数理解析研究所講究録
第1599巻 2008年 50-56

自明でない法則を用いた形式言語における概念分化

真理大學 数理學院 資訊科學系 植村 仁 (Jin Uemura)
Aletheia University
College of Mathematical Sciences, Computer and Information Science

1 序

本論文は形式言語の正例からの学習 [2] に 2 つの疑問を投げかけ, それらを解決するための学習の枠組みを提案する.

形式言語の正例からの学習のうち, 入力される正例の集合を説明する仮説を複数の言語で構成するものがある [4]. その際学習対象となる言語族によっては, 仮説を構成する言語の数に上限を設定しなければならないものもある [1]. この上限を定める定数は例を説明するための言語を無限に増やさないために必要になるものであるが, この定数は任意の正整数でよく, どのような数であるべきかという議論が欠落している. この定数によらない言語の学習についての研究は少ない [3]. このような学習において複数言語を適切に組み合わせ仮説を構成する一般的方法はないだろうか. これが第一の疑問である.

第二の疑問は, 正例の集合をカバーするような仮説を出力すればそれで十分であるかどうかというものである. 人間が学習する際には例を一通りに説明すればよいというわけではなく, その内部にどのような構造があるのかということに立ち入ることもある. 言い換えるならば, 機械学習において例の集合を詳しく説明する, 十分に分化された仮説を生成するためにはどのようにすればよいだろうかということである.

これらの疑問に 1 つの回答を与えるため, 本論文が提案する枠組みでは入力される例を同じ性質に分け階層化するように, 複数言語による仮説を構成する. この際に生じる問題を解決するため, 自明でない法則というものを導入する.

本論文では諸定義の後, まず自明でない法則を導入し, 入力例の集合を十分に分化することについて議論する. その後学習の枠組みとアルゴリズムを定義し, この分化を伴う学習が可能となるための十分条件について議論する. この十分条件は主に 2 つの条件からなり, その 1 つは正例からの学習でよく知られた学習の十分条件である有限の弾力性 [4] であることを述べる. 最後に本研究で未解決の問題について議論する.

2 諸定義

基本的な定義と, 例で用いる言語の族を定義する.

2.1 諸定義

アルファベットとは定数記号の有限集合であり, Σ で表す. 定数記号は主に $a, b, c, \ldots$ で表す. 変数記号を $X, Y, Z, X', X_0, X_1, \ldots$ などで表すことにしよう. 変数記号の集合はアルファベットと交わりをもたない可算無限集合であるとする. 長さ 0 の文字列を ε で表す.

言語と表記する際には Σ 上の言語であると暗黙に仮定する. Σ^* は Σ 上のすべての文字列からなる言語となる. サンプルとは, Σ 上の語の有限集合である. 集合 S の濃度を $\sharp S$ で表す.

2.2 部分列パターン言語

$m \geq 1$ とし, $X_0, X_1, \ldots, X_m$ を変数記号, $a_1, a_2, \ldots, a_m$ を Σ に含まれる定数記号とする. 部分列パターンとは, $X_0 a_1 X_1 \cdots a_m X_m$ または X_0 となる有限長の文字列である. 部分列パターン p の言語 $L(p)$ を以下のように定義する.

$$p = X_0 a_1 X_1 \cdots a_m X_m \quad \text{のとき}, L(p) = \Sigma^* \{a_1\} \Sigma^* \cdots \Sigma^* \{a_m\} \Sigma^*$$
$$p = X_0 \quad \text{のとき}, L(p) = \Sigma^*$$

定数文字列 x が 定数文字列 y の部分文字列であるとは, y の文字をいくつか消去することで x が得られるということである. $L(p)$ は, p の変数を消去して得られる定数文字列を部分文字列としてもつ定数文字列全体からなる言語となる. 例えば, $\Sigma = \{a, b, c\}$ のとき,

$$L(XaY) = \{a, aa, ba, ab, ca, ac, aaa, baa, caa, aab, aac, \ldots\}$$

となる.

3 自明でない法則と例の分化

3.1 例を階層化し分類する際の問題点

サンプル E を同じ性質をもつものに分けるとはどういうことだろうか. $\mathcal{C}$ を例を説明する言語の族としよう. 2 つの定数文字列 x, y が同じ性質をもつということを表現するには, 例えば

$$\forall L \in \mathcal{C}, x \in L \iff y \in L$$

というものが考えられるだろう.

