

# ジェスチャパターンの教師なし追加学習・認識

## Incremental Unsupervised Clustering of Gesture Patterns

岡田 将吾†  
Shogo Okada†

西田 豊明†  
Toyoaki Nishida†

### 1 はじめに

人間とインタラクション可能な知的システムやロボットを実現するためには、人間の用いる言語情報だけでなく、人間の表出するジェスチャなどの非言語情報を理解するメカニズムが必要である。近年までに、人間とロボットやコンピュータとの柔軟なインタラクション(HCI)の実現を目指し、人間の表出するジェスチャや動作の認識・理解に関する研究が盛んに行われている[1]。

多くの研究で提案されたシステムでは、認識対象のジェスチャの予め決定されており、モーションキャプチャやカメラから測定したジェスチャデータに人手でラベルを付与し、これらを訓練データに用いて識別器を構成し、認識を行う。このシステムを搭載したロボットやインターフェイスを通じてインタラクションを行う場合、予め定義された特定種類のジェスチャを人間側が行うことを強いられる。逆にいえばこれらのシステムでは、定義外のジェスチャを認識することが出来ない。

一方で人間がコミュニケーション中に表出するジェスチャは多種多様である上、個人によってその種類や数は異なるため、事前に定義することは難しい。この背景より本研究では、認識対象のジェスチャの種類や数を定義せず、事前知識のない状態から、ジェスチャから得られる時系列パターンを追加的に学習、認識する手法を提案する。

本研究では、静的パターンの追加クラスタリング手法である Self-Organizing Incremental Neural Network

(SOINN)[2]を基盤として本手法を提案する。SOINNは非定期的な入力を学習可能であり、分布に複雑な形状を持つクラスに対しても、ノイズを除去しつつ適切なクラス数及びデータの分布を近似可能である。

しかしながら SOINN は静的パターン(固定長の多次元ベクトル)の追加クラスタリング手法であり、ジェスチャなどから得られる時系列数の異なるパターン群を直接学習することが出来ない。そこで本研究では以下の二点で SOINN を拡張し、HB-SOINN を提案する。

(1) 時系列パターンを Hidden Markov Model (HMM) [3] で学習し、学習後の HMM のパラメータを SOINN の入力とする。

(2) 未学習パターンと思しき入力パターン  $P_a$  については、 $P_a$  から学習された HMM の各状態の出力分布に従い、パターンをランダムサンプリングにより生成し、生成したパターンを SOINN の学習空間に入力することで、 $P_a$  に関する学習強度を強める。

評価実験では、モーションキャプチャを用いて取得した、数種類のハンドジェスチャから得られるデータを用いて提案手法の追加学習性能に関する評価を行う。なお本研究の最終目的は、[1] 非分節の連続動作からジェスチャ部分を自動的にオンラインで分節化し、[2] 分節化したジェスチャを追加的に学習する手法を実現することである。本論文ではこの内 [2] 分節化されたジェスチャを追加学習する手法の実現方法および評価実験について述べる。

### 2 関連研究

教師付き学習では、訓練データへの教師ラベルの付加の作業に手間がかかることや、動作やジェスチャのカテゴリ数自体が未知であるといった問題に対処するために、時系列パターンの教師なしクラスタリング手法が提案されている。これらの手法は体系的に [4] にまとめられている。[4] に記載される手法はバッチ的に与えられた訓練データをクラスタリングする手法であり、オンラインで入力されるパターンの処理や、追加学習に対応していない。

これに対し、クリッチらは人間の全身動作を永続的に観察することで、追加的に学習・認識・生成可能な見まね学習手法を提案している [5]。この研究では、オンライン学習可能な Factorial-HMM (FHMM) が提案されており、FHMM 同士の距離に基づいた階層的クラスタリングを利用した追加学習手法を提案している。この手法では入力されたパターン群から木構造が形成され、入力パターンを全て木構造の枝として保持する。[5] の手法では、認識時に木構造を利用することでパターン

