帰納推論について

宮原, 哲浩
九州大学教養部情報科学

https://doi.org/10.15017/6796277

出版情報：言語科学. 25, pp.148-155, 1990-03-01. 九州大学言語文化部言語研究会
バージョン：
権利関係：

帰納推論について

宮原哲浩

九州大学教養部情報科学

1．はじめに

帰納推論(inductive inference)とは，与えられたデータから，それを説明する一般的な規則を推測する過程である．たとえば，次のような文字列データが与えられたとしよう．

010, 0010000, 0001000000000

この例から，

「1の両側に0が並んだ文字列で1の右側の0の個数は左側のものの二乗である」

という規則を導き出すような過程が帰納推論である．しかし，上の例に

0000100000000000000000000000

が加わると推測は正しくなくなるので，帰納推論の結果はいつも正しいわけではない．この場合には，別の推測に変更する必要が生じる．たとえば，

「1の左側にx個の0，右側にy個の0 が並んだ文字列で，$y=x^3-5x^2+11x-6$である」

という規則に変更すれば，これは上のすべてのデータを説明している．もちろん，この次に与えられるデータがこの規則に合っているとは限らないので，さらに推測を変更しなければならない可能性がある．したがって，帰納推論とは一般に無限に続く過程であるといえる．言語学習の過程やソフトウェアの進化などは帰納推論とみなせるので，このことは不自然なことではない．

帰納推論は正確にいうと次の5 つによって定義されると考えることができる[3]．

(1) 推論の対象となる規則の集合
(2) 規則の表現
(3) 規則に対する例の提示(例示)の仕方
(4) 推論の方法
(5) 正しい推論の基準．

推論の対象となる規則としては，通常，計算可能関数や形式言語が取り扱われる．また，規則の表現としては，関数の計算をするチューリング機械や言語を規定する文法などが考えられる．チューリング機械は，計算機の数学的モデルであり，プログラムだと思ってよい．たとえば，文法推論は，規則として言語，その表現として文法を扱う帰納推論である．

関数fに対する例示には，入出力の対(x, f(x))の列や，f を計算するチューリング機械の動作系列などが考えられる．推論機械に関数f の入出力の対(x, f(x))を与えることは，プログラムの自動合成システムに，入力xとfを計算するプログラムの入力x上での出力f(x)を与えることに対応する．よって，例からのプログラム自動合成は関数の帰納推論とみなすことができる．形式言語は語の集合であるから，言語Ｌの例には，Ｌに属する語と属さない語の二種類が考えられ，それぞれ正データ，負データと呼ばれる．一般に例示は無限列である．

推論は

「入力要求→推測の計算→出力」

という過程を無限に繰り返すことによって行うものとする．つまり推論は，次々に例を入力として受け取り，次々とその時点での推測を計算し，その結果を出力していく．各時点での推測の計算はそれまでに受け取った例の有限列に基づいていると考えられるので，以下では，推論の方法としては例の有限列から推測を計算するチューリング機械を用いるものとする．このチューリング機械を推論機械(inference machine)と呼ぶ．問題によっては，ある性質をもつ推論機械だけに限定する場合もある．たとえば，与えられた例の有限列と矛盾しない推測だけを出力する推論機械に限定することがある．

正しい推論の基準とは，帰納推論が成功したと

『言語科学』　第25号（1990）P.148－155

みなす基準のことであり，帰納推論において最も重要な概念である．基本的な基準として，帰納推論が無限に続く過程であることを念頭においた，Gold [8, 9] による極限における同定 (identification in the limit) の概念がある．推論機械Mが規則Rを極限において同定するとは，Rの例示が与えられたときに，Mの生成する出力の列がある表現τに収束し，すなわち，ある時点以降のMの出力がすべてτであり，しかもτがRの正しい表現となることをいう．また，規則の集合Γが帰納推論可能であるとは，ある推論機械Mが存在して，MはΓの任意の規則Rを極限において同定することをいう．

