文法推断研究的历史和现状*

张瑞岭

(中国科学院软件研究所计算机科学开放研究实验室 北京 100080)

E-mail: zrl@ox.ios.ac.cn

摘要 文法推断属于形式语言的归纳学习问题,它研究如何从语言的有限信息出发,通过归纳推断得到语言的语法定义.文章综述了文法推断研究的历史和现状.首先阐述文法推断的理论模型,接着罗列上下文无关文法类及其非平凡子类、隐马尔可夫模型以及随机上下文无关文法的推断方法,最后简介文法推断的应用,并展望其发展趋势.

关键词 示例学习,归纳推断,形式语言的学习,文法推断.

中图法分类号 TP18

文法推断(grammatical inference)属于形式语言的归纳学习问题.语言的归纳学习致力于研究如何从语言的有限信息出发,通过归纳推断得到语言的定义.当待学习语言的最终定义用不同的形式表示时,其归纳学习过程也有所不同.严格地说,文法推断是指待学习语言的定义用文法表示时的归纳学习过程.

文法推断从语言的有限信息(输入)出发,通过一个归纳推断过程,最终得到语言的文法描述(输出).其中输入信息一般包含一个正例样本集,或同时包括一个反例样本集,还可能包含其他附加信息.文法推断问题的提出,可追溯到20世纪50年代末,所以,相对于计算机科学中某些研究领域而言,文法推断是一个历史悠久的研究领域;同时,由于文法推断问题本身固有的复杂性,其发展还不够成熟,还有大量的理论和技术问题需要人们去研究和探索,所以它又是一个新兴的研究领域.

在文法推断研究的最初20年内,特别是在1970~1980年间,取得了丰硕的研究成果.但从80年代初开始,有关文法推断问题的研究论文越来越少,原因不在于该问题的研究已很成熟,而是由于文法推断问题本身固有的困难,使其研究处于低迷状态.近几年,许多研究者试着将其他一些技术,如遗传算法和神经网络运用到文法推断中,特别是将概率统计理论结合到文法推断中,使文法推断进入实际应用成为可能,于是文法推断的研究又活跃起来,并从1993年起召开了4届关于文法推断专题的国际会议.其他一些专题会议中也包含了文法推断的研究成果,如类比及归纳推断、算法学习理论和计算学习理论,另外,在ACM关于计算理论的年度会议中也包含部分相关论文.

到目前为止,涉及文法推断问题的综述性文章主要有文献[1~4].本文旨在介绍文法推断的理论模型和方法,特别介绍当前的研究热点和趋势.在介绍研究成果时特别列出有关上下文无关文法及其非平凡子类文法(即正规文法的超类)以及随机文法的推断方法.

1 有关定义

一个上下文无关文法(context-free grammar,简称CFG)G用一个四元组表示,即

$$G=(N,\Sigma,P,X),$$

其中N和Σ分别为非终极符集合和终极符集合,P为产生式集合,X为开始非终极符.字母表$V=N\cup\Sigma$.Σ上

* 本文研究得到国家自然科学基金、国家863高科技项目基金和国家"九五"高科技攻关项目基金资助.作者张瑞岭,1969年生,博士生,主要研究领域为形式规约,机器学习.

本文通讯联系人:张瑞岭,北京100080,中国科学院软件研究所计算机科学开放研究实验室

本文1998-09-29收到原稿,1999-03-15收到修改稿

串集合(包括空串)记为 Σ^*,空串用 ε 表示.$\Sigma^+=\Sigma^*-\{\varepsilon\}$.$V$ 上的终极符和非终极符串组成的集合(包括 ε)记为 V^*,$V^+=V^*-\{\varepsilon\}$.一个串 α 的长度记为 $|\alpha|$.P 中每个产生式形如 $A\to\beta$,这里 $A\in N,\beta\in V^*$.我们用 $\alpha\Rightarrow\beta$ 表示 $\exists\ \gamma,\delta,\eta,A$ 满足 $\alpha=\gamma A\delta,\beta=\gamma\eta\delta$ 且 $A\to\eta$ 是一个产生式.$\alpha\overset{*}{\Rightarrow}\beta$ 表示$\exists$ 串 $\alpha_0,\alpha_1,\ldots,\alpha_m(m\geqslant 0)$满足

$$\alpha=\alpha_0\to\alpha_1\Rightarrow\ldots\Rightarrow\alpha_m=\beta.$$

这里,序列 $\alpha_0,\ldots,\alpha_m$ 称为从 α 到 β 的一个推导.文法 G 产生的语言记为 $L(G)$,即

$$L(G)=\{s\mid s\in\Sigma^*\},X\overset{*}{\Rightarrow}s\},$$

也就是对 $L(G)$中任一实例串 s,存在从 X 到 s 的一个推导,该推导过程可以用一个树表示,该树就称为 G 关于 s 的语法树(或称派生树),派生树的根结点为开始非终极符 X,所有叶结点连接起来即为 s.派生树的形状(即不考虑其内部结点的标记)称为 s 的轮廓(skeleton).推导过程也可以用依次使用的产生式序列表示,即一个实例 s 的推导过程记为 $D(s)=(r_1,\ldots,r_m)$,r_i 为 P 中产生式.

若一个 CFG G 的产生式集合 P 中的所有产生式形如 $A\to a$ 或 $A\to aB,A,B\in N,a\in\Sigma$,则称 G 为正规文法(regular grammar).一个语言 L 是正规语言,当且仅当存在一个正规文法 G,有 $L=L(G)$.同时,一个语言 L 是正规语言,当且仅当存在一个确定的有限状态自动机(deterministic finite automata,简称 DFA)A,其接受语言 $L(A)=L$.在下面的描述中,正规文法、正规语言和 DFA 等名词可互相替换.

对于一个语言 L,若终极符串 $s\in L$,则称 s 为 L 的正例;若 $s\notin L$,则称 s 为 L 的反例.语言 L 的正例样本集 S^+是由 L 的有限个正例组成的集合;语言 L 的反例样本集 S^-是由 L 的有限个反例组成的集合.