† 京都大学 情報学研究科 知能情報学専攻

マッチングの回数を軽減している．一方，HB-SOINN では必要なプロトタイプベクトルのみが競合学習の過程を経て自動的に生成され，これらとの最近傍識別によりパターン認識を行うことで，パターンマッチングの回数を軽減している．本論文では提案する HBSOINN のアルゴリズムを 3 章で説明し，4 章で [5] で提案された手法およびパッチクラスタリング手法と HB-SOINN との比較評価実験について説明し，5 章でまとめる．

### 3 提案手法

提案手法は時系列パターンの固定次元への写像手法 (3.1 節) と静的パターンの追加クラスタリング手法 SOINN (3.2 節) の二つの手法で構成される．HB-SOINN の概要を図 1 に示す．

#### 3.1 HMM に基づく時系列パターンの固定次元への写像

まず  $k$  番目に入力されたジェスチャから得られる時系列パターンを以下のように定義する．

$$O_k = \{o_{k,1}, o_{k,2}, \dots, o_{k,t}, \dots, o_{k,T_k}\} \quad (1)$$

ここで  $o_{k,t}$  は  $d$  次元のベクトルであり，時刻  $t$  に得られる特徴ベクトルに相当する．また  $T_k$  は時系列データの時系列数を示す．ここで用いる HMM は単一正規分布の出力分布を保持し  $S$  個の状態を持つ left-to-right モデルとした．また HMM のパラメータ推定法には，一般的に用いられる Baum-Welch アルゴリズムを用い，Baum-Welch アルゴリズムのパラメータ推定精度を向上させるため，パラメータの初期値設定に Segmental-Kmeans 法を用いた．

まずパターン  $O_k$  を 1 つの HMM で学習する．この HMM の番号を  $k$  番と定義する．学習後の  $k$  番の HMM のパラメータ  $\lambda_k$  は  $\lambda_k = \{\pi_k, A_k, B_k\}$  と表すことが出来る．パラメータの中で  $\pi_k$  は初期確率集合， $A_k$  は遷移確率集合， $B_k$  は出力確率集合である．さらに出力確率集合は各状態の単一正規分布から生成されるため，出力確率集合から得られるパラメータ集合は  $(\mu_{k,s}, \Sigma_{k,s}) | 1 \leq s \leq S$  で表される． $\mu_{k,j}$  は平均ベクトル， $\Sigma_{k,j}$  は共分散行列である．これら全てのパラメータ集合を特徴ベクトルとして SOINN へ入力すると，次元が大きくなるため，過学習の問題が生じる [6]．この問題を避けるために本研究では，ジェスチャの軌跡の特徴が現れる平均ベクトルのみを入力ベクトルとした．ここで共分散行列  $\Sigma_{k,j}$  も軌跡を表す特徴量であるが  $\Sigma_{k,j}$  を特徴ベクトルに含むと次元が大きくなるため， $\Sigma_{k,j}$  については特徴ベクトルに含めるのではなく， $\Sigma_{k,j}$  を持つ分布からランダムサンプルを生成するという形で学習空間に入力する．

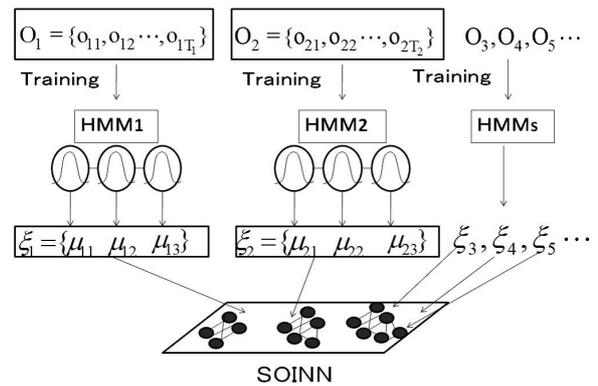


図 1: HBSOINN の概要 (学習後の HMM のパラメータが SOINN への入力となる.)

