

## ショートノート

### コンシステントラベリングを用いたパターン連想†

黒川 清†† 石川 聖二†† 加藤 清史††

脳の情報処理を工学的に実現するための研究は、神経回路網理論や知識工学の分野で活発に行われているが、本論文では脳の基本的機能の1つである画像パターン連想の概念的なことを絞って、これをコンシステントラベリングの概念を用いて実現する方法について述べる。コンシステントラベリングとは、ネットワーク拘束解析問題を抽象した概念である。画像上の画素をユニット、画素値をラベルとすれば、画像は、ユニット-ラベル対の集合として表現できる。画像がある物体を表せば、これらのユニット-ラベル対の間に拘束関係が生じる。このことを利用すれば、連想過程は拘束解析問題としてコンシステントラベリングの言葉で記述できる。コンシステントラベリングを用いたパターン連想のモデルを構成し、2値画像について実験を行った。その結果満足いく動作が得られた。

#### 1. はじめに

脳の情報処理を工学的に実現するための研究は、神経回路網理論<sup>1)</sup>や知識工学<sup>2)</sup>として活発に行われているが、本論文では脳の基本的機能である連想の概念的なことを絞って、これをコンシステントラベリングの概念を用いて実現する方法について考察する。コンシステントラベリングとは、ラテンスクエアパズル、部分グラフ同型問題、グラフ準同型問題、コンピュータビジョンにおけるシーンのラベル付け問題等のネットワーク拘束解析問題を抽象した概念である<sup>3)</sup>。我々は、ひずみを持つ画像パターンマッチング問題への本概念の適用を検討している<sup>4)</sup>が、同様の考え方により、拘束解析問題として連想をとらえれば、連想機能もコンシステントラベリングの言葉で記述できる。本論文では、コンシステントラベリングによる連想方式の提案とその有効性について、実験例を交えて検討する。

#### 2. コンシステントラベリングによる連想方式

コンシステントラベリング問題<sup>3)</sup>は、ユニット-ラベル拘束関係からユニットへの矛盾のないラベル付けを求める問題である。ユニット  $u$  の集合を  $U$ ,  $u \in U$  に付加可能なラベル  $l$  の集合を  $L$ , ユニット間の拘束関係を与える  $u$  の組の集合を  $T$ , ユニット-ラベルのペア間の拘束関係を与える組の集合を  $R$  で表せば、4つ

組  $(U, L, T, R)$  はコンパチビリティモデルと呼ばれ、この解、すなわち  $U$  の要素に対する、 $R$  が与える拘束関係を満足する1対1のラベル付けは、コンシステントラベリングと呼ばれる。これは縦型探索によって求めることができる。

今、デジタル画像を  $f(i, j)$  ( $i=1, 2, \dots, I; j=1, 2, \dots, J$ ) で表せば、画像とは、画素  $(i, j)$  とグレイレベル  $f(i, j)$  のペアの集合とみなすことができる。そこで画素をユニット  $u (\in U)$ , グレイレベルをラベル  $l (\in L)$  とする。連想問題を考えるために基本となる画像パターンを取り、これを  $f_k(i, j) (k=1, 2, \dots, N)$  で表せば、ユニット拘束関係およびユニット-ラベル拘束関係は、これらの個々の画像パターンに対して定義される。画像パターン  $f_k(i, j)$  に対するユニット拘束集合を  $T_k = \{(u_{r1}^k, u_{r2}^k, \dots, u_{rn}^k)\}$ , ユニット-ラベル拘束集合を  $R_k = \{(u_{r1}^k, l_{r1}^k, u_{r2}^k, l_{r2}^k, \dots, u_{rn}^k, l_{rn}^k)\}$  で表せば、 $N$  個の画像パターンがメモリに記憶された状態は、

$$\begin{aligned} T &= \bigcup_{k=1}^N T_k \\ &= \{(u_{r1}, u_{r2}, \dots, u_{rn})\}, \\ R &= \bigcup_{k=1}^N R_k \\ &= \{(u_{r1}, l_{r1}, u_{r2}, l_{r2}, \dots, u_{rn}, l_{rn}) \mid \\ &\quad (u_{r1}, u_{r2}, l_{r2}, \dots, u_{rn}) \in T\} \end{aligned}$$

