

扩展的动态逻辑程序及其在产品个性化推荐中的应用^①

夏 杰, 王 洁, 蔡文君

(北京工业大学 多媒体与智能软件北京重点实验室, 北京 100124)

摘 要: 动态逻辑程序能很好的处理知识库更新问题, 但它不能描述和处理具有偏好的知识更新问题. 因此, 本文在动态逻辑程序的基础上, 提出了一种新的扩展的动态逻辑程序, 它通过对规则头部使用有序析取的方法使其能够描述和处理具有偏好的知识更新问题, 进一步增强了知识的表达和推理能力, 并且定义了其最优回答语义. 同时将这种新的扩展的动态逻辑程序应用于产品推荐系统中, 使用户获得的推荐信息具有个性化特点, 达到个性化推荐的目的. 最后以一个产品个性化推荐实例讨论扩展的动态逻辑程序在产品个性化推荐中的应用.

关键词: 知识库更新; 扩展的动态逻辑程序; 用户偏好; 最优回答集; 个性化推荐

Extended Dynamic Logic Programs and Its Application in Product Personalized Recommendation

XIA Jie, WANG Jie, CAI Wen-Jun

(Key Laboratory of Multimedia and Intelligent Software, College of Computer Science, Beijing University of Technology, Beijing 100124, China)

Abstract: Dynamic logic program is an extension of answer set program for processing knowledge base updated problems, but it cannot describe and deal with knowledge base with preferences updated problems. Therefore, this paper proposes an extended dynamic logic programming language which is based on the dynamic logic program, extending the head of the rule to the form of ordered disjunctive and its optimal answer set semantics is also obtained. The new logic program has more powerful expressiveness and reasoning ability to describe and handle knowledge base with preferences updated problems; however, it does not increase the computational complexity. Applying the method mentioned above to the existing product recommendation system, so that users can get the recommended information with personalized features, to achieve the purpose of the personalized recommendation. Finally the example discusses the extension of dynamic logic program in the application of products personalized recommendation.

Key words: knowledge base update; extended dynamic logic program; user preferences; optimal answer set; personalized recommendation

1 引言

在当今海量信息的互联网中, 信息寻找这个过程乏味, 繁琐且代价昂贵^[1,2]. 特别是在电子商务中, 对商家而言, 产品不能有针对性的提供给潜在买家; 对消费者而言, 不能轻松的得到需要的产品信息, 因此, 推荐系统便成了一种必备的工具, 几乎所有的电商网站都给用户提供了产品推荐信息, 而这些产品信息也是用户所需要的.

推荐系统^[14]已广泛应用在网络商店, 新闻预订等

网络应用中. 推荐系统的目标是为用户提供有价值的信息, 而它的用户模型对其有很重要的影响. 因为用户模型能够代表用户的品味和喜好等信息. 用户模型可以是隐式的, 如用户注册时的一些信息等, 也可以是显示的, 用户对产品信息反馈, 如对产品的投票等. 隐式的用户模型不需要用户的直接反馈信息, 而这也同时带来了模型有效性的问题. 显示模型则相对有效, 因为这是用户需求的直接体现. 怎样处理用户的短期兴趣和长期兴趣是对推荐系统的一项重大挑战.

^① 基金项目:北京市教委基金(05007011200704)

收稿时间:2013-02-25;收到修改稿时间:2013-04-16

目前大多数推荐系基本上都是通过利用用户间的关系来实现推荐,但是把显式的用户模型及其自身更新计算进去的方法关注的是用户自身的动态行为,是一个完全不同的途径,很值得研究.

回答集程序设计(ASP)^[3,4]最初由 Gelfond 等于 1998 年提出,它完全不同于传统的过程程序,程序员通过形式语言描述问题,并使用推理机制发现问题的解,这种程序方式已经成功的应用于设计常识推理等各类问题建模等领域.目前,许多学者致力于对回答集程序进行扩展,以处理概率逻辑程序^[11,12],加权约束逻辑程序^[13]等不确定信息.由于 ASP 具有描述性和非单调语义,特别适用于对常识进行描述和推理,逐渐应用于数据整合,产品配置,医疗以及关于行为的推理等许多方面.

Alferes 在 2005 年提出了动态逻辑程序(DLP)^[5,6]以解决知识库更新问题,但它不能描述和处理具有偏好的知识更新问题.

