時系列データのクラス分離性を強化するshapelets学習法

山口　晃広†　　植野　　研†　　鹿島　久嗣††

† 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
†† 京都大学 大学院情報学研究科 知能情報学専攻　〒 606-8501 京都府京都市左京区吉田本町
E-mail: †akihiro5.yamaguchi@toshiba.co.jp

あらまし　時系列データから shapelets と呼ばれる波形パターンを獲得し特徴量を学習するクラス分類手法が，説明性があり高い分類性能を両立するため注目を集めている．これらの従来手法では古典的なシグモイド損失関数を用いている一方で，コンピュータビジョンなどの分野ではクラス間の特徴量を積極的に分離する損失関数が提案されている．本研究では，スケーリングを自動調整することでクラス分離性を強化する既存手法に着想を得て shapelets 学習に適用可能な損失関数を提案する．更に，理論的に特徴量をクラス間で引き離しながら縮小させる正則化を提案し shapelets の説明性を維持する．UCR データセットを用いた実験では，少数の shapelets における説明性と AUC の性能向上を示す．

キーワード　クラス分類，時系列データ, shapelets, クラス分離性, 説明性, 機械学習

1　は じ め に

IoT の普及に伴い，機械学習を用いた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，学習時に正または負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．一般的なクラス分類とは異なり時系列分類では，時系列の並びや波形パターンの形状が重要であり，波形パターンの出現位置のずれや振幅の違い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，shapelets と呼ばれる時系列分類に用いる複数の波形パターンを発見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波形パターンに表れるというコンセプトに基づく．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，良好な分類性能を達成する．より重要なことは，医療・インフラ・製造などの産業分野の専門家に説明性を提供できることである．これらの専門家は，時系列データに関する専門知識を有し波形パターンに着目して分析することが多いため，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．そのため，このような産業分野の専門家に比較的受け入れられやすい．

Shapelets の研究は，学習データを構成する部分時系列を探索することで shapelets を獲得する探索ベースの手法に端を発しているが [3]，shapelets を学習データの部分時系列に制限されない任意の波形パターンとして分類器と同時に学習する shapelets 学習法が提案されている [2]．Shapelets 学習法では，探索ベースの手法 [3] よりも分類性能を向上し学習アルゴリズムの計算量を削減することができる [2, 4, 5, 6]．文献 [5] では，shapelets 学習法に探索ベースの手法を統合し，少ない shaplelets の個数で Area Under the Curve (AUC) を向上する手法も提案された．

従来の shapelets 学習法ではシグモイド関数を介した交差エントロピー損失関数（以降，シグモイド損失関数）が用いられており，クラス間で特徴量を積極的に分離させるわけではないため，十分な汎化性能が得られない可能性がある．一般にシグモイド損失関数では，クラス間で完全に分離可能な学習データに確率的勾配降下法（Stochastic Gradient Descent; SGD）を用いれば，クラス間のマージンは最大となるがその収束速度は非常に遅いことが理論的に証明されており [7]，実用上は殆ど達成できない．特に学習データが少数しかない場合には過学習が起こりテストデータにおける分類性能は悪くなる傾向がある．クラスラベルの付与にはコストがかかるため，データ収集が容易になってきたとしてもこのケースは起こりうる．それゆえ，shapelets 学習法を改良する一つの方向性として，特徴量の分離性を強化することで汎化能力やロバスト性を向上させることが考えられる．

時系列分類から離れた主にコンピュータビジョンの分野などでは，特徴量をクラス間で積極的に分離する識別的特徴量学習に向けて，ソフトマックス関数やシグモイド関数に基づく分類損失関数を拡張する研究が盛んに行われており，これによるクラス分類性能の改善も示されている [8, 9]．しかしながら，従来の shapelets 学習法では古典的な分類損失関数が依然用いられており，このような方向性の研究は未だ行われていない．

本研究では，特徴量のクラス分離性を強化する shapelets 学習法（Learning Time-series Shapelets Enhancing Discriminability; LTSED という技術）を提案する．文献 [10] に着想を得て，LTSED では SGD による勾配更新が積極的に行われるようにシグモイド損失関数のスケールパラメータを動的かつ自動的に調整する．一般的な深層学習を用いた識別的特徴量学習とは異なり，shapelets 学習法の特徴量は shapelet と時系列インスタンスとの距離に一致する．そこで，shapelets が一方のクラスの時系列データに類似しながら 2 クラス間の特徴量が離れることを理論的に保証する正則化を更に提案する．これらにより，

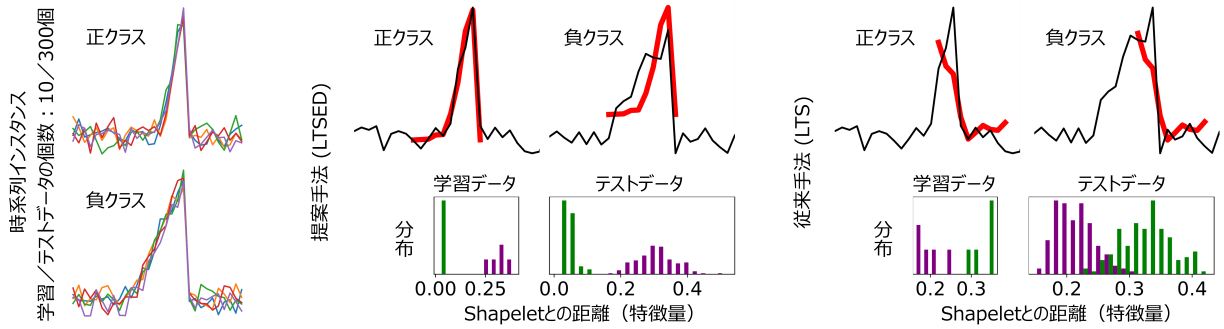

図 1　LTSED で学習した特徴量（shapelet との距離）は従来手法 [2] の場合よりもクラス間で大きく分離する．その結果，LTSED でのみテストデータにおいても特徴量が正しく分離する．