ここで，帰納推論可能という概念は単一の規則に対しては意味がなく，規則の集合に対してだけ有効となる点に注意しよう．

冒頭で示した例は，規則を単に文字列の集合すなわち言語とみなせば，文法推論の例になっており，文字列を整数の対とみなせば，関数の帰納推論の例にもなっている．帰納推論とよく似た概念に外挿 (extrapolation) がある．これは知能テストなどで，よくみかける

「数列1, 3, 5, 7の次にくる数は何か」

といった問題に対応している．帰納推論では例を説明する規則を求めることが目的であるのに対し，列の外挿では与えられた有限列の次の要素を予測することが目的である．冒頭の例も，少し修正すれば，外挿問題とみなすことができる．

以下の節では，関数・言語・モデルの帰納推論について，その理論的成果の概要を紹介する．

2．関数の帰納推論

計算機による例からのプログラムの自動合成は関数の帰納推論としてとらえることができる．

最近の計算機の性能向上にはまことに著しいものがあるので，次のような疑問が浮かぶかもしれない．

「もし将来，計算機資源がいくらでも使えるようになったとしたら，無限ループに入らないようなプログラムであればどんなものでも，例から自動的に合成するような夢のシステムを構築することができないだろうか？」

Gold [8, 9] 等によって始められた帰納推論の数学的理論は，計算理論による定式化の下で，上のような疑問に正確に答えようとするものである．この疑問に対する否定的な解をはじめとして，これまでに得られた数多くの成果のうちから代表的なものをいくつか紹介しよう．

適当な符号化を用いることにより，データとしては自然数 0, 1, 2, ... だけを考察すれば十分であるから，本節での推論の対象となる規則は，任意の自然数を自然数に写す関数で，プログラムによって計算可能であるとする．計算可能関数f の例示としては自然数の無限列 f(0), f(1), ... だけを考え，fの表現としてはfを計算するプログラムを考える．

正しい推論の基準としては，極限における同定以外にも考えられる．他の基準と区別するために，「極限において同定する」ということをＥＸ推論する (EX-identify) ということにする．このようにいうのは，推測が関数を計算する（EXplain）プログラムに収束するからである．計算可能関数の集合であって，ＥＸ推論可能なものの全体を，同じ記号ＥＸを使って表す．

実際の計算機で実現された推論機械 (たとえば，Shapiro [16] のモデル推論システム) では原理的には，枚挙できるすべての推測を枚挙して，各時点でそれまでに受け取ったデータを説明できる推測で最初に見つかったものを出力するいう戦略をとる．この戦略は，枚挙手法 (enumerative method) と呼ばれている．計算可能関数から成る枚挙可能集合の部分集合，すなわち，枚挙手法が適用可能な集合の全体をＮＵＭで表す．次の結果はこの分野において初めて得られたものである．

Gold [8]　計算可能関数から成る枚挙可能集合の部分集合はＥＸ推論可能である．すなわち，ＮＵＭ $\subseteq$ ＥＸである．

このような最も素朴とも思える枚挙手法を用いる戦略は，帰納推論を行う際にいつでも通用するのだろうか．この問題はGold [8] (1965) 以来，数年間未解決問題であったが，否定的に解決された．

Barzdin [5]　枚挙手法が適用できないがＥＸ推論可能な計算可能関数の集合が存在する．すなわち，ＮＵＭ $\subsetneq$ ＥＸ．

こうして，ＥＸ推論するという概念が我々が想像するよりもはるかに広い概念であることがわかった．無限ループしないプログラムは，規則としての計算可能関数の表現である．すべての計算可能関数(Recursive function)の集合$\mathcal{R}$を計算可能関数で枚挙することは不可能であることが知られているので，$\mathcal{R}$を枚挙手法によって帰納推論することはできない．

それでは，$\mathcal{R}$はＥＸの意味では推論可能であろうか．この問題も否定的に解決された．つまり，本節の始めに述べたような夢のシステムは存在しないことが証明された．

Gold[8] すべての計算可能関数をＥＸ推論するような推論機械は存在しない．すなわち，$\mathcal{R} \notin EX$．

そうすると，推論が成功したとみなす基準を一体どこまで弱くすれば，$\mathcal{R}$が推論可能となるのだろうか，という問題が生じる．基準を少しずつ弱くしていってみよう．まず，時点n で計算可能関数fのデータf(0),....,f(n)を与えたときのＭの推測をＭ(f[n])で表す．Ｍ(f[n])があるプログラムｐに収束したとしよう．ｐが最大限m 個の例外を除いて(m=*の場合は，有限個の例外を除いて)fを正しく計算するとき，ＭはｆをEX^m 推論するという．

