多重极小一般普化*

叶 风 徐晓飞

(哈尔滨工业大学计算机科学与工程系 哈尔滨 150001)

E-mail: yf@mlg.hit.edu.cn

摘要 文章提出一种广义θ-包含意义下的广义最小一般普化,称为多重极小一般普化.这一操作能够有效地减少普化程度,从而使过度普化问题较好地得以解决.为了有效地计算极小一般多重普化,文章研究了示例集上的普化范式与极小一般普化的关系,提出了一种基于概念聚类的归纳学习算法(clustering-based multiple minimum general generalization,简称CMGG).该算法能够有效地产生多重极小一般普化,并准确地反映出学习示例间的内在联系.

关键词 归纳学习,归纳逻辑程序设计,多重极小一般普化,最小一般普化.

中图法分类号 TP181

在归纳逻辑程序设计(inductive logic programming,简称ILP)这一机器学习的热点研究领域中,普化是进行归纳学习的主要操作[1],而最小一般普化(least general generalization,简称LGG)则是一种常用的普化方式,是在包含(θ-subsumption)意义下的子句最小一般化方法[2].学习算法利用LGG产生示例的普化,并将其作为归纳结论.LGG是一种有效的归纳方法.在数据挖掘与知识发现等领域中,LGG都有重要应用[1,3].

LGG方法存在的主要问题是归纳结论的过度普化(over generalization),即所产生的归纳结论覆盖过多的未知事实.归纳结论的覆盖面通常应限制在一定范围内(如已知事实),这就是归纳学习中普遍存在的最小性要求(minimality criteria)[4].归纳的最小性要求使归纳结论最接近示例所蕴涵的逻辑信息,使归纳过程不致过多地引入归纳偏向,这尤其对正例学习是必需的.然而在逻辑蕴含意义下,归纳普化是非常困难的问题,LGG方法在这一意义下是不完全的.现已证明,即使在θ-包含意义下,若不限制归纳结论的搜索空间,不存在能产生真普化、有限且完备的普化算子[5].因此,过度普化问题不存在彻底解决方案.普化必须在一定限制下进行,如,放弃能产生归纳"跳跃"的具有最小真普化性质的归纳算子等.现实可行的解决方案是获取能够有效地降低普化程度的归纳算子.Arimura等人提出的单位子句上的K-最小多重普化算子[6],在特定的范围内,以多单位子句的形式有效地产生简单示例集上的具有较低普化程度的归纳结论.

本文针对上述问题提出广义θ-包含概念,并在这一基础上提出子句集上的多重极小一般普化(multiple and minimum general generalization,简称MGG)的理论与方法,有效地缓解了过度普化问题.本文的结论是对文献[6]的结果的进一步推广.为实现MGG方法,本文引入一种确定子句间相关程度的启发函数,据此给出一种基于概念聚类的算法(clustering-based multiple minimum general generalization,简称CMGG),使MGG方法得以有效实现.实验表明,MGG方法是解决LGG普化问题的有效方案.

1 相关概念

1.1 项、子句与语言

项是构成原子的基本成分.对项的结构及其性质的研究是MGG方法的基础.令A为有穷集,$|A|$表示A的

* 本文研究得到国家863高科技项目基金资助.作者叶风,1960年生,博士生,主要研究领域为机器学习,人工智能逻辑基础,专家系统.徐晓飞,1962年生,博士,教授,博士生导师,主要研究领域为计算机集成制造,分布式数据库.

本文通讯联系人:叶风,哈尔滨150001,哈尔滨工业大学计算机科学与工程系专家系统研究室

本文1998-05-29收到原稿,1998-08-25收到修改稿

基数；有穷集 Σ 表示函数符号集，常数作为零元函数也于其中；X 为与 Σ 不交的变量集.

定义 1. t 称为项，如果：

(1) $t\in\Sigma$，t 为零元函数或 $t\in X$；

(2) 若 $t_1,\ldots,t_n$ 为项，$f^{(n)}\in\Sigma$，则 $f(t_1,\ldots,t_n)$ 为项.

项称为基项，若该项不含变元. T 表示项集，GT 表示基项集. 项上的二元关系 $\leqslant$ 定义为：$t,t'\in T$，$t\leqslant t'$ 当且仅当存在替换 θ 使得 $t=t'\theta$. t 为 t' 的一个例子，t' 为 t 的普化，即 t' 比 t 更为一般. 由项 s 生成的关于项的语言记为 $L_t(s)$，$L_t(s)=\{\omega\mid\omega\in GT,\omega\leqslant s\}$. 由 L_t 的定义知，$s\leqslant s'$ 当且仅当 $L_t(s)\subseteq L_t(s')$. 如果 V 是有穷项集，$V\subseteq L_t(v)$，则称 v 是 V 的普化. 如果对任意 V 的普化 v' 都有 $v\leqslant v'$，则称 v 是 V 的最小一般普化 LGG，记为 $LGG(V)$.

