knowledge Bases

WANG Yan-Yan[1,2], CHEN Qun[1,2], ZHONG Ping[1,2], LI Zhan-Huai[1,2]

[1](School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China)

[2](Key Laboratory of Big Data Storage and Management (Northwestern Polytechnical University), Ministry of Industry and Information Technology, Xi'an 710129, China)

Abstract: The inference techniques for probabilistic knowledge bases have recently attracted significant attentions. In most off-the-shelf existing systems, the inference is mainly implemented based on batch processing and thus not suited for online querying. This paper proposes an online inference approach based on approximate factors for probabilistic knowledge bases, so as to provide a way to reuse those inferred results to calculate the marginal probability for the query variable. In this approach, a subgraph is extracted first, taking the query variable as center; then some approximate factors are attached to simulate the influences from the variables outside the subgraph; and finally, the marginal probability of the query variable is calculated by the clique tree algorithm. Experiments show that compared with existing algorithms, the presented approach can achieve a better tradeoff between accuracy and time.

Key words: probabilistic knowledge base; online inference; approximate factor; Markov logic network

随着信息提取和数据库管理系统的不断发展,构建大规模知识库已成为工业界和学术界的研究热点,典型系统有 Yago[1]、Freebase[2]、Google knowledge graph[3]、DBPedia[4]、NELL[5]等.知识库构建(knowledge base construction,简称 KBC)的主要过程为:从结构和非结构数据源(比如文本、表格、图片)中获取关系型事实.比如,

* 基金项目: 国家重点研发计划(2016YFB1000703); 国家自然科学基金(61332006, 61732014, 61672432, 61472321, 61502390)

Foundation item: National Key Research and Development Program of China (2016YFB1000703); National Natural Science Foundation of China (61332006, 61732014, 61672432, 61472321, 61502390)

收稿时间: 2017-04-10; 修改时间: 2017-05-18, 2017-07-28; 采用时间: 2017-09-05; jos 在线出版时间: 2017-12-01

CNKI 网络优先出版: 2017-12-04 06:46:36, http://kns.cnki.net/kcms/detail/11.2560.TP.20171204.0646.002.html

谷歌知识图谱主要从 Freebase、维基百科以及大规模网页内容中抽取知识,到 2016 年底,它已包含 700 亿条实体关系.这种以结构化形式存储信息的方式能够促进信息的有效处理和查询.

然而,由于信息提取算法固有的概率特性和人类知识的局限性,知识库系统通常存在不确定性和不完全性.因此,通常需要以概率的方式去推断知识库中的缺失事实.而马尔可夫逻辑网(Markov logic network,简称 MLN)[6]是将一阶逻辑(first-order logic)和概率图模型(probabilistic graph model)相结合的一种统计关系型模型,它可用于处理不确定的规则和事实.于是,基于 MLNs 的概率知识库系统运用而生,典型用例有 Deepdive[7]、Elementary[8]、ProbKB[9]等.它们主要采用命题推断的思想,其过程包括两个阶段:1) 实例化(grounding)阶段——基于 MLNs 构建因子图;2) 推断(inference)阶段——在因子图上执行边缘推断.在 Inference 阶段,已有的大多数概率知识库系统主要采用基于马尔可夫链蒙特卡罗(Markov chain Monte Carlo,简称 MCMC)的算法[10](比如吉布斯采样)在整个因子图上执行边缘推断(批量式),即计算因子图中每个变量的边缘概率.

目前,我们正采用 MLN 模型构建商品决策支持系统 Poolside[11].其主要目标是:为用户提供针对性的推荐服务,并可支持含有模糊概念的查询,比如推荐一款性价比高的华为手机.在响应用户的查询时,若采用以上批处理的方式执行边缘推断则耗时较多(商品数目通常较多),而且也没有必要(用户并不可能对所有商品都感兴趣).尽管基于查询驱动的 k-hop 算法[12,13]可加快推理过程,但它很难同时在时间和精度上取得权衡:若跳跃步数 k 取值较大则精度较高,但推理较慢;若跳跃步数 k 取值较小则推理较快,但精度较差.但事实上,随着用户的不断访问,概率知识库中的已推断事实会逐渐增加,若能重复利用它们的推断结果计算其他查询事实的概率,则可在保证精度的情况下降低计算量.

基于以上分析,本文提出了一种基于近似因子的在线推理方法(online inference based on approximate factors),记为 OIAF 算法.OIAF 算法主要包括 3 个步骤:子图提取、添加并估算近似因子和边缘推断.其具体过程为:首先,从因子图中提取查询变量的子图(含已推断变量);然后,为子图中的已推断变量添加近似因子以模拟子图外变量的影响,并基于已推断变量的概率估算近似因子的取值;最后,采用团树算法在含有近似因子的子图上执行边缘推断.本文的主要贡献总结如下.

1) 针对概率知识库的在线查询场景,提出了一种基于近似因子的在线推理算法.它主要利用已推断结果计算查询变量的边缘概率.
2) 在真实和模拟数据集上进行大量实验,进而说明相对于 k-hop 算法,OIAF 算法可在时间和精度上取得较好的权衡.

本文第 1 节综述概率知识库推理技术的相关工作.第 2 节介绍概率知识库的背景知识,包括马尔可夫逻辑网及其推理过程的两个阶段:Grounding 阶段和 Inference 阶段.第 3 节给出本文的研究问题和相关的符号定义,并对 OIAF 算法的工作流程进行介绍.第 4 节通过实验说明 OIAF 算法可在时间和精度上取得较好的权衡.第 5 节对本文内容进行总结.

1 相关工作

目前,概率知识库的推理主要基于马尔可夫逻辑网模型,它已广泛应用在社交网络分析(social network analysis)[14]、实体解析(entity resolution)[15]、信息提取(information extraction)[16]等领域.MLNs 是将一阶逻辑与概率图模型相结合的一种统计关系型模型.它主要包括两种类型的推断方法:命题推断(propositional inference)和合一推断(lifted inference).命题推断主要采用概率图模型的推理算法[17],比如置信传播、变分推断以及马尔可夫链蒙特卡罗等,其主要思想为:基于 MLNs 构建概率图模型(比如因子图、马尔可夫网),然后采用以上方法执行边缘推断.合一推断则基于一阶逻辑推理的思想,主要算法有合一一阶置信传播(lifted first-order belief propagation)[18]、加权一阶模型计算(weighted first-order model counting,简称 WFOMC)[19]等.相对于合一推断,目前,命题推断的应用较为广泛.

