時系列データの shapelets を学習する partial AUC の最大化手法

山口 晃広† 真矢 滋† 丸地 康平† 植野 研†

† 株式会社東芝 研究開発センター システム技術ラボラトリー 〒 212-8582 川崎市幸区小向東芝町 1

E-mail: †akihiro5.yamaguchi@toshiba.co.jp

あらまし IoT の活用に向けて時系列データにおけるクラス分類の研究が進められている．特に近年，分類器を学習しながら shapelets と呼ばれる分類に有効な波形パターンを同時に学習する手法が研究されており，分類性能が高く説明性のある手法として近年注目を集めている．一方，医療・製造・保守などの産業分野では，False Positive Rate（FPR）を低い範囲に絞って Area Under the ROC curve（AUC）を測定する partial AUC（pAUC）が実用上重要な性能指標となる場合が多い．本論文では，full AUC も含めて任意の FPR の範囲における pAUC を最大化するように shapelets と分類器とを同時に学習する手法を提案する．実験では，UCR 時系列データセットを用いて pAUC の性能向上を示すとともに，各産業分野への適用事例でも有効性を示す．

キーワード 時系列データ，Partial AUC，Shapelet

1 は じ め に

IoT の活用に向けて，時系列データを機械学習により分類する時系列クラス分類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正例または負例として与えられて，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クラス分類では，時間軸上での順序関係や波形パターンの形状が重要であり，時系列データ上のノイズや波形パターンの時間軸上のシフトなど，時系列データの特徴を扱う必要がある．

Shapelet とはクラス分類に有効な波形パターンである．時系列クラス分類では，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 [1, 2]．これらの shapelets 手法では，分類に有効な特徴は時系列全体ではなく少数の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，分類に直接寄与す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・保守などの分野で受け入れられやすい．

初期に提案された shapelets 手法では時系列データの中から部分時系列を探索することで shaplets を発見していたが [2]，近年 shapelets を学習する手法が提案されている [1, 3]．このような shapelets 学習手法では，勾配降下法により shapelets と分類器の両方を同時に学習する．これにより，計算量を削減し [1]，時系列データのノイズにロバストな shapelets を発見し [4, 5]，正解率や F 値による分類性能も向上する [1, 3]．

Receiver Operating Characteristic（ROC）曲線は，データマイニングや機械学習の評価において重要な役割を果たす．特に，ROC 曲線下の面積に相当する Area Under the ROC curve（AUC）は，クラス分類の閾値に依存せず，クラス間のサンプル数に偏りがある場合でも適切に評価できるため，2 クラス分類からランキングまで幅広く性能測定に用いられている．

多くの実応用では，AUC における False Positive Rate（FPR）を特定の範囲に制限した partial AUC（pAUC）に興味がある．特に，医療・製造・保守などの産業分野では，FPR の範囲を低く抑えたもとで pAUC を最大化することが重要である．そのため，pAUC を最適化する手法が提案されているが [6, 7]，時系列データの特徴を扱えない．しかし，このような産業応用では下例のように時系列データに対して pAUC を向上する技術が求められる．

- 深刻な心臓病や癌などの医療診断では，見逃しは患者の生命に関わるため見逃し率を低い範囲に抑えたうえで正しく診断する必要がある [8, 9]．一方，心電図の時系列データを用いた医療診断は，患者の体を傷つけず安価でもあるため多くの患者へ適用できる．そのため，心電図の時系列データに対して pAUC を向上する技術が求められる．
- 半導体製造工場に異常検知技術を導入する場合，誤報の発生が製造装置の不要なダウンタイムを発生させるだけでなく作業員の意識も低下させることから，誤報率を許容範囲に抑えて検知率を向上することが必要となる [10]．一方，製造工程は時系列として表され，製造装置にセンサを取りつけて時系列データを分析する機会も増えている [11, 12]．そのため，センサ時系列データに対して pAUC を向上する技術が求められる．
- 異常予兆を検知することで修理の要否を判定する予知保全では，保守をする際には保守員が現地へ赴き機器を分解するなどのコストが発生するので，誤報率を低く抑えたもとで検知率を向上する必要がある．一方，このようなプロアクティブな保守を安価なセンサデバイスの取り付けで実現できれば，製品やサービスの価値を向上できる．そのため，安価なセンサから得られる時系列データに対して pAUC を向上する技術が求められる．

そこで，我々は pAUC を最適化する shapelets と分類器の両方を同時に学習する手法 “LTSpAUC” を提案する．pAUC を最適化する場合としない場合で学習される shapelets や分類器の違いについて，以下の簡単な例を用いて説明する．

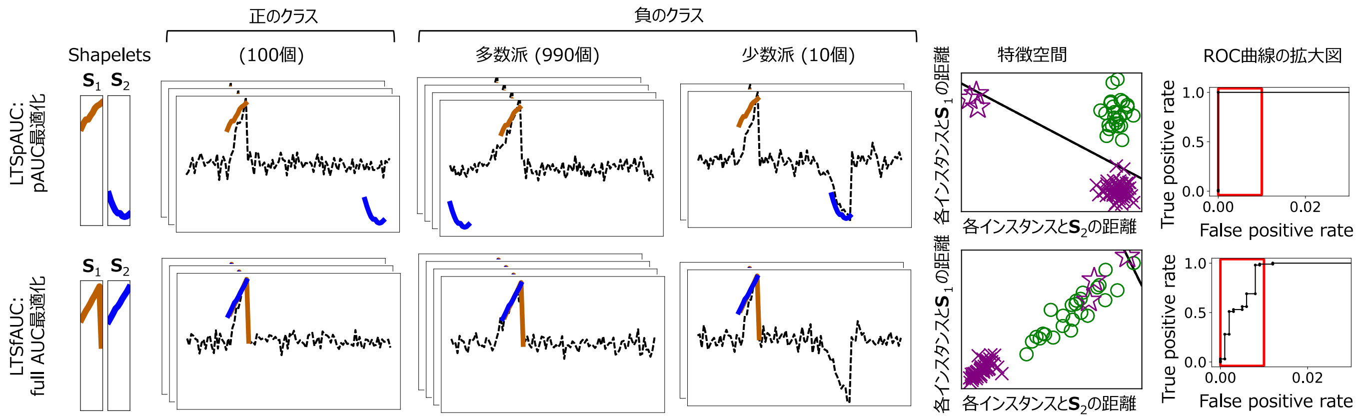

図 1　pAUC を最適化する LTSpAUC（上段）と full AUC を最適化する LTSfAUC（下段）の違い．LTSpAUC では負例の少数派を唯一区別できる下に尖ったピーク波形を shapelet として発見できるため，特徴空間でも負例の少数派と正例を分類でき，pAUC を向上する．