下面是一些 CF 文法子类的定义.

(1) G 是线性文法(linear grammar)当且仅当其产生式形如 $A\to uBv$.$A\to u$,其中 $A,B\in N,u,v\in\Sigma$.G 是偶线性文法(even linear grammar,简称 ELG)当且仅当 G 是线性文法,且形如 $A\to uBv$ 的产生式满足$|u|=|v|$.

(2) G 是简单确定文法(simple deterministic grammar)当且仅当其产生式形如 $A\to a\alpha$,其中 $a\in\Sigma,\alpha\in N^*$,且$|\alpha|\leqslant 2$;同时,若 $A\to a\alpha$ 和 $A\to a\beta\in P$,则有 $\alpha=\beta$.

(3) G 是极简单文法(very simple)当且仅当 G 为 Greibach 范式文法,且对每个终极符 $a\in\Sigma$,P 中正好有一个形如 $A\to a\alpha(\alpha\in V^*)$的产生式.极简单语言类是简单确定语言类的真子集.

本文中出现的形式语言中有关概念的定义和进一步解释见参考文献[5].

2　文法推断的理论模型

文法推断的研究可以追溯到 20 世纪 50 年代末.最初由 Chomsky[6]于 1957 年提出.但首先系统而严格地陈述这一研究课题的是 Gold.他于 1969 年在文献[7]中给出文法推断的经典理论模型,即语言的极限识认模型.

2.1　语言的极限识认模型

Gold 把文法推断问题抽象成这样一个过程:在任何时刻 t,向推断设备输入信息单元 i_t,推断设备输出一个猜测(即推断结果)$H(i_1,\ldots,i_t)$.$H(i_1,\ldots,i_t)$表示从信息单元 $i_1,i_2,\ldots,i_t$ 所得到的猜测结果.若在有限时间之后,推断设备输出的猜测不因提供更多的信息单元而改变,且该猜测是关于待学习语言的正确描述,则可以说推断设备使用的是一个成功的推断算法,这个过程也称为"语言的极限识认".这里,$i_1,i_2,\ldots$ 称为样本序列(或训练序列).设待学习语言为 L,L 的正例表示(positive presentation)序列是指其样本序列中任一信息单元 $i_k(k=1,2,\ldots)$为 L 的正例串;L 的完整表示(complete presentation)序列是指其样本序列中任一 i_k 为一个二元组$(w,b)\in\Sigma^*\times\{0,1\}$,若 $b=1$,则 w 为 L 的正例,否则 w 为 L 的反例.

定义. 设 $\mathscr{L}$ 为一递归可枚举语言类,推断设备 ID 从正例表示(或完整表示)序列极限识认 $\mathscr{L}$ 是指,$\forall L\in\mathscr{L}$ 和 L 的任一正例表示(或完整表示)序列 $I=i_1,i_2,\ldots$,在任意时刻 t,ID 接受 i_t 并输出猜测 $H_t=H(i_1,\ldots,i_t)$,这里,H 是可计算的,存在一个时刻 k,有 $h_k=h_{k+1}=\ldots$,且 h_k 是对 L 的正确描述.

定理(Gold). 设 L 为任一无穷语言(即包含无穷多个句子),C_F 为有限语言类,则从正例表示序列不能极限识认语言类 $C_F\cup\{L\}$;从完整表示序列可以极限识认任何可枚举语言类.

Gold 定理告诉我们,如果只提供正例,连正规语言也不能极限识认.这是一个令人沮丧的结论,但这一点也

很好理解.因为对于无穷语言,如果仅提供正例,一旦某个时刻 t,推断设备输出的猜测 h_t 是待学习语言 L 的超集,即 h_t 描述的语言 $L'\supset L$,则在随后的时间里,猜测 h_t 会始终不变,而且不会有其他信息告诉推断设备去修正其猜测.这种情况又称为过分泛化(overgeneralization).

2.2 交互式学习模型

Gold 的结论使人很自然地想到,除了提供正例样本外,还得提供别的有用信息.Gold 提出的反例信息是一种方法,但很多场合反例很难提供,而且包含反例信息后,加大了问题的复杂性[8].Angluin 提出一种在多项式时间内通过提问进行语言正确识认(exact identification using queries in polynomial time)的模型[9,10].该模型中包括教师和学习者两个角色,其中教师负责回答学习者(即推断系统)提出的关于待学习语言 L 的有关问题,典型的提问类型有两种.

(1) 成员问题(membership).推断设备提供一个句子串 w,教师回答该串是否为待学习语言 L 的合法句子;

(2) 等价问题(equivalence).推断设备提供一个文法 G',教师回答该文法所产生的语言是否等价于待学习语言 L.若回答"no",则同时提供一个反例串 w,即 $w\in L(G')$ 但 $w\notin L$,或 $w\in L$ 但 $w\notin L(G')$.等价问题的实质是回答当前推断结果 G' 是否为最终结果.

Angluin 在文献[11]中证明,如果只使用等价问题,CF 语言类不可能在多项式时间内识认;更进一步地,Angluin和 Khariton 在文献[12]中证明,如果使用成员问题和等价问题,CF 文法类识认问题的难度与通常的解密问题相当,到目前为止尚未找到多项式时间的算法.

2.3 近似学习模型

最近几年,真正得到广泛应用的是 Valiant 的近似学习模型(probably approximately correct learning,简称 PAC)[13].在该模型中,输入样本随机地从 Σ^* 中选取,Σ^* 的概率分布 D 确定但未知.推断过程的成功与否由正确性参数 ε 和信心参数 δ 这两个参数衡量.这两个参数也同样作为输入提供给推断算法.一个成功的推断算法将以高概率(至少 $1-\varepsilon$)寻找到具有最小错误率(小于 δ)的文法.

Valiant 的概率模型得到广泛的应用,特别是在语音识别、信息检索和机器翻译等领域.在这些应用场合,语言由随机文法表示.对于一个通常文法,如果赋予每个产生式相应的概率,就得到一个随机文法.一个随机文法会相应地赋予其产生的每个串一个概率,从而定义了串集合的概率分布.与上下文无关文法相对应的是随机上下文无关文法(stochastic CFG,简称 SCFG);与有限状态自动机相对应的则是隐马尔可夫模型(hidden Markov model,简称 HMM).在后面的章节中将分别介绍相应于这些表示类的推断算法.