本文主要采用命题推断的思想,其过程主要包括两部分:Grounding 阶段和 Inference 阶段.基于它的概率知识库系统主要有 Alchemy[20],Tuffy[21],ProbKB[9],Deepdive[22]等.最初的 Alchemy 系统提出了一系列算法,主要用

于马尔可夫逻辑网的统计关系学习和概率逻辑推断,并且主要在内存中实现.然而 Alchemy 系统的推理过程耗时较多,为此 Tuffy 提出了混合架构的思想:Grounding 阶段在 RDBMS 中实现,Inference 阶段在内存中执行.但 Tuffy 并不适用于含有大规模规则的 MLN,于是,ProbKB 通过利用关系表存储规则的方式实现了推理规则的批量式应用.另外,考虑到概率知识库的推理过程通常是增量式的(随着新数据的到来),Deepdive 提出了增量式构建知识库(incremental knowledge base construction)的方法.然而,以上系统并未支持概率知识库的在线推理.

针对概率知识库的在线推理,文献[12,13]提出了基于查询驱动的 k-hop 算法,但它很难同时在时间和精度上取得权衡.针对以上挑战,本文提出了一种基于近似因子的在线推理方法.

2 概率知识库简介

概率知识库是支持不确定型事实和规则的知识库,是基于马尔可夫逻辑网进行定义的[9].本节主要介绍马尔可夫逻辑网及其推理过程的两个阶段:Grounding 阶段和 Inference 阶段.

2.1 马尔可夫逻辑网

马尔可夫逻辑网[6]是表示不确定事实和规则的数学模型.它由一阶加权子句构成,其权重反映了该一阶子句成立的可能性.MLN 的一个简单例子为

$$3.0\ \ smoke(x)\wedge friends(x,y)\Rightarrow smoke(y),\ \ 2.0\ \ smoke(x)\Rightarrow cancer(x) \tag{1}$$

其中,第 1 个规则蕴含抽烟者的朋友也抽烟,而另外一个规则蕴含抽烟会患癌症.然而,这两条规则并不是绝对成立,权重 3.0 和 2.0 分别用于度量规则成立的可能性程度:权重值越大,说明规则越不容易被违反.下面给出 MLNs 的形式化定义.

定义 1(马尔可夫逻辑网). MLNs 是由一组加权子句$\{(F_i,w_i),i=1,\ldots,n\}$构成的集合,其中,$F_i$ 为一阶子句,而 w_i 为其权重且 $w_i\in\mathbb{R}$.

2.2 Grounding阶段

在概率知识库中,基于 MLNs 构建因子图的过程称为 Grounding.首先介绍因子图的相关知识.因子图由变量节点 $X=\{x_1,x_2,\ldots,x_N\}$和因子节点$\Phi=\{\phi_1,\phi_2,\ldots,\phi_M\}$组成,其中,因子$\phi_i$ 为定义在 $X_i\subset X(X_i$ 为 X 的子集)上的函数,主要用于表示 X_i 中元素之间的关系.在 Grounding 阶段构建的因子图中,变量节点表示概率知识库中的事实,因子节点表示事实之间的因果关系.然后举例说明如何基于 MLNs 构建因子图.图 1 以 Friends & Smokers 为例,简单阐述了 Grounding 的基本过程.

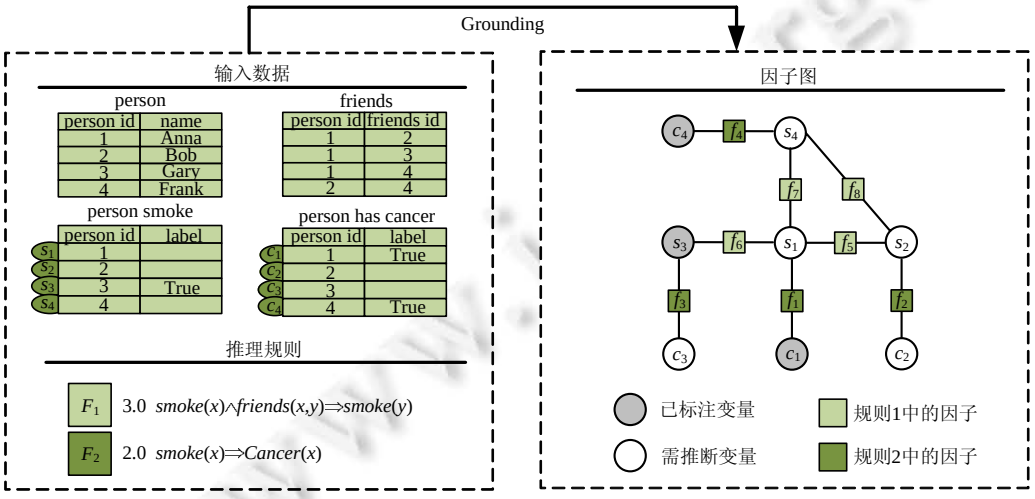


Fig.1 Schematic illustration of grounding

图 1 Grounding 示意图

比如,在规则 F_2"2.0 $smoke(x)\Rightarrow cancer(x)$"中,若 x 取值为 Anna,则可得到 2 个变量节点 $smoke(Anna)$和 $cancer(Anna)$(分别记为 s_1,c_1,如图 1 所示)以及 1 个因子节点 $f_1(s_1,c_1)$,其函数值取决于规则的权重且表示为

$$f_1(s_1,c_1)=\begin{cases}1, & \text{if } (s_1,c_1)=(1,0)\\ \mathrm{e}^{2.0}, & \text{otherwise}\end{cases} \tag{2}$$

依此类推,实例化规则 F_1 和 F_2 中的变量,进而可构建如图 1 右半部分所示的因子图.

下面给出 Grounding 的形式化定义.

定义 2(Grounding). 给定马尔可夫逻辑网 $L=\{(F_i,w_i),i=1,\ldots,n\}$及常量集 C,则规则 F_i 的 Grounding 结果集为 $G(F_i)=\{g=F_i[\bar{a}/\bar{x}]\,|\,\bar{a}\in C\}$(其中,$F_i[\bar{a}/\bar{x}]$ 表示将 F_i 中的自由变量 $\bar{x}$ 用 C 中相应类型的元素 $\bar{a}$ 替代);马尔可夫逻辑网 L 的 Grounding 结果集为 $G(L)=\{(g,w_i)|\exists(F_i,w_i)\in L:g\in G(F_i)\}$,$w_i$ 为命题子句 g 相对应规则 F_i 的权重.

2.3 Inference阶段

Grounding 阶段构建的因子图定义了一个关于随机变量 X 的概率分布[9]:

$$P(X=x)=\frac{1}{Z}\prod_i f_i(X_i)=\frac{1}{Z}\exp\left(\sum_i w_i n_i(x)\right) \tag{3}$$

其中,$n_i(x)=|\{g|g\in G(F_i),x\vDash g\}|$,即为 $G(F_i)$中命题子句成立的数目;Z 为规范化常数.Inference 阶段的主要任务为:基于此概率分布,计算因子图中变量节点的边缘概率.

在因子图中,计算边缘概率的方法主要分为两种类型:精确计算和近似计算.精确计算方法主要有变量消除、团树算法等,而近似计算方法主要有马尔可夫链蒙特卡罗和变分推断等.另外,概率图模型中的精确推理和近似推理都是 NP 难问题[23].