推測の列が収束するというのは，推測が表すプログラムが文字列として収束するということを意味する．そこでさらに基準を弱くして，推測の無限列が収束しなくても，ある時点から以降はプログラムの動作として正しければよい(Behaviory Correct)という基準を考えよう．ある時点以降のすべての時点nで，推測Ｍ(f[n])が最大限m個の例外を除いて(m=*のときは，有限個の例外を除いて，)fを正しく計算するプログラムであるときに，ＭはfをBC^m推論するという．

計算可能関数の集合であって，EX^m推論可能，BC^m推論可能なもの全体のなす族をそれぞれ，EX^m，BC^mで表す．このような族は，それぞれの基準における帰納推論が可能な範囲を表しており，推論方式(同定タイプ，identification type)と呼ばれている．様々な成功の基準を満たす推論機械の推論能力を比較するためには，対応する推論方式の包含関係を明らかにすればよい．このような観点から，推論方式のつくる階層を解明する研究が進んでいる[3,12]．推論方式の階層が様々な基準を満たす推論機械の能力の差を表しているということは，形式言語理論でいえば，Chomskyの階層のような言語のクラスの階層が各種の受理装置の能力の差を表しているということに対応する．たとえば，次のような階層がわかっている．

Case & Smith[7] $EX = EX^0 \subsetneq EX^1 \subsetneq EX^2 \subsetneq \ldots \subsetneq EX^* \subsetneq BC^0 \subsetneq BC^1 \subsetneq BC^2 \subsetneq \ldots \subsetneq BC^*$．$\mathcal{R} \in BC^*$．

つまり，すべての計算可能関数の集合$\mathcal{R}$は，BC^* 推論まで基準を弱くしてはじめて推論可能になる．この基準は現実的ではないが，集合に対する推論可能性の定義により，$\mathcal{R}$の任意の部分集合はBC^*推論可能となるので，BC^*は最上位の推論方式であることがわかる．

外挿を行う外挿機械(extrapolating machine)とは，各時点nで，計算可能関数ｆのデータ f(0)，f(1)，....，f(n)を基にして，次の値 f(n+1)に対する予測値を必ず出力するチューリング機械のことである．外挿機械Ｍが，ある時点以降のすべての時点nで，次の値f(n+1)を正しく予測する(Ｎext Ｖalue-extrapolate)とき，ＭはfをＮＶ外挿するという．計算可能関数の集合であって，ＮＶ外挿可能なものの全体をＮＶで表す．

外挿機械を利用して推論機械を構成することができ，また逆に，推論機械を利用して外挿機械を構成することができるように思われる．それでは，外挿と帰納推論の能力は等しいのだろうか．また，ＮＶ外挿可能な集合とはどのような関数の集合であろうか．

この問に答えるためには，h 従順(h-easy)という概念が必要である．h を計算可能関数とする．計算可能関数fがh従順であるとは，f を計算するプログラムで，有限個を除いたすべての自然数 x に対してf(x)の計算をh(x)ステップ以内で終了するようなものが存在するときをいう．

Barzdin & Freivald[5]，Blum & Blum[6] $\{\Gamma \subseteq \mathcal{R} \mid$ ある計算可能関数hがあって，Γのすべての関数はh従順$\} = NV = NUM$．

この結果によって，ＮＶ外挿と枚挙手法の能力

は等しく，どちらも，すべての関数がh 従順であるような集合に対して適用できることがわかった．

Blum & Blum[6]は，計算量を表す尺度として，さらにh穏健(h-honest)，O穏健(O-honest)という概念を用いて，推論方式の特徴付けを与え，推論可能性と計算量との間に密接な関係があることを示した．

