適応型記憶による類似波形を用いたストリームのオンライン予測

山口　晃広† 　真矢　滋† 　稲木　達哉† 　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の普及に伴い，交通流，株価，電力消費量などのストリームから将来の値を高精度に予測するオンライン予測技術が求められている．一方，ストリームの重要な性質としてデータの傾向が時間変化するコンセプトドリフトに対応する研究が近年盛んに行われている．本研究では，(a) 複数のウインドウ長を持つ短期記憶と矛盾が無いようにクラスタリングを行う長期記憶とを適応的に配合し，(b) この適応型記憶でストリームから得られる波形を管理し類似波形の局所回帰を用いることで，コンセプトドリフトに対応するオンライン予測方法を提案する．様々なコンセプトドリフトを持つ人工データと，交通流，株価，電力消費量の実データを用いた評価により，提案方法はデータセットごとにメタパラメータを変えることなく従来方法と比べて高い予測精度を達成することを確認した．

キーワード　ストリーム，オンライン予測，コンセプトドリフト

1. はじめに

IoT の普及に伴い，交通流や電力使用量や株価など，変化する環境の中で終わりなく流れ続けるデータストリーム（以下ストリーム）をリアルタイムに分析する技術に関心が高まっている．ストリームの分析では，データ量が増え続けても記憶容量を一定以下に抑える必要がある．加えて，環境の変化などによりデータの傾向が時間変化する性質であるコンセプトドリフトへの対応も課題となる．

コンセプトドリフトには急激に変化するタイプ，徐々に変化するタイプ，類似したパターンが繰り返し発生するタイプなどがあり [1]，それらに対応する技術への関心が高まっている．スライディングウインドウ（以下ウインドウ）は，これに対応する良く知られた記憶管理であり，短期的に最近のデータを保持する．しかし，ウインドウの長さは，長いとデータの急激な変化に適応することが難しく，逆に短いとノイズの影響を受けやすい．また，古いデータを捨ててしまうため，長期的に同様のパターンが繰り返し発生する場合に対応が難しい．近年，ウインドウの長さを適応的に変える短期記憶に加えて古いデータを長期記憶に圧縮して保持するクラス分類方法が提案され，実データや様々なコンセプトドリフトを持つ人工データにおいて高い識別性能を達成している [2, 3]．

一方，交通流，株価，電力消費量では，近い将来の値をリアルタイムに予測するオンライン予測技術が求められる．例えば交通流の予測では，路側器から 5 分間隔で配信される交通流の情報から数ステップ先の交通流を予測することで，リアルタイムな自動車のルート案内や交通制御などへの活用が期待される [4, 5]．これらの予測では，ウインドウを用いて短期的なデータから回帰により将来の値を予測する方法がよく研究されている．自己回帰（AR）など線形回帰を用いる方法が良く知られているが，[6] など非線形なカーネル回帰をウインドウのデータに適用する研究なども行われている．また，コンセプトドリフトに対応するため適応的にウインドウ長を変えて予測する方法も提案された [7]．一方，過去の履歴を用いる方法として，過去に出現した波形を保持しておき，現在の波形と類似する波形を用いて将来の値を予測する方法が，交通流，株価，電力消費量などの予測に用いられている [8, 9, 10]．

しかしながら，従来のオンライン予測方法では様々なコンセプトドリフトを持つストリームを精度よく予測することは難しい．ウインドウを用いる従来のオンライン予測方法では過去の履歴を用いないため，類似したパターンが長期的に繰り返し発生する場合に過去の情報を有効に活用できない．一方，過去の履歴を用いる従来方法では予測前に過去の一定期間から履歴をあらかじめ抽出するため，コンセプトドリフトを考慮しない．[2, 3] ではコンセプトドリフトを考慮しながら長期的な記憶を圧縮して保持する．しかし，[2, 3] はクラス分類が対象である．また，短期記憶と長期記憶に矛盾が発生した場合には長期記憶から矛盾した部分を取り除き短期記憶で上書きしてしまうため，取り除かれた情報をそれ以降活用できない．

本研究では，コンセプトドリフトを持つストリームに対して，近い将来の値を高精度に予測する汎用的なオンライン予測方法を提案する．短期記憶と長期記憶を適応的に用いるクラス分類方法 [2, 3] のアイデアを発展させて，本提案ではオンライン予測に適応する．特に [2, 3] の記憶に矛盾が発生した場合に古い記憶が消えてしまう課題を克服するため，長期記憶を矛盾が無いように複数のクラスタに分けて管理する．この適応型記憶でストリームから得られる波形を管理し，類似波形の局所回帰により将来を予測する．これにより，非線形なストリームやデータ傾向の変化に柔軟に対応する．様々なコンセプトドリフトを持つ人工データと，交通流，株価，電力消費量の実データを用いた評価により，提案方法ではデータセットごとにメタパラメータを変えることなく [6, 7] を含む比較方法と比べて高い予測性能が得られることを確認する．

2. 関連研究

オンライン予測技術として良く知られている方法では，ウイ

ンドウによる短期記憶のみを用いて回帰モデルにより予測器が将来の値を予測する．予測器としては，自己回帰（AR）など線形回帰を用いる方法が広く知られているが，実データは非線形であるため精度が悪い場合が多い．精度を向上するため，ニューラルネット，サポートベクトル回帰，カーネルリッジ回帰などの非線形な回帰を用いる方法も提案されている [11, 5, 6]．非線形回帰では予測器の学習に時間がかかるため，[6] ではウインドウをスライドするたびに逐次的にカーネルリッジ回帰の予測器を学習する方法が提案された．しかし，これらの方法では短期記憶から外れた古いデータを捨ててしまうため，予測に有効なパターンが古いデータにあってもそれを活用できない．また，ウインドウ長は固定されており，データの急激な変化とノイズへの頑健性とを両立することが難しい．

予測精度を向上するため，過去の履歴を用いる方法も提案されている．例えば平日の通勤時には交通量が増えるなど，曜日やイベントなどのカレンダー情報を利用する方法があるが [4, 8]，これらは特定分野の知識を用いており汎用性に欠ける．一方，過去に発生した波形を保持しておき，現在の波形に類似した過去の波形を k 近傍法により求め，それらの類似波形を用いて局所的に将来の値を予測する方法が提案されている．このような類似波形を用いる方法は，交通流，株価，電力消費量など様々な分野のオンライン予測に用いられており，各分野で良好な性能を達成している [8, 9, 10]．しかし，これらの従来方法では，あらかじめ予測する前に一定期間内に発生した波形を保持しておき予測時にそれを用いる．そのため，データの傾向がその時から変わると対応できない．これに対応する自明な方法として，固定長のウインドウでストリームから得られる波形を管理する方法が挙げられる．しかし，データに合わせたウインドウ長のチューニングが必要である．

