

## [特別講演] DP マッチング概説 ～ 基本と様々な拡張 ～

内田 誠一†

† 九州大学大学院システム情報科学研究所  
〒 819-0395 福岡市西区元岡 744  
E-mail: uchida@is.kyushu-u.ac.jp

あらまし 弾性マッチングの一種である DP マッチングは、開発されて既に 30 年以上経った現在でも、パターン認識・画像処理の基本技術として多用されている。本稿では、1 次元パターン間の DP マッチング、1 次元と 2 次元パターンの DP マッチング、2 次元パターン間の DP マッチングを述べた後、いくつかの高速化手法、ならびに学習理論との組み合わせについて紹介する。

キーワード 動的計画法, DP マッチング

### DP Matching – Fundamentals and Applications –

Seiichi UCHIDA†

† Faculty of Information Science and Electrical Engineering, Kyushu University  
744 Motoooka, Nishi-ku, Fukuoka-shi, 819-0395 Japan  
E-mail: uchida@is.kyushu-u.ac.jp

**Abstract** Dynamic programming (DP) matching has been developed in early 1970's and vastly employed in various pattern recognition and image processing problems as an efficient algorithm to provide optimal elastic matching (nonlinear correspondence) between two patterns. This paper describes DP matching for 1D-1D, 1D-2D, and 2D-2D pattern matching problems with techniques for reducing computational complexity. Several combinations with learning algorithms are also described.

**Key words** dynamic programming, DP matching

#### 1. ま え が き

2 パターンをマッチングする際、一方に微小な歪み(ずれ)があると整合性が急に悪くなる。このため、マッチングの際、一方のパターンをゴムのように非線形伸縮させ、最も整合した時点をマッチング結果とすることがある。これは弾性マッチングと呼ばれる技術であり、音声やジェスチャなどの時系列パターン(1次元パターン)認識や、顔・文字・指紋などの画像パターン(2次元パターン)認識、画像圧縮のための動き補償など、さまざまな分野で必要不可欠とされる。

この弾性マッチングには動的計画法(dynamic programming, DP)による手法 - DP マッチング が広く用いられている。時間伸縮を伴う音声時系列パターンのマッチングを目的として 1970 年代に迫江らにより開発された [1] ~ [3]。(開発競争の様子については文献 [4] に詳述されている。) 理論的明快さ、アルゴリズム実装の容易さ、そして非常に少ない計算量ながら最適マッチングが求まる(長さ  $I$  と  $J$  の 1 次元パターンであれば、 $O(IJ)$  の計算量で済む) という、非常に優れた性質を持つ。こ

のため 1 次元パターン認識、特に時系列パターン認識(例えばオンライン文字認識やジェスチャ認識)において爆発的に広がり、当該分野における基本的道具となっている [5]。

本稿では、1 次元パターン間の DP マッチングのアルゴリズムについて、その導出および実用における幾つかのノウハウについて述べる。またその発展形である 1 次元と 2 次元パターンの DP マッチング、および 2 次元パターン間の DP マッチングについて概説する。さらにいくつかの高速化手法、ならびに学習理論との組み合わせについて紹介する。

#### 2. 1 次元パターン間の DP マッチング

##### 2.1 基本アルゴリズム

2 つの 1 次元パターン  $X = x_1, x_2, \dots, x_i, \dots, x_I$ ,  $Y = y_1, y_2, \dots, y_j, \dots, y_J$  間の弾性マッチングを考える。ここで  $x_i$  と  $y_j$  は特徴ベクトルである。弾性マッチングとは、 $X$  の第  $i$  要素  $x_i$  と  $Y$  の第  $j$  要素  $y_j$  との対応付け  $j = u_i$  ( $i = 1, \dots, I$ ) を最適化する問題である。

この要素間の対応付けのコスト(局所距離)を

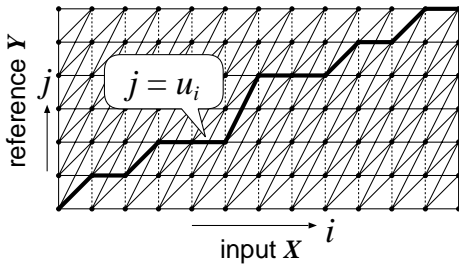


図1 1次元パターン間のDPマッチング

$$d_i(u_i) = \|x_i - y_{u_i}\| \quad (1)$$

とすると、最適化問題は次のように定式化される。

$$\left. \begin{array}{l} \text{minimize} \quad F = \sum_{i=1}^I d_i(u_i), \\ \text{w.r.t.} \quad u_1, \dots, u_I, \\ \text{subject to} \quad 0 \leq u_i - u_{i-1} \leq 2, u_1 = 1, u_I = J \end{array} \right\} (2)$$

第一の制約条件は単調連続性制約と呼ばれ、(i)  $u_i$  は  $u_{i-1}$  より小さくなることなく(単調性)、(ii)  $u_i$  は  $u_{i-1}$  より大きくなり過ぎることはない(連続性)、という条件である。連続性条件は対応付け  $j = u_i$  を関数(歪み関数と呼ばれる)としてみたときの傾き  $du_i/di$  を2以下にするための条件であり、2倍以上の速度変化を許さない条件となっている。第二、第三は始端・終端条件である。