LTSED で学習された shapelets は説明性を維持しつつ，その各 shapelet に基づく特徴量はクラス間で積極的に引き離される．

図 1 の黒線は上に尖った波形周辺を拡大した時系列インスタンスである．上に尖った波形の傾きは負例では正例よりも緩やかに上昇するため，上に尖った波形に着目することで正例と負例を正しく分類できる．LTSED と従来手法 [2] のどちらでも，上に尖った波形周辺を（赤太線で描かれた）shapelets と特定することで学習データの全てのインスタンスを正しく分類する特徴量が得られている．しかしながら，LTSED における特徴量の分布は 2 クラス間でより離れている．その結果，従来手法 [2] では誤分類が発生するテストデータにおいても，LTSED では全インスタンスを正しく分類する特徴量が得られている．このように，LTSED では shapelets の説明性を損なわずに特徴量のクラス分離性を強化することで汎化性能を向上できることが，この簡単な例から確認できる．

1.1　本研究の主な貢献

本論文の主な貢献を以下に示す．なお，本内容は基本的に文献 [11] に準じ，式の導出や一部の証明及び関連研究については文献 [11] を参照いただきたい．

- クラス分離性を強化する shapelets 学習法を提案する．
- 提案する shapelets に対する正則化では，shapelets を適切な元の時系列データに類似させることで，shapelets の説明性を維持しながら理論的にクラス分離性を高める．
- コンピュータビジョンの分野における研究 [10] に着想を得て，シグモイド関数のスケーリングをパラメータフリーで自動調整する方法を shapelets の学習にも適用可能な形で提案する．
- UCR 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

2.1　記号と定義

時系列インスタンスを 2 クラス $\mathcal{Y}:=\{1,-1\}$ 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 $y_i\in\mathcal{Y}$ とする．図 1 のように，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．第 i 番目の時系列インスタンス $\mathbf{t}_i\in\mathbb{R}^Q$ の j 番目の値を $t_{i,j}$ と記述し，k 番目の shapelet $\mathbf{s}_k\in\mathbb{R}^K$ の l 番目の値を $s_{k,l}$ 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．第 j 番目の部分時系列 $(t_{i,j},t_{i,j+1},\cdots,t_{i,j+L-1})$ を $\mathbf{t}_{i,j:j+L-1}$ と略記する．従来の shapelets 学習法 [2, 4, 5, 6, 12, 13, 14] と同様に，j 番目の部分時系列 $\mathbf{t}_{i,j:j+L-1}$ と $\mathbf{s}_k$ との二乗ユークリッド距離を測り，$j=1,2,\cdots,J$ の中で最小なその距離を $\mathbf{t}_i$ と $\mathbf{s}_k$ の距離（非類似度）として以下で定義する．

$$x_{i,k}:=\min_{j=1,2,\cdots,J}\frac{1}{L}\sum_{l=1}^{L}(t_{i,j+l-1}-s_{k,l})^2. \tag{1}$$

ここで，i 番目のインスタンスにおける特徴ベクトル $\mathbf{x}_i\in\mathbb{R}^K$ は $(x_{i,1},x_{i,2},\cdots,x_{i,K})$ であり，特徴量 $x_{i,k}$ は明らかに非負である．

本研究では，shapelets の説明性を損なわずにクラス分離性を強化するように，shapelets $\mathbf{S}$ と分類器の重み（以降，分類重み） $\mathbf{w}$ とを学習する．

2.2　従来の shapelets 学習法

本節では文献 [2] の定式化を説明する．バイアス項を含めた分類重み $\mathbf{w}\in\mathbb{R}^{K+1}$ と特徴ベクトル $\mathbf{x}_i\in\mathbb{R}^K$ とを用いて，次の線形モデルで $\hat{y}_i$ を予測する．

$$\hat{y}_i:=\sum_{k=1}^{K}w_kx_{i,k}+w_0. \tag{2}$$

最終的な定式化は，特徴ベクトル集合 $\mathbf{X}$ を経由した shapelets $\mathbf{S}$ と分類重み $\mathbf{w}$ とを同時に学習する次の数理最適化問題となる．

$$\underset{\mathbf{S}\in\mathbb{R}^{K\times L},\mathbf{w}\in\mathbb{R}^{K+1}}{\text{minimize}}\sum_{i=1}^{I}F_i,\quad F_i:=\mathcal{L}(y_i,\hat{y}_i)+\frac{\alpha}{I}\sum_{k=1}^{K}w_k^2. \tag{3}$$

ここで，$\alpha\geqq 0$ は分類重みに対する ℓ_2 正則化パラメータである．クラスラベルが $\mathcal{Y}=\{1,-1\}$ であることに注意し [1]，次の交差エントロピー損失関数 $\mathcal{L}(\cdot,\cdot)$ を用いる．

$$\mathcal{L}(y_i,\hat{y}_i):=-\frac{1+y_i}{2}\ln(\sigma(\hat{y}_i))-\frac{1-y_i}{2}\ln(1-\sigma(\hat{y}_i)). \tag{4}$$

ここで，シグモイド損失関数 $\sigma(\cdot)$ は次で定義される．

1：$y_i=-1$ を $y_i=0$ と置換することで式 (4) は [2] の定式化と等価となる．

$$\sigma(\hat{y}_i) := \left(1+e^{-\hat{y}_i}\right)^{-1}. \tag{5}$$

式 (5) を介して式 (4) で定義されるシグモイド損失関数は，実際にはクラス間のマージンを十分に拡大できないため [7]，式 (3) における従来の定式化では，shapelet に基づく式 (1) の特徴量はクラス間で十分に分離されず特に学習インスタンスの数が限られる場合など汎化能力に課題があると考えられる．

2.3 教師有りの shapelets 初期化

式 (3) は非凸最適化問題であるため，SGD アルゴリズムにおける初期 shapelets の選択は最終的な分類性能に大きな影響を及ぼす．従来の shapelets 学習法 [2, 4, 6, 12, 13] では，元の部分時系列をクラスタリングしたセントロイドとして shapelets を初期化するため，特に shapelets の個数 K が少ない場合には分類に寄与するセントロイドつまりは初期 shapelets が殆ど含まれず分類性能が低下してしまう．

この課題を解決するため，文献 [5, 14] では，クラスラベルを活用して shapelets を初期化し，少数の shapelets でも分類性能を向上させる．特に文献 [5] では，教師有り特徴選択として多数の部分時系列から初期 shapelets を特定した後，SGD アルゴリズムで元の部分時系列に制限されない shapelets を学習する．

これらの従来手法では，少数の shapelets でも比較的高い分類性能を達成できるが，古典的なシグモイド損失関数を依然採用しており，汎化能力において更なる改善の余地がある．

