

線形分離オートマトンを用いたパターン認識手法の 理論的基礎について

沼井裕二[†] 星野憲司^{††} 宇田川祥彰[‡] 小林聡[†]

[†] 電気通信大学大学院電気通信学研究科情報工学専攻 ^{††} 電気通信大学電気通信学部情報工学科
[‡] エヌ・ティ・ティ アイティ株式会社

本論文では、実数ベクトル系列を受理するモデルである線形分離オートマトン (LSA) の理論的結果を示すと同時に、LSA を利用したパターン認識アルゴリズムを提案し、実験・評価する。論文の前半では、実数ベクトル系列を受理するシステムである LSA に関していくつかの定理を導出し、理論的基礎を考察する。LSA は、有限オートマトンの自然な拡張になっている。後半では、LSA に確率的遷移を付与したモデル (PLSA) によってパターン認識を行うアルゴリズムを提案し、実験・評価する。

On Theoretical Foundation of Pattern Recognition with Linear Separation Automata

Yuji NUMAI[†] Kenji HOSHINO^{††} Yoshiaki UDAGAWA[‡] Satoshi KOBAYASHI[†]

[†] Department of Computer Science,
Graduate School of Electro-Communications, The University of Electro-Communications
^{††} Department of Computer Science,
Faculty of Electro-Communications, The University of Electro-Communications
[‡] NTT-IT Corporation

In this paper, we propose automata that accept series data of real-valued vectors, called Linear Separation Automata (LSA); and deduce some theoretical results. Moreover, we also propose the pattern recognition algorithm of series data of real-valued vectors with LSA; and evaluate the algorithm. In the first half of the paper, an acceptor that accepts series data of real-valued vectors, LSA, is introduced and its theoretical properties are studied. It is naturally extended from finite automata. In the latter, the pattern recognition algorithm using LSA with probabilistic transitions is proposed; and an experimental result of this algorithm is shown.

1 イントロダクション

時系列データから何らかのパターンを認識、あるいは予測するという試みは、これまで数多く行われている (例えば、[1, 2])。特に、実際のデータにおいては、動画、工場の計測データ、遺伝子発現など、実数ベクトルを特徴量として利用することが多く、実数値データのパターン認識問題は重要な研究課題である。

本論文では、認識モデルとしてオートマトンを扱う。計算モデルとしては基本的かつ単純であるが、実数値の系列を扱えるように拡張したモデルも数多く提案されている (例えば、[3, 4])。

本論文の前半では、各状態で実数ベクトルを線形に分離することで実数ベクトルの系列を受理する線形分離オートマトン (Linear Separation Automata: LSA) に関していくつかの定理を導出し、理論的基礎を考察する。後半では、LSA に確率的遷移を付与し

た確率的線形分離オートマトン (Probabilistic Linear Separation Automata: PLSA) による実数ベクトル系列の認識アルゴリズムを提案し、実験・評価する。既存研究と比較すると、確率モデルを利用した精度の高い認識システムのための基礎を構築しようとしているのが、本論文の特徴である。

本論文の前半は、LSA の基礎理論の考察に当てられる。LSA は、各状態に d 次元実数ベクトルの重みと実数値の閾値の列を持つようなオートマトンである。ある状態に d 次元実数ベクトルが入力されると、入力と重みの内積が計算され、その値と閾値列とを比較して遷移先が決定される。よって、各状態では、入力ベクトルを線形に分離していることになる。LSA は、全体として、実数ベクトルの系列を受理することが可能である。

LSA について定義を終えたら、いくつかの理論的結果を示す。その過程で、LSA が有限オートマトン

の自然な拡張であり、各種の定理も有限オートマトンの場合と同様の方法で導出できることが明らかにされる。

LSA を拡張し、確率的に遷移先が決まる状態を持つことができるようにしたものを、確率的線形分離オートマトン (PLSA) という。確率を付与した理由は、音声認識 [5] やバイオインフォマティクス (遺伝子の発見やタンパク質配列モチーフのモデル化など) [6] での成功にみられるように、ノイズのある実データの認識に確率モデルを利用することで、認識の精度を上げることが可能となる、というものである。