最適化問題(2)は次のように表現できる。

$$\begin{aligned} \min F &= \min_{\substack{u_1, \dots, u_I \\ 0 \leq u_i - u_{i-1} \leq 2}} \sum_{i=1}^I d_i(u_i) \\ &= \min_{\substack{u_2, \dots, u_I \\ 0 \leq u_i - u_{i-1} \leq 2}} \left[ \sum_{i=3}^I d_i(u_i) \right. \\ &\quad \left. + \left\{ d_2(u_2) + \min_{\substack{u_1 \\ 0 \leq u_2 - u_1 \leq 2}} d_1(u_1) \right\} \right] \end{aligned}$$

ここで、右辺  $\{ \}$  内を

$$g_2(u_2) = d_2(u_2) + \min_{\substack{u_1 \\ 0 \leq u_2 - u_1 \leq 2}} d_1(u_1) \quad (3)$$

と置くと、

$$\min F = \min_{\substack{u_2, \dots, u_I \\ 0 \leq u_i - u_{i-1} \leq 2}} \left[ \sum_{i=3}^I d_i(u_i) + g_2(u_2) \right]$$

となり、見かけ上、制御変数  $u_1$  を消去できる。(見かけ上というのは、式(3)で  $\min$  を与える  $u_1$  がまだ確定できていないためである。) 上式をさらに変形して、

$$\begin{aligned} \min F &= \min_{\substack{u_3, \dots, u_I \\ 0 \leq u_i - u_{i-1} \leq 2}} \left[ \sum_{i=4}^I d_i(u_i) \right. \\ &\quad \left. + \left\{ d_3(u_3) + \min_{\substack{u_2 \\ 0 \leq u_3 - u_2 \leq 2}} g_2(u_2) \right\} \right] \end{aligned}$$

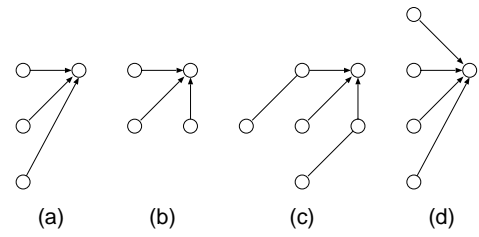


図2 傾斜制限

として、右辺  $\{ \}$  内を

$$g_3(u_3) = d_3(u_3) + \min_{\substack{u_2 \\ 0 \leq u_3 - u_2 \leq 2}} g_2(u_2)$$

と置くと、

$$\min F = \min_{\substack{u_3, \dots, u_I \\ 0 \leq u_i - u_{i-1} \leq 2}} \left[ \sum_{i=4}^I d_i(u_i) + g_3(u_3) \right]$$

となり、今度は制御変数  $u_2$  を消去できる。このように

$$g_i(u_i) = d_i(u_i) + \min_{\substack{u_{i-1} \\ 0 \leq u_i - u_{i-1} \leq 2}} g_{i-1}(u_{i-1}) \quad (4)$$

を繰り返し計算することで次々に変数を消去でき、結局

$$\min F = \min_{u_I} g_I(u_I)$$

として目的関数  $F$  の最小値が得られる。境界条件  $u_I = J$  があるため、結局  $\min F = g_I(J)$  となる。

式(4)はDP漸化式と呼ばれ、これがDPマッチングの基本式となっている。このDP漸化式を、 $i=1$  から  $I$  まで順にすべての  $j$  において計算すること最適解が得られる。このことは、図1のような  $I \times J$  個の格子点を持つトレリス(DP平面と呼ばれる)上のすべての点において、式(4)中の最小値選択に従った局所的な経路選択を順次行っていることに相当する。この処理は探索幅  $J$ 、深さ  $I$  の幅優先探索であり、したがって計算量は  $O(IJ)$  となる<sup>(注1)</sup>。

DPマッチングの結果として得られるのは、パターン間距離(DP距離)  $\min F$  と対応関係  $u_1, \dots, u_i, \dots, u_I$  の二つである。DP距離は非線形伸縮に対して不変なマッチング距離であり、従って変形を伴うパターンの認識に有効である。長さが違う2パターン(すなわち  $I \neq J$ ) においても計算可能な距離である点も重要である。また対応関係はパターン  $X$  を基準とした  $Y$  の変形状況を表現したものであり、構造解析的手法や統計的解析手法との接点となりうる。

(注1): 余談になるが、上の計算に必要なメモリ量について、学生に答えさせてみると面白い。多くの学生諸君は、 $I \times J$  の2次元配列を確保すると答える。時々少し気の利いた学生がいて、 $g_i(u_i)$  の計算時には  $g_{i-1}(i-1)$  のみが必要であることに気づき、 $2 \times J$  の2次元配列を確保すると答える。しかし実は  $2+J$  で済む。(後述する整合窓を考えれば、もっと減らせる。) 詳細は文献[13]を参照されたい。メモリが存分に使える世の中ではあるが、こうした工夫こそが大事であると筆者は信じている。なお、バックトラックにより最適な  $u_i$  も求める必要がある場合は、この限りではない。