コンセプトドリフトへの対応には，クラス分類やオンライン予測など特定のタスクを対象とした方法と特定のタスクに依らない方法が，これまで提案されている．特定のタスクに依らない代表的な方法として，コンセプトドリフトを検知して動的にウインドウの長さを変える方法がある [12, 13]．しかし，これらの方法ではコンセプトドリフトを陽に検知するための適切な閾値を設定する必要がある．また，データの傾向が急激に変わる場合には有効であるが，徐々に変わる場合や長期的に同様の傾向が繰り返し発生する場合には不向きである．特定のタスクに依らない別なアプローチとして，複数のモデルを環境に応じて適応的に組み合わせるアンサンブルに基づく方法も提案されている [14, 15]．しかし，[13, 14, 15] よりも [2, 3] の方が様々なコンセプトドリフトを持つ人工データや複数分野の実データにおいてクラス分類の性能が高いことが経験的に確認されている [2, 3]．

オンライン予測を対象としてコンセプトドリフトに対応する方法も提案されている．[16] では，複数のタイムスケールからなる予測を組み合わせ，複数の非線形な予測モデルから最近の傾向に合う予測モデルを選んで予測する．しかし，近い将来ではなく遠い将来の予測を対象としている．また，予測モデルはリアルタイムに生成されるが，一定の記憶容量以下に抑える仕組みが無いため予測モデルの数が増え続ける可能性がある．一方，[7] では複数の異なるウインドウ長による予測結果を重みづけて統合する方法が提案された．[7] では，コンセプトドリフトを陽に検知する必要が無く [12] の方法よりも精度が高い．しかし，ウインドウによる短期記憶のみを用いて古いデータを捨ててしまうため，長期的に有効なデータがあってもそれを活用できない．また，予測器は線形であり非線形な予測器との性能比較は明らかにされていない．

近年，クラス分類を対象として短期記憶と長期記憶の 2 つの記憶管理を適応的に用いる k 近傍法が提案された [2, 3]．短期記憶では，[7] と同様のアイデアで複数の異なるウインドウ長により予測する．長期記憶では，古くなったデータを圧縮して保持し，長期的に繰り返しパターンが発生する場合に対応する．ウインドウ長や記憶の選択といったメタパラメータのチューニングも不要であり，様々なコンセプトドリフトからなる人工データや実データで有効性が確認された．しかし，クラス分類の方法でありオンライン予測に直接適用できない．また，短期記憶と長期記憶に矛盾が発生した場合には短期記憶で長期記憶を上書きしてしまう．そのため，上書きされた記憶が現在の傾向に合っていても再利用できない．

3. 問題設定

ストリームはタプルの列 $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t, \cdots)$ である．ここで，タプル $\mathbf{x}_i$ は J 次元ベクトル (i.e.,$\mathbf{x}_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \cdots, \mathbf{x}_{i,J})$) である．現在時刻 t において N ステップ先の値を予測する場合，オンライン予測問題とは $\mathbf{x}_{t+N}$ のある変数 j の予測値 $\hat{\mathbf{x}}_{t+N,j}$ を求める問題である．予測ステップ N，タプルの次元 J，予測対象となる変数 j は，ユーザにより予測前に与えられる定数である．以降，$y_{i+N} = \mathbf{x}_{i+N,j}$ 及び $\hat{y}_{i+N} = \hat{\mathbf{x}}_{i+N,j}$ と記述する．本研究では近い将来の予測を対象とし N は 1 から 5 ステップ程度を想定する．

時刻 t に $\hat{y}_t$ が予測された N ステップ後に，y_t が得られる．これにより予測器は予測値と観測値との誤差である予測誤差を測定できる．予測誤差を式 1 のようにユークリッド距離で測る．

$$\mathrm{loss}(\hat{y}_t, y_t) = (\hat{y}_t - y_t)^2 \tag{1}$$

時刻 t における長さ D の波形は $\mathbf{s}_t = (\mathbf{x}_{t-D+1}, \cdots, \mathbf{x}_t)$ であり $D \times J$ 次元のベクトルである．記憶には，波形 $\mathbf{s}_i$ と N ステップ先の値 y_{i+N} の組からなる $D \times J + 1$ 次元のベクトル $\mathbf{z}_i = (\mathbf{s}_i, y_{i+N})$ がサンプルとして保持される．記憶全体で保持できる最大サンプル数は予測前にあらかじめ決まっている．

4. 提案方法

本節では，まず始めに提案方法の概要を述べ，次に提案方法の各構成要素を説明する．

4.1 全体構成

図 1 のように記憶管理は短期記憶と長期記憶からなる．短期記憶は，サンプルを要素とするウインドウで管理される．ウインドウ長は 1 つに固定されず複数の異なるウインドウ長を用いる．一方，長期記憶ではウインドウに入りきらない古いサンプ

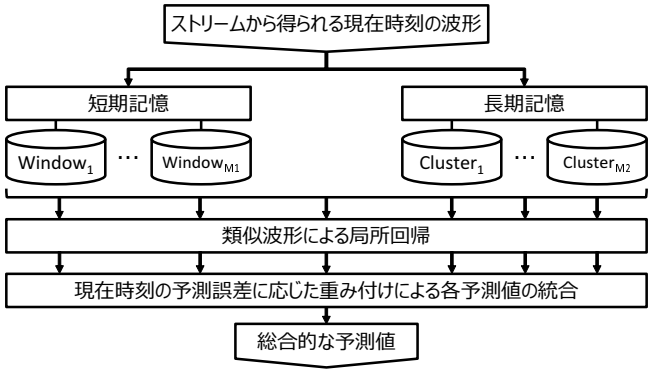

図 1　提案する予測方法の全体像

ルを圧縮して保持する．特に [2, 3] と異なり，本提案では長期記憶を矛盾の無いようにクラスタリングする．本来，波形 $\mathbf{s}_t$ が類似すれば N ステップ先の観測値 y_{t+N} も類似し同様の傾向を示すはずである．しかし，そのような傾向を示さない場合には，それらのサンプルの間に矛盾があると考え，それらを別々のクラスタで管理する．その結果，各クラスタ内では矛盾の無いようにサンプルが保持される．短期記憶と長期記憶については 4.3 節と 4.4 節でそれぞれ詳しく述べる．