ＮＵＭ⊊ＥＸであるが，ＮＵＭとＥＸの間には，どのような基準があるのだろうか．推論機械Ｍが出力する推測M(f[n])が，すべての時点n でそれまでに受け取ったデータ f(0),...,f(n)と矛盾しない(CONSistent)というのは，望ましい基準である．この基準を満たすＭがfをＥＸ推論するとき，ＭはｆをＣＯＮＳ推論するという．このように，応用上望ましい基準を要求すると，一般に推論能力は小さくなることがわかっている [10,20]．計算可能関数fに対する例示としては，f(0)，f(1)，... だけを考えても，一般には推論能力は変わらないことがわかっている．しかし，ＥＸより下位の基準，すなわち，ＣＯＮＳのような応用上望ましい基準の下では，データを任意の順序で与えると推論能力が変化することがある[10]．任意の(ARBitraly) 順序で与えられた例示$(x_0, f(x_0))$, $(x_1, f(x_1))$,...のそれぞれに対して，Ｍがｆを ＣＯＮＳ推論するとき，ＭはｆをＣＯＮＳ^{arb}推論するという．

推論方式ＣＯＮＳ，ＣＯＮＳ^{arb}に対して，次の関係がわかっている．

Wiehagen[20]，Jantke & Beick[10]
ＮＵＭ⊊ＣＯＮＳ^{arb}⊊ＣＯＮＳ⊊ＥＸ．

３．言語の帰納推論

本節では，帰納推論の対象として言語を取り上げる．言語の帰納推論が関数を対象とする場合と異なる点は，例示の仕方にある．関数ｆの例示は入出力対の列であるのに対し，言語Ｌの例示には２通りの方法が考えられる．１つは正データだけからなる列を与えるもので，もう１つは正・負両データからなる列を与えるものである．言語Ｌの特徴関数fとは，語から0または1への関数で，$w \in L$ならば$f(w)=1$，そうでないならば$f(w)=0$であるものである．正・負両データからのＬの帰納推論はその特徴関数 fの帰納推論とみなすことができる．したがって，言語を対象とする場合に固有な理論上の問題は，正データからの帰納推論にあるといえる．そこで，ここでは，正データからの言語の帰納推論に焦点をあてて議論を進めることにする．

言語Ｌの完全提示とは，すべての語と，その語がＬに属しているかどうかを表す0または1の対の無限列 $(w_1, t_1), (w_2, t_2), ...$である．言語Ｌの正提示とは，Ｌの語がすべて現れる無限列$w_1, w_2, ...$である．ここで，無限列は重複を許すものとする．

言語族$\Gamma = L_1, L_2, ...$が完全データ(正データ) から帰納推論可能であるとは，ある推論機械Ｍが存在して，Γの任意の言語L_iの任意の完全(正)提示が与えられたとき，Ｍの生成する推測の列が，$L_j = L_i$ となるjに収束することをいう．ここで，言語L_iの添字iはこの言語の規定するある文法の名前を表している．L_iの表現とは$L_j = L_i$となる添字jのことである．

Gold[9] は言語の族は完全データから推論可能であっても，正データから推論可能であるとは限らないことを示した．Σを記号の有限集合とする．言語の族ΓがΣ上のすべての有限言語と少なくとも1 つの無限言語を含むとき，Γは超有限であるという．次の定理は，Goldによって証明されたもので，帰納推論の初期の研究で得られた最も重要な結果である．

Gold[9] 超有限な言語族は正データから帰納推論可能でない．

この定理から，正データから帰納推論可能な言語族は超有限でないものに限られることがわかり，Chomsky の階層の最下位にある正規言語の族でさえも正データから推論できないことがわかる．この事実は，帰納推論の応用にとって否定的に思われる．しかし，正データからの帰納推論が完全データからのものより能力が劣っているということと，興味ある族を推論できるかどうかとは別の問題である．実際，Angluin[1,2]は正データから推論可能な族を特徴付ける定理を証明するとともに，自明でなく興味深い族が存在することを示した．彼女が導入したパターン言語の族は，正データから推論可能な族で最も重要なものである．