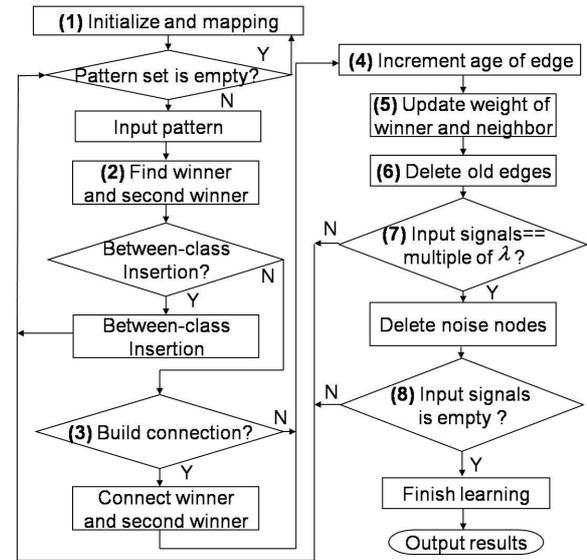


図 2: Adjusted SOINN アルゴリズムの手順.

またジェスチャの時系列長の伸縮が顕著に現れる場合、遷移確率も重要となる特徴量である．しかしながら、今回扱うジェスチャは 1 ストローク程の短い時系列パターンとし、上記のような時系列の伸縮に特徴があるジェスチャを扱わない．この理由のため、遷移確率を特徴ベクトルに含まなかった．この手法については後の Algorithm 1 で述べる．

上記の操作の結果，時系列パターン  $O_k$  は固定状態を持つ HMM に学習された後，各状態の出力分布の平均ベクトル  $\mu_{k,j}$  からなる固定長ベクトル集合に変換される．このベクトルの次元数は  $S \times d$  となる．

#### 3.2 HB-SOINN のアルゴリズム

HB-SOINN の基盤として用いる SOINN アルゴリズムには一層のネットワークによる Adjusted SOINN[7] を用いる (以降では ASOINN と呼ぶ)．ASOINN ではノードと呼ばれるプロトタイプ集合と，ノードをつなぐエッジ集合により，オンラインで追加的に入力され

るパターンの分布を近似する。

図2にHB-SOINNの処理手順を示す。HB-SOINNのアルゴリズムはASOINNアルゴリズムにStep(1)を加え拡張したものである。まず最初にHB-SOINNにとって重要な手順(1)のアルゴリズムをAlgorithm 1で述べる。図3は手順(1)のフローチャートを示している。

以降のAlgorithm 1の各手順(1)から(3)は、図3中の(1)から(3)に対応している。

Algorithm 1: SOINN空間の初期化(図2の手順(1))

- (1) 新たなジェスチャパターン  $O_k$  が  $k$  番目のHMMに入力され、3.1節の手法により固定次元のパターン  $\xi_k$  に変換される。
- (2) SOINN空間にノードが存在しない場合、最初に入力されたパターン  $\xi_1$  と二番目に入力されたパターン  $\xi_2$  を二つのノード ( $A = \{c_1, c_2\}$ ) としてノード集合  $A$  が初期化される。エッジ集合  $C$  ( $C \subset A \times A$ ) は空集合とする。
- (3) SOINN空間に2つ以上のノードが存在する場合、一番距離の近いノード (*winner*)  $s_1$  を探索するもし  $\xi_k$  と  $s_1$  の参照ベクトル同士の距離が閾値  $Th_m$  を超える場合、入力パターン  $\xi_k$  をSOINN空間中の全てのノード(プロトタイプ)群と充分距離が大きいと見なされる。このような入力パターン  $\xi_k$  を未知のカテゴリのパターンである可能性が高いと見なし、このパターンについての学習強度を強める。