で定義される  $T, R$  を用いて表現できる。想起のために入力されるパターン (ある画像パターンの部分) は、

$$\begin{aligned} X &= \{(u_{x1}, l_{x1}, u_{x2}, l_{x2}, \dots, u_{xn}, l_{xn}) \mid \\ &\quad (u_{x1}, u_{x2}, \dots, u_{xn}) \in T, \end{aligned}$$

† Associating Patterns by Consistent Labeling by KIYOSHI KUROKAWA, SEIJI ISHIKAWA and KIYOSHI KATO (Department of Electric, Electronic and Computer Engineering, Faculty of Engineering, Kyushu Institute of Technology).

†† 九州工業大学工学部計算機工学教室

$$\text{not}((l_{x1}=\text{nil}) \cap (l_{x2}=\text{nil}) \cap \dots \cap (l_{xn}=\text{nil}))$$

のように、ユニット-ラベル拘束集合を用いて表現される。ここで、nil はラベル不明を表す。X が与えられると、R の要素は X によって拘束を受ける。X に矛盾する要素を R から除いた集合を  $R^*$  とする。すなわち、

$$R^* = \{r \mid r \in R, r \sim \forall x' \in X\}.$$

ここで、“ $\sim$ ” は無矛盾を表す。

無矛盾であることをチェックし、 $R^*$  を生成するアルゴリズム REMOVE は次のように表現できる。ただし、 $r \equiv (u_{r1}, l_{r1}, u_{r2}, l_{r2}, \dots, u_{rn}, l_{rn}) \in R$ ,  $x \equiv (u_{x1}, l_{x1}, u_{x2}, l_{x2}, \dots, u_{xn}, l_{xn}) \in X$  とする。また一般に  $s = (u_{s1}, l_{s1}, u_{s2}, l_{s2}, \dots, u_{sn}, l_{sn})$  のとき、 $s|_{\tau} = (u_{s1}, u_{s2}, \dots, u_{sn})$  と定義する。

#### REMOVE

```
begin
  copy R to R*;
  for all r ∈ R* do
    for all x ∈ X do
      if r|τ = x|τ
        then if for a certain i
              (lri ≠ lxi) and (lxi ≠ nil)
                then remove r from R*
    end.
end.
```

REMOVE が生成する集合  $R^*$  を使えば、本連想問題は、コンパチビリティモデル  $(U, L, T, R^*)$  で表されるコンシステントラベリング問題に帰着する。全体の流れは次のように記述できる。

#### ASSOCIATION 1

```
begin
  input pattern X;
  perform REMOVE with X and R to
    obtain R*;
  perform consistent labeling with (U, L, T, R*)
end.
```

本アルゴリズムにおいて、REMOVE 処理によって得られる  $R^*$  を  $R^*_1$  で表す。

先述のように、コンシステントラベリング問題の解は縦型探索によって求められるが（そのアルゴリズムは一般的記述<sup>5)</sup>があるので省略する）、これは時間のかかる処理である。そこで、ユニット  $u$  がラベル  $l$  を持つ確からしさが大きければ、 $(u, l)$  を確定ペアとして入力 X に含める、という発見的手法を導入し、探索の効率化を計る。この手法としてアルゴリズム MAJOR-

ITY を導入する。

MAJORITY は、ラベルが nil のユニット  $u_{xi}(l_{xi} = \text{nil})$  を含む要素  $x$ （ただし確定したラベルを持つユニットが多いもの）を X から取り出し、R の中ですべてのユニットが  $x$  のユニットと一致する要素  $r$  と、ラベルどうしの比較を行う。一致するラベルの個数がある値  $\theta$  より大きければ、 $u_{xi}$  に対応する  $r$  のユニット  $u_{ri}$  が持つラベル  $l_{ri}$  を  $u_{xi}$  のラベルの候補とし、集合  $L_{xi}$  に入れる。候補ラベルは一通りとは限らないから、 $u_{xi}$  について上述の手続きを繰り返した後、 $L_{xi}$  内で頻度の最も高いラベルを  $u_{xi}$  のラベルとする。以下にアルゴリズムを示す。

