数理解析研究所講究録
1041 巻 1998 年 183-190

1 変数パタン言語の多項式時間オンライン学習

稲子 希望 (Nozomu Inago)　有村 博紀 (Hiroki Arimura)
九州大学 システム情報科学研究科 情報理学専攻
〒 816-8580 福岡県 春日市 春日公園 6-1
e-mails:{*inago,arim*}*@i.kyushu-u.ac.jp*

概要: 1 変数パタン言語の族は, 正例から極限において同定可能であることが知られている. しかし, この族が多項式時間オンライン学習可能であるかどうかは, 未解決の問題である. 本研究では, 1 変数パタン言語の族が多項式時間オンライン学習可能であることを示す. さらに, これを改良して, より効率のよいアルゴリズムを与える.

1 はじめに

本稿では, 1 変数パタン言語のオンライン学習可能性について考察する. オンライン学習 [5] では, 学習者は未知の概念の例を順にうけとり, 現在の仮説にしたがってその例が未知の概念のに含まれるかどうかを予測する. 予測が間違っていた場合は, 誤予測がおきたといい, 学習者は誤予測を起こした例にもとづいて仮説を更新しながら, 学習過程をくり返す. 学習の目標は, 予測過程における誤予測の回数をできるだけ小さくすることである.

学習対象は, 1 変数パタン言語の族である. パタンとは定数文字と変数からなる文字列であり, その言語は, パタン中の変数に空でない定数文字列を代入して得られるすべての定数文字列の集合である. 1 変数パタン言語は, $0x10x1$ のような高々一種類の変数を含むパタンによって生成される言語である. 1 変数パタン言語は, Angluin[1] によって導入されて以来, さまざまな学習モデルにおいてその学習可能性が詳細に調べられている [6, 3, 4]. しかし, この族が Littlestone[5] のモデルで多項式時間オンライン学習可能であるかどうかは, 未解決の問題である. そこで, 本稿では 1 変数パタン言語のオンライン学習可能性について調べる.

はじめに, 任意個の 1 変数パタン言語の積を仮説として用いた, 誤予測数 $O(n^4)$ の学習アルゴリズムを与える. ここに, n は誤予測を起こした最長の例のサイズである. このアルゴリズムは, Angluin の 1 変数パタンオートマトンを用いて仮説となる 1 変数パタン言語の積をコンパクトに表現する.

つぎに, 誤予測数が $O(m \log n)$ となる, さらに効率よい学習アルゴリズムを与える. ここに, m は未知パタンのサイズであり, n は誤予測を起こした最長の例のサイズである. このアルゴリズムは, 学習アルゴリズム WINNOW[5] で用いられている重み付多数決法のアイディアに基づいている.

2 準備

2.1 基本的定義

集合を A とする. A 上の文字列とは, A の要素の有限列 $s = a_1 a_2 \cdots a_n$ である. ここに, $1 \leq i \leq n$ について $a_i \in A$ である. A の長さは $n \geq 0$ である. A の部分集合 B に対して, $|s|_B$ で B の要素の文字列 s 中の総出現数を表す.

正整数を i, m とする. このとき, s の i 番目の文字を $s[i] = a_i$ と定義し, i 番目の文字から始まる長さ m の部分文字列を $sub(s, i, m) = a_i a_{i+1} \cdots a_{i+m-1}$, s の長さ m の接頭語を $pref(s, m) = a_1 a_2 \cdots a_m$ と定義する. s の接頭語全体の集合を $Pref(s) = \{pref(s, m) \mid 0 \leq m \leq |s|\}$ と書く.

空語を ε と書く. A 上の有限文字列のなす集合を A^* と書き, $A^+ = A^* - \{\varepsilon\}$ と定義する. A の要素数を $|A|$ と書き, 文字列 $s \in A^*$ の長さを $|s|$ と書く.

2.2 パタン言語

はじめに, Angluin (1980) にしたがって1変数パタン言語を定義する. Σ を**定数文字**のアルファベットとし, x を**変数**とする. 以下では, 常に $|\Sigma| \geq 2$ と仮定する. **1変数パタン** (one-variable pattern) は, 定数文字と変数 x からなる文字列 $p \in (\Sigma \cup \{x\})^+$ である. 1変数パタンのクラスを $P_1 = (\Sigma \cup \{x\})^+ - \Sigma^+$ で表す. 以後, 混乱しない限り単にパタンと呼ぶ.