記憶からサンプル $(\mathbf{s}_i, y_{i+N})$ の集合が適切に与えられれば，予測方法は以下のような類似波形を用いた局所回帰により実現される．まず，現在時刻 t_1 における波形 $\mathbf{s}_{t_1}$ に類似する K 個の波形 $\mathbf{s}_{i_1}, \mathbf{s}_{i_2}, \cdots, \mathbf{s}_{i_K}$ をサンプル集合から取り出す．次に，$\mathbf{s}_{i_1}, \mathbf{s}_{i_2}, \cdots, \mathbf{s}_{i_K}$ から $y_{i_1+N}, y_{i_2+N}, \cdots, y_{i_K+N}$ を推定する回帰式を求め，その回帰式に現在時刻の波形 $\mathbf{s}_{t_1}$ を代入することで N ステップ先の予測値 $\hat{y}_{t_1+N}$ を予測する．このように局所的なサンプルに基づいて予測を行うことで，非線形なストリームに対応するだけでなくデータの傾向が局所的に変わる場合にも素早く柔軟に対応することが期待される [17]．詳しい方法は 4.2 節で述べる．

各記憶で管理する様々なサンプル集合を用いて予測を行い，その結果を重みづけて統合し最終的な予測値を出力する．現在時刻の予測誤差に基づき予測結果の重み付けを適応的に調整することで，様々なコンセプトドリフトを持つストリームに対応する．詳しくは 4.5 節で述べる．

4.2　類似波形を用いたオンライン予測

サンプル $(\mathbf{s}_i, y_{i+N})$ の集合と現在時刻 t_1 における波形 $\mathbf{s}_{t_1}$ が与えられたとき，類似波形を用いた予測方法をまずは定式化する．この方法は，カレンダー情報の併用などのドメイン固有の部分を取り除いた類似波形の局所回帰による従来方法 [8] と同様のアイデアである．

現在時刻の波形 $\mathbf{s}_{t_1}$ とサンプル集合のある波形 $\mathbf{s}_i$ との距離をユークリッド距離（i.e., $||\mathbf{s}_{t_1} - \mathbf{s}_i||^2$）で測る．そのとき，サンプル集合の波形の中で $\mathbf{s}_{t_1}$ との距離が近い K 個の類似波形を $\mathbf{s}_{i_1}, \mathbf{s}_{i_2}, \cdots, \mathbf{s}_{i_K}$ とする．$\mathbf{X} = (\mathbf{s}_{i_1}^{\mathrm{T}}, \mathbf{s}_{i_2}^{\mathrm{T}}, \cdots, \mathbf{s}_{i_K}^{\mathrm{T}})$ として $\mathbf{y} = (y_{i_1+N}, y_{i_2+N}, \cdots, y_{i_K+N})$ とする．ここで $\mathbf{X}$ は D 行 K 列の行列である．以降，$\mathbf{X}$ を平均 0 と標準偏差 1 で標準化したもとで議論する．このとき，類似波形を用いた局所回帰は式 2 で最適化される線形回帰の係数 $\mathbf{w}$ を求める問題である．

$$\min_{\mathbf{w}} ||\mathbf{y} - \mathbf{w}\mathbf{X}||^2 + \lambda||\mathbf{w}||^2 \tag{2}$$

従来方法 [8, 9, 10] とは異なり，本研究ではウインドウ長を短くしてコンセプトドリフトに対応する．これによるオーバーフィットを避けるため第 2 項に正則化項を加えている．λ は正の定数である．式 2 はリッジ回帰と同じ形式であり，$\mathbf{w}$ は式 3 で算出できる．

$$\mathbf{w} = \left(\left(\mathbf{X}\mathbf{X}^{\mathrm{T}} + \lambda \mathbf{I} \right)^{-1} \mathbf{X}\mathbf{y}^{\mathrm{T}} \right)^{\mathrm{T}} \tag{3}$$

ここで $\mathbf{I}$ は D 行 D 列の単位行列である．

予測値 $\hat{y}_{t_1+N}$ は式 3 の $\mathbf{w}$ を用いて式 4 で予測される．

$$\hat{y}_{t_1+N} = \mathbf{w}\mathbf{s}_{t_1}^{\mathrm{T}} \tag{4}$$

4.3　短期記憶

ウインドウの長さにはトレードオフの問題がある．その最適な長さはデータの傾向によって異なるため，最適なウインドウ長を人手で決めることは難しい．そのため，提案方法では [2, 3, 7] と同様に，様々なウインドウサイズを用意しておき，現在時刻の予測誤差を小さくするように各ウインドウ長による予測値を重みづける．これにより，データの傾向ごとにウインドウ長をチューニングする問題を解決する．

短期記憶で保持する全サンプルの集合を $\mathrm{STM}_{\mathrm{all}}$ とする．ウインドウ長が l_m の場合のサンプル集合 STM_m は式 5 となる．

$$\mathrm{STM}_m = \{\mathbf{z}_t \mid t_0 - l_m < t \leqq t_0, \mathbf{z}_t \in \mathrm{STM}_{\mathrm{all}}\} \tag{5}$$

ここで t_0 はウインドウにサンプルが挿入された最新時刻である．ウインドウ長 l_m は，最小ウインドウ長 $L_{\min}$ とウインドウ長の増分 L_{step} と最大ウインドウ長 $L_{\max}$ を与えられて，式 6 のように決める．

$$l_m = \begin{cases} L_{\min} & (m = 1) \\ L_{m-1} + (m-1)L_{\mathrm{step}} & (m = 2, 3, \cdots, M_{\mathrm{STM}}) \end{cases},$$
$$M_{\mathrm{STM}} = \underset{m=1,2,\cdots}{\arg\max}\{l_m \mid l_m \leqq L_{\max}\} \tag{6}$$

つまり，$\mathrm{STM}_1 \subsetneq \mathrm{STM}_2 \subsetneq \cdots \subsetneq \mathrm{STM}_{M_{\mathrm{STM}}} = \mathrm{STM}_{\mathrm{all}}$ である．

現在時刻 t_1 にタプル $\mathbf{x}_{t_1}$ が観測されたとき，$\mathrm{STM}_{\mathrm{all}}$ には $\mathbf{z}_{t_0} = (\mathbf{s}_{t_0}, \mathbf{x}_{t_0+N,j})$ が最新のサンプルとして挿入される．ここで $t_0 = t_1 - N$ である．短期記憶で管理する全サンプルはウインドウで管理され，$\mathrm{STM}_{\mathrm{all}}$ は式 7 となる．

$$\mathrm{STM}_{\mathrm{all}} = \begin{cases} \bigcup_{t=1}^{t_0} \mathbf{z}_t & (t_0 \leqq L_{\max}) \\ \bigcup_{t=t_0-L_{\max}+1}^{t_0} \mathbf{z}_t & (L_{\max} < t_0) \end{cases} \tag{7}$$

