Kobe University Repository : Kernel

PDF issue: 2026-07-25

AMSS: 時系列データの効率的な類似度測定手法

中村, 哲也
瀧, 敬士
野宮, 浩揮
上原, 邦昭

(Citation)
電子情報通信学会論文誌. D, 情報・システム,J91-D(11):2579-2588

(Issue Date)
2008-11-01

(Resource Type)
journal article

(Version)
Version of Record

(Rights)
Copyright (c) 2008 IEICE

(URL)
https://hdl.handle.net/20.500.14094/90000827

KOBE UNIVERSITY

論　文

AMSS：時系列データの効率的な類似度測定手法

中村　哲也[†a)]　瀧　敬士[†]　野宮　浩揮[†]　上原　邦昭[†b)]

AMSS: A Similarity Measure for Time Series Data

Tetsuya NAKAMURA[†a)], Keishi TAKI[†], Hiroki NOMIYA[†], and Kuniaki UEHARA[†b)]

あらまし　近年，時系列データの分類に関する研究が盛んに行われている．しかし，既存の類似度測定手法は，データ整合，ノイズ，計算コストなどの問題がある．また，座標値のみの計算ではあらゆる時系列データを安定して分類できないという問題もある．これは，時系列データがもつ波形の振幅，振動数，概形などの特徴も考慮する必要があるからである．本論文では，前者の解決のために類似度測定手法 Angular Metrics for Shape Similarity（AMSS）を提案する．AMSS は時系列データをベクトル列として扱い，ベクトルの角度を比較して類似度を計測している．角度の比較にはコサイン類似度を用い，ノイズのような類似しない部分を無視している．また，動的計画法で系列間の類似度を計算して，データ整合の問題を解決している．一方，後者の解決のために，それぞれの特徴に合わせた分類アルゴリズムを複数用意し，アンサンブル学習の枠組みを導入したメタアルゴリズムにより，それらを組み合わせて分類を行う．その結果，個々の識別器を単独で利用するよりも分類精度が向上することを示す．

キーワード　データマイニング，時系列，類似度，分類

1. まえがき

時系列データマイニングでは，分類，クラスタリング，インデクシングのために，系列間の類似度を計測する様々な手法が提案されている [1], [2]．近年，国際会議において時系列分類の精度を競うワークショップ [3] が開催されるなど，再び注目を集めている．ワークショップでは，物体の形状や，画像の輪郭などのデータを時系列として分類する手法なども提案され，従来の音声認識等の分野だけでなく，他の分野でも用いられるようになっている．

時系列データマイニングでは，一般的な類似度測定手法として，ユークリッド距離が知られている [4]．しかし，長さの違う時系列を対象とする場合，系列間の距離を計測することができないという問題がある．この問題点を解決するために，Dynamic Time Warping（DTW）が用いられている [5]．DTW は時系列の長さを整合し，対応するデータ点を調べて比較するアルゴリズムである．時系列間の類似度は，データ点のすべての組合せの類似度を格納した類似度行列を構築し，その中で最適な組合せとなるワーピングパスを発見して，類似度を足し合わせている．また，大規模データベースを対象とする場合には計算時間がばく大となるため，木構造を利用したインデクシングにより計算の効率化を図るなど，アルゴリズムを改良する研究も数多くされている [13]．

しかし，DTW には重大な問題点がある．まず，完全にすべてのデータ点をマッチングさせる性質から，ノイズに弱いという問題がある．ノイズを含むデータに対しては，最長共通部分列（LCSS）に基づく類似度が提案されている [1]．LCSS は類似しない軌跡の部分列は無視し，類似する部分列だけで類似度を計測する．このため，LCSS は DTW よりもノイズに強く，実世界で移動する物体の軌跡などの類似度測定に用いられている [6]．しかし，LCSS の類似度は，時系列中でのマッチングする部分列の割合で表され，どのくらい部分列が似ているかという情報はもっていない．したがって，部分的に完全に一致している時系列よりも，全体的におおよそ類似している時系列の方が，類似度が高くなることがある．また，部分列が似ているかどうか判断するために，しきい値を設定する必要がある

† 神戸大学大学院工学研究科，神戸市
Graduate School of Engineering, Kobe University, 1–1 Rokkodai-cho, Nada-ku, Kobe-shi, 657–8501 Japan
a) E-mail: nakamura@ai.cs.scitec.kobe-u.ac.jp
b) E-mail: uehara@kobe-u.ac.jp

が，データに合わせて一意に決定することは困難である．

更に，DTW と LCSS はユークリッド空間中で時系列の座標値を用いるため，類似する時系列データ間でも，座標間の相対位置が異なれば類似度が適切に求められないという問題がある．例えば，図 1 の Sequence 1，Sequence 2 は同じ軌跡だが，比較すべき座標値はずれ，その向きも異なるため，類似軌跡と判定されないことも起こり得る．このような軌跡の類似度を適切に計測するためには，正規化が必要になる．

上記の問題を解決するために，本論文では，時系列データ間の類似度を計測する新たな手法 Angular Metrics for Shape Similarity（AMSS）を提案する．本手法では，DTW のように動的計画法を実装してデータ整合の問題を解決している．また，座標系の影響を受けずに類似度を計測するために，時系列データをベクトルの系列として扱っている．類似度の計測では，ベクトルの方向に着目している．すなわち，方向はベクトル間のコサイン類似度で評価でき，どれだけ類似しているかを計測できる．更に，コサイン類似度によって，ノイズを無視することができる．具体的には，明らかに方向の違うノイズとの類似度は 0 となるため，ノイズが無視されることになる．この特徴は LCSS と同様だが，しきい値設定が必要ないという利点がある．