3 OIAF 算法

本节首先给出本文的研究问题和相关的符号定义(第 3.1 节),然后详细介绍 OIAF 算法的基本流程(第 3.2 节),最后分析算法的复杂度(第 3.3 节).

3.1 问题描述及符号定义

问题描述. 给定概率知识库 Grounding 阶段构建的全局因子图和已推断变量,本文的目标是利用已推断结果计算查询变量的概率.

第 3.2 节需涉及到 k-hop 子图的概念(示意图可参见后文图 3),在此给出其形式化定义[13].

定义 3(k-hop 子图). 给定因子图 F 中的因子节点$\{f_i|0\leqslant i\leqslant N\}$、查询变量 v^q 以及跳跃步数 k,若因子节点 $f_i=\{v_{i1}\vee v_{i2}\vee\ldots\vee v_{im}\}$中的元素 $v_{ij}(0\leqslant j\leqslant m)$都满足 $distance(v_{ij},v^q)\leqslant k(distance(p,q)$表示两节点之间需要跳跃的最小步数),那么它将包含在 k-hop 子图中.

3.2 OIAF算法流程

针对以上研究问题,本节提出了 OIAF 算法,其基本流程如图 2 所示.

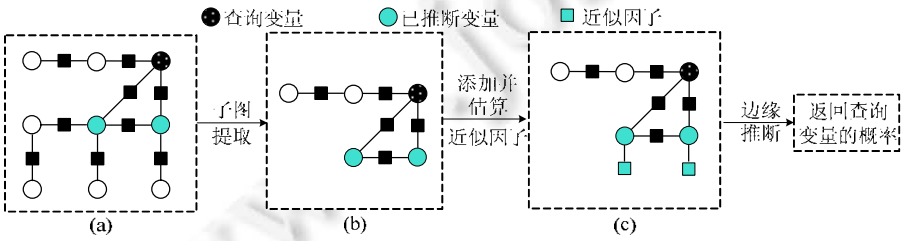


Fig.2 Workflow of OIAF

图 2 OIAF 算法的基本流程

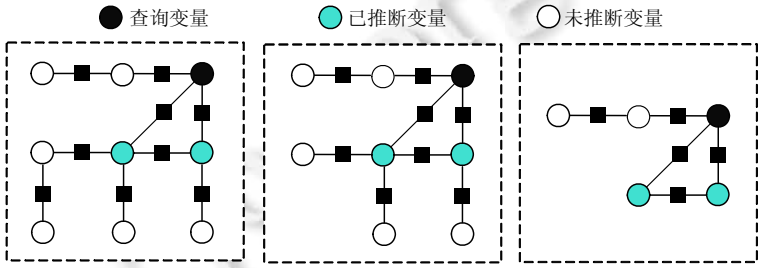
它主要包括 3 个步骤:(1) 子图提取;(2) 添加并估算近似因子;(3) 边缘推断.OIAF 算法首先从全局因子图(如图 2(a)所示)中提取查询变量的子图(含已推断变量,如图 2(b)所示);然后,为子图中的已推断变量添加近似因子以模拟子图外变量的影响(如图 2(c)所示),并基于已推断变量的概率估算近似因子的取值;最后,在含有近似因子的子图(如图 2(c)所示)上执行边缘推断并返回查询变量的边缘概率.下面将详细介绍这 3 部分内容.

3.2.1 子图提取

在因子图中,若假定所有变量都一样,则变量 p 和 q 之间的重要性[13]可表示为

$$I_p(q)=\begin{cases}C, & \text{if } d(p,q)=1\\ C/d^N, & \text{if } d(p,q)>1\end{cases} \tag{4}$$

其中,C 为常数,N 为正实数,$d(p,q)$为变量 p 和 q 之间需要跳跃的最小步数.也就是说,随着两节点之间距离的增大,它们之间的相互影响将会减弱.已有的 k-hop 算法正是基于此思想.它通过在查询变量的 k-hop 子图(如图 3(b)所示)上执行边缘推断,从而实现概率知识库的在线推理.然而,此算法通常存在以下两种情况:(1) 若跳跃步数 k 的取值较大,则推理精度较高,但时间会较长;(2) 若跳跃步数 k 的取值较小,则推理较快,但精度相对较低.对此,本节提出了受约束的 k_{cons}-hop 子图(如图 3(c)所示),它本质上是 k-hop 子图(如图 3(b)所示)剪枝之后的结果.



(a) 全局因子图 (b) 2-hop 子图 (c) 2_{cons}-hop 子图

Fig.3 An illustration of extracting subgraph

图 3 提取子图示意图

k_{cons}-hop 子图的提取原则为:以查询变量为中心,若在搜索路径上(k 跳之内)遇到已推断变量,则搜索停止;否则执行最大跳跃步数 k.虽然受约束 k_{cons}-hop 子图相对于 k-hop 子图较小,但通过利用已推断变量的信息可进一步提高推理的精确性,详见第 3.2.2 节.提取 k_{cons}-hop 子图的具体过程如算法 1 所示.

算法 1. 子图提取.

输入:全局因子图 F,查询变量 v^q,跳跃步数 k.

输出: k_{cons}-hop 子图 F_{sg}.

1. BEGIN
2. F_{sg}=*breadthFirstSearch*(F,v^q,k) //利用广度优先搜索提取子图
3. RERURN F_{sg}
4. END

3.2.2 添加并估算近似因子

若直接在提取的受约束 k_{cons}-hop 子图上执行边缘推断,则会忽略子图之外变量的影响,从而导致误差较大.为了进一步提高推理的精确度,则可借助子图中已推断变量的概率信息.假定子图中已推断变量的概率较为精确,则其在一定程度上反映了全局因子图中变量对它的影响.若通过给 k_{cons}-hop 子图中已推断变量添加近似因子的方式能够使在含有近似因子的子图上推断得到的已推断变量的概率与已知概率相等,近似因子则从一定程度上反映了子图之外变量的影响,这是本文算法的基本原理.在概率知识库中,因子图中的因子往往对应于相应的规则,故添加近似因子相当于增加了相应的规则.另外,我们将含有近似因子(approximate factor)的子图简称为 AF 子图.本文采用非线性方程组的思想求解近似因子的权重(即为近似因子相应规则的权重),下面给出其

形式化表示.