1.1　提案するアプローチの有効性を示す例

100 個の正例と 990 個の負例の多数派と 10 個の負例の少数派からなるデータセットに対して，時系列インスタンスを図 1 の破線で示す．負例の多数派では正例よりも上に尖ったピークが緩やかなので，上に尖ったピーク波形に着目することで，正例と負例の殆どを正しく分類できる．一方，負例の少数派では上に尖ったピークは正例のものと同じだが，下に尖ったピークは負例の少数派にのみ表れる．つまり，下に尖ったピーク波形が負例の少数派と正例とを唯一区別できる部分時系列である．

この例では，FPR を 0.01 以下に抑えたもとで pAUC を最適化する shapelets と分類器を学習することを目的とする．full AUC（つまり FPR [0,1] の範囲での pAUC）を最適化する比較手法 “LTSfAUC” と，FPR [0,0.01] の範囲で pAUC を最適化する LTSpAUC とを比較する．図 1 の 1 列目に示すように，LTSpAUC（図上段）と LTSfAUC（図下段）はそれぞれ 2 つの shapelets を学習した．図 1 の 2–4 列目では，各手法に対して時系列データ上で最もマッチングする位置に shapelets を太線で示している．以上の結果から，LTSpAUC だけが負例の少数派を区別できる shapelet（つまり下に尖ったピーク波形パターン）を学習できていることが分かる．

図 1 の 5 列目では，各時系列インスタンスを 2 つの shapelets への距離 [1] から算出される 2 次元特徴ベクトルとして特徴空間へマッピングしている．正例，負例の多数派，負例の少数派をそれぞれ “○”, “×”, and “☆” で幾つかプロットしている．各黒線の分類境界は FPR=0 を満たす最も低い閾値で引かれている．特徴空間での以上の結果から，LTSfAUC では正例と負例の少数派を区別できないが，LTSpAUC では負例の多数派だけでなく負例の少数派も正例と正しく区別できていることが分かる．

図 1 の 6 列目は，各手法による ROC 曲線の一部をそれぞれ表し，赤枠は FPR が 0.01 以下の範囲を表す．負例の少数派を誤分類しても full AUC としては無視できるほどであり，各手法の full AUC はいずれも 1.0 である．一方，FPR を [0,0.01] の範囲に絞った pAUC としては，LTSpAUC と LTSfAUC とで 1.0 と 0.58 となり大きく性能差が生じる．

LTSfAUC と同様に，pAUC を最適化しない従来の shapelets 手法 [1, 2, 3] でも，負例の少数派を正例と区別する shapelets を発見できない．その結果，pAUC の性能も劣化する．それゆえ，この単純な例では pAUC を最適化する shapelets の発見が pAUC の性能向上に有効であることが分かった．

1.2　本論文の主な貢献

本論文の主な貢献を以下に示す．なお，本内容は基本的に [13] に準じ，証明などに関しては [13] を参照いただきたい．

- Full AUC も含めて，任意の FPR の範囲で pAUC を最適化するように shapelets と分類器の両方を同時に学習する手法 LTSpAUC を提案する．
- 従来の shapelets 手法 [1, 2, 3, 4, 5] では計算量は時系列長の 2 乗以上のオーダーだが，文献 [14] と同様に min 関数の劣微分を用いて計算量を時系列長の線形オーダーにする．
- Shapelets の幾つかを正または負クラスの時系列データへ明示的に近づきやすくするように，制約式を導入して数理最適化モデルを拡張する．また，それらの制約式を導入しても射影劣勾配降下法で効率的に解けることを理論的に示す．
- 負例の数が多い UCR 時系列データセットを用いて，最新の shapelets 学習手法 [1, 3] と比べて，FPR の範囲を低く抑えた pAUC が有意に向上することを示す．更に，医療・製造・保守の各産業分野への適用事例でも有効性を示す．

2　関 連 研 究

2.1　Shapelets 学習手法

Learning Time-series Shapelets（LTS） [1] として知られる時系列クラス分類手法では，shapelets と分類器の両方を確率的勾配降下法（SGD）により同時に学習する．LTS は様々な時系列データセットで高い正解率を達成し [1, 15, 16]，これまで様々な拡張が研究されている．Shapelets が時系列インスタンス上に現れる位置を学習することで，正解率の低下を抑えながら学習を高速化する拡張手法が提案された [15]．また，教師無

1：時系列インスタンスと shapelet との距離は 3.1 節で定義する．

しの時系列クラスタリングへの拡張手法も提案された [4, 5].

しかしながら，full AUC や pAUC を最適化する shapelets 手法は未だ研究されていない．他のアプローチとして，正解率に基づく損失関数を自動調整することで正例と負例の数に偏りのある時系列クラス分類に対応する Cost-Sensitive Learning Time-series Shapelets（CSLTS） [3] が提案された．しかし，CSLTS は full AUC や pAUC を最適化せず，文献 [3] でも F 値のみが評価されている．

2.2 pAUC 最適化手法

任意の FPR の範囲における pAUC を最適化する手法が提案されている．文献 [7] では，文献 [6] から数理最適化モデルを改善し，産業分野の適用事例で pAUC の有効性を示した．また，この手法 [7] はコンピュータビジョンの分野にも適用されアンサンブル学習に拡張された [17]．pAUC のようにインスタンス単位に分解できない性能指標を最適化する手法も幾つか提案されている．オンライン学習の問題設定に合わせた手法が提案され [18]，ユーザが pAUC を近似するためのアンカーポイントを指定しなければならないがインスタンスの数にスケールする手法が提案された [19].

しかしながら，時系列データを利用する機会が増えているにも関わらず，これらの従来手法は我々の知る限り時系列データを扱えない．それゆえ，我々は pAUC の最大化と shapelets の学習とを統合する手法を開発する．

3 準　　備

3.1 表記や定義

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．全インスタンスと正例と負例の数をそれぞれ I，P，N（つまり $I=P+N$）とする．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 $(\mathbf{T}_{i,j},\mathbf{T}_{i,j+1},\cdots,\mathbf{T}_{i,j+L-1})$ 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．

