DEIM Forum 2020 I6-5

時系列データを用いたデータストリームマイニングアルゴリズムの性能評価手法の検討

Examination of Performance Evaluation Method in Data Stream Mining in Time Series Data

丸尾　海月†　　新美　礼彦†

† 公立はこだて未来大学　041–8655 北海道函館市亀田中野町 116 番地 2
E-mail: †{b1016149,niimi}@fun.ac.jp

あらまし　データストリームマイニングのアルゴリズムを評価する場合，静的なデータを加工してデータストリームとして利用することがある．しかしデータストリームには局所的に変化する性質のものもあるため，静的なデータを加工する方法では正しく性能が評価できない可能性が考えられる．本稿では時系列データを擬似データストリームとして扱い，この問題の解決を図る．提案手法の有効性を検証するため，静的データセットと時系列データセットを用いて，データストリームマイニングアルゴリズムに適用し，提案手法の利点と欠点を考察した．

キーワード　時系列データ，データストリーム，性能評価

1　はじめに

昨今の情報過多に伴い，得られる全てのデータをデータベースに入力し，それら全てを用いて解析するのは困難となっている．IDC の調査によると，2017 年時点で，世界のデータ量の 15%はデータストリームであり，2025 年には 30%のデータがデータストリームになるであろうという予測 [1] がされている．

データストリームは動的な大規模データであり，従来の静的なデータを対象としたデータマイニング手法をそのまま適用することは困難である．そのため，動的なデータを対象としたアルゴリズムが古くから研究されている．今後更に増加するであろうデータストリームを分析するために，データストリームマイニングに関する研究が盛んになると考えられる．

データストリームマイニングでは，次々にやってくるデータストリームを処理しなければならない．そのため，従来の静的なデータを対象としたデータマイニングのように，十分な時間とリソースを掛けることができない．そこで，データストリームマイニングでは，厳密解を求めることを諦めて，近似解を求めるという手法が取られている．

データストリームを対象としたアルゴリズムを評価する場合，最も望ましいのはデータストリームを使用することである．しかし，データストリームを用いる場合は前処理に十分な時間を割くことが出来ないという問題や，データストリームを調達してくるのが難しいという問題が存在する．アルゴリズムを評価する際，重要なのはアルゴリズムであってデータではない．そのため，データは軽視されがちであり，データストリームを対象としたアルゴリズムであったとしても，静的なデータを用いるか，自作のデータストリームを使うことがしばしば存在する．静的なデータを用いる場合，アルゴリズムにはデータストリームを用いる場合と同じように，逐次的にデータが入力される．静的なデータを対象とするデータマイニングアルゴリズムでは，全てのデータを使用してパラメータを更新するため，データの順序が結果に影響することがない．しかし，データストリームマイニングアルゴリズムでは全てのデータを使用せずにパラメータを更新するため，データの順序によって結果が異なってしまうという問題点が生じる．

本稿では，静的なデータセットを用いてデータストリームマイニングアルゴリズムを評価する場合，正しく性能を評価出来ているのかを検証し，時系列データセットを用いてデータストリームマイニングアルゴリズムを性能評価できるかを検討する．

1.1　データストリームと時系列データの違い

本稿では，データストリームと時系列データを以下のように定義する．有村らは「データストリームは膨大な量のデータが，高速なストリームを通じて，時間的に変化しながら，終わりなく到着しつづけるもの」[2] と定義している．時系列データは，データストリームがいつ到着したかという情報を付加して保存したものとする．つまり，データストリームがデータベース上に保存された時点で，時系列データになったとみなす．

データストリームには，時間的に変化するという性質があるため，時系列データも時間的に変化するという性質を保持していると考える（表 1 参照）．データストリームには終わりなく到着しつづけるという性質があるため，無限に生成されていくと考えられるが，時系列データでは，データベース上に保存されているデータが全てだと考えるため，データストリームが保有していた，終わりなく到着しつづけるという性質は失われる．加えて，データストリームは動的なデータであったが，時系列データはデータベース上に保存された状態から変化することがないため，静的なデータに変わる．