正データからの帰納推論とは，ある言語に属している語，つまり正しい例だけからの推論であり，言語に属さないものに関する情報も用いることができる完全データからの帰納推論より自然である．これは，通常の例からの学習においては，誤りの例をまったく必要としないことがあるということからも妥当である．しかし，上の定理からもわかるように，ある状況下では，誤りの例が本質的に必要であることも事実である．

ここで，正データからの帰納推論の困難さについて考察してみよう．言語族$\Gamma = L_1, L_2, \ldots$を対象としているとしよう．言語$L_i$ の正データからの帰納推論の過程において出力された推測g が正しくないとき，つまり$L_i \neq L_g$であるときには，次の二通りの場合が考えられる．

(I)　$L_i - L_g \neq \phi$,

(II)　$L_i - L_g = \phi$

それぞれを第I種，第II種と呼ぶことにしよう．

第I種の誤りに対しては，L_g が誤りであることを示す反例$w \in L_i - L_g$が必ずL_iの正提示中にいつかは現れるので，いずれ別の推測に変更できる．第II種の誤りの場合は，推測が目標のものより一般的過ぎる，つまり$L_i \subsetneq L_g$である．L_g はL_i のどんな正提示とも矛盾しないので，正提示だけからではg が誤りであることすら認識できない．

これに対し，完全提示は第I種の誤りだけでなく第II種の誤りについてもその誤りを指摘できる．したがって，第II種の誤りが正データからの帰納推論の困難さの本質であるといえる．上の定理の証明は，超有限な族においては，任意の有限言語と無限言語の間には，無数の有限言語が存在するので，正提示からでは第II種の誤りを防ぐことができないことによっている．

さて，Angluin[1, 2]が示した結果を紹介しよう．

$\Gamma = L_1, L_2, \ldots$を言語の族とする．次の条件を満たす集合$T_i$を$L_i$の有限証拠集合という：

(1)　T_iはL_iの有限部分集合であり，かつ

(2)　T_iを含み，しかもL_iの真部分集合であるようなL_jが存在しない．

ここで，T_iがL_iの有限証拠集合であるとき，L_iはT_iを含む極小の言語となっている点に注意しよう．言語の族$\Gamma = L_1, L_2, \ldots$が条件1を満たすとは，任意の$i$に対し，$L_i$の有限証拠集合の要素を枚挙する手続きが存在することをいう．

Angluin[1, 2]　言語の族$\Gamma = L_1, L_2, \ldots$が正データから帰納推論可能であるための必要十分条件はΓが条件1 を満たすことである．

この定理より，有限証拠集合が第II種の誤りを防ぐのに重要な役割を果たすことがわかる．すなわち，言語L_iの正データをT_iを含むように十分多く与えると，その時点で第II種の誤りはなくなるのである．

Angluin は，さらに正データからの帰納推論可能性に関するいくつかの条件を考察している．ここでは，次の条件2 についてのみ紹介する．他のものについては，彼女の論文を参照されたい．

言語族$\Gamma = L_1, L_2, \ldots$が条件2を満たすとは，任意の空でない語の有限集合$S$に対し，$S$を含む$\Gamma$の言語は有限個である，すなわち

$C(S) = \{L \mid S \subseteq L$ かつ，あるiに対して$L = L_i\}$

が有限個の言語からなることをいう．

Angluin[1, 2]　言語族$\Gamma = L_1, L_2, \ldots$が条件2を満たしているならば，$\Gamma$は正データから帰納推論可能である．

条件2 は正データからの推論可能性の十分条件であるが，必要条件ではないことも証明されている．しかし条件2は，条件1と違って，添字付けと無関係であるため適用が容易である．実際，上の定理を用いて，パターン言語の族など多くの言語族の正データからの推論可能性を示すことができる．

XをΣとは別の記号からなる記号の可算集合とする．Σの要素を定数，Xの要素を変数と呼ぶ．パターンとは定数と変数からなる長さ1 以上の文字列である．パターンpが表わす言語$L(p)$とは，p 中の各変数を任意の空でない定数列で置き換えて得られる定数列全体のことをいう．

たとえば，$\Sigma = \{0, 1\}$，$X = \{x, y, \ldots\}$とする．$p = 00x$はパターンであり，$L(p)$は0, 1からなる長さ3 以上の文字列で先頭が00であるもの全体である．$q = xx$とすると，$L(q)$は0, 1 からなる空でない文字列を 2回繰り返して並べたもの全体を表わす．