3 提案手法の定式化

本章では，シグモイド損失関数のスケールパラメータと分類重みの正規化とを導入した後，shapelets を時系列データに類似させる際に分類性能を劣化させない条件を導出し，それに基づいて shapelets に対する正則化（以降，shapelets 正則化）を提案する．その後，LTSED の数理最適化問題をまとめ，shapelets 正則化のクラス分離性を理論的に解析する．

3.1 スケールパラメータと正規化した分類重みの導入

4.1 節で後述するようにシグモイド損失関数のスケールパラメータを動的かつ自動的に調整するため，式 (5) の代わりにスケールパラメータ $\beta>0$ を持つ次のシグモイド関数を用いる．

$$\sigma(\hat{y}_i) := \left(1+e^{-\beta\hat{y}_i}\right)^{-1}. \tag{6}$$

式 (2) の代わりに，バイアス項を除く分類重みに文献 [15] の正規化を適用した次の線形モデルを用いる．

$$\hat{y}_i := \sum_{k=1}^{K} \tilde{w}_k x_{i,k} + w_0, \quad \tilde{w}_k := \frac{w_k}{\|\mathbf{w}_{1:K}\|_2}. \tag{7}$$

ここで，$\|\mathbf{w}_{1:K}\|_2 := \sqrt{\sum_{k'=1}^{K} w_{k'}^2}$ である．なお，この正規化により LTSED では式 (3) のハイパーパラメータ α を除外する．

3.2 Shapelets 正則化の定式化

分類損失を最小化するように無制限に任意の形状を学習した shapelets は，実際の時系列データと類似していない可能性がある．特に，LTSED では式 (6) のスケールパラメータを動的に調整して特徴量をクラス間で積極的に引き離そうとするため，shapelets が実際の時系列データから乖離しやすくなり，shapelets 本来の意図に反して説明性が損なわれてしまう．そこで，クラス分離性を悪化させることなく，shapelets が元の時系列データから乖離してしまうことを抑制する shapelets 正則化を提案する．Shapelets 正則化では，式 (1) より特徴量 $x_{i,k}$ は shapelet と時系列データとの距離に一致することに注意して，クラス間のマージンが拡大するように特徴量 $x_{i,k}$ を縮小させる．

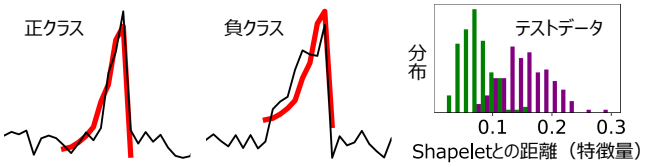

図 2　Shapelet に基づく特徴量に標準的な ℓ_2 正則化を適用すると，特徴量のクラス間の分離性は低下する．

まず，shapelet に基づく特徴量 $x_{i,k}$ に対して標準的な ℓ_p 正則化を適用してしまうと，クラス分離性が悪化することを説明する．図 2 の黒線は図 1 の黒線と同じ時系列インスタンスであり，図 2 の赤太線は特徴量の縮小に対して標準的な ℓ_2 正則化を適用して学習した shapelet である．Shapelet はどちらのクラスの時系列データにも類似してしまうため，この特徴量のクラス間の分離性は図 2 (右) の分布に示すように悪い．

次の定理は，文献 [5] の Proposition 4.1 と同様に，時系列インスタンス $\mathbf{t}_i$ と shapelet $\mathbf{s}_k$ との距離を縮めること（つまり，shapelet に基づく特徴量 $x_{i,k}$ を縮小すること）は，$y_i w_k < 0$ の条件下で分類性能に悪影響を及ぼさないことを理論的に保証する．

定理 1. 任意の $i=1,2,\cdots,I$ 及び $k=1,2,\cdots,K$ に対して，i 番目の時系列インスタンス $\mathbf{t}_i$ と k 番目の shapelet $\mathbf{s}_k$ との距離は，式 (1) で定義される特徴量と一致することに注意する．$\mathbf{t}_i$ と $\mathbf{s}_k$ の距離が縮まるとき次の 3 ケースがある:

(a) $y_i w_k < 0$ の場合，式 (4) の分類損失は減少する．
(b) $y_i w_k > 0$ の場合，式 (4) の分類損失は増加する．
(c) $w_k = 0$ の場合，式 (4) の分類損失は変化しない．

3.2.1 不連続なステップ関数を用いたナイーブな方法

定理 1 に従い，簡易的ではあるが実用的でない不連続な定式化から始める．分類重み $\mathbf{w}$ と式 (1) の特徴ベクトル集合 $\mathbf{X}$ とクラスラベル集合 $\mathbf{y}$ を用いて，shapelets 正則化 Ω を次で定義する．

$$\begin{aligned} \Omega(\mathbf{w},\mathbf{X},\mathbf{y}) &:= \sum_{i=1}^{I} \Omega_i(\mathbf{w},\mathbf{x}_i,y_i), \\ \Omega_i(\mathbf{w},\mathbf{x}_i,y_i) &:= \sum_{k=1}^{K} h(-y_i w_k) x_{i,k}. \end{aligned} \tag{8}$$

ここで，$x_{i,k}$ は式 (1) より非負である．図 3 の青破線に示すように，本節の $h(\cdot)$ は次の不連続なステップ関数である．

$$h(v) := \begin{cases} 0 & \text{if } v \leq 0, \\ 1 & \text{if } 0 < v. \end{cases} \tag{9}$$

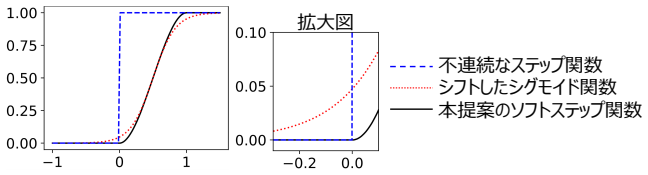

図 3　3 種類のステップ関数.