$\xi_k$  から学習された  $k$  番目のHMMの各状態  $j$  の出力分布から  $N_R$  個のランダムサンプル  $p_{k,j,l}$  ( $l \in \{1 \dots N_R\}$ ) を生成し、これをSOINNへ入力する。多次元ガウス分布からランダムサンプリングする方法にはBoxMuller法[6]を用いた。ランダムサンプルから構成される入力パターンは  $\xi'_{k,l} = \{p_{k,1,l}, \dots, p_{k,S,l}\}$   $l \in \{1 \dots N_R\}$  となる。結果として  $\xi'_{k,l}$  がSOINN空間へ入力パターン集合  $IP$  となる。

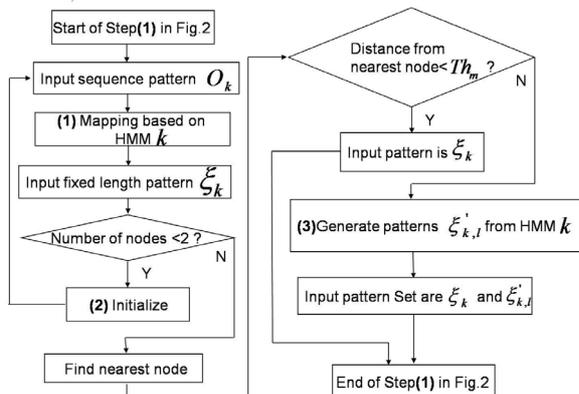


図3: 図2における手順(1)の詳細

る  $\xi_k$  と  $s_1$  の参照ベクトル同士の距離が閾値  $Th_m$  を超えない場合は上記の操作を行わず  $\xi_k$  のみが入力パターン集合  $IP$  となる。

次にAlgorithm 2においてHB-SOINNアルゴリズムの手順について述べる。Algorithm 2の各ステップのインデックスは、図2中のインデックスに対応している。

Algorithm 2: HB-SOINNのアルゴリズム

- (1) Algorithm 1を実行した後、入力パターン集合  $IP$  から一つのパターンがSOINN空間に入力される。
- (2) 入力サンプルに対する勝者ノード (*winner*)  $s_1$  と第二勝者ノード (*second winner*)  $s_2$  を以下の式に従い決定する。 $s_1 = \arg \min_{c \in A} \|\xi - W_c\|$   $s_2 = \arg \min_{c \in A \setminus \{s_1\}} \|\xi - W_c\|$   
 入力サンプル  $\xi$  とノード ( $s_1$  または  $s_2$ ) との距離が類似閾値 ( $T_{s_1}$  または  $T_{s_2}$ ) より大きい場合、入力サンプルを新ノードとして  $A$  に追加する。その後、新しい入力サンプルの学習のためにステップ(2)に戻る。類似閾値  $T$  はノード  $i$  が勝者ノードまたは第二勝者ノードである場合、周辺ノードの位置に応じて  $T_i$  を更新する。
- (3)  $s_1$  と  $s_2$  との間のエッジが存在しなければ、新たに作成して  $C$  に追加する。存在する場合は該当するエッジの年齢を0にリセットする。
- (4)  $s_1$  に繋がる全てのエッジの年齢を加算する。
- (5) 勝者ノードと勝者ノードに隣接するノードの位置ベクトルを、入力パターンに近づける。
- (6) 閾値  $a_d$  を超える年齢のエッジを削除する。その結果、隣接関係を持たないノードが現れた場合は、該当するノードを削除する。
- (7) 入力サンプル数が  $\lambda$  の倍数となった場合、隣接ノードが存在しない孤立したノードを削除する。
- (8)  $IP$  の中に入力パターンがなくなるまで、手順(2)に戻り学習を繰り返す。 $IP$  の中に入力パターンがなくなったら手順(1)に戻る。学習終了時点で特徴空間上に存在するノード集合  $A$  の中で、エッジにより繋がっているノード集合が1つのクラスに対応する。