子句及其语言有类似项的定义. 令 P 为谓词符号集，$A(t_1,\ldots,t_n)$ 称为原子公式，如果 $t_1,\ldots,t_n$ 为项，$A\in P$ 为 n 元谓词. 文字是原子公式或其否定. 子句为文字的有穷集合，子句也表示其中文字的析取，其中变元为全称约束. 子句集上的二元关系 $\geqslant_\theta$（θ-包含）定义如下.

定义 2. C,D 为子句，$C\theta$-包含 D，记为 $C\geqslant_\theta D$，当且仅当存在替换 θ 使得 $C\theta\subseteq D$.

C 称为 D 的普化，相应地，D 为 C 的特化，亦称为 C 的一个例子. C 称为 D 的真普化，记为 $C>D$，如果 $C\geqslant_\theta D$ 且 $D\not\geqslant_\theta C$. 由子句 E 生成的语言记为 $L_c(E)$，$L_c(E)=\{F\mid E\geqslant_\theta F\}$. 由子句集 S 生成的语言仍记为 $L_c(S)$，$L_c(S)=\bigcup_{E\in S}L_c(E)$. 如果 U 是有穷子句集，$U\subseteq L_c(u)$，则称 u 是 U 的普化；如果对任意 U 的普化 u'，都有 $u'\geqslant_\theta u$，则称 u 为 U 的最小一般普化，记为 $LGG(U)$. 本文在可区分的场合，对子句和项使用公共术语. 类似地，$C\geqslant_\theta D$ 当且仅当 $LGG(\{C\})\geqslant_\theta LGG(\{D\})$，常把这种情形称为 C 覆盖 D.

1.2 最小一般普化的计算

θ-包含关系下子句的最小一般普化 LGG 是 ILP 中最常用的普化方法，算法由 Plotkin 给出[2]，计算按下述递归方式进行.

项间的 LGG 计算：

(1) $LGG(\{s,t\})=X$，如果 $s=f(s_1,\ldots,s_n)$，$t=g(t_1,\ldots,t_m)$，$f\neq g$，X 是现行计算中未出现的新变量，在后续计算中，项对 $\{s,t\}$ 的 LGG 均以 X 代之；

(2) $LGG(\{s,t\})=f(LGG(\{s_1,t_1\}),\ldots,LGG(\{s_n,t_n\}))$，如果 $s=f(s_1,\ldots,s_n)$，$t=f(t_1,\ldots,t_n)$.

文字间的 LGG 计算：

(3) $LGG(\{p(s_1,\ldots,s_n),p(t_1,\ldots,t_n)\})=p(LGG\{s_1,t_1\}),\ldots,LGG(\{s_n,t_n\}))$，$p$ 为 n 元谓词；

(4) $LGG(\{p(s_1,\ldots,s_n),q(t_1,\ldots,t_m)\})=$无定义，如果 p,q 为不同符号文字.

子句间的 LGG 计算：

(5) $LGG(\{C\})=C$；

(6) $LGG(\{C_1,C_2\})=\{l\mid l_1(C_1,l_2\in C_2,l=LGG(\{l_1,l_2\},LGG(\{l_1,l_2\}$有定义$)\}$；

(7) $LGG(\{C_1,\ldots,C_n\})=LGG(\{C_1,LGG(C_2,\ldots,C_n)\})$.

例 1：$C_1=\{p(a)\leftarrow q(a),q(f(a))\}$，$C_2=\{p(b)\leftarrow q(f(b))\}$.

$LGG(\{C_1,C_2\})=\{p(X)\leftarrow q(Y),q(f(X))\}$；$C_1$ 与 C_2 另有一 LGG 解 $C'=\{p(X)\leftarrow q(f(X))\}$，但在 θ-包含关系下，C 与 C' 等价，因为 $C\geqslant_\theta C'$ 且 $C'\geqslant_\theta C$.

2 多重普化

最小一般普化是在 θ-包含关系下定义的，而不是在逻辑蕴含意义下定义的，这是因为，在计算上，在逻辑蕴涵意义下的最小一般普化计算存在着很大的困难，而 θ-包含下的最小一般普化计算较逻辑蕴涵要容易得多. 但是，LGG 仍存在着过度普化问题.

例 2：表连接的逻辑程序表述是典型的 ILP 学习问题. 现有示例集 $E=\{\{app([],[],[])\},\{app([b],[a],[b,a])\},\{app([a],[],[a])\},\{app([],[a],[a])\},\{app([a,b],[c,d],[a,b,c,d])\}\}$. 按上述 LGG 算法，得到 $LGG(S)=\{app(X,Y,Z)\}$. 然而，若以 $\{app(X,Y,Z)\}$ 为归纳结论，则几乎无意义，因为 $app(X,Y,Z)$ 过于一般，

可以覆盖关于 app 的一切正反例.