表 1　データストリームと時系列データの特徴

	データストリーム	時系列データ
データの性質	動的	静的
保存するか	しない	する
有限か	無限	有限

1.2　時系列データと静的なデータの違い

時系列データとは，時系列情報を含むデータのことである．前項で述べたように，時系列データは時間的に変化するという性質を保持していると考える．時系列データから，時系列情報を削除することで，静的なデータとなると考える．静的なデータにした場合，時間的に変化するという性質は失われ，時系列データが保持していたデータの前後関係は消滅する（表 2 参照）．

表 2　時系列データと静的なデータの特徴

	時系列データ	静的なデータ
時系列情報	保有する	保有しない
前後関係	保有する	保有しない

1.3　一般のデータストリームマイニングアルゴリズムのベンチマーク用データセットの作り方

一般的なデータストリームマイニングアルゴリズムでは，ベンチマークに使用されているデータセットは，静的なデータを加工して作られている場合や，各々の研究者が作成している場合が多い．静的なデータを加工してデータストリームマイニングアルゴリズムを性能評価する場合，アルゴリズムにデータを渡す順番を指定する必要がある．

1.4　問　題　点

静的なデータを対象とするデータマイニングアルゴリズムでは，全てのデータを使用してパラメータを更新するため，データの順序が結果に影響することがない．しかし，データストリームマイニングアルゴリズムでは全てのデータを使用せずにパラメータを更新するため，データの順序によって結果が異なってしまうという問題点が生じる．静的なデータには，データに前後関係の情報は含まれてはいない．しかし，データストリームマイニングアルゴリズムではデータを流し込む順番が結果を左右してしまうため，静的なデータを使用してデータストリームマイニングアルゴリズムをベンチマークする場合，静的データにあるはずのない前後関係を考慮してデータマイニングがなされてしまうという問題点が生じる．時系列データであれば，データの前後関係が保持されていると考えられる．

よって，時系列データであれば，データストリームマイニングアルゴリズムを正しく性能評価が出来るのではないかと考える．

2　関 連 研 究

データストリーム中の頻出アイテムを近似的に抽出するオンライン型アルゴリズムである Lossy Counting Algorithm に先頭系列頻度を用いて，頻出部分系列を高速近似抽出するオンラインアルゴリズムを提案しているが，評価実験では人工的に作成したデータを用いている [3]．

逆単調性を満たす全体頻度なる出現頻度を用いて，大規模時系列データを対象とした頻出パターンのオンライン型高速抽出アルゴリズムを提案しているが，評価実験では乱数で作成したアイテム集合を用いている [4]．

Frequent に，k-reduced bag の考えを用いて改良した KRB というアルゴリズムを提案しているが，評価実験では zipf 則に基づく人工データを用いている [5]．

3　提 案 手 法

時系列データセットを用いてデータストリームマイニングアルゴリズムの性能評価を行う方法を提案する．時系列順にデータストリームとすることで，実在するデータストリームに近い疑似データストリームをアルゴリズムに与えることが出来ると考える．時系列データはデータストリームをデータベース上に保存したものなので，よりデータストリームに近いデータを用いてアルゴリズムを評価することができる．

提案手法を評価するために，静的なデータセットを用いてデータストリームマイニングアルゴリズムを性能評価する場合と比較するために実験を行う．本稿では，データストリームマイニングアルゴリズムとして，オンラインアルゴリズムの Online Passive-Aggressive [6] を用いて実験を行う．

Online Passive-Aggressive は実装が容易であり， オンラインアルゴリズムの特徴であるノイズに弱いという特徴を持っている．ノイズに弱いということは，データに対して機敏に反応するため，本実験に適していると判断した．

4　Online Passive-Aggressive

Online Passive-Aggressive はオンライン学習における教師あり学習の一手法である．Online Passive-Aggressive では，ヒンジ損失関数を用いて最適化問題を逐次的に解いていき，更新する重みベクトルを求める．

入力には，データ点集合 $\mathbf{x}_i$ と正解ラベル y_i が与えられる．データが入力された時点での重みベクトル $\mathbf{w}_i$ を用いて，入力されたデータ点 d_i が正解であるか，不正解であるかを予測し，正解ラベル y_i と比較して重みベクトル $\mathbf{w}_{i+1}$ を更新する．ヒンジ損失関数は以下の通りである．

$$\ell_{\text{hinge}}(\mathbf{x}_i, y_i, \mathbf{w}_i) = \max\left(0, 1 - y_i \mathbf{w}_i^\top \mathbf{x}_i\right)$$