$L_{\max} < t_0$ においてサンプル $\mathbf{z}_{t_0}$ が短期記憶に挿入された場合，短期記憶の最古のサンプル $\mathbf{z}_{t_0-L_{\max}}$ は捨てられず長期記憶に移される (注1)．

4.4　長期記憶

提案方法では [2, 3] と異なり，長期記憶を複数のクラスタで

(注1)：オンライン予測における観測値にはノイズがのることで，長いウインドウ長が一時的に不要でもその後直ぐに必要となる場合もある．そのため，[2, 3] とは異なり，短期記憶に入りきらない古いサンプルのみを長期記憶に移動する．

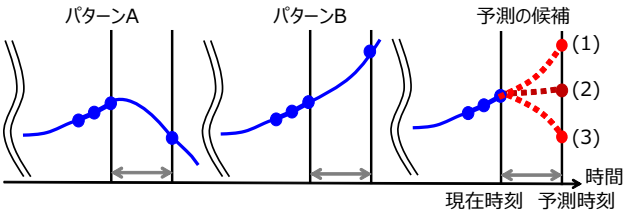

図 2　長期記憶のクラスタリングによる予測値の候補

管理する．その動機を図 2 の例を用いて説明する．パターン A と B は波形が類似するにも関わらず N ステップ先の傾向は下降または上昇し異なる．もし，パターン A と B のサンプルを区別せずに混ぜて予測に用いると，図 2 の (2) のような 1 つの予測値しか得られず，N ステップ先が下降するケース (3) や上昇するケース (1) が予測値の候補として区別して得られない．そのため，長期記憶ではパターン A と B のようにお互いに矛盾するサンプルを異なるクラスタで分割管理する．これにより，図 2 の (1)(2)(3) のような異なるケースの予測を区別できる．

各クラスタではサンプルを更にクラスタリングして圧縮することで長期記憶の全サンプル数を $L_{\max}$ 以下に抑える．このアプローチは [18] に基づき，本研究では圧縮された各サンプルをミクロクラスタと呼ぶ．サンプル数が $L_{\min}$ 以下のクラスタについては，それらに属するサンプルを外れ値のように扱い，そのクラスタ単体では予測に用いない．

短期記憶からサンプルが送られてきたとき，長期記憶の処理は 2 つに分けられる．1 つ目の場合は，短期記憶から初めて長期記憶にサンプルが保持される時と長期記憶の記憶容量が一杯になった時に行われるクラスタの構築処理である．2 つ目の場合は，長期記憶にサンプルが既に保持されておりその記憶容量に余裕がある時に行われるクラスタの逐次更新処理である．以降ではそれぞれの処理を説明する．

4.4.1　クラスタの構築処理

波形 $\mathbf{s}_{i_1}$ と $\mathbf{s}_{i_2}$ が類似するにも関わらず N ステップ先の観測値 y_{i_1+N} と y_{i_2+N} が互いに異なる場合にサンプル $\mathbf{z}_{i_1}$ と $\mathbf{z}_{i_2}$ を別々のクラスタに割り当てる．ここでは，長期記憶に保持される全サンプルを対象にする．但し，初めて長期記憶にサンプルを保持する時は短期記憶の全サンプルを長期記憶へコピーしそれらを対象にする．サンプル $(\mathbf{s}_{i_1}, y_{i_1+N})$ と $(\mathbf{s}_{i_2}, y_{i_2+N})$ の距離を式 8 で定義し，M_{LTM} 個のクラスタを生成する．

$$|y_{i_1+N} - y_{i_2+N}| \exp\left(\frac{-\alpha||\mathbf{s}_{i_1} - \mathbf{s}_{i_2}||^2}{\mathrm{var}(\{\mathbf{s}_i\}_{(\mathbf{s}_i, y_{i+N}) \in \mathrm{LTM}_{\mathrm{all}}})}\right), \quad (8)$$

ここで，$\mathrm{LTM}_{\mathrm{all}}$ は長期記憶に保持されているサンプル集合であり，var(.) は分散を表し，α は正の定数である．クラスタリングアルゴリズムには k-means++[19] を用いる．

次に，先のクラスタリング処理で構築した各クラスタに含まれるサンプルを圧縮する．$m = 1, 2, \cdots, M_{\mathrm{LTM}}$ に対して，LTM_m は m 番目のクラスタのサンプル集合を表す．各 m 番目のクラスタに対して，$\frac{L_{\max}}{M_{\mathrm{LTM}}} \leqq |\mathrm{LTM}_m|$ ならばミクロクラスタ数が $\frac{L_{\max}}{2M_{\mathrm{LTM}}}$ 個となるように LTM_m に k-means++[19] によるクラスタリングを適用する．その距離にはサンプル間のユークリッド距離を用いる．クラスタリングを行った場合，LTM_m では各クラスタの重心が圧縮されたサンプル（i.e., ミクロクラスタ）となる．

4.4.2　クラスタの逐次更新処理

本処理では，4.4.1 節の処理によりクラスタが構築されており長期記憶の記憶容量にも空きがある場合を扱う．サンプル $(\mathbf{s}_i, y_{i+N})$ が長期記憶に送られてきた時に，長期記憶に矛盾が生じないように適切なクラスタへ $(\mathbf{s}_i, y_{i+N})$ を挿入する．具体的には，$\hat{y}_{i+N}$ を各クラスタのサンプル集合で予測したとき，予測誤差が最小となるクラスタを選ぶ．もしそのクラスタにサンプルの数が十分にあれば，4.2 節と同じ方法で $\hat{y}_{i+N}$ を予測する．もしそのクラスタにサンプル数が十分になければ，そのサンプル集合から $\mathbf{s}_i$ の類似波形を取り出しそれらの N ステップ先の平均値を予測値 $\hat{y}_{i+N}$ とする．