ただし、

$$f(l_{ri}, l_{xi}) = \begin{cases} 1 & \dots & l_{ri} = l_{xi} \\ 0 & \dots & \text{otherwise} \end{cases}$$

$\theta, \lambda$ : しきい値

$L_{xi}$ : ユニット  $u_{xi}$  の候補ラベルの集合

$\text{maj}(L_{xi})$ :  $L_{xi}$  中で頻度の最も高いラベルを決定する関数

である。また、集合  $S$  の要素数を  $n(S)$  で表すものとして、 $s = (u_{s1}, l_{s1}, u_{s2}, l_{s2}, \dots, u_{sn}, l_{sn})$  のとき、

$$g(s) = n(\{u_{si} \mid l_{si} \neq \text{nil}\})$$

と定義する。（ $X, R$  の定義より、 $x \in X, r \in R$  についてそれぞれ  $1 \leq g(x) \leq n, g(r) = n$  が成り立つ。）

#### MAJORITY

```
begin
  copy X to X';
  while X' is renewed by the assignment do
    begin
      let Lxi := {φ};
      for all x ∈ X' such that lxi = nil and
        g(x) > λ do
        for all r ∈ R such that r|τ = x|τ do
          if  $\sum_{j=1}^n f(l_{rj}, l_{xj}) > \theta$  then put lri into
            Lxi;
        if Lxi ≠ {φ} then
          for all x ∈ X' such that lxi = nil do
            assign maj(Lxi) to lxi
          end
        end.
    end.
```

MAJORITY 処理を含む連想の流れは、次のように表現できる。

#### ASSOCIATION 2

```
begin
```

input pattern  $X$ ;  
 perform MAJORITY with  $X$  to obtain  $X'$ ;  
 perform REMOVE with  $X'$  and  $R$  to  
 obtain  $R^*$ ;  
 perform consistent labeling with  $(U, L, T, R^*)$   
 end.

画像  $P$  上で値が未知の画素数を  $m(P)$  で表せば、 $m(P) \geq 0$  である。MAJORITY を実行すれば  $m(X') \leq m(X)$  なるパターン  $X'$  が得られる。 $m(X')=0$  のこともあるが、一般には  $m(X') \neq 0$  である。このパターンを使って  $R$  に対し REMOVE 処理を施せば、新たに集合  $R^* \equiv R^*_2$  が得られる。 $R^*_1$  がパターン  $X$  を使って  $R$  から求められるのに対し、 $R^*_2$  は、 $X$  を包含するパターン  $X'$  を利用して、 $R$  から求められる。すなわち  $R^*_2$  のほうがより強い拘束条件から得られる集合である。よって、一般に  $n(R^*_1) \geq n(R^*_2)$  が成り立つから、MAJORITY 処理の導入によって探索効率を上げることができる。なお、MAJORITY 処理の副次的効果として、MAJORITY 処理のみで連想が完結する場合がある。また MAJORITY 処理の一部を変更すれば、画像パターンに多少のノイズが重畳しても処理可能になる (実験結果参照)。

### 3. 実験

5×5 画素の 2 値画像パターンを対象として連想モデルを作成し、ASSOCIATION 1 および ASSOCIATION 2 により、連想実験を行った。ただし本論文では後者の実験結果を示す。実験に使用したパターンは図 1 に示す “○”, “×”, “◇” の 3 種類のパターンである。

ユニットを  $u_i (i \in U) (i=1, 2, \dots, n_1 (=25))$ , ラベルは 2 値パターンを想定しているので、 $L = \{-1, 0, 1\}$  とする。画像上では ●=1, ○=-1 とし、nil は ·=0 で表すものとする。また  $L^- = \{-1, 1\}$  と定義する。ユニット拘束集合  $T$  はすべての画素のペアを要素とし、各画素の値が互いにどのような関係にあるかをユニット-ラベル拘束集合  $R$  で表現する。

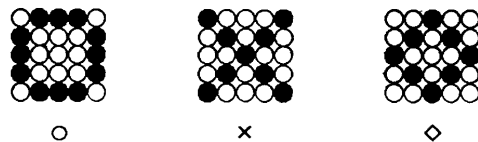


図 1 記憶パターン  
 Fig. 1 Storage patterns.