一方，時系列データによっては，座標値による類似度測定のみでデータ間の類似性をうまく取り扱えるとは限らない場合がある．このため，波形の振動数に柔軟に対応することを目的として，波形の角度から類似性を計測する手法も存在する [15]．また，波形の概形を評価するためにデータの平均値やメジアン値，周波数で評価するためにウェーブレット係数など，数多くの特徴を利用するアプローチも存在する [12]．しかしながら，多く特徴を用いる場合には，どのように適切な特徴を選択するかという問題が生じてくる．

このような問題を解決する手法として，アンサンブル学習の枠組みを導入する．一般に，アンサンブル学習で用いられる学習器自体は単純で，計算量も妥当なものであるが，異なる特徴を用いて各学習器を学習させ，それらを組み合わせてより複雑な学習モデルを実現しようとしている．本研究でも，アンサンブル学習の一種である Boosting に基づくアプローチを利用して，与えられたデータに対して適切な特徴を選択し，分類精度の向上を目指す手法を提案する [11]．最後に，本手法の正当性を評価する実験を行い，類似度測定における効果を示す．

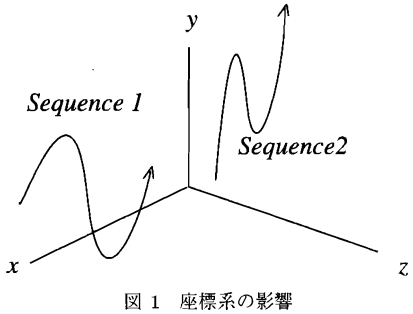

図 1　座標系の影響
Fig. 1　Effect of coordinates.

2. Angular Metrics for Shape Similarity

AMSS は時系列データをベクトル系列に変換し，系列間でベクトルを比較して類似度を計算する手法である．ここで，AMSS がどのように類似度を計算するかを示す．時系列を，データ長 m の C とし，これら隣接する点を直線で結ぶ．

$$C = c_1, c_2, \ldots, c_m \tag{1}$$

時系列中の点の座標は，原点からのベクトルで表される位置ベクトルと等しい．ここで，ベクトル $\vec{c}_1$ は $(c_2 - c_1)$ と表すことができるため，時系列 C は，ベクトル系列 $\hat{C}$ と等価に表現することができる（図 2）．

$$\begin{aligned}\hat{C} &= (c_2 - c_1), (c_3 - c_2), \ldots, (c_m - c_{m-1}) \\ &= \vec{c_1}, \vec{c_2}, \ldots, \vec{c_M} \qquad (M = m - 1)\end{aligned} \tag{2}$$

同様に，Q を長さ n の時系列とする．

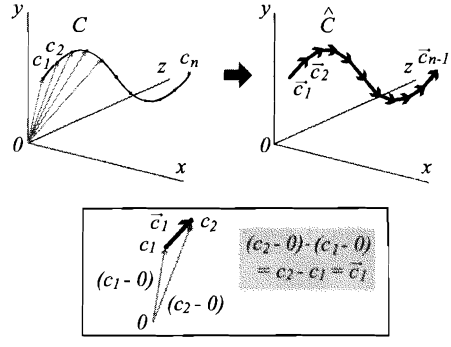

図 2　時系列データのベクトル系列への変換
Fig. 2　A time series is converted to a vector sequence.

$$Q = q_1, q_2, \ldots, q_n \tag{3}$$

$$\hat{Q} = (q_2 - q_1), (q_3 - q_2), \ldots, (q_n - q_{n-1}) = \vec{q_1}, \vec{q_2}, \ldots, \vec{q_N} \quad (N = n-1) \tag{4}$$

ベクトル $\vec{q_i}$ と $\vec{c_j}$ はそれぞれ $\hat{Q}$ と $\hat{C}$ に含まれているものとする（図 3(a)）．また，θ は q_i と c_j の間の角と仮定する（図 3(b)）．このとき，ベクトル $\vec{q_i}$, $\vec{c_j}$ 間の類似度は以下のように計算される．

$$Sim(\vec{q_i}, \vec{c_j}) = \cos\theta = \frac{\vec{q_i} \cdot \vec{c_j}}{|\vec{q_i}||\vec{c_j}|}$$

$$Sim(\vec{q_i}, \vec{c_j}) = 0 \quad (\text{if } \theta > \pi/2) \tag{5}$$

なお，ベクトルの方向がどれだけ類似しているかを表すため，ここではコサイン類似度を用いている．

類似度を計測した例を図 4 に示す．図 4(a) において，$|\vec{c_1}|$ と $|\vec{q_1}|$ はそれぞれ長さが 1 と $\sqrt{2}$ である．$\vec{c_1}$ と $\vec{q_1}$ の間の角度は $\pi/4$ となる．したがって，$Sim(\vec{c_1}, \vec{q_1})$ は 0.71 と計算される．同様に，図 4(b) の $\vec{c_1}$ と $\vec{q_2}$ の類似度 $Sim(\vec{c_1}, \vec{q_2})$ は 0.50 となる．この結果は，$\vec{q_1}$ が $\vec{q_2}$ より $\vec{c_1}$ に似ていることを示している．