重みベクトルの更新するための最適化問題は以下の通りである．

$$\mathbf{w}_{i+1} = \arg\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_i\|^2$$

subject to

$$\ell_{hinge}(\mathbf{x}_i, y_i, \mathbf{w}_i) = 0$$

この最適化問題から，重みの更新式は以下のようになる．

$$\mathbf{w}_{i+1} = \begin{cases} \mathbf{w}_i + \frac{\ell_{\text{hinge}}(\mathbf{x}_i, y_i, \mathbf{w}_i)}{\|\mathbf{x}_i\|^2} y_i \mathbf{x}_i & (\ell_{hinge}(\mathbf{x}_i, y_i, \mathbf{w}_i) > 0) \\ \mathbf{w}_i & (otherwise) \end{cases}$$

4.1 Online Passive-Aggressive-I

Online Passive-Aggressive では，ヒンジ損失関数が 0 となる重みに更新する．そのため，ノイズに対して敏感であり，それまで学習した結果を破棄して，ノイズに合わせて大きく学習してしまうという問題点があった．そこで，ある程度の誤りを許容する，Online Passive-Aggressive-I という方法が提案された．どの程度の誤りを許容するかというパラメータ $C > 0$ を指定する必要がある．重みベクトルの更新するための最適化問題は以下の通りである．

$$\mathbf{w}_{i+1} = \arg\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_i\|^2 + C\xi$$

subject to

$$\ell_{hinge}(\mathbf{x}_i, y_i, \mathbf{w}) \leqq \xi \quad and \quad \xi \geqq 0$$

この最適化問題から，重みの更新式は以下のようになる．

$$\mathbf{w}_{i+1} = \begin{cases} \mathbf{w}_i + \frac{\min(C, \ell_{\text{hinge}}(\mathbf{x}_i, y_i, \mathbf{w}_i))}{\|\mathbf{x}_i\|^2} y_i \mathbf{x}_i & \\ & (\ell_{hinge}(\mathbf{x}_i, y_i, \mathbf{w}_i) > 0) \\ \mathbf{w}_i & (otherwise) \end{cases}$$

4.2 Online Passive-Aggressive-Ⅱ

Online Passive-Aggressive-I では，ヒンジ損失関数が 0 でなかった場合のペナルティを線形に考えていたが，Online Passive-Aggressive-Ⅱ では，ペナルティを 2 乗で考える．重みベクトルの更新するための最適化問題は以下の通りである．

$$\mathbf{w}_{i+1} = \arg\min_{\mathbf{w}} \frac{1}{2} \left\|\mathbf{w} - \mathbf{w}_{(i)}\right\|^2 + C\xi^2$$

subject to

$$\ell_{hihge}\left(\mathbf{x}_i, y_i, \mathbf{w}\right) \leqq \xi$$

この最適化問題から，重みの更新式は以下のようになる．

$$\mathbf{w}_{i+1} = \begin{cases} \mathbf{w}_i + \frac{\ell_{\text{hinge}}(\mathbf{x}_i, y_i, \mathbf{w}_i)}{\|\mathbf{x}_i\|^2 + \frac{1}{2C}} y_i \mathbf{x}_i & (\ell_{hinge}(\mathbf{x}_i, y_i, \mathbf{w}_i) > 0) \\ \mathbf{w}_i & (otherwise) \end{cases}$$

5 実　　験

本実験では，実験を簡潔にするため，2 クラス分類問題を想定して行う．静的なデータセットを用いて，データストリームマイニングアルゴリズムを学習した場合と，時系列データセットを用いてデータストリームマイニングアルゴリズムを学習した場合との，再現率及び適合率の推移を比較する．

静的なデータを用いてデータストリームマイニングアルゴリズムを実行する時，データをアルゴリズムに与える順序を決める必要がある．仮にランダムに決めるとした場合，正例と負例が極端に配置された順序になる可能性は否定できない．正例と負例が極端に配置された順序の場合でも，アルゴリズムが正しく評価できているかを検証するために，与える順序がどの程度影響するのかを検証するために，静的データに下記のような加工を施したデータを用いて実験を行う．

- ターゲットラベルに対して昇順でソート (先に負例)
- ターゲットラベルに対して降順でソート (先に正例)
- データの順序をランダマイズ

時系列データは，時系列自体がデータを与える順序を担うと考える．本稿の定義では，時系列データから時系列情報を削除することで静的データとなるので，時系列データから時系列を無視して静的データとして扱った場合．どの程度再現率と適合率に影響するのかを検証する．時系列データに行う加工は以下の通りである．

