

H-037

連続的非言語インタラクション系列のオンライン分節化とクラスタリングに基づく非言語パターン発見

Gesture pattern discovery based on clustering and on-line segmentation of continuous interaction data.

岡田 将吾 †
Shogo Okada†

石橋 聡 †
Satoshi Ishibashi†

西田 豊明 †
Toyoaki Nishida†

1 はじめに

人間同士のコミュニケーションにおける非言語情報の分析において、分析者は分析対象となる非言語パターンを大量のインタラクションデータから分節化しアノテーションを行わなければならない。この分析作業の負担を軽減するため、本研究では、多次元時系列データから頻出パターンを高速に抽出可能な手法を提案する。非言語パターンの長さは予め特定出来ないため、多次元時系列データから客観的な最小単位である基本構成要素をオンラインで分節化・抽出し、抽出した基本構成要素を HB-SOINN[1] を用いてオンラインでクラスタリングし、基本構成要素の抽出・ラベリングを行う。

評価実験では、提案手法は従来のオンラインセグメンテーション手法に比べて、連続非言語データから構成要素を高速かつ高精度でセグメンテーションを行うことが可能であることを示す。また提案手法で得られた構成要素の列から、頻出する非言語情報を発見可能であることを示す。

2 提案手法

本研究では、2段階構成のシステムによって多次元時系列データから基本構成要素を抽出する。本手法のシステム構成を図1に示す。1段階目のセグメンテーションの手法には、SWAB アルゴリズム [2] を、多次元時系列データからの基本構成要素検出が可能のように拡張した、MD-SWAB(Multi-Dimensional SWAB) を用いる。2段階目のクラスタリング部では、セグメンテーション部から出力される時系列データのセグメントに対してクラスタリングを行う。クラスタリングの手法には、オンライン教師無しクラスタリング手法である HB-SOINN を用いる。次節以降では、システムを構成する各コンポーネントについての詳細を述べる。

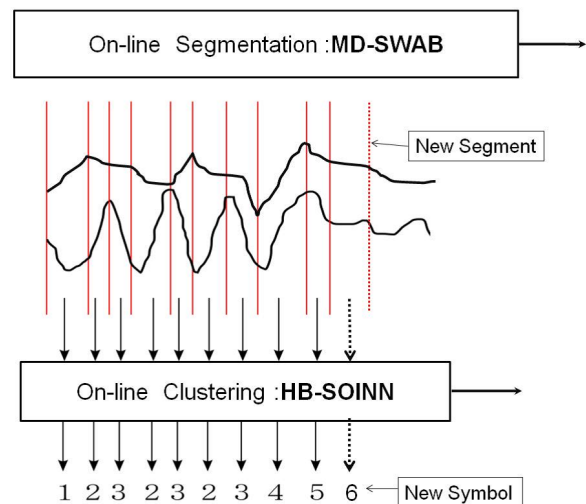


図 1: The overview of proposed system

2.1 SWAB アルゴリズム

SWAB アルゴリズムは、Sliding Window アルゴリズムと Bottom-Up アルゴリズムを基盤とするアルゴリズムである。まず Sliding Window アルゴリズムでは、ある地点から窓幅を広げていき、窓幅に含まれる区間の時系列データを、直線近似した際の二乗誤差が閾値を超えた場合に、その区間の終点をセグメント点とする。

一方 Bottom-Up アルゴリズムでは、時系列データを最小のセグメント列の集合と見なし、統合した際の誤差が最少になるセグメント同士を統合する処理を行う。ここで得られる最少誤差がある閾値を上回るまで、この処理を繰り返す。両手法における近似誤差は最小二乗法によって算出される。

Sliding Window アルゴリズムはオンライン処理が可能であるが、データの局所的な部分のみを観測して近似直線を求めるため、入力データによっては良い近似の精度が得られないことがある。一方 Bottom-Up アルゴリズムは大局的視点からデータ全体を観察して処理を行うため、Sliding Window に比べよいデータの近似が得られるが、バッチアルゴリズムであるためにデー

† 京都大学 情報学研究科 知能情報学専攻

タ全体を観測しなければ動作せず，オンライン処理が出来ない．両者の手法の利点を統合した手法として提案された SWAB アルゴリズムが提案された．

以下で SWAB アルゴリズムの詳細を述べる．

- Step1. 大きさ w のバッファを確保する．バッファの初期サイズ w_0 は，バッファ内にセグメントが 5, 6 程度含まれるように定める．
- Step2. バッファ w に対して Bottom-Up アルゴリズムを実行する．
- Step3. 2 の結果から左端のセグメントを結果として出力し，バッファから取り出す．
- Step4. バッファの右端から Sliding Window アルゴリズムの手法によって仮のセグメント点を検出し，その点までのデータをバッファの右に追加する．
- Step5. Step2 ~ Step4 の操作をデータが入力される間繰り返す．