更に，もし $\theta > \pi/2$ であれば，$Sim(\vec{q_i}, \vec{c_j})$ は無条件で 0 となる．図 4(c) において，ベクトル $\vec{c_1}$ と $\vec{q_3}$ の方向は極めて異なっている．それゆえ，$Sim(\vec{c_1}, \vec{q_3})$ は 0 となり，同様にベクトル間の角度が $\pi/2$ 以上の組合せでは，類似度は 0 としている．このように，角度を比較することによって，類似しない部分系列を特定することができる．結果として，ベクトル間の角度に基づく類似度を通して，本手法の有効性を見ることができる．

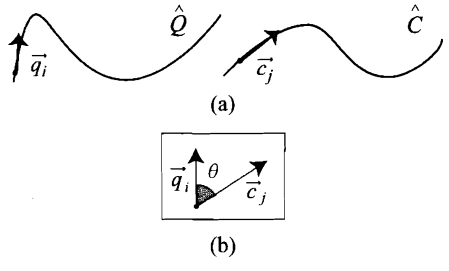

図 3　ベクトル間の類似度測定
Fig. 3　How to measure the similarity between two vectors.

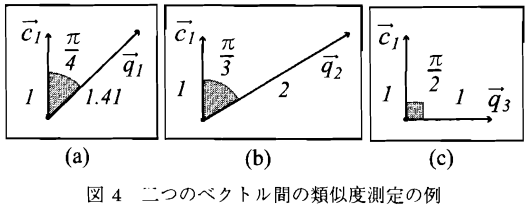

図 4　二つのベクトル間の類似度測定の例
Fig. 4　An example of measuring the similarity between two vectors.

比較する時系列中のベクトル同士のすべての組合せについて類似度を計算したあと，それらを足し合わせて，$\hat{Q}$ と $\hat{C}$ の類似度が求められる．この計算には動的計画法を用いて，全体の類似度が最大となるように，対応するベクトルの組合せを決定している．

$\hat{Q}$ (frame 1 to frame i) と $\hat{C}$ (frame 1 to frame j) の類似度は以下の式で計算される．

$$\begin{aligned} S(i,j) = Max[&S(i-1,j-1) + Sim(i,j),\\ &S(i-1,j) + Sim(i,j),\\ &S(i,j-1) + Sim(i,j)] \end{aligned} \tag{6}$$

$$S(i,1) = S(1,j) = -\infty$$

$$S(1,1) = Sim(1,1)$$

ここで，$S(i,j)$ は $\hat{Q}$ の frame i と $\hat{C}$ の frame j までの最大の類似度となっている．したがって，長さ N, M の時系列 Q, C 間の類似度は，$S(N,M)$ で計算される．

$$AMSS(\hat{Q}, \hat{C}) = S(N,M) \tag{7}$$

2.1　動的計画法によるノイズの無視

式 (6) で最大の類似度が得られるように，系列間で対応するベクトルの組合せを考える（図 5(a)）．この対応を，各ベクトル間の類似度を要素にもつ行列中に示したものをワーピングパスと呼ぶ（図 5(b)）．ワーピングパスによって，時系列間の対応が決定され，類似度が計算される．以下では，DTW や LCSS と比較して，AMSS でのノイズを無視する方法を説明する．

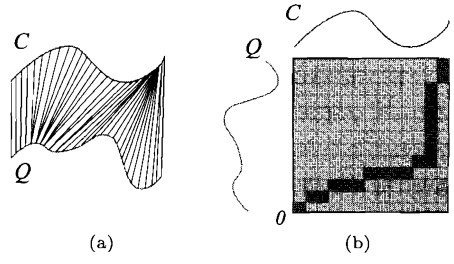

図 5　(a) ベクトル対応の例 (b) ワーピングパス
Fig. 5　(a) An example of correspondence of vectors. (b) An example of warping path.

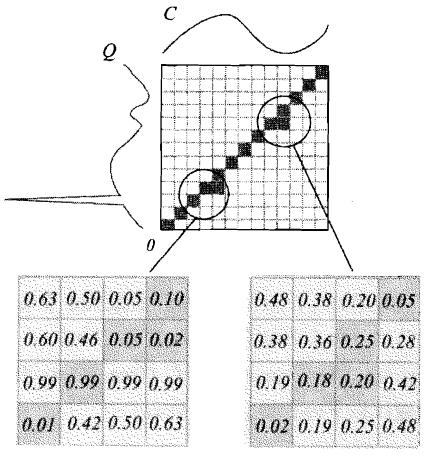

図 6 DTW のワーピングパスの例
Fig. 6 An example of warping path in DTW.

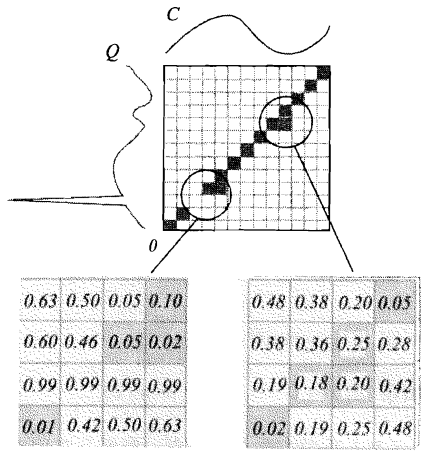

図 7 LCSS のワーピングパスの例
Fig. 7 An example of warping path in LCSS.