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

文献 [1, 3] と同様に，式 (1) を用いて 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．

FPR の範囲 $[\alpha,\beta]$ が与えられて，$N_\alpha=\lceil\alpha N\rceil+1$ とし $N_\beta=\lfloor\beta N\rfloor$ とする．また，全インスタンスの K 次元特徴ベクトルの集合と，そのうちの正例と負例の特徴ベクトルの集合をそれぞれ $\mathbf{X}\in\mathbb{R}^{I\times K}$ と，$\mathbf{X}^+\in\mathbb{R}^{P\times K}$ と $\mathbf{X}^-\in\mathbb{R}^{N\times K}$ とする．分類器の K 次元重みベクトル $\mathbf{w}$ と K 次元特徴ベクトル $\mathbf{x}$ が与えられて，次のスコア関数を考える．

$$f(\mathbf{x}):=\langle\mathbf{w},\mathbf{x}\rangle=\sum_{k=1}^{K}\mathbf{w}_k\mathbf{x}_k, \quad (2)$$

ここで，$\mathbf{x}$ が正のクラスに属する場合に f は高い値を持つ．スコアに同点が無いと仮定すると，FPR $[\alpha,\beta]$ の範囲におけるスコア関数 f による pAUC は次のように測られる [2].

$$\frac{1}{c}\sum_{p=1}^{P}\sum_{n=N_\alpha}^{N_\beta}\mathbb{I}\Big(f(\mathbf{X}_p^+)>f(\mathbf{X}_{(n)_f}^-)\Big), \quad (3)$$

ここで，定数 c は $(\beta-\alpha)PN$ であり，$\mathbb{I}$ は引数が真の場合に 1 で偽の場合に 0 となる indicator 関数であり，$(n)_f$ はスコア関数 f の値の降順に N 個の負例を並べた際に n 番目のインスタンスを指し示す番号である．

本論文の目的は，式 (3) を最大化する shapelets $\mathbf{S}$ と重みベクトル $\mathbf{w}$ を学習することである．

3.2 pAUC 最適化手法の概要

本節では，LTSpAUC の定式化の一部で利用する文献 [7] の定式化を簡単に説明する．我々のアプローチとは異なり，正例の特徴ベクトルの集合 $\mathbf{X}^+$ と負例の特徴ベクトルの集合 $\mathbf{X}^-$ はそれぞれ学習する前に固定で与えられる．

要素数が N_β となる $\mathbf{X}^-$ の任意の部分集合を $\mathbf{Z}$ として，$n=1,2,\cdots,N_\beta$ に対して $\mathbf{Z}$ の n 番目の特徴ベクトルを $\mathbf{Z}_n$ と記述する．$\mathbf{X}^+$ と $\mathbf{Z}$ の相対的な順序関係を表す行列 $\mathbf{O}\in\{0,1\}^{P\times N_\beta}$ を次のように定義する．

$$\mathbf{O}_{p,n}=\begin{cases}1 & \text{if } \mathbf{X}_p^+ \text{ is ranked below } \mathbf{Z}_n,\\ 0 & \text{if } \mathbf{X}_p^+ \text{ is ranked above } \mathbf{Z}_n.\end{cases} \quad (4)$$

反対称性や推移性を満たすような妥当な順序関係を表す行列 $\mathbf{O}$ の全ての集合を $\mathcal{O}$ と記述する．明らかに正しい順序関係を表す $\mathbf{O}^*_{p,n}$ は全てのペア (p,n) に対して $\mathbf{O}^*_{p,n}=0$ となる．この行列 $\mathbf{O}\in\mathcal{O}$ を用いて，次のように pAUC の損失を定義できる．

$$\Delta(\mathbf{O}^*,\mathbf{O})=\frac{1}{c}\sum_{p=1}^{P}\sum_{n=N_\alpha}^{N_\beta}\mathbf{O}_{p,(n)_\mathbf{O}}, \quad (5)$$

ここで，$(n)_\mathbf{O}$ は行列 $\mathbf{O}$ の順序関係を満たすように負例を並べた際に n 番目のインスタンスを指し示す番号である．

インスタンス単位に分解できない損失関数を扱う従来手法 [20] のアイデアに基づき，負例の特徴ベクトルの集合を部分集合 $\mathbf{Z}$ に制限した際に特徴ベクトルの集合 $\mathbf{X}$ と行列 $\mathbf{O}\in\mathcal{O}$ のマッピングを次のように定義する．

$$\phi_\mathbf{Z}(\mathbf{X},\mathbf{O})=\frac{1}{c}\sum_{p=1}^{P}\sum_{n=N_\alpha}^{N_\beta}(1-\mathbf{O}_{p,n})\big(\mathbf{X}_p^+-\mathbf{Z}_n\big). \quad (6)$$

式 (6) の差を次のように簡単に記述する．

$$\Phi_\mathbf{Z}(\mathbf{X},\mathbf{O})=\phi_\mathbf{Z}(\mathbf{X},\mathbf{O}^*)-\phi_\mathbf{Z}(\mathbf{X},\mathbf{O}). \quad (7)$$

固定された特徴ベクトル $\mathbf{X}$ に対する pAUC の最適化は次のように定式化 [3] できる [7].

2：αN や βN が整数でない一般の定義は文献 [6] の 2 章を参照いただきたい．

3：式 (8) の詳細な導出は文献 [7] の 5.2 節を参照いただきたい．

$$\begin{aligned}\underset{\mathbf{w}}{\text{minimize}} \quad & \max_{\mathbf{Z}\in\mathcal{Z},\mathbf{O}\in\mathcal{O}} \Delta(\mathbf{O}^*,\mathbf{O}) - \langle \mathbf{w},\Phi_{\mathbf{Z}}(\mathbf{X},\mathbf{O})\rangle \\ \text{s.t.} \quad & \|\mathbf{w}\|_2 \leqq \lambda,\end{aligned} \tag{8}$$

ここで，$\mathcal{Z}$ は $\mathbf{Z}$ の集合であり，λ は正則化パラメータである．

4 提案手法 LTSpAUC

4.1 数理最適化モデルの定式化

LTSpAUC は，式 (1) の shapelets に基づく特徴ベクトルを式 (8) へ組み込むことで，次のように自然に定式化される．

$$\begin{aligned}\underset{\mathbf{S},\mathbf{w}}{\text{minimize}} \quad & F(\mathbf{S},\mathbf{w}), \\ & F(\mathbf{S},\mathbf{w}) = \max_{\mathbf{Z}\in\mathcal{Z},\mathbf{O}\in\mathcal{O}} H(\mathbf{X},\mathbf{Z},\mathbf{O};\mathbf{w}), \\ & H(\mathbf{X},\mathbf{Z},\mathbf{O};\mathbf{w}) = \Delta(\mathbf{O}^*,\mathbf{O}) - \langle \mathbf{w},\Phi_{\mathbf{Z}}(\mathbf{X},\mathbf{O})\rangle, \\ & \mathbf{X}_{i,k} = \min_{j=1,2,\cdots,J} \frac{1}{L}\sum_{l=1}^{L}(\mathbf{T}_{i,j+l-1}-\mathbf{S}_{k,l})^2, \\ & \qquad \text{for } 1\leqq i\leqq I, 1\leqq k\leqq K \\ \text{s.t.} \quad & \|\mathbf{w}\|_2 \leqq \lambda,\end{aligned} \tag{9}$$