如果将 S 进行适当分组,再进行 LGG 普化,则得到:

$LGG(\{\{app([],[],[])\},\{app([],[a],[a])\}\})=\{app([],X,X)\}$;

$LGG(\{\{app([b],[a],[b,a])\},\{app([a],[],[a])\},\{app([a,b],[c,d],[a,b,c,d])\}\})=\{app([A|X],Y,[A|X])\}$.

将两个普化结论合起来便可覆盖 S,这一归纳结论已接近正确的表连接表述,即是本文的二重极小一般普化例.

上例引出进行多重普化的必要性. 较之单普化(LGG),多重普化将显著地降低普化程度,有利于得到满意的归纳结论. 以下内容是有关多重普化的形式讨论. 首先将 θ-包含概念推广到子句集. 令 S,S' 为子句集.

定义 3. S $G\theta$-包含 S',记为 $S\geqslant_{G\theta}S'$,当且仅当对任意 $D\in S'$,都存在 $C\in S$ 及替换 θ,使得 $C\theta\subseteq D$,其中 S 和 S' 都是子句集.

性质 1. 若 S,S' 为单子集,则 $S\geqslant_{G\theta}S'$ 当且仅当 $S'\geqslant_{\theta}S'$.

性质 2. 若 $S\geqslant_{G\theta}S'$,则 $S\models S'$.

因此,S 为 S' 的普化. 本文将 S 称为 k 子句集,如果 S 是由至多 k 个子句构成的集合.

定义 4. k 子句集 S 称为 S' 的 k 重极小一般普化(k-minimum general generalization,简称 k-MGG),记为 k-$MGG(S')$,当且仅当下列条件成立:

(1) $S\geqslant_{G\theta}S'$;

(2) 对任意 k 子句集 T,如果 $T\geqslant_{G\theta}S'$ 且 $S\geqslant_{G\theta}T$,则 $T\geqslant_{G\theta}S$.

性质 3. 若 S 为 k 子句集,则 k-$MGG(S)=S$.

因此,在多重普化中只要不限制普化子句的基数,普化程度在$\geqslant_{G\theta}$关系下就可降到最低,即自身. 多重普化的优越性在于此. 当然,k-$MGG(S)=S$ 这样的解是平凡的. 实用中,学习产生的归纳结论既要覆盖现有事实,又要具有一定的信息压缩能力. 因而 k 要取在一定范围之内.

性质 4. k 重极小一般普化不唯一.

例 3: $S=\{\{p(a,a)\},\{p(a,b)\},\{p(b,b)\}\}$,则 S 的二重极小一般普化解有 3 个:

(1) $S_1=\{p(X,X),p(a,b)\}$;

(2) $S_2=\{p(a,a),p(Y,b)\}$;

(3) $S_3=\{p(a,Z),p(b,b)\}$.

因此,子句集上通常没有多重最小一般普化解.

CS 表示子句的全集,等价关系$\equiv$定义为 $C\equiv D$,当且仅当 $C\geqslant_{G\theta}D$ 且 $D\geqslant_{G\theta}C$. $CS/\equiv$为由关系$\equiv$归纳的商集,$[C]\in CS/\equiv$为子句 C 的等价类,则$\geqslant_{G\theta}$是 CS 上的偏序关系. ILP 中的归纳操作即为确定子句间$\geqslant_{G\theta}$关系的存在与否.

定理 1. $\geqslant_{G\theta}$为 $CS/\equiv$上的偏序关系.

证明:(1) 对任意$[C]\in CS/\equiv$,取替换 $\theta=\{\}$,则 $C\theta\subseteq C$,于是$[C]\geqslant_{G\theta}[C]$;

(2) $[C],[D]\in CS/\equiv$,若$[C]\geqslant_{G\theta}[D]$且$[D]\geqslant_{G\theta}[C]$,由$\equiv$的定义,$[C]\equiv[D]$;

(3) $[C],[D],[E]\in CS/\equiv$,若$[C]\geqslant_{G\theta}[D]$且$[D]\geqslant_{G\theta}[E]$,则对$\forall e\in E$,存在 $d\in D$ 及 θ,使得 $d\theta\subseteq e$,并且对 $\forall d'\in D$,存在 $c\in C$ 及 θ',使得 $c\theta'\subseteq d'$. 不妨取 $d=d'$,则$(c\theta')\theta\subseteq d\theta\subseteq e$. 于是,$[C]\geqslant_{G\theta}[E]$. 因此,$(CS/\equiv,\geqslant_{G\theta})$是偏序集. □

在偏序$\geqslant_{G\theta}$下,S,k-$MGG(S)$与 $LGG(S)$呈现下述关系.