バッファが無限に大きくなると SWAB アルゴリズムは Bottom-Up アルゴリズムに近くなり，現在のフレームと出力されるセグメントとの時刻差が大きくなってしまふ．逆にバッファが小さくなりすぎると Sliding Window アルゴリズムに近くなり精度が劣化する．以上の問題を避けるため，バッファの大きさには上限と下限を設定する．今回は上限を $2w_0$ ，下限を $w_0/2$ と設定した．

2.2 MD-SWAB アルゴリズム

本節では，SWAB アルゴリズムを多次元時系列データに適用して基本構成要素の検出ができるように拡張した手法，MD-SWAB(Multi-Dimensional SWAB) の詳細について述べる．MD-SWAB は，入力される多次元時系列データの各次元に対して並行して SWAB アルゴリズムを実行し，それぞれのセグメント結果を統合することで，基本構成要素の分節点の検出を行う．

まず，多次元時系列データの各次元に対して並行して SWAB アルゴリズムを実行し，次元毎のセグメント列を得る．次に，それぞれのセグメント列をデータ変換部で merge-cost というデータに変換し，さらにそれらを足し合わせた sum-error という時系列データを得る．sum-error を分節点出力部に入力することで，多次元時系列データのセグメント列を得る．

MD-SWAB アルゴリズム

- Step1. 入力データの各次元に対し，並行して SWAB アルゴリズムを実行する．

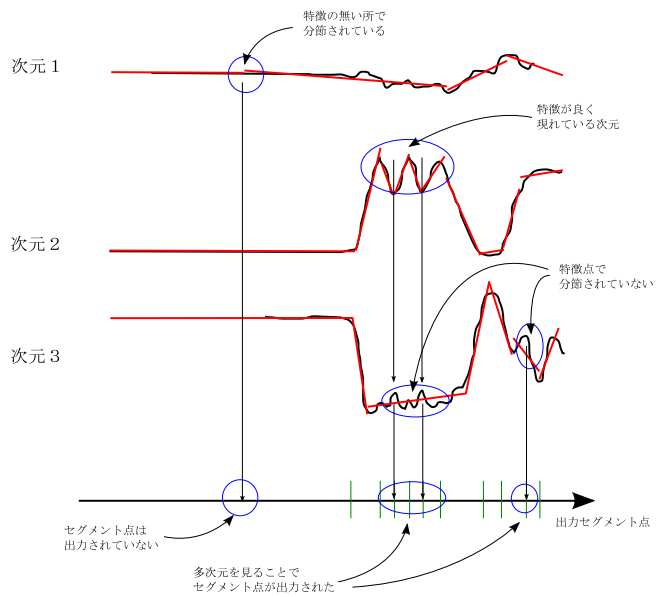


図 2: ノイズが取り除かれている図

- Step2. 次元毎の SWAB アルゴリズムが出力する分節点について，merge-cost(分節点の前後のセグメントをマージした区間のエラー値) を算出して保存する．
- Step3. 入力時系列データを，各フレームにおいていずれかの次元で分節点が出力された場合はその点についての merge-cost，また分節点がない場合は 0 を値とする時系列データ (以下，"sum-error") に変換する．同フレームにおいて複数次元で分節点が出力されている場合は，それぞれの merge-cost を足し合わせた値を取るものとする．
- Step4. 窓枠の大きさ W と比例定数 C を設定する．sum-error について，窓内のフレーム辺りの値を算出する．その値が $C \times \text{max-error}$ を超えた場合，窓の右端を最終的なセグメント点とする．

時系列データ sum-error に対し窓 W 内の密度を算出し，閾値を超えた所でセグメント点を出力し，窓を移動させる．

動作の特徴がよく現れているセグメント点に対する merge-cost の値は大きくなり，ノイズなどの無意味なセグメント点に対する merge-cost の値はそれほど大きくなりません．従って，一次元のみから出力されたノイズのような分節点を含む窓枠内では sum-error の値が小さくなり，その分節点を取り除くことができる．