1変数パタンを $p, q, r \in P_1$ とする. このとき, $p\{x := q\}$ で, パタン p 中の変数 x の出現すべてを q で置き換えて得られるパタンを表す. この置き換えを代入という. あるパタン r に対して $p\{x := r\} = q$ となるとき, q は p に**マッチする**といい, $q \preceq p$ と書く. p の言語を, p 中の変数 x に空でない定数文字列を代入して得られる文字列全体の集合 $L(p) = \{w \in \Sigma^+ \mid w \preceq p\}$ と定義する.

パタン $p \in P_1$ に対して, 三つ組 $\tau(p) = (I(p), J(p), K(p))$ を定義する. ここに, $I(p) = |p|_\Sigma$ は p 中の定数の出現数, $J(p) = |p|_x$ は変数の出現数, $K(p)$ は変数 x の最左出現位置である. 例えば, $\Sigma = \{0, 1\}$ 上のパタンを $p = 0x1xx0x$ とすると, $\tau(p) = (3, 4, 2)$ である. 非負整数 i, j, m に対して, 関数 h_m を $h_m(i, j) = i + j \cdot m$ と定義する.

添字が $\tau(p) = (i, j, k)$ である任意のパタンを $p \in P_1$ とすると, $h_m(i, j)$ は p の変数に長さ m の文字列を代入して得られる文字列の長さとなってる. つまり $|p\{x := t\}| = h_{|t|}(I(p), J(p))$ である. つぎの補題は基本的である.

補題 1 パタンを $p \in P_1$ とし, 文字列を $s, t \in \Sigma^*$ とする. このとき, *(1)* $p\{x := t\} = s$ が成立することと, *(2)* $h_{|t|}(I(p), J(p)) = |s|$ かつ, 任意の $1 \leq d \leq |p|$ と $q = pref(p, d-1)$ に対して,

$$\begin{cases} s[h+1] = p[d] & (p[d] \in \Sigma \text{のとき}) \\ sub(s, h+1, |t|) = t & (p[d] = x \text{のとき}) \end{cases}$$

が成立することは等価である. ここに $h = h_{|t|}(I(q), J(q))$ である.

2.3 学習モデル

本稿では, 学習モデルとしてオンライン学習モデル [5] を用いる. Σ を定数記号のアルファベットとし, 未知の1変数パタンを $p \in P_1$ とする. 任意の有限文字列 $s \in \Sigma^+$ を例という. 例 s は, $s \in L(p)$ のとき正例といい, $s \notin L(p)$ のとき負例という.

オンライン学習では, つぎのような試行をくりかえしながら学習が進行する. 一つの試行は, つぎの3つの段階からなる. まず学習者は, 例を1個うけとり, つぎに現在の仮説をもとにして, 与えられた例が正例であるか負例であるかにしたがって, **予測** $d \in \{0, 1\}$ を出力する. 最後に予測の**正否** $r \in \{0, 1\}$ をうけとり, 予測が正しくなかったとき $(r = 0)$ は仮説を更新する. 予測が正しくないことを**誤予測**という. 学習アルゴリズム $\mathcal{A}$ が一変数パタン言語を**多項式時間オンライン学習**するとは, 学習の任意の時点において, 誤予測の回数 MB と1回の試行の計算時間が今までにうけとった最長の例の長さ n と未知パタン p のサイズ m の多項式 $poly(m, n)$ でおさえられることをいう.

ある概念クラスが多項式時間オンライン学習可能ならば, 多項式時間評価可能な任意の仮説を用いて等価性質問学習 [2] や PAC 学習 [7] でも学習可能である [2, 5].

3 1変数パタンオートマトン

共通のアルファベット Δ をもつ二つの有限オートマトン $A_i = \langle V_i, \Delta, \delta_i, q_0, F_i \rangle$ $(i = 1, 2)$ に対して, 包含関係 $\subseteq$ と積の演算 $\cap$ を定義する. ここで R_δ は, $R_\delta = \{(q, c, r) \mid \delta(q, c) = r\}$ で定義される遷移関係である.

- $A_1 \subseteq A_2 \Leftrightarrow V_1 \subseteq V_2, F_1 \subseteq F_2, R_{\delta_1} \subseteq R_{\delta_2}$.
- $A_1 \cap A_2 = \langle V_1 \cap V_2, \Delta, \delta, q_0, F_1 \cap F_2 \rangle$ *such that* $R_\delta = R_{\delta_1} \cap R_{\delta_2}$.

定義より, $A_1 \subseteq A_2$ のとき $L(A_1) \subseteq L(A_2)$ であり, $L(A_1 \cap A_2) \subseteq L(A_1) \cap L(A_2)$ であるが, 逆は一般に成立しない.