假定提取的 k_{cons}-hop 子图为 $F_{sg}=\{f_i|i=1,\ldots,n\}$、子图中的已推断变量集为 $I_{sg}=\{v_j|j=1,\ldots,m\}$及其相应的概率为 $P_{sg}=\{p(v_j)|v_j\in I_{sg}\}$、添加的近似因子集为 $F_{af}=\{f_j^*(v_j)\,|\,v_j\in I_{sg}\}$,其中,

$$f_j^*(v_j)=\begin{cases}1, & v_j=0\\ \mathrm{e}^{w_j}, & v_j=1\end{cases} \tag{5}$$

可记为 $f_j^*(v_j)=[1,\mathrm{e}^{w_j}]$ (w_j 为相应近似因子的权重且为未知参数),则用于求解近似因子权重的非线性方程组为

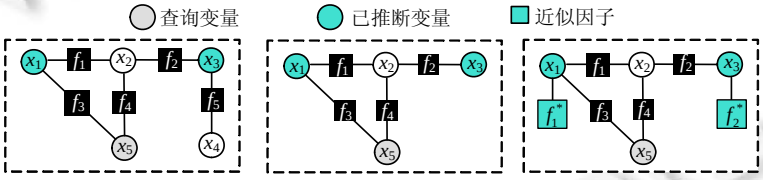
$$\begin{cases}\dfrac{1}{Z}\displaystyle\sum_{V\setminus v_1}\left[\prod_{i,j}f_i f_j^*\right]=p(v_1)\\ \ldots\\ \dfrac{1}{Z}\displaystyle\sum_{V\setminus v_m}\left[\prod_{i,j}f_i f_j^*\right]=p(v_m)\end{cases} \tag{6}$$

其中,Z 为规范化常数,V 为子图中的变量集.此方程组中未知变量的个数取决于已推断变量(近似因子)的数目,且未知变量为$\{w_j\in\mathbb{R}|j=1,\ldots,m\}$(近似因子 $f_j^*(v_j)$ 的取值与 w_j 相关).

下面通过例子说明非线性方程组的具体构建过程.比如图 4(a)为原始因子图,其中,变量 x_1,x_3 为已推断变量,变量 x_5 为查询变量.假定提取的 k_{cons}-hop(k 取值为 2)子图如图 4(b)所示,为了使查询变量 x_5 的推断概率较为精确,分别为已推断变量 x_1,x_3 添加近似因子 $f_1^*(x_1)=[1,\mathrm{e}^{w_1}]$ 和 $f_2^*(x_3)=[1,\mathrm{e}^{w_2}]$ (权重分别为 w_1,w_2)以模拟其余变量的影响(如图 4(c)所示).于是,求解以上近似因子权重的非线性方程组为

$$\begin{cases}\dfrac{1}{Z}\displaystyle\sum_{x_2}\sum_{x_3}\sum_{x_5}f_1(x_1,x_2)f_2(x_2,x_3)f_3(x_1,x_5)f_4(x_2,x_5)f_1^*(x_1)f_2^*(x_3)=p(x_1)\\ \dfrac{1}{Z}\displaystyle\sum_{x_1}\sum_{x_2}\sum_{x_5}f_1(x_1,x_2)f_2(x_2,x_3)f_3(x_1,x_5)f_4(x_2,x_5)f_1^*(x_1)f_2^*(x_3)=p(x_3)\end{cases} \tag{7}$$

其中,$Z=\displaystyle\sum_{x_1}\sum_{x_2}\sum_{x_3}\sum_{x_5}f_1f_2f_3f_4f_1^*f_2^*$.



(a) 原始因子图 (b) 2_{cons}-hop 子图 (c) AF 子图

Fig.4 An illustration of adding approximate factors

图 4 添加近似因子示意图

添加并估算近似因子的具体过程如算法 2 所示.此算法以 k_{cons}-hop 子图 F_{sg} 和子图中已推断变量集 I_{sg} 的概率为输入,输出为 AF 子图 F_{sg}^*. 第 2 行用于初始化近似因子集 F_{af} 和 AF 子图 F_{sg}^*. 第 3 行~第 6 行对 I_{sg} 中的每个已推断变量 v 创建相应的因子函数 $f_v=[1,\mathrm{e}^{w_v}]$ (w_v 为其相应近似因子的权重且为未知参数),并将其添加到 F_{af}. 第 7 行基于已推断变量的概率构建如公式(6)所示的非线性方程组,从而求得近似因子的权重.第 8 行利用计算得到的权重实例化 F_{af} 中的未知参数.第 9 行将实例化后的 F_{af} 添加到 F_{sg}^* 中.第 10 行最终返回 AF 子图 F_{sg}^*.

算法 2. 添加并估算近似因子.

输入:k_{cons}-hop 子图 F_{sg},子图中已推断变量集 I_{sg} 的概率.

输出:AF 子图 F_{sg}^*.

1. BEGIN

2. $F_{af} \leftarrow \varnothing, F_{sg}^{*} \leftarrow F_{sg}$
3. FOR EACH $v \in I_{sg}$ DO
4. $f_v \leftarrow [1, e^{w_v}]$
5. $F_{af}.add(f_v)$
6. END FOR
7. create equations like Eq.(6) and solve it
8. instantiation of the parameters in F_{af}
9. $F_{sg}^{*}.add(F_{af})$
10. RETURN F_{sg}^{*}
11. END

优化方案. 在算法 2 中,我们采用子图中的所有变量联合求解近似因子的权重,但当子图较大或已推断变量较多时,求解过程耗时较多.事实上,在求解近似因子的权重时,子图中变量的影响程度与距离有关:距离较近的变量影响较大,距离较远的变量则影响较小.比如在图 5(a)中,变量 x_2 与近似因子 f_1^* 的距离较近,故它对 f_1^* 的权重求解影响较大;变量 x_6 与近似因子 f_1^* 的距离较远,故它对 f_1^* 的权重求解影响较小.因此在求解某近似因子的权重时,我们可适当忽略子图中与它距离较远的变量,从而加快近似因子权重的求解过程.对此,本节提出了分组求解的优化技术.其基本原理为:以每个近似因子相邻的已推断变量为中心,剪枝掉其 2-hop 子图之外的所有变量(且需保证剪枝之后子图中变量节点的数目不能超过 N_{vars}(通常取 20 即可),否则继续剪枝以满足条件),然后,在剪枝后的子图上求解相应近似因子的权重.另外,若两个近似因子均在同一剪枝后的子图中,则同时求解它们的权重.比如,图 5(b)为求解近似因子 f_1^* 权重的子图,图 5(c)则为求解近似因子 f_2^*, f_3^* 权重的子图.此优化技术的有效性将会在第 4.4 节得到验证.