## 2.2 傾斜制限

DP 漸化式 (4) 中の最小値選択範囲は傾斜制限と呼ばれ、歪み関数  $j = u_i$  の傾斜を制限する。この傾斜制限には図 2 に示すように様々な種類がある。式 (2) の単調連続性制約に基づく DP マッチングでは、同図 (a) の傾斜制限を利用している。

傾斜制限の使い分けは重要である。図 2(a) は非対称型と呼ばれる形式であり、1 状態のスキップを許した left-to-right 型 HMM と等価なマッチングを実現する。最適経路上の局所距離  $d$  の累積加算回数は  $I$  回となり、 $J$  に依らない。このため入力パターン  $X$  を認識する際、標準パターン  $Y$  の長さ  $J$  に応じたマッチング距離  $\min F$  の正規化が不要である。同図 (b) は対称型と呼ばれ、Levenshtein 距離を求める際に利用される。非対称型と異なり 2 パターン  $X, Y$  を交換しても同じ距離 ( $\min F$ ) が求まる。局所距離加算回数が最小で  $\max(I, J)$ 、最大で  $I + J$  と変動するので、 $\min F$  による最短距離識別を行う際には注意が必要である。(c) も対称型と呼ばれるが (b) とは大きく異なり、最適経路の傾斜を  $1/2 \sim 2$  に制約する。時系列パターンであれば、 $1/2 \sim 2$  倍の速度変動に限り補償することに相当する。同図 (d) は単調性制約のない傾斜制限である。西村ら [6] は、時間的に逆行するような対応付けを許すべく、ジェスチャ認識にこのような傾斜制限を利用している。

傾斜制限は  $i$  もしくは  $(i, j)$  毎に変えてもよい。これは局所的に制約条件を変えることに相当する。その結果、HMM の状態遷移確率制御に似た効果が得られる。局所的に連続性を無くし、対応関係に大きなギャップを許すこともできる。

## 2.3 整合窓

傾斜制限により対応付けの範囲は制約されるが、それでも意図しない極端な対応付けが起こりうる。こうした極端な対応付けを起こさないための一つの単純な方式として整合窓がある。これは一般に、線形対応付け  $j = u_i = iJ/I$  からあまり離れすぎないようにする条件  $|u_i - iJ/I| \leq W$  として定義される。ここで  $W$  は窓幅と呼ばれる正定数である。DP 漸化式も整合窓内で計算するだけでよいので、計算量低減効果も見込める ( $O(IJ)$  を  $O(IW)$  にできる)。傾斜制限と同様、 $W$  も  $i$  もしくは  $(i, j)$  毎に変えてもよい。

## 2.4 局所距離

式 (1) では局所距離  $d_i(u_i)$  を  $\|x_i - y_{u_i}\|$  と定義したが、他にも様々なバリエーションが考えられる。絶対値距離や if-then ルールのような微分不可能な局所距離を利用できるのも DP の特徴の一つである。例えば閾値を用いた局所距離

$$d_i(u_i) = \begin{cases} 0 & \text{if } \|x_i - y_{u_i}\| \leq \epsilon \\ \infty & \text{otherwise} \end{cases}$$

も使用できる。文献 [7] では、上式の局所距離を利用した類似区間検出法が提案されている。

## 2.5 始端終端フリー化

図 1 のトレリスの上下に距離 0 のエッジを加えて図 3 のように改造すれば、始端終端条件のない最適マッチングアルゴリズムが実現する。これはしばしば始端・終端フリー DP マッチングと呼ばれ、入力パターン  $X$  中に  $Y$  と似た区間を探す、い

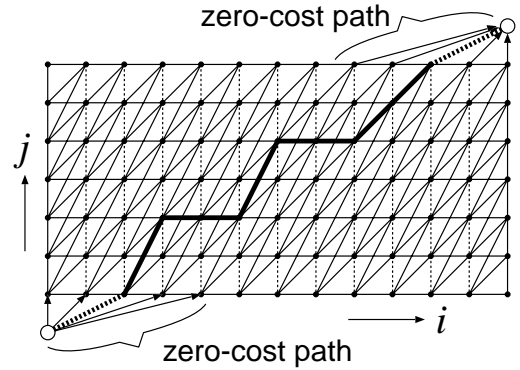


図 3 始端・終端フリーな DP マッチング

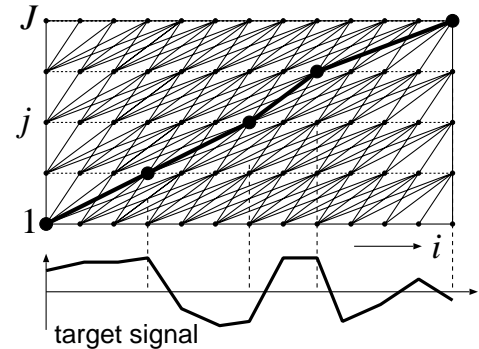


図 4 DP によるセグメンテーション

わゆるスポッティング処理に応用できる。岡 [8] はこの方式を連続 DP と呼び、様々な応用を提案している。オンライン文字データなどのパターンの始終端付近の不安定部分 (フック) をスキップするためにも利用できよう。