Angluin[1] は, 与えられた文字列を生成するような1変数パタン全体をコンパクトに表現する手段として, **1変数パタンオートマトン**を導入した. 文字列を $s \in \Sigma^+$ と, その部分文字列 $t \in \Sigma^+$ に対して, 変数 x に t を代入すると s になるような1変数パタン全体の集合を $P_1(s;t) = \{p \in P_1 \mid p\{x := t\} = s\}$ と定義する. つぎに, $P_1(s;t)$ を言語として受理する有限オートマトン $A(s;t)$ を定義する.

定義 1 有限オートマトン $A(s;t) = \langle V, \Delta, \delta, q_0, F \rangle$ をつぎのように定義する.

- 状態 $(i,j) \in V = \{0,\ldots,|s|\} \times \{0,\ldots,|s|\}$. $A(s;t)$ の状態 (i,j) は, 初期状態からこの状態に到達するまでに i 個の定数と j 個の変数を読みこんだことを意味する.
- アルファベット $\Delta = \Sigma \cup \{x\}$.
- 遷移関数 $\delta : V \times \Delta \to V$:

$$\delta((i,j),c) = (i+1,j) \quad if\ s[h_{|t|}(i,j)\ +1] = c\ (c \in \Sigma),$$
$$\delta((i,j),x) = (i,j+1) \quad if\ sub(s,\ h_{|t|}(i,j)\ +1,\ |t|) = t.$$

- 初期状態 $q_0 = (0,0)$.
- 最終状態の集合 $F = \{(i,j) \in V \mid h_{|t|}(i,j) = |s|, j \geq 1\}$.

状態 $\delta(q,p)$ が定義されているとき, $\delta(q,p)\downarrow$ と書き, それ以外のとき $\delta(q,p) = \perp$ と書く. δ は, 通常のやり方で $\delta : V \times \Delta^* \to V$ へ拡張する.

定理 2 (Angluin 1980[1]) $L(A(s;t)) = P_1(s;t)$.

長さ n の定数文字列を s とおく. 3つ組の集合 F_n を, $F_n = \{(i,j,k) \mid 0 \leq i \leq n-1,\ 1 \leq j \leq n,\ 1 \leq k \leq i+1,\ j|(n-i)\}$ と定義する. また, 3つ組 (I,J,K) に対して, $\psi(s;I,J,K) = sub(s,k,(|s|-I)/J)$ と定義する. このとき, 任意のパタン p に対して, p が s にマッチするならば, $\tau(p) \in F_n$ が成立する. さらに, もしある文字列 t に対して $p\{x := t\} = s$ となるならば, $t = \psi(s;\tau(p))$ が成立する.

定義 2 先の s に対して, $(I,J,K) \in F_n$ と仮定, $t = \psi(s;\tau(p))$ とおく. このとき, 有限オートマトン $A(s;t)$ に以下の操作をおこなって得られる有限オートマトンを $B(s;I,J,K)$ で表す.

- $(i,j) \in \{0,\ldots,I\} \times \{0,\ldots,J\}$ 以外の状態を取り除く.
- $\delta((u,0),x) = (u,1)$ $(u = 0,\ldots,K-2)$ の遷移を取り除く.
- $\delta((K-1,0),c) = (K,0)$ $(c \in \Sigma)$ の遷移を取り除く.
- 最終状態は (I,J) のみにする.

オートマトン C は, ある $s \in \Sigma^*$ と $(I,J,K) \in F_n$,$n = |s|$, に対して $C \subseteq B(s;I,J,K)$ をみたすとき, 1変数パタンオートマトンという. このとき, 明らかに $L(C)$ は同じ3つ組 τ をもつパタンしか含まない. この3つ組を, C のインデックスといい $\tau(C)$ で表す.

この定義より, 明らかに $B(s;I,J,K) \subseteq A(s;t)$ である. $B(s;I,J,K)$ は $A(s;t)$ の受理する文字列を $\tau(p) = (I,J,K)$ となる p だけに限定したものなので, $B(s;I,J,K)$ は $1pa$ であり, $\tau(B(s;I,J,K)) = (I,J,K)$ がわかる. また, もし $p \in L(A(s;t))$ ならば $p \in L(B(s;\tau(p)))$ である.