今, サンプル E を

$$E = \{a, b, c, \varepsilon\}\{a\}\{a, b, c, \varepsilon\} = \{a, aa, ba, ab, ca, ac, aaa, baa, aab, caa, \ldots, cac\}$$

として, 複数の部分列パターン言語で例を説明してみると, 次のようなものが挙げられる.

$$E \subseteq L(XaY)$$
$$E - \{a\} \subseteq L(XaYaZ) \cup L(XaYbZ) \cup L(XbYaZ) \cup L(XaYcZ) \cup L(XcYaZ)$$
$$E - \{a, aa, ab, ba, ac, ca\}$$
$$\subseteq L(XaYaZaX') \cup L(XbYaZaX') \cup L(XaYaZbX') \cup \cdots \cup L(XcYaZcX')$$

特に元 bab に焦点を当てると, $bab \in L(XbYaZbX')$ となり, この言語は $L(XbYaZ)$, $L(XaYbZ)$, $L(XbYbZ)$ に包含され, これらはさらに $L(XaY)$ に包含されるというようになっている.

このように, サンプルの各元は個々の要素にまで分けられてしまい, 説明する言語に関する記述がサンプルの文字列長の合計よりもはるかに長くなる. このようなことは正規言語やパターン言語などの既知の言語族においても同様に起こる. 2 つの元を, 言語族の任意の言語に所属するかどうかで分類するためにはこの問題を解決しなければならない.

本論文ではこれを解決するため, 単純な包含関係の代わりに自明でない法則という概念を導入しこの問題を解決する.

3.2 自明でない法則の定義

言語 L によって説明される 言語 E の部分を $Ex(L,E)$ $(= L \cap E)$ と表す. E は説明される側である「例の集合」を表す.

定義 3.1 (自明でない法則). $\mathcal{C}$ を言語族, $L_1, L_2 \in \mathcal{L}$, E を言語とする. $L \rightarrow_E L'$ を以下のように定義する.

$$
\begin{aligned}
& L \rightarrow_E L' \\
\iff\ & Ex(L,E) \neq \emptyset, \\
& Ex(L,E) \subsetneq Ex(L',E), \\
& L \not\subseteq L' \text{ かつ } L' \not\subseteq L
\end{aligned}
$$

L, L' は包含関係に対し比較不能となること, L' に属し L に属さない E の元が存在することに注意せよ.

3.3 例の分化

定義 3.2 (分化・十分に分化). $\mathcal{C}$ を言語族, $F \subseteq \mathcal{C}$, E を空でない言語とする (有限言語とは限らない). $\mathcal{C}$ を用いて言語の集合 F が E を分化するとは以下の 1-4 を満たすことであり, 十分に分化するとは 1-5 を満たすことである.

1. $\exists! \hat{L} \in F$ *s.t.* $E = Ex(\hat{L}, E)$
2. F の任意の 2 つの言語は互いに包含関係にない
3. $\forall L, L' \in F$ $(L \neq L')$ に対して, $Ex(L,E) \neq Ex(L',E)$
4. $\forall L \in F$ $(L \neq \hat{L}), \exists (\hat{L} =) L_{i_0}, L_{i_1}, \ldots, L_{i_n} (= L) \in F, L_{i_{j+1}} \rightarrow_E L_{i_j}$ $(j = 0, \ldots, n-1)$
5. $\forall L' \in \mathcal{C} - F, \forall L \in F$, L, L' は互いに包含関係にないか共通部分をもたず, $L' \rightarrow_E L$ とならない

となることである.

4 分化学習

4.1 分化を伴う学習に関する定義

帰納的言語の添え字付族

言語族 $C = L_0, L_1, \cdots$ が帰納的言語の添字付き族であるとは, 次のような計算可能関数 $f : N \times \Sigma^* \to \{0,1\}$ が存在することをいう.

$$f(i,x) = \begin{cases} 1, & \text{if } x \in L_i \\ 0, & \text{if } x \notin L_i \end{cases}$$

添字 i は言語を表すオートマトンや形式文法, そしてパターンなどを (プログラムのゲーデル数のように) 自然数に対応づけることにより得られる数である. 以下, 言語族 $\mathcal{L}$ は帰納的言語の添え字付き族とする.