在本文中,以动态逻辑程序为基础,首先把有序析取引入动态逻辑程序规则的头部,扩展成扩展的动态逻辑程序.然后把显示的用户模型用扩展的动态逻辑程序表示并加入到已存的推荐系统中,以提高推荐质量,达到个性化推荐的目的.在这个显示的用户模型中,用户可以提出自己的需求,偏好等,并且扩展的动态逻辑程序有很好的性质,既其最优回答集正好对应个性化推荐的结果.

2 扩展的动态逻辑程序

Alferes 等在 2005 年提出了动态逻辑程序^[5,6].

动态逻辑程序由多个广义的逻辑程序组成,而广义逻辑程序是回答集程序^[7-9]的扩展.

广义的逻辑程序 P 是规则的头部允许出现缺省文字,形如: $(not) l \leftarrow l_1, \dots, l_i, not l_{i+1}, \dots, not l_m$ 的有限条规则的集合,其中头部为文字 l 或缺省文字 not l, $m \geq i \geq 0$.我们把上述规则叫做广义的规则.

例如:

$$not\ tv-on \leftarrow power-failure \quad (1)$$

$$power-failure \leftarrow \quad (2)$$

把由多个广义的逻辑程序组成的一个程序序列称作动态逻辑程序,记为 $P = (P_1, \dots, P_s)$, 其中 P_i 为一个广义的逻辑程序.对于动态逻辑程序 P,有如下定义:

记 $P = (P_1, \dots, P_s)$ 为一个动态逻辑程序, M 是一个解

释,我们定义:

若规则 r1 与规则 r2 冲突,即 $head(r1) = not\ head(r2)$, 标记为 $r1 \times r2$.

$$Rej(P, M) = \{r \mid r \in P_i, \exists r' \in P_j, i \leq j, r \times r', M \models B(r')\}$$

$$Def(P, M) = \{not\ A \mid \neg \exists r \in \rho(P), H(r) = A, M \models B(r)\}$$

$$M = least(\rho(P) - Rej(P, M) \cup Def(P, M))$$

其中 $\rho(P)$ 是 P 中所有规则的集合, M 是程序 P 的稳定模型.

动态逻辑程序在一定程度上为知识更新问题的描述提供了一种简洁合适的表示方法,但是在知识更新后,多项可选且具有偏好的结果并不能用以上规则来表示.例如:

某个销售盘子和碗的电子商务网站,通过对近几年的盘子和碗的销售数据的静态分析,列出了一些受欢迎的盘子和碗的产品类型.我们把这些产品记做 M, 假设 $M = \{rec(d1), rec(d2), rec(b1), rec(b2), rec(d7), rec(b6)\}$, 其中 d_i 表示第 i 种盘子, b_i 表示第 i 种碗.在初始情况下网站会随机推荐一些碗和盘子给用户.客户持有现金,需要购买盘子和碗,并且希望优先购买盘子,此时知识库有更新,获得的更新知识有两点:客户可支配的钱和选择商品具有偏好.解决上述知识库更新问题的步骤如下:首先,通过用户能支配的钱 S 筛选出价格不大于 S 的商品.然后是选择商品推荐给客户,选择筛选出来的商品有两种情景:(1)筛选出来的商品中只有碗或者盘子,则直接选择,可以用广义的逻辑程序表示.(2)筛选出来的商品中两种商品都有,显而易见广义的逻辑程序规则是不能表达这种情景的.

为了解决情景(2)出现的问题,本文在动态逻辑程序的基础上,提出了一种扩展的动态逻辑程序,即对广义逻辑程序 P_i 进行扩展,在广义的逻辑程序的头部使用有序析取^[10],定义了扩展的动态逻辑程序的稳定集及最优回答集.

扩展的动态逻辑程序由 $P = (P_1, \dots, P_s)$ 组成, P_i 是有序析取的广义逻辑程序,其一般形式为:

$$C_1 \oplus \dots \oplus C_m \leftarrow C_{m+1}, \dots, C_n$$

其中, C_1, \dots, C_n 为客观文字(原子或原子的非)或者缺省文字(前置 not 的客观文字), $n \geq m \geq 0$.

$C_1 \oplus \dots \oplus C_m$ 表示 $C_1 \succ C_2 \succ \dots \succ C_m$, 即文字的优先序列,其中文字 C_i 称为规则的选项.