定義 3 長さ n の定数文字列 s に対して, オートマトンの集合 $\mathcal{B}(s)$ をつぎのように定義する.

$$\mathcal{B}(s) = \{B(s;I,J,K) \mid (I,J,K) \in F_n\}.$$

$P_1(s) = \{p \in P_1 \mid s \in L(p)\}$ と定義する．また，$1pa$ の集合 $\mathcal{B}$ に対して $\mathcal{B}$ のあらわす言語を $\mathcal{L}(\mathcal{B}) = \bigcup_{B \in \mathcal{B}} \mathrm{L}(B)$ と定義する．

定理 3 (Angluin 1980[1]) $\mathcal{L}(\mathcal{B}(s)) = P_1(s)$,

補題 4 長さ n の文字列 s に対して，$\mathcal{B}(s)$ は $O(n^4)$ 時間で構成可能である．

補題 5 (Angluin 1980[1]) パタンオートマトン A_1, A_2 に対して，$L(A_1) \cap L(A_2) = L(A_1 \cap A_2)$.

演算 $\cap$ をパタンオートマトンの集合上につぎのように拡張する:

$$\mathcal{B}_1 \sqcap \mathcal{B}_2 = \{B_1 \cap B_2 | B_1 \in \mathcal{B}_1, B_2 \in \mathcal{B}_2, \tau(B_1) = \tau(B_2)\}.$$

補題 6 $\mathcal{L}(\mathcal{B}_1 \sqcap \mathcal{B}_2) = \mathcal{L}(\mathcal{B}_1) \cap \mathcal{L}(\mathcal{B}_2)$.

4 学習アルゴリズム

本節では，1 変数パタンの多項式時間オンライン学習について議論する．目標クラスは 1 変数パタンのクラス P_1 であり，目標となる未知パタンを $p_* \in P_1$ とする．仮説として $1pa$ の集合 $\mathcal{B}_h$ を用いる．ただし $\mathcal{B}_h$ 中の全ての $1pa$ B は常に，どのような入力に対しても初期状態から到達できないような遷移は除去されているようにしておく．仮説 $\mathcal{B}_h$ の表す仮説言語を $M(\mathcal{B}_h)$ とする．ここに $M(\mathcal{B}_h)$ は，$\mathcal{B}_h$ が保持しているすべてのパタン言語の積集合であるる．すなわち，$M(\mathcal{B}_h) = \bigcap_{p \in \mathcal{L}(\mathcal{B}_h)} L(p)$ である．$\mathcal{B}_\perp$ を，任意の $1pa$ の集合 $\mathcal{B}$ に対して $\mathcal{B}_\perp \sqcap \mathcal{B} = \mathcal{B}$ であり，かつ $M(\mathcal{B}_\perp) = \emptyset$ となるような特別な $1pa$ の集合とすると，学習アルゴリズムはつぎのようになる．

algorithm $LEARN\_P_1$
入力: 例の列 $s_1, s_2, \ldots$, 正否の列 $r_1, r_2, \ldots$
出力: 予測の列 $d_1, d_2, \ldots$
begin
  $\mathcal{B}_h := \mathcal{B}_\perp$
  for $i := 1$ **to** ∞ **do**
  begin
    例 $s_i \in \Sigma^+$ をうけとる
    if $s_i \in M(\mathcal{B}_h)$ **then** $d_i := 1$ **else** $d_i := 0$
      予測 d_i を出力
    d_i の正否 $r_i \in \{0, 1\}$ をうけとる
      if $r_i = 0$ **then** $\mathcal{B}_h := \mathcal{B}_h \sqcap \mathcal{B}(s)$ { 誤予測のとき仮説を更新 }
  end
end

このアルゴリズムにおいて，つねに $p_* \in \mathcal{L}(\mathcal{B}_h)$ であり，誤予測がおきたときの例は必ず正例となっている．

補題 7 学習の任意の時点において，そのときの仮説を $\mathcal{B}_h$ とすると，$M(\mathcal{B}_h)$ はそれまでうけとったすべての正例を含み，すべての負例を含まない．

予測

うけとった例 $s \in \Sigma^+$ に対して予測をおこなう．$s \in M(P_h)$ のとき，すなわち任意の $p \in \mathcal{L}(\mathcal{B}_h)$ に対して $s \preceq p$ のとき s は正例であると予測し，そうでないとき負例であると予測する．まず，ある $1pa$ $B \in \mathcal{B}_h$ に着目し，任意の $p \in L(B)$ に対して $s \preceq p$ であるかを調べる．下記のような手続きにより関数 $T : V \to \{0, 1\}$ の値を計算する．