## 2.6 セグメンテーションへの利用

マッチングではないが、DP の考え方自体はセグメンテーションにも応用可能である。パターン  $X$  を最適セグメンテーションする場合の DP 平面を図 4 に示す。ここでは縦軸がセグメント数になる。大津の多値化 [9] では、ヒストグラムを最適階段近似するために DP セグメンテーションを用いている。他にも音声スペクトラムのセグメンテーションによるフォルマント抽出 [10] や曲線の多角形近似 [11] にも応用されている。なお、図 4 に従って  $J$  セグメントへの最適分割を求めると、既に  $j (< J)$  セグメントへの最適分割についても求まっている点に注意されたい。

## 2.7 Recognition-Based Segmentation

前節の DP セグメンテーションと DP マッチングを組み合わせると、セグメンテーション・非線形伸縮の補償・認識の 3 処理を最適化の枠組みで一括して行うアルゴリズムが得られる。これは Recognition-based segmentation と呼ばれ、早くから文字列認識 [12] や連続音声認識 [13] に利用されてきた。

Recognition-based segmentation は、始端・終端フリーの DP 平面を標準パターン毎に準備し、図 5 のようにそれらを結合することで構築できる。すべての DP 平面における処理がフレーム  $i$  に同期して進行できることも、時系列パターンマッチングを行う上では都合がよい。

Recognition-based segmentation には、しばしば FSA 制御

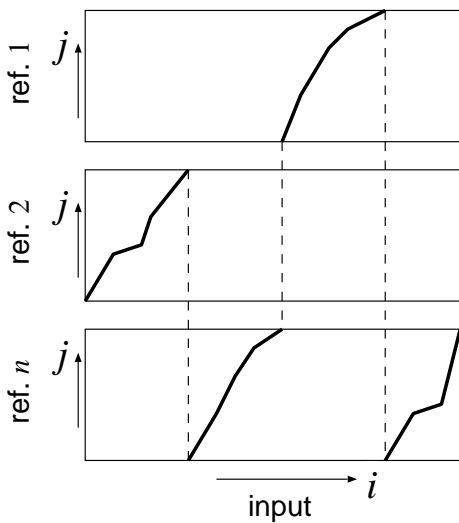


図 5 Recognition-based segmentation

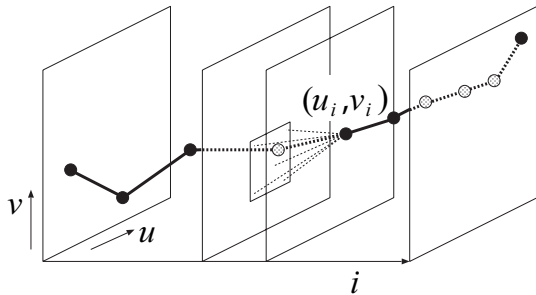


図 6 1次元-2次元パターン間のマッチング

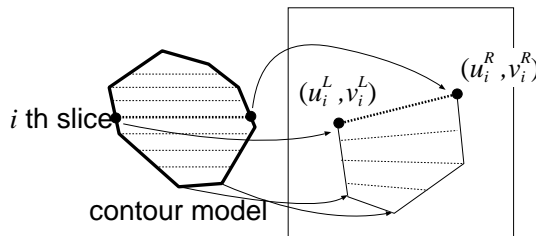


図 7 閉輪郭のフィッティング問題

が導入される．この場合，各 DP 平面を FSA における一つの状態のように考え，状態間の遷移すなわち DP 平面間の結合の有無を FSA により既定する．この FSA 制御は，文法モデルが必要となる文字列認識や音声認識において多用される．階層化することも可能であり，例えば音声認識では，音素レベル処理を最下層，文レベル処理を最上層とする FSA 制御に基づいた認識手法がすでに一般化している [14]．FSA だけではなく，文脈自由文法 (CFG) も扱うこともできる．このことは CFG パーザの最も基本的なものである CYK パーザが 2 段 DP マッチング [3] と等価な処理であることから理解できる．

### 3. 1次元-2次元パターン間の DP マッチング

2. では，1次元パターン間のマッチングのために，対応付け関数  $j = u_i$  を考えた．この  $u_i$  を 2次元ベクトル  $(u_i, v_i)$  とすれば，1次元パターンと 2次元パターンのマッチングが可能となる．こうした応用についても古くから検討されており，RS マッチング [15] や MAP マッチング [16] はその嚆矢であろう．