本論文の後半では、実数ベクトル系列を PLSA で認識するアルゴリズムを提案し、実験・評価する。与えられた PLSA (これはある性質 P を表現していると思わせる) に、当該課題向けに調整した Viterbi アルゴリズム [7] を適用すれば、入力された実数ベクトルの時系列が、性質 P を持つかどうかを判別することができると思われる。このアルゴリズムの実験を行い、結果を評価する。

以降では、まず LSA, PLSA を定義し、理論的結果を示す (2 章)。そして、実数ベクトル系列を PLSA で認識するアルゴリズムを提案し、実験・評価する (3 章)。最後に結論と今後の課題を述べる (4 章)。

2 線形分離オートマトンの理論的基礎

2.1 線形分離オートマトン

線形分離オートマトン (LSA) M は、次のように定義される。

$$M = (d, n, Q, p, F, W, H, \delta)$$

d は M に入力される実数ベクトルの次元数、 Q は状態の有限の順序集合、 $p \in Q$ は初期状態、 $F \subseteq Q$ は受理状態の集合、 W は Q の各状態の重みベクトルの順序集合、 H は Q の各状態の閾値ベクトルの順序集合、 $\delta: Q \times \mathbf{R}^d \rightarrow Q$ は状態遷移関数である。 n は Q, W, H の間の拘束条件で、 $n = |Q| = |W| = |H|$ でなければならない。

W, H, δ について詳しく説明する。まず、 $Q = \{q_1, q_2, \dots, q_n\}$ とおく。このとき、 $W = \{w_1, w_2, \dots, w_n\}$ 、 $H = \{h_1, h_2, \dots, h_n\}$ である。ただし、 $w_i \in \mathbf{R}^d, h_i = (h_{i,1}, h_{i,2}, \dots, h_{i,j})$ ($h_{i,k} \in \mathbf{R}, j \in \mathbf{N}, r < s \Rightarrow h_{i,r} < h_{i,s}$) が成り立つ。そして、 δ は次のような形で表される関数である。ただし、 $q_i, p_i \in Q, x \in \mathbf{R}^d$ とする。

$$\delta(q_i, x) = \begin{cases} p_1 & \text{if } x \cdot w_i < h_{i,1} \\ p_2 & \text{if } h_{i,1} \leq x \cdot w_i < h_{i,2} \\ \vdots & \\ p_j & \text{if } h_{i,j-1} \leq x \cdot w_i < h_{i,j} \\ p_{j+1} & \text{if } h_{i,j} \leq x \cdot w_i \end{cases}$$

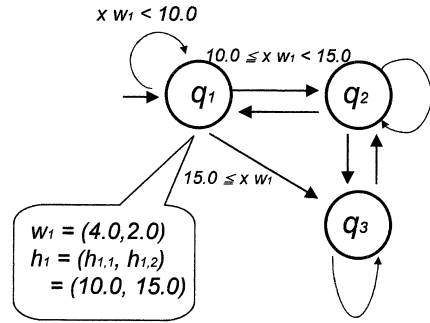


Figure 1: LSA の例

LSA の例を Figure 1 に示す。 q_1 に実数ベクトル $x_1 = (1.0, 2.0)$ が入力されたとき、 $x_1 \cdot w_1 = 8.0 < 10.0 = h_{1,1}$ より、 q_1 に遷移する。