3 CF 文法类及其子类的推断方法

抽象地说,推断设备进行推断时包含两个子过程,即猜测生成和猜测选择.所谓猜测即为可能的推断结果,比如对 CF 文法推断来说,一个猜测就是一个 CF 文法.其中猜测生成方法可以笼统地分为枚举法和构造法两类.枚举法[14,15]基于这样一个事实,即大部分重要的文法子类能有效地进行枚举.枚举法通过对所考虑的文法类中的文法进行枚举,由此构造猜测空间.在文献[15]中以自定义的文法复杂度的顺序进行上下文无关文法的枚举.其文法形式可以是 Chomsky 范式、Greibach 范式或算符文法.为提高效率,它对每个文法作预处理,以避免花费过多的时间去处理那些不可能被接受的文法.这类文法包括与已处理过的某文法的结构相同(两个文法结构相同是指文法的非终结符之间可建立对应关系)、无效文法(如从文法的某一非终结符出发,无法生成终结符串,或出现循环推导,如 $X\Rightarrow\ldots\Rightarrow X, X\in N$)等等.在文献[14]中以结构包含关系为偏序关系,进行文法枚举.对 CFG G_1 和 G_2,文法 G_1 结构包含于文法 G_2 是指 G_1 产生的语法树形状集合是 G_2 的相应集合的子集.枚举法的优点是直观、彻底(因为其本质是穷尽所有可能解,所以从理论上可以找到最优解);其致命的弱点是计算复杂性较高,很难用于实际问题的解决.因此,枚举法文法推断的研究较少,本文不再作进一步介绍.

构造法是对样本进行系统化的分析,从样本实例中所包含的结构信息,构造出文法规则.通过枚举法或构造法生成猜测空间后,下一步就是在猜测空间中进行搜索,即推断设备从众多猜测中选出与当前样本集合协调的猜测(最优或较优猜测).Angluin 在其综述性文章[2]中列举了一些搜索方法.这些方法可能相互间有类似之处,

它们的思想都是采用各种手段，尽可能快地找出最优或较优猜测. 比如，启发式方法是根据一系列的启发规则进行抉择，这些启发规则通常与待推断文法类的性质，或者与具体的应用领域有关. 在此方法中，猜测空间中的猜测通常具有某种偏序关系，如通用性. 比如文法 G_1 比文法 G_2 通用性弱是指 $L(G_1)\subset L(G_2)$. 于是，如果一个猜测 H 不能包含某些正例，则所有比 H 通用性弱的猜测就不予考虑.

以上所述推断方法只是对抽象推断过程的分类，在实际方法中，猜测生成和猜测选择过程可能融合在一起或交替地反复进行. 下面，我们介绍一些具体的文法推断方法，并且侧重于应用广泛且较难对付的文法类，即上下文无关文法类或其非平凡子类.

3.1　从自然实例进行推断

所谓自然实例就是指通常的句子串，它是相对于下面提到的结构化实例而言的. Solomonoff 于 1959 年在文献[16]中提出一种文法推断方法，其思想是根据 CF 语言的“子字重复”性质，对每个正例样本 $w\in S^+$，首先从 w 中删除某个子串后得到串 w'，询问 w' 是否为 L 的实例；若是，再将删除的子串重复多次后插入 w' 中得到串 w''，再询问 w'' 是否为 L 的实例，若是，则可推断出一个递归规则. 比如，若多个形如 a^nb^n 的串是 L 的实例，则 L 的文法中包含产生式形如 $A\to aAb$. 很显然，该方法对 S^+ 的选择要求较高. 若 S^+ 太小，则会遗漏很多重要的递归规则；若 S^+ 太大，则计算复杂性令人不能忍受；同时该方法对某些 CF 文法是推断不出来的，如 $A\to aAb|cAd$[1]. 事实上，Solomonoff 并未给出严格的推断算法，更谈不上算法的实现，但他给后来的研究者一些启发.

另一个从自然实例推断的例子是 B. Knobe 和 K. Knobe 提出的方法[17]，这个方法实际上是一系列启发式方法的大杂烩，其计算复杂性较高. 还有一个例子是 Tanatsugu 提出的构造性算法[18]，该算法从正例和反例出发，进行 CF 文法推断，它包括 3 个过程. 首先从正例样本中抽出自嵌套(self-embedding)结构；然后从给定的样本中推断出线性文法；最后用一定的方式从线性文法构造出 CF 文法. 文献[18]中认为该算法适用于整个 CF 语言类，但未分析其计算复杂性. 本文认为，该算法的主导方法是枚举，其实质非常类似于 Solomonoff 的方法.

为了研究如何获取形式规约以及在构造新规约时如何复用已有规约，董韫美提出了一个基于复用的上下文无关文法推断方法[19~21]. 该方法同样以自然实例作为推断出发点，其主要特点是：(1) 在进行复杂文法推断(即待推断的语言规模较大)时可复用已知语言；(2) 推断出的文法结构自然. 复用的思想大大降低了文法推断的复杂度.

3.2　从结构化实例进行推断

基于 CF 文法推断问题的复杂性，许多作者在研究 CF 文法推断问题时，往往在提供样本实例的同时，附加额外的信息.

Crespi-Reghizzi 在文献[22]中介绍了一种方法，即从带括号的实例出发推断算符优先文法[23]. 对一个 CFG G，将 G 中每个产生式 $A\to\alpha$ 代之以 $A\to(\alpha)$ 得到文法 G'，G' 生成的实例串称为 G 的带括号实例串.

带括号实例串中包含了实例串的轮廓信息. 我们知道，对 CFG 来说，其文法结构决定其实例串的派生树形状，反过来说，其所有实例的派生树形状决定其文法结构. 因此，对推断算法来说，最有效的信息就是实例串的轮廓(即派生树形状)信息. 我们把这种将轮廓信息包含在内的实例串称为结构化实例.