procedure *predict*
入力：非冗長な1*pa* B, 定数文字列 $s \in \Sigma^+$
出力：$if\ \forall p \in L(B) : s \preceq p \quad then \quad 1$
$else \quad 0$
begin
$(I,J,K) = \tau(B)$
if $(I,J,K) \notin F_{|s|}$ **then rerurn**(0)
$t = \psi(s;I,J,K)$
for $j = 0$ **to** J **do**
for $i = 0$ **to** I **do**
begin
つぎのいずれかの条件を満たすとき $c_{i,j} = 1$ それ以外のとき $c_{i,j} = 0$
(a)$i = 0$
(b)$\forall c \in \Sigma : \delta((i-1,j),c) = \perp$
(c)$T(i-1,j) = 1 \ \wedge \ s[h_{|t|}(i-1,j) \ +1] = b \quad such\ that\ \delta((i-1,j),b) = (i,j)$
つぎのいずれかの条件を満たすとき $x_{i,j} = 1$ それ以外のとき $x_{i,j} = 0$
(d)$j = 0$
(e)$\delta((i,j-1),x) = \perp$
(f)$T(i,j-1) = 1 \ \wedge \ sub(s,\ h_{|t|}(i,j-1) \ +1,\ |t|) = t$
$T(i,j) = c_{i,j} \ \wedge \ x_{i,j}$
end
rerurn$(T(I,J))$
end

このとき, $D(i,j) = \{p \in \Delta^* \mid \delta((0,0),p) = (i,j)\}$ とすると, 下の補題が成り立つ.

補題 8 $(I,J,K) \in F_{|s|}$ とすると, 任意の状態 (i,j) に対して,

$$T(i,j) = 1 \ \Leftrightarrow \ \forall p \in D(i,j) : p\{x := t\} = pref(s,h)\ such\ that\ h = h_{|t|}(i,j).$$

証明: $n = i + j$ $(i = 0,\ldots,I,\ j = 0,\ldots,J)$ に関する帰納法で証明する.
$n = 0$ のとき $T(i,j) = 1$ であり, $\varepsilon\{x := t\} = pref(s,0)$ である. よって成り立つ.
$1 \leq n \leq I + J$ のとき. $i' + j' = n - 1$ であるような任意の状態 (i',j') に対して, 命題が成り立っていると仮定する.
$(\Rightarrow)$ $i + j = n$ となるような任意の状態 (i,j) に対して, $T(i,j) = 1$ とする. 一般に, 状態 (i,j) が遷移先となっているのは, $(i-1,j)$ からの定数による遷移と $(i,j-1)$ からの変数による遷移の 2 種類がある. 定数による遷移があるときは $T(i-1,j) = 1$ なので $p\{x := t\} = pref(s,h_{|t|}(i-1,j))$ であり, かつ $\delta((i-1,j),b) = (i,j)$ に対して $s[h_{|t|}(i-1,j) \ +1] = b$ である. 同様に変数による遷移があるときは $p\{x := t\} = pref(s,h_{|t|}(i,j-1))$ かつ $sub(s,\ h_{|t|}(i,j-1) \ +1,\ |t|) = t$ となっているので, 成り立つ.
$(\Leftarrow)$ $i + j = n$ となるような任意の状態 (i,j) に対して,

$$\forall p \in D(i,j) : p\{x := t\} = pref(s,h)\ such\ that\ h = h_{|t|}(i,j)$$

とする. 任意の $p \in D(i,j)$ に対して $p = q \cdot a$ $(q \in \Delta^*, a \in \Delta)$ とすると, $a \in \Sigma$ のとき $q \in D(i-1,j)$ であり, $a = x$ のとき $q \in D(i,j-1)$ である. $q\{x := t\} = pref(s,h_{|t|}(i(q),j(q)))$ なので $T(i(q),j(q)) = 1$ がわかる. $a \in \Sigma$ ならば 遷移 $\delta((i-1,j),b) = (i,j)$ に対して $a = b$ であり, かつ $s[h_{|t|}(i-1,j)+1] = b$ であるから, 遷移 $\delta((i-1,j),b) = (i,j)$ が存在するときは $a = s[h_{|t|}(i-1,j)+1]$ がわかる. 同様に $a = x$ ならば, 遷移 $\delta((i-1,j),b) = (i,j)$ が存在するときは $sub(s,h_{|t|}(i,j-1)+1,|t|) = t$ となっている. ■

$h_{|t|}(I,J) = |s|$ なので, 補題 8 より,

$$predict(B,s) = 1 \ \Leftrightarrow \ \forall p \in L(B) : p\{x := t\} = s$$

がわかる.

補題 9 ある試行においてうけとった例 s の長さを n とすると, 予測は $O(n^4)$ 時間で計算できる.