ここで，Δ と $\Phi_{\mathbf{Z}}$ はそれぞれ式 (5) と式 (7) で定義され，λ は正則化パラメータである．

Shapelets の発見を補助するため，重みベクトル $\mathbf{w}$ を制約付けることで式 (9) の数理計画モデルを拡張し，shapelets の幾つかを正例または負例の部分時系列に類似しやすくするよう明示的に指定できる．これは例えば特定のクラスにフィッティングする shapelets を発見したいときに役立つ．まず，shapelet $\mathbf{S}_k$ と重みベクトルの要素 $\mathbf{w}_k$ との関係を定理としてまとめる．

定理 1. 任意の時系列インスタンス $\mathbf{T}_i$ と $k=1,2,\cdots,K$ に対して，重みベクトルの要素 $\mathbf{w}_k$ の値に応じて以下の 3 つのケースが成り立つ: (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 (2) のスコアは上昇する; (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 (2) のスコアは減少する; (c) $\mathbf{w}_k=0$ の場合，$\mathbf{S}_k$ は式 (2) のスコアに寄与しない．

LTSpAUC は式 (2) のスコア関数を改善するように shapelets $\mathbf{S}$ と重みベクトル $\mathbf{w}$ の両方を学習する．そのため，$\mathbf{w}_k<0$ または $\mathbf{w}_k>0$ の場合，定理 1 より shapelet $\mathbf{S}_k$ は正例または負例にフィッティングする傾向が強くなる．それゆえ，次の制約式をオプションとして追加できるようにする．

$$\begin{cases}\mathbf{w}_k \leqq 0, \quad 1\leqq k\leqq K^+ & \text{if } K^+\geqq 1, \\ \mathbf{w}_k \geqq 0, \quad K^+<k\leqq K^++K^- & \text{if } K^-\geqq 1,\end{cases} \tag{10}$$

ここで，K^+ と K^- はそれぞれ正例または負例の時系列データにフィッティングしやすいよう制約づけられた shapelets の数である．ユーザは $K^++K^-\leqq K$ を満たすように非負の整数である K^+ と K^- を決定できる．$K^+=0$ または $K^-=0$ の場合，式 (10) における 1 番目と 2 番目の制約をそれぞれ無視する．

我々の今の目的は，オプションの制約式 (10) を持つ数理最適化モデル (9) に対して，shapelets $\mathbf{S}$ と重みベクトル $\mathbf{w}$ の両方を学習することである．我々は，制約式を満たしながら目的関数 $F(\mathbf{S},\mathbf{w})$ の勾配の逆方向に shapelets $\mathbf{S}$ と重みベクトル $\mathbf{w}$ を繰り返し更新することで，この数理最適化モデルを解く．次の 2 つ節では，$\mathbf{S}$ と $\mathbf{w}$ の更新ステップをそれぞれ述べる．

4.2 Shapelets の更新

Shapelets $\mathbf{S}$ に対する目的関数 $F(\mathbf{S},\mathbf{w})$ の勾配は，max 関数の劣微分と微分の連鎖律を用いて次のように計算できる．

$$\begin{aligned}\frac{\partial F(\mathbf{S},\mathbf{w})}{\partial \mathbf{S}_{k,l}} &= \sum_{i=1}^{I}\frac{\partial H(\mathbf{X},\bar{\mathbf{Z}},\bar{\mathbf{O}};\mathbf{w})}{\partial \mathbf{X}_i}\frac{\partial \mathbf{X}_i}{\partial \mathbf{S}_{k,l}}, \\ (\bar{\mathbf{Z}},\bar{\mathbf{O}}) &= \arg\max_{\mathbf{Z}\in\mathcal{Z},\mathbf{O}\in\mathcal{O}} H(\mathbf{X},\mathbf{Z},\mathbf{O};\mathbf{w}).\end{aligned} \tag{11}$$

$\partial H(\mathbf{X},\bar{\mathbf{Z}},\bar{\mathbf{O}};\mathbf{w})/\partial \mathbf{X}_i$ は，Δ が $\mathbf{X}$ に依存しないことに注意して $\langle \mathbf{w},\Phi_{\bar{\mathbf{Z}}}(\mathbf{X},\bar{\mathbf{O}})\rangle$ を微分することで以下のように計算できる．

$$\begin{aligned}\frac{\partial H(\mathbf{X},\bar{\mathbf{Z}},\bar{\mathbf{O}};\mathbf{w})}{\partial \mathbf{X}_i} &= -\frac{\partial \langle \mathbf{w},\Phi_{\bar{\mathbf{Z}}}(\mathbf{X},\bar{\mathbf{O}})\rangle}{\partial \mathbf{X}_i} \\ &= \begin{cases} -\left(\frac{1}{c}\sum_{n=N_\alpha}^{N_\beta}\bar{\mathbf{O}}_{i,n}\right)\mathbf{w} & \text{if } \mathbf{X}_i\in\mathbf{X}^+, \\ \left(\frac{1}{c}\sum_{p=1}^{P}\bar{\mathbf{O}}_{p,i}\right)\mathbf{w} & \text{if } \mathbf{X}_i\in\bar{\mathbf{Z}}, \\ \mathbf{0} & \text{otherwise.}\end{cases}\end{aligned} \tag{12}$$

従来の shapelets 学習手法 [1, 3, 4, 5] の計算量は時系列長の 2 乗のオーダーであった．一方，LTSpAUC では文献 [14] と同様に min 関数を微分可能な関数で近似せず劣勾配を直接計算する．これにより，計算量を時系列長の線形オーダーまで削減する．つまり，$\partial \mathbf{X}_i/\partial \mathbf{S}_{k,l}$ は次のように導出される．

$$\begin{aligned}\frac{\partial \mathbf{X}_i}{\partial \mathbf{S}_{k,l}} &= \frac{2(\mathbf{S}_{k,l}-\mathbf{T}_{i,j^*+l-1})}{L}, \\ j^* &= \arg\min_{j=1,2,\cdots,J}\frac{1}{L}\sum_{l=1}^{L}(\mathbf{T}_{i,j+l-1}-\mathbf{S}_{k,l})^2.\end{aligned} \tag{13}$$