正提示

文字列 x が L の正例であるとは, x が L の元となることである. 文字列の無限列 $x_0, x_1, \cdots$ が言語 L の正提示であるとは, $\{x_n \mid n \geq 0\} = L$ が成立することである. 文字列の無限列 $x_0, x_1, \cdots$ の 0 番目から n 番目までの有限列を $\sigma[n]$ で表し, 初期断片と呼ぶ.

学習機械

学習機械 M とは, 次々に入力を要求し, 次々に出力を生成する実行的手続きのことであり, M の出力を推測と呼ぶ. 文字列の無限列 σ に対して, その初期断片 $\sigma[n]$ が入力された後, M が生成する出力を $M(\sigma[n])$ で表す. 推論アルゴリズム M が入力の列 σ に対して, 添字 $g \in N$ に収束するとは, ある $m \in N$ が存在し, 任意の $n \geq m$ に対して, $M(\sigma[n]) = g$ となることをいう.

正例からの分化学習

学習機械が言語 L を 言語族 C を用い極限において正例から分化学習するとは, 1) L の正例を次々に入力として受け取り, 蓄積された例を C を用いて分化する言語の有限集合を推測として出力し続け, 2) ある時点から十分に分化されたもののみを出力し, 3) その推測の無限列が収束することが, 4) L の任意の正提示に対し成立することである.

また, 言語族 C' が C を用いて正例から分化学習可能であるとは, ある学習機械が存在して, 任意の言語 $L \in C'$ を正例から極限において分化学習することである.

4.2 分化学習アルゴリズム

言語 L を C' の元とし, $C = L_0, L_1, \ldots$ としよう. 以下正例からの分化学習をするアルゴリズムを挙げる. その正当性は節の最後で証明する.

入力: ある言語 L の正提示
出力: C を用いて入力された例の集合 E を分化する C の有限部分集合 F の列
$R := \emptyset$
$E := \emptyset$
for $u = 1$ to ∞
begin
  Read the next input and add to E;
  Let $F_1, \ldots, F_k$ be the sets of indexes of languages of C satisfying
    1) $\sharp F_m \leq u$ and 2) $\forall n \in F_m,\ n \leq u$ $(m = 1, \ldots, k)$;
  Let $R := F_1, \ldots, F_k$;
  for $j = 1$ to k
  begin
    if validDiff(F_j,E) is false then remove F_j from R;
    $ex[j]$:= exDiv(F_j,E)
  end;
  for $j = 1$ to k
    if fine($ex[t]$, $ex[j]$) is true for some $t = 1, \ldots, k,\ t \neq j$;
      then check F_j;
  output the first $F \in R$ without a check
end

ただし, validDiff(F_j,E) は F_j が E を分化していれば真, そうでなければ偽を返す関数とする. また, exDiv(F_j,E) は 以下の条件を満たす $\{S_1, \ldots, S_v\}$ を返す.

1. $S_1 \cup \cdots \cup S_v = E$
2. $S_1, \ldots, S_v$ の任意の 2 つは交わりをもたない
3. $\forall S_w$ $(w = 1, \ldots, v)$, $\forall x, y \in S_w$, $\forall h \in F_j$, $x \in L_h \iff y \in L_h$

fine($ex[t]$, $ex[j]$) は $ex[t]$ が $ex[j]$ より細分化されていれば真, そうでなければ偽を返す. つまり, $ex[j]$ の元のいくつかの和をとることで $ex[t]$ を構成できるときに真を返す.

4.3 分化学習の十分条件

定義 4.1 (有限の弾力性 [4]). 言語族 C が有限の弾力性をもつとは, 以下のような条件を満たす語の無限列 $x_0, x_1, \cdots$ と言語の無限列 $L_0, L_1, \cdots \in C$ が存在しないことである.

$$\begin{aligned} \{x_0, x_1, \cdots, x_n\} &\subseteq L_n \text{ かつ} \\ x_{n+1} &\notin L_n \ (n \in N). \end{aligned}$$

補題 4.2 (有限の弾力性との関係). E を空でない言語とする (有限言語とは限らない). 言語

族 C が有限の弾力性をもち, $Ex(L_{t_0}, E) \neq \emptyset$ であるとき,

$$L_{t_0}, L_{t_1}, \ldots \in C$$
$$L_{t_i} \rightarrow_E L_{t_{i+1}} \quad (i \geq 1)$$

となるような無限列は存在しない.