記憶は、記憶パターンが提示されると、それが画素のペア  $(u_i, u_j) \in T$  に与えるラベルの組  $(l_i, l_j) (l_i, l_j \in L^-)$  を、 $(u_i, l_i, u_j, l_j)$  の形で  $R$  に登録することによって行われる。本実験では、図 1 のパターンのユニット拘束関係を与える集合をそれぞれ  $T_1, T_2, T_3$  とすれば、 $T_1 = T_2 = T_3 \equiv T, n(T) \equiv n_T = {}_{25}C_2 = 300$ , また  $n(R) \equiv n_R = 702$  となる。想起は、記憶により生成された  $R$  と入力パターン  $X$  をもとに、ASSOCIATION 2 によって行う。ただし MAJORITY で参照されるしきい値については、ユニット拘束集合の要素が画素のペアであるから、 $\theta = \lambda = 0.5$  とする。

実験結果の一部を図 2 に示す。図 2 (a) は、図 1 に示されたパターン “○” の部分パターンが入力された場合であるが、MAJORITY 処理のみで正しく想起が行われている。図 2 (b) は、図 1 中央のパターンの部分パターン上で、第 2 行第 2 列の画素がノイズのために ●→○となったものであるが、この場合も MAJORITY 処理のみで正しく想起が行われている。ノイズが修正されたのは、この実験に限り、MAJORITY 処

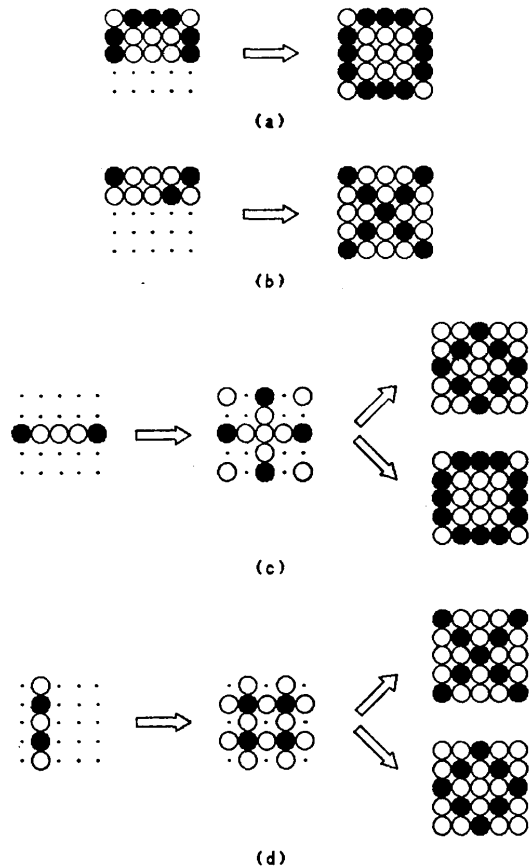


図 2 想起過程の例  
 Fig. 2 Examples of the associative process.

理の中で  $l_{xi} = \text{nil}$  のユニットばかりでなく、 $l_{xi} \neq \text{nil}$  のユニットも処理の対象としたためである。図 2(c), 2(d)のように2つのパターンに共通な部分が入力された場合は、MAJORITY 処理による想起パターン(中央)を経て、REMOVE 処理で得られた  $R^*$  に対するコンシステントラベリングの解として2種類のパターンが想起されている。

#### 4. 考 察

探索効率の尺度として、ASSOCIATION 1 を適用した場合、ASSOCIATION 2 を適用した場合とで、ユニット-ラベル拘束集合(それぞれ  $R^*_1, R^*_2$ )の要素数と  $R$  の要素数を比較した。その結果、 $r_i = n(R^*_i) / n(R)$  とすれば、図 2 の 3 例(a, c, d)については、それぞれ a:  $r_1 = 0.61, r_2 = 0.43$ , c:  $r_1 = 0.90, r_2 = 0.74$ , d:  $r_1 = 0.87, r_2 = 0.68$  であり、その平均は  $\bar{r}_1 = 0.79, \bar{r}_2 = 0.62$  となり、MAJORITY の導入が探索効率を上げるのに効果のあることが確かめられた。なお図 2(b)は、REMOVE のみでは正しい想起は行われないため比較対象から除いている。