可以证明，一个 CFG G 的所有实例串对应的派生树组成的集合(记为 $DT(G)$)是一个合理的树集合，所谓合理是指这个树集合可以由某个树自动机识别，而且即使 $DT(G)$ 中各派生树的内部结点未被标识，该集合同样是合理树集合[24]. 基于此，从结构化串推断 CF 文法的问题，可以归结为树自动机的识认问题.

Angluin 在文献[9]中提出一个有效的推断算法，通过等价问题和成员问题识认确定的有限自动机(DFA). Sakakibara 在文献[24]中通过将 Angluin 对 DFA 的推断算法推广到树自动机，从而表明，从结构化样本实例出发，CF 语言类可以通过结构化成员问题和结构化等价问题在多项式时间内识认.

3.3　CF 文法子类的推断

鉴于 CF 文法推断问题的复杂性，许多研究者将注意力转向 CF 文法子类的研究，而且取得了可喜的成果.

Ishizaka 在文献[25]中给出关于简单确定文法的推断算法，同时表明，通过等价问题和成员问题，简单确定语言可以在多项式时间内识认.

Yokomori 在文献[26]中研究了简单确定语言的真子类——极简单语言的学习,得出这样的结论:极简单语言类可以从正例出发,在多项式时间内识别.

Takada 在文献[27]中表明,ELG(偶线性文法)的推断问题可以归结为 DFA 的推断问题,并给出了一个实现归结过程的多项式时间算法.作者认为,将一种文法类的推断问题归结为另一种相对简单的文法类的推断问题是一个自然且有效的方法.所以对 Takada 的方法不妨多费些笔墨.

Takada 的基本思想如下:设 F 为一个字母表,对于一个 ELG $G=(N,\Sigma,P,X)$,把 P 中每个产生式用 F 中的字母唯一标志.设 $f=f_1f_2\ldots f_k\in F^*$,则 $X\overset{*}{\Rightarrow}_f w$ 表示从 G 的开始非终极符 X 开始,依次使用标志为 f_1,f_2,…,f_k 的产生式,派生出串 w.设 C 为字母表 F 上的一个语言,文法 G 在控制集 C 下产生的语言定义为

$$L_C(G)=\{w\in\Sigma^*\mid X\overset{*}{\Rightarrow}_f w, f\in C\}.$$

于是,控制集 C 中一个句子对应着 $L_C(G)$ 中一个句子的派生过程.Takada 证明:任何一个字母表 Σ 上的偶线性语言 L,可以由 Σ 上的一个通用 ELG G^0 在一个控制集 C 下产生,且 C 为正规语言,即对 Σ 上任一 ELL L,有

$$L=L_C(G^0)=\{w\in\Sigma^*\mid X^0\overset{*}{\Rightarrow}_{f_C} w, f\in C\}.$$

其中 C 为一正规语言,G^0 的定义为

$$G^0=(\{X^0\},\Sigma,P^0,X^0),$$

其中 $$P^0=\{X^0\to aX^0b\mid a,b\in\Sigma\}\cup\{X^0\to ab\mid a,b\in\Sigma\}\cup\{X^0\to a\mid a\in\Sigma\}\cup\{X^0\to\varepsilon\}.$$

可以看出,$L(G^0)$是由 Σ 上任意串组成的语言.设待推断文法为 G,对于 $L(G)$中任一句子 w,它所对应的 C 中的句子记为 c_w,是由 G^0 生成 w 所依次使用的产生式的标志构成的串;反过来,对于 C 中任一句子 f,设 $f=f_1f_2\ldots f_k$,则从 G^0 中 X^0 出发,依次使用 P^0 中标志为 $f_1f_2\ldots f_k$ 的产生式所产生的串 w_c,即为 f 所对应的 $L(G)$ 中的句子.Takada 还给出了从正规语言 C 向相应的 ELL L 进行翻译的多项式算法,即已知 C,求出 ELG G,使 $L(G)=L_C(G^0)$.于是一个 ELG 的推断问题可以归结为正规语言(即相应的控制集 C)的学习问题.

Sempere[28]等人同样设计出一个算法,利用 ELG 的一个特性,将 ELG 的推断问题归结成正规语言的学习问题.归结策略来自于 ELL 的一个特性,进一步的解释见参考文献[28].

一些研究者还对其他一些 CF 子类的推断问题做了研究,限于篇幅,本文不作介绍.

3.4 遗传算法和神经网络技术用于文法推断

从 90 年代初开始,文法推断的研究逐渐活跃起来,这一方面是受文法推断所展现的和潜在的应用前景所激励;另一方面是计算机科学中其他研究领域的进展给文法推断的研究注入了灵感.前文提到,文法推断问题的复杂性源于其庞大的猜测空间,计算机研究者很自然地考虑到将遗传算法搜索技术用于文法推断研究[29,30];同样地,由于某些神经网络与特定文法类在识别能力上具有某种等价关系,甚至可以设计出特定类型的神经网络,使其网络结构与文法规则具有一定的对应关系,因此,文法推断问题可以转化为神经网络的学习问题[31,32].

3.4.1 遗传式文法推断

遗传算法是模拟生物进化过程的计算模型,其主要特点是:① 体现生物系统中生存竞争、优胜劣汰的原则;② 体现知识更新的随机性.遗传算法作为一种新的全局优化搜索算法,以其简单通用、鲁棒性强、适于并行处理等特点,而被广泛采用.一个遗传算法包含如下要素:(1) 编码;(2) 初始群体的设定;(3) 适应度函数的设计;(4) 遗传操作设计;(5) 控制参数设定.CF 文法推断遗传式算法(GIGA)同样包含这 5 个要素.