4.3 重みベクトルの更新

まず，重みベクトル $\mathbf{w}$ に対する目的関数 $F(\mathbf{S},\mathbf{w})$ の勾配を計算する．$\mathbf{w}$ は shapelets $\mathbf{S}$ に依存しないので，この勾配は文献 [7] と同様であり次のように計算できる．

$$\frac{\partial F(\mathbf{S},\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial H(\mathbf{X},\bar{\mathbf{Z}},\bar{\mathbf{O}};\mathbf{w})}{\partial \mathbf{w}} = -\Phi_{\bar{\mathbf{Z}}}(\mathbf{X},\bar{\mathbf{O}}). \tag{14}$$

式 (14) の勾配の逆方向に更新した重みベクトルを $\mathbf{v}$ と記述する．

次に，$\mathbf{w}$ の制約を満たすように $\mathbf{v}$ を射影する．この射影は次のように定式化できる．

$$\begin{aligned}\underset{\mathbf{w}}{\text{minimize}} \quad & \|\mathbf{w}-\mathbf{v}\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{w}\|_2 \leqq \lambda \text{ and constraints (10).}\end{aligned} \tag{15}$$

制約式 (10) を取り除くため，次の定理により最適化問題 (15) を書き換える．制限された定理は文献 [21, 22] で既に研究されているが，本論文では我々の問題に合うようにより汎用化した定理を導出する．

定理 2. 最適化問題 (15) に対して，次の最適化問題を考える．

$$\underset{\mathbf{w}}{\text{minimize}}\,\|\mathbf{w}-\tilde{\mathbf{v}}\|_2^2 \qquad \text{s.t.} \qquad \|\mathbf{w}\|_2 \leqq \lambda,$$

$$\tilde{\mathbf{v}}_k = \begin{cases} \min\{0,\mathbf{v}_k\} & \text{if } 1 \leqq k \leqq K^+, \\ \max\{0,\mathbf{v}_k\} & \text{if } K^+ < k \leqq K^+ + K^-, \\ \mathbf{v}_k & \text{if } K^+ + K^- < k \leqq K. \end{cases} \tag{16}$$

問題 (16) を最小化する $\mathbf{w}$ は，問題 (15) の最適解でもある．

定理 2 により，問題 (16) を解くことで問題 (15) の解 $\mathbf{w}$ が得られる．つまり，$\mathbf{v}$ から $\tilde{\mathbf{v}}$ を計算した後で，ℓ_2 球への射影を $\mathbf{w}=\min\{1,\lambda/\|\tilde{\mathbf{v}}\|_2\}\tilde{\mathbf{v}}$ で計算すれば良い．

4.4 アルゴリズム

Algorithm 1 LTSpAUC

Input: 時系列データセット: $\mathbf{T}\in\mathbb{R}^{I\times Q}$; クラスラベル: $\{-1,1\}^I$; FPR の範囲: $[\alpha,\beta]$; Shapelet の長さ: L; Shapelets の数: (K,K^+,K^-); 正則化パラメータ: λ; 学習率: η; 繰り返し回数: M

Output: Shapelets: $\mathbf{S}\in\mathbb{R}^{K\times L}$; 重みベクトル: $\mathbf{w}\in\mathbb{R}^K$

```
1:  Initialize S and set w ← 0.
2:  for m = 1,2,···,M do
3:      Calculate feature vectors X based on S in Eq. (1).
4:      Find (Z̄,Ō) = arg max_{Z∈𝒵,O∈𝒪} H(X,Z,O;w).
5:      Compute v ← w − (η/√m) ∂H(X,Z̄,Ō;w)/∂w.
6:      Compute w from v by solving Eq. (16).
7:      Choose i ∈ {1,2,···,I} at random in FPR [α,β].
8:      for k = 1,2,···,K do
9:          for l = 1,2,···,L do
10:             Compute S_{k,l} ← S_{k,l} − (η/√m) ∂H(X,Z̄,Ō;w)/∂S_{k,l}.
11:         end for
12:     end for
13: end for
14: return S,w
```

LTSpAUC の擬似コードを Algorithm 1 に示す．文献 [1] と同様に 1 行目の shapelets の初期化では，部分時系列に k-means++クラスタリング手法 [23] を適用し，その centroids を初期 shapelets とする．4 行目では，文献 [7] の Algorithm 2 を用いて $(\mathcal{Z},\mathcal{O})$ の組み合わせからペア $(\bar{\mathbf{Z}},\bar{\mathbf{O}})$ を効率的に見つける．5 行目で勾配 (14) の逆方向に $\mathbf{w}$ を更新した後，6 行目で ℓ_2 球への射影により問題 (16) を解く．7 行目で負例のインスタンスを選ぶ場合，FPR $[\alpha,\beta]$ の範囲に含まれる負例を分類するように shapelets を学習することが重要なため，スコア関数 f の値の降順に負例をソートし N_α 番目から N_β 番目までのインスタンスをランダムに選ぶ．10 行目では，式 (11)–(13) を用いて勾配を計算し，その逆方向に $\mathbf{S}$ を更新する．

計算量をまとめる．文献 [1] と同様に K と L は小さいため省略する．各繰り返しで，重みベクトル $\mathbf{w}$ の更新は文献 [7] と同様に $O(N\log N+PN_\beta+I)$ の計算量で，shapelets $\mathbf{S}$ の更新は $O(N\log N+PN_\beta+Q)$ の計算量である．そのため，全体の計算量は $O((N\log N+PN_\beta+I+Q)M)$ となる．4.2 節で述べたように，LTSpAUC は時系列の長さ Q に線形でスケールできる．

5 性能評価実験

5.1 実験設定

時系列クラス分類の標準的なデータセットである UCR 時系列データセット [24] を用いる．FPR の狭い範囲 [0,0.05] における pAUC を測定するため，多くの負例が必要と考えられる．そのため，インスタンスの数が最少のクラスを正クラスとして残りのクラスをまとめて負クラスとする [4]．本実験では 500 個以上の負例がある 39 個のデータセットを使用する．

提案手法 **LTSpAUC** を次の最新の shapelets 学習手法と比較する．**LTS** [1] は，他の時系列クラス分類手法と比べて高い正解率を達成する [1, 15, 16]．**CSLTS** [3] は，コスト考慮型学習を導入することで LTS を拡張した手法であり，正例と負例でコストが異なるとき高い F 値を達成する．**LTSfAUC** は，LTSpAUC を修正して pAUC の代わりに full AUC を最適化する．