定理 2. $LGG(S)\geqslant_{G\theta}k$-$MGG(S)\geqslant_{G\theta}S$.

证明:只需证明前半部分,后半部分由 k-MGG 的定义直接得到. 若不然,存在子句 $C\in k$-$MGG(S)$,使得对任意替换 θ,$LGG(S)\theta\not\subseteq C$. 对 k-$MGG(S)$中所有这样的 C 作下述替换:$(k\text{-}MGG(S)-\{C\})\cup\{C'\}$,其中 $C'=C\cup LGG(S)$,替换后形成 k 子句集 T. 注意到 $C\geqslant_{G\theta}C'$ 且 $C'\not\geqslant_{G\theta}C$,$C$ 为 C'的真普化. 于是,$LGG(S)\geqslant_{G\theta}k$-$MGG(S)>_{G\theta}T$. 此外,对$\forall e\in S$,$\exists C\in k$-$MGG(S)$及替换 θ,使得 $C\theta\subseteq e$,相应地,$\exists C'\in T$,$C'=C$ 或 $C'=C\cup LGG(S)$,前

者 $C'\theta\subseteq e$；而后者，只需注意 LGG 的定义，存在替换 θ' 使得 $LGG(S)\theta'\subseteq e$，于是，$C'\theta\theta'\subseteq e$. 因此，$T$ 为 S 的普化，$T\geqslant_{G\theta}S$. 但这与 k-$MGG(S)$的极小性矛盾. □

因此，关系$\geqslant_{G\theta}$比$\geqslant_\theta$更强，多重极小一般普化能比最小一般普化更好地降低结论的一般性程度，多重普化的合理性也在此得以体现.

性质 5. $LGG(S)=1-MGG(S)$.

k-MGG 因此也称为广义最小一般普化.

3　普化范式与多重极小一般普化

本节首先研究子句集上的一个重要性质——紧致性，这一性质是所谓普化范式的基础，而普化范式与多重极小一般普化又有密切关系. 利用这种关系可建立基于多重极小一般普化的学习算法.

定义 5. 称 CS 关于集合包含具有紧致性，当且仅当若 $L_c(D)\subseteq\bigcup_{g\in G}L_c(g)$，则存在 $g\in G$ 使得 $L_c(D)\subseteq L_c(g)$. 其中 $D\in CS$，$G\subseteq CS$.

CS 上的紧致性是计算 k-MGG 的基础.

定义 6. 项 t 的层数递归定义如下：

(1) t 为 1 层的，如果 t 为常量或变量；

(2) t 为 $n+1$ 层的，如果 $t=f(t_1,\ldots,t_n)$，$n=\mathrm{Max}(\{t_1$ 的层数，$\ldots,t_n$ 的层数$\})$.

引理 1. 若 $|\Sigma|>k>0$，则 CS 关于集合包含具有紧致性. 其中 Σ 是构造 CS 的函数符号集.

证明：$L_c(D)\subseteq\bigcup_{g\in G}L_c(g)$. 令 $|\Sigma|=m$，$\Sigma=\{f_1,\ldots,f_m\}$，$D$ 是子句，$G=\{g_1,\ldots,g_k\}\subseteq CS$. 施归纳于 D 中项的层数. 归纳基始：D 中项的最大层数为 1，则 D 中项为常量或变量.

(1) 若 D 中项均为常量，则 $L_c(D)=\{D\}\subseteq\bigcup_{g\in G}L_c(g)$. 当然，存在 $g\in G$，使得 $D\in L_c(g)$；

(2) 否则，D 中项有一个为变量，不妨设只有一个变量且为 X，则 $L_c(D)=L_c(\{\{D\{X/f_1\}\},\ldots,\{D\{X/f_m\}\})$，其中$\{X/f_i\}$为替换. 即将 Σ 中各函数(包括常量)以最一般的形式代入. 由于各 f_i 不同名，可将 f_i 看成常量. 类似于(1)，对各 $D\{X/f_i\}$都存在一个 $g_i\in G$ 及替换 θ_i，使得 $g_i\theta_i\subseteq D\{X/f_i\}$. 因为 $m>k$，由抽屉原理，必有一个 $g\in G$ 覆盖 D 的两个例化，分别为 $D\{X/f_i\}$与 $D\{X/f_j\}$，$D\{X/f_i\}\in L_c(g)$，$D\{X/f_j\}\in L_c(g)$. 由于 $f_i\neq f_j$，故 g 中对应的能覆盖 f_i 与 f_j 的项必为变量. 于是，通过对这个变量的各种可能替换，g 能覆盖所有 D 的例化，即 $D\{X/f_i\}\in L_c(g)$. 因而 $L_c(D)=\{D\{X/f_1\}\},\ldots,(D\{X/f_m\})\subseteq L_c(g)$.