パターン言語族が正データから帰納推論可能であることは，次のようにしてわかる．w を任意の語，p を任意のパターンとする．変数への置き換えは空でない定数列に限られているので，w∈L(p)ならば，|p|≦|w|である．ここで，|p| はパターン pの変数も含めた長さを表わすものとする．パターンpが変数の名前を付け替えるとパターンqと等しいならば，L(p)＝L(q)である(たとえば，p＝0x1x，q＝0y1yのとき，L(p)＝L(q)である)．したがって，長さが |w|以下のパターンは無限個あるが，変数の名前の付け替えを無視すると，それらが表わす言語の種類は有限であることがわかる．ゆえに，任意の空でない語の有限集合Sに対して，Sを含むパターン言語の個数は有限である．すなわち，パターン言語族は条件2 を満たしている．したがって，パターン言語族は正データから帰納推論可能であることが証明された．

正データから帰納推論可能であるだけではなく，様々な応用の可能性を含んでいるという点から，パターン言語は重要視されている．詳しくは，文献[11, 18, 19]を参照されたい．

4．モデル推論

これまでに考えてきた帰納推論の対象は，関数や形式言語，プログラム等であった．しかも，これらの対象を個別的に扱ってきた．また，対象のクラスが広いものは，推論の方法が非能率的であったり，推論の方法が能率的であるものは対象のクラスが狭いという難点をもっていた．

最近，Shapiro[16, 17]によって，節形式の第一階言語，特にホーン節の集合を対象にした，統一的で効率のよい帰納推論方式が提案された．この方式に基づくモデル推論システム MIS(Model Inference System) は，関数や形式言語，プログラムなどを同じ枠組みの中で扱えるだけでなく，プログラム開発における半自動のデバッガとしても機能するものである．

Shapiro のモデル推論の枠組みの基本はPopper[14, 15] の「理論は事実によって反駁されるべきである」という理念にある．これは，これまでに考えてきた推測を正しいと見做すための基準としての極限における同定という概念にも反映されているものであるが，彼はその表現に含まれている理論，事実，反駁という概念を第一階述語論理の言葉で捉え直すことから始めている．また，他の帰納推論に対するアプローチと異なる点は，対象として述語論理とそのモデルを選んだ点にある．これが彼の理論を実用的なものにした理由である．すなわち，述語論理には，次のような際だった特徴がある．

(1) 自然な意味をもつ．そのために，不都合が生じた場所を容易に検出できる．

(2) 構文と意味の間に密接な関係がある．そのために，反駁された推測を弱めることが簡単にできる．

(3) 単調でありモジュール単位で扱える．そのために，推測の変更による能力変化やその影響範囲を大まかに予知することができる．

(4) プログラミング言語と見做せる．実際 Prolog に代表されるような論理型プログラミング言語とみることができる．

モデル推論では，ある一階言語L 上のエルブランモデルM を推論の対象とする．モデル推論アルゴリズム（推論機械のこと）IAに入力として与えられる例は，L 上のグランドアトムa とa のM における真理値V（つまり，a∈MならV=true, a∉MならV=False）との対<a, V>であり，M に関する事実と呼ばれる．M に関する事実の無限列でL 上のすべてのグランドアトムが列のどこかに現れるものをM の枚挙と呼ぶ．IAの出力はL 上のプログラム（本節では論理プログラムのことであり，Popperのいう理論に対応する）であり，推測と呼ばれる．帰納推論アルゴリズムIAに，エルブランモデルMの枚挙が与えられたとき，IMが出力する推測の無限列がある時点から以降はずっと，同じプログラムP であり（このとき，IMの出力がP に収束するという），M(P)=Mであるとき，IAはM を極限において同定するという．