ここで，変数 v は実数値をとる．式 (9) を代入した式 (8) において，定理 1 の条件 (a) を満たさない項は常に 0 となる．したがって，shapelets 正則化 Ω は，非負の特徴量 $x_{i,k}$ を縮小する際にはクラス分類性能を理論的に劣化させない．

式 (9) のステップ関数 $h(v)$ は $v=0$ で不連続である．その結果，勾配降下法に基づく最適化過程において，$k=1,2,\cdots,K$ で分類重み w_k が 0 付近で変化すると，shapelets 正則化 Ω が不連続的に変化し不安定になる傾向がある．この課題を解決するため，次節では式 (9) のステップ関数を滑らかなソフトステップ関数に置き換える．

3.2.2　ソフトステップ関数に改良した方法

本節では，分類性能が悪化しないことを理論的に保証しつつ，最適化過程を安定させるためにソフトステップ関数を導入する．滑らかなステップ関数や活性化関数としてはシグモイド関数が一般的ではあるが，定理 1 の条件 (b) の下でも非負の特徴量 $x_{i,k}$ を縮小してしまい理論的にクラス分類性能が悪化する．これは，図 3 の赤点線で示すように，シグモイド関数は（たとえ水平にシフトしたとしても）変数が $(-\infty,\infty)$ の範囲で常に 0 より大きい値を出力するためである．したがって，変数が負の場合には厳密に 0 を出力する滑らかなステップ関数が必要となる．

図 3 の黒実線で示すように，滑らかでかつ $v<0$ で常に $h(0)=0$ となるソフトステップ関数 $h(v)$ を導入する．区間の右端 $\gamma>0$ が与えられると，$h(\cdot)$ は区間の左側で 0 となり右側で 1 となり区間 $[0,\gamma]$ で 3 乗多項式となるように次で定義される．

$$h(v):=\begin{cases}0 & \text{if}\ \ v\leqq 0,\\ 3\left(\frac{v}{\gamma}\right)^2-2\left(\frac{v}{\gamma}\right)^3 & \text{if}\ \ 0<v<\gamma,\\ 1 & \text{if}\ \ \gamma\leqq v,\end{cases}\qquad (10)$$

ここで，$\gamma>0$ はステップ関数の値がどの程度徐々に変化するかを決める．特に，$\gamma\to 0$ の場合に式 (9) における不連続なステップ関数に一致し，$\gamma\to\infty$ の場合には常に $h(v)=0$ となる．本研究では全ての実験で $\gamma=100$ を用いる．

ソフトステップ関数はコンピュータグラフィックスで良く用いられており，近年計算時間を短縮するためにソフト決定木にも採用された [16]．一方，本研究ではクラス分類性能が悪化しないことを理論的に保証するためにこのソフトステップ関数を用いる．

最終的には，式 (9) の代わりに式 (10) を代入した式 (8) の shapelets 正則化を用いる．この最終的な shapelets 正則化 Ω は，定理 1 の条件 (a) を満たさない場合常に 0 となるため，特徴量 $x_{i,k}$ が縮小してもクラス分類性能を劣化させない．また，この Ω は式 (10) のソフトステップ関数を介することで常に滑らかであり，最適化過程も安定化される，

3.3　数理最適化問題の定式化

まとめると，LTSED の定式化は特徴ベクトル集合 $\mathbf{X}$ を介した shapelets $\mathbf{S}$ と分類重み $\mathbf{w}$ とを次のように同時に最適化する．

$$\underset{\substack{\mathbf{S}\in\mathbb{R}^{K\times L}\\ \mathbf{w}\in\mathbb{R}^{K+1}}}{\text{minimize}}\sum_{i=1}^{I}G_i,\quad G_i:=\mathcal{L}(y_i,\hat{y}_i)+\lambda\Omega_i(\mathbf{w},\mathbf{X},\mathbf{y}).\qquad (11)$$

ここで，分類損失関数 $\mathcal{L}(\cdot,\cdot)$ は，式 (1), (6), (7) を介した式 (4) であり，shapelets 正則化は式 (10) を介した式 (8) であり，正則化パラメータ $\lambda\geqq 0$ はインスタンスごとに分解した目的関数 G_i の 2 つの項のバランスをとる．3.1 節で述べたように式 (3) のハイパーパラメータ α を使用しない．

LTSED では式 (4) の分類損失を削減するように shapelets と分類重みと式 (8) の shapelets 正則化とを同時に最適化するため，定理 1 から文献 [5] と同様に次の性質が導かれる．

性質 1. 分類重み $\mathbf{w}$ が与えられと，$k=1,2,\cdots,K$ に対して k 番目の shapelet は $w_k<0$ または $w_k>0$ のときにそれぞれ正または負のクラスの時系列データに類似する傾向がある．したがって，$w_k<0$ または $w_k>0$ のときに k 番目の shapelet はそれぞれ正または負のクラスに属するとする．

3.4　Shapelets 正則化の解析

本節では，式 (10) を介した式 (8) における shapelets 正則化 Ω のクラス分離性を解析する．次の定理により，勾配 $\partial\Omega/\partial\mathbf{X}$ における降下方向への特徴ベクトル集合 $\mathbf{X}$ の更新は，特徴ベクトル集合 $\mathbf{X}$ をクラス間分離を改善する方向へ移動させる．

定理 2. 各 $i=1,2,\cdots,I$ に対して $y_i w_k<0$ となる $k\in\{1,2,\cdots,K\}$ が存在する場合，勾配 $\partial\Omega_i/\partial\mathbf{x}_i$ の降下方向に特徴ベクトル $\mathbf{x}_i$ を更新すると，分類境界からの $\mathbf{x}_i$ の距離が改善する．また，その距離の改善分は次式で与えられる．

$$\sum_{k=1}^{K}-y_i\tilde{w}_k\eta h(-y_i w_k),\quad \tilde{w}_k:=\frac{w_k}{\|\mathbf{w}_{1:K}\|_2},\qquad (12)$$

ここで，$\eta>0$ は学習率であり，$h(\cdot)\geqq 0$ は式 (10) のステップ関数である．逆に，全ての $k=1,2,\cdots,K$ に対して $y_i w_k\geqq 0$ となる場合，特徴ベクトル $\mathbf{x}_i$ は勾配 $\partial\Omega_i/\partial\mathbf{x}_i$ で更新されない．

証明. 特徴ベクトル $\mathbf{x}_i$ と分類境界との距離が勾配 $\partial\Omega_i/\partial\mathbf{x}_i$ によって更新される量は更新前後の差として次で計算される．

$$\begin{aligned}&y_i\left(\left\langle\tilde{\mathbf{w}},\mathbf{x}_i-\eta\frac{\partial\Omega_i}{\partial\mathbf{x}_i}\right\rangle+w_0\right)-y_i(\langle\tilde{\mathbf{w}},\mathbf{x}_i\rangle+w_0)\\ &=\sum_{k=1}^{K}-y_i\tilde{w}_k\eta h(-y_i w_k),\quad \tilde{\mathbf{w}}:=\frac{\mathbf{w}_{1:K}}{\|\mathbf{w}_{1:K}\|_2}.\end{aligned}\qquad (13)$$