所谓编码是指,解空间中解(又称个体)的编码.CF 文法推断问题的解就是一个完整的 CF 文法,因此,GIGA 中的编码问题首先是文法的表示问题.我们知道,同一个 CF 语言可以用不同"形式"的 CF 文法表示.如自由形式,即每个产生形如 $A\to\alpha,\alpha\in V^*$.若对 α 的格式作不同的限制,就会得到不同的范式.其中使用较多的是 Chomsky 范式,即每个产生式形如 $A\to BC$ 或 $A\to a$,其中 $A,B,C\in N,a\in\Sigma$.确定文法的形式后,对文法的进一步编码则大致可分为两类.一类是直接编码,即每个解用其所有产生式连接在一起得到的字符串表示;另一类是间接编码,即根据一定的规则,将产生式转换为树、图、递归转移网络或 0、1 序列.

由于遗传算法的群体型操作的需要,必须为遗传操作准备一个由若干可能解组成的初始群体.通常,在 GIGA 中,初始群体是由随机产生的多个解组成,每个解又是由如下方法得到的.首先,分别假定终极符集合 Σ' 和

非终极符集合 N',在 Σ' 和 N' 上随机产生 k 个产生式,当然,每个产生式必须符合编码阶段选定的范式形式,这里,k 大于或等于目标文法中的产生式数,甚至要求 Σ' 和 N' 分别等价于目标文法的终极符集合和非终极符集合.

适应度函数用于在进化过程中评估个体的优劣.在 GIGA 中适应度函数一般描述个体(即一个 CF 文法)对样本的协调程度,也就是从对正例的接受情况和对反例的拒绝情况两方面综合评估.

遗传操作一般包括杂交(crossover)和变异(mutation)两种.GIGA 中杂交操作一般是将两产生式右部某位置的符号(即终极符或非终极符)进行交换;变异操作是将某产生式右部某位置的符号(终极符或非终极符)替换成别的符号(终极符或非终极符).

控制参数设定包括设定群体大小(即每代处理的个体数量)以及从当前代生成下一代的个体时,当前代中的个体分别进行杂交或变异的概率.

从以上描述我们可以看出,遗传式文法推断方法有如下特点:(1) 推断的出发点包括正例和反例;(2) 最好能预先知道待获取文法的一些性质,如产生式数、终极符集合和非终极符集合;(3) 获取的文法一般是某种范式.

3.4.2　神经网络技术用于文法推断

神经网络(neural networks)是近年来人工智能的一个前沿研究领域.它具有如下特点:大规模并行结构;信息的分布存储和并行处理;良好的自适应性、自组织性和容错性;较强的学习、记忆、联想和识别能力.

到目前为止,将神经网络技术用于上下文无关语言学习的文献较少,相对较多的是用于正规语言的学习.由于 RNN(recurrent neural network)与有限自动机(严格地说是与随机正规文法[32])在识别能力上具有某种等价关系[33],于是,正规语言的学习转化为训练 RNN[31,32].神经网络用于文法推断的典型情形是,将通过正例和反例训练得到的神经网络作为分类器,用于判断任一串是正例还是反例.将神经网络用于文法推断的另一情形是,将神经网络的结构对应于文法结构,在这种情况下,通过学习得到的神经网络可以转换成相应的文法[34].

4　随机文法推断

最近几年,文法推断研究侧重于随机文法类的推断.该趋势的根源一方面在于传统精确推断模型和方法的乏力;另一方面在于随机模型和方法在电子工程的很多领域的有效应用.随机文法是在通常文法的基础上加入了概率模型.随机文法推断过程的实质是以概率估算为手段加速结构的推断过程,即用某种"合适"函数评估猜测的可能性,以此加速猜测空间的搜索过程,从而减低计算复杂性.这里的"合适"函数考虑的因素通常包括文法本身的复杂性(Feldman 在文献[35]中给出了文法复杂性的一个形式定义)和文法与样本之间的吻合程度.

Horning 在其博士论文中提出一种枚举方法从概率样本(即每个样本串附带一个概率)极限识认 SCFG[36].其思想是搜索所有可能的猜测,用定义的"合适"函数进行取舍.类似的方法有文献[15]和文献[37]中所提到的方法.显然,由于枚举方法固有的特性,这些方法的效率很低.

另一个较早的 SCFG 的推断方法是 Cook 等人提出的登山法[38].他们首先定义了一种衡量串复杂度的方法,在此基础上定义文法的复杂度.其推断的大致过程是:首先从给定的带概率的样本中构造一个初始文法(此文法的复杂度较高),然后根据一系列启发规则(如替换重复出现的串)对初始文法逐步求精,直至文法无法简化,最终获得一个复杂度最低的文法.与此方法类似,Stolcke 等人[39]提出一个模型合并方法,其过程是:首先从给定样本构造初始文法;然后通过模型合并(如合并 HMM 中的状态或合并 SCFG 中的非终极符)进行文法推广.是否合并以及何时停止合并由文法与样本的吻合程度决定.

以上方法的共同特点是均侧重文法结构的推断.最近的研究则侧重概率参数的估计,即文法结构的推断过程代之以概率参数的估计过程.根据最大可能性原则,目前,关于概率参数的估计算法主要有 HMM 的前-后算法(forward-backward,简称 F-B)[40]和 SCFG 的内-外算法(inside-outside,简称 I-O)[41].以 HMM 为例,从初始 HMM(一个状态全联结 HMM)开始,用 F-B 算法进行概率估算,使其达到局部最优;最后删去概率为 0 或概率很小的状态转换边,从而得到最终 HMM 的结构.下面分别介绍 HMM 和 SCFG 的参数估计过程.

4.1 HMM

一个HMM λ 由一个五元组定义，即 $\lambda=(S,\Sigma,T,O,\pi)$，其中 S 为状态集合，Σ 为字母集，T 为状态转换概率分布，O 为字母输出概率分布，π 为开始状态概率分布. 即 $S=\{s_1,\ldots,s_n\}$，s_i 为 λ 中的状态；$T=\{t_{ij}\mid 1\leqslant i,j\leqslant n\}$，$t_{ij}$ 为从状态 q_i 转换到 q_j 的概率，即如果我们把时间序列 $t=1,2,3,\ldots$ 与状态转换联系起来，设 t 时刻所处状态为 q_i，则 $t_{ij}=Pr(q_t=s_j\mid q_{t-1}=s_i)$；$O=\{o_j(a)\mid 1\leqslant j\leqslant n,a\in\Sigma\}$，$o_j(a)$ 为在状态 q_j 时输出字母 a 的概率；$\pi=\{\pi_i\mid 1\leqslant i\leqslant n\}$，$\pi_i$ 为状态 q_i 是开始状态的概率.

