DEIM2022 D34-2(day3 p21)

Shapeletsを学習する時系列分類手法の局所的最適解向上

山口　晃広†　　植野　　研†

† 株式会社東芝 研究開発センター システム AI ラボラトリー　〒 212-8582 川崎市幸区小向東芝町 1
E-mail: †akihiro5.yamaguchi@toshiba.co.jp

あらまし　機械学習による時系列インスタンスのクラス分類問題において，分類に用いる波形パターン（shapelets）を学習する手法が，分類性能が高く説明性もあるため近年注目を集めている．これらの手法は非凸最適化問題として定式化され勾配降下法により局所最適解を見つける．そのため，shapelets の初期値や時系列インスタンスの学習順序を適切に選ぶことが重要であり，本研究ではこれらの課題に向けて shapelets 学習法を改良する．1 つ目の shapelets の初期値問題について，本研究では教師有り特徴選択の枠組みを導入し，時系列データを構成する大量の部分時系列から分類損失を削減するものを初期 shapelets として選ぶ．その後，その分類損失を更に削減するように学習データの部分時系列に限定されない shapelets を学習する．2 つ目の時系列インスタンスの学習順序については，カリキュラム学習の一種である self-paced learning を shapelets 学習法に導入し，shapelets と分類器に加えて学習データの信頼度を同時に学習し，高信頼のインスタンスから優先的に学習する．UCR 時系列データセットを用いた実験では，shapelets の個数が少ない場合の shapelets の説明性と AUC の性能向上における有効性を示す．

キーワード　時系列分類, Shapelet, 特徴選択, SPL

1　はじめに

IoT の活用に向けて，機械学習による時系列データのクラス分類手法の研究が進められている．2 クラス分類の場合は，学習時に 1 または 0 のクラスラベルが付与された複数の時系列インスタンスを与えて，テスト時に未知の時系列インスタンスのクラスラベルを予測する問題となる．一般のクラス分類とは異なり時系列分類では，時間軸における順序関係や波形パターンの形状が重要であり，時系列データのノイズや波形パターンの出現位置のズレなどを扱える時系列分類に特化した手法がこれまでに提案されている [1].

時系列分類手法の中で，時系列分類に有効な複数の波形パターン（shapelets）を発見することで分類器を学習する研究に注目が集まっている [2, 3]．これらの shapelets 手法では，分類に有効な特徴は長い時系列全体ではなく少数の短い部分時系列に表れるというアイデアに基づく．特長として，学習が終わればテスト時の分類は高速であり，高い分類性能を達成する．また，分類に用いる波形パターンを専門家に提示できるため説明性があり，医療・製造・保守分野などの専門家に受け入れられやすい．特に少数の shapelets を発見できれば，専門家は学習モデルに用いられる特徴を全て理解し機械学習の結果を専門知識と照らし合わせて納得して利用できる．

Shapelets 手法は，学習データから切り出される多数の部分時系列を探索することで shapelets を発見するアプローチから始まったが [3]，近年 shapelets を任意の波形パターンとして学習する shapelets 学習法が提案されている [2]．探索ベースの手法とは異なり，shapelets 学習法では確率的勾配降下法（stochastic gradient descent; SGD）により shapelets と分類器の両方を同時に学習する．そのため，目標とする分類損失を削減するように，学習データの部分時系列に制限されない shapelets を生成することができる．これまで，F 値や partial Area Under the Curve (AUC) の改善を目標とした 2 クラス分類損失関数を導入する手法も提案されている [4, 5].

しかしながら，これらの shapelets 学習法は非凸最適化問題として定式化され [2, 4, 5, 6]，SGD による解法では局所的最適解しか求まらない．そのため，shapelets の初期値を適切に選ぶことが 1 つ目の重要な課題となる．この初期 shapelets の選択は分類性能に大きな影響を与えることが知られているが [2]，従来手法 [2, 4, 5, 6] では部分時系列に教師無しクラスタリングを適用しそのセントロイドを初期 shapelets としておりクラスラベルを活用できていなかった．2 つ目の課題としては，時系列インスタンスの学習順序を適切に選んで勾配降下法を適用することである．従来手法 [2, 4, 5, 6] では SGD の学習過程でランダムに時系列インスタンスを選んでいたため，信頼性の低いインスタンスの悪影響を受けやすい課題があった．

1 つ目の課題については，大量の部分時系列から分類損失を削減するものを shapelets として選ぶことが考えられ，このアプローチは教師有り特徴選択とみなせる．高次元データからの特徴選択は，データマイニングや機械学習の分野で広く研究されており，大きく教師の有無で区分できる．教師有り特徴選択ではクラスラベルを活用し損失関数を削減するように最適化できるため，教師無し特徴選択よりも一般に高い分類性能を達成する．しかしながら，従来の shapelets 学習法 [2, 4, 5, 6] では教師無しクラスタリングを用いて大量の部分時系列から初期 shapelets を得るため分類失関数を考慮できていない．

2 つ目の課題については，非凸最適化問題の解法に SGD の代わりに self-paced learning (SPL) という学習アルゴリズムを用いることが考えられる．人間や動物の学習過程のように，SPL では最初は簡単な事例を優先的に学習し徐々に紛らわしく

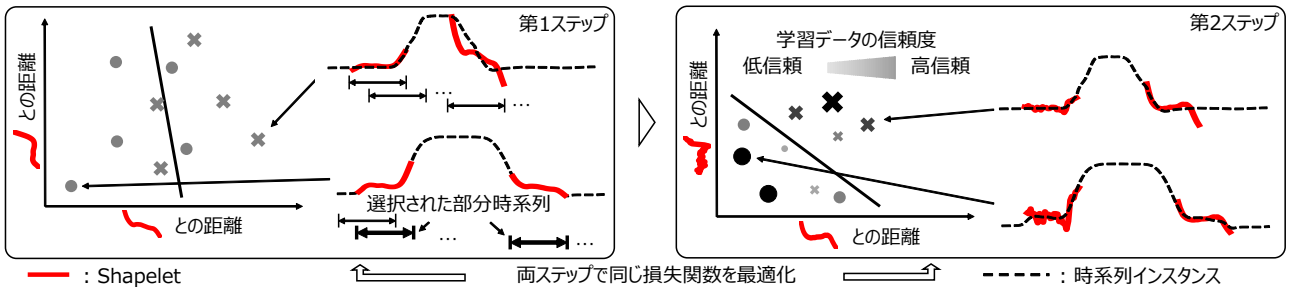

図 1　LTSSFS の概要: 第 1 ステップで分類損失を考慮して部分時系列から初期 shapelets を選択し，第 2 ステップで shapelets と分類器と信頼度を SPL で同時に学習する．

難しい事例を含めて学習を進める．これにより，学習初期における低信頼の学習データの悪影響を避けて最終的な局所的最適解が改善することが知られている [7]．このアプローチはカリキュラム学習 [8] と類似するが，SPL では学習過程に応じて学習データの信頼度を自動的に獲得する．

そこで，本研究ではこれらの課題を解決する shapelets 学習法 “LTSSFS” を提案する．図 1 に示すように，LTSSFS では 2 ステップで共通の損失関数を削減するようにそれぞれ定式化される．第 1 ステップでは，教師有り特徴選択を shapelets の初期値問題に導入し，大量の部分時系列から分類損失を削減するものを初期 shapelets として発見する．第 2 ステップでは，SPL に基づき学習データの信頼度を学習しながら，第 1 ステップで得られた初期 shapelets から更に分類損失を削減するように shapelets を調整する．

1.1　本研究の主な貢献

本研究の主な貢献を以下に示す．なお，本内容は基本的に [9] に準じ，証明や計算量や追加実験などに関しては，第 1 ステップは [9] を第 2 ステップは [10] をそれぞれ参照いただきたい．

- 学習データの部分時系列からの初期 shapelets を選定する段階と，それら初期 shapelets から任意の形状の shapelets を学習する段階との 2 ステップで共通の損失関数を最適化する shapelets 学習法を提案する．
- Shapelets の初期値を改善するため，学習データに含まれる大量の部分時系列から損失関数を削減するように初期 shapelets を選定する問題を教師無し特徴選択として定式化し計算資源を考慮した解法も提案する．
- 低信頼な学習データの悪影響を避けるため，SPL を shapelets 学習法に導入し，shapelets と分類器だけでなく学習データの信頼度も同時に学習する定式化とその解法とを提案する．
- UCR 時系列データセットを用いて，shapelets が少数の場合の説明性と AUC の性能向上における有効性を示す．

2　準　　備

本研究では時系列インスタンスを 2 クラス $\mathcal{Y}:=\{1,0\}$ に分類する問題を扱う．学習データのインスタンスの数を I 個として，i 番目の真のクラスラベルを $\mathbf{y}_i\in\mathcal{Y}$ とする．図 1 に示すように，長い数列として表される時系列インスタンスに対して shapelet は短い数列として表される．長さ Q の時系列データセットを $\mathbf{T}\in\mathbb{R}^{I\times Q}$ とし，K 個の長さ L の shapelets を $\mathbf{S}\in\mathbb{R}^{K\times L}$ とする．i 番目の時系列インスタンス $\mathbf{T}_i$ の j 番目の値を $\mathbf{T}_{i,j}$ と記述し，k 番目の shapelet $\mathbf{S}_k$ の l 番目の値を $\mathbf{S}_{k,l}$ と記述する．各時系列インスタンスに対して長さ L の部分時系列は $J:=Q-L+1$ 個ある．j 番目の部分時系列 $(\mathbf{T}_{i,j},\mathbf{T}_{i,j+1},\cdots,\mathbf{T}_{i,j+L-1})$ を $\mathbf{T}_{i,j:j+L-1}$ と記述する．

2.1　Shapelets 学習法

本節では，提案手法の一部の基となっている文献 [2] の定式化を説明する．j 番目の部分時系列 $\mathbf{T}_{i,j:j+L-1}$ と $\mathbf{S}_k$ とのユークリッド距離を測り，$j=1,2,\cdots,J$ の中で最小のユークリッド距離を $\mathbf{T}_i$ と $\mathbf{S}_k$ との距離として以下で定義する．

$$\mathbf{X}_{i,k}=\min_{j=1,2,\cdots,J}\frac{1}{L}\sum_{l=1}^{L}(\mathbf{T}_{i,j+l-1}-\mathbf{S}_{k,l})^2. \qquad (1)$$

i 番目のインスタンスに対する特徴ベクトルを $\mathbf{X}_i:=(\mathbf{X}_{i,1},\mathbf{X}_{i,2},\cdots,\mathbf{X}_{i,K})$ で定義する．バイアス項を含めた分類器の重み $\mathbf{w}\in\mathbb{R}^{K+1}$ が与えられると，次の線形モデルで $\hat{\mathbf{y}}_i$ を予測する．

$$\hat{\mathbf{y}}_i=\sum_{k=1}^{K}\mathbf{w}_k\mathbf{X}_{i,k}+\mathbf{w}_0. \qquad (2)$$

最終的には，分類器の重み $\mathbf{w}$ に加えて特徴ベクトル $\mathbf{X}$ を経由して shapelets $\mathbf{S}$ を同時に学習する非凸最適化問題として次のように定式化される．

$$\underset{\mathbf{S}\in\mathbb{R}^{K\times L},\,\mathbf{w}\in\mathbb{R}^{K+1}}{\text{minimize}}\quad\sum_{i=1}^{I}\mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)+\alpha\sum_{k=1}^{K}\mathbf{w}_k^2, \qquad (3)$$

ここで，$\alpha\geqq 0$ は ℓ_2 正則化パラメータである．損失関数 $\mathcal{L}$ としては，次のようなロジスティック損失関数を用いる．

$$\mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)=-\mathbf{y}_i\ln\sigma(\hat{\mathbf{y}}_i)-(1-\mathbf{y}_i)\ln(1-\sigma(\hat{\mathbf{y}}_i)), \qquad (4)$$

ここで，シグモイド関数 $\sigma(\hat{\mathbf{y}}_i)$ は $\left(1+e^{-\hat{\mathbf{y}}_i}\right)^{-1}$ で定義される．

Shapelets 学習法 [2] は，クラスラベルを用いない時系列クラスタリングへ適応する研究 [6] や，コスト考慮型学習や pAUC 最大化への対応など 2 クラス分類損失関数を改良する研究などが行われている [4, 5]．これらの従来手法 [2, 4, 5, 6] も非凸最適化問題として定式化され SGD を用いて shapelets を学習する．そのため，shapelets の初期値や時系列インスタンスの学習順

序を適切に選ぶ必要がある．しかしながら，これらの従来手法では shapelets の初期値を教師無しクラスタリングで選ぶため，良質の初期 shapelets が選ばれにくい問題がある．これに当てはまらない従来手法として，時系列データに出現する shapelets の位置を学習する手法では shapelets の時間軸のズレに直接対応できず [11, 12]，学習時だけでなくテスト時にも shapelets を生成する手法では多数の shapelets が必要となり [13, 14]，TF-IDF に基づき初期 shapelets を選定する手法では目標とする分類損失関数を考慮できない [15]．また，[11, 12] を除いたいずれの手法も shapelets の学習に SGD を使用しているため低信頼な学習データの悪影響を受けやすいという問題もある．

2.2 Self-paced learning

人間や動物の学習過程のように，SPL では良い局所最適解を得るため分類し易いインスタンスから優先的に学習を始めて徐々に難しいインスタンスを含めて学習を進める [7]．カリキュラム学習とは異なり [8]，SPL では学習インスタンスの信頼度 $\mathbf{v}\in[0,1]^I$ を数理最適化問題の定式化に取り込むことで動的かつ自動的に信頼度 $\mathbf{v}$ を学習する．その定式化は次のように分類器の重み $\mathbf{w}$ と信頼度 $\mathbf{v}$ とを同時に最適化する．

$$\underset{\mathbf{w}\in\mathbb{R}^{K+1},\mathbf{v}\in[0,1]^I}{\text{minimize}} \quad \sum_{i=1}^{I}\mathbf{v}_i\mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)+\lambda_{\mathbf{y}_i}f(\mathbf{v}_i), \tag{5}$$

ここで，$\lambda_{\mathbf{y}_i}>0$ を SPL パラメータと呼び，関数 f を SPL 関数と呼ぶ．SPL 関数 f には狭義単調減少関数 [1] を選ぶことで，$\lambda_{\mathbf{y}_i}$ が増加すると信頼度 $\mathbf{v}_i$ の最適値も上昇するので，より難しいインスタンスが学習過程に含まれるようになる．そのため，学習過程に応じて SPL パラメータ $\lambda_{\mathbf{y}_i}$ を徐々に増加させる [7]．

従来研究では様々な SPL 関数が提案されている．$f(\mathbf{v}_i)=-\mathbf{v}_i$ と定義した場合が最も単純で，信頼度 $\mathbf{v}_i$ の最適値は i 番目のインスタンスを使用するか $(\mathbf{v}_i=1)$ 否か $(\mathbf{v}_i=0)$ の 2 値になる [7]．その後，信頼度の最適値が [0,1] 区間の連続値をとれるような SPL 関数が提案され [16, 17]，学習過程に応じて動的に変化する SPL 関数も提案された [18]．SPL の適用事例としても様々な研究があり，行列分解 [17] やマルチメディア検索 [16] や畳み込みニューラルネットワーク [18] などで有効性が示されている．しかしながら，時系列分類に SPL を導入した研究事例は我々の知る限り無い．

2.3 特徴選択

特徴選択は教師有り，教師無し，半教師有りに区分できる [19]．教師有り特徴選択はラッパー，フィルタ，埋め込みに更に区分できる．ラッパーや埋め込みに区分される教師有り特徴選択では，クラスラベルだけでなく最適化する目的関数を直接考慮できるため，一般に高い分類性能を達成する傾向がある．しかしながら，従来の shapelets 学習法では大量の部分時系列に教師無しクラスタリングを適用することでそのセントロイドを初期 shapelets としていた．このような従来のアプローチは教師無し特徴選択とみなせるため分類性能を劣化させると考えられる．

LASSO のように ℓ_1 正則化を用いた埋め込み法は広く研究されてはいるが，ユーザが所望する数の特徴を選択するには正則化パラメータのチューニングが難しい．一方，所望する数の特徴を直接得るには ℓ_0 制約条件付き最適化問題として定式化すればよい．このような問題は一般に NP 困難となるが，限られた計算資源のもとで全ての特徴に一度にアクセスすることなく高次元の特徴を選択できる効率的な解法も提案されており，ロジスティック損失関数を含む分類損失を扱える特徴選択手法も提案されている [20]．しかしながら，本研究において有効な特徴を選択することは実用的には難しく，教師有りラッパーの Recursive Feature Elimination (RFE) [21] が，目的関数に従った有効な特徴を選択できることを確認した．そこで，本研究では RFE に基づく実用的なリソース管理手法を提案する．

3 教師有り特徴選択による初期 shapelets の発見

本章では提案手法 LTSSFS の第 1 ステップを述べる．第 2 ステップと全体のアルゴリズムについては，4 章で後述する．

3.1 第 1 ステップにおける数理最適化問題の定式化

大量の部分時系列から分類損失関数を最小化する K 個の初期 shapelets を選ぶように数理最適化問題を定式化する．部分時系列のインデックス集合を $\mathcal{S}$ とし，ペア $(\tilde{i},\tilde{j})\in\mathcal{S}$ は部分時系列 $\mathbf{T}_{\tilde{i},\tilde{j}:\tilde{j}+L-1}$ に対応する．次段落で後述するように，$\mathcal{S}$ には学習データに含まれる長さ L の全ての部分時系列のうち冗長なものを除外して保持するが，$\mathcal{S}$ の数 $(|\mathcal{S}|)$ は依然多い．高次元ベクトルとして表される分類器の重み $\tilde{\mathbf{w}}\in\mathbb{R}^{|\mathcal{S}|+1}$ を用いて，本数理最適化問題では次のように分類損失を最小化する．

$$\begin{aligned}
&\underset{\tilde{\mathbf{w}}\in\mathbb{R}^{|\mathcal{S}|+1}}{\text{minimize}}\, F, \quad F=\sum_{i=1}^{I}\mathcal{L}(\mathbf{y}_i,\tilde{\mathbf{y}}_i)+\alpha\sum_{(\tilde{i},\tilde{j})\in\mathcal{S}}\tilde{\mathbf{w}}_{(\tilde{i},\tilde{j})}^2,\\
&\tilde{\mathbf{y}}_i=\sum_{(\tilde{i},\tilde{j})\in\mathcal{S}}\tilde{\mathbf{w}}_{(\tilde{i},\tilde{j})}\tilde{\mathbf{X}}_{i,(\tilde{i},\tilde{j})}+\tilde{\mathbf{w}}_0, \qquad (6)\\
&\tilde{\mathbf{X}}_{i,(\tilde{i},\tilde{j})}=\min_{j=1,2,\cdots,J}\sum_{l=1}^{L}\frac{(\mathbf{T}_{i,j+l-1}-\mathbf{T}_{\tilde{i},\tilde{j}+l-1})^2}{L},
\end{aligned}$$

ここで，損失関数 $\mathcal{L}(\mathbf{y}_i,\cdot)$ は式 (4) と同じである．損失関数 $\mathcal{L}(\mathbf{y}_i,\cdot)$ は凸なので，制約式が無ければ式 (6) は凸最適化問題である．多数 $|\mathcal{S}|$ 個の部分時系列から少数 K 個の初期 shapelets を選ぶため，以下の ℓ_0 制約式を追加する．

$$\sum_{(\tilde{i},\tilde{j})\in\mathcal{S}}\mathbb{I}\Big(\tilde{\mathbf{w}}_{(\tilde{i},\tilde{j})}\neq 0\Big)\leqq K, \tag{7}$$

ここで，インジケータ関数 $\mathbb{I}(\cdot)$ は引数が真のとき 1 を返し偽のとき 0 を返す．分類器の重みが $\tilde{\mathbf{w}}_{(\tilde{i},\tilde{j})}=0$ であれば，全ての $i=1,2,\cdots,I$ に対して，特徴量 $\tilde{\mathbf{X}}_{i,(\tilde{i},\tilde{j})}$ は $\tilde{\mathbf{y}}_i$ に影響を与えないため損失 $\mathcal{L}(\mathbf{y}_i,\tilde{\mathbf{y}}_i)$ にも影響を与えない．そのため，制約式 (7) を用いて式 (6) の目的関数に応じた K 個の部分時系列を初期 shapelets として選定できる．

従来手法 [22, 23] のように，同一の時系列データセットから取り出される部分時系列の大部分はお互いに類似していることに着目し，ランダムに取り出した部分時系列の集合から類似していない部分時系列のみをインデックス集合 $\mathcal{S}$ に保持する．こ

1：つまり $0\leqq v<\tilde{v}\leqq 1$ であれば $f(v)>f(\tilde{v})$ が成り立つ関数 f を選ぶ．

のヒューリスティックな方法は，冗長のない部分時系列に絞ることで計算を効率化できるが，類似したより良い部分時系列が後から取り出された場合に見逃す可能性がある．しかしながら，LTSSFS には第 2 ステップがありこの初期 shapelets を更に調整するため，従来手法 [22, 23] よりも問題とならない．

3.2 第 1 ステップにおける数理最適化問題の解法

制約式 (7) を持った数理最適化問題 (6) は，高次元で非凸であり NP 困難な問題である．そのため，解法には RFE を適用し制約式 (7) を満たすまで以下の手続きを繰り返す．

(1) 制約式 (7) を無くして式 (6) を解く．

(2) 分類器の重み $\tilde{\mathbf{w}}_{(\tilde{i},\tilde{j})}$ の絶対値が小さいものを 0 に固定することで，その $(\tilde{i},\tilde{j})$ 番目の特徴を取り除く．

この手続きは，重みの絶対値 $|\tilde{\mathbf{w}}_{(\tilde{i},\tilde{j})}|$ が大きい次元 $(\tilde{i},\tilde{j})$ は，$\tilde{\mathbf{y}}$ に大きく影響するため，損失関数 $\mathcal{L}$ にも大きく影響するというアイデアに基づく．制約式 (7) が無い式 (6) は標準的な ℓ_2 正則化付きロジスティック損失関数の最小化問題と等しいため，[24] などの最適化ソルバを用いて効率的に計算できる．

3.1 節で述べたように，冗長のない部分時系列のみに $\mathcal{S}$ を制限したとしても，計算資源やデータサイズによっては式 (6) の全次元を同時に扱うことは難しい．そのため，LTSSFS では部分時系列のインデックス集合 $\mathcal{S}$ を更新しながら同時に重要ではない特徴を破棄することで，$\mathcal{S}$ に含まれる部分時系列の数を一定以下に抑える．$|\mathcal{S}|$ が最大数 τ に達したとき，式 (6) が制約式 (7) の代わりに次の制約式を満たすように RFE の手続きを適用し重要でない特徴を除外する．

$$\sum_{(\tilde{i},\tilde{j})\in\mathcal{S}} \mathbb{I}\Big(\tilde{\mathbf{w}}_{(\tilde{i},\tilde{j})} \neq 0\Big) \leqq \frac{\tau}{2}. \tag{8}$$

つまり，リソース容量が一杯になったときに次元数を半分に減らす．$|\mathcal{S}|$ の最大数 τ は，利用する計算資源によって決まり，本実験では $\tau = 10000$ を用いた．

本解法の疑似コードを Algorithm 1 に示す．冗長ではない部分時系列を判定するため，冗長と判定した部分時系列のインデックス集合についても $\mathcal{R}$ に保持している．従来手法 [22] のように，部分時系列がお互いに類似しているか否かは，部分時系列のランダムなペアに対してユークリッド距離 [2] を計算し，その分布の 25%タイルを判定閾値として 1 行目で決定される．

Algorithm 1 教師有り特徴選択に基づく初期 shapelets の選定

Input: 時系列インスタンス: $\mathbf{T}\in\mathbb{R}^{I\times Q}$; クラスラベル: $\mathcal{Y}^I$; Shapelet の長さ: L; Shapelets の数: K; ℓ_2 正則化パラメータ: α

Output: 初期 shapelets: $\mathbf{S}\in\mathbb{R}^{K\times L}$

```
Determine similarity threshold.
S ← ∅ and R ← ∅.
for n = 1,2,···,IQL do
    Choose segment T_{ĩ,j̃:j̃+L−1} uniformly at random.
    if T_{ĩ,j̃:j̃+L−1} is not similar to any previously stored time-series segments in S∪R. then
        Calculate X̃_{i,(ĩ,j̃)} and ỹ_i in Eq. (6) for i = 1,2,···,I.
        S ← S∪{(ĩ,j̃)}.
        if |S| ≧ τ then
            Select eliminated time-series segment indexes E by RFE so that Eq. (6) satisfies constraint (8).
            S ← S\E and R ← R∪E.
        end if
    end if
end for
Select eliminated time-series segment indexes E by RFE so that Eq. (6) satisfies constraint (7).
k = 1.
for each element (ĩ,j̃) ∈ S\E do
    S_k ← T_{ĩ,j̃:j̃+L−1} and k ← k+1.
end for
return S
```

4 SPL を用いた最終的な shapelets の学習

3 章で述べた第 1 ステップの最適化から求まる初期 shapelets は，学習データの部分時系列に制限されていた．本章で述べる第 2 ステップでは，第 1 ステップと同じ損失関数を更に削減するように初期 shapelets を調整し，学習データの部分時系列に限定されない波形パターンを最終的な shapelets として SPL を用いて学習する．

4.1 第 2 ステップにおける数理最適化問題の定式化

第 2 ステップでは，分類損失を最小化するように shapelets $\mathbf{S}$ と分類器の重み $\mathbf{w}$ と SPL における学習データの信頼度 $\mathbf{v}$ とをそれらの相互依存を考慮して同時に学習する．この数理最適化問題は，SPL の式 (5) を shapelets 学習法の式 (3) に組み込み自然に定式化される．

$$\begin{aligned} &\underset{\mathbf{S}\in\mathbb{R}^{K\times L},\mathbf{w}\in\mathbb{R}^{K+1},\mathbf{v}\in[0,1]^I}{\text{minimize}} \quad \sum_{i=1}^{I} G_i, \\ &G_i = \mathbf{v}_i\mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i) + \lambda_{\mathbf{y}_i} f(\mathbf{v}_i) + \frac{\alpha}{I}\sum_{k=1}^{K}\mathbf{w}_k^2, \end{aligned} \tag{9}$$

ここで，ロジスティック損失関数 $\mathcal{L}$ は式 (1) と (2) を介して式 (4) から計算される．また，ℓ_2 正則化パラメータ $\alpha \geqq 0$ と SPL パラメータ $\lambda_{\mathbf{y}_i} > 0$ は，式 (3) と式 (5) のものとそれぞれ同じである．SPL 関数 $f(\cdot)$ は次式で定義される．

$$f(\mathbf{v}_i) = \frac{q}{1+q}\mathbf{v}_i^{\frac{1}{q}+1} - \mathbf{v}_i, \tag{10}$$

ここで，$q > 0$ でありそのとき f は狭義単調減少関数となることが示せる [10]．

4.2 第 2 ステップにおける数理最適化問題の解法

式 (9) の解法は交互最適化アルゴリズムに基づく．最適化する変数を 2 つのブロックに分割し，片方のブロックに含まれる変数を固定してもう片方のブロックに含まれる変数を最適化し，このプロセスを交互に繰り返すことで全変数を学習する．片方のブロックには学習データの信頼度 $\mathbf{v}$ が含まれ，残りのブロックには shapelets $\mathbf{S}$ と分類器の重み $\mathbf{w}$ が含まれる．以降では，これらのブロックの学習方法について述べる．

2：各部分時系列に対して z 正規化（平均 0 標準偏差 1）を事前に適用する．

4.2.1 学習データの信頼度 $\mathbf{v}$ の学習

ここでは，shapelets $\mathbf{S}$ と分類器の重み $\mathbf{w}$ とを固定して，学習データの信頼度 $\mathbf{v}$ を最適化する．目的関数は $\sum_{i=1}^{I} G_i$ として分解できるため G_i を個別に最適化すれば十分である．このとき i 番目のインスタンスの信頼度 $\mathbf{v}_i$ の最適値は式 (11) で求まる．

定理 1. 変数 $\mathbf{S}$ と $\mathbf{w}$ を固定したとき，$i=1,2,\cdots,I$ に対して式 (9) の最適値 $\mathbf{v}_i$ は次の閉形式で求まる．

$$\mathbf{v}_i = \begin{cases} \left(1-\frac{\mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)}{\lambda_{\mathbf{y}_i}}\right)^q & \text{if } \mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i) \leqq \lambda_{\mathbf{y}_i}, \\ 0 & \text{if } \mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i) > \lambda_{\mathbf{y}_i}. \end{cases} \tag{11}$$

4.2.2 Shapelets $\mathbf{S}$ と分類器の重み $\mathbf{w}$ の学習

学習データの信頼度 $\mathbf{v}$ を固定して shapelets $\mathbf{S}$ と分類器の重み $\mathbf{w}$ を最適化する．この最適化問題は非凸のままであるが，目的関数は $\sum_{i=1}^{I} G_i$ と分解できるため SGD アルゴリズムを用いて解く．Shapelets $\mathbf{S}$ や分類器の重み $\mathbf{w}$ に対して，インスタンスごとに分解した目的関数 G_i の勾配は，i 番目のインスタンスの信頼度 $\mathbf{v}_i$ が定数として含まれることを除いて従来の shapelets 学習法 [2, 4, 5, 6] と同様であり次の式で求められる．

$$\begin{aligned} \frac{\partial G_i}{\partial \mathbf{S}_{k,l}} &= \mathbf{v}_i \frac{\partial \mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)}{\partial \hat{\mathbf{y}}_i} \frac{\partial \hat{\mathbf{y}}_i}{\partial \mathbf{X}_{i,k}} \frac{\partial \mathbf{X}_{i,k}}{\partial \mathbf{S}_{k,l}}, \\ \frac{\partial \mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)}{\partial \hat{\mathbf{y}}_i} &= \sigma(\hat{\mathbf{y}}_i) - \mathbf{y}_i, \\ \frac{\partial \hat{\mathbf{y}}_i}{\partial \mathbf{X}_{i,k}} &= \mathbf{w}_k, \quad \frac{\partial \mathbf{X}_{i,k}}{\partial \mathbf{S}_{k,l}} = \frac{2(\mathbf{S}_{k,l} - \mathbf{T}_{i,j^*+l-1})}{L}, \\ j^* &= \arg\min_{j=1,2,\cdots,J} \frac{1}{L} \sum_{l=1}^{L} (\mathbf{T}_{i,j+l-1} - \mathbf{S}_{k,l})^2, \end{aligned} \tag{12}$$

$$\frac{\partial G_i}{\partial \mathbf{w}_k} = \begin{cases} \mathbf{v}_i \left(\frac{\partial \mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)}{\partial \hat{\mathbf{y}}_i} \mathbf{X}_{i,k} + \frac{2\alpha \mathbf{w}_k}{I} \right) & \text{if } k \neq 0, \\ \mathbf{v}_i \frac{\partial \mathcal{L}(\mathbf{y}_i,\hat{\mathbf{y}}_i)}{\partial \hat{\mathbf{y}}_i} & \text{if } k = 0, \end{cases}$$

ここで，勾配 $\partial \mathbf{X}_{i,k} / \partial \mathbf{S}_{k,l}$ については，[5] と同様に劣勾配を用いて導出することで時系列長に対する計算量を線形オーダーまで削減できる [9, 10]．

4.3 全体のアルゴリズム

全体のアルゴリズムを Algorithm 2 に示す．1 行目は第 1 ステップに対応し，2–15 行目が第 2 ステップに対応する．2 行目の分類器の重み $\mathbf{w}$ の初期化では，過学習を避けるため Algorithm 1 の結果を利用せず零ベクトルに初期化する．

4 行目では SPL パラメータ λ_y と SPL 関数のパラメータ q の値を動的に設定する．学習データの信頼度 $\mathbf{v}$ が徐々に増加するように（λ_y が増加し q が減少するように）以下のように設定する．

$$\lambda_y = \sum_{i=1}^{\left\lceil \frac{\tilde{m}}{\tilde{M}} |\mathcal{I}_y| \right\rceil} \mathcal{L}(\mathbf{y}_{(i)_y}, \hat{\mathbf{y}}_{(i)_y}), \quad q = 100^{\frac{1}{2} - \frac{\tilde{m}-1}{\tilde{M}-1}}, \tag{13}$$

ここで，$\mathcal{I}_y$ はクラス y のインスタンスの集合であり，$(i)_y$ はロジスティック損失 $\mathcal{L}$ の昇順にソートした際の $\mathcal{I}_y$ のインデックスである．また，$\tilde{M}$ は交互最適化の最大繰り返し回数であり，$\tilde{m}$ は現時点での繰り返し回数である．5–13 行目で shapelets $\mathbf{S}$ と分類器の重み $\mathbf{w}$ を勾配降下法により学習し，14 行目で学習データの信頼度 $\mathbf{v}$ を式 (11) で学習する．

Algorithm 2 Overall LTSSFS algorithm

Input: 時系列インスタンス: $\mathbf{T} \in \mathbb{R}^{I \times Q}$; クラスラベル: $\mathcal{Y}^I$; Shapelet の長さ: L; Shapelets の数: K; ℓ_2 正則化パラメータ: α; 勾配降下法の学習率: η; 全繰り返し回数: M; 内側ループの繰り返し回数: $\tilde{M}$

Output: Shapelets: $\mathbf{S} \in \mathbb{R}^{K \times L}$; 分類器の重み: $\mathbf{w} \in \mathbb{R}^{K+1}$; 学習データの信頼度: $\mathbf{v} \in [0,1]^I$

1: Discover intelligent initial shapelets $\mathbf{S}$ by Algorithm 1.
2: Initialize $\mathbf{w} \leftarrow \mathbf{0}$ and $\mathbf{v} \leftarrow \mathbf{1}$.
3: **for** $\tilde{m} = 1,2,\cdots,\tilde{M}$ **do**
4: Set λ_y for $y \in \mathcal{Y}$ and q according to Eq. (13).
5: **for** $m = 1,2,\cdots,M/\tilde{M}$ **do**
6: **for** $i = 1,2,\cdots,I$ **do**
7: **for** $k = 1,2,\cdots,K$ **do**
8: $\mathbf{S}_{k,l} \leftarrow \mathbf{S}_{k,l} - \eta \frac{\partial G_i}{\partial \mathbf{S}_{k,l}}$ in Eq. (12) for $l = 1,2,\cdots,L$.
9: $\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \frac{\partial G_i}{\partial \mathbf{w}_k}$ in Eq. (12).
10: **end for**
11: $\mathbf{w}_0 \leftarrow \mathbf{w}_0 - \eta \frac{\partial G_i}{\partial \mathbf{w}_0}$ in Eq. (12).
12: **end for**
13: **end for**
14: $\mathbf{v}_i \leftarrow \arg\min_{\mathbf{v}_i \in [0,1]} G_i$ in Eq. (11) for $i = 1,2,\cdots,I$.
15: **end for**
16: **return** $\mathbf{S}$,$\mathbf{w}$,$\mathbf{v}$

4.4 説明性評価に向けた解析

発見された shapelets の解釈を補助するため，従来研究 [5] のように，shapelets と時系列インスタンスとの距離と分類器の重み $\mathbf{w}_k$ との関係を定理としてまとめる．

定理 2. 任意の $i = 1,2,\cdots,I$ 及び $k = 1,2,\cdots,K$ に対して，クラスが $\mathbf{y}_i = 1$ の場合に分類器の重み $\mathbf{w}_k$ の符号に応じて次の 3 ケースが成り立つ．クラスが $\mathbf{y}_i = 0$ の場合は $\mathbf{w}_k$ の符号が逆転する．

(a) $\mathbf{w}_k < 0$ の場合，$\mathbf{T}_i$ と $\mathbf{S}_k$ の距離が減少すると式 (4) の分類損失は減少する．

(b) $\mathbf{w}_k > 0$ の場合，$\mathbf{T}_i$ と $\mathbf{S}_k$ の距離が減少すると式 (4) の分類損失は増加する．

(c) $\mathbf{w}_k = 0$ の場合，$\mathbf{S}_k$ は式 (4) の分類損失に寄与しない．

LTSSFS は，式 (4) の分類損失を削減するように shapelets $\mathbf{S}$ と分類器の重み $\mathbf{w}$ を同時に学習する．そのため，$\mathbf{w}_k < 0$ または $\mathbf{w}_k > 0$ の場合，定理 2 より shapelet $\mathbf{S}_k$ はクラス 1 またはクラス 0 の時系列データとの距離が近くなるように学習される傾向がある．そのため以降の評価では，shapelet $\mathbf{S}_k$ は $\mathbf{w}_k < 0$ の場合にクラス 1 に属し $\mathbf{w}_k > 0$ の場合にクラス 0 に属すると決める．

5 分類性能評価

5.1 実験設定

提案手法 **LTSSFS** を次の shapelets 学習法と比較する．**LTS** [2] は，時系列分類において高い正解率を達成する [2, 11, 25]．**CSLTS** [4] は，コスト考慮型のロジスティック損失関数を導入することで LTS を拡張した手法であり，2 クラス間でインスタンス数の偏りがある（不均衡データである）とき高い F 値を達成する．**SFS1st** は，提案手法において第 2 ステップ

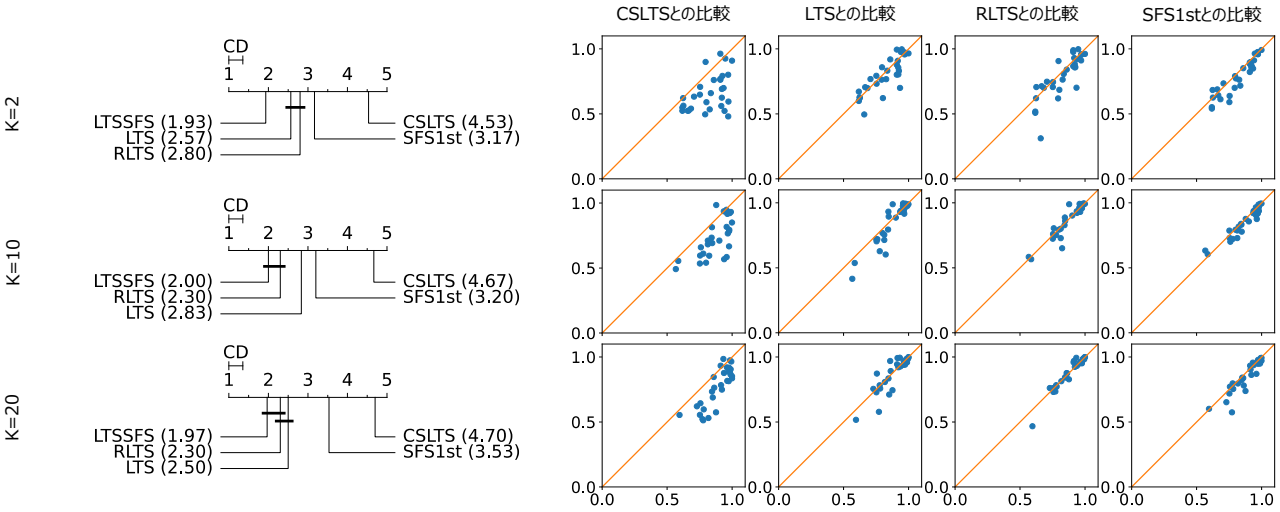

図 2　$K=2$(上段), 10(中段), 20(下段) における AUC の比較．左: CD diagrams．右: AUC 値．

を行わず，第 1 ステップ（Algorithm 1）の初期 shapelets を直接用いる．**RLTS** は，提案手法の第 2 ステップは行うが，第 1 ステップ（Algorithm 1）を行わず従来手法 [2, 4, 5, 6] と同様に教師無しクラスタリングにより初期 shapelets を選ぶ．

時系列分類で標準的な UCR データセット [26] の中に，2 クラス分類が対象でデータサイズが小さく（$IQ^2 \leqq 30{,}000{,}000$）欠損の無いデータセットが 30 個あり，それら 30 個のデータセットを用いる．学習データとテストデータの分け方はデフォルトの設定を使用し，同様の実験を各 5 回繰り返す．CSLTS では，クラス 1 の見逃しに対してより厳しいペナルティを課すため，学習データ数が少ないクラスを 1 とし多いクラスを 0 とする．学習データの 10%をハイパーパラメータのチューニング用に取っておく [3]．各手法に対して，学習率 η は {0.01,0.1,1,10} でチューニングし，ℓ_2 正則化パラメータ α は {0.01,1,100} でチューニングする．CSLTS では追加のハイパーパラメータがあり $\theta \in \{1{,}100\}$ と $D \in \{0.1{,}10\}$ でチューニングする．全手法で，shapelets の長さ L を $0.2 \times Q$ とし，勾配降下法の繰り返し回数の合計 M を 600 とし，交互最適化の繰り返し回数 $\tilde{M}$ を 20 とする．AUC は分類器の閾値に依存せず均衡／不均衡データのいずれでも適切な分類性能指標である．次節では，shapelets の数 K を {2,10,20} で変化させたときの AUC を比較する．

5.2　実 験 結 果

図 2 は，30 個のデータセットに対するテストデータの AUC の結果である．左側は Critical Difference (CD) diagrams [27] であり，右側には LTSSFS とその他の比較手法とで AUC 値をプロットしている．全体として，LTSSFS の性能が他の比較手法よりも良いことが分かる．

図 2 (左) の CD diagrams では，括弧内の値はランキングの平均値を表しており値が小さいほど性能が良い．太棒で繋がれた手法らは信頼区間 95%でお互いに有意差が無いことを表している．まず，SFS1st と比べて LTSSFS や RLTS や LTS の性能が有意に良いため，LTSSFS の第 2 ステップは除去できず，shapelets を部分時系列から探索するだけでは不十分であり shapelets を学習することが性能改善に重要であることが分かる．一方，LTSSFS を RLTS と比べて，shapelets の数が大きいとき（$K=10$ と 20）では有意差はないものの全てのケース（$K=2$ と 10 と 20）で LTSSFS の性能が良いことから，shapelets 学習法に第 1 ステップを導入することは性能改善に重要であることも分かる．RLTS を LTS と比べて，$K=10$ で RLTS の性能が有意に良いことから，SPL を導入することが有効であることも分かる．最終的に，第 1 ステップと第 2 ステップの両方を行う LTSSFS の性能が全比較手法の中で最良であり，特に shapelets の数が少ない（$K=2$）ときその差は有意となる．

図 2 (右) では，横軸と縦軸は LTSSFS とその他の比較手法の AUC の値をそれぞれプロットしている．橙色の対角線は 2 つの手法が同じ AUC 値であることを意味し，この対角線から下側に離れるほど LTSSFS が他手法よりも性能が良いことを意味する．第 1 ステップを行わない全ての LTS と CSLTS と RLTS では，shapelets の数が少ない（$K=2$）とき性能が非常に低下する場合があることが分かる．対して，第 1 ステップを行う LTSSFS と LTS1st ではこのような性能低下が起こりにくい．これは，従来の shapelets 学習法で教師無しクラスタリングによって初期 shapelets を決める際に shapelets の数（つまりクラスタ数）が少ないと，分類に有効な初期 shapelets が含まれにくくなるためである．この結果からも，shapelets の数が少ない（$K=2$）ときの第 1 ステップの有効性を確認できる．以降では，$K=2$ の場合における LTSSFS の説明性を評価する．

6　ケーススタディ評価

本章では，shapelets の数 K が 2 個の場合でケーススタディを通して LTSSFS が学習した shapelets を評価する．実験設定は 5.1 節と同じである．分類性能の比較には最良の従来手法である LTS を用いる．一般性を損なうことなく shapelets のイ

3 : もし学習データの数が少なすぎて 10%のインスタンスを取れない場合は，学習データをそのまま用いてハイパーパラメータをチューニングする．

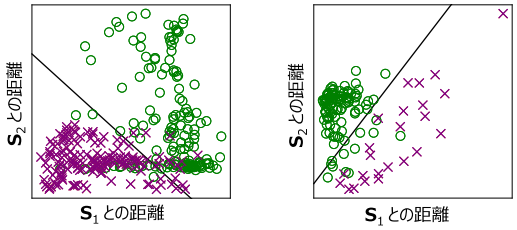

図 3　LTSSFS の特徴空間．左: GunPointAgeSpan データセット．右: Toe-segmentation データセット．

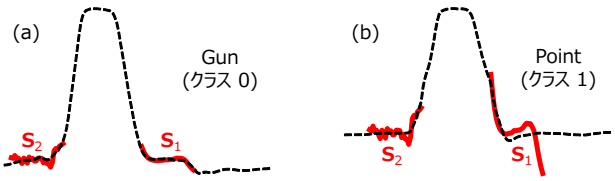

図 4　GunPointAgeSpan データと LTSSFS が学習した shapelets.

ンデックスは分類器の重みの絶対値の降順にソートする（つまり，$|\mathbf{w}_1| \geqq |\mathbf{w}_2|$）．4.4 節で前述したように，$\mathbf{w}_k$ の符号から shapelet $\mathbf{S}_k$ が属するクラスを決める．

6.1　GunPointAgeSpan のデータセット

このデータセット [26] は，元々は 2003 年にリリースされて時系列分類で広く用いられている GunPoint データセットの拡充版である．この拡充版データセットには，年齢や性別の異なる人間が銃を引く動作と指を指す動作をセンシングしそれらの時系列データが収集されている．時系列分類の目的は，銃を引く動作（クラス 0）と指を指す動作（クラス 1）とを分類することである．

図 3 (左) は，LTSSFS の 2 次元特徴空間を表す．各軸は式 (1) で定義される shapelet と時系列インスタンスとの距離である．テストデータにおけるクラス 0 とクラス 1 のインスタンスに対して，それぞれ “×” と “○” とで特徴ベクトルをプロットする．黒線は学習データで学習した分類境界である．テストデータにおける LTSSFS と LTS の AUC はそれぞれ 0.93 と 0.83 であり，大きく分類性能が改善されている．

図 4 (a) と (b) では，それぞれクラス 0 とクラス 1 の時系列インスタンスを黒点線で描き，それらにベストマッチングする位置に shapelets を赤太線で重ね描きしている．$\mathbf{w}_1$ と $\mathbf{w}_2$ の符号はいずれも正であったため，2 つの shapelets $\mathbf{S}_1$ と $\mathbf{S}_2$ ともにクラス 0 に属する．元々の GunPoint データセットでは，銃を引く動作と指を指す動作との違いは，ホルスターから銃を取り出すところとホルスターに銃を戻すところに現れることが分かっている [3]．本実験により，この拡充版のデータセットにおいても，同様の 2 箇所に波形パターンの違いがあることが分かった．特に LTSSFS では，shapelets の数 K が 2 個と少数であっても，過不足なくこれら 2 箇所の違いが shapelets として捉えられている．

6.2　Toe-segmentation のデータセット

このデータセットの目的は，正常と異常な歩行を分類することである [28]．本実験では，UCR データセット [26] に含まれる y 軸の動作をセンシングした時系列データ（ToeSegmentation2）を

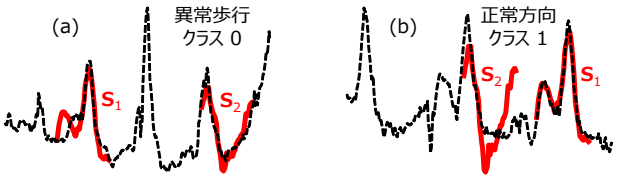

図 5　Toe-segmentation データと LTSSFS が学習した shapelets.

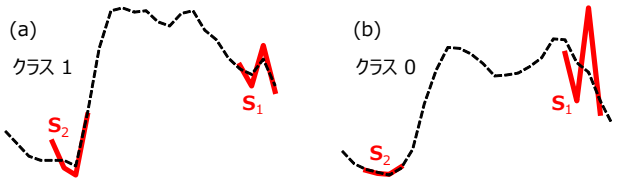

図 6　Italy Power Demand データと LTSSFS が学習した shapelets.

用いる．異常と正常な歩行をそれぞれクラス 0 とクラス 1 とする．

図 3 (右) は，LTSSFS の 2 次元特徴空間を表す．各軸は式 (1) で定義される shapelet と時系列インスタンスとの距離である．テストデータにおけるクラス 0 とクラス 1 のインスタンスに対して，それぞれ “×” と “○” とで特徴ベクトルをプロットする．テストデータにおける LTSSFS と LTS の AUC はそれぞれ 0.93 と 0.70 であり，大きく分類性能が改善されている．

図 5 (a) と (b) では，それぞれクラス 0 とクラス 1 の時系列インスタンスを黒点線で描き，それらにベストマッチングする位置に shapelets を赤太線で重ね描きしている．$\mathbf{w}_1$ と $\mathbf{w}_2$ の符号はそれぞれ負と正であった．そのため，shapelets $\mathbf{S}_1$ と $\mathbf{S}_2$ はそれぞれ正常クラス 1 と異常クラス 0 に属する．分類器の重みの絶対値が最も大きい（つまり，最も分類に寄与する）shapelet $\mathbf{S}_1$ に着目すると，$\mathbf{S}_1$ は正常歩行に現れる大小のこぶからなる典型的な波形パターンを捉えていることが分かる．このように，LTSSFS では shapelets の数が 2 個と少なくても適切な波形パターンを shapelets として発見できる．

6.3　Italy Power Demand のデータセット

このデータセット [26] にはイタリアの 12ヶ月にわたる電力需要が 1 日単位の時系列インスタンスとして収集されている [29]．目的は時系列インスタンスを収集した日が 10 月から 3 月に属するか 4 月から 9 月に属するかを分類することである．テストデータにおける LTSSFS と LTS の AUC はそれぞれ 0.80 と 0.62 であり，大きく分類性能が改善されている．図 6 (a) と (b) では，異なるクラスの時系列インスタンスにベストマッチングした位置で shapelets を重ねて描いている．Shapelets の長さが短く（$L=4$），shapelets の数も少ない（$K=2$）にも関わらず，LTSSFS では分類に有効な波形パターンを発見できることが分かる．

7　ま　と　め

本研究では，初期 shapelets の選定と shapelets の学習とで一貫して共通の損失関数を最小化する shapelets 学習法 LTSSFS を提案した．第 1 ステップでは，初期 shapelets の選定に教師有り特徴選択を導入し，RFE に基づき学習データに含まれる多

数の部分波形から分類に有効なものを初期 shapelets として発見する．第 2 ステップでは，shapelets 学習法に SPL を導入し，shapelets と分類器の重みと学習データの信頼度とをそれらの相互依存を考慮して同時に最適化し，分類損失を更に削減させる．UCR データセットを用いて，LTSSFS は LTS や CSLTS といった shapelets 学習法と比較して AUC を改善することを確認した．また，UCR データセットを用いたケーススタディを通して shapelets 数 K が 2 個の場合に LTSSFS で学習した shapelets の説明性を確認した．

文　　献

[1] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. “Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures”. In: *Proc. VLDB Endow.* (2008).

[2] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme. “Learning Time-series Shapelets”. In: *KDD*. ACM, 2014.

[3] L. Ye and E. Keogh. “Time Series Shapelets: A New Primitive for Data Mining”. In: *KDD*. ACM, 2009.

[4] S. Roychoudhury, M. Ghalwash, and Z. Obradovic. “Cost Sensitive Time-Series Classification”. In: *ECML PKDD*. Springer, 2017.

[5] A. Yamaguchi, S. Maya, K. Maruchi, and K. Ueno. “LTSpAUC: Learning Time-series Shapelets for Optimizing Partial AUC”. In: *SDM*. SIAM, 2020.

[6] Q. Zhang, J. Wu, H. Yang, Y. Tian, and C. Zhang. “Unsupervised Feature Learning from Time Series”. In: *IJCAI*. AAAI Press, 2016.

[7] M. P. Kumar, B. Packer, and D. Koller. “Self-Paced Learning for Latent Variable Models”. In: *International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2010.

[8] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. “Curriculum Learning”. In: *ICML*. ACM, 2009.

[9] A. Yamaguchi and K. Ueno. “Learning Time-series Shapelets via Supervised Feature Selection”. In: *SDM*. SIAM, 2021.

[10] A. Yamaguchi, S. Maya, and K. Ueno. “RLTS: Robust Learning Time-series Shapelets”. In: *ECML PKDD*. Springer, 2020.

[11] L. Hou, J. T. Kwok, and J. M. Zurada. “Efficient Learning of Timeseries Shapelets”. In: *AAAI*. AAAI Press, 2016.

[12] H. Zou, Y. Zhou, J. Yang, W. Gu, L. Xie, and C. J. Spanos. “WiFi-Based Human Identification via Convex Tensor Shapelet Learning”. In: *AAAI*. AAAI Press, 2018.

[13] Q. Ma, W. Zhuang, and G. Cottrell. “Triple-Shapelet Networks for Time Series Classification”. In: *ICDM*. IEEE Computer Society, 2019.

[14] Q. Ma, W. Zhuang, S. Li, D. Huang, and G. Cottrell. “Adversarial Dynamic Shapelet Networks”. In: *AAAI*. AAAI Press, 2020.

[15] Z. Fang, P. Wang, and W. Wang. “Efficient Learning Interpretable Shapelets for Accurate Time Series Classification”. In: *ICDE*. IEEE Computer Society, 2018.

[16] L. Jiang, D. Meng, T. Mitamura, and A. G. Hauptmann. “Easy Samples First: Self-Paced Reranking for Zero-Example Multimedia Search”. In: *MM*. ACM, 2014.

[17] Q. Zhao, D. Meng, L. Jiang, Q. Xie, Z. Xu, and A. G. Hauptmann. “Self-Paced Learning for Matrix Factorization”. In: *AAAI*. AAAI Press, 2015.

[18] H. Li and M. Gong. “Self-Paced Convolutional Neural Networks”. In: *IJCAI*. AAAI Press, 2017.

[19] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, et al. “Feature Selection: A Data Perspective”. In: *ACM Comput. Surv.* (2017).

[20] H. Yang, R. Fujimaki, Y. Kusumura, and J. Liu. “Online Feature Selection: A Limited-Memory Substitution Algorithm and Its Asynchronous Parallel Variation”. In: *KDD*. 2016.

[21] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. “Gene Selection for Cancer Classification Using Support Vector Machines”. In: *Mach. Learn.* (2002).

[22] J. Grabocka, M. Wistuba, and L. Schmidt-Thieme. “Fast classification of univariate and multivariate time series through shapelet discovery”. In: *Knowl. Inf. Syst.* (2016).

[23] S. Roychoudhury, F. Zhou, and Z. Obradovic. “Leveraging Subsequence-orders for Univariate and Multivariate Time-series Classification”. In: *SDM*. SIAM, 2019.

[24] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. “LIBLINEAR: A Library for Large Linear Classification”. In: *J. Mach. Learn. Res.* (2008).

[25] X. Li and J. Lin. “Evolving Separating References for Time Series Classification”. In: *SDM*. SIAM, 2018.

[26] H. A. Dau, E. Keogh, K. Kamgar, C.-C. Yeh Michael, Y. Zhu, et al. *The UCR Time Series Classification Archive*. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`. 2018.

[27] J. Demšar. “Statistical Comparisons of Classifiers over Multiple Data Sets”. In: *Journal of Machine learning research* (2006).

[28] L. Ye and E. Keogh. “Time Series Shapelets: A Novel Technique That Allows Accurate, Interpretable and Fast Classification”. In: *Data Min. Knowl. Discov.* (2011).

[29] E. Keogh, L. Wei, X. Xi, S. Lonardi, J. Shieh, and S. Sirowy. “Intelligent Icons: Integrating Lite-Weight Data Mining and Visualization into GUI Operating Systems”. In: *ICDM*. IEEE Computer Society, 2006.