

ストリームの適応記憶型オンライン予測

Online prediction of self-adaptive memory over data stream

山口 晃広^{*1} 真矢 滋^{*1} 稲木 達哉^{*1} 植野 研^{*1}
 Akihiro Yamaguchi Shigeru Maya Tatsuya Inagi Ken Ueno

^{*1}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
^{*1}Corporate Research & Development Center TOSHIBA Corporation

We propose online prediction method for data streams with heterogeneous concept drift. The proposed method forecasts short-range future values by integrating both short-term memory of various window sizes and long-term memory of consistent clusters adaptively. On synthetic datasets with heterogeneous concept drift and real-world datasets of traffic flow, stock prices, and electricity demand, experimental results show superior accuracy without optimizing meta-parameters among datasets over several baseline methods.

1. はじめに

IoTの普及に伴い、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。スライディングウィンドウ（以下ウィンドウ）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、ウィンドウの長さは、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、ウィ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手法が提案され、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従来手法と比較して優れた分類性能を達成している [2, 3]。

一方、交通流、株価、電力消費量などのストリームでは、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、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。これらの予測では、ウィ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手法がよく研究されている。線形回帰を用いる手法が良く知られているが、[4] など非線形なカーネル回帰をウィ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複数のウィンドウ長で予測しその結果を適応的に重み付けて予測する手法も提案された [5]。一方、過去の履歴を用いる手法として、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手法が、交通流、株価、電力消費量などの予測に用いられている [6]。

しかしながら、従来のオンライン予測手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。ウィンドウを用いる従来のオンライン予測手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、波形など過去の履歴を用いる従来手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。これに対応する自明な方法として、固定長のウィ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ィンドウ長のチューニングが必要である。[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、様々なコンセプトドリフトに対応するオンライン予測手法を提案する。短期記憶と長期記憶を適応的に用いるクラス分類手法 [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に付けて管理する。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、提案手法ではメタパラメータのチューニング無しに [4, 5] を含む比較手法と比べて高い予測性能が得られることを確認する。

2. 問題設定

ストリームはタプルの列 $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t, \dots)$ であり、タプル \mathbf{x}_i は J 次元ベクトル (i.e., $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,J})$) 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 $\hat{x}_{t+N,j}$ 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 $\hat{y}_{i+N} = \hat{x}_{i+N,j}$ 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 \hat{y}_t と観測値

連絡先: 山口晃広, 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, akihiro5.yamaguchi@toshiba.co.jp

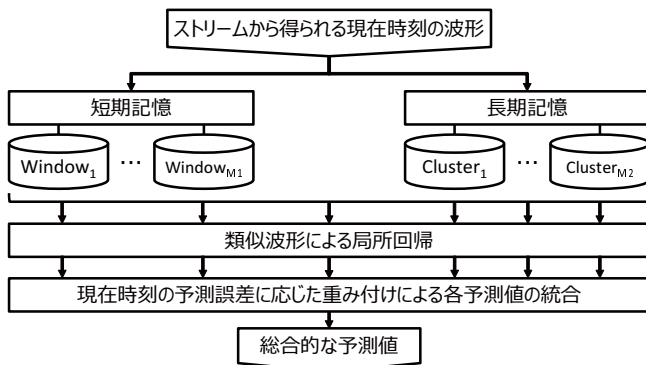


図 1: 提案する予測手法の全体像

y_t との誤差である予測誤差 $(\hat{y}_t - y_t)^2$ を測定できる。

時刻 t における長さ D の波形は $\mathbf{s}_t = (\mathbf{x}_{t-D+1}, \dots, \mathbf{x}_t)$ であり $D \times J$ 次元のベクトルである。記憶には、波形 \mathbf{s}_t 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。

3. 提案手法

本節では、提案手法の全体構成と長期記憶を順に説明する。

3.1 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ィンドウで管理される。[5] と同様に複数のウィンドウ長を用いることで様々なサンプル集合で予測することができる。一方、長期記憶ではウィンドウに入りきらない古いサンプ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。長期記憶については 3.2 節で詳しく述べる。

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、予測手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 $\mathbf{s}_{i_1}, \mathbf{s}_{i_2}, \dots, \mathbf{s}_{i_K}$ をサンプル集合から取り出す。次に、 $\mathbf{s}_{i_1}, \mathbf{s}_{i_2}, \dots, \mathbf{s}_{i_K}$ から $y_{i_1+N}, y_{i_2+N}, \dots, y_{i_K+N}$ を推定する回帰式を求め、その回帰式に現時刻の波形 \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手法 [6] と同様のアイデアである。但し、従来手法と異なり本研究ではウィ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回帰式には正則化項を加えたリッジ回帰を用いる。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 [7]。

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、その結果を適応的に重みづけて統合し最終的な予測値を出力する。予測に用いるサンプル集合は、異なるウィンドウ長で区切られた短期記憶のサンプル集合、長期記憶の各クラスタを構