HB-SOINNについて4つのパラメータ ( $a_d, \lambda, Th_m, N_R$ ) の設定が必要である。 $\lambda$  はノイズとおぼしきノードを削除する周期であり、 $a_d$  はノイズなどの影響で誤って挿入されたエッジを削除する周期である。これらのパラメータの決定方法は4.3節で述べる。ASOINN、SOINN

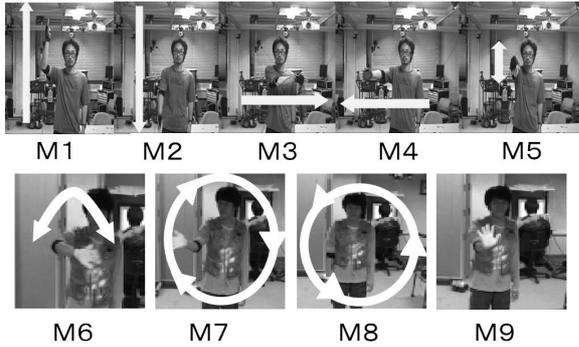


図 4: 実験に使用したジェスチャの例

のアルゴリズムや、アルゴリズムの動作例などの詳細な記述は [2][7] に譲る。

## 4 評価実験

本章では、HB-SOINN の追加クラスタリング精度を検証するため、モーションキャプチャより取得したジェスチャから得られるデータを用いた評価実験を行った。

### 4.1 ジェスチャデータセット

本実験では 2 人の実験者の連続動作からジェスチャ部分（始点と終点）を予め切り出して、データセットを作成した。図 4 にデータセットの例を示す。データセットは 9 種類のジェスチャ(M1-M9) から構成されている。

各ジェスチャは光学式のモーションキャプチャによって計測された。このモーションキャプチャでは 1 秒あたり 30 フレームのデータが取得される。1 フレームあたりに取得されるデータはマーカの三次元位置座標である。実験者は右手に 1 つ、右腕に 1 つ、右肘に 1 つの計 3 つのマーカを装着してジェスチャを行った。時刻  $t$  に得られるマーカ  $i$  の三次元座標を  $u_{i,t} = (x_{i,t}, y_{i,t}, z_{i,t})$  として、これを  $t = 0$  における位置座標で正規化した座標は  $\hat{u}_{i,t} = u_{i,t} - u_{i,0}$  と表せる。したがって式 (1) における時刻  $t$  の特徴ベクトルは 9 次元ベクトル (3 マーカ  $\times$  3 次元) となり、 $o_{k,t} = \{\hat{u}_{1,t}, \hat{u}_{2,t}, \hat{u}_{3,t}\}$  と表せる。

### 4.2 比較手法

HB-SOINN の追加クラスタリング性能を検証するため、追加クラスタリング手法である Adaptive FHMM に基づく手法 [5] とバッチ型クラスタリング手法である階層的クラスタリング [4] との性能比較を行う。

HB-SOINN では学習経過に応じてプロトタイプノードのみを出力する機構を持つ。このため分類に有用なパターンの情報を誤って削除する可能性があり、その場合、後の認識性能を劣化させる。一方で階層的クラスタリングではバッチ的に全ての学習データを用いてクラスタリングを行うため、上記の問題は起きない。こ

の手法と追加クラスタリング手法である HB-SOINN とのクラスタリング精度を比較することで提案手法のクラスタリング精度に対する頑健性を評価する。

#### 4.2.1 Adaptive FHMM

Adaptive FHMM のアルゴリズムを以下で説明する。

*Adaptive FHMM* :