HMM的参数估计可归结为下面3个基本问题(设当前HMM为 λ). 给定一个串 w，

1. 计算 w 的概率 $Pr(w\mid\lambda)$；

2. 寻找最可能路径 $s=q_{i_1},\ldots,q_{i_l}$ 使 $Pr(s\mid w,\lambda)$ 最大，$Pr(s\mid w,\lambda)$ 表示 w 被 λ 接受时其接受路径为 s 的概率；

3. 估计 λ 的参数(即 T,O 和 π)，使 $Pr(w\mid\lambda)$ 最大.

以上问题可用动态编程技术有效地解决. 其中解决第2个问题的有效算法是Viterbi算法[42]；采用F-B算法解决第3个问题，其中用到 F 变量和 B 变量($w=a_1\ldots a_T$). F 变量 $F_k(s_i)$ 定义为 $F_k(s_i)=Pr(a_1\ldots a_k,s_i\mid\lambda)$，即在时刻 k(此时止输出字符序列为 w 的前缀串 $a_1\ldots a_k$)时所处状态为 s_i 的概率；B 变量 $B_k(s_i)=Pr(a_{k+1}\ldots a_T\mid s_i,\lambda)$，即 λ 从状态 s_i 始输出 w 后缀串 $a_{k+1}\ldots a_T$ 的概率. 第1个问题的解决用到 F 变量，具体过程如下：

(1) 初始化

$$F_1(s_i)=\pi_i o_i(a_1),\qquad \text{其中 } 1\leqslant i\leqslant n;$$

(2) 归纳

$$F_{k+1}(s_j)=\Big\{\sum_{i=1}^{n}F_k(s_i)t_{ij}\Big\}o_j(a_{k+1}),\qquad \text{其中 } 1\leqslant j\leqslant n;$$

(3) 终止

$$Pr(w\mid\lambda)=\sum_{i=1}^{n}F_T(s_i).$$

F-B算法是一种期望值最大化(expectation maximization，简称EM)算法. 在介绍其过程之前，先定义概率 $\xi_t(i,j)$，即在 λ 接受 w 的过程中，t 时刻状态为 s_i 且 $t+1$ 时刻状态为 s_j 的概率.

$$\xi_t(i,j)=Pr(q_t=s_i,q_{t+1}=s_j\mid w,\lambda).$$

$\gamma_t(i)$ 表示在 λ 接受 w 的过程中 t 时刻所处状态为 s_i 的概率，即

$$\gamma_t(i)=\sum_{j=1}^{n}\xi_t(i,j).$$

F-B算法处理第3个问题的过程如下：

(1) 设 λ_{old} 为初始模型；

(2) 根据当前 λ_{old} 和给定串 w，

① 对任意状态对 s_i,s_j，计算并设置 λ_{new} 中的 t_{ij} 的值为

$$t_{ij}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}=\frac{\lambda\text{接受}w\text{过程中经过从状态}s_i\text{到}s_j\text{路径的可能性}}{\lambda\text{接受}w\text{过程中经过从状态}s_i\text{向其他状态或本身转换路径的可能性}};$$

② 对每个状态 s_j 和输出字符 a，计算并设置 λ_{new} 中 $o_j(a)$ 的值为

$$o_j(a)=\frac{\sum_{t=1(t\text{时刻输出字符为}a)}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}=\frac{\lambda\text{接受}w\text{过程中经过状态}s_j\text{且输出字符为}a\text{的可能性}}{\lambda\text{接受}w\text{过程中经过状态}s_j\text{的可能性}};$$

(3) 用 λ_{new} 替换 λ_{old}，重复步骤(2)直至 λ_{old} 无大变化.

Baum等人[43]证明：(1) 若初始 λ_{old} 正好是最可能模型(即期望值最大)，则步骤(2)的结果是 $\lambda_{old}=\lambda_{new}$；(2) 否则，$\lambda_{new}$ 比 λ_{old} 可能性大，即 $Pr(w\mid\lambda_{new})>Pr(w\mid\lambda_{old})$.

4.2 随机上下文无关文法

一个SCFG G_ζ 可以描述为一个CFG附带一个函数 q,即 $G_\zeta=(G,q)$,这里,$G=(N,\Sigma,P,X)$是一个CFG,q是一个函数 $q:P\to[0,1]$且满足

$$\forall\ A\in N,\sum_{A\to\alpha\in P}q(A\to\alpha)=1.$$

对于句子 s,其相应的一个推导过程 $D(s)=(r_1,\ldots,r_m)(r_i\in P)$ 的概率为

$$Pr(s,D(s)|G_\zeta)=q(r_1)\ldots q(r_m).$$

对 Σ^* 上的任一串 w,w 被 G 接受的概率为

$$Pr(w|G_\zeta)=\begin{cases}\sum_{D(w)}Pr(w,D(w)|G_\zeta), & \text{如果 } w\in L(G)\\ 0, & \text{否则}\end{cases}$$

于是,$Pr(w|G_\zeta)(w\in\Sigma^*)$构成 Σ^* 上的概率分布.

SCFG的参数估计同样包含类似于HMM的3个基本问题.

(1) 对于一个给定串 w,计算由SCFG G_ζ 所赋予的概率 $Pr(w|G_\zeta)$;

(2) 寻找 G_ζ 关于 w 的最可能的派生树;

(3) 参数估计(即估计 G_ζ 中 q),使 $Pr(w|G_\zeta)$最大.