DTW, LCSS, AMSS におけるワーピングパスを，それぞれ図 6, 図 7, 図 8 に示す．なお，時系列 Q は時系列 C に類似しているが，前半部分にはノイズを含み，後半部分では軌跡が少し異なっているものとする．

図 6 における数値は，時系列中データ間の距離を示し，0 に近いほど類似していることを表している．距離が最小になるようにワーピングパスをとっているが，DTW ではすべての点に関して対応を求める完全性により，ノイズとの距離も含めなくてはならない．このため，図 6 の左下にあるように，0.99 の距離さえも加算され，全体の距離に影響を及ぼしている．一方，後半部分においては，軌跡は異なっていても，ユークリッド空間中での距離が近いため，距離は 0.2 程度になっている．このように，たとえ類似しない軌跡であっても，時系列データの座標によっては類似しているとみなされることある．

次に，LCSS の例を示す．数値は DTW と同様に距離を示している．LCSS では，しきい値によって類似するかどうかを決定するため，図 7 の左下のような極端に距離の離れているノイズは確実に無視できる．しかし，右下に示された，軌跡の異なる部分では，しきい値によってノイズとみなされたり，類似する部分列とみなされることがある．例えば，しきい値が 0.3 以上の場合には，類似していない部分列にもかかわらず，類似しているとみなされてしまう．このように，しきい値設定には困難さが伴うことが分かる．

最後に，AMSS のワーピングパスの例を示す．AMSS ではベクトル間の類似度を計測するため，数値は 1.0 に近いほど類似していることになる．逆に，図 8 の左下のように，ノイズの類似度は 0.0 となっている．ワーピングパスにおいては，DTW と同様にすべての対応を求める完全性をもっているが，ノイズの類似度は 0.0 であるため，加算されても全体の類似度には影響を及ぼさない．このことから，座標系において距離が近くても，軌跡が違う，すなわち方向が違う部分系列に関しては，類似しないものとして認識できることが分かる．

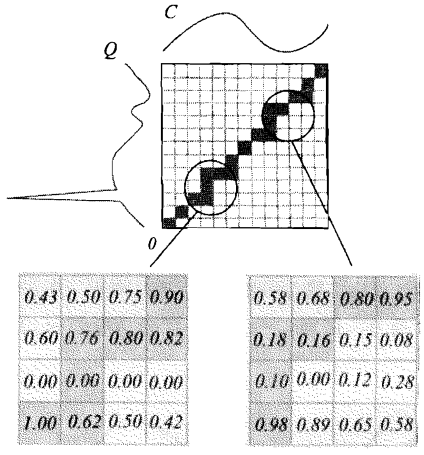

図 8 AMSS のワーピングパスの例
Fig. 8 An example of warping path in AMSS.

2.2 ワーピングパス制約

正しい分類には，適切なワーピングパスの選択が必要となる．しかし，系列によっては間違った対応がとられる場合もある．図 5 では，系列 Q の前半部分が

系列 C のほとんどのベクトルと対応づけられてしまい，Q の後半部分の多くのベクトルは C の一つのベクトルに対応づけられている．このようなベクトルの対応が起こると，同じクラスにもかかわらず，違うクラスとして分類されたり，逆に異なるクラスのものが類似していると認識されることがある．このような問題を解決するために，ワーピングパスに制約を設ける手法が提案されている．

本研究では，板倉らの制約 (Itakura Parallelogram) を用いている [7]．本制約では，式 (6) を式 (8) に置き換えて実装されている．このワーピングパスの制約により，一つのベクトルにいくつものベクトルが対応されることを防いでいる．

$$S(i,j) = Max[S(i-1,j-1) + 2Sim(i,j),$$
$$S(i-2,j-1) + 2Sim(i-1,j) + Sim(i,j),$$
$$S(i-1,j-2) + 2Sim(i,j-1) + Sim(i,j)] \quad (8)$$

実際のパスの様子を，図 9 に示す．水平のパスをとり続けることは，例えば，時系列 Q の j 番目のベクトルだけに C のベクトルが対応することを表している．垂直のパスをとり続けることはその逆である．これは，前述の一つのベクトルに多くのベクトルが対応される誤った対応づけであり，避けなければならない．

図 9 (a) に見られるように，板倉らの制約では，(i,j) に到達するためのパスを以下の三つに制限している．

（1） $(i-1,j-1)$ から (i,j) への斜めのパス

（2） $(i-2,j-1)$ から $(i-1,j)$ への斜めのパスをとり，次に水平のパスをとる

（3） $(i-1,j-2)$ から $(i,j-1)$ への斜めのパスをとり，次に垂直のパスをとる

この制約により，ワーピングパスがとり得る範囲は図 9 (b) の灰色の範囲に制限される．この制約を用いて，ベクトルの間違った対応づけを防ぎ，更には類似度測定の範囲を限定して，計測の効率化につながるようになっている．

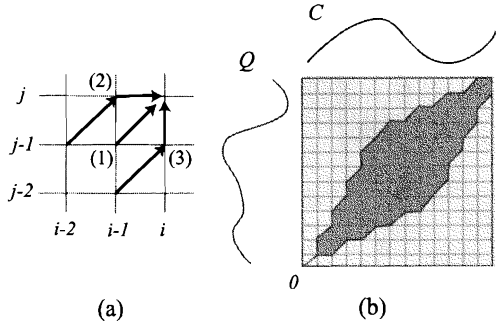

図 9 ワーピングパス制約 'Itakura Parallelogram'
Fig. 9 A warping path constraint 'Itakura Parallelogram'.