各データセットでインスタンスをシャッフルしながら 10-fold の交差検証を 5 回繰り返す．各 fold では学習データの 10%をメタパラメータのチューニング用に取っておく．各手法は各 fold に対してチューニング用データで FPR [0,0.05] の pAUC が最高となるメタパラメータを採用する．メタパラメータである shapelets の数を $K\in\{20,40,60,80,100\}$ で正則化パラメータを $\lambda\in\{0.01,1,100\}$ でそれぞれチューニングする．CSLTS では追加のメタパラメータがあり $\theta\in\{1,10,100\}$ と $D\in\{0.1,10\}$ でチューニングする [5]．全手法で，勾配降下法の繰り返し回数 $M=1000$ と，shapelets の長さ $L=0.1\times Q$ と設定する．特に明記しない限り，式 (1) の距離計算では z-normalization を適用し，LTSpAUC と LTSfAUC ではオプションの制約式 (10) を使用しない．

5.2 実験結果

各手法に対して，テスト用データを用いて Critical Difference (CD) diagrams [25] により pAUC と full AUC を比較する．図 2 (a) は，39 個のデータセットに対する full AUC における CD diagram を表す．図の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．図 2 (a) より，LTSfAUC と LTS の間には有意差はないが LTSfAUC の性能が最良であることが分かる．

図 2 (b) は，39 個のデータセットに対する pAUC の結果を表す．LTSpAUC の性能が最良であるが，全ての手法間で有意な差は見られない．推測される原因としては，負例が多い 39 個のデータセットを選んだが FPR を狭い範囲に絞ることで負例にまだ過学習している可能性が考えられる

これを確認するため，負例が 1000 個を越える 17 個のデータセットに絞り込み，その結果を図 2 (c) に示す．負例の多いデータセットに絞り込むことで LTSpAUC の rank が良くなる

4：最少のクラスのインスタンスが 50 個よりも少ない場合，正クラスのインスタンスが 50 個以上になるまで次に小さいクラスを正クラスに順次追加していく．
5：CSLTS は正例のペナルティを重くするように設計されているので [3]，本論文では正例と負例の役割を入れ替えて実験する．

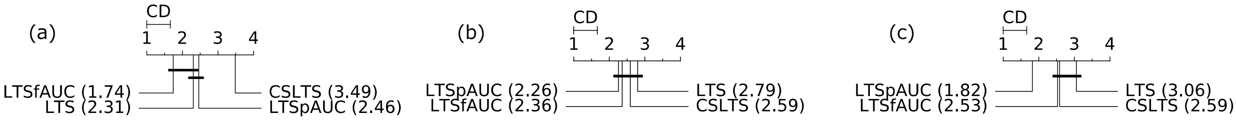

図 2　テスト用データによる critical difference diagrams: (a) 全 39 データセットでの full AUC, (b) 全 39 データセットでの pAUC, (c) 負例の多い 17 データセットでの pAUC

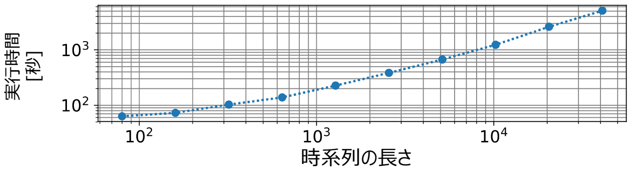

図 3　時系列長におけるスケーラビリティ

ことが分かる．その結果，LTSpAUC は他の比較手法よりも有意に性能が高いことも分かる．これは，FPR を狭い範囲に絞った pAUC を最適化する場合には，過学習を避けるため多くの負例が理想的には必要であることを意味している．

5.3　時系列長におけるスケーラビリティ

図 3 は，図 1 の人工時系列データの長さを変えながら，LTSpAUC（Algorithm 1）の実行時間を測定した結果である．図 3 で時系列が長い場合，実行時間は時系列長にほぼ線形にスケールしている．これにより，実験結果が 4.4 節の計算量の解析結果と一致することが分かる．

6　産業分野への適用事例

時系列データで pAUC の向上が必要とされる医療・製造・保守の 3 分野における適用事例を示す．5.2 節で示したように pAUC を最適化しない手法の中では LTSfAUC が pAUC でも full AUC でも最良であったため，LTSfAUC を LTSpAUC と比較する．

$k=1,2,\cdots,K$ に対して，shapelet $\mathbf{S}_k$ が負例の少数派を区別するのにどの程度特化しているかを次の式で測る．

$$\mathrm{CB}(\mathbf{S}_k)=\frac{\text{mean of }\left\{\left|\mathbf{w}_k\tilde{\mathbf{X}}^-_{\tilde{n},k}\right|\right\}_{\tilde{n}\in[1,|\tilde{\mathbf{T}}^-|]}}{\text{mean of }\left\{\left|\mathbf{w}_k\mathbf{X}^-_{n,k}\right|\right\}_{n\in[1,|\mathbf{T}^-|]}}, \qquad (17)$$

ここで，$\mathbf{T}^-$ と $\tilde{\mathbf{T}}^-$ はそれぞれ負例の全てと負例の少数派を表し，$\tilde{\mathbf{X}}^-$ は全負例における特徴ベクトルの集合を表す．$\tilde{\mathbf{T}}^-_{\tilde{n}}$ と $\mathbf{S}_k$ に対して，式 (17) の $|\mathbf{w}_k\tilde{\mathbf{X}}^-_{\tilde{n},k}|$ はスコア関数 (2) への寄与度を表す．そのため，$k^*=\mathrm{argmax}_{k=1,\cdots,K}\mathrm{CB}(\mathbf{S}_k)$ を満たす $\mathbf{S}_{k^*}$ は，負例の少数派を区別するのに最も特化した shapelet と言える．

6.1　医療分野における PVC の検知

心室期外収縮（PVC）の検知は，生命に関わる心臓病の診断において重要なタスクである．正常または異常のインスタンスをそれぞれ正または負クラスに割り当てる．PVC を見逃さないことが重要なため，FPR を [0,0.05] と低い範囲に絞り込んだ pAUC を考慮する．UCR データセット [24] の中から非侵襲な

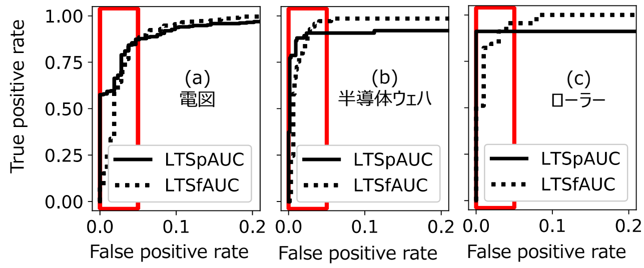

図 4　ROC 曲線の拡大図．いずれのデータセットでも LTSpAUC は full AUC を同等か低下させるが pAUC を向上する．

心電図の時系列データセット（ECG5000）を用いる．負例は次の 3 つの異常タイプからなる [6]: (A1) R-on-T 型の PVC, (A2) 上室期外または異所性拍動, (A3) PVC [26]．5.1 節の実験設定とは異なり，ここでは特定の異常タイプにのみ表れる shapelets を発見したい．そのため，shapelets が異常な時系列データにフィッティングしやすくなるよう $K^-=K=40$ としてオプションの制約式 (10) を利用する．