易知,当 $m=1$ 时,头部仅有一项,此时程序退化

成一般的广义逻辑程序; 当 $m=0$ 时, 头部为空, 规则退化为一个一般的约束规则; 当 $m=n$ 且 $m>1$, 体部为空时, 此规则为头部具有有序析取的事实. 显然, 这种扩展的动态逻辑程序知识表达的能力大大增强, 尤其对多项可选且具有偏好的结果的选择. 如上关于碗和盘子的例子, 有一个客户想买盘子和碗, 但是手上的钱有限只能买其中的一种, 并且这个客户表示若只能买一样, 那么优先买盘子, 用有序析取的广义逻辑程序可表示为:

$$rec(dish) \oplus rec(bowl) \leftarrow \quad (3)$$

下面逐步给出扩展的动态逻辑程序的相关定义.

为了消除程序中的有序析取, 有定义 1-2:

定义 1: 规则 $r = C_1 \oplus \dots \oplus C_n \leftarrow body$. 对于 $k \leq n$, 若规则 r 头部的 $k-1$ 个原子不成立且规则 r 头部的第 k 个原子成立, 则规则 r 化简为如下形式:

$$r^k = C_k \leftarrow body, not C_1, \dots, not C_{k-1}$$

我们把 r^k 称为规则 r 的第 k 个选项. 如规则(1)的第二个选项为 $rec(bowl) \leftarrow not rec(dish)$

定义 2: P 表示一个有序析取的广义逻辑程序, 如果 P' 是通过 P 把中的每个规则替换为它的第 k 个选项得到的, 则 P' 是 P 的一个“分离程序”.

例如, 考虑如下简单程序 P_1 :

$$A \oplus B \leftarrow not C$$

$$B \oplus C \leftarrow not D$$

按定义 2, 程序 P_1 有四个分离程序:

$$A \leftarrow not C \quad B \leftarrow not C, not A$$

$$B \leftarrow not D \quad B \leftarrow not D$$

$$A \leftarrow not C \quad B \leftarrow not C, not A$$

$$C \leftarrow not D, not B \quad C \leftarrow not D, not B$$

通过上述变形, 使包含有序析取的程序转化成一般的逻辑程序.

定义 3: P 表示一个动态逻辑程序 $P = (P_1, \dots, P_s)$, 如果文字的集合 A 是程序 P 的分离程序 P' 的一致个回答集, 则 A 是程序 P 的一个回答集.

根据定义 1,2,3, 程序 P 有如下回答集:

$$M_1 = \{A, B\}$$

$$M_2 = \{B\}$$

$$M_3 = \{C\}$$

上述三个答集并不都是最满足我们的意图的. 很明显回答集 $\{A, B\}$ 对两条规则都给了最优的选择, 而

回答集 $\{B\}$ 对规则 2 给了最优的选择而对规则 1 给了第二选择, 回答集 $\{C\}$ 对规则 2 给了最优的选择. 为了区分回答集的满意度, 我们有定义 4-6:

定义 4: 有序析取的广义逻辑程序的规则为 $r = C_1 \oplus \dots \oplus C_n \leftarrow body$. 对于 $k \leq n$, 定义 C_k 的权值为 $W_{C_k} = k$

例如, 考虑如下程序 P :

$$1) A \oplus B \leftarrow not C$$

$$2) B \oplus C \leftarrow not D$$

在规则 1 中, A 权值为 1, B 权值为 2, 在规则 2 中 B 权值为 1, C 权值为 2.

定义 5: 对于文字集合 S , $W_S = \sum C_i (C_i \in S)$, 表示 S 在程序 P 中的重要程度. 用 S_1 和 S_2 表示两个文字集合, 当且仅当 $W_{S_1} < W_{S_2}$ 时, 我们说文字集 S_1 优先于 S_2 .

如文字集 $S_1 = \{A, B\}$, $S_2 = \{B, C\}$, $S_1 = 1+1=2$, $S_2 = 2+2=4$, 我们认为文字集 S_1 优先于 S_2 .

定义 6: 文字集合 S 是扩展的动态逻辑程序 P 的答集, 规则 $r = C_1 \oplus \dots \oplus C_n \leftarrow A_1, \dots, A_m, not B_1, \dots, not B_k$ 是扩展的动态逻辑程序 P 的规则, 若 $r = C_1 \oplus \dots \oplus C_n \leftarrow A_1, \dots, A_m, not B_1, \dots, not B_k$, 或 $\exists B_i \in S (1 \leq i \leq k)$, 则规则 r 的权值为 $W_r = 1$. 否则 $W_r = 0$. 记所有规则的权重记为 W_R .

定义 7: 记 $W = W_S + W_R$, 当且仅当 S 是程序 P 的一个回答集, 而且不存在程序 P 的回答集 S' 使得 $W' < W$, 文字集合 S 是扩展的动态逻辑程序 P 的优先回答集.