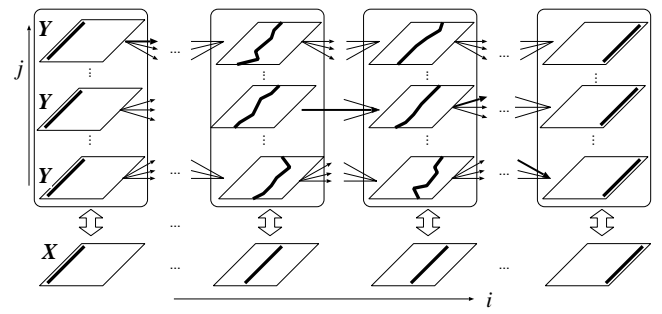


図 8 2次元パターン間の DP マッチング

またマッチングに限らず，2次元パターン中からの 1次元的な構造 (例えばエッジ [17]) の検出にも利用される．

図 6 は 1次元-2次元パターンマッチングの様子である．累積距離を格納する  $g_i$  は  $g_i(u_i)$  から  $g_i(u_i, v_i)$  に拡張されるが，それ以外は 1次元パターン間の場合とさほど変わらない．計算量，すなわち DP 漸化式計算回数は  $O(IJ^2)$  である．

1次元-2次元 DP マッチングにより SNAKES すなわち輪郭モデルの 2次元平面へのフィッティングのアルゴリズムも得られる [18]．ただし，閉輪郭を一箇所で切断して開輪郭化し，1次元-2次元 DP マッチング法を利用する方式では不十分である．2 端点  $(u_1, v_1), (u_I, v_I)$  を一致させられないためである．そこで，図 7 のように閉輪郭のスライスを考え，そのスライスの系列により閉輪郭を 1次元化する [19]．各スライス  $(u_i^L, v_i^L)$  と  $(u_i^R, v_i^R)$  の 4 変数で表されるために， $g_i$  は  $g_i(u_i^L, v_i^L, u_i^R, v_i^R)$  となる．このことから予想されるように，計算量は  $O(IJ^4)$  と急に増加する．

### 4. 2次元パターン間の DP マッチング

1次元パターンの DP マッチングが提案された直後から，2次元パターンすなわち画像パターン用の DP マッチング法が数々提案されてきた [20]．しかし，定番と呼ばれるような手法は未だ存在しないのが現状である．これは，1次元 DP マッチングをそのまま 2次元 DP マッチングに拡張した問題が以下に示すように NP-hard 問題となり [21]，事実上解けないためである．

4 隣接画素間の単調連続性制約の下，ラスタスキャンの順  $(1, 1), (1, 2), \dots, (1, N), (2, 1), (2, 2), \dots, (2, N), \dots, (N, N)$  に 1画素ずつ対応先を決定していく過程を考えると，画素  $(i, j)$  のマッチング先の決定に  $N$  段前の  $(i-1, j)$  の決定が影響することになる．従ってこの 2次元 DP マッチングが本質的には  $N$  次マルコフ性を持った最適決定問題であることがわかる．ここで列すなわち  $N$  画素をまとめてブロック化すると， $N$  次マルコフ的な決定過程を単純マルコフ決定過程に変換でき，形式的には DP の枠組みで解くことが可能になる [22], [23]．図 8 は，大きさ  $N \times N$  の 2次元パターン間の DP マッチングの様子である．各列の可能な対応付けをすべて考えると  $N$  に関する指数オーダーとなるため，NP-hard 問題になることがわかる．このため実用的には区分線形化 [24] や後述するビームサーチのような計算量低減の工夫が必要となる．

ところで，ラスタスキャンとジグザグスキャン  $(1, 1), (1, 2),$

..., (1, N), (2, N), (2, N - 1), ..., (2, 1), (3, 1), ..., (N, N) では、それによって構成される DP アルゴリズムの計算量が著しく異なる。後者では最高で  $2N$  次マルコフ性が発生するためである。このように 2 次元 DP マッチングでは、計算量の最小化のために、計算順序の最適化が必要である。このメタ最適化問題は Secondary Optimization Problem [25] と呼ばれており、グラフ表現されたモデルの 2 次元平面へのマッチング問題でも重要視されている [26]。

## 5. DP マッチングの高速化手法

### 5.1 ビームサーチ

DP マッチングには様々な高速化法がある。2.3 の整合窓も一種の高速化手法である。他の代表的な高速化法としては、まずビームサーチが挙げられる。枝刈りとも呼ばれ、各  $i$  において、最適経路になる見込みのない  $g_i(u_i)$  について DP 漸化式を計算せずにスキップする方法である。局所探索の導入に相当し、最適解の保証はなくなるが、実用上十分な解が得られる場合が多い。

ビームサーチの組み込みにはアルゴリズム的な工夫が必要である。見込みの無い  $g_i(u_i)$  のところに  $\infty$  を代入してビームサーチとしている学生のプログラムを見たことがあるが、これでは全く効果はない。ビームサーチは動的リストとビーム駆動と組み合わせることで初めて効果が出てくる。前者は枝刈りの結果生き残った  $g_i(u_i)$  についてのみ、そのインデックス  $u_i$  と共にメモリに格納する方式である。後者はそのリスト中の  $g_i(u_i)$  から DP 漸化式を駆動し、適宜  $g_{i+1}(u_{i+1})$  を計算する方法である [13]。複雑にはなるが、2 次元 DP マッチングにもビームサーチは導入可能である [23]。