归纳假设：当 D 中项的层数至多为 n 时，结论成立. 下面证明当 D 中项的层数至多为 $n+1$ 时，结论仍成立. 不妨考虑 D 中只出现一个层数为 $n+1$ 的项 $t=t'\left[\begin{matrix}f(t_1,\ldots,t_n)\\ i\end{matrix}\right]$，表示 t 由层数为 n 的项 t' 构造而得，方法是将 t' 中位置为 i 处的变量代之以项 $f(t_1,\ldots,t_n)$，其中 $f\in\Sigma$，$t_1,\ldots,t_n$ 为变量或常量.

(3) 若 $t_1,\ldots,t_n$ 均为常量，则将 $f'=f(t_1,\ldots,t_n)$作为一个新的常量加入 Σ 中(这种做法不改变 $L_c(D)$等的内容)，同时将 t 改为 $t'\left[\begin{matrix}f'\\ i\end{matrix}\right]$. 于是，$D$ 中项的最大层数为 n 且 $|\Sigma|=m+1>k$，由归纳假设，结论成立；

(4) 否则 $t_1,\ldots,t_n$ 之一为变量，不妨设 t_1 为唯一变量. 将 t_1 分别替换成 f_i，$k\geqslant i\geqslant 1$，形成相应的 D_i，并将 k 个 $f(f_i,t_2,\ldots,t_n)$作为新的常量加入 Σ 中. 因而，D_i 中项的最大层数为 n 且 $|\Sigma|=m+k>k$. $L_c(D)=L_c(\{D_1\},\ldots,\{D_m\})=L_c\{D_1\}\cup\ldots\cup L_c\{D_m\}$. $\forall i,1\leqslant i\leqslant m$，$L_c\{D_i\}\subseteq\bigcup_{g\in G}L_c(g)$，由归纳假设，存在 $g\in G$，使得 $L_c\{D_i\}\subseteq L_c(g)$. 应用抽屉原理并以类似(2)的方法，得到结论. □

下面我们均假定 $|\Sigma|>k$，因为这一假定符合 ILP 学习的实际情况.

定义 7. 子句集 S，T. S 称为 T 的普化范式，当且仅当对$\forall C\in S$，$C=LGG(T-L_c(S-\{C\}))$.

普化范式一方面说明了普化结论的极小性，另一方面指出了普化结论中的各子句对覆盖示例集的独立贡献.

定理 3. 设 S，T 是子句集，$|S|=k$，$T\subseteq L_c(S)$，则 S 是 T 的 k 重极小一般普化当且仅当 S 是 T 的普化范式.

证明：必要性. S 是 T 的普化范式. 若不然，存在 $C\in S$，使得 $C\neq LGG(T-L_c(S-\{C\}))$. 令 $C_0=LGG(T-$

$L_c(S-\{C\})$),则由 C_0 的最小性,$L_c(C_0)\cap T\supset L_c(C)\cap T$ 或 $L_c(C)\cap T\supset L_c(C_0)\cap T$. 于是,前者导致 $T\not\subseteq L_c(S)$,与 $T\subseteq L_c(S)$矛盾;而后者导致 $T\subset L_c(S)$,与 S 的极小性矛盾.

充分性. S 是 T 的 k 重极小一般普化. 若不然,另有 k 子句集 S',使得 $T\subseteq L_c(S')$,$S\geqslant_{GB}S'$,$S'\not\geqslant_{GB}S$,即 $S>S'$. 于是,$T\subseteq L_c(S')\subset L_c(S)$.

(1) 若 $|S'|<|S|$,由引理 1,对$\forall\ C'\in S'$,$L_c(C')\subseteq\bigcup_{C\in S}L_c(C)$,存在 $C\in S$,使得 $L_c(C')\subseteq L_c(C)$. 取子句集 $S''=\{C|C\in S$,存在 $C'\in S'$使得 $L_c(C')\subseteq L_c(C)\}$,$|S''|=|S'|$. 由此,$T\subseteq L_c(S')\subseteq L_c(S'')$,$S''$覆盖 T 且为 S 的真子集. 任取 $C\in(S-S'')$,则有 $T-L_c(S-\{C\})=\varnothing$. 于是,$LGG(T-L_c(S-\{C\}))\neq C$,即 S 不是 T 的普化范式;

(2) 若 $|S'|=|S|$,由引理 1,对$\forall\ C'\in S'$,$L_c(C')\subseteq\bigcup_{C\in S}L_c(C)$,存在 $C\in S$,使得 $L_c(C')\subseteq L_c(C)$. 再由假设 $S>S'$ 知,必存在 $C\in S$,$C'\in S'$,使得 $L_c(C')\subset L_c(C)$. 然而,$L_c(S'-\{C'\})\subseteq L_c(S-\{C\})$. 因此,$T-L_c(S-\{C\})\subseteq T-L_c(S'-\{C'\})$,此外,因为 $T\subseteq L_c(S)$,$T-L_c(S-\{C\})\subseteq L_c(C)$. 同理,$T-L_c(S'-\{C'\})\subseteq L_c(C')$. 于是,$T-L_c(S-\{C\})\subseteq T-L_c(S'-\{C'\})\subseteq L_c(C')\subset L_c(C)$. 由此,$C\neq LGG(T-L_c(S-\{C\}))$,这与 S 为普化范式矛盾. □