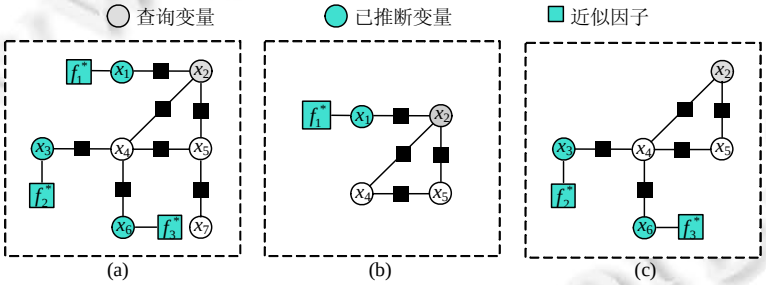


Fig.5 An illustration of grouping

图 5 分组示意图

3.2.3 边缘推断

最后,在 AF 子图上执行边缘推断,进而返回查询变量的边缘概率.目前,大部分概率知识库系统主要采用基于马尔可夫链蒙特卡罗的算法(比如吉布斯抽样)在大规模因子图上执行边缘推断任务.然而当因子图较小时,若想获得较为精确的结果,采样算法通常比精确推理算法(比如团树算法)耗时要多.考虑到本文提取的 AF 子图通常较小,故采用团树算法执行边缘推断.边缘推断的具体过程如算法 3 所示.它以 AF 子图 F_{sg}^{*} 和查询变量 v^q 为输入,输出为查询变量的边缘概率 $p(v^q)$.

算法 3. 边缘推断.

输入:AF 子图 F_{sg}^{*},查询变量 v^q.

输出:查询变量的边缘概率 $p(v^q)$.

1. BEGINE

2. $p(v^q)=CliqueTree(F_{sg}^*, v^q)$
3. RETURN $p(v^q)$
4. END

3.3 复杂度分析

本节主要分析 OIAF 算法的复杂度.不妨设子图提取、添加并估算近似因子和边缘推断的时间复杂度分别为 T_1,T_2,T_3.下面分别对它们进行分析.

(1) 令全局因子图为 F,最大跳跃步数为 k,则利用广度优先搜索提取 k_{cons}-hop 子图 F_{sg} 的时间复杂度为 $T_1=O(b^{k+1})$(https://en.wikipedia.org/wiki/Breadth-first_search)(k 的取值通常较小,默认值为 2),其中,b 为图 F 的分支系数.

(2) 估算近似因子的过程主要包括两部分:构建方程组和求解方程组.令在分组求解中估算近似因子的组数为 n,各组中变量节点的最大阈值为 N_{vars},则构建非线性方程组所需的时间为 $O(n2^{N_{vars}})$,而求解非线性方程组所需的时间为 $O(nm^3)$(采用 MINPACK 中的子程序 HYBRD(http://www.netlib.org/minpack/hybrd.f)进行求解),其中,m 为各方程组中未知变量的最大数目.但当 N_{vars} 给定时,估算近似因子的时间复杂度为 $T_2=O(n(2^{N_{vars}}+m^3))=O(nm^3)$.

(3) 文献[23]指出,概率图模型中的精确推理和近似推理都是NP难问题,故本文所采用的团树算法在最坏情况下需要指数时间.然而,实际中提取的 2_{cons}-hop 子图通常较小,故可有效执行推理.另外,该算法的空间复杂度为 $S=O(b^{d+1}+m^2)$.

4 实验及分析

本文第 4.1 节对实验配置(包括运行环境、数据集等)进行详细说明.第 4.2 节为 OIAF 算法和 k-hop 算法的对比实验.第 4.3 节和第 4.4 节分别说明跳跃步数 k 和分组优化技术对 OIAF 算法性能的影响.

4.1 实验配置

实验所用的编程语言为 Python,且运行环境配置为 Intel(R) Core(TM) i5-6300HQ 2.30GHz 处理器,16GB 内存,Ubuntu 16.04 LTS 64 位操作系统.实验选用了两部分数据集:(1) Poolside 商品知识库[11],主要为用户提供针对性的推荐服务;(2) Friends & Smokers 知识库[18],主要用于社交网络.数据集的相关统计信息见表 1.

Table 1 Statistics of the datasets used in the experiment

表 1 实验中所用数据集的统计信息

数据集	#规则	#变量节点	#因子节点
Poolside	14	2 671	4 086
Friends & Smokers	2	52 096	81 846

在实验中,若选取因子图中的所有变量作为查询变量,则耗时较多且没有必要.于是,本文分别从 Poolside 和 Friends & Smokers 中随机选取 83 和 153 个变量作为查询节点.已有在线推理算法的精度评估标准为:在线推理结果与全局推断结果的绝对误差.但若仅仅将这些查询变量的平均绝对误差作为精确性的评估标准,则很难得知 OIAF 算法(或 k-hop 算法)在低误差和高误差区间上的具体分布情况.于是,我们将误差区间[0,1]切分为 7 部分(如后文图 6 所示):第 1 部分为[0,0.005](低误差区间);最后一部分为(0.03,1](简记为 others,高误差区间);其余部分为(0.005,0.01],(0.01,0.015]),(0.015,0.02],(0.02,0.025]和(0.025,0.03];并分别统计了在各误差区间上查询变量所占的比例.同理,时间区间[0,∞)(单位为 s)被划分为 6 部分:第 1 部分为[0,2](低响应区间);最后一部分为[10,+∞)(简记为 others,高响应区间);其余部分为(2,4],(4,6],(6,8]和(8,10].以上数据集中已推断变量的选取比例均为 20%.另外,考虑到精度和时间之间的相互影响(类似于信息检索中精确率和召回率之间的关系),本文对它们的权衡采取类似的准则:$F_\beta=(1+\beta^2)p_e p_t/(\beta^2 p_e+p_t)$,其中,$p_e$ 为低误差区间[0,0.005]内的百分比(精度的度量),p_t 为低响应区间[0,2]内的百分比.由于相对于精度,在线算法的时间响应更为重要,因此本文选取$\beta=2$.F_2 的取值越大,

说明精度和时间之间的权衡越好.

OIAF 算法的默认设置为 k_{cons}-hop 子图中的跳跃步数 k 为 2,采用分组优化技术求解近似因子的权重.

4.2 OIAF算法 vs. k-hop算法

本节主要通过 OIAF 算法和 k-hop 算法在 Poolside 和 Friends & Smokers 数据集上的对比实验来说明 OIAF 算法可以在精度和时间上取得较好的权衡.

图 6 为两种算法在 Poolside 数据集上误差和时间的对比结果,表 2 为此对比实验在各误差和时间区间内所占百分比的详细信息,表 3 为误差和时间的相关统计信息(比如均值和标准差).