成するサンプル集合、長期記憶の全体のサンプル集合、短期記憶と長期記憶の全体のサンプル集合のうち、サンプル数が一定数 (L_{\min}) 以上のものである。各サンプル集合を用いて現時刻 t から N ステップ前の波形 \mathbf{s}_{t-N} を用いて予測値 \hat{y}_t を算出し、[5] のように予測誤差が小さいサンプル集合を重視するように重み付けを決定する。次に、各サンプル集合に対して現時刻 t の波形 \mathbf{s}_t を用いて改めて予測して重み付け平均を最終予測値とする。このように、現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、メタパラメータのチューニング無しに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。

3.2 長期記憶

短期記憶から溢れた古いサンプルは長期記憶に保持される。長期記憶はクラスタ内に矛盾が無いように複数のクラスタで分割管理され、各クラスタではサンプルを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。

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。

3.2.1 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 1 で定義し、k-means++[8] のクラスタリング手法を用いて M_{LTM} 個のクラスタを生成する。

$$|y_{i_1+N} - y_{i_2+N}| \exp\left(\frac{-\alpha \|\mathbf{s}_{i_1} - \mathbf{s}_{i_2}\|^2}{\text{var}(\{\mathbf{s}_i\}_{(\mathbf{s}_i, y_{i+N}) \in LTM_{\text{all}}})}\right), \quad (1)$$

ここで、 LTM_{all} は長期記憶に保持されているサンプル集合であり、 $\text{var}(\cdot)$ は分散を表し、 α は正の定数である。

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。 $m = 1, 2, \dots, M_{LTM}$ 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、 $\frac{L_{\max}}{M_{LTM}} \leq |LTM_m|$ ならばマイクロクラスタ数が $\frac{L_{\max}}{2M_{LTM}}$ 個となるように LTM_m に k-means++[8] 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 はマイクロクラスタの重心からなる集合に置き換えられる。

3.2.2 クラスタの逐次更新処理

本処理では、3.2.1 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 (サンプル数が L_{\min} 以上の場合)、類似波形による局所回帰で \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 (サンプル数が L_{\min} 未満の場合)、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。

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正則化項の係数
$L_{\min} = 200$	予測に用いられる最小サンプル数
$L_{\max} = 500$	各記憶に保持できる最大サンプル数
$M_{LTM} = 3$	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 1 の定数

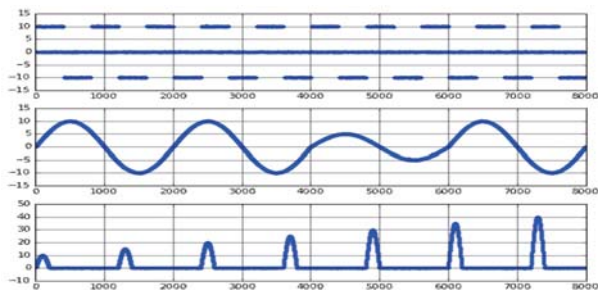


図 2: 人工データセット (1 行目: SuddenReocc, 2 行目: GradualAReocc, 3 行目: GradualBReocc)

4. 実験評価

本実験全体を通して、提案手法のメタパラメータには表 1 の値を統一して用いる。短期記憶のウィンドウ長は 200, 250, 350, 500 からなり、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。

4.1 比較手法

本実験では、3. 節で述べた全ての機能を実現する提案手法 (FullFunc) を以下の手法と比較する。

NoCluster は、提案手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。**NoLTM** は、提案手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。短期記憶のウィンドウ長は最大 $2L_{\max}$ まで適応的に変える。**ARWin** は、最近のデータをウィンドウに保持し適応的にウィンドウ長を変えて線形回帰を行う従来手法である [5]。ウィンドウ長は 3, 4, ..., 10 の範囲で変える。**KRWin** は、最近のデータを固定長のウィ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手法である [4]。各データセットに対して 1000–2000 ステップの期間で精度が最良のメタパラメータを選ぶ。**NRWin** は、固定のウィンドウ長を用いた類似波形による局所回帰であり、従来手法 [6] からドメイン固有の部分を取り除いた手法である。ウィンドウ長は提案手法で変更する範囲のウィンドウ長か $2L_{\max}$ である。

4.2 データセット

4.2.1 人工データ

(Sudden) 急激に変化するタイプ、(GradualA) ストリーム全体が徐々に変化するタイプ、(GradualB) 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、(Reocc) 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図 2 に示す以下の 3 つの人工データを用いる。

SuddenReocc 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 $(0, \pm 10, 0, \dots)$ を繰り返す。ここで記号 (\pm) は y_t が 0 と ± 10 へ 10 回変化するごとに符号を反転する。

GradualAReocc 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。

GradualBReocc 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 $(10, 15, 20, \dots)$ 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。

4.2.2 実データ

交通流、電力、株価における以下の実データを用いる。

Traffic は、California 州の道路を走行する車速度である *1。観測場所は VDS: 407750 であり、1 番目のレーンを表す Lane 1 Speed (mph) 列のみを用いる。2017/08/13–2017/08/26 の期間を用いて、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 *2。[5] と同様に終値に対応する Close 列のみを評価に用いる。1997/5/19–2017/5/15 の期間のデータを用いて、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 *3。本研究では nswdemand 列のみを用いる。期間は 1996/05/07–1998/12/05 で、総ステップ数は 45,312 ステップである。