上述定理指出,要获得 k 重极小一般普化,只需将示例集 T 作适当的 k 类划分,形成 k 个子集,再对每个子集作 LGG,k 个 LGG 形成覆盖 T 的普化范式,也是 T 的 k 重极小一般普化. 因此,k 重极小一般普化问题转化为对示例集的合理 k 类划分问题.

4 多重极小一般普化算法

由第 3 节讨论可知,产生 k 重极小一般普化的关键是进行合理的 k 类划分. 然而,就归纳学习而言,我们仅知道学习示例,而对目标概念的深层知识并不知道,因而难于得到划分的标准. 在这种情况下,必须给出归纳结论的适当语义. 归纳结论的一个明显表现是其反映示例的聚类特征,每一聚类表示目标概念的一个子概念,相应于示例划分中的一个子集. 由此,我们可以作出这样的假定:即归纳结论具有聚类意义下的语义. 据此,本文基于多重极小一般普化的学习算法的基本思路是:首先对示例集 E 进行适当的聚类(k 类),聚类结果形成 E 的一个 k 类划分,然后对 E 划分中的各子集作 LGG,并形成 k 子句集作为 k 重极小一般普化的近似.

为进行概念聚类,必须定义一种相似性测度,这种测度应能准确反映示例间的相似性或差异性. 为此,考察一下 LGG 计算的定义,不难发现,$LGG(E)$实际捕捉了示例间的公共结构与特征,反映了示例间的共性. 因此,可将 LGG 作为元素间相似性的基准. 相似性的对立面是差异性. 对于给定的两个子句 C 与 D,它们的差异主要来自以下几个方面:同名文字间对应位置上项的差异、子句中文字数量的差异、不同名文字数量的差异以及子句中变量间限制的差异等(在此,变量限制是指一个变量在子句中出现两次以上. 如在子句 $P(X)\leftarrow Q(X)$中,变量 X 是限制的,而 X 在子句 $P(X)\leftarrow Q(Y)$中是不限制的. 这有别于变量的全称约束). 根据以上 4 种差异,定义子句间差异的函数 Dif.

首先定义变量在子句(项)中所处深度的概念,变量深度的不同决定了这一变量对子句的影响不同. 一般地,变量所处位置越小,其影响越大.

定义 8. X 为一变量名,tc 为子句或项. X 在 tc 中的深度 $d(X,tc)$递归定义为:

(1) $d(X,tc)=0$,如果 tc 是与 X 同名的变量;

(2) $d(X,tc)=\infty$,如果 tc 是与 X 不同名的变量;

(3) $d(X,tc)=1+\min\{d(X,t_i)|t_c=f(t_1,\ldots,t_n),1\leqslant i\leqslant n\}$;

(4) $d(X,tc)=1+\min\{d(X,t_i)|tc-(\neg)P(t_1,\ldots,t_n),1\leqslant i\leqslant n,tc$ 为文字$\}$;

(5) $d(X,\{\})=\infty$;

(6) $d(X,tc)=1+\min\{d(X,l_i)|tc=\{l_1,\ldots,l_n\},1\leqslant i\leqslant n,tc$ 为子句$\}$.

例如,变量 X 在 $C=\{Even(s(X))\leftarrow Odd(X)\}$中的深度为 1,而 X 在 $C'=\{Even(s(s(X)))\leftarrow Odd(s(X))\}$的深度为 2. 明显地,在 θ-包含意义下,X 在 C 中起的作用要大于 X 在 C'中起的作用.

定义 9. C,D 为子句,C 与 D 的差别函数 Dif 定义为

$$Dif(C,D)=k\left(\sum_{X/t\in\theta_1}\frac{1}{d(X,LGG(C,D))}+\sum_{X/t\in\theta_2}\frac{1}{d(X,LGG(C,D))}\right)$$
$$+k'(|C-LGG(\{C,D\})\theta_1|+|D-LGG(\{C,D\})\theta_2|).$$

其中 θ_1 与 θ_2 为替换，使得 $LGG(\{C,D\})\theta_1\subseteq C$ 和 $LGG(\{C,D\})\theta_2\subseteq D$，$X/t$ 表示替换 θ 中的一个代换项，$|E|$ 表示子句 E 中文字个数，k 与 k' 是两个可调参数，用于确定两类差异的权重.