次に，モデル推論可能性を特徴づけるh 従順という概念を定義する．h を帰納的関数（計算可能関数）とし，T をプログラムとする．アトムa がT からn ステップ以下で演繹できるとき，$T \vdash_n a$ と書く．L 上のすべてのグランドアトムの枚挙を一つ固定し，$a_1, a_2, \cdots$とする．$T \vdash a_n$を満たす$n>0$に対して有限個の例外を除いて，

$$\min\{j \mid T \vdash_j a_n\} \leq h(n)$$

となるとき，T はh 従順であるという．エルブランモデルM はM(T)=Mなるh 従順なプログラムT があるとき，h 従順であるという．

Shapiroの枚挙型モデル推論アルゴリズムを下に示す．

h を帰納的関数とし，$T_1, T_2, \cdots$をL 上のすべてのプログラムの枚挙とする．

$S_{false}:=\{\Box\}$，$S_{true}:=\{\ \}$，$k:=1$

repeat

　次の事実 $F_n=\langle a, V\rangle$ を読む．

　$S_V:=S_V\cup\{a\}$

　while　$\exists b\in S_{false}$: $T\vdash_n b$ または

　　　　$\exists a_i\in S_{true}$: $T\nvdash_{h(i)} a_i$　　do

　　　　$k = k+1$とする．

　推測T_kを出力する．

forever.

このアルゴリズムについて，次の結果が知られている[16]．

(1) h を帰納的関数とし，MをLの h従順なエルブランモデルとする．このとき，枚挙型モデル推論アルゴリズムはMを極限において同定する．

(2) 枚挙型モデル推論アルゴリズムに対して一様に存在する帰納的関数h に対して，このアルゴリズムがエルブランモデルM を極限において同定するならば，M は h従順である．

この枚挙型のモデル推論アルゴリズムには，まだ非現実的なところがある．すなわち，現在の推測T_kと次の推測T_{k+1}との間には，通常の枚挙ではなんら関係はない．つまり，推測の候補の枚挙はこれから推論しようとしているモデルに無関係に定まっており，また当然それまでにアルゴリズムが受け取った事実とも無関係である．そのために，たとえば現時点で正しい推測に非常に近いところまで到達していたとしても，次に検査する候補は全く関係のないものであることもあるわけである．Shapiro はこうした枚挙によるアルゴリズムの効率の悪さを改良するために，矛盾点追跡アルゴリズムおよび，精密化演算子という二つの重要な手法を考案している．

矛盾点追跡アルゴリズムとは，推測が強すぎるとき，すなわちモデルM で偽である文を導出するとき，その推測の中の誤った文を特定するものである．このアルゴリズムを適用して，誤りである文を推測から取り除いた結果，逆に，推測が弱すぎるようになることがある，つまりモデルM で真である文を導出できなくなることがある．精密化演算子はこのとき起動させる．この演算子は，文p を入力として受取り，p よりも具体的な文q を返す．このq をp の精密化という．彼は，この二つの手法を用いて，次のような一般的な漸増的モデル推論アルゴリズムを得ている．

$T:=\{\Box\}$

repeat

　次の事実 $F_n=\langle a, V\rangle$ を読む．

　$S_V:=S_V\cup\{a\}$

　while　$\exists b\in S_{false}$: $T\vdash_n b$　　do

　　矛盾点追跡アルゴリズムを適用して

　　反駁された仮説をTから除去する．

　while　$\exists a_i\in S_{true}$: $T\nvdash_{h(i)} a_i$　　do

　　以前に反駁された仮説に対する精密化を

　　Tに加える．

　until　$\forall b\in S_{false}$: $T\nvdash_n b$ かつ

　　　　$\forall a_i\in S_{true}$: $T\vdash_{h(i)} a_i$.

　推測Tを出力する．

forever.

5．おわりに

本稿では，プログラムの自動合成や文法推論の理論的基礎を与える帰納推論について概観してきた．帰納推論については，このように，多くの理論的な研究成果が得られている．しかし，真の意味での応用可能性が帰納推論に期待できるようになったのは，ごく最近のことである．たとえば，4節で説明したShapiro のモデル推論システムは，多くの注目を浴びている．また，帰納推論は，単独に用いるのではなく，他の推論と併用して初めて，その真価を発揮できるであろう．その意味で，類推の機械化の研究は重要である[23]．

帰納推論の理論のごく一部だけを紹介してきた．詳しくは，この分野に対する概説である[3, 12, 21, 22, 24]等を参照されたい．

参考文献

[1] Angluin,D. : Finding Patterns Common to a Set of Strings,Proc. 11th Annual symp. Theory of Computing (1979),130-141.