- 時系列を無視し，ターゲットラベルに対して昇順でソート (先に負例)
- 時系列を無視し，ターゲットラベルに対して降順でソート (先に正例)
- 時系列を無視し，ランダマイズ

加工を行わない場合は，静的なデータではインデックス順，時系列データでは時系列順の順序となる．

5.1 対象のデータストリームマイニングアルゴリズム

本稿では，オンラインアルゴリズムを対象に実験を行う．オンライン学習とは，データが 1 つずつ逐次的に与えられる状況下において，データが与えられる度にパラメータを更新する学習方法である．実験で使用するアルゴリズムは，以下の通りである．

- Online Passive-Aggressive
- Online Passive-Aggressive-I
- Online Passive-aggressive-Ⅱ

実装は Python で行った．

5.2 使用するデータセット

本実験では，静的なデータセットとして，Fisher の Iris Data Set，時系列データセットとして，UEA & UCR Time Series Classification Repository にある，クラス数が 2 のデータセットを用いて実験を行う．

5.3 実験に当たっての前処理

Fisher の Iris Data Set は 3 クラスのデータであるが，Online Passive-aggressve でクラス分類を簡潔に行うため，3 クラスのデータを 2 クラスの組に変更し．学習データと評価データに分割する．

5.4 提案手法のパラメータ設定

今回の実験では，データを擬似データストリームとして扱う．そのため，エポック数は 1 となる．Online Passive-Aggressive-I 並びに，Online Passive-Aggressive-Ⅱ では， 誤りの許容量 $C > 0$ をパラメータとして指定するが，今回は $C = 0.1$ とする．

6 ま と め

実験は未完了であるため，今後は Online Passive-Aggressive を対象に，静的データと時系列データを用いて，評価実験を行い，再現率と適合率の推移を比較する．

予想される結果としては，静的データにおいて，正例と負例を極端に配置した場合は，先に来る例に合わせて学習し，安定した後に逆の例に変わるため，変化した直後に大きく重みが変化すると考えられる．その後は同じ例しか来ないため安定し，大きく重みが変化することはないと考えられる．Online Passive-Aggressive は，今までの重みで上手く分類ができないデータが到着すると，今までの学習結果を忘れるほど大きく重みを変更するため，極端に正例と負例を配置した場合，後に来た例に対してのみ大きく学習するため，先に来た例に対しての再現率や適合率は低くなると考えられる．次に，静的データにおいて，正例と負例を均等に配置した場合，始めは不安定に重みが更新されるかもしれないが，徐々に安定していき，重みの変化量は振動が減衰するように徐々に小さくなっていくと考えられる．Passive-Aggressive には，正しく分類できている場合でも自信がない場合は重みを更新するという特性があるため，分類が出来た後も重みを更新すると考えられるからである．

次に，正例と負例をランダマイズした場合，始めは不安定に重みが更新されるが，その後は大きくは更新されず，不定期に重みが変化するのではないかと考えられる．

時系列データを用いた場合を想定する．時系列データを用いる場合，時系列データには前後関係の情報が含まれているため，重みの変化の収束が静的データを用いる場合よりも速いのではないかと考える．時系列データの時系列を無視して実験を行った場合は，時系列データに含まれていた前後関係の情報が無くなるため，静的なデータを使う場合と大差が無くなるのではないかと予測する．

本稿では．時系列データを用いてデータストリームマイニングアルゴリズムを評価する方法を提案した．実験が未完了であるため，今後は時系列データを使用することの優位性を示すため，本稿で示した実験を行う．

文　献

[1] Reinsel David, Gantz John, and Rydning John. The digitization of the world from edge to core, 2018. An IDC White Paper.

[2] 博紀有村, 拓也喜田. データストリームのためのマイニング技術. 情報処理, Vol. 46, No. 1, pp. 4–11, jan 2005.

[3] 順平村田, 宏治岩沼, 龍一石原, 英知鍋島. F-043 精度保証付きオンライン型高速近似系列マイニング (人工知能・ゲーム, 一般論文). 情報科学技術フォーラム講演論文集, Vol. 8, No. 2, pp. 499–503, aug 2009.