前两个问题可以用与CKY和Earley分析方法类似的动态编程方法解决.解决第3个问题的标准方法是内-外(inside-outside,简称I-O)算法.其中 I 变量和 O 变量的定义分别为($w=a_1\ldots a_{|w|}$):

$$I(i,j,A)=Pr(A\overset{*}{\Rightarrow}a_i\ldots a_j|G_\zeta),$$

$$O(i,j,A)=Pr(S\overset{*}{\Rightarrow}a_1\ldots a_{i-1}Aa_{j+1}\ldots a_{|w|}|G_\zeta),$$

其中 $A\in N,1\leqslant i\leqslant j\leqslant|w|$.内变量即为由非终结符 A 能产生 $a_i\ldots a_j$ 子串的概率;外变量即为 G 生成句型 $a_1\ldots a_{i-1}Aa_{j+1}\ldots a_{|w|}$ 的概率.上面的I-O算法要求文法为Chomsky范式,这在很多场合很不方便,而且其时间复杂度至少为 $O(n^3)$,这里,n 为样本串的典型长度.为避开这些问题,Sakakibara等人[44]将HMM的F-B算法推广到树文法,由此设计了从结构化实例样本推断SCFG的新方法,其名为Tree-Grammar EM.Sakakibara等人进一步设计出对付非结构化实例的推断方法.它采用如下循环迭代过程来估计样本的结构.

(1) 用初始文法分析样本,得到部分结构化的串集;

(2) 由部分结构化的串样本,用Tree-Grammar EM方法估计一个新SCFG G_ζ;

(3) 用当前文法 G_ζ 去分析并获取更精确的结构化样本集合;

(4) 重复步骤(2)和步骤(3),直至 G_ζ 的结构不变.

5 文法推断的应用

文法推断研究产生的动因之一是构造人类学习自然语言的模型,尽管后来的研究者对能否达到这一目的表示怀疑.实际上,文法推断的主要应用是结构化模式识别(structural pattern recognition),在这些场合,某些特定的模式类用一个文法描述,而且手工构造这些文法非常困难,所以需要自动化或部分自动化来构造这些文法.Crespi-Reghizzi等人提出用文法推断进行程序语言语法的设计[45].

近年来,随着越来越多的文法推断算法的出现,文法推断的应用越来越广泛.主要有信息检索、语音和自然语言处理、基因分析、数据采集和知识获取等.还有一些研究者直接用文法推断辅助文法构造[46~48].

6 文法推断的研究趋势

本文介绍了文法推断的理论模型和推断方法,特别介绍了其表示类为上下文无关文法类或其子类、随机上下文无关文法和隐马尔可夫模型的推断方法.我们可以把文法推断的研究成果归纳为两方面.理论方面侧重于文法推断抽象过程的研究,以建立合理的文法推断理论模型,并试图寻求在特定的抽象模型下,文法推断的可计算性和计算复杂性等理论问题的答案;实践方面则致力于寻找有效的且可行的文法推断方法,并尝试着将这些

方法用于解决实际问题.文法推断研究的发展历程可归纳为:从文法推断理想框架(在此框架下,推断算法是一个黑盒,或者说是完全自动化的,推断过程无需人的参与)到交互模型(此时加入人对推断过程的干预,以求降低计算复杂性),再到近似模型(降低对推断结果的要求,即对近似解或次优解的认可).

可以说,到目前为止,虽然文法推断的研究取得了很大的进展,但它作为一个研究领域,还远不够成熟,甚至有一些基本概念还有待统一.已有的研究工作或着眼于文法推断抽象过程和理论模型的研究,或着眼于某个具有特殊性质的特定文法类,虽然它能在某个已有理论模型下被有效推断(识认),或它的推断问题能被归约为某些更简单文法类(如正规文法、可逆文法[49])的推断问题,但实际应用环境很难限制到该文法类;还有一些研究者考虑附加更多的输入信息(如实例串的结构信息),同样地,在实际应用环境中,这些信息仍然很难提供.

尽管如此,由于文法推断所展现的和潜在的应用,将会激发更多的研究者尝试从新的角度或用新的方法寻找实际可用的文法推断方法.特别是,随着计算机科学其他学科和领域的发展,研究者会尝试将一些新理论和新技术运用到文法推断中,如将遗传算法和神经网络用于文法推断就是典型的例子.同时我们认为,概率模型和方法将继续作为研究的主流.可以说,文法推断的研究将是一个有意义且富有挑战性的工作.

致谢 本文的研究工作是在董韫美院士的指导下完成的.其中多篇主要文献由肖勇先生查得,作者在此表示衷心的感谢.

参考文献

1 Fu K S, Booth T R. Grammatical inference: introduction and servey(Part 1, Part 2). IEEE Transactions on Systems, Man and Cybernetics, 1975,SMC-5(1):95~111;1975,SMC-5(4):409~423

2 Angluin D, Smith C H. Inductive inference: theory and methods. Computing Surveys, 1983,15(3):237~269

3 Miclet L. Grammatical inference. In: Bunke H, Sanfeliu A eds. Syntactic and Structural Pattern Recognition——Theory and Applications. Singapore: World Scientific Publishing Company, 1986. 237~290

4 Sakakibara Y. Grammatical inference: an old and new paradigm. Lecture Notes in Artificial Intelligence 997, 1995. 1~24

5 Solomaa A. Formal Languages. London: Academic Press, 1973

6 Chomsky N. Syntactic Structures. The Hague: Mouton and Co., 1957

7 Gold E M. Language identification in the limit. Information and Control, 1967,10(5):447~474

8 Gold E M. Complexity of automation identification from given data. Information and Control, 1978,37(4):302~320

9 Angluin D. Learning regular sets from queries and counter-examples. Information and Computation, 1987,75(1):87~106

10 Angluin D. Queries and concept learning. Machine Learning, 1988,2(3):319~342

11 Angluin D. Negative results for equivalence queries. Machine Learning, 1990,5(2):121~150

12 Angluin D, Kharitonov M. When won't membership queries help? In: Proceedings of the 23rd Annual ACM Symposium on Theory of Computing. New York: ACM Press, 1991. 444~454

13 Valiant L G. A theory of learnable. Communications of the ACM, 1984,27(11):1134~1142

14 Giordano J Y. Inference of context-free grammars by enumeration: structural containment as an ordering bias. LNAI 862, 1994. 212~221