枝刈りすなわち  $g_i(u_i)$  に見込みがあるか否かの判定は、一般に  $g_i(u_i)$  に対するしきい値処理に基づく。従って、最適経路とそうでない経路の  $g_i(u_i)$  に大きな差があることが望ましい。特徴量が平坦であれば  $d_i(u_i)$  に差が出にくく、 $g_i(u_i)$  の差も小さくなるため（最適経路が曖昧になる）、ビームサーチの効きが悪くなる。こうした場合には事前知識を導入する正則化的な工夫がなされる。例えば最適経路が平滑であることがわかっているならば、目的関数を

$$F = \sum_{i=1}^I (d_i(u_i) + \lambda \|u_i - u_{i-1}\|)$$

のように変更することでビームサーチの効果が改善される ( $\lambda$  は正定数)。

### 5.2 粗密 DP

粗密 DP (Coarse-to-fine DP) [27] は解の最適性を保ったまま高速化を図る方法として興味深い。図 9 に示すように、この方法では、(i) 各  $i$  段の  $J$  個の状態を 1 つにまとめて複合状態とし、(ii) DP マッチングを行い、(iii) 最適経路が通過した複合状態を分割して (ii) に戻る、という処理を繰り返し行う。ステップ (iii) において最適経路上の各複合状態内に 1 状態しか含まれなくなれば終了である。複合状態に付随する局所距離が、その複合状態内の状態の局所距離  $d_i(u_i)$  の下限値であれば、こ

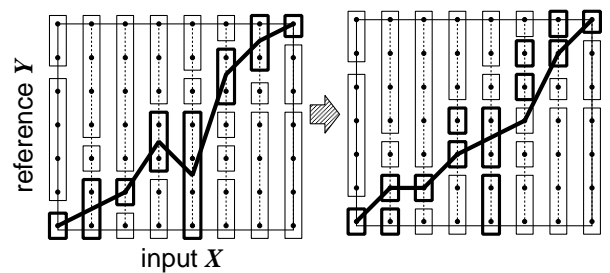


図 9 粗密 DP

の方法で求まる最適経路が原問題のそれに一致することが保証される。計算量低減効果を高めるためには、タイトな下限値を簡単な計算量で求めることが要請される。

### 5.3 解析的 DP マッチング

DP マッチングの高速化の新しい考え方として、解析的 DP マッチング [28] が提案されている。この方法は、局所距離  $d_i(j)$  を各  $i$  において次のように 2 次関数近似する。

$$d_i(j) = \alpha_i j^2 + \beta_i j + \gamma_i$$

この近似の結果、前出の目的関数  $F = \sum_i d_i(u_i)$  は  $u_i$  に関して微分可能になり、解析的な枠組みで解けることになる。単純には  $\partial F / \partial u_i = 0$  をすべての  $u_i$ , ( $i = 1, \dots, I$ ) について連立する解析方法が考えられるが、DP の枠組みでも解くことができる。この場合に重要な点は、 $d_i(j)$  が 2 次関数ならば  $g_i(j)$  も 2 次関数  $g_i(j) = a_i j^2 + b_i j + c_i$  とになる点である。詳細は略するが、これらの係数  $a_i, b_i, c_i$  は  $g_{i-1}(i)$  の係数  $a_{i-1}, b_{i-1}, c_{i-1}$  と  $\alpha_i, \beta_i, \gamma_i$  から一意に計算できるので、この計算を  $i = 1$  から  $I$  まで計算すれば  $g_i(j)$  が求まり、したがって最適解  $\min F = \min_j g_i(j)$  が求まる。

このように、解析的 DP マッチングは、従来の DP マッチングのような探索幅  $J$ 、深さ  $I$  の幅優先探索に基づく組合せ最適化的な手法ではない。このことは  $J$  が非常に大きくなっていく問題、例えば前出の 2 次元パターンマッチング問題についても DP が適用可能であることを示唆している。また、この 2 次元パターンマッチング問題を考えると、上述の連立解法においては必要となる大規模な逆行列計算を回避できる可能性がある。

## 6. 学習理論とのリンク

### 6.1 主成分分析とのリンク

2.1 で述べたように、DP マッチングにより求めた対応関係  $u_1, \dots, u_i, \dots, u_I$  は  $X$  を基準とした  $Y$  の変形を表している。従って、あるクラスについて一つの標準パターン  $X$  と多くの学習パターン  $\{Y\}$  を準備し、各  $Y$  を  $X$  と DP マッチングすれば、そのクラスに起こる変形を自動的に収集できることになる。

こうして得られた変形の共分散行列を求めると、変形に関するマハラノビス距離 [29], [30] や 2 次識別関数 [31] が利用できる。さらにそのクラスに起こりやすい変形が張る部分空間 [32] や tangent plane [33] を形成できる。DP マッチングをはじめとする弾性マッチングでは、一般に合わせ過ぎが問題になる。合

わせ過ぎはそのクラス内にほとんど起こらない変形を許したことに相当するので、マハラノビス距離や部分空間を用いてマッチングを評価することで、合わせ過ぎを検出することが可能となる。