上式は，$y_i w_k<0$ となる $k\in\{1,2,\cdots,K\}$ が存在する場合に正の値となり，それ以外では 0 となる．ゆえに定理が成り立つ．□

4　提案手法の解法

本章では，3 章で述べた定式化の解法について述べる．4.1 節では，シグモイド損失関数のスケーリングパラメータを自動調整する方法を提案する．その後の節では，SGD における勾配を導出し，LTSED のアルゴリズムとその計算量をまとめる．

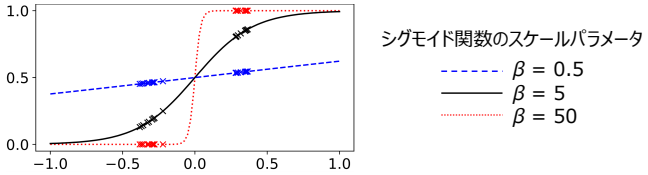

図 4　異なるスケールパラメータ β におけるシグモイド関数.

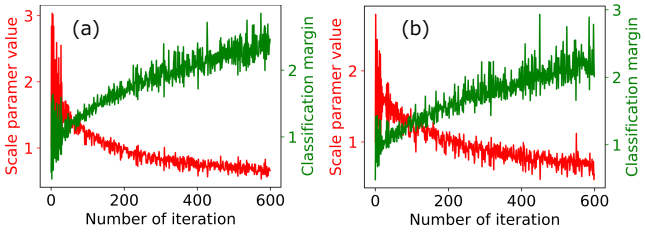

図 5　スケールパラメータ β と特徴ベクトルの分類境界からの平均距離の推移（左：GunPoint，右：ECGFiveDays）.

4.1　シグモイド損失関数のスケーリングパラメータの自動調整

式 (6) におけるシグモイド損失関数のスケールパラメータ β を適切に調整することは，shapelets $\mathbf{S}$ と分類重み $\mathbf{w}$ を効率的に学習するために重要であり，特に 3.1 節で述べた分類重みの正規化を行う場合はより重要となる．文献 [10] に着想を得て，LTSED ではハイパーパラメータのチューニングを必要とせずに自動的かつ動的に β をスケーリングする．

クラス分離性を強化する基本的なアイデアは，勾配の更新が十分に大きくなるように（つまり，shapelets $\mathbf{S}$ と分類重み $\mathbf{w}$ が十分に変化するように），スケールパラメータ β を設定することである．図 4 では，式 (6) のスケールパラメータ β の値を変化させたときのシグモイド関数 $\sigma(\cdot)$ の曲線を描いている．また，サンプリングした $(\hat{y}_i,\sigma(\hat{y}_i))$ の値を “×” でプロットしている．これにより，β が小さすぎる場合や大きすぎる場合（つまり，$\beta=0.5$ や $\beta=50$ の場合）では，勾配 $\partial\sigma(\hat{y}_i)/\partial\hat{y}_i$ は 0 に近づき，勾配の更新が効かなくなることが分かる．

数学的には，$\{|\hat{y}_i|\}_{i=1}^{I}$ の中央値を $\bar{y}$ として，シグモイド関数が (0,0.5) で対称であることに注意して，$\hat{y}_i=\bar{y}$ における勾配 $\partial\sigma(\hat{y}_i)/\partial\hat{y}_i$ の絶対値が最大となるようにスケールパラメータ β の設定値 β^* を次で定式化する．

$$\beta^* := \arg\max_{\beta>0}\left|\frac{d\sigma(\hat{y}_i)}{d\hat{y}_i}\right|_{\hat{y}_i=\bar{y}}\Bigg| = \arg\max_{\beta>0}\frac{\beta e^{-\beta\bar{y}}}{(1+e^{-\beta\bar{y}})^2}. \quad (14)$$

これを解くため，右辺の目的関数を β で微分してその結果を 0 と置くと次式となる．

$$\frac{d}{d\beta}\left(\frac{\beta e^{-\beta\bar{y}}}{(1+e^{-\beta\bar{y}})^2}\right) = \frac{(2\beta\bar{y}+(1-\beta\bar{y})(e^{\beta\bar{y}}+1))e^{\beta\bar{y}}}{(e^{\beta\bar{y}}+1)^3} = 0. \quad (15)$$

なお，式 (14) の等号と式 (15) の左側の等号は，それぞれ代数演算で導出できる．更に，式 (15) の右側の等号関係を次のように代数演算で簡潔に書き直す．

$$z(\beta) := (\beta\bar{y}-1)e^{\beta\bar{y}}-\beta\bar{y}-1=0. \quad (16)$$

上式を解くため，Newton-Raphson アルゴリズムを使用する．その際に必要となる $z(\beta)$ の 1 次微分と 2 次微分は次で計算される．

$$\frac{dz(\beta)}{d\beta}=\bar{y}\left(\beta\bar{y}e^{\beta\bar{y}}-1\right), \frac{d^2z(\beta)}{d\beta^2}=\bar{y}^2(\beta\bar{y}+1)e^{\beta\bar{y}}. \quad (17)$$

このアルゴリズムにおける 1 回分の更新は次式で計算される．

$$\beta^{\text{new}} \longleftarrow \beta-\left(\frac{d^2z(\beta)}{d\beta^2}\right)^{-1}\frac{dz(\beta)}{d\beta}. \quad (18)$$

式 (16) における 2 次のテイラー展開から，このアルゴリズムの適切な初期値 β^{init} を次のように設定する．

$$-2-\beta\bar{y}+\frac{\beta^2\bar{y}^2}{2}=0 \implies \beta^{\text{init}}=\frac{1+\sqrt{5}}{\bar{y}}. \quad (19)$$

式 (19) の適切な初期値を用いて，式 (17) と (18) で更新した場合，少ない回数の更新で大抵収束する．そのため，Newton-Raphson アルゴリズムの最大更新回数を 10 回と設定する．

図 5 では，UCR データセットの中の良く用いられる 2 種類のデータセットに対して，本節の自動調整方法の振る舞いを検証する．横軸は，SGD アルゴリズムの繰り返し回数である．赤と緑の曲線は，スケールパラメータ β の変化と，学習データにわたる特徴ベクトルの分類境界からの平均距離とをそれぞれ示している．各データセットにおいて期待通り，スケールパラメータ β の値が減少しながら，特徴量が分類境界から正しい方向へ離れていくことが分かる．