まとめると，$m = 1, 2, \cdots, M_{\mathrm{LTM}}$ に対して，式 9 で定義される $(\mathbf{s}_i, y_{i+N})$ と LTM_m との距離 (注2) が最小となる m 番目のクラスタに $(\mathbf{s}_i, y_{i+N})$ を挿入する．

$$\begin{cases} \infty & (|\mathrm{LTM}_m| = 0) \\ \mathrm{loss}(y_{i+N}, \frac{1}{K'}\sum_{k=1}^{K'} y_{i_k+N}) & (0 < |\mathrm{LTM}_m| < L_{\min}) \\ \mathrm{loss}(y_{i+N}, \mathbf{w}\mathbf{s}_i^{\mathrm{T}}) & (L_{\min} \leqq |\mathrm{LTM}_m|) \end{cases}, \quad (9)$$

$$K' = \min\{K, |\mathrm{LTM}_m|\}$$

ここで，$\mathbf{w}$ は 4.2 節で述べた類似波形の局所回帰により式 3 で計算される係数ベクトルである．y_{i_k+N} は 4.2 節で述べた類似波形 $\mathbf{s}_{i_k}$ に対応する N ステップ先の値である．

4.5　予測結果の統合

本節では，短期記憶と長期記憶から得られる様々なサンプル集合から予測に用いるものを選び，それらのサンプル集合による各予測値の重み付けを現在時刻の予測誤差に応じて適応的に調整して，最終的な予測値を算出する方法を説明する．

短期記憶から得られるサンプル集合 $\mathrm{STM}_1, \cdots, \mathrm{STM}_{M_{\mathrm{STM}}}$，長期記憶から得られるサンプル集合 $\mathrm{LTM}_1, \cdots, \mathrm{LTM}_{M_{\mathrm{LTM}}}$，それらの和集合であるサンプル集合 $\mathrm{LTM}_{\mathrm{all}}$，$\mathrm{STM}_{\mathrm{all}} \cup \mathrm{LTM}_{\mathrm{all}}$ からなる $M_{\mathrm{STM}} + M_{\mathrm{LTM}} + 2$ 個のサンプル集合のうち，サンプル数が $L_{\min}$ 以上のものを予測に用いる．

サンプル数が $L_{\min}$ 以上のサンプル集合の数を M_{used} として，現在時刻を t_1 とする．まず，各サンプル集合と $\mathbf{s}_{t_1-N}$ を用いて 4.2 節で述べた方法により M_{used} 個の予測結果 $\hat{y}_{t_1}^{(1)}, \hat{y}_{t_1}^{(2)}, \cdots, \hat{y}_{t_1}^{(M_{\mathrm{used}})}$ を得る．予測に用いられる m 番目のサンプル集合による予測結果の重み $\mathbf{a}_m$ を，式 10 で算出する．

$$\mathbf{a}_m = \exp\left(-\beta\widetilde{\mathrm{loss}}(\hat{y}_{t_1}^{(m)}, y_{t_1})\right), m = 1, 2, \cdots, M_{\mathrm{used}} \quad (10)$$

ここで，$\widetilde{\mathrm{loss}}(\hat{y}_{t_1}^{(m)}, y_{t_1})$ は $\{\mathrm{loss}(\hat{y}_{t_1}^{(m)}, y_{t_1})\}_{m=1}^{M_{\mathrm{used}}}$ を $[0, 1]$ の範囲にリスケーリングした値であり，β は正の定数である．

次に，各サンプル集合と $\mathbf{s}_{t_1}$ を用いて 4.2 節の方法により M_{used} 個の予測結果 $\hat{y}_{t_1+N}^{(1)}, \hat{y}_{t_1+N}^{(2)}, \cdots, \hat{y}_{t_1+N}^{(M_{\mathrm{used}})}$ を得る．最終

(注2)：式 9 は距離の公理を満たさず $(\mathbf{s}_i, y_{i+N})$ と LTM_m との近さを測るために用いられる．

表 1　本実験で統一して用いるメタパラメータの値

$D = 5$	波形の長さ
$K = 100$	k 近傍法における近傍の数
$\lambda = 1$	リッジ回帰における式 2 の定数
$L_{\min} = 200$	予測に用いられる最小サンプル数
$L_{\text{step}} = 50$	短期記憶におけるウインドウ長の増分
$L_{\max} = 500$	短期記憶または長期記憶に保持可能な最大サンプル数
$M_{\text{LTM}} = 3$	長期記憶のクラスタ数
$\alpha = 10^{-3}$	クラスタリングの距離における式 8 の定数
$\beta = 0.5$	予測結果の重み付けにおける式 10 の定数

的に統合された予測結果は式 11 のように重み付き平均として算出される．

$$\hat{y}_{t_1+N} = \sum_{m=1}^{M_{\text{used}}} \mathbf{a}_m \hat{y}^{(m)}_{t_1+N} \Bigg/ \sum_{m=1}^{M_{\text{used}}} \mathbf{a}_m \qquad (11)$$

5. 実験評価

本実験全体を通して，提案方法のメタパラメータには表 1 の値を統一して用いる．短期記憶と長期記憶を合わせて保持できる最大サンプル数は $2L_{\max}$ となる．また，予測ステップ数 N が 5 でタプルの次元 J が 1 の場合で実験する．

5.1 比較方法

本実験では，4. 節で述べた全ての機能を実現する提案方法（**FullFunc**）を以下の方法と比較する．

NoCluster は，提案方法において長期記憶を複数に分けず [2, 3] のように 1 つのクラスタで管理する．この方法は，4.4 節で述べた長期記憶を矛盾のない複数のクラスタに分ける効果を確認するために比較される．

NoLTM は，提案方法において短期記憶と長期記憶を分けずに短期記憶のみを用いる．この場合，短期記憶に保持できる最大サンプル数は $2L_{\max}$ である．短期記憶のウインドウ長は最大 $2L_{\max}$ まで適応的に変える．この方法は，長期記憶を用いる効果を確認するために比較される．

ARWin は，最近のデータをウインドウに保持し適応的にウインドウ長を変えて線形回帰を行う従来方法である [7]．ウインドウ長は $3, 4, \cdots, 10$ の範囲で変える．この方法は，オンライン予測におけるコンセプトドリフトに対応する従来方法の 1 つとして比較される．

KRWin は，最近のデータを固定長のウインドウに保持しカーネルリッジ回帰により非線形回帰を行う従来方法である [6]．メタパラメータについては，各データセットに対して 1000 ステップから 2000 ステップまでの期間で次の範囲で精度が最良のものを選ぶ．ウインドウ長は提案方法で変更する範囲のウインドウ長に $2L_{\max}$ を加えた範囲 (i.e.,200, 250, 350, 500, 1000) から選ぶ (注3)．カーネル関数には式 12 のガウシアンカーネルを用い，[20] を参考に $\{\gamma||\mathbf{x}_t - \mathbf{x}_{t'}||^2\}_{t,t'=1000}^{2000}$ のメディアンが $10^{-2}, 10^{-1}, \cdots, 10^2$ となる範囲で γ を選ぶ．