2.3 メタアルゴリズム

既に述べたように，時系列データを単一の座標値のみで表現しても，うまく類似性を測定できない場合がある．このため，多くの特徴を組み合わせて，様々な時系列データの分類精度を向上させるために，Boosting に基づくアンサンブル学習の枠組みを導入する [11]．アンサンブル学習とは，複数の学習器を組み合わせて，より性能の良いアンサンブル学習器を構成する手法である．

本アプローチは通常の Boosting とは次の 2 点が大きく異なっている．まず，通常の Boosting では 2 値問題を対象としているが，本アプローチでは複数のクラスを扱えるよう，多値問題に拡張している．更に，Boosting では単一の分類アルゴリズムを用いるのに対し，本アプローチでは多くの特徴を扱うために複数の分類アルゴリズムを組み合わせている．

具体的には，まず，それぞれの特徴ごとに分類アルゴリズムを用意し，訓練データを分類する．訓練データには分類の難しさを表す重みが与えられる．分類過程では，ある分類アルゴリズムで誤分類だったデータはその他の分類アルゴリズムに送られる．逆に，他の分類アルゴリズムで誤分類だったデータはこの分類アルゴリズムに送られる．この際，他の分類アルゴリズムから送られてきた事例には大きい重みが与えられる．以上の操作を 1 回のラウンドとして，本研究では，この訓練データによる分類過程を 5 ラウンド繰り返している．そして，分類アルゴリズムがテストデータを分類する場合，類似した訓練データにおける分類精度が高ければ，この分類結果は信頼できるとみなされる．逆に分類精度が低ければ，他の分類アルゴリズムの結果が採用される．結果として，各分類アルゴリズムは自身にとって正確に分類できるデータを分類するように特化されることになる．

3. 実　　験

本章では，以下の点に従って，我々の手法が有効であることを示す：

（1） AMSS の類似度としての正当性

（2） ワーピングパス制約の有効性

（3） メタアルゴリズムの有効性

まずはじめに，AMSS が新規の効果的な類似度測定手法として有効であるかを確かめるために，他の類似度測定手法との分類精度を比較する．次に，ワーピングパスの制約を導入したことにより，分類精度が向上していることを示す．最後に，メタアルゴリズムを導入したことにより，分類精度が向上していることを示す．なお，実験では時系列データマイニングの手法を確認するために公開されているデータセットを用いている [8], [9].

3.1 分類実験

本手法を評価するために，一次元時系列の 1-Nearest Neighbor による分類性能を，他手法と比較する．実験には，UCR Time Series Classification/Clustering Page [9] で配布されている，20 個のデータセットを用いている．この実験は時系列の類似度測定の正当性を確かめるのに最も簡単で効果的な方法である．

分類性能の比較として，ユークリッド距離，DTW, LCSS の分類結果を用いる．なお，DTW の分類結果は [9] で公開されているものであり，DTW に Sakoe-Chiba band によるワーピング制約を施した結果となっている [14]．また，LCSS のしきい値は，20 個の異なるデータセットにそれぞれ固定値で設定することが困難なため，2 点 x, y が以下の不等式を満たせば類似性があると設定している [1].

$$y/(1+\epsilon) \leq x \leq y(1+\epsilon) \qquad (9)$$

本実験では $\epsilon = 0.1$ としている．これは，時系列データによらず，2 点間の差異が 10%以内であれば類似することを表している．

表 1 は，時系列データセットにおける誤分類率を示している．データセットに対して最も誤差が少なかった値を太字で表しており，上から AMSS が優位なデータ，DTW が優位なデータ，ユークリッド距離が優位なデータ，LCSS が優位なデータと並べている．

結果から，各類似度測定手法により，優位なデータセットが存在することが分かる．例えば，DTW は，CBF に代表される振動的なデータに強く，AMSS は，Fish に代表される座標値が極めて類似しているデータに強いことが分かる（図 10 参照）．これは，DTW の場合，類似度を座標で算出しているため，軌跡が振動的であろうと大体の位置関係が似ていればマッチングがとれるためである．また，AMSS の場合は角度から類似度を算出しているため，座標値の小さな変化でも類似度としては大きな変化となっているため，このような結果となっている．

また，AMSS は 4 手法の中で最も多い九つのデータセットにおいて，最も優れている．DTW も九つの

表 1 最近傍分類におけるユークリッド距離，DTW, LCSS, AMSS の性能
Table 1 Classification performance of Euclidean Distance, Warping Window DTW, LCSS, AMSS with 1-Nearest Neighbor.

Dataset	Euclidean	DTW	LCSS	AMSS
Gun-Point	0.087	0.087	0.040	**0.000**
Trace	0.240	0.010	0.090	**0.000**
Fish	0.217	0.160	0.114	**0.040**
OSU Leaf	0.483	0.384	0.306	**0.103**
Swedish Leaf	0.213	0.157	0.290	**0.104**
Coffee	0.25	0.179	0.214	**0.143**
Adiac	0.389	0.391	0.675	**0.345**
Beef	0.467	0.467	0.500	**0.433**
50 Words	0.369	**0.242**	0.352	**0.242**
Lighting-2	0.246	**0.131**	0.213	0.180
Face (four)	0.216	**0.114**	0.307	0.261
Face (all)	0.286	**0.192**	0.393	0.275
Lighting-7	0.425	**0.288**	0.438	0.301
CBF	0.148	**0.004**	0.084	0.522
Synthetic Control	0.120	**0.017**	0.273	0.527
Wafer	**0.005**	**0.005**	0.017	0.011
ECG	**0.120**	**0.120**	0.190	0.170
OliveOil	**0.133**	0.167	0.500	0.200
Two Patterns	0.09	0.0015	**0.000**	0.090
Yoga	0.170	0.155	**0.149**	0.158
平均誤分類率	0.234	**0.164**	0.257	0.205
標準偏差	**0.134**	0.135	0.182	0.160