MAJORITY 処理は、図 2(b)でみられるように、ノイズ修正という副次的効果があるが、正しい画素値をノイズとして修正してしまう可能性もある。今後、このような誤動作をなくするための検討が必要である。

従来のアナログ連想方式<sup>6),7)</sup>とは異なり、コンシステントラベリングによる連想方式はデジタルである。パターンの記銘は、ユニット拘束集合  $T$  に対する、そのパターンが与えるユニット-ラベル拘束集合  $R$  の生成によって行われる。そのため  $T$  をうまく取れば、記銘パターンの数が増えても、分離性の高い連想を行うことが可能であると思われるが、 $T$  (あるいは  $R$ ) の要素数が増大すると探索時間が増大する。本連想システムの実現性の立場から、 $T$  の与え方が検討されなければならない。

$R$  に対するコンシステントラベリングは縦型探索によって求められるが、これは逐次的方法であり、画像の画素数が増えると処理時間が膨大になる。探索回数を減らす方法は検討されている<sup>3)</sup>が、逐次的な探索は実際の画像パターンへの適用には限界があるように思われる。今後、本連想方式を実行するための並列的なアルゴリズムを検討する必要がある。

#### 5. おわりに

ネットワーク拘束解析問題の一般化である、コンシステントラベリングを用いた連想方式を提案し、その有効性を実験的に確かめた。本方式によれば、記銘パターン数が増えても画素間拘束の与え方によっては、分離性のよい連想が実現できるのではないかと考えている。本方式の並列化とともに今後の検討課題である。

#### 参 考 文 献

- 1) 合原一幸: ニューラルコンピュータ, 東京電機大学出版局 (1988).
- 2) 大須賀節雄: 知識情報処理, オーム社 (1986).
- 3) Haralick, R. M. and Shapiro, L. G.: The Consistent Labeling Problem: Part 1, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-1, No. 2, pp. 173-184 (1979).
- 4) Ishikawa, S., Tamura, H. and Kato, K.: On Matching Distorted Patterns by Consistent Labelling, *IEICE Technical Report, PRU 88-32*, pp. 15-22 (1988).
- 5) Nilsson, N. J.: *Principles of Artificial Intelligence*, Springer-Verlag (1980).
- 6) 中野 繁: アソシアトロン, 昭晃堂 (1979).
- 7) Kohonen, T.: Correlation Matrix Memories, *IEEE Trans. Comput.*, Vol. C-21, No. 4, pp. 353-359 (1972).

(平成元年 6 月 22 日受付)

(平成 2 年 1 月 16 日採録)



黒川 清 (学生会員)

昭和 40 年生。昭和 63 年九州工業大学工学部情報工学科卒業。現在同大学院博士前期課程電気工学専攻在学中。コンピュータビジョンの研究に従事。



石川 聖二 (正会員)

1974 年東京大学工学部計数工学科卒業。1979 年同大学院工学系研究科修了。工学博士。同年九州工業大学講師。1983 年～1984 年シェフィールド大学客員研究員。現在九州工業大学工学部助教授。主に画像理解の研究に従事。計測自動制御学会、電子情報通信学会各会員。

**加藤 清史 (正会員)**

昭和8年生. 昭和34年東京大学工学部応用物理学科卒業. 昭和36年東京大学数物系修士課程修了. 昭和39年東京大学数物系博士課程中退. 昭和39年九州工業大学助教授.

現在九州工業大学工学部教授・電気工学科・計算機工学教室勤務. 工学博士. 画像情報処理の応用, 教育へのコンピュータ利用, 計算機による音楽情報処理などの研究に従事. 著書「デジタルシステムの基礎」(昭和56年オーム社). 計測自動制御学会, 電子情報通信学会, 日本ロボット学会, 日本特殊教育学会などの会員.

---

**論 文 誌 編 集 委 員 会**

委員長	村井 真一			
副委員長	益田 隆司			
委員	浮田 輝彦	大田 友一	小池 誠彦	
	小谷 善行	佐藤 興二	島津 明	
	戸川 隼人	永田 守男	原田 紀夫	
	疋田 輝雄	松田 晃一	三浦 孝夫	
	毛利 友治	吉澤 康文	米崎 直樹	