例如程序 P_1 有三个回答集 $M_1 = \{A, B\}$, $M_2 = \{B\}$, $M_3 = \{C\}$, 有

$$W_1 = W_S + W_R = 1+1+0=2$$

$$W_2 = W_S + W_R = 2+1=3$$

$$W_3 = W_S + W_R = 2+1=3$$

则根据定义 7, 最优回答集为 M_1 , 正好是推荐结果, 这也与直观感觉相符.

3 在产品推荐中的应用

近些年, 产品推荐已经成为人工智能领域的一个成果颇多的研究课题, 也成为人工智能技术在商业上的成功应用之一.

用扩展的动态逻辑程序建立用户模型, 把用户模型

和大众推荐结果结合起来形成一个广义的动态逻辑程序, 这个程序得到稳定模型, 进而得到最优答集. 由于用户模型的存在, 这个最优回答集必然是包含用户偏好因素的, 以此提高推荐质量, 达到个性化推荐的目的.

在这一节中我们通过一个实例来展示本文提出的扩展的动态逻辑程序的一些特性. 我们考虑一个在线电影推荐系统, 其数据库中包含了最近几十年的各种电影, 其中通过统计等其它方法得到了一个优秀电影集合, 我们叫它原始的推荐集, 记做 M , 假设我们的原始推荐 M 如下表所示:

表 1 原始推荐电影集合

ID	Title	Year	Geners
15	Cutthroat Island	1995	Action, Adventure
497	Much Ado About Nothing	1993	Comedy, Romance
527	Schindler's List	1993	Drama, War
551	Nightmare Before Christmas, The	1993	Children's, Comedy
589	Terminator 2: Judgment Day	1991	Action, Sci-Fi
1249	Nikita	1990	Thriller
1267	Manchurian Candidate, The	1962	Film-Noir, Thriller
1580	Men in Black	1997	Action, Adventure
1608	Air Force One	1997	Action, Thriller
1856		1998	
1912		1998	
2394	Prince of Egypt, The	1998	Animation, Musical
2396	Shakespeare in Love	1998	Comedy, Romance

现在我们用扩展的动态逻辑程序来建立用户模型.

首先, 用户不想看到有关 drama 类型电影的推荐, 那么可以用如下规则来表示:

$$\text{not rec}(X) \leftarrow \text{genre}(X, \text{Drama}) \quad (4)$$

若用户没有别的需求, 那么

$$P_1 : \text{not rec}(X) \leftarrow \text{genre}(X, \text{Drama})$$

此时扩展的动态逻辑程序 $P = \{M, P_1\}$, 显而易见回答集为

$$M_{R_1} = \{\text{rec}(15); \text{rec}(497); \text{rec}(551); \text{rec}(589); \text{rec}(1249); \text{rec}(1267); \text{rec}(1580); \text{rec}(1608); \text{rec}(1856); \text{rec}(1912); \text{rec}(2394); \text{rec}(2396)\}$$

然后用户需要一些好电影的推荐, 为此他定义了好电影, 即动作, 冒险和魔幻的电影, 并且用户需要至少一部这样电影的推荐:

$$P_2 : \text{good}(X) \leftarrow \text{genre}(X, \text{Action}), \text{genre}(X, \text{Adventure}), \text{genre}(X, \text{Fantasy}).$$

$$\text{rec}(X) \leftarrow \text{good}(X), \text{not n_rec}(X).$$

$$\text{n_rec}(X) \leftarrow \text{good}(X), \text{not rec}(X).$$

$$\text{rec_at_least_one} \leftarrow \text{good}(X), \text{rec}(X).$$

$$\leftarrow \text{not rec_at_least_one}.$$

随后用户定义了一般的电影:

$$P_3 : \text{common}(X) \leftarrow \text{genre}(X, \text{Action}).$$

$$\text{rec}(X) \leftarrow \text{common}(X)$$

并且定义了其它电影:

$$P_4 : \text{else}(X) \leftarrow X$$

$$\text{rec}(X) \leftarrow \text{else}(X)$$

用户想看电影, 选电影的原则是从好到一般到其他, 则用扩展的广义逻辑程序规则表示为:

$$P_5 : \text{good}(X) \oplus \text{common}(X) \oplus \text{else}(X)$$

此时扩展的动态逻辑程序 $P = \{M, P_1, P_2, P_3, P_4, P_5\}$,

根据扩展的动态逻辑程序的回答集求解方法可得出扩展的动态逻辑程序有两个回答集, 分别为: $M_1 = \{15, 589, 1580, 1608, 2394\}$, M_2 为所有电影. M_1 的含义是所有普通电影的集合, M_2 的含义是所有电影的集合. 由于 $2 = W_{M_1} < W_{M_2} = 3$, 所以程序 P 的优先答集为 M_1 , 集合 M_1 作为推荐结果推荐给用户, 即在只有普通电影和其它电影的情况下, 把普通电影推荐给用户, 符合直观意思.