図 4 (a) は，LTSpAUC と LTSfAUC によるテストデータでの ROC 曲線の拡大図を表す．FPR が 0.05 以下の範囲を赤枠で囲っている．いずれの手法でも full AUC は 0.97 であるが，pAUC は LTSpAUC と LTSfAUC とでそれぞれ 0.71 と 0.54 である．これにより，pAUC が大きく改善されることを確認した．

PVC の少数派を捉えるのに貢献している shapelets を調べる．異常タイプ (A1) R-on-T 型の PVC と異常タイプ (A3) PVC とは，負例の中でそれぞれ 86% と 9% を占めており，異常タイプ (A3) は PVC 少数派と言える．異常の時系列データセット $\mathbf{T}^-$ に対して，式 (17) における負例の少数派 $\tilde{\mathbf{T}}^-$ をその PVC 少数派として，LTSpAUC による式 (17) の CB を計算する．図 5 (a) は，LTSpAUC により学習された shapelets の中で CB の上位 20 個を表している．図 6 では，図 5 (a) の最左の shapelet（つまり最高の CB の値を持つ shapelet $\mathbf{S}_{k^*}$）を，正常または PVC 少数派の時系列インスタンスに最もマッチングする箇所に重ねて描いている．期待通り，その shapelet は負クラス（つまり異常クラス）の時系列データにフィッティングしている．更に，その shapelet は QRS 波の一部を捉えている．R-on-T の間隔が縮まる以外に，QRS 波の幅が広がることが PVC の重要な兆候であり [27]，図 6 の結果はその医学的な知見と一致していることが分かる．

6.2　製造分野における半導体ウェハの異常検知

異常検知及び分類（FDC）は，半導体製造の自動化に向けて

6：分類不能とラベル付けされた 24 個のインスタンスを除外する．

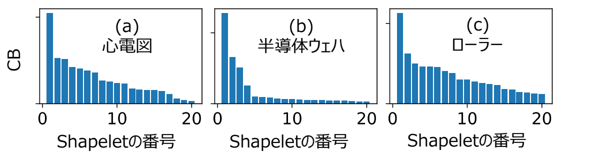

図 5　LTSpAUC による上位 20 個の式 (17) における CB．いずれのデータセットでも 1 番目の CB は 2 番目の CB より約 2 倍高い．

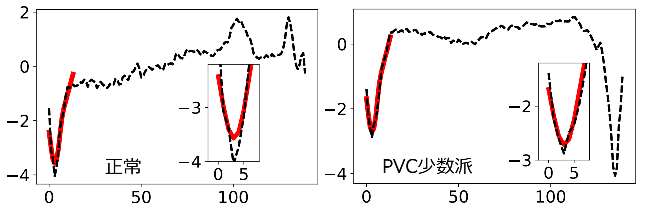

図 6　心電図の時系列データで LTSpAUC が発見した shapelet

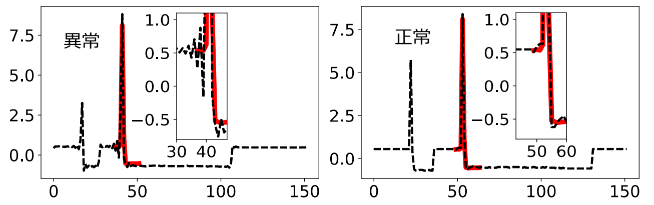

図 7　半導体ウェハの時系列データで LTSpAUC が発見した shapelet

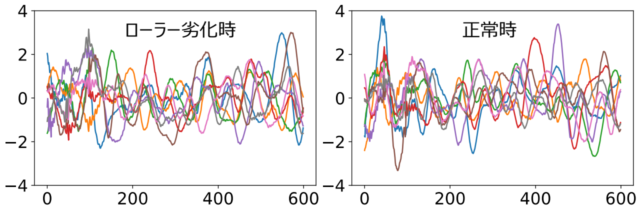

図 8　ローラーの時系列データセットの一部

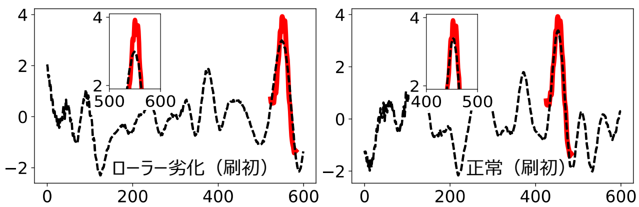

図 9　ローラーの時系列データで LTSpAUC が発見した shapelet

最も重要なタスクの 1 つである．正常または異常のインスタンスをそれぞれ負または正クラスに割り当てる．製造業者は誤報が多発するリスクを抑えるため [10]，FPR の範囲を [0,0.05] と低く絞り込んだ pAUC を考慮する．UCR データセット [24] の中から半導体ウェハの時系列データセット（wafer）を用いる．そのエッチング加工処理は，250 以上のステップからなるため様々な波形パターンを含む．5.1 節の実験設定とは異なり，FDC ではセンサ値の平均や標準偏差を考慮するのが一般的なため，各手法は z-normalization 無しで式 (1) を計算する．

図 4 (b) は，LTSpAUC と LTSfAUC によるテストデータでの ROC 曲線を表す．LTSpAUC と LTSfAUC に対して，full AUC は 0.95 と 0.98 であるが pAUC は 0.86 と 0.78 である．これにより，pAUC は大きく改善されることを確認した．

pAUC の性能差に貢献している shapelets を調べる．比較手法 LTSfAUC による式 (2) のスコアに従って誤分類した負例の上位 5%を選び，それらを式 (17) における負例の少数派 $\tilde{\mathbf{T}}^-$ として，提案手法 LTSpAUC による式 (17) の CB を計算する．図 5 (b) は，LTSpAUC で学習した shapelets に対して CB の上位 20 個を表している．図 7 では，最高の CB の値を持つ shapelet $\mathbf{S}_{k^*}$ を，時系列インスタンスに最もマッチングする箇所に重ねて描いている．$\mathbf{S}_{k^*}$ により発見された波形パターンは正常な時系列データにフィットし，そのパターンは異常な時系列データでのみ摂動する．加えて，そのパターンはこのデータセットの中で稀にしか現れないが，その稀なケースの分類には有効である．これにより，LTSpAUC は pAUC の最適化に有効で稀な波形パターンを shapelet として発見できることを確認した．