(注3)：実際のメモリ容量としてはカーネルのグラム行列とその逆行列を保持する必要があるためサンプル数の 2 乗のオーダーが必要となるが，本実験ではサンプル数の方を揃えて評価する．

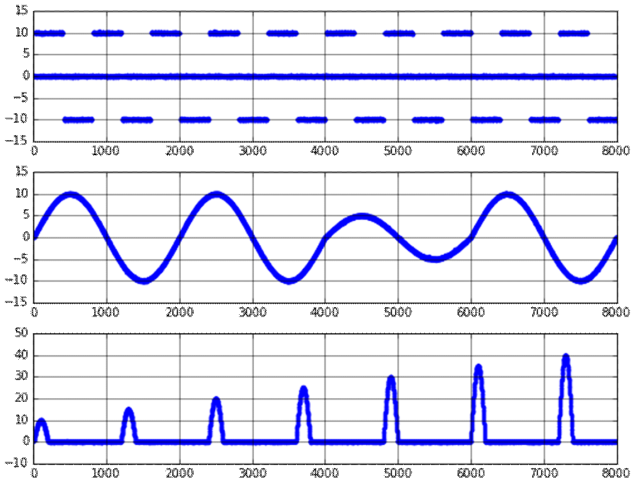

図 3　人工データセット（1 行目 : SuddenReoccurring, 2 行目 : GradualAReoccurring, 3 行目 : GradualBReoccurring）

$$\text{kernel}(\mathbf{x}, \mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2) \qquad (12)$$

正則化項 λ は $10^{-2}, 10^{-1}, \cdots, 10^2$ の範囲から選ぶ．この方法は，ウインドウを用いる非線形オンライン予測における従来方法の 1 つとして比較される．

NRWin は，類似波形の局所回帰によるオンライン予測に固定のウインドウ長を適用した従来方法である．類似波形の局所回帰は 4.2 節で述べた方法と同様であり，従来方法 [8] からドメイン固有の部分を取り除いたものに相当する．ウインドウ長は提案方法で変更する範囲のウインドウ長か $2L_{\max}$ である (i.e.,200, 250, 350, 500, 1000)．この方法は，類似波形を用いるオンライン予測における従来方法の 1 つとして比較される．

NAWin は，類似波形を用いる方法として 4.2 節で述べた方法の代わりに類似波形の N ステップ先の平均値を予測値として用いる．他は NRWin と同様であり，従来方法 [9, 10] からドメイン固有の部分を取り除いたものに相当する．この方法は，類似波形を用いるオンライン予測の中で 4.2 節で述べた方法の有効性を確認するため比較される．

5.2 データセット

5.2.1 人工データ

提案方法や比較方法の違いを確認するため，(Sudden) 急激に変化するタイプ，(Gradual) 徐々に変化するタイプ，(Reoccurring) 同様のパターンが繰り返し発生するタイプのコンセプトドリフトを持つ人工データを用意する．Gradual タイプには，(A) 全体の傾向が変化するタイプと [1]，(B) 部分的な波形パターン（時系列モチーフ）が成長していくタイプがある [21]．そのため，Gradual タイプについては (A) と (B) のそれぞれのタイプの人工データを用意する．Reoccurring タイプは，Sudden タイプまたは Gradual タイプのコンセプトドリフトを持つ人工データに混ぜる．いずれの人工データも，総ステップ数は 8000 ステップであり，平均 0 で標準偏差 0.1 の正規分布に従ったランダムノイズが付加される．

図 3 に示す以下の 3 つの人工データを用いる．

SuddenReoccurring は，急激な変化が繰り返し発生する．

短期間（20 ステップごと）にコンセプトドリフトが発生する．それによる y_t の変化は $(0, \pm 10, 0, \pm 10, 0, \cdots)$ を繰り返す．ここで記号 $(\pm)$ は y_t が 0 と ± 10 へそれぞれ 10 回変化するごとに符号を反転する．

GradualAReoccurring は，ストリーム全体の傾向が滑らかに変化する．傾向の変化は非線形であり，振幅が $(10, 10, 5, 10)$ からなる正弦曲線を 1 周期ごとにつなぎ合わせた曲線である．

GradualBReoccurring は，時系列モチーフが徐々に成長する．時系列モチーフの形状は上に凸な半周期分の正弦曲線であり，その高さは $(10, 15, 20, \cdots)$ と成長する．時系列モチーフの出現間隔は 1000 ステップ間隔であり $L_{\max}$ より長い．

5.2.2 実データ

交通流，電力，株価のそれぞれの分野における以下のオープンデータを実データとして用いる．

Traffic は，California 州の道路を走行する車両の速度である (注4)．このデータセットは交通流の予測における研究で広く用いられている [5, 8]．観測場所は VDS: 407750 である．1 番目のレーン上の車速を表す Lane 1 Speed (mph) 列のみを用いる．2017/08/13 から 2017/08/26 までの期間のデータを用いる．総ステップ数は 4,031 ステップである．

Stock は，日経株価の終値である (注5)．このデータセットは従来方法 ARWin の評価でも用いられている [7]．[7] と同様に，本データセットのうち終値に対応する Close 列のみを評価に用いる．1997/5/19 から 2017/5/15 までの期間のデータを用いる．総ステップ数は 5,046 ステップである．

Electricity は，New South Wales 州の電力需要量である (注6)．このデータセットはコンセプトドリフトに対応するクラス分類の研究に広く用いられてきた．本研究ではクラスラベルを用いず，nswdemand 列のみを用いる．期間は 1996/05/07 から 1998/12/05 である．総ステップ数は 45,312 ステップである．

5.3 予測性能の比較結果

表 2 は，各方法が各データセットを予測したときの性能を示している．表 2 の括弧内の値は，各データセットの後半の期間において，式 1 の予測誤差を平均した値（平均誤差）である．様々な分野のデータセットを用いるため，データセットによって平均誤差のスケールに違いが生じる．そこで各データセットに対して，提案方法の平均誤差を 1 としたときの比較方法の平均誤差の倍率を評価する．表 2 の括弧以外の数値はその倍率を表している．その倍率を全データセットで平均した値を表 2 の Overall に示す．全データセットにわたる各方法の性能を Overall の値で評価する．表 2 の太字は，各データセットまたはデータ全体で最も性能の良かった場合を表している．NRWin*は固定のウインドウ長に*を用いた場合における従来方法 NRWin を表す．NAWin は各データセットにおいて最も性能が良かった固定のウインドウ長を用いた場合を示す．