- (1) 新規の入力動作パターン  $O_i$  を HMM  $\lambda_i$  で学習する。
- (2) 現在存在するグループ群と  $\lambda_i$  を Kullback-Leibler 距離 [3] に基づいてモデル同士の類似度を算出し、最近傍のグループを探索し、最近傍グループ  $G_c$  に  $\lambda_i$  を加える。
- (3)  $G_c$  に含まれる動作パターン (HMM) を完全リンクの階層的クラスタリング [4] により分類し、十分に距離の近いパターン集合が存在する場合、そのパターン集合を新しいグループ  $G_n$  として生成し、木構造上の  $G_c$  の直下に子グループとして配置する。
- (4)  $G_n$  に含まれる動作パターン群から HMM および FHMM を学習し  $\lambda_{G_n}$  を推定する。

この手順を繰り返すことで、追加的に入力されるパターン群から自動的に木構造が生成され、クラスタリングが行われる。また [5] では、パターンの数やパターンの分布の分散に応じてマルコフチェーンモデルを追加することで、オンライン学習可能な FHMM を提案している。

このアルゴリズムでは、新しいグループを追加するためのパラメータ  $K_{maxGD}$ 、階層的クラスタリングに用いられる閾値  $K_{cutoff}$ 、クラスを形成するためのパターン数の最小値  $N_{cutoff}$ 、マルコフチェーンを追加するための閾値  $Th_{Addchain}$  を設定する必要があるが、HB-SOINN のパラメータと同様に予備実験 (4.3 節) で決定する。Adaptive FHMM のアルゴリズムの詳細は [5] に譲る。

#### 4.2.2 階層的クラスタリング

階層的クラスタリングはパターン同士の距離を定義可能であれば適用可能であるため、異なる時系列数を持つパターン群のクラスタリングにも利用可能である [4]。本研究では予備実験において検討を行った結果、最もクラスタリング結果が良好であった対称型の漸化式に基づく DTW (Dynamic time warping) をパターン間の距離計算に、階層的クラスタリングの種類に Ward's 法を用いた。この手法を DTW+Ward's と定義する。

#### 4.3 予備実験によるパラメータ設定

HB-SOINN の追加学習性能を評価するための事前実験を行う。M5, M4, M1 の各ジェスチャから各 10 個ずつパターンを取得し, 3 つのジェスチャ × 10 で計 30 個のデータを SOINN に入力した。この 30 個のデータをランダムに一個ずつ与え, クラスタリング精度を purity (式(3)) の大きさに評価した。この実験をデータの入力順序を入れ替えながら 10 回行い。平均の purity が最も高い場合のパラメータを本実験で用いた。

予備実験の結果, HB-SOINN のパラメータは  $\lambda = 200, a_d = 10, Th_m = 30, N_R = 5$  と決定された。また Adaptive FHMM のパラメータは  $K_{maxGD} = 2.0, K_{cutoff} = 0.8, N_{cutoff} = 6, Th_{Addchain} = 80$  と決定され, 両手法に用いる HMM の状態数は 4 と決定された。

#### 4.4 本実験の設定

本実験では 2 人の実験協力者によるジェスチャを 180 パターン (9 ジェスチャ × 20 回) 用意した。本実験では予備実験と同様, ランダムに 1 パターンずつ HB-SOINN および Adaptive FHMM に入力しクラスタリングを行う。クラスタリング後, HB-SOINN ではプロトタイプノードが出力される。180 パターンを最近傍識別し, 各カテゴリにパターンを分類する。クラスタリング精度の計算には purity を用いる。purity は以下の式で算出される,

$$purity(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j| \quad (2)$$

式(3)で  $N = 180$  であり,  $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$  はクラスタリング後のクラスタの集合を表し,  $C = \{c_1, c_2, \dots, c_9\}$  はジェスチャの正解クラスタを表す。

purity は異なるカテゴリのパターンが同じクラスタに存在しなければ高い値をとることから, 各クラスタにつき 1 パターンが分類された場合, つまり 180 個のクラスタが出力された場合に purity は最大値の 1.0 を取ってしまう。したがって, クラスタリングの評価には出力クラス数がどれだけ真のクラスタ数 9 に近いということも purity と同時に評価する。