証明. 背理法により証明する. C が有限の弾力性をもち, かつ $L_{t_0} \rightarrow_E L_{t_1} \rightarrow_E \cdots$ となるような言語の無限列が存在すると仮定する.

$Ex(L_{t_0}, E) \neq \emptyset$ より, $Ex(L_{t_0}, E)$ の元を 1 つ取り, x_0 とすることができる. $i \in N$ に対し, $L_{t_i} \rightarrow_E L_{t_{i+1}}$ $(i \in N)$ の定義より, $L_{t_{i+1}}$ に属し, L_{t_i} に属さない E の元が存在する. これを x_{i+1} とする. $Ex(L_{t_0}, E) \subsetneq Ex(L_{t_0}, E) \subsetneq \cdots \subsetneq Ex(L_{t_i}, E)$ であるから, $\{x_0, \ldots, x_i\} \subseteq L_{t_i}$ となる. ところがこれは C が有限の弾力性をもつことになり矛盾する. ■

この補題は, 入力の無限列に対しても, それを説明する言語が自明でない法則の列を成すとき, その列が有限となることを表している.

定義 4.3 (有限共有性). 言語族 C が有限共有性をもつとは, 任意の言語 $L \in C$ に対して,

$$\sharp\{L' \mid L \cap L' \neq \emptyset, L \not\subseteq L', L' \not\subseteq L\} < \infty$$

が成立することである.

定理 4.4 (分化学習可能であるための十分条件). 言語族 C' が帰納的言語の添え字付き族 C を用いて正例から分化学習可能であるためには, C が以下を満たせばよい.

1. 有限共有性をもつ
2. 有限の弾力性もつ
3. 各言語の対の包含関係が決定可能である

(C' に条件はない.)

証明. 略証のみを与える. まず最も外側のループ内の計算可能性について吟味する. validDiff は自明でない法則と分化の定義より, 有限言語の包含関係と言語の包含関係が計算可能であれば計算可能となる. exDiv は F_j と E が有限集合であることから計算可能であることが分かる. fine は有限言語の有限の組み合わせを調べることで計算できるのでこれも計算可能である.

正例からの分化学習の 2) ある時点から十分に分化されたもののみを出力し, 3) その出力の無限列が収束することについては有限共有性と有限の弾力性から, u が十分に大きくなった時点から出力が十分に分化されたものとなる. ■

5 結論

本論文の分類学習の定式化では学習対象の言語を同定せず, 十分に分化することを学習の成功基準としたため, 学習対象となる言語族には制限がない形で学習可能性に関する結果を導く

ことができた. しかし, 学習対象を分化し, かつ同定する問題には触れなかった. また分化学習の十分条件を 1 つ発見するに至ったが, より弱い十分条件や必要十分条件は見つかっていない. 本論文で挙げた十分条件を満たす言語族の発見がこの問題の足がかりになると期待される.

自明でない法則の前件と後件にはそれぞれ 1 つの言語しかないもののみを扱った. 積または和を用い, 前件もしくは後件に複数の言語が出現するような自明でない法則を用いた場合にどのような性質が立ち現れるかについては将来の課題としたい.

参考文献

[1] H. Arimura, T. Shinohara and S. Otsuki: *Finding minimal generalizations for unions of pattern languages and its application to inductive inference from positive data*, Lecture Notes in Computer Science, vol. **775**, 646-660, (1994).

[2] E. M. Gold: *Language identification in the limit*, Information and Control, vol. **10**, 447-474, (1967).

[3] T. Shinohara and H. Arimura: *Inductive inference of unbounded unions of pattern languages from positive data*, Proc. the 7th International Workshop on Algorithmic Learning Theory, Lecture Notes in Artificial Intelligence, **1160**, 256-271(1996).

[4] K.Wright: *Identification of unions of languages drawn from positive data*, Proc. the 2nd Annual Workshop on Computational Learning Theory, 328-333, (1989)