从上例可知, 扩展的动态逻辑程序过滤掉了不相关的信息, 体现了用户个人偏好, 在一定程度上提高了推荐的质量, 对应的推荐结果符合用户偏好.

4 总结及展望

在本文中提出了一种新的扩展的动态逻辑程序, 它通过对规则头部使用有序析取的方法使其能够描述和处理具有偏好的知识更新问题, 进一步增强了知识的表达和推理能力, 并且定义了其最优回答集语义我们提出了扩展的动态逻辑程序. 并且以扩展的动态逻辑程序为基础建立了用户的显示模型, 结合已经存在的推荐系统, 提高了对用户推荐信息的质量.

为了进一步证明本文方法的有效性, 下一步的工

作是把本文的方法集成到一个实验的应用程序中,通过大量的实验数据来进一步验证。

参考文献

- 1 Schafer JB, Konstan J, Riedl J. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 2001(1).
- 2 Hart J, Kamber M. 数据挖掘概念与技术. 北京:机械工业出版社, 2001.
- 3 Gelfond M, Lifschitz V. Logic programs with classical negation. *The 7th International Conference on Logic Programming*, MIT Press, 1990:579–597.
- 4 Gelfond M, Lifschitz V. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing*, 1991, 9(3):365–385.
- 5 Alferes JJ, Banti F, Brogi A, Leite JA. The refined extension principle for semantics of dynamic logic programming. *Studia Logica*, 2005, 79(1).
- 6 Alferes JJ, Leite JA, Pereira LM, Przymusinska H, Przymusiński T. Dynamic updates of non-monotonic knowledge bases. *Journal of Logic Programming*, 2000, 45(1-3).
- 7 Gelfond M, Lifschitz V. The stable model semantics for logic programming. In: Kowalski R, Bowen K, Eds. *Logic Programming: Proc. 5th International Conference and Symposium*, 1988:1070–1080.
- 8 Gelfond M, Lifschitz V. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing*, 1991, 9:365–387.
- 9 Gelder V, Ross A, Schlipf J. The Well-founded Semantics for General Logic Programs. *Journal of ACM*, 1991, 38(3):620–650.
- 10 Brewka G, Niemelä I, Syrjänen T. *Logic Programming with Ordered Disjunction*. *Logics in Artificial Intelligence*, 2002-Springer.
- 11 王洁, 鞠实儿. 概率逻辑程序. *计算机科学*, 2003, 30(7):1–3.
- 12 Wang J, Ju S, Liu C. Agent-oriented probabilistic logic programming. *Journal Computer Science & Technology*, 2006, 21(3):412–417.
- 13 Simons P, Soinen T, Niemela I. Stable model semantics of weight constraint rules. *Proc. of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning*, El Paso, TX, *Lecture Note in Computer Science*. Berlin: Springer, 1999, 1730:317–331.
- 14 Pazzani MJ, Billsus D. Content-based recommendation systems. In Brusilovsky P, Kobsa A, Nejdl W, ed. *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*. 325–341. Springer-Verlag.

(上接第 179 页)

- 5 李虹, 王惠南, 董海艳, 章晓国. 血管内超声图像的血液噪声抑制和对比度增强算法. *华南理工大学学报*, 2009, 39(1): 48–53.
- 6 杨东华, 李久贤, 卞治国. Marr 边缘检测算法的研究. *中国图像图形学报*, 2006, 11(6):823–826.
- 7 王慧锋, 战桂礼, 罗晓明. 基于数学形态学的边缘检测算法研究及应用. *计算机工程与应用*, 2009, 45(9):223–226.
- 8 王昱镔, 雷加印, 毛征, 王亚丽, 吴良. 基于熵和 GVF 的血管内超声图像边缘检测. *国外电子测量技术*, 2008, 27(11): 14–19.
- 9 李毅波, 马东立, 刘中铁, 张朔. 基于三次 B 样条的机翼气动隐身优化设计. *航空计算技术*, 38(4):45–50.