在 Dif 定义中，第 1 项 $\left(\sum_{X/t\in\theta_1}\frac{1}{d(X,LGG(C,D))}+\sum_{X/t\in\theta_2}\frac{1}{d(X,LGG(C,D))}\right)$ 表达了 C 和 D 各自与公共结构 $LGG(\{C,D\})$ 间项的差异，也隐含了限制变元间的差异，这种差异主要体现在替换 θ_1 与 θ_2 的各个代换项上；第 2 项 $(|C-LGG(\{C,D\})\theta_1|+|D-LGG(\{C,D\})\theta_2|)$ 表达了 C 和 D 各自与公共结构 $LGG(\{C,D\})$ 间文字数量的差异与不同名文字数量的差异.

性质 6. (1) $Dif(C,C)=0$；(2) $Dif(C,D)=\infty$，如果 $LGG(\{C,D\})=\{\}$.

下面利用 Dif 给出基于概念聚类的多重极小一般普化算法 CMGG.

(1) 输入子句集 $T=\{C_1,\ldots,C_n\}$，k

(2) 循环 直至 $|T|\leqslant k$ do

(3)　　取 $C,D\in T,C\neq D$，使得 $Dif(C,D)=\min\{Dif(A,B)|A\in T,B\in T,A\neq B\}$；

(4)　　对所有 $A\in T$，如果 $LGG(\{C,D\})\geqslant_\theta A$，则 $T=T-\{A\}$；

(5)　　$T=T\cup LGG(\{C,D\})$；

(6) 输出 T. // k 重极小一般普化结论.

性质 7. 算法 CMGG 输出输入子句集上的多重极小一般普化结论.

算法 CMGG 以最邻近规则进行聚类. 下例说明算法的执行过程，取 $k=k'=1$.

例 4：$T=\{\{app([\,],[\,],[\,])\},\{app([b],[a],[b,a])\},\{app([a],[\,],[a])\},\{app([\,],[a],[a])\},\{app([a,b],[c,d],[a,b,c,d])\}\}$. 考虑 E 的二重极小一般普化.

(1) 首次循环，$\{app([\,],[\,],[\,])\}$ 与 $\{app([\,],[a],[a])\}$ 为最近邻，$Dif(app([\,],[\,],[\,]),app([\,],[a],[a]))=2$，形成 $T=\{\{app([b],[a],[b,a])\},\{app([a],[\,],[a])\},\{app([a,b],[c,d],[a,b,c,d])\},\{app([\,],X,X)\}\}$；

(2) 二次循环，$\{app([b],[a],[b,a])\}$ 与 $\{app([a,b],[c,d],[a,b,c,d])\}$ 为最近邻，$Dif(app([b],[a],[b,a]),app([a,b],[c,d],[a,b,c,d]))=5$，形成 $T=\{\{app([A|B],C,[A|D])\},\{app([\,],X,X)\}\}$.

算法有效地得到 T 的二重极小一般普化 $\{\{app([A|B],C,[A|D])\},\{app([\,],X,X)\}\}$. 注意到在首次循环，$\{app([\,],[\,],[\,])\}$ 与 $\{app([a],[\,],[a])\}$ 也是一对最近邻，它们的 LGG 是 $\{app(X,[\,],X)\}$，这样可形成另一种二重极小一般普化.

算法 CMGG 在以最邻近规则进行聚类时，有时也难免生成过于普化的结论. 如果在算法的第(3)步再考虑 LGG 相对于示例集的覆盖面因素，这一问题就能得到较好的处理. 将算法的第(3)步改为：取 $C,D\in T,C\neq D$，使得 $Dif(C,D)+Cov(LGG(\{C,D\}))=\min$，其中 $Cov(LGG(\{C,D\}))=|\{e|e\in T,e\in L_e(LGG(\{C,D\}))\}|$，即综合考虑子句差别与 LGG 结论的普化程度.

例 4 得到的归纳结论 $\{app([A|B],C,[A|D]),app([\,],X,X)\}$ 已经非常接近正确的表连接表述. 注意到算法 CMGG 得到的每一部分归纳结论实质都表征了一类示例的结构特征，如 $app([\,],X,X)$ 表示 app 第 1 元为常量 $[\,]$，第 2 元与第 3 元相同的示例，$app([A|B],C,[A|D])$ 表示 app 第 1 元与第 3 元的头元素相同的示例. 利用这些结构特征能够有效地产生相当一些类问题的正确归纳结论，特别是单位子句程序类等. 利用例 4 产生的结构，我们能够容易地构造出这一问题的最终表述：$\{app([\,],X,X),app([A|B],C,[A|D])\leftarrow app(B,C,D)\}$.