LSA は、各状態において、入力された d 次元実数ベクトルを線形に分離する。

2.2 確率的線形分離オートマトン

$M = (d, n, Q, p, F, W, H, \delta)$ を LSA とする。ここで、 Q とは別の状態集合 P 、および、確率的状態遷移関数 $\Delta: P \times (P \cup Q) \rightarrow [0, 1]$ (ただし $\sum_{q' \in P \cup Q} \Delta(q, q') = 1$) を M に追加しよう。すると、確率的線形分離オートマトン (PLSA) M' が、以下のように定義できる。

$$M' = (d, n, Q, p, P, F, W, H, \delta, \Delta)$$

d は M に入力される実数ベクトルの次元数、 Q は入力されたベクトルに依存して遷移先が決まる状態 (決定的状態) の有限の順序集合、 $p \in P \cup Q$ は初期状態、 P は Δ に依存して確率的遷移を行う状態 (確率的状態) の集合、 $F \subseteq Q$ は受理状態の集合、 W は Q の各状態の重みベクトルの順序集合、 H は Q の各状態の閾値ベクトルの順序集合、 δ は決定的状態遷移関数、 Δ は確率的状態遷移関数である。 W, H, δ, Δ の定義は、 $\delta(q_i, x) \in P \cup Q$ となることを除けば、前述したものと同一である。 n は Q, W, H の間の拘束条件で、 $n = |Q| = |W| = |H|$ でなければならない。

PLSA の例を Figure 2 に示す。この例では、一番下の dead state を無視すれば、 $N = |P| = |Q|$ が成り立っている。

2.3 理論的結果

ある有限オートマトンが受理する言語 (正則言語) に関して、Myhill-Nerode の定理 [8] により、その言語を受理する状態数最小の有限オートマトンが一意に定まる。一方、LSA が受理する言語に関しても、上述の定理を LSA 向けに調節された形の定理を証明することが可能である。つまり、以下の定理が成り立つ。

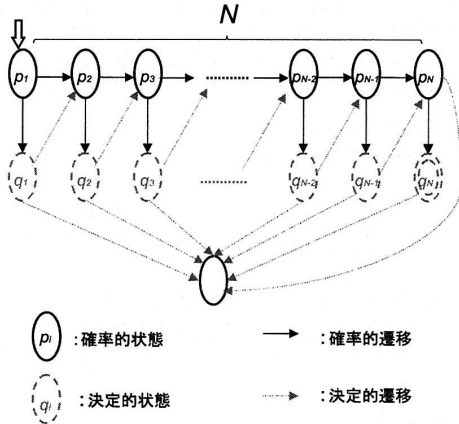


Figure 2: PLSA の例

Theorem 1. ある LSA が与えられたとき、 $(\mathbf{R}^d)^*$ 上で定義した Myhill-Nerode 関係に関して同値であるような状態の対を統合すると、同じ受理能力を持つ状態数最小の LSA が得られる。なお、ある言語を受理する状態数最小の LSA は、状態の名前の違いを無視すれば、一通りに定まる。

□

これに基づき、LSA の状態数最小化アルゴリズムを得ることができる。このアルゴリズムは、有限オートマトンの場合と同様のものである。つまり、考えられるすべての状態対に関して、それらが Myhill-Nerode 関係に関して同値であるかを調べ、同値であるような状態対は統合する。これにより、状態数最小の LSA が求められる。

3 確率的線形分離オートマトンによるパターン認識アルゴリズム

本章では、確率的線形分離オートマトンが表す性質を持つ実数ベクトル系列を判別するためのアルゴリズムを提案し、実験・評価する。

3.1 アルゴリズム

PLSA に実数ベクトル列が入力されたとき、受理状態に到達する最も確からしい確率を求めることを考える。

PLSA $M = (d, n, Q, p, P, F, W, H, \delta, \Delta)$ に d 次元実数ベクトル列 x_1, x_2, \dots, x_m を入力したとする。 i 番目まで入力されたときに、状態 q_k に到達する最も確からしい確率を $v_k(i)$ とする (ただし $q_k \in Q = \{q_1, \dots, q_n\}$)。任意の $q, q' \in Q \cup P$ について、 q から q' へのパスのうち、全て確率的遷移を通過して (確率パス)、かつその各遷移の持つ確率値の積が最大