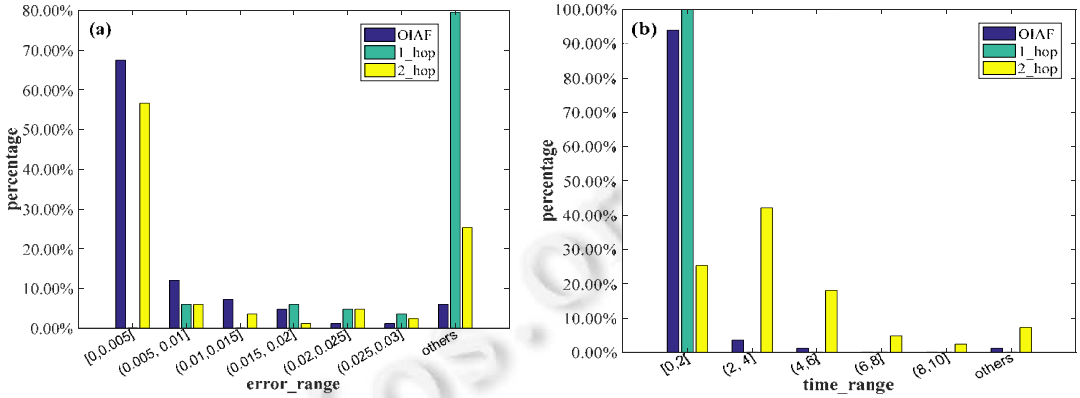


Fig.6 Comparison results of error and time on Poolside (OIAF vs. k-hop)

图 6 Poolside 上各误差和时间的对比结果(OIAF vs. k-hop)

Table 2 Percentage of each error and time interval on Poolside (OIAF vs. k-hop)

表 2 Poolside 上各误差和时间区间内所占的百分比(OIAF vs. k-hop)

误差区间	OIAF (%)	1-hop (%)	2-hop (%)	时间区间(s)	OIAF (%)	1-hop (%)	2-hop (%)
[0,0.005]	67.47	0.0	56.63	[0,2]	93.98	100	25.3
(0.005,0.01]	12.05	6.02	6.02	(2,4]	3.61	0.0	42.17
(0.01,0.015]	7.23	0.0	3.61	(4,6]	1.2	0.0	18.07
(0.015,0.02]	4.82	6.02	1.2	(6,8]	0.0	0.0	4.82
(0.02,0.025]	1.2	4.82	4.82	(8,10]	0.0	0.0	2.41
(0.025,0.03]	1.2	3.61	2.41	Others	1.2	0.0	7.23
Others	6.02	79.52	25.3	–	–	–	–

Table 3 Statistics of error and time on Poolside (OIAF vs. k-hop)

表 3 Poolside 上误差和时间的统计信息(OIAF vs. k-hop)

	统计量	OIAF	1-hop	2-hop
误差	平均值	0.007 2	0.115 5	0.025 3
	标准差	0.012 6	0.098 8	0.042 6
时间	平均值	1.179 4	0.734 4	20.488 1
	标准差	1.568 1	0.182 4	94.325 7

误差对比结果如图 6(a)所示,其横坐标为误差区间,纵坐标为在每个误差区间上查询变量所占的百分比.分析得知:OIAF 算法在低误差区间[0,0.005]上的比例高达 67.47%,而在高误差区间内的比例较低(仅为 6.02%);1-hop 算法则正好相反,它在低误差区间上的比例极低,而在高误差区间上的比例极高(高达 79.52%);2-hop 算法相对于 1-hop 算法,精确性虽然有所提升,但它在低误差区间和高误差区间上的表现均没有 OIAF 算法好.时间对比结果如图 6(b)所示,其横坐标为时间区间,纵坐标为在每个时间区间上查询变量所占的百分比.结果表明:1-hop 算法的推理速度最快(集中在低响应区间[0,2]上的比例为 100%);OIAF 算法的推理速度次之,它在低响应

区间上的比例达到 93.98%;而 2-hop 算法的推理速度相对较慢.分析得知:1-hop 算法的推理速度较快但误差较大;2-hop 算法虽然在准确性上有所提升但推理相对较慢,而 OIAF 算法在时间和精度上的表现均较好.另外,在精度和时间的权衡方面,OIAF,1-hop 和 2-hop 算法的 F_2 值分别为 0.871 3,0.0 和 0.284 5.综合分析可知,OIAF 算法可在时间和精度上取得较好的权衡.

图 7 为两种算法在 Friends & Smokers 数据集上的误差和时间对比结果,表 4 为此对比实验在各误差和时间区间内所占百分比的详细信息,表 5 为误差和时间的相关统计信息(比如均值和标准差).

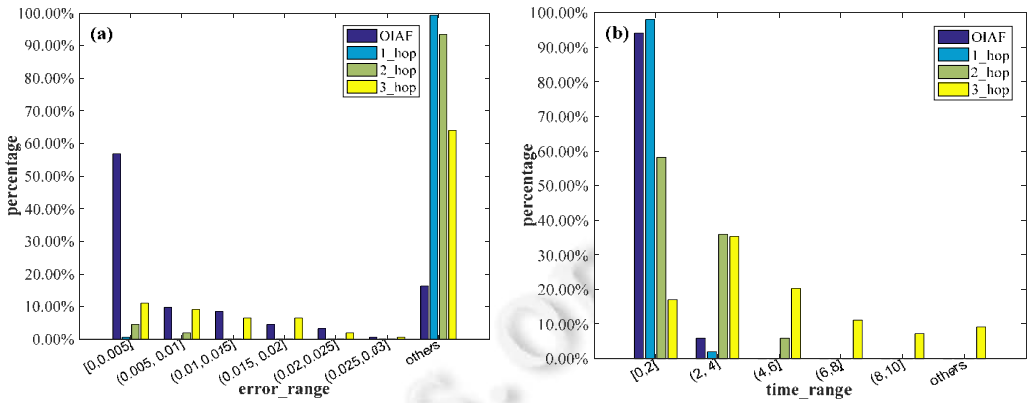


Fig.7 Comparison results of error and time on Friends & Smokers (OIAF vs. *k*-hop)

图 7 Friends & Smokers 上误差和时间的对比结果(OIAF vs. *k*-hop)

Table 4 Percentage of each error and time interval on Friends & Smokers (OIAF vs. *k*-hop)

表 4 Friends & Smokers 上各误差和时间区间内所占的百分比(OIAF vs. *k*-hop)

误差区间	OIAF (%)	1-hop (%)	2-hop (%)	3-hop (%)	时间区间(s)	OIAF (%)	1-hop (%)	2-hop (%)	3-hop (%)
[0,0.005]	56.86	0.65	4.58	11.11	[0,2]	94.12	98.04	58.17	16.99
(0.005,0.01]	9.8	0.0	1.96	9.15	(2,4]	5.88	1.96	35.95	35.29
(0.01,0.015]	8.5	0.0	0.0	6.54	(4,6]	0.0	0.0	5.88	20.26
(0.015,0.02]	4.58	0.0	0.0	6.54	(6,8]	0.0	0.0	0.0	11.11
(0.02,0.025]	3.27	0.0	0.0	1.96	(8,10]	0.0	0.0	0.0	7.19
(0.025,0.03]	0.65	0.0	0.0	0.65	Others	0.0	0.0	0.0	9.15
Others	16.34	99.35	93.46	64.05	–	–	–	–	–

Table 5 Statistics of error and time on Friends & Smokers (OIAF vs. *k*-hop)

表 5 Friends & Smokers 上误差和时间的统计信息(OIAF vs. *k*-hop)

	统计量	OIAF	1-hop	2-hop	3-hop
误差	平均值	0.018 5	0.357 9	0.263 3	0.159 8
	标准差	0.039 3	0.155 8	0.223 1	0.219 1
时间	平均值	1.253 3	0.805 0	2.011 9	5.040 7
	标准差	0.471 9	0.326 8	0.983 6	4.496 0

图 7(a)和图 7(b)表明:相对于 *k*-hop 算法(*k* 分别取 1,2,3),OIAF 算法在低误差区间上的比例明显较高,且推理速度也相对较快(仅慢于 1-hop 算法).另外,在评测精度和时间之间的权衡时,OIAF,1-hop,2-hop 和 3-hop 算法的 F_2 值分别为 0.832 1,0.031 7,0.174 2 和 0.153 6.由此可进一步说明,相对于 *k*-hop 算法,OIAF 算法可在精度和时间上取得较好的权衡.