証明: 現在の $1pa$ の集合を $\mathcal{B}_h$ とする. ある $B \in \mathcal{B}_h$ に対して $\tau(B) = (I, J, K)$ とすると $(I, J, K) \in F_n$ のときのみ $predict(B, s)$ の計算に高々 $O(n + I \cdot J) = O(n \cdot J)$ 時間必要である. よって任意の $1pa$ $B \in \mathcal{B}$ に対して $predict(B, s)$ を計算する時間は高々 $\sum_{(I,J,K)\in F_n} O(n \cdot J) = O(n^4)$ である. ■

仮設の更新

補題 10 ある試行においてうけとった例 s の長さを n とすると, 仮設の更新時間は $O(n^4)$ である.

証明: 現在の $1pa$ の集合を $\mathcal{B}_h$ とする. まず $\mathcal{B}(s)$ の計算時間は定理 4 より $O(n^4)$ である. $B \in \mathcal{B}_h$, $\tau(B) = (I, J, K)$ とすると $(I, J, K) \in F_n$ の $1pa$ B に対してのみ $B \cap B(s; I, J, K)$ を計算する. B および $B(s; I, J, K)$ の状態数は $O(I \cdot J)$ なので, $B \cap B(s; I, J, K)$ の計算時間は $O(I \cdot J)$ となる. よって $\mathcal{B}_h \sqcap \mathcal{B}(s)$ の計算時間は, $\sum_{(I,J,K)\in F_n} O(I \cdot J) = O(n^4)$ である. ■

誤予測の回数

補題 11 $1pa$ B, 定数文字列 $s \in \Sigma^+$ に対して $\tau(B) = (I, J, K)$, $(I, J, K) \in F_{|s|}$ とし, $1pa$ $B, B' = B \cap B(s; I, J, K)$ の遷移関数をそれぞれ δ, δ' とする. このとき, $predict(B, s) = 0$ ならば $R_{\delta'} \subset R_\delta$ である.

証明: $predict(B, s) = 0$ とする. $R_{\delta'} \subseteq R_\delta$ は自明である. $predict(B, s) = 0$ なので, 手続き $predict$ において条件に合わなかった遷移が少なくとも 1 つは存在する. よって $R_{\delta'} \neq R_\delta$ である. ■

定理 12 最初にうけとった正例を s_0 とすると, $LEARN\_P_1$ の誤予測の回数 MB は高々 $O(|s_0|^4)$ である.

証明: 誤予測がおきたとき, 補題 11 より, 1 つ以上の遷移が更新によって B から除去されることがわかる. したがって MB は $\mathcal{B}(s_0)$ 中のすべての $1pa$ の遷移の総和でおさえられている. $1pa$ $B(s_0; I, J, K)$ の遷移の数は高々 $O(I \cdot J)$ であるから, その総和は, $\sum_{(I,J,K)\in F_{|s_0|}} O(I \cdot J) = O(|s_0|^4)$ である. ■

定理 9 , 補題 10 , 補題 12 より, 1 変数パタンのクラスは多項式時間オンライン学習可能である.

5 WINNOW を利用した学習アルゴリズム

本節では, 単調選言形のクラスを効率よく学習するアルゴリズム WINNOW[5] を用いて, 1 変数パタンのクラスをより効率よく学習することを考える.

5.1 WINNOW アルゴリズム

N 次元論理関数のオンライン学習を考える. WINNOW は単調選言形のクラス C_{MD} を効率よくオンライン学習するアルゴリズムであり, しきい値 θ, 更新要素 $\alpha > 1$ および N 個の重み $w_1, \ldots, w_N$ を保持している. 一般に $\alpha = 1.5$, $\theta = N$ に設定する. 重みの初期値はそれぞれ 1 である. 例 $u_1 \cdots u_N \in \{0, 1\}^N$ をうけとると, WINNOW は $\sum_{i=1}^{N} w_i \cdot u_i \geq \theta$ のとき予測を $d = 1$ とし, そうでないとき $d = 0$ とする. 誤予測がおきたとき, $u_i = 1$ となっている変数の重み w_i を, $d = 1$ のときは α 倍し, $d = 0$ のときは $1/\alpha$ 倍する.

定理 13 (Littlestone 1988[5]) 目標関数を $f(x_1, \ldots, x_N) = x_{i_1} \vee \cdots \vee x_{i_M}$ とすると $WINNOW$ の誤予測数は, $O(M \log N)$ である.