コンピュータビジョンなど他分野においてクラス分離性を強化する識別的特徴量学習は多数提案されている．その中で文献 [10] は，Triplet 損失 [17] や Contrastive 損失 [18] のようにインスタンスのペアリングを必要とせず，ArcFace [19] や CosFace [20] のようにマージンなどのハイパーパラメータも必要としない．しかしながら，文献 [10] の定式化では特徴ベクトルの正規化が本質的に必須であるが，shapelets に基づく特徴ベクトルの正規化は不適切である．これは，shapelet に基づく特徴量が時系列データとの距離に対応しており，図 1 に示すように特徴ベクトルが 1 次元にしかならない場合もあるためである．そのため，勾配の更新が大きくなるようにスケールパラメータを自動調整するというアイデアは文献 [10] と共通しているが，本節の式 (14)–(19) にわたる定式化は文献 [10] とは根本的に異なる．

4.2　勾配の導出

式 (11) の数理最適化問題において，shapelets $\mathbf{S}$ と分類重み $\mathbf{w}$ は SGD アルゴリズムで解かれるため，本章では $\mathbf{S}$ と $\mathbf{w}$ に対するインスタンスごとの目的関数 G_i の勾配を導出する．これらの勾配は微分の連鎖率により以下のように書き下せる [2]．

$$\frac{\partial G_i}{\partial s_{k,l}}=\frac{\partial\mathcal{L}(y_i,\hat{y}_i)}{\partial\hat{y}_i}\frac{\partial\hat{y}_i}{\partial x_{i,k}}\frac{\partial x_{i,k}}{\partial s_{k,l}}+\lambda\frac{\partial\Omega_i}{\partial x_{i,k}}\frac{\partial x_{i,k}}{\partial s_{k,l}},$$
$$\frac{\partial G_i}{\partial w_k}=\frac{\partial\mathcal{L}(y_i,\hat{y}_i)}{\partial\hat{y}_i}\frac{\partial\hat{y}_i}{\partial w_k}. \quad (20)$$

ここで，これらの勾配の構成要素は以下のように導出される．

2：式 (10) を介した式 (8) は w_k に対して微分可能であるが，実験的に性能が向上しないため，式 (20) の勾配 $\partial G_i/\partial w_k$ に勾配 $\partial\Omega_i/\partial w_k$ を含めない．

Algorithm 1 LTSED

Input: Time-series instances: $\mathbf{T}$; Class labels: $\mathbf{y}$; Shapelet length: L; Number of shapelets: K; Regularization parameter: λ; Learning rate: η; Number of iterations: M

Output: Shapelets: $\mathbf{S}$; classifier weights: $\mathbf{w}$

1: Initialize $\mathbf{S}$ and $\mathbf{w}$.
2: **for** $m=1,2,\cdots,M$ **do**
3: Set β according to Section 4.1.
4: **for** $i=1,2,\cdots,I$ **do**
5: $s_{k,l} \leftarrow s_{k,l} - \eta \frac{\partial G_i}{\partial s_{k,l}}$ for $l=1,\cdots,L$ and $k=1,\cdots,K$.
6: $w_k \leftarrow w_k - \eta \frac{\partial G_i}{\partial w_k}$ for $k=0,1,\cdots,K$.
7: **end for**
8: **end for**
9: **return** $\mathbf{S}$,$\mathbf{w}$

$$\frac{\hat{y}_i}{\partial x_{i,k}} = \frac{w_k}{\|\mathbf{w}_{1:K}\|_2}, \quad \frac{\partial \Omega_i}{\partial x_{i,k}} = h(-y_i w_k),$$
$$\frac{\partial \hat{y}_i}{\partial w_k} = \begin{cases} \sum_{k'=1}^{K} \left(\frac{x_{i,k'}}{\|\mathbf{w}_{1:K}\|_2} - \frac{x_{i,k'} w_k w_{k'}}{\|\mathbf{w}_{1:K}\|_2^3} \right) & \text{if } k \neq 0, \\ 1 & \text{if } k = 0. \end{cases} \quad (21)$$

残りの構成要素は文献 [5, 13] と同様である．特に文献 [5, 12, 13] と同様に，劣勾配を介して勾配 $\partial x_{i,k}/\partial s_{k,l}$ を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．

4.3 アルゴリズム

Algorithm 1 は LTSED の疑似コードを表す．1 行目の変数の初期化では，文献 [5] の Algorithm 1 を用いて，教師あり特徴選択により分類に寄与する部分時系列で shapelets を初期化する．3 行目で 4.1 節で述べたようにスケールパラメータ β を設定し，5–6 行目で SGD アルゴリズムを用いて shapelets $\mathbf{S}$ と分類重み $\mathbf{w}$ を更新する．アルゴリズムの計算量は，文献 [5] の Algorithm 1 の計算量を R とすると $O(IQM+R)$ となる．

5 性能評価

5.1 実験設定

提案手法 **LTSED** を次のベースラインと比較する．ベースラインの最初の 4 つは LTSED を一部変更した手法である．**NoSReg** は shapelets 正則化を用いない（つまり，$\lambda=0$ と設定する）．**HardReg** は式 (10) の代わりに式 (9) の不連続なステップ関数を用いる．**L1Reg** と **L2Reg** は shapelets 正則化の代わりに標準的な ℓ_1 と ℓ_2 正則化をそれぞれ用いる．**LTS** [2] は良く知られた shapelets 学習法であり高い正解率を達成する [2, 21, 22]．**CSLTS** [4] はパラメータを自動調整するコスト考慮型のシグモイド損失関数を導入することで 2 クラス間の不均衡データに対して高い F 値を達成する．**LTSSFS** [5] は教師有りの特徴選択により良質な初期 shapelets を獲得し self-paced learning で学習することで shapelets が少数であっても高い AUC を達成する．**SFS1st** は文献 [5] の Algorithm 1 を用いて探索ベースで shapelets を獲得する．