また逆に，特徴量が小さく検出されなかった分節点は，他に大きく特徴量の表れている次元があれば，検出することができる．多次元のセグメンテーション結

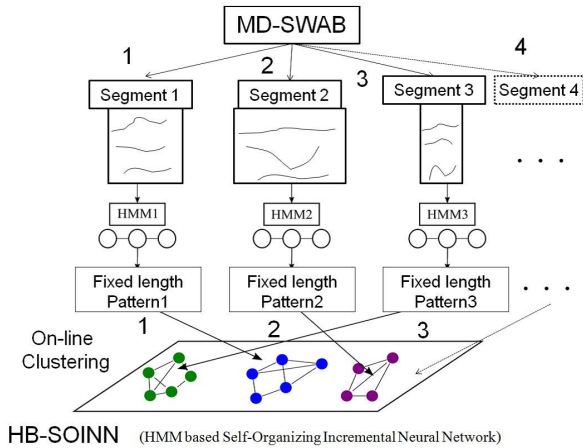


図 3: HB-SOINN の概要

果を統合することで具体的に上記のそのような点が取り除かれている様子を、図 2 に示す。MD-SWAB で設定すべきパラメータは閾値 $\max\text{-error}$ ，初期バッファサイズ w_0 ，窓枠 W ，比例定数 C の 4 つである

2.3 HB-SOINN による構成要素のラベリング

HB-SOINN(HMM Based SOINN) [1] は、HMM を用いて固定長パターンのオンライン教師無し学習手法である SOINN [3] を、時系列データの異なるデータに適用できるように改良した手法である。図 3 に、HB-SOINN の概要を示す。

- Step1. セグメンテーション部からオンラインで出力される時系列データのセグメントを入力として受け取る。
- Step2. 入力データに対してオンラインでクラスタリングを行う。
- Step3. 入力された時系列データの各セグメントがどのクラスタに属するかを出力させることで、基本構成要素列を得る。

とう手順によって、セグメンテーション部から出力される時系列データのセグメントをクラスタリングし、基本構成要素の列を出力する。

3 評価実験

本研究の提案したセグメンテーション手法の精度を検証するために、モーションキャプチャにより取得したジェスチャデータを入力としてセグメンテーションを行った。さらに Kohlmorgen によって提案されたオンラインセグメンテーション手法:Dynamic HMM Online Segmentation(DHOS)[4] を用いて同じ設定で実験を行

い、本手法との性能比較を行った。インタラクション行動には、指示者が行動者を連続ハンドジェスチャによって移動させるというタスクを設定した。ハンドジェスチャの 3 次元座標データを取得するため、マーカは指示者の右手の指先・手の甲・肘に 2 箇所、計 4 点に貼り付けて計測した。取得したデータは、60fps で約 1 分 40 秒の 12 次元の時系列データである。

また、軸毎に線の色を、マーカ毎に線のパターンを同じものにしていく。今回の実験ではジェスチャ者は一定の地点に固定した。さらに両肩にマーカを貼り付け、フレーム毎に 3 次元座標軸をジェスチャ者の肩のラインに合わせて回転させることで、ジェスチャ者の向きが変わったり移動したりしても、同じジェスチャをした場合は同じ 3 次元座標データに変換することが可能である。また、セグメンテーションは各マーカの 3 次元座標の時系列データのみから行い、ビデオデータやジェスチャに対応して行われた行動の意味などのジェスチャの意味に関連する情報はセグメンテーション部には入力しないものとする。今回の実験では、基本構成要素のセグメンテーションに対する正解データを手動で作成した。正解データを作成する際には、ジェスチャの 1 ストロークを基本構成要素だとみなし、ビデオデータと波形を同時に閲覧しながら、主に波形に山と谷がよく現れている点を、基本構成要素の分節点の正解とした。

3.1 セグメンテーション精度の評価結果

本手法から得られたセグメンテーション結果と、正解データとを比較することで本手法の評価を行う。評価には、再現率と適合率、及び F 値を用いる。セグメンテーション結果に対する再現率を R ，適合率を P ，F 値を F とすると、 R, P, F は以下の式で算出される。

$$R = \frac{\text{正解と判定された結果の個数}}{\text{正解データの全セグメントの個数}} \quad (0 \leq R \leq 1)$$

$$P = \frac{\text{正解と判定された結果の個数}}{\text{出力された全セグメントの個数}} \quad (0 \leq P \leq 1)$$

$$F = \frac{2 \times R \times P}{R + P} \quad (0 \leq F \leq 1)$$

本手法によって得られた各セグメント点が正解かどうかを判定するため、正解データとセグメント点の許容誤差 t を定める。正解データのセグメント点とのフレームの差が $t = 10$ 以内であれば、本手法が出力したセグメント点は正解であると判定する。