4.3 跳跃步数*k*对OIAF算法的影响

本节实验主要比较不同跳跃步数 *k* 对 OIAF 算法的影响.在上述实验中,*k* 的默认取值为 2.图 8 测评了 *k* 取不同值(1,2,3)时,OIAF 算法在 Friends & Smokers 数据集上误差和时间的对比结果,表 6 为此对比实验在各误差和时间区间内所占百分比的详细信息.

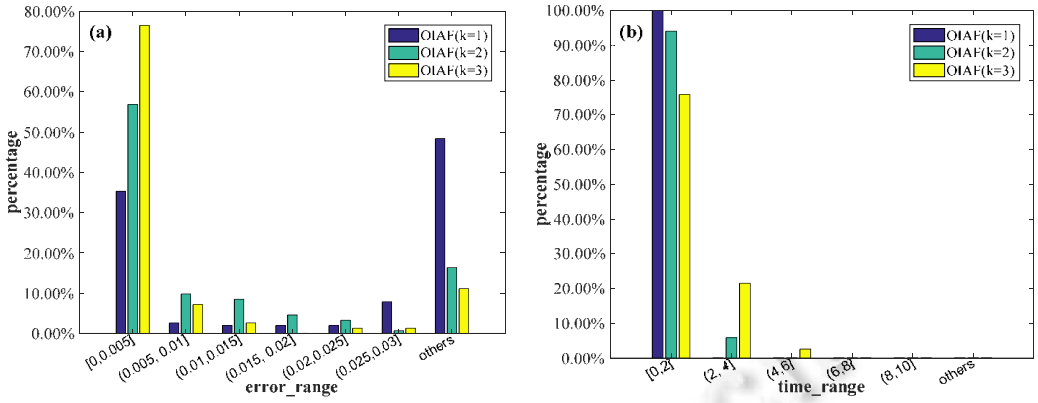


Fig.8 Comparison results of error and time for OIAF (k=1,2,3) on Friends & Smokers

图 8 OIAF(k=1,2,3)在误差和时间上的对比结果(Friends & Smokers)

Table 6 Percentage of each error and time interval for OIAF (k=1,2,3) on Friends & Smokers

表 6 OIAF(k=1,2,3)在各误差和时间区间内所占的百分比(Friends & Smokers)

误差区间	k=1 (%)	k=2 (%)	k=3 (%)	时间区间(s)	k=1 (%)	k=2 (%)	k=3 (%)
[0,0.005]	35.29	56.86	76.47	[0,2]	100	94.12	75.82
(0.005,0.01]	2.61	9.8	7.19	(2,4]	0.0	5.88	21.57
(0.01,0.015]	1.96	8.5	2.61	(4,6]	0.0	0.0	2.61
(0.015,0.02]	1.96	4.58	0.0	(6,8]	0.0	0.0	0.0
(0.02,0.025]	1.96	3.27	1.31	(8,10]	0.0	0.0	0.0
(0.025,0.03]	7.84	0.65	1.31	Others	0.0	0.0	0.0
Others	48.37	16.34	11.11	–	–	–	–

图 8(a)表明:在低误差区间[0,0.005]上,k=1 时比例最低,k=2 时次之,k=3 时最高.图 8(b)表明:整体来看,k=1 时推理速度最快,k=2 时次之,k=3 时推理相对较慢.另外,当 k 取 1,2,3 时,计算得知,OIAF 算法的 F_2 值分别为 0.731 7,0.832 1 和 0.759 5.综合分析得知:当 k=2 时,OIAF 算法在精度和时间上的均衡较好.

4.4 分组求解对OIAF算法的影响

本节实验主要说明分组求解优化技术对 OIAF 算法的影响.图 9 给出了在数据集 Friends & Smokers 上,分组优化技术对 OIAF 算法的影响,表 7 为此对比实验在各误差和时间区间内所占百分比的详细信息.

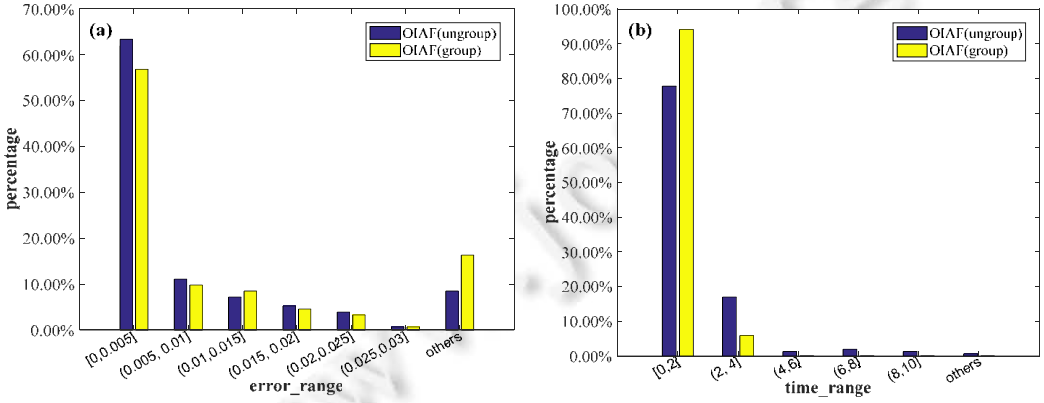


Fig.9 Performance for the OIAF approach with the optimization technique on Friends & Smokers

图 9 分组求解优化技术对 OIAF 算法的影响(Friends & Smokers)

Table 7 Percentage of each error and time interval for OIAF (ungroup/group) on Friends & Smokers

表 7 OIAF(ungroup/group)在各误差和时间区间内所占的百分比(Friends & Smokers)

误差区间	未分组(ungroup) (%)	分组(group) (%)	时间区间(s)	未分组(ungroup) (%)	分组(group) (%)
[0,0.005]	63.4	56.86	[0,2]	77.78	94.12
(0.005,0.01]	11.11	9.8	(2,4]	16.99	5.88
(0.01,0.015]	7.19	8.5	(4,6]	1.31	0.0
(0.015,0.02]	5.23	4.58	(6,8]	1.96	0.0
(0.02,0.025]	3.92	3.27	(8,10]	1.31	0.0
(0.025,0.03]	0.65	0.65	Others	0.65	0.0
Others	8.5	16.34	–	–	–

图 9(a)表明:若采用分组优化技术,则在低误差区间的比例只有稍微下降.但从图 9(b)可以看出,分组优化技术明显加快了推理过程.另外,在权衡精度和时间时,OIAF 算法在未分组和分组情形下的 F_2 值分别为 0.744 0 和 0.832 1.综合分析得知,分组求解对 OIAF 算法具有一定的有效性.