以上の処理は、長さ（次元）の異なるパターンについても適用できる。これはDP マッチングの次元正規化能力に依る。従来、固定次元のパターンにのみ統計的手法が適用されてきたが、このようにDP マッチングを介することで可変次元のパターンを扱えるようになる。

## 6.2 識別関数理論とのリンク

DP による最適経路をパターン  $X$  から  $Y$  への非線形写像として考えると、その経路を様々に変えることで  $X$  からその写像パターンとしての  $Y$  を様々に合成できることになる。パターン空間で考えると、 $X$  を中心としてその歪みパターンを含む集合が DP により構成できることになる。そこで、単純に学習パターンについて識別面を求めるのではなく、各学習パターンについて DP によりその歪みパターン集合を求め、それらについて識別面を求めることが考えられる。こうした考え方は学習パターンが少ない場合に特に有効であろう。文献 [34] では、この考えに基づいた線形識別関数の誤り訂正学習法が提案されている。また文献 [35] では誤り訂正学習の代わりに、線形計画法を用いた方法が提案されている。

## 6.3 Kernel Machines とのリンク

局所距離  $d_i(j)$  を  $x_i y_j$  として、DP 漸化式中の  $\min$  を  $\max$  にすれば、DP マッチングにより 2 ベクトル  $X, Y$  間の内積（すなわち類似度）を求めることができる。このことから DP をカーネルと見ることができ、種々の Kernel Machines において利用できる。SVM の枠組みにおいては既に実際に利用されている [36], [37]。これらは、前節で紹介した手法 [34], [35] と同様の効果を狙ったものと考えられよう。

## 7. ま と め

DP マッチングの基本と関連する話題について述べた。DP マッチングは多くの教科書にも載っており、すでに使い古された技術のように思われるかも知れない。しかし、扱っている問題がもし最適マッチング問題として定式化されたならば、その解法を模索する際、単純ながらも最適解を与える DP の適用可能性について一考の価値があろう。HMM や MRF [38] がそうであるように、表面的には複雑な定式化が成されていることも案外多いようである。

## 謝 辞

DP マッチング研究に取り組むきっかけを与えて下さり、また 18 年という長きにわたりご指導下さった、九州大学システム情報科学研究所 迫江博昭教授に心より感謝申し上げます。

## 文 献

- [1] 迫江博昭, 千葉成美, “動的計画法を利用した音声の時間正規化に基づく連続音声認識,” 日本音響学会誌, Vol. 27, no. 9, pp. 483-490, 1971.
- [2] H. Sakoe and S. Chiba, “A Dynamic Programming Algorithm Optimization for Spoken Word Recognition,” IEEE Trans. on Acoust. Speech and Signal Proc., Vol. ASSP-26, No.1, pp. 43-49, 1978.