となるもののその値を $P(q, q')$ と書く (確率パスがない場合は $P(q, q') = 0$ 。また、 $P(q, q) = 1$)。

M が入力を全て読み終えたときに状態 q_1, \dots, q_n にいる最も確からしい確率 $v_1(m), \dots, v_n(m)$ は Algorithm 1 で求められる。これは、Viterbi アルゴリズム [7] を PLSA 向けに調整したものといえる。

なお、任意の $q, q' \in Q \cup P$ に対する $P(q, q')$ は、重み付き有向グラフの全点対間最短経路問題を解く Floyd-Warshall アルゴリズム [9] を PLSA 向けに調整したものによって、計算することができる。

Algorithm 1 PLSA-Viterbi

Input: d 次元実数ベクトル列 x_1, \dots, x_m および
 PLSA $M = (d, n, \{q_1, \dots, q_n\}, p, P, F, W, H, \delta, \Delta)$
 Output: x_1, \dots, x_m を M に入力して受理状態に到達する最も確からしい確率

```

for  $k = 1$  to  $n$  do  $v_k(0) = P(p, q_k)$ ; end
for  $i = 1$  to  $m$  do
  for  $k = 1$  to  $n$  do
     $v_k(i) = \max\{v_j(i-1) \cdot P(\delta(q_j, x_i), q_k) \mid j = 1, \dots, n\}$ ;
  end
end
Output  $\max\{v_f(m) \mid q_f \in F\}$ ;

```

3.2 実験手法

以下の方法により、ある性質を持った PLSA が与えられたときの、PLSA-Viterbi によるパターン検出率を求める。

- Figure 2 と同じ形の、長さ N の PLSA を与える。ただし、各決定的状態 $q_i \in Q$ に対して、 $|w_i| = 1, h_i = (h_{i,1} = 0)$ となるようにパラメータを正規化しておく。
- PLSA に受理されるような d 次元実数ベクトル系列 N_L 本からなる集合 L_P を作成する。また、別にランダムな d 次元実数ベクトル系列 N_L 本からなる集合 L_R を作成する。ただし、含まれている系列の長さの度数分布は両方とも同じであり、かつ、各ベクトルの長さを 1 に正規化しておく。 $L_P \cup L_R$ を訓練集合とする (L_P が正例集合、 L_R が負例集合)。
- $L_P \cup L_R$ の各系列 α を PLSA-Viterbi に入力し、各々について、受理状態 q_N に到達するパスの確率の最大値 $PV(\alpha)$ を計算する。
- PLSA による分類のための閾値 h を、訓練集合の正例および負例の分類の成功率の幾何平均を最大化するように定める。つまり、正例について $S_P(h) = \frac{|\{\alpha \in L_P \mid h \leq \log PV(\alpha)\}|}{|L_P|}$ 、負例について $S_R(h) = \frac{|\{\alpha \in L_R \mid \log PV(\alpha) < h\}|}{|L_R|}$ としたとき、 $h = \operatorname{argmax}_h \sqrt{S_P(h) \cdot S_R(h)}$ である。

Table 1: PLSA-Viterbi の実験結果

$\Delta(p_i, q_i)$ の値の範囲	閾値 h	検出率 (%)	
		$P(T_P)$	$P(T_R)$
$0.80 \leq \Delta < 1.00$	-6.7	90.2	0.16
$0.60 \leq \Delta < 0.80$	-9.0	68.1	3.37
$0.40 \leq \Delta < 0.60$	-9.7	72.9	22.0
$0.20 \leq \Delta < 0.40$	-9.2	63.0	55.0
$0.00 \leq \Delta < 0.20$	-6.9	53.4	53.4