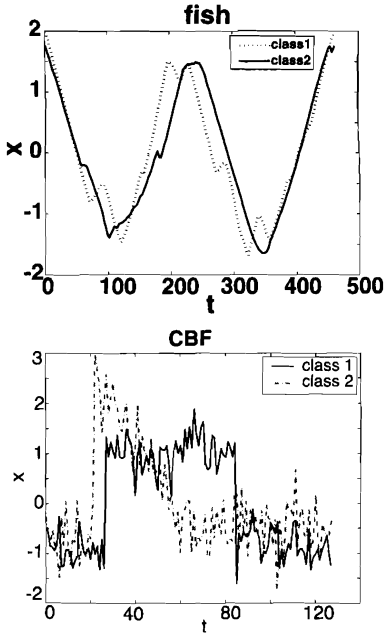

図 10 “Fish” と “CBF” の軌跡
Fig. 10 Shapes of trajectories of data “Fish” and “CBF”.

データセットで優れた結果となっているが，AMSS は Gun-Point と Trace で誤差 0 となっており，その有効性を確認できる．しかし，平均，標準偏差で比較すると，DTW の方が全体として精度が良く，安定的な結果を出している．これは，AMSS は角度の変化により類似度を算出するため，CBF などの振動するような軌跡データでは，繰り返し現れる同じ方向のベクトルの誤対応を起こし，分類性能が極端に悪くなっているためである．以上述べてきたように，すべてのデータに特化した類似度測定手法はなく，時系列の形状によっては，どの類似度測定手法が優位な結果を示すか分からない．

一方，時系列データの形状は，Fish のように比較的滑らかな軌跡であるデータ，CBF のようにデータ全体にノイズが付加されたような変化の激しいデータというように，少なくとも 2 種類が存在している．これらを一つのみの特徴，または一つの類似度測定手法で安定して分類することは不可能である．**3.3** では，多くの特徴を用いて分類精度の向上を図るメタアルゴリズムの導入について議論する．

3.1.1 多次元データセットにおける分類実験

上記の実験では，実験対象のデータは一次元のものが用いられている．本項では，AMSS が多次元のデータセットに対しても，有効に類似性を検索できるかどうかを確かめるための実験を行う．本実験では，表 2 に示される二次元時系列のデータセットを用いている [8]．データセットは 9 種類，150 のデータを含んでいる．分類実験を行うために，データを 3 分割して交差検定を行う．具体的には，ランダムにテストデータ 100 個と訓練データ 50 個に分割し，1-NN にて分類実験を行う．これらのデータは長さが異なるために，ユークリッド距離では類似性を測定できず，DTW，LCSS，AMSS とのみ比較している．

表 3 に着目すると，分類精度に大きく差が開いていることが分かる．

二次元の時系列分類において，軌跡が類似するデータ同士の比較を行う際，図 1 のようにユークリッド空間で同じ動作を異なる向きで行えば，DTW ではそれらが類似していると判断できない．これは，軌跡が類似していても，方向が違えば対応するデータ点間の距離が大きくなるため，類似性が低くみなされるためである．LCSS では，DTW よりも誤差は小さくなっているが，AMSS には及ばない結果となっている．AMSS は，時系列をベクトルの系列とみなすため，系列中の局所的な方向に着目することができる．したがって，時系列全体の方向が異なっていても，系列中でのベクトルの方向が類似していれば，類似度が高くなり，軌跡の似ている時系列同士を類似しているとみなすことができる．

表 2　二次元時系列のデータセット
Table 2　Dataset of 2-dimensional time series.

データ	データ数	データ長
ASL_clean	26	61～94
buoy_sensor	8	215～255
cameraMouse	15	924～1719
flutter	4	255
MarineMammals	4	109～169
mariohWords	52	309～1841
RandomWalk	13	256
robot_arm	4	256
TrackingPoor	23	435～759
all	150	61～1841

表 3　二次元時系列の分類実験結果
Table 3　Classification performance in 2-dimensional time-series dataset.

	DTW	LCSS	AMSS
error rate	0.700	0.219	0.079

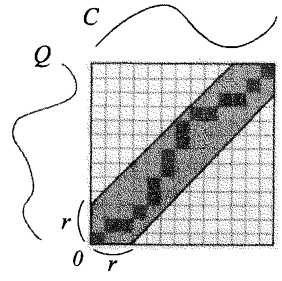

図 11　ワーピングパス制約 'Sakoe-Chiba band'
Fig. 11　A warping path constraint 'Sakoe-Chiba band'.

3.2　ワーピングパス制約

制約を用いた場合と制約なしの場合を比較して，ワーピングパス制約の効果を示す．また，他に制約として Sakoe-Chiba band との比較も行った [10]．図 11 に示されるように，Sakoe-Chiba band によってワーピングパスがとり得る範囲は，行列の対角線と平行な 2 本の直線で囲まれる灰色の領域に制限される．

帯域幅 r は，時系列の長さの 0～100%のもので，訓練データにおける分類精度の良いものを実験的に用いている．帯域幅 0%では，ワーピングパスが類似度行列の対角線と等しくなり，すべてのベクトルが 1 対 1 で対応付けされる．帯域幅 100%のときは，制約を設けない場合と等しくなる．

表 4 ワーピングパス制約の性能
Table 4 Performance of warping path constraints.