例 5：关于自然数乘法的示例集 $E=\{\{mul(0,1,0)\},\{mul(0,2,0)\},\{mul(1,1,1)\},\{mul(1,2,2)\},\{mul(1,4,4)\},\{mul(2,2,4)\leftarrow dec(2,1),mul(1,2,),plus(2,2,4)\},\{mul(3,1,3)\leftarrow dec(3,2),mul(2,1,2),plus(2,1,3)\}\}$. 取 $k=3$.

(1) 算法形成 $\{\{mul(0,1,0)\},\{mul(0,2,0)\}\}$ 的聚类 $mul(0,A,0)$；

(2) 算法形成{{*mul*(1,1,1)},{*mul*(1,2,2)},{*mul*(1,4,4)}}的聚类 *mul*(1,*B*,*B*).

若不考虑 *LGG*(*C*,*D*)结论的普化程度因素,下一次将对 *mul*(0,*A*,0)与 *mul*(0,*B*,*B*)进行聚类,产生 *mul*(*X*,*Y*,*Z*),结果与 LGG 方式一样. 而在考虑 *LGG*(*C*,*D*)结论的普化程度因素后,有

(3) 算法对{{*mul*(2,2,4)←*dec*(2,1),*mul*(1,2),*plus*(2,2,4)},{*mul*(3,1,3)←*dec*(3,2),*mul*(2,1,2),*plus*(2,1,3)}}进行聚类,形成 *mul*(*C*,*D*,*E*)←*dec*(*D*,*F*),*mul*(*F*,*D*,*G*),*plus*(*G*,*D*,*E*).

本例中 *E* 取的是自然数乘法的逻辑程序{*mul*(0,*A*,0). *mul*(0,*B*,*B*). *mul*(*C*,*D*,*E*)←*dec*(*D*,*F*),*mul*(*F*,*D*,*G*),*plus*(*G*,*D*,*E*).}的基例化子集. 目的是考察算法 CMGG 的普化性能. *E* 所取示例是一种"代表集"[7],示例集给出了进行归纳所必须的示例,在"代表集"上进行多重极小一般普化,可取得满意结果. 本文提出的算法还对一批典型示例进行了多重普化学习,如自然数 *plus*(+),*lesseg*(≤)等,均获满意结果.

5 结 论

子句集上的多重极小一般普化是对最小一般普化归纳的直接推广. 本文首先证明了多重普化能够有效降低归纳结论的一般性程度,从而使多重极小一般普化成为一种适合的归纳方法. 然后,证明了子句集上的 *k* 重极小一般普化等价于该集上的 *k* 普化范式. 由此,通过引入关于子句间差异的启发函数 Dif,提出了一种基于概念聚类方法的多重极小一般普化算法 CMGG. 实验表明,该算法准确地形成了令人满意的多重极小一般普化归纳结论.

参考文献

1 Muggleton S, Raedt L D. Inductive logic programming: theory and method. Journal of Logic Programming, 1994, 19(20): 629～679

2 Plotkin G G. A note on inductive generalization. In: Meltzer B, Michie D eds. Machine Intelligence. Edinburgh University Press, 1970, (5): 153～163

3 Dzeroski S. Inductive logic programming and knowledge discovery in databases. In: Fayyad U M, Shapiro G, Smyth P *et al* eds. Advances in Knowledge Discovery and Data Mining. Cambridge, CA: AAAI Press, 1996. 117～152

4 Wrobel S. First order theory refinement. In: DeReadt L ed. Advances in Inductive Logic Programming. Amsterdam: IOS Press, 1996. 14～33

5 Patrick R J, Nienhuys-Cheng S. Existence and nonexistence of complete refinement operators. In: Bergadano F, Raedt L D eds. Proceedings of the 7th European Conference on Machine Learning. Lecture Notes in Artificial Intelligence. Berlin: Springer Verlag, 1994. 307～322

6 Arimura H, Shinohara T, Otsuki S *et al*. A generalization of the least general generalization. In: Furukwa K, Michie D, Muggleton S eds. Machine Intelligence. Oxford: Clarendon Press, 1994, (13): 59～85

7 Ling C X. Logic program synthesis from good examples. In: Muggleton S ed. Inductive Logic Programming. London: Academic Press, 1992. 113～127

Multiple Minimum General Generalization

YE Feng XU Xiao-fei

(*Department of Computer Science and Engineering Harbin Institute of Technology Harbin* 150001)

Abstract In this paper, the authors present a kind of generalized least general generalization, called MGG (multiple minimum general generalization), under generalized θ-subsumption. MGG does effectively reduce the generalization of inductive hypotheses to extent, such that the problem of over-generalization is satisfactorily overcome. For computing MGG efficiently, the relation between normal generalization and MGG is studied and an algorithm CMGG (clustering-based multiple minimum general generalization) based on concept clustering is proposed, which can effectively figure out MGG and reflect accurately the internal relation of the set of learning examples.

Key words Inductive learning, inductive logic programming, multiple minimum general generalization, least general generalization.