6.3　保守分野におけるローラーの劣化検知

スキャナ，コピー機，プリンタなどオフィスオートメーション機器では，ローラーは紙を搬送する重要な部品である．ほこりの堆積や摩耗を防ぐためローラーには保守が必要であり，安価なセンサを用いてプロアクティブに保守することで，製造や保守にかかるコストを削減できる．ここでは，正常時または劣化時のインスタンスをそれぞれ負または正クラスに割り当てる．不要な保守を行うと逆にコストが増大するため，FPR の範囲が [0,0.05] と低い pAUC を考慮する．我々はスキャナ（ScanSnap f1-7160）のローラーの近くに低コストの加速度センサ（MMA7361）を取り付けて，ローラーを劣化させるため 28 万枚の紙をスキャンした．スキャンする際，ハミング窓で前処理した 1 枚単位のセンサ信号を時系列インスタンスとして定期的に収集した．スキャナの警告に基づき，スキャンした枚数が 17 万枚以下の時を負例（正常時）とし 20 万枚以上の時を正例（劣化時）とする．図 8 は収集した時系列データセットの一部である．6.2 節と同様に，各手法は z-normalization 無しで式 (1) を計算する．

図 4 (c) は各手法による ROC 曲線を表す．LTSpAUC と LTSfAUC に対して，full AUC は 0.95 と 0.99 であるが pAUC は 0.91 と 0.81 であり，pAUC は大きく改善されることを確認した．

pAUC の性能差に貢献している shapelets を調べる．LTSfAUC による式 (2) のスコアに従って誤分類した負例の上位 5%を選び，それらを式 (17) の $\tilde{\mathbf{T}}^-$ として，LTSpAUC による式 (17) の CB を計算する．図 5 (c) は，LTSpAUC で学習した shapelets に対して CB の上位 20 個を表している．図 9 では，最高の CB の値を持つ shapelet $\mathbf{S}_{k^*}$ を，時系列インスタンスに最もマッチングする箇所に重ねて描いている．ローラーの動きは刷り初めと連続して印刷している場合とで異なり，刷り初めのインスタンスは少数である．Shapelet $\mathbf{S}_{k^*}$ はその刷り初めに特有の波形の一部を捉えている．更に，ローラーが劣化すると滑りやすくなるため，加速度センサから得られる波形のピークは丸まりやすくなる．図 9 より，発見された shapelet がこれらの現象を表していることが分かる．

7 まとめ

任意の FPR の範囲で pAUC（または full AUC）を最適化するように shapelets と分類器の両方を同時に学習する手法 LTSpAUC を提案した．また，幾つかの shapelets が特定のクラスの時系列データに明示的に近づきやすくする方法を提案し，それを射影劣勾配降下法で効率的に解けることを理論的に示した．加えて，その計算量は時系列長の線形オーダーまで削減される．負例の数が多い UCR 時系列データセットを用いた実験評価では，FPR の範囲を低く抑えた pAUC が，最新の shapelets 学習手法である LTS や CSLTS のものよりも有意に高いことを確認した．更に，医療・製造・保守の各産業分野への適用事例では，pAUC の向上を確認するとともに，負例の少数派を分類することに特化した shapelets の発見が産業分野の知見や現象などに一致することを示した．

文献

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

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

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

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

[5] 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).

[6] H. Narasimhan and S. Agarwal. “A Structural SVM Based Approach for Optimizing Partial AUC”. In: *ICML*. ACM, 2013.

[7] H. Narasimhan and S. Agarwal. “SVMpAUCtight: A New Support Vector Method for Optimizing Partial AUC Based on a Tight Convex Upper Bound”. In: *KDD*. ACM, 2013.

[8] L. E. Dodd and M. S. Pepe. “Partial AUC Estimation and Regression”. In: *Biometrics* (2003).

[9] Z. Wang and Y.-C. I. Chang. “Marker selection via maximizing the partial area under the ROC curve of linear risk scores”. In: *Biostatistics* (2011).

[10] W. Ye, Y. Lin, M. Li, Q. Liu, and D. Z. Pan. “LithoROC: Lithography Hotspot Detection with Explicit ROC Optimization”. In: *ASPDAC*. ACM, 2019.

[11] R. T. Olszewski. “Generalized Feature Extraction for Structural Pattern Recognition in Time-series Data”. PhD thesis. Carnegie Mellon University, 2001.

[12] H. J. Chang, D. S. Song, P. J. Kim, and J. Y. Choi. “Spatiotemporal Pattern Modeling for Fault Detection and Classification in Semiconductor Manufacturing”. In: *IEEE Transactions on Semiconductor Manufacturing* (2012).

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

[14] A. Yamaguchi and T. Nishikawa. “One-Class Learning Time-Series Shapelets”. In: *Big Data*. IEEE, 2018.

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

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

[17] S. Paisitkriangkrai, C. Shen, and A. van den Hengel. “Pedestrian Detection with Spatially Pooled Features and Structured Ensemble Learning”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2016).

[18] P. Kar, H. Narasimhan, and P. Jain. “Online and Stochastic Gradient Methods for Non-decomposable Loss Functions”. In: *International Conference on Neural Information Processing Systems*. Curran Associates, Inc., 2014.

[19] E. Eban, M. Schain, A. Mackey, A. Gordon, R. Rifkin, and G. Elidan. “Scalable Learning of Non-Decomposable Objectives”. In: *AISTATS*. PMLR, 2017.

[20] T. Joachims. “A Support Vector Method for Multivariate Performance Measures”. In: *ICML*. ACM, 2005.

[21] J. Kim, R. D. C. Monteiro, and H. Park. “Group Sparsity in Nonnegative Matrix Factorization”. In: *SDM*. SIAM, 2012.

[22] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. “Proximal Methods for Sparse Hierarchical Dictionary Learning”. In: *ICML*. ACM, 2010.

[23] D. Arthur and S. Vassilvitskii. “K-means++: The Advantages of Careful Seeding”. In: *SODA*. Society for Industrial and Applied Mathematics, 2007.

[24] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, et al. *The UCR Time Series Classification Archive*. `www.cs.ucr.edu/~eamonn/time_series_data/`. 2015.

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

[26] Y. Chen, Y. Hao, T. Rakthanmanon, J. Zakaria, B. Hu, and E. Keogh. “A General Framework for Never-ending Learning from Time Series Streams”. In: *Data Mining and Knowledge Discovery* (2015).

[27] J. Roelandt, P. Klootwuk, J. Lubsen, and M. J. Janse. “Sudden death during longterm ambulatory monitoring”. In: *European Heart Journal* (1984).