5.2 WINNOW を利用した1変数パタン学習アルゴリズム

整数の組 (I,J) に対して, $U(I,J)=\{0,\ldots,I\}\times\{0,\ldots,J\}$ と定義すると, 学習アルゴリズムはつぎのようになる. まず, 学習アルゴリズムは, 最初の誤予測がおこるまで $d=0$ と予測し続ける. 最初の誤予測がおきたとき, このときうけとった例 $s_0\in\Sigma^+$ の長さを $n_0=|s_0|$ とする. 任意の $(I,J,K)\in F_{n_0}$ に対して論理変数 $u_{i,j,0}^{(I,J,K)}$: $((i,j)\in U(I-1,J))$ および $u_{i,j,1}^{(I,J,K)}$: $((i,j)\in U(I,J-1))$ を設定する. これらの論理変数の集合を V_{s_0} で表す. 同時にそれぞれの論理変数に対応する重み $w_{(i,j),0}^{(I,J,K)}, w_{(i,j),1}^{(I,J,K)}$ を設定し, 値を1に初期化する. 論理変数の数は $N=|V_{s_0}|=O({n_0}^4)$ である. 次回の試行からは, その試行でうけとった例を $s\in\Sigma^+$ とすると, 以下のような作業をおこなう.

- 例の変換: 任意の $(I,J,K)\in F_{n_0}$ に対してつぎをおこなう. $(I,J,K)\notin F_{|s|}$ のとき, (I,J,K) をラベルとして持つすべての論理変数 $u_{(i,j),0}^{(I,J,K)}, u_{i,j,1}^{(I,J,K)}$ に1を割り当てる. $(I,J,K)\in F_{|s|}$ のとき, $t_0=\psi(s_0;I,J,K)$, $t=\psi(s;I,J,K)$ として, つぎのように値を割り当てる.

$$u_{i,j,0}^{(I,J,K)}:=\begin{cases}0 & s[h_{|t|}(i,j)+1]=s_0[h_{|t_0|}(i,j)+1]\text{ のとき}\\ 1 & \text{上記でないとき,}\end{cases}$$

$$u_{i,j,1}^{(I,J,K)}:=\begin{cases}0 & sub(s,h_{|t|}(i,j)+1,|t|)=t\text{ のとき}\\ 1 & \text{上記でないとき.}\end{cases}$$

- 予測: 変換された例に対して,

$$\sum_{(I,J,K)\in F_{n_0}}\left(\sum_{(i,j)\in U(I-1,J)}w_{i,j,0}^{(I,J,K)}u_{i,j,0}^{(I,J,K)}+\sum_{(i,j)\in U(I,J-1)}w_{i,j,1}^{(I,J,K)}u_{i,j,1}^{(I,J,K)}\right)\geq N$$

のとき $d=1$ とし, そうでないとき $d=0$ とする. 予測としては, $\neg d$ を出力する.

- 更新: 予測の成否 $r\in\{0,1\}$ をうけとり, $r=0$ のとき, すなわち, 誤予測がおきたときは仮説の更新をおこなう. 仮説の更新は1が代入されている論理変数に対応する重みを, $d=1$ ならば α 倍し, $d=0$ ならば $1/\alpha$ 倍する.

定理 14 最初にうけとった正例の長さ n_0 に対して, 学習アルゴリズムの1回の試行の計算時間は, その試行でうけとった例の長さを n とすると $O(n\cdot{n_0}^2\log n_0+{n_0}^4)$ である.

証明: まず, 例の変換に必要な計算時間を考える. 1つの $(I,J,K)\in F_{n_0}$ に着目する. 任意の $(i,j)\in U(I-1,j)$ に対して論理変数 $u_{i,j,0}^{(I,J,K)}$ への値の割り当ては定数時間で計算できる. 先に文字列 s 上の $t=\psi_{i,j,1}^{(I,J,K)}$ の出現位置を計算しておけば, 任意の $(i,j)\in U(I,J-1)$ に対して $u_{i,j,1}^{(I,J,K)}$ への値の割り当ても定数時間で計算できる. t の出現位置は $O(n)$ で計算可能なので, 1つの (I,J,K) に対する例の変換は $O(n+|U(I-1,J)|+|U(I,J-1)|)=O(n+I\cdot J)$ で計算できる. よって, 全体の例の変換の計算時間は,

$$\sum_{(I,J,K)\in F_{n_0}}O(n+I\cdot J)\ =\ |F_{s_0}|\cdot O(n)+\sum_{(I,J,K)\in F_{n_0}}O(I\cdot J)=O(n\cdot{n_0}^2\log n_0+{n_0}^4).$$

つぎに予測と更新の計算時間を考える. 論理変数の数, および重みの数は N なので, 予測と更新はそれぞれ $O(N)=O({n_0}^4)$ 時間で計算できる. よって, 1試行の計算時間は $O(n\cdot{n_0}^2\log n_0+{n_0}^4)$. ■

1変数パタン $p\in P_1$ に対して V_{s_0} 上の単調選言形を $H_p=\bigvee_{d=1}^{m}u_{I(d),J(d),b(d)}^{(I,J,K)}$ と定義する. ここに $m=|p|$, $(I,J,K)=\tau(p)$, $I(d)=I(pref(p,d-1))$, $J(d)=J(pref(p,d-1))$ である. また, $b(d)$ は, $p[d]=x$ のとき1, $p[d]\in\Sigma$ のとき0の値をとる.