データセット全体では，提案方法 (FullFunc, NoCluster) が最良の性能を示した．提案方法の中では，FullFunc，NoCluster, NoLTM の順に性能が良く，短期記憶と長期記憶を分けることと，長期記憶を矛盾の無い複数のクラスタに分けることの有効性をそれぞれ確認した．

比較方法の中では，NRWin が FullFunc と NoCluster に次いで性能が良かった．しかし，データセットにより最適なウインドウ長は異なるため，そのチューニングがデメリットとしてあげられる．一方，提案方法では，NRWin と同じく類似波形による局所回帰を用いながら，ウインドウ長の適応的な自動チューニングにより，そのデメリットを克服できる．ARWin は提案方法と同じくウインドウ長のチューニングを必要としない．しかし， ARWin は GradualBReoccurring 以外のデータセットでは提案方法と NRWin より予測性能は良くなかった．NAWin は Electricity のデータセットでは全ての方法の中で最高の性能を示したが他のデータセットでは性能は良くなかった．人工データでは，GradualAReoccurring と GradualBReoccurring のデータセットで NRWin と比べて NAWin の性能は大きく低下した．これは，傾向が徐々に変化する場合に履歴のみでなく 4.2 節で述べた方法のように履歴と回帰を組み合わせることが有効であることを示している．KRWin はストリーム全体の傾向が滑らかに変化する GradualAReoccurring では性能が良かったが，データセット全体では最も性能が悪かった．以上の従来方法の比較により，類似波形の局所回帰における予測性能への有効性を確認した．

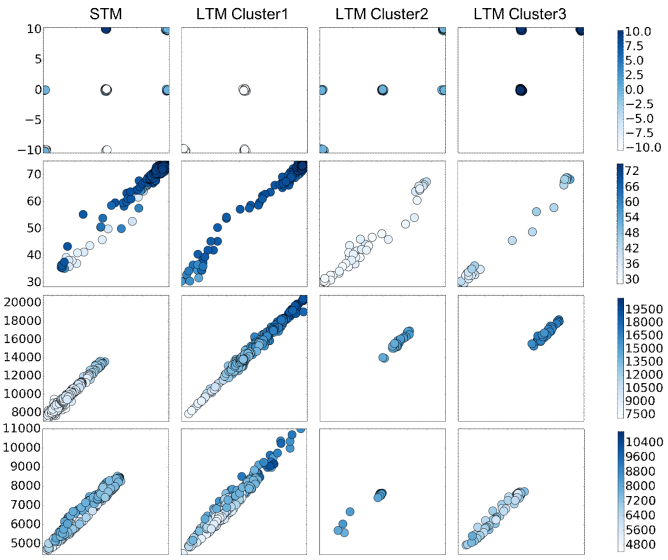

図 4 STM または LTM の各クラスタのサンプル集合（1 行目:Sudden-Reoccurring, 2 列目:Traffic, 3 行目:Stock, 4 行目:Electricity）

5.4 記憶管理の結果

図 4 は，短期記憶（STM）と長期記憶の 3 つのクラスタ（LTM Cluster1, LTM Cluster2, LTM Cluster3）に保持されたサンプルを表示している．1–4 行は，それぞれ SuddenReoccurring, Traffic, Stock, Electricity のデータセットの場合を示している．サンプル $(\mathbf{x}_{i-D+1}, \cdots, \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+N})$ に対して，丸点の座標が $(\mathbf{x}_{i-1,1}, \mathbf{x}_{i,1})$ に対応し，丸点の色が $\mathbf{x}_{i+N,1}$ に対

(注4)：http://pems.dot.ca.gov/
(注5)：https://finance.yahoo.com/quote/%5EN225/
(注6)：https://moa.cms.waikato.ac.nz/datasets/

表 2　予測性能の比較結果

	SuddenReoccurring	GradualAReoccurring	GradualBReoccurring	Traffic	Stock	Electricity	Overall
FullFunc	1.00 (1.8)	**1.00 (0.074)**	**1.00 (0.32)**	**1.00 (0.47)**	**1.00 (94.8)**	1.00 (535)	**1.00**
NoCluster	1.06 (1.9)	1.03 (0.076)	1.03 (0.33)	1.04 (0.49)	1.12 (106)	0.99 (531)	1.04
NoLTM	1.17 (2.1)	1.23 (0.091)	1.09 (0.35)	1.09 (0.51)	1.47 (139)	0.99 (531)	1.17
ARWin	1.44 (2.6)	1.35 (0.10)	1.06 (0.34)	1.64 (0.77)	2.91 (276)	1.20 (643)	1.60
KRWin	1.33 (2.4)	1.05 (0.078)	12.81 (4.1)	1.79 (0.84)	6.49 (615)	1.21 (649)	4.11
NRWin200	1.39 (2.5)	1.35 (0.10)	1.09 (0.35)	1.23 (0.58)	1.68 (159)	0.98 (524)	1.29
NRWin250	1.83 (3.3)	1.35 (0.10)	1.12 (0.36)	1.43 (0.67)	1.69 (160)	1.00 (537)	1.40
NRWin350	1.72 (3.1)	1.35 (0.10)	1.09 (0.35)	1.26 (0.59)	1.71 (162)	1.00 (537)	1.36
NRWin500	0.94 (1.7)	1.35 (0.10)	1.06 (0.34)	1.32 (0.62)	1.59 (151)	1.04 (554)	1.22
NRWin1000	**0.89 (1.6)**	1.31 (0.097)	1.06 (0.34)	1.19 (0.56)	1.46 (138)	1.06 (566)	1.16
NAWin	1.22 (2.2)	6.49 (0.48)	4.69 (1.5)	2.66 (1.25)	4.77 (452)	**0.95 (507)**	3.46

応する．

1 行目の SuddenReoccurring のデータセットにおける LTM を見ると，原点 (i.e.,$(\mathbf{x}_{i-1,1}, \mathbf{x}_{i,1}) = (0, 0)$) 付近のサンプルの $\mathbf{x}_{i+N,1}$ が-10 付近と 0 付近と 10 付近の値で Cluster1 と Cluster2 と Cluster3 に分かれて描画されている．これは，SuddenReoccurring において，0 から-10 の方向へ急激に変化する場合と 0 付近から変化しない場合と 0 から+10 の方向に急激に変化する場合とを複数のクラスタに分けて管理できていることを表している．