Dataset	no constraint	Sakoe	Itakura
50 Words	0.281	0.386	**0.242**
Adiac	0.667	0.483	**0.345**
Beef	0.700	**0.400**	0.433
CBF	0.528	0.654	0.522
Coffee	0.250	0.357	**0.143**
ECG	0.230	0.53	**0.170**
Face (all)	**0.251**	0.691	0.275
Face (four)	0.704	0.409	**0.261**
Fish	0.194	0.434	**0.040**
Gun-Point	0.060	0.053	**0.000**
Lighting-2	0.459	0.459	**0.180**
Lighting-7	0.425	0.863	**0.301**
OliveOil	0.900	0.266	**0.200**
OSU Leaf	0.368	0.458	**0.103**
Swedish Leaf	0.495	0.431	**0.104**
Synthetic Control	**0.527**	0.833	**0.527**
Trace	0.250	0.220	**0.000**
Two Patterns	0.097	0.746	**0.090**
Wafer	0.009	**0.006**	0.011
Yoga	0.160	0.413	**0.158**

表 4 は，それぞれの制約における分類の誤差を示したものである．結果から，板倉らの制約が優れていることが分かる．Sakoe-Chiba band では幅の狭い帯域の場合，データ整合が正しく行われず，間違った対応付けがなされることがある．一方，幅の広い帯域の場合は，前述の一つのベクトルに多くのベクトルが対応される問題が生じることがある．また，Coffee の例を見ると，Sakoe-Chiba band よりも制約なしの方が誤差が少ない．これは訓練例で有効だった帯域幅が，テストデータには有効でなかったと考えられる．このように，Sakoe-Chiba band よりも，板倉らの制約がより的確で，誤差が少なく分類できることが分かる．

3.3 メタアルゴリズム

3.1 の実験結果のように，AMSS はノイズが付加されたような激しい変化を伴うデータに対して，他の手法と比べて精度が悪くなっている．そのため，メジアン値，移動平均値を用いた平滑化により激しい変化を軽減させることを考える．なお，メジアン値は，時系列データ中の各項について，その項を中心とした前後数項の中央値であり，移動平均値は，その前後数項との平均値である．

CBF に対して平滑化を行った例を図 12 に示す．メジアンフィルタ（median）では，特徴的な極値が残って，全体的に時系列の中央値に寄って縮約されている．移動平均法（moving average）では，振動のような変化が取り除かれ，クラスごとの特徴を示すような時系列全体の概形を描く軌跡となっている．これらの特徴値により，連続的なノイズによる誤対応がなく，時系列データを比較することができる．

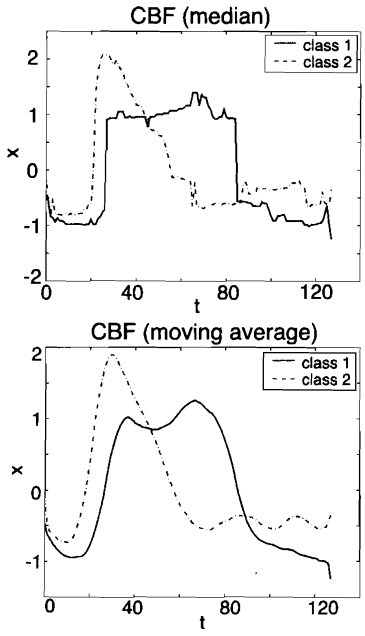

図 12 “CBF” での平滑化の例
Fig. 12 Examples of smoothing on data “CBF”.

一方，平滑化により精度が下がるデータセットも存在する．例えば，Gun-Point はノイズを含まず，逆に平滑化によってクラス間の差異が縮められてしまい，正確に分類できていたデータが誤分類される可能性がある．したがって，一概に平滑化を採用することはできず，どのデータセットにどの平滑化が有効かを決定することもできない．そのため，もとの時系列データの値，メジアン値，移動平均値をそれぞれ特徴値として用いるメタアルゴリズムを導入する．

メタアルゴリズムを用いた場合とメタアルゴリズムなしの場合を比較して，メタアルゴリズムの効果を示す．今回の実験では，メジアン値，移動平均値ともに，前後 5 項を評価するように実験的に決定した．また，移動平均法では，平滑化した結果に，もう一度移動平均法で平滑化を行い，より変動の少ないデータを得られるようにしている．

表 5 はメタアルゴリズムを用いた場合の AMSS と AMSS 単体の分類の誤差を示したものである．結果として，15 データセットで分類誤差が減少していることが分かる．CBF では，移動平均法を適用したデー

表 5 Boosting アプローチによる実験結果
Table 5 Experimental results of AMSS with Boosting approach.