$L(p)$ の補集合を $\overline{L}(p)$ で表す. すなわち $\overline{L}(p)=\Sigma^+-L(p)$ である.

定理 15 $s\in\overline{L}(p)\Leftrightarrow H_p=1$.

証明: $s \in L(p) \Leftrightarrow H_p = 0$ を証明する. 最初におきた誤予測のときにうけとった例 s_0 は正例なので, $s_0 \preceq p$ である. $t_0 = \psi(s_0; \tau(p))$ に対して $p\{x := t_0\} = s_0$ であるから, 補題 1 より,

$$\forall d \in \{1, \ldots, |p|\} : p[d] \in \Sigma \Rightarrow s[h_0 + 1] = p[d] \qquad (*)$$

が成り立つ. ここに $h_0 = h_{|t_0|}(I(q), J(q)), q = pref(p, d-1)$ である.
($\Rightarrow$) $s \in L(p)$ とする. $t = \psi(s; \tau(p))$ に対して $p\{x := t\} = s$ であるから, 補題 1 より,

$$\forall d \in \{1, \ldots, |p|\} : p[d] \in \Sigma \Rightarrow s[h + 1] = p[d]$$

が成り立つ. ここに $h = h_{|t_0|}(I(q), J(q)), q = pref(p, d-1)$ である. 補題 1 と (*) より,

$$\forall d \in \{1, \ldots, |p|\} : u^{\tau(p)}_{I(d), J(d), b(d)} = 0$$

となるから $H_p = 0$ である.
($\Leftarrow$) $H_p = 0$ とすると, $t = \psi(s, \tau(p))$ なので $|t| = (|s| - I(p))/J(p)$ である. よって $h_{|t|}(I(p), J(p)) = I(p) + J(p) \cdot |t| = |s|$ である. さらに $H_p = 0$ と (*) より, 補題 1 の (2) が成り立つ. よって $s \in L(p)$ である. ■

定理 16 *1* 変数パタンのクラスは *WINNOW* を利用してオンライン学習可能である. このときの誤予測数は, 目標パタン $p_* \in P_1$ の長さを m, 最初に誤予測がおきたときの例の長さを n_0 とすると, $O(m \log n_0)$ である.

証明: WINNOW は単調選言形 H_p を正しくオンライン学習する. 定理 15 より, 1 変数パタン言語 $L(p_*)$ の補集合 $\overline{L}(p_*)$ のクラスは, WINNOW を利用してオンライン学習可能であり, 予測を反転することにより, 1 変数パタンのクラスも同様にオンライン学習可能である. このときの誤予測数は定理 13 より, $O(m \log N) = O(m \log({n_0}^4)) = O(m \log n_0)$ である. ■

6 おわりに

本稿では, 1 変数パタンのオンライン学習について考察し, この族が, 多項式時間オンライン学習可能であることを示した. また, WINNOW を応用して, 未知パタンのサイズ m と誤予測を起こした最長の例のサイズ n に対して, 誤予測数 $O(m \log n)$ でうごく効率の良い学習アルゴリズムを与えた.

参考文献

[1] Angluin, D.: Finding pattern common to a set of string, *Jurnal of Computer and System Science*, **21**, 46–62, (1980).

[2] Angluin, D.: Queries and concept learning, *Machine Learning*, **2**, 319–342, (1988).

[3] Erlebach, T., Rossmanith,P., Shtadtherr,H., Strger,A., Zeugmann,T.: Learning one-variable pattern langueges Very efficient on average,in parallel,and by asking queries, In *Proc. the 8th International Workshop on Alogorithmic Learning Theory*, 260–276, (1997).

[4] Kearns, M., Pitt,L.: A polynomial-time algorithm for learning k-variable pattern langueges from examples, In *Proc. the 2nd Annual Workshop on Computational Learning Theory*, 57–71, (1989).

[5] Littlestone, N.: Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm, *Machine Learning*, **2**, 285–318, (1988).

[6] Marron, A.: Learning pattern langueges from a single initial exsample and from queries, In *Proc. the 1988 Workshop on Computational Learning Theory*, 345–358, (1988).

[7] Valiant, L.: A theory of learnable, In *Communication of the ACM*, **27**, 1134-1142, (1984).