また追加クラスタリングはデータを与える順番に依存して結果が異なるという, 性質を持つため, データの入力順序を変化させて計 1000 回の実験を行い。データを与える順番に対するクラスタリング精度の頑健性も同時に評価・考察した。

### 5 実験結果

本章では本実験の結果とその考察について述べる。

表 1: HB-SOINN と Adaptive FHMM により得られた purity の平均値の比較

	HB-SOINN	Adaptive FHMM
purity×100[%]	98.02±4.45	85.18±7.95
No.Classes	12.29±1.61	32.36±3.09

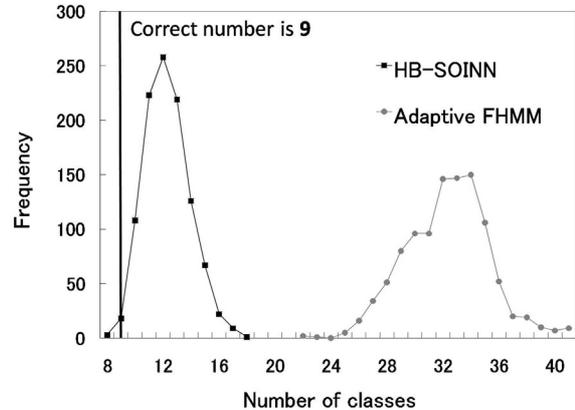


図 5: 1000 回の実験における出力クラス数の分布

#### 5.1 Adaptive FHMM との比較実験結果

表 1 には 1000 回の実験の結果, HB-SOINN と Adaptive FHMM により得られた purity の平均値と出力クラス数の平均をそれぞれ示す。表 1 より purity の平均値の点で HB-SOINN の方が Adaptive FHMM より約 13% 良好であることが示された。

次にクラスタリング後の出力クラス数について議論する。図 5 は 1000 回の実験における出力クラス数の分布を示す。横軸は出力クラス数, 縦軸はその出力クラス数が得られた実験回数 (頻度) をそれぞれ示す。また HB-SOINN の頻度が最も大きかったのは 12 クラスで頻度は 258 回であった。表 1 より, 出力クラス数の幅は最少が 8 で最大が 18 であり平均出力クラス数は  $12.29 \pm 1.61$  であった。

一方 Adaptive FHMM の頻度が最も大きかったのは 34 クラスで頻度は 150 回であった。また出力クラス数の幅は最少が 22 で最大が 50 であり平均出力クラス数は  $32.36 \pm 3.09$  であった。上記の結果から, 実際のジェスチャの正解カテゴリ数は 9 であるため, HB-SOINN の出力クラス数の方が明らかに正確のカテゴリ数に近い値を出力出来ていることが示された。なお Adaptive FHMM では出力クラス数が多くなりすぎているため, クラス数を制御可能なパラメータ  $K_{maxGD}, K_{cutoff}$  を変化させてクラス数を減らす操作を行ったが, その場合, 逆に異なるカテゴリのパターンが同じクラスに分類され Purity が低下するといった現象が確認された。