Dataset	AMSS	Meta-AMSS
50 Words	0.242	**0.208**
Adiac	0.345	**0.304**
Beef	0.433	**0.400**
CBF	0.522	**0.061**
Coffee	0.143	**0.000**
ECG200	0.170	**0.140**
Face (all)	0.275	**0.184**
Face (four)	0.261	**0.227**
Fish	0.040	**0.035**
Gun-Point	**0.000**	0.006
Lightning-2	0.180	**0.157**
Lightning-7	0.301	**0.283**
OliveOil	0.200	**0.166**
OSULeaf	**0.103**	0.140
Swedish Leaf	**0.104**	0.106
Synthetic Control	0.527	**0.116**
Trace	0.000	0.000
Two Patten	0.090	**0.006**
Wafer	0.011	0.011
Yoga	0.158	**0.146**

タの分類が効果的に採用され，大幅に誤差が減少し，DTW に追従する結果が得られている．Swedish Leaf や OSULeaf など，誤差が減少しなかったデータセットもあるが，その誤差の変化は 0.2〜4%と小さく，今後，パラメータ設定の修正で改善される範囲となっていることが分かる．

4. むすび

本論文では，時系列データの新規の類似度測定手法である AMSS を提案した．AMSS は類似度を適切に計測でき，ノイズや座標系の影響を受けないという特徴がある．分類実験では，AMSS が他の手法よりも有効な類似性検索手法であることを示した．また，従来のワーピングパス制約が AMSS にも有効であることを示した．最終的に，AMSS がユークリッド距離や DTW に対抗し得ると結論づけた．

また，AMSS は変化の激しいデータでは，繰り返し同じ方向のベクトルが現れ，部分系列を誤って対応させてしまうという問題が生じたが，メタアルゴリズムによりデータの特徴をうまく選択することにより解決した．今後，更に扱う特徴を増やすことにより，総合的に高精度な分類が可能な手法が実現できると考えられる．

しかしながら，メタアルゴリズムを行うことにより，計算コストはもとの AMSS に比べて膨大となり，ワーピング制約を用いた効率化のみでは問題が残る．このため，更に効率的な手法として，上界関数を導入したフィルタリングにより，計算コストを削減する手法を考えている．上界関数とは，本来の関数より必ず大きな値となる近似値を求める関数である．具体的には，ベクトル系列を合成ベクトル系列に変換して，上界関数を用いて近似値を求め，類似性の低いデータを前もってフィルタリングしている．この効率化については別稿で議論する．

文　献

[1] G. Das, D. Gunopulos, and H. Mannila, "Finding similar time series," Proc. First European Symposium on Principles of Data Mining and Knowledge Discovery, pp.88–100, 1997.

[2] S.-W. Kim, S. Park, and W.W. Chu, "An index-based approach for similarity search supporting time warping in large sequence databases," Proc. 17th International Conference on Data Engineering, pp.607–614, 2001.

[3] "Workshop and challenge on time series classification," 2007. http://www.cs.ucr.edu/~eamonn/SIGKDD2007TimeSeries.html

[4] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," Proc. 1994 ACM SIGMOD International Conference on Management of Data, pp.419–429, 1994.

[5] D.J. Berndt and J. Clifford, "Finding patterns in time series: A dynamic programming approach," in Advances in Knowledge Discovery and Data Mining, ed. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, pp.229–248, AAAI, 1996.

[6] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing multi-dimensional time-series with support for multiple distance measures," Proc. 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.216–225, 2003.

[7] F. Itakura, "Minimum prediction residual principle applied to speech recognition," IEEE Trans. Acoust. Speech Signal Process., vol.23, pp.67–72, 1975.

[8] E. Keogh, "The UCR Time Series Data Mining Archive," 2006. http://www.cs.ucr.edu/~eamonn/TSDMA/index.html

[9] E. Keogh, X. Xi, L. Wei, and C.A. Ratanamahatana, "The UCR time series classification/clustering homepage," 2006. http://www.cs.ucr.edu/~eamonn/time_series_data/

[10] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," IEEE Trans. Acoust. Speech Signal Process., vol.26, pp.46–49, 1978.

[11] H. Nomiya and K. Uehara, "Multistrategical im-

age classification for image data mining," Proc. International Workshop on Multimedia Data Mining (MDM/KDD 2007), pp.22–30, 2007.

[12] A. Borisov, K. Torkkola, and E. Tuv, "Best subset feature selection for massive mixed-type problems," Proc. 7th International Conference on Intelligent Data Engineering and Automated Learning, pp.1048–1056, 2006.

[13] S. Park, W. Chu, J. Yoon, and C. Hsu, "Efficient search for similar subsequences of different length in aequence database," Proc. 16th International Conference on Data Engineering, pp.22–32, 2000.

[14] C.-A. Ratanamahatana and E. Keogh, "Making time-series classification more accurate using learned constraints," Proc. SIAM Internatioal Conference on Data Mining, pp.11–22, 2004.

[15] Y. Kim, D. Kim, S. Chung, and P.-K. Chong, "Target classification in sparse sampling acoustic sensor networks using DTWC algorithm," Proc. International Conference on Intelligent Pervasive Computing, pp.236–241, 2007.

（平成 20 年 1 月 21 日受付，5 月 7 日再受付）

中村　哲也

平 18 神戸大・工・情報知能工学卒．現在，同大大学院工学研究科博士前期課程在学中．

瀧　　敬士

平 19 神戸大・工・情報知能工学卒．現在，同大大学院工学研究科博士前期課程在学中．

野宮　浩揮

平 16 神戸大学大学院自然科学研究科情報知能工学専攻修士課程了．現在，同大学院情報・電子科学専攻博士課程に在学中．機械学習，特に視覚的学習の研究に従事．

上原　邦昭　（正員）

昭 53 阪大・基礎工・情報工学卒．昭 58 同大大学院博士後期課程単位取得退学．同産業科学研究所助手，講師，神戸大学工学部情報知能工学科助教授，同都市安全研究センター教授を経て，現在，同大学院工学研究科教授．工博．人工知能，特に機械学習，マルチメディア処理の研究に従事．人工知能学会，計量国語学会，日本ソフトウェア科学会，AAAI 各会員．