5 总 结

针对概率知识库的在线查询场景,本文提出了一种基于近似因子的在线推理方法.其主要思想为:重复利用已推断结果计算查询变量的概率.该算法通过子图提取和添加近似因子的方式,在含有近似因子的子图上执行边缘推断,进而计算查询变量的概率.相对于已有算法,该算法能够在时间和精度上取得较好的权衡.另外,在对概率知识库进行增量式推理时,需要根据已推断结果来更新节点信息,故本文算法可进一步推广到增量式推理场景中.

References:

[1] Suchanek FM, Kasneci G, Weikum G. Yago: A core of semantic knowledge. In: Proc. of the 16th Int'l Conf. on World Wide Web (WWW 2007). Banff: ACM Press, 2007. 697−706. [doi: 10.1145/1242572.1242667]

[2] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: A collaboratively created graph database for structuring human knowledge. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD 2008). Vancouver: ACM Press, 2008. 1247−1250. [doi: 10.1145/1376616.1376746]

[3] Singhal A. Introducing the Knowledge Graph: Things, Not Strings. Official Google Blog, 2012.

[4] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. Dbpedia: A nucleus for a Web of open data. In: Proc. of the 6th Int'l Semantic Web Conf., 2nd Asian Semantic Web Conf. (ISWC2007, ASWC 2007). Busan: Springer-verlag, 2007. 722−735. [doi: 10.1007/978-3-540-76298-0_52]

[5] Carlson A, Betteridge J, Kisiel B, Settles B, Hruschka ER, Mitchell TM. Toward an architecture for never-ending language learning. In: Proc. of the 24th AAAI Conf. on Artificial Intelligence (AAAI 2010). Atlanta: AAAI Press, 2010. 1306−1313.

[6] Richardson M, Domingos P. Markov logic networks. Machine Learning, 2006,62(1):107−136. [doi: 10.1007/s10994-006-5833-1]

[7] Zhang C. DeepDive: A data management system for automatic knowledge base construction. Madison: University of Wisconsin-Madison, 2015.

[8] Niu F, Zhang C, Ré C, Shavlik J. Elementary: Large-Scale knowledge-base construction via machine learning and statistical inference. Int'l Journal on Semantic Web and Information Systems, 2012,8(3):42−73. [doi: 10.4018/jswis.2012070103]

[9] Chen Y, Wang DZ. Knowledge expansion over probabilistic knowledge bases. In: Proc. of the 2014 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD 2014). Snowbird: ACM Press, 2014. 649−660. [doi: 10.1145/2588555.2610516]

[10] Wick M, Mccallum A, Miklau G. Scalable probabilistic databases with factor graphs and MCMC. Proc. of the Vldb Endowment, 2010,3(1):794−804. [doi: 10.14778/1920841.1920942]

[11] Zhong P, Li ZH, Chen Q, Wang YY, Wang LP, Ahmed MHM, Fan FF. Poolside: An online probabilistic knowledge base for shopping decision support. In: Proc. of the 26th ACM Int'l Conf. on Information and Knowledge Management (CIKM 2017). Singapore: ACM Press, 2017. 2559−2562. [doi: 10.1145/3132847.3133168]

[12] Zhou X, Chen Y, Wang DZ. ArchimedesOne: Query processing over probabilistic knowledge bases. Proc. of the VLDB Endowment, 2016,9(13):1461−1464. [doi: 10.14778/3007263.3007284]

[13] Li K, Zhou X, Wang DZ, Grant C, Dobra A, Dudley C. In-Database batch and query-time inference over probabilistic graphical models using UDA—GIST. The VLDB Journal, 2017,26(2):177−201. [doi: 10.1007/s00778-016-0446-1]

[14] Singla P, Kautz H, Luo J, Gallagher A. Discovery of social relationships in consumer photo collections using Markov logic. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPRW 2008). Anchorage: IEEE Computer Society, 2008. 1−7. [doi: 10.1109/CVPRW.2008.4563047]

[15] Singla P, Domingos P. Entity resolution with Markov logic. In: Proc. of the 6th Int'l Conf. on Data Mining (ICDM 2006). Hong Kong: IEEE Computer Society, 2006. 572−582. [doi: 10.1109/ICDM.2006.65]

[16] Poon H, Domingos P. Joint inference in information extraction. In: Proc. of the 22nd AAAI Conf. on Artificial Intelligence. Vancouver: AAAI Press, 2007. 913−918.

[17] Bishop CM. Pattern Recognition and Machine Learning (Information Science and Statistics). New York: Springer-Verlag, 2006.

[18] Singla P, Domingos P. Lifted first-order belief propagation. In: Proc. of the 23th AAAI Conf. on Artificial Intelligence. Chicago: AAAI Press, 2008. 1094−1099.

[19] Van den Broeck G, Taghipour N, Meert W, Davis J, De Raedt L. Lifted probabilistic inference by first-order knowledge compilation. In: Proc. of the 22nd Int'l Joint Conf. on Artificial Intelligence. Barcelona: IJCAI Press, 2011. 2178−2185. [doi: 10.5591/978-1-57735-516-8/IJCAI11-363]

[20] Kok S, Singla P, Richardson M, Domingos P. The Alchemy system for statistical relational AI. 2007. http://www.cs.washington.edu/ai/alchemy

[21] Niu F, Ré C, Doan AH, Shavlik J. Tuffy: Scaling up statistical inference in Markov logic networks using an RDBMS. Proc. of the VLDB Endowment, 2011,4(6):373−384. [doi: 10.14778/1978665.1978669]

[22] Shin J, Wu S, Wang F, De SC, Zhang C, Ré C. Incremental knowledge base construction using deepdive. Proc. of the VLDB Endowment, 2015,8(11):1310−1321. [doi: 10.14778/2809974.2809991]

[23] Koller D, Friedman N, Wrote; Wang FY, Han SQ, Trans. Probabilistic Graph Model. Beijing: Tsinghua University Press, 2015 (in Chinese).

附中文参考文献:

[23] Koller D,Friedman N,著;王飞跃,韩素青,译.概率图模型.北京:清华大学出版社,2015.

王艳艳(1991－),女,山西吕梁人,博士生,主要研究领域为概率知识库.

钟评(1985－),男,博士生,主要研究领域为数据质量.

陈群(1976－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为大数据管理,物联网信息管理.

李战怀(1961－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据库理论与技术.