2–4 行目の実データに着目する．2 行目では，サンプル間で $(\mathbf{x}_{i-1,1}, \mathbf{x}_{i,1})$ が類似しているにも関わらず $\mathbf{x}_{i+N,1}$ が異なる場合に，それらを別々のクラスタに分けて管理していることが分かる．また，3–4 行目では，STM にはサンプルの無い $(\mathbf{x}_{i-1,1}, \mathbf{x}_{i,1})$ の領域（STM の右上）であっても，LTM ではその領域（LTM Cluster1 の右上）にサンプルがあることを確認できる．これは，短期記憶から古くなり消えた情報を長期記憶で圧縮して管理できていることを表している．

以上により，提案方法の記憶管理が期待通りに動作していることを確認した．

6. おわりに

本研究では，様々なコンセプトドリフトを持つストリームのオンライン予測方法を提案した．予測器は類似波形の局所回帰に基づき，記憶管理については様々なコンセプトドリフトに対応するクラス分類方法 [2, 3] における適応型記憶のアイデアを発展させた．提案方法では，短期記憶と長期記憶を用いた記憶管理に加えて，長期記憶を複数のクラスタにより管理し，クラスタ内に矛盾が無いようにクラスタリングを行えることを確認した．急激に変化するタイプ，ストリームの全体や一部が徐々に変化するタイプ，類似した変化が繰り返し発生するタイプのコンセプトドリフトを持つ人工データと，交通流，株価，電力消費量の実データを用いた評価により，提案方法ではデータセットごとにメタパラメータを変えることなくコンセプトドリフトに対応する複数のオンライン予測方法と比較して高い予測精度を達成することを確認した．

文　献

[1] Gama, J. a., Žliobaitė, I., Bifet, A., Pechenizkiy, M. and Bouchachia, A.: A Survey on Concept Drift Adaptation, *ACM Comput. Surv.*, Vol. 46, No. 4, pp. 44:1–44:37 (2014).

[2] Losing, V., Hammer, B. and Wersing, H.: KNN Classifier with Self Adjusting Memory for Heterogeneous Concept Drift, *ICDM*, IEEE Computer Society, pp. 291–300 (2016).

[3] Losing, V., Hammer, B. and Wersing, H.: Self-adjusting Memory: How to Deal with Diverse Drift Types, *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, AAAI Press, pp. 4899–4903 (2017).

[4] Pan, B., Demiryurek, U. and Shahabi, C.: Utilizing Real-World Transportation Data for Accurate Traffic Prediction, *ICDM*, IEEE Computer Society, pp. 595–604 (2012).

[5] Jeong, Y.-S., Byon, Y.-J., Castro-Neto, M. M. and Easa, S. M.: Supervised Weighting-Online Learning Algorithm for Short-Term Traffic Flow Prediction, *Trans. Intell. Transport. Sys.*, Vol. 14, No. 4, pp. 1700–1707 (2013).

[6] Van Vaerenbergh, S., Vía, J. and Santamaría, I.: Nonlinear System Identification using a New Sliding-Window Kernel RLS Algorithm, *Journal of Communications*, Vol. 2, No. 3, pp. 1–8 (2007).

[7] Yoshida, S., Hatano, K., Takimoto, E. and Takeda, M.: Adaptive Online Prediction Using Weighted Windows, *IEICE Transactions*, Vol. 94-D, No. 10, pp. 1917–1923 (2011).

[8] Dell'Acqua, P., Bellotti, F., Berta, R. and Gloria,

A. D.: Time-Aware Multivariate Nearest Neighbor Regression Methods for Traffic Flow Prediction, *IEEE Trans. Intelligent Transportation Systems*, Vol. 16, No. 6, pp. 3393–3402 (2015).

[9] Ban, T., Zhang, R., Pang, S., Sarrafzadeh, A. and Inoue, D.: Referential kNN Regression for Financial Time Series Forecasting, *Neural Information Processing - 20th International Conference, ICONIP*, Lecture Notes in Computer Science, Vol. 8226, Springer, pp. 601–608 (2013).

[10] Al-Qahtani, F. H. and Crone, S. F.: Multivariate k-nearest neighbour regression for time series data - A novel algorithm for forecasting UK electricity demand, *The 2013 International Joint Conference on Neural Networks, IJCNN 2013, Dallas, TX, USA, August 4-9, 2013*, pp. 1–8 (2013).

[11] Dougherty, M. S. and Cobbett, M. R.: Short-term inter-urban traffic forecasts using neural networks, *International Journal of Forecasting*, Vol. 13, No. 1, pp. 21 – 31 (1997).

[12] Bifet, A. and Gavalda, R.: Learning from Time-Changing Data with Adaptive Windowing., *SDM*, Vol. 7, SIAM, pp. 443–448 (2007).

[13] Bifet, A., Pfahringer, B., Read, J. and Holmes, G.: Efficient Data Stream Classification via Probabilistic Adaptive Windows, *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, ACM, pp. 801–806 (2013).

[14] Jaber, G., Cornuéjols, A. and Tarroux, P.: Online Learning: Searching for the Best Forgetting Strategy Under Concept Drift, *Proceedings, Part II, of the 20th International Conference on Neural Information Processing - Volume 8227*, ICONIP 2013, Springer-Verlag New York, Inc., pp. 400–408 (2013).

[15] Bifet, A., Holmes, G. and Pfahringer, B.: Leveraging Bagging for Evolving Data Streams, *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*, ECML PKDD'10, Springer-Verlag, pp. 135–150 (2010).

[16] Matsubara, Y. and Sakurai, Y.: Regime Shifts in Streams: Real-time Forecasting of Co-evolving Time Sequences, *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, ACM, pp. 1045–1054 (2016).

[17] Zhang, P., Gao, B. J., Zhu, X. and Guo, L.: Enabling Fast Lazy Learning for Data Streams, *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, IEEE Computer Society, pp. 932–941 (2011).

[18] Zhang, T., Ramakrishnan, R. and Livny, M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases, *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, ACM, pp. 103–114 (1996).

[19] Arthur, D. and Vassilvitskii, S.: K-means++: The Advantages of Careful Seeding, *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, Society for Industrial and Applied Mathematics, pp. 1027–1035 (2007).

[20] Haworth, J., Shawe-Taylor, J., Cheng, T. and Wang, J.: Local online kernel ridge regression for forecasting of urban travel times, *Transportation Research Part C*, Vol. 46, No. Complete, pp. 151–178 (2014).

[21] Zhu, Y., Imamura, M., Nikovski, D. and Keogh, E.: Matrix Profile VII: Time Series Chains: A New Primitive for Time Series Data Mining, *ICDM*, IEEE Computer Society, p. to be appeared (2016).