総じて purity と出力クラス数の比較結果から HB-SOINN のクラスタリング精度における有効性と, デー

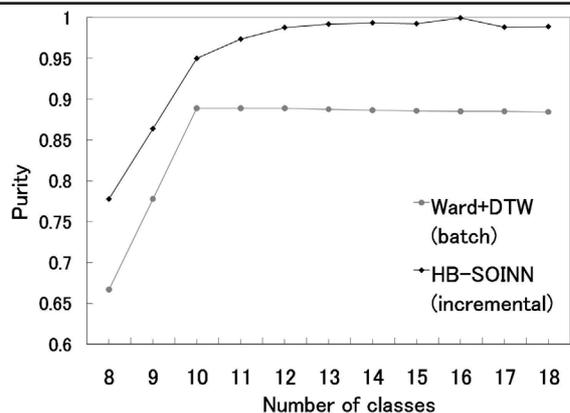


図 6: HB-SOINN と DTW+Ward's 法における出力クラス数が 8 から 18 の場合の purity の比較

タの入力順序に対するクラスタリングの頑健性が示された。

## 5.2 DTW+Ward's 法との比較実験結果

ここでは以下の方法で HB-SOINN との比較を行う。図 5 より HB-SOINN の出力クラス数の幅は最少が 8 で最大が 18 であることが示されたため、階層的クラスタリング法で出力クラス数が 8 から 18 の場合の purity と、HB-SOINN のそれを図 6 において比較する。図 6 より、いずれの出力クラス数における purity も HB-SOINN は DTW+Ward's 法より良好であることが示された。したがって HB-SOINN は追加クラスタリング手法であるにも関わらず、バッチクラスタリング手法の DTW+Ward's 法のクラスタリング精度を上回った。

この結果から 4.2 節で述べた、有用なパターン情報を、誤って削除するケースが最小限に抑えられていることがわかる。180 個のジェスチャパターンの学習後に得られた HB-SOINN のプロトタイプの数、平均 32.5 点 (1000 回の実験) であった。HB-SOINN の purity が比較的によく良好であったことから、HB-SOINN は分類に必要なプロトタイプを生成可能であることが示された。

基本的に階層的クラスタリングはデータ数の 2 乗のオーダーで計算量が増加するため、少数のプロトタイプで認識を行う HB-SOINN は、大規模なデータセットをクラスタリングする場合に、計算量の面で有意性を主張出来る可能性がある。今後、大規模な時系列データセットを用いて計算量に関する検証を行う予定である。

## 6 結論

本研究では個人により表出される多種多様なジェスチャを学習・獲得することを目指し、ジェスチャより得られる時系列パターンの追加クラスタリング手法 HB-SOINN を提案した。

HB-SOINN では HMM を用いて時系列パターンを固定次元特徴ベクトルへ変換し SOINN へ入力する。ま

た HMM の出力分布に従ってパターンをサンプリングにより生成し、これを学習データとして用いることで、未知学習パターンへの学習強度を強める機能を新たに導入した。

ジェスチャデータセットを用いた評価実験では、追加クラスタリング手法の Adaptive FHMM よりもクラスタリング精度 (purity) や出力クラス数の点で有効性を示した。またバッチクラスタリング手法である DTW+Ward's 法よりもクラスタリング精度の点で良好であることを示した。今後はオンラインセグメンテーション手法と HB-SOINN を統合することで、連続動作データからのジェスチャパターンの抽出と追加学習を目指す。

## 参考文献

- [1] Mitra, S. and Acharya, T.: Gesture Recognition: A Survey, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, Vol. 37, No. 3, pp. 311–324 (2007).
- [2] Shen, F. and Hasegawa, O.: An Incremental Network for On-line Unsupervised Classification and Topology Learning, *Neural Networks*, Vol. 19, No. 1, pp. 90–106 (2006).
- [3] Rabiner, L. R.: A tutorial on hidden markov models and selected applications in speech recognition, *Proc. IEEE*, pp. 257–286 (1989).
- [4] T. Warren: Clustering of time series data—a survey, *Pattern Recognition*, Vol. 38, No. 11, pp. 1857–1874 (2005).
- [5] Kulić, D., Takano, W. and Nakamura, Y.: Incremental Learning, Clustering and Hierarchy Formation of Whole Body Motion Patterns using Adaptive Hidden Markov Chains, *Int. J. Rob. Res.*, Vol. 27, No. 7, pp. 761–784 (2008).
- [6] Bishop, C. and service), S. O.: *Pattern recognition and machine learning*, Springer New York. (2006).
- [7] Shen, F. and Hasegawa, O.: A fast nearest neighbor classifier based on self-organizing incremental neural network, *Neural Networks*, Vol. 21, No. 10, pp. 1537–1547 (2008).