実験に用いた MD-SWAB のパラメータは $w_0 = 180, Th = 250000, W = 10, C = 20$ ，HB-SOINN のパラメータは $\lambda = 100, a_d = 10, Th_m = 10$ と事前実験の結果より設定した。再現率・適合率・F 値を表 1 に示す。一般に再現

	再現率	適合率	F 値
MD-SWAB	0.776	0.731	0.753
DHOS	0.832	0.416	0.555

表 1: 再現率・適合率・F 値

率が高くなると適合率が低く、適合率が高くなると再現率が低くなる傾向があるため、両方の値を組み合わせた F 値によって比較する。表 1 の通り、比較手法では再現率は高い値を示したが適合率は低くなっているため、F 値を見ると精度が良くないことが分かる。一方提案手法では再現率・適合率共にある程度の値を保っており、F 値を見ても。

両者の F 値を比較すると、提案手法の方が再現率・適合率を共に考慮した時により精度でセグメンテーション結果が得られたことが分かる。さらに本実験タスクを同性能の計算機 (CPU: Intel Xeon 3.0GHz) で計算したところ、MD-SWAB では、全てのセグメンテーションを完了するのに 121 秒、DHOS では 1867 秒を要した。この結果より MD-SWAB の計算時間は DHOS の計算時間に比べて明らかに高速であることが示された。

3.2 構成要素のラベリング結果

MD-SWAB によってオンラインでセグメントされた構成要素群は HB-SOINN によってクラスタリングされインデックス (シンボル) が付与される。最終的にジェスチャから得られる多次元時系列データはシンボル列として出力される。ここでは同じ動作から得られたシンボル列同士が類似しているかどうかを考察する。指示者が行ったジェスチャの内、(1) "come on", (2) "to right", (3) "go there", (4) "no motion" について結果を記載する。表 2 には各ジェスチャ区間から出力されたシンボル列を記載した。表 2 より同カテゴリのジェスチャ区間から出力されたシンボル列 p は類似していることがわかる。この結果より同じカテゴリのジェスチャパターンが表すシンボル列を自動的に発見出来る可能性が高い。しかしながら他のジェスチャについては類似していないシンボル列が出力されたケースもあり、これについては今後の課題とする。

4 結論

本論文では、人間の表出するジェスチャなどの多次元時系列データのオンラインセグメンテーションを行い、得られた各セグメントに対してオンラインクラスタリングを行うことによって、基本構成要素を抽出する手法について述べ、関連手法と比較することで提案

Kind	Num	label
come on	2	$p_1: (7\ 6\ 7\ 8\ 6\ 6\ 7\ 8\ 7)$ $p_2: (7\ 13\ 8\ 8\ 6)$
to right	4	$p_1: (4\ 5\ 4\ 5\ 4\ 5\ 4)$ $p_2: (11\ 5\ 4\ 5\ 4)$ $p_3: (12\ 5\ 4\ 5\ 12\ 5)$ $p_4: (26\ 5\ 4\ 5\ 4\ 5)$
go there	5	$p_1: (3\ 3\ 15\ 3)$ $p_2: (3\ 15\ 3\ 15\ 3)$ $p_3: (3\ 3\ 15\ 3\ 15\ 3)$ $p_4: (3\ 15\ 3\ 15\ 3\ 6)$ $p_5: (3\ 15\ 3\ 15\ 3\ 15\ 3)$
no motion	12	10 patterns are $p_1 - p_{10}(1)$ 2 patterns are $p_{11}, p_{12}(21)$

表 2: 各ジェスチャ区間から出力されたシンボル列

手法の有効性を検証した。またセグメンテーションの結果、得られるシンボル列からジェスチャパターンを発見出来る可能性を示した。今後、表 2 に示したシンボル列を自動的に抽出するモチーフ発見 [5] の手法を適用する予定である。

参考文献

- [1] Okada, S. and Nishida, T.: Incremental clustering of gesture patterns based on a self organizing incremental neural network, *IEEE International Joint Conference on Neural Networks(IJCNN 2009)* (2009).
- [2] Keogh, E., Chu, S., Hart, D. and Pazzani, M.: An On-line Algorithm for Segmenting Time Series, *IEEE International Conference on Data Mining*, Vol. 0, p. 289 (2001).
- [3] 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).
- [4] Kohlmorgen, J. and Lemm, S.: A Dynamic HMM for On-line Segmentation of Sequential Data, *Advances in Neural Information Processing Systems(NIPS)* (2001).
- [5] Chiu, B., Keogh, E. and Lonardi, S.: Probabilistic discovery of time series motifs, *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 493-498 (2003).