15 Wharton R M. Grammar enumeration and inference. Information and Control, 1977,33(3):253~272

16 Solomonoff R J. A new method for discovering the grammars of phrase structure languages. Information Processing, New York: UNESCO, 1959. 285~290

17 Knobe B, Knobe K. A method for inferring context-free grammars. Information and Control, 1976,31(2):129~146

18 Tanatsugu K. A grammatical inference for context-free languages based on self-embedding. Bulletin of Informatics and Cybernetics, 1987,22(2):149~163

19 董韫美.获取上下文无关文法的一种交互式算法.计算机学报,1996,19(3):168~173
(Dong Yun-mei. An interactive algorithm for acquisition of context-free grammars. Chinese Journal of Computers, 1996,19(3):169~173)

20 蒙韫美. 基于复用的上下文无关文法推断. 软件学报，1996，7(863 专刊)：178～181
(Dong Yun-mei. Context-free grammatical inference based on reusing. Journal of Software, 1996,7(863 Special Issue): 178～181)

21 张瑞岭. 一个上下文无关文法获取过程的设计和实现. 软件学报，1998，9(8)：601～605
(Zhang Rui-ling. Design and implementation of a procedure for acquisition of context-free grammars. Journal of Software, 1998,9(8): 601～605)

22 Crespi-Reghizzi S. An effective model for grammar inference. In: Gilchrist B ed. Information Processing 71, New York: UNESCO, 1972. 524～529

23 Aho A V, Sethi R, Ullman J D. Compilers: Principles, Techniques and Tools. Reading, MA: Addison-Wesley Publishing Company, 1986

24 Sakakibara Y. Learning context-free grammars from structural data in polynomial time. Theoretical Computer Science, 1990,76(2): 223～242

25 Ishizaka H. Polynomial time learnability of simple deterministic languages. Machine Learning, 1990,5(2): 151～164

26 Yokomori T. Polynomial-time learning of very simple grammars from positive data. In: Proceedings of the 4th Workshop on Computational Learning Theory(COLT'91). San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1991. 213～227

27 Takada Y. Grammatical inference for even linear languages based on control sets. Information Processing Letters, 1988,28(4): 193～199

28 Sempere J M, Garcia P. A characterization of even linear languages and its application to the learning problem. LNAI 862, 1994. 38～44

29 Dupont P. Regular grammatical inference from positive and negative samples by genetic search: the GIG method. LNAI 862, 1994. 236～245

30 Wyard P. Representational issues for context-free grammar induction using genetic algorithms. LNAI 862, Berlin: Springer-Verlag, 1994. 222～235

31 Alquezar R, Sanfeliu A. A hybrid connectionist-symbolic approach to regular grammatical inference based on neural learning and hierarchical clustering. In: Carrasco R C, Oncina J eds. LNAI 862, Berlin: Springer-Verlag, 1994. 203～211

32 Carrasco R C, Forcada M L, Santamaria L. Inferring stochastic regular grammars with recurrent neural networks. In: Miclet L, de La Higuera C eds. LNAI 1147, Berlin: Springer-Verlag, 1997. 274～281

33 Cleermans A, Servan-Schreiber D, McClelland J L. Finite-state automata and simple recurrent networks. Neural Computation 1, 1989. 372～381

34 Roques M. Dynamic grammatical representations in guided propagation networks. LNAI 862, Berlin: Springer-Verlag, 1994. 189～202

35 Feldman J. Grammatical inference and complexity. Information and Control, 1972,2(3): 244～262

36 Horning J J. A study of grammatical inference [Ph. D. Thesis]. Stanford University, 1969

37 Van der Mude A, Walker A. On the inference of stochastic regular grammars. Information and Control, 1978,2(5): 310～329

38 Cook C M *et al*. Grammatical inference by hill-climbing. Information Sciences, 1976,2(1): 59～80

39 Stolcke A, Omohundro S. Inducing probabilistic grammars by Bayesian model merging. LNAI 862, Berlin: Springer-Verlag, 1994. 106～118

40 Rabiner L R. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of IEEE, 1989,77(2): 257～286

41 Lari K, Young S J. The estimation of stochastic context-free grammars using the inside-outside algorithm. Computer Speech and Language, 1990,4(1): 35～56

42 Viterbi A J. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Transactions on Information Theory, 1967,IT-13(2): 260～269

43 Baum L E, Sell G R. Growth functions for transformations on manifolds. Pacific Journal of Mathematics, 1968,27(2): 211～227

44 Sakakibara Y *et al*. Stochastic context-free grammars for tRNA modeling. Nucleic Acids Research, 1994,22(1):5112～5120

45 Crespi-Reghizzi S, Melkanoff M A, Lichten L. The use of grammatical inference for designing programming language. Journal of ACM, 1973,16(2):83～90

46 Young S J, Shih H H. Computer assisted grammar construction. LNAI 862, Berlin: Springer-Verlag, 1994. 282～290

47 Dong Yun-mei. Collection of SAQ Report, No. 1～7. Technical Report No. ISCAS-LCS-95-09, Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, 1995

48 Dong Yun-mei *et al*. Collection of SAQ Report, No. 8～16. Technical Report No. ISCAS-LCS-96-1, Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, 1996

49 Angluin D. Inference of reversible languages. Journal of ACM, 1982,29(3):741～765

Grammatical Inference: Retrospect and Prospect

ZHANG Rui-ling

(*Laboratory of Computer Science Institute of Software The Chinese Academy of Sciences Beijing* 100080)

Abstract Grammatical Inference (GI) is a problem of inductive learning of formal languages, which deals with how to obtain the grammatical description of a formal language from the given finite data drawn from the language. In this paper, the author provides a survey of the history and recent advances in GI field. The author first presents some learning models for GI, then enumerates the methods for GI with an emphasis on the results concerning the inference of context free grammar class and its some subclasses, hidden Markov models, and stochastic context-free grammar class. At last, the author briefly gives some applications of GI as well as the future directions of GI research.

Key words Learning from examples, inductive inference, learning of formal languages, grammatical inference.