4.3 予測性能の比較結果

表 2 は、各手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ける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、提案手法の平均誤差を 1 としたときの比較手法の平均誤差の倍率を評価し、表 2 の括弧以外の数値はその倍率を表す。その倍率を全データセットで平均した値を表 2 の Overall に示し、各手法の性能を Overall の値で評価する。NRWin* は固定のウィンドウ長に*を用いた場合の従来手法 NRWin である。

データセット全体では、提案手法 (FullFunc, NoCluster) が最良の性能を示した。提案手法の中では、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。

従来手法の中では、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ィ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、提案手法では、NRWin と同じく類似波形による局所回帰を用いながら、ウィ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 は提案手法と同じくウィンドウ長のチューニングを必要としない。しかし、ARWin は GradualBReocc 以外のデータセットでは提案手法と NRWin より予測性能は良くなかった。KRWin はストリーム全体の傾向が滑らかに変化する GradualAReocc 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。以上の従来手法の比較により、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。

4.4 記憶管理の結果

図 3 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 SuddenReocc, Traffic, Stock, Electricity のデータセットの場合を示してい

*1 <http://pems.dot.ca.gov/>*2 <https://finance.yahoo.com/quote/%5EN225/>*3 <https://moa.cms.waikato.ac.nz/datasets/>

表 2: 予測性能の比較結果

	SuddenReocc	GradualAReocc	GradualBReocc	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

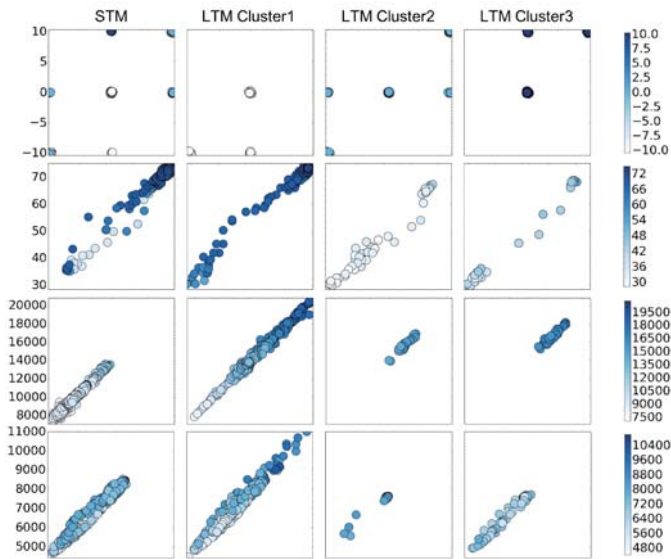


図 3: 短期記憶 (STM) または長期記憶 (LTM) の各クラスターのサンプル集合 (1 行目: SuddenReocc, 2 列目: Traffic, 3 行目: Stock, 4 行目: Electricity)

る。サンプル $(\mathbf{x}_{i-D+1}, \dots, \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+N})$ 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 $\mathbf{x}_{i+N,1}$ に対応する。

1 行目の SuddenReocc 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、SuddenReocc 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ー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かかわらず $\mathbf{x}_{i+N,1}$ が異なる場合に、それらを別々のクラスター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。

5. 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手法を提案した。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手法 [2, 3] における適応型記憶のアイデアを発展させた。提案手法では、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ーにより管理し、クラスター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、提案手法ではメタパラメータのチューニング無しにコンセプトドリフトに対応する従来手法と比較して高い予測精度を達成することを確認した。

参考文献

- [1] Gama, J. a., Žliobaitė, I., Bifet, A., Pechenizkiy, M. and Bouchachia, A.: A Survey on Concept Drift Adaptation, *CSUR*, 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*, pp. 291-300 (2016).
- [3] Losing, V., Hammer, B. and Wersing, H.: Self-adjusting Memory: How to Deal with Diverse Drift Types, *IJCAI*, pp. 4899-4903 (2017).
- [4] 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).
- [5] Yoshida, S., Hatano, K., Takimoto, E. and Takeda, M.: Adaptive Online Prediction Using Weighted Windows, *IEICE Trans*, Vol. 94-D, No. 10, pp. 1917-1923 (2011).
- [6] Dell'Acqua, P., Bellotti, F., Berta, R. and Gloria, A. D.: Time-Aware Multivariate Nearest Neighbor Regression Methods for Traffic Flow Prediction, *IEEE Trans. ITS.*, Vol. 16, No. 6, pp. 3393-3402 (2015).
- [7] Zhang, P., Gao, B. J., Zhu, X. and Guo, L.: Enabling Fast Lazy Learning for Data Streams, *ICDM*, pp. 932-941 (2011).
- [8] Arthur, D. and Vassilvitskii, S.: K-means++: The Advantages of Careful Seeding, *SODA*, pp. 1027-1035 (2007).