時系列分類で標準的な UCR データセット [23] 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 種類あり，文献 [5, 13] と同じくそれら 30 種類のデータセットを用いる．更に，45 種類のマルチクラスのデータセットに対して最小の 2 クラスを取り出すことで，合計 75 種類の UCR データセットを使用する．30 種類のデータセットと 45 種類のデータセットをそれぞれ **D1** と **D2** と記述する．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 10 回繰り返す．CSLTS では，正クラスの見逃しに対してより厳しいペナルティを課すため，小さいクラスを正とし大きいクラスを負とする．Shapelet の長さ L を $0.2\times Q$ とし，Shapelets の個数 K を 2 とし，勾配降下法の繰り返し回数 M を 600 とする．LTSED と HardReg と L1Reg と L2Reg とにおいて shapelets に対する正則化パラメータ λ は 1 とし，従来手法では分類重みに対する ℓ_2 正則化パラメータ α は 1 とする．各手法に対して次のハイパーパラメータは学習データで AUC が最高となる組み合わせを選ぶ．学習率 η は $\{0.01,0.1,1\}$ の中から選択する．CSLTS では追加のハイパーパラメータがあり $\theta \in \{1,100\}$ と $D \in \{0.1,10\}$ の組み合わせから選択する．AUC は分類器の閾値に依存せず均衡データ及び不均衡データのいずれでも適切な分類性能指標である．文献 [5, 13] と同様に以降では AUC で分類性能を評価する．

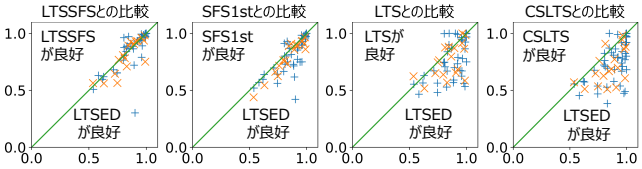

図 6　LTSED と従来手法とにおける AUC 値の比較.

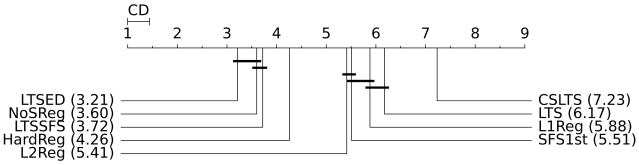

図 7　75 種類のデータセットにおける AUC の CD ダイアグラム.

5.2 分類性能の評価

図 6 の横軸と縦軸は，それぞれ LTSED と従来手法のテストデータにおける AUC の値をプロットしている．D1 と D2 のデータセットにおける結果は，橙色の “×” と青色の “+” とにそれぞれ対応する．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 LTSED が従来手法よりも性能が良いことを意味する．第一観として，LTSED が従来手法よりも優れていることが分かる．

図 7 は 75 種類のテストデータセットに対する Critical Difference (CD) ダイアグラム [24] である．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．提案手法（LTSED と NoSReg）が最も優れており，どの従来手法よりも LTSED の性能が有意に良いことが分かる．

図 8 (左) は D1 の 30 種類のテストデータセットに対する CD

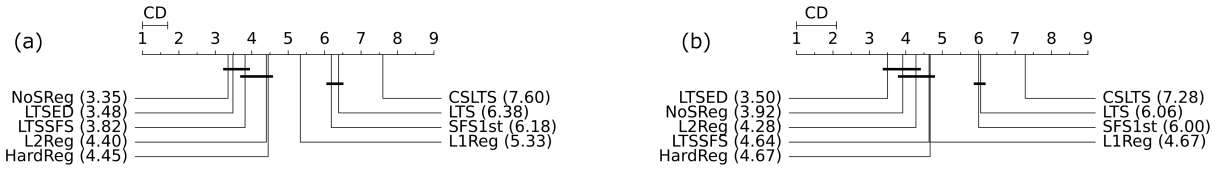

図 8　D1 データセットにおける AUC の CD ダイアグラム（左：30 種類，右：18 種類）

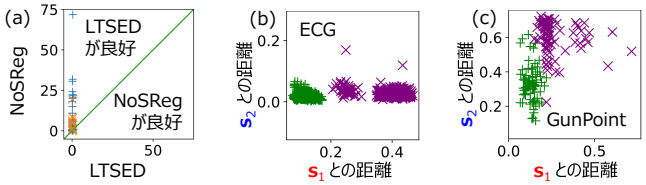

図 9　(a) 式 (22) の SD 値．(b)(c) ECG と GunPoint の特徴空間．

ダイアグラムである．学習データが少数の場合，分類器は過学習する傾向があり，クラス間の分離性を高めて汎化能力を向上させることがより有効と考えられる．これを確認するために，更に学習データのインスタンス数が 100 以下になるように 18 種類のデータセットを選択し，図 8 (右) にその結果を示す．期待通りこれにより，LTSED は特に学習データが少数の場合に従来手法よりも性能が有意に良くなることが分かる．

最後に，本研究で提案するアプローチの有効性を図 7 と図 8 をとおして評価する．LTSED と NoSReg が SFS1st よりも有意に良いことから，shapelets を探索するだけでなく学習することが AUC の向上に有効であることが分かる．また，提案する shapelets 正則化は，不連続な定式化や標準的な ℓ_1 や ℓ_2 正則化（HardReg, L1Reg, L2Reg）とは異なり，分類性能を悪化させないことも分かる．Shapelets 正則化により，shapelets が元の時系列データから乖離してしまうことを抑制できるかについては 5.3 節と 6 章で評価する．

5.3　Shapelets の時系列データからの乖離度合いの評価

各 $k=1,2,\cdots,K$ 及び $i=1,2,\cdots,I$ に対して，式 (1) の $x_{i,k}$ は時系列インスタンス $\mathbf{t}_i$ と shapelet $\mathbf{s}_k$ の距離であり，$\mathbf{s}_k$ のクラスは性質 1 に基づいて定義されることに注意する．Shapelets 正則化が元の時系列データからの shapelets の乖離をどの程度抑制できるかを定量的に測るため，shapelets とそれらが属するクラスの時系列データとの平均的な距離 SD を用いる．

$$\mathrm{SD}(\mathbf{w},\mathbf{X},\mathbf{y}) := \frac{\sum_{k=1}^{K}\sum_{i=1}^{I} h(-y_i w_k) x_{i,k}}{\sum_{k=1}^{K}\sum_{i=1}^{I} h(-y_i w_k)}. \qquad (22)$$

ここで，$h(\cdot)$ は式 (9) の不連続なステップ関数である．図 9 (a) は，D1 と D2 のテストデータセットにおける SD の値をプロットしている．LTSED による SD の値は NoSReg の場合よりもはるかに小さいことから，LTSED では shapelets 正則化により shapelets の時系列データからの乖離を抑制できていることが分かる．この結果は 3.2 節で期待される効果と一致する．