- [3] H. Sakoe, “Two-Level DP-Matching Algorithm—A Dynamic Programming Based Pattern Matching Algorithm for Continuous Speech Recognition,” IEEE Trans. on Acoust. Speech and Signal Proc., Vol. ASSP-27, No.6, pp.588-595, 1979.
- [4] 迫江博昭, “DP マッチングとの 30 年,” 信学技報, SP2002-155, Dec. 2002.
- [5] 大田友一, 山田博三, “動的計画法によるパターンマッチング,” 情報処理, vol. 30, no. 9, pp. 1058-1066, 1989.
- [6] 西村拓一, 野崎俊輔, 向井理朗, 岡 隆一, “連続 DP への非単調性導入によるジェスチャ動画からの戸惑い動作のスポッティング認識,” 信学論 (D-II), vol. J81-D-II, no. 1, pp. 18-26, Jan. 1998.
- [7] 森 明恵, 内田誠一, 倉爪 亮, 谷口 倫一郎, 長谷川 勉, 迫江博昭, “論理判定型 DP マッチングによる類似区間検出とその基本動作抽出への応用,” 画像の認識理解シンポジウム (MIRU2006), IS2-2, July 2006.
- [8] 岡 隆一, “連続 DP を用いた連続単語認識,” 音響学会音声研資料, SP78-20, 1978.
- [9] 大津展之, “判別および最小二乗規準に基づく自動しきい値選定法,” 信学論 (D), vol. J63-D, no. 4, pp. 349-356, April 1980.
- [10] L. Welling and H. Ney, “Formant estimation for speech recognition,” IEEE Trans. Speech Audio Proc., vol. 6, no. 1, pp. 36-48, Jan. 1998.
- [11] J.-C. Perez and E. Vidal, “Optimum polygonal approximation of digitized curves,” Pattern Recog. Lett., vol. 15, pp. 743-750, 1994.
- [12] R. G. Casey and E. Lecolinet, “A survey of methods and strategies in character segmentation,” IEEE Trans. Pat. Anal. Mach. Intell., vol. 18, no. 7, pp. 690-706, 1996.
- [13] 迫江博昭, 藤井宏美, 吉田和久, 巨理誠夫, “フレーム同期化, ビームサーチ, ベクトル量子化の統合による DP マッチングの高速化,” 信学論 (D-II), vol. J81-D-II, no. 6, pp.1251-1258, June 1988.
- [14] H. Ney and S. Ortman, “Progress in dynamic programming search for LVCSR,” Proc. IEEE, vol. 88, no. 8, pp. 1224-1240, Aug. 2000.
- [15] 迫江博昭, “Rubber String Matching 法による手書き文字認識,” 信学技報, PRL74-20, Sep. 1974.
- [16] 山田博三, 山本和彦, “DP 整合法による超音波心臓動画像の認識,” 信学論 (D), vol. 71-D, no. 4, pp. 678-684, 1988.
- [17] U. Montanari, “On the optimal detection of curves in noisy pictures,” Communications of the ACM, vol. 14, no. 5, pp. 335-345, May 1971.
- [18] D. Geiger, A. Gupta, and J. Vlontzos, “Dynamic programming for detecting, tracking and matching deformable contours,” IEEE Trans. Pat. Anal. Mach. Intell., vol. 17, no. 3, pp. 294-302, 1995.
- [19] 上田修巧, 間瀬健二, 末永康仁, “弾性輪郭モデルとエネルギー最小化原理による輪郭追跡手法,” 信学論 (D-II), vol. J75-D-II, no. 1, pp. 111-120, Jan. 1992.
- [20] S. Uchida and H. Sakoe, “A survey of elastic matching techniques for handwritten character recognition,” IEICE Transactions on Inf. Syst., vol. E88-D, no.8, pp.1781-1790, 2005.
- [21] D. Keysers and W. Unger, “Elastic image matching is NP-complete,” Pattern Recog. Lett., vol. 24, no. 1-3, pp. 445-453, 2003.
- [22] 内田誠一, 迫江博昭, “動的計画法に基づく単調連続 2 次元ワーブ法の検討,” 信学論 (D-II), vol. J81-D-II, no. 6, pp. 1251-1258, June 1998.
- [23] S. Uchida and H. Sakoe, “An efficient two-dimensional warping algorithm,” IEICE Trans. Inf. & Syst., vol. E82-D, no. 3, pp. 693-700, March 1999.
- [24] 内田誠一, 迫江博昭, “区分線形 2 次元ワーブ法の検討,” 信学論 (D-II), vol. J83-D-II, no. 12, pp. 2622-2629, Dec. 2000.
- [25] U. Bertele and F. Brioschi, *Nonserial Dynamic Programming*, Academic Press, 1972.
- [26] Y. Amit and A. Kong, “Graphical templates for model registration,” IEEE Trans. Pat. Anal. Mach. Intell., vol. 18, no. 3, pp. 225-236, 1996.
- [27] C. Raphael, “Coarse-to-fine dynamic programming,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 2, pp. 1379-1390, 2001.
- [28] 内田誠一, 迫江博昭, “1 次元パターンへの解析的 DP マッチング,” 画像の認識理解シンポジウム (MIRU2006), OSA-1, 2006.
- [29] S. Uchida and H. Sakoe, “Eigen-deformations for elastic matching based handwritten character recognition,” Pattern Recognit., vol. 36, no. 9, pp. 2031-2040, 2003.
- [30] 三苫寛人, 内田誠一, 迫江博昭, “弾性マッチングと固有変形を用いたオンライン文字認識,” 信学論, vol. 45, no. 12, pp. 2845-2855, Dec. 2004.
- [31] H. Mitoma, S. Uchida, and H. Sakoe, “Online character recognition based on elastic matching and quadratic discrimination,” Proc. Int. Conf. Doc. Anal. Recog., vol. 1 of 2, pp. 36-40, Sept. 2005.
- [32] 三苫寛人, 内田誠一, 迫江博昭, “部分空間法と DP マッチングに基づくオンライン文字認識手法,” 信学技報, PRMU2005-223, Feb. 2006.
- [33] 内田誠一, 迫江博昭, “カテゴリー固有変形の線形結合モデルに基づく弾性マッチング法,” 信学論 (D-II), vol. J87-D-II, no. 2, pp. 639-648, Feb. 2004.
- [34] 迫江博昭, “発生速度の変動を許容する識別関数の計算法,” 昭 47 信学大全, p.53, 1972.
- [35] 廣瀬智貴, 片山喜規, 迫江博昭, “線形計画法と動的計画法によるワーブに対して安定な識別関数の学習法,” 信学技報, PRMU99-169, Dec. 1999.
- [36] 野間健一, 中井 満, 下平 博, 嵯峨山茂樹, “非線形時間伸縮を用いた Support Vector Machine による時系列パタンの認識,” 信学技報, PRMU2000-128, Dec. 2000.
- [37] C. Bahlmann, B. Haasdonk, and H. Burkhardt, “On-line handwriting recognition with support vector machines - A kernel approach,” Proc. Int. Workshop Frontiers of Handwriting Recog., pp.49-54, 2002.
- [38] H. Derin and H. Elliott, “Modeling and segmentation of noisy and textured images using Gibbs random fields,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 9, no. 16, pp. 39-55, 1987.