[2] Angluin,D. : Inductive Inference of Formal Languages from Positive Data,Inform. & Contr. 45 (1980),117-135.

[3] Angluin,D., Smith,C.H. : Inductive Inference:Theory and Methods, Computing Surveys 15 (1983), 237-269.(大谷木重夫 訳, 帰納的推論―理論と方法, bit別冊 acm computing surveys '83 コンピュータ・サイエンス,共立出版 (1985),107-135.)

[4] Barzdin,J.M. : Prognostication of Automata and Functions, Information Processing 71,North-Holland (1972),81-84.

[5] Barzdin,J.M., Freivald,R.V. : On the Prediction of General Recursive Functions, Soviet Math. Dokl. 13 (1972), 1224-1228.

[6] Blum,L., Blum,M. : Toward a Mathematical Theory of Inductive Inference, Inform. & Contr. 28 (1975), 125-155.

[7] Case,J., Smith,C. : Comparison of Identification Criteria for Machine Inductive Inference,Theoretical Computer Science 25 (1983), 193-220.

[8] Gold,E.M. : Limiting Recursion, J. Symbolic Logic 30 (1965), 28-48.

[9] Gold,E.M. : Language Identification in the Limit, Inform. & Contr. 10 (1967), 447-474.

[10] Jantke,K.P., Beick,H.-R. : Combining Posturates of Naturalness in Inductive Inference, Elektron. Informationsverarb. Kybernet. 17 (1981), 465-484.

[11] Jantke,K.P.: Polynomial Time Inference of General Pattern Languages, Proc. STACS'84, LNCS-166, Springer-Verlag (1984),314-325.

[12] Klette,R., Wiehagen,R. : Research in the Theory of Inductive Infernce by GDR Mathematicians — A Survey, Information Sciences 22 (1980), 149-169.

[13] Nix,R.P.: Editing By Example, RR-280, Dept. Comp. Sci. Yale Univ., 1983.

[14] Popper,K.R. : The Logic of Scienfific Discovery, Basic Books, New York, 1959. (大内義一,森博　訳,科学的発見の論理(上)(下),恒星社　厚生閣,1971.)

[15] Popper,K.R. :Conjectures and Refutations :the Growth of Scientific Knowledge, Harper Torch Books, New York, 1968. (藤本隆志,石垣壽郎,森博　訳,推測と反駁―科学的知識の発展,法政大学出版局,1980.)

[16] Shapiro,E.Y. : Inductive Inference of Theories From Facts, TR-192, Dept. Comp. Sci. Yale Univ., 1981.(有川節夫　訳,知識の帰納的推論, 共立出版,1986.)

[17] Shapiro,E.Y. : Algorithmic Program Debugging, The MIT Press, 1983.

[18] Shinohara,T. : Polynomial Time Inference of Pattern Languages and Its Application, Proc. 7th IBM Symp. Math. Found. of Comp. Sci. (1982), 191-209.

[19] Shinohara,T. : Polynomial Time Inference of Extended Regular Pattern Languages, Proc. RIMS Symp. Software Sci. Eng., LNCS-147, Springer-Verlag (1982),115-127.

[20] Wiehagen,R. : Limes-Erkennung rekursiver Funktionen durch spezielle Strategien, Elektron. Informationsverarb. Kybernet. 12 (1976), 93-99.

[21] 有川, 篠原, 宮原 : 帰納推論の理論, 大須賀, 佐伯編, 知識の獲得と学習, オーム社, 1987,147-197.

[22] 有川, 原口 : 例によるプログラムの合成, 大須賀, 佐伯編, 知識の獲得と学習, オーム社, 1987,199-219.

[23] 有川, 原口 : 類推の理論, 大須賀, 佐伯編, 知識の獲得と学習, オーム社, 1987, 221-251.

[24] 横森 貴: 帰納推論アルゴリズム, 数理科学, 300 (1988), 23-29.