2. と同様の手法で、テスト集合 $T_P \cup T_R$ を新たに作成する。ただし $|T_P| = |T_R| = N_T$ とする。
- $T_P \cup T_R$ の各系列 α に対して $PV(\alpha)$ を計算し、 h と比較して入力进行分类し、検出率 $P(T_P), P(T_R)$ を計算する。ただし、 $P(T) = \frac{|\{\alpha \in T | h \leq PV(\alpha)\}|}{|T|}$ である。

3.3 実験結果

$N = 15, N_L = N_T = 100000, d = 4$ として実験を行った。ただし、確率的状態遷移関数 $\Delta(p_i, q_i)$ に関してとりうる値の範囲を限定した PLSA を 5 つ生成して、それぞれに対して実験を行った。

Table 1 がその結果である。 $\Delta(p_i, q_i)$ 、つまり Figure 2 における確率的状態 p_i から決定的状態 q_i への下向きの遷移の確率が高いほど、識別の精度が高いことがわかる。特に、 $\Delta \geq 0.60$ なら、 $P(T_R)$ が一桁に抑えられていることが注目に値する。

Figure 2 の形の PLSA のパラメータを、ある性質を持つ実数ベクトル系列の正例と負例から学習する場合、ベクトル系列のノイズがあまり多くなければ、 $\Delta(p_i, q_i)$ の値は高くなると推測される。ゆえに、PLSA-Viterbi はパターン識別アルゴリズムとして有効に機能するといえる。

4 結論と今後の課題

本論文ではまず、線形分離オートマトン (LSA) 及び確率的線形分離オートマトン (PLSA) を定義し、理論的結果を示した。LSA は、各種の定理や最小化アルゴリズムを有限オートマトンの場合と同様の方法で得られる。これは、LSA が有限オートマトンの自然な拡張であることを意味する。また、実数ベクトル系列を PLSA で認識するアルゴリズム PLSA-Viterbi を提案し、実験・評価した。その結果、PLSA-Viterbi はパターン識別アルゴリズムとして有効に機能することを示すことができた。

最後に今後の課題を述べる。

本論文では基本的には、LSA に関する理論的結果を提示した。今後は、パターン認識の直接的な基盤となっている PLSA に関する理論的含意についても明らかにしなければならない。

現実の認識問題に PLSA-Viterbi を適用するには、ある性質をもつ実数ベクトル系列の正例および負例から、その性質をもつような PLSA を学習する必要がある。例えば、次のような学習手法が考えられる。

- ある性質をもつ実数ベクトル系列の正例および負例をアラインメントする。
- アラインメントデータから、当該性質を持つような PLSA のパラメータ及びパターン認識のための閾値を学習する。

PLSA-Viterbi の検出率は、学習によって得られた PLSA のパラメータの精度に大きく依存するだろうから、学習をいかに正確に行うかが重要になるだろう。

References

- T. Matsunaga and M. Oshita, "Recognition of walking motion using support vector machine," *ISICE2007*, pp. 337–342, 2007.
- T. Mohri and H. Tanaka, "Weather prediction by memory-based reasoning," *Journal of Japanese Society for Artificial Intelligence*, vol. 10, no. 5, pp. 798–805, 1995.
- R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, no. 1, pp. 3–34, 1995.
- N. Lynch and F. Vaandrager, "Forward and backward simulations for timing-based systems," *Proceedings of the Real-Time: Theory in Practice*, pp. 397–446, 1992.
- L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.
- R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis: Probabilistic models of proteins and nucleic acid*. Cambridge University Press, 1998.
- A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE transactions on Information Theory*, vol. IT-13, no. 2, pp. 260–269, 1967.
- A. Nerode, "Linear automaton transformations," *Proceedings of the American Mathematical Society*, vol. 9, pp. 541–544, 1958.
- R. W. Floyd, "Algorithm 97 shortest path," *Communications of the ACM*, vol. 5, no. 6, p. 345, 1962.