6　ケーススタディ評価

本章では UCR データセット [23] の中で特に正解となる shapelets が良く知られている 2 種類のデータセットに対して，ケーススタディをとおして LTSED における shapelets 正則化を評価する．実験設定は 5.1 節と同じである．Shapelet $\mathbf{s}_k$ の分類への寄与度を分類重み w_k の絶対値の大きさで測定し，一般性を損なうことなく shapelets のインデックスは分類重みの絶対値の降順にソートする（つまり，$|w_1| \geq |w_2|$）．また，性質 1 に基づいて shapelets のクラスを決定する．

6.1　ECGFiveDays データセット

この心電図のデータセットにおいて，医学的に有意な差は T 波に現れることが知られている．図 9 (b) は，LTSED の 2 個の shapelets に基づく 2 次元特徴空間である．各軸は式 (1) で測る距離である．以降では，正と負のクラスにおけるテストインスタンスをそれぞれ緑色の “+” と紫色の “×” としてプロットする．これより，LTSED はテストデータにおける特徴量をクラス間で明確に分離できていることが分かる．

図 10 (a) と (b) では，それぞれ正と負のクラスにおける時系列インスタンスを黒線で描き，それらにベストマッチングする位置に LTSED で学習した shapelets を重ね描きしている．以降では，$\mathbf{s}_1$ と $\mathbf{s}_2$ とをそれぞれ赤太線と青太線で描く．比較のため図 10 (c) には，同じ時系列インスタンスに対してベストマッチングする位置に NoSReg で学習した shapelets を重ね描きしている．図 10 (b) と (c) を比較することで，LTSED と NoSReg のどちらも shapelets で T 波を捉えてはいるが，LTSED の方は shapelets の時系列データからの乖離を抑制できており shapelets 正則化の効果を確認できる．従来の shapelets 学習法 [14, 25] は冗長な多数の shapelets を必要としたが，LTSED はたった 2 つの shapelets で T 波を発見できることも分かる．

6.2　GunPoint データセット

この 2003 年に公開されたデータセットでは，人が銃を向ける動作と指を指す動作とをそれぞれ正と負のクラスとして分類する．図 9 (c) は，LTSED の 2 次元特徴空間であり，各軸は式 (1) の距離である．図 10 (d) と (e) は，それぞれ正と負のクラスにおける時系列インスタンスに対してそれらにベストマッチングする位置に LTSED で学習した shapelets を重ね描きしている．比較のため図 10 (f) には，同じ時系列インスタンスに対してベストマッチングする位置に NoSReg で学習した shapelets を重ね描きしている．図 10 (e) と (f) を比較することで，shapelets 正規化が shapelets の時系列データからの乖離を抑制していることが分かる．このデータセットでは「ホルスターから銃を抜く」部分と「ホルスターに銃を戻す」部分に 2 クラス間の違いが現れることが知られている [3]．図 10 (d) と (e) より，LTSED はこれらの部

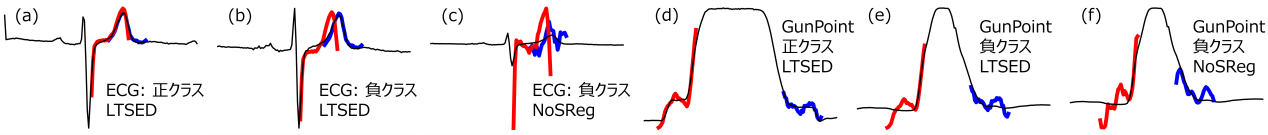

図 10　時系列インスタンス（黒線）及び，shapelets $\mathbf{s}_1$（赤太線）と $\mathbf{s}_2$（青太線）.

分を過不足なく 2 つの shapelets として発見できることも分かる.

7 ま と め

本研究では，shapelet に基づく特徴量のクラス分離性を強化するように学習する LTSED を提案した．文献 [10] に着想を得て，LTSED では SGD による勾配更新が積極的に行われるようにシグモイド損失関数のスケールパラメータを動的かつ自動的に調整する．Shapelets 正則化では，元の時系列データからの shapelets の乖離を抑えつつ，理論的にクラス分離性を向上させる．UCR データセットを用いて，shapelets の個数が 2 個の場合に AUC と shapelets の説明性との観点で提案手法の有効性を確認した.

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

[6] Q. Zhang, J. Wu, P. Zhang, G. Long, and C. Zhang. “Salient Subsequence Learning for Time Series Clustering”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* (2018).

[7] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. “The Implicit Bias of Gradient Descent on Separable Data”. In: *J. Mach. Learn. Res.* (2018).

[8] B. Chen, W. Deng, and H. Shen. “Virtual Class Enhanced Discriminative Embedding Learning”. In: *NeurIPS*. Curran Associates Inc., 2018.

[9] W. Liu, Y. Wen, Z. Yu, and M. Yang. “Large-Margin Softmax Loss for Convolutional Neural Networks”. In: *ICML*. JMLR.org, 2016.

[10] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li. “AdaCos: Adaptively Scaling Cosine Logits for Effectively Learning Deep Face Representations”. In: *CVPR*. IEEE Computer Society, 2019.

[11] A. Yamaguchi, K. Ueno, and H. Kashima. “Learning Time-series Shapelets Enhancing Discriminability”. In: *SDM*. SIAM, 2022.

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

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

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

[15] T. Salimans and D. P. Kingma. “Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks”. In: *NeurIPS*. Curran Associates Inc., 2016.

[16] H. Hazimeh, N. Ponomareva, P. Mol, Z. Tan, and R. Mazumder. “The Tree Ensemble Layer: Differentiability meets Conditional Computation”. In: *ICML*. PMLR, 2020.

[17] F. Schroff, D. Kalenichenko, and J. Philbin. “FaceNet: A unified embedding for face recognition and clustering”. In: *CVPR*. IEEE Computer Society, 2015.

[18] R. Hadsell, S. Chopra, and Y. LeCun. “Dimensionality Reduction by Learning an Invariant Mapping”. In: *CVPR*. IEEE Computer Society, 2006.

[19] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. “ArcFace: Additive Angular Margin Loss for Deep Face Recognition”. In: *CVPR*. IEEE Computer Society, 2019.

[20] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, et al. “CosFace: Large Margin Cosine Loss for Deep Face Recognition”. In: *CVPR*. IEEE Computer Society, 2018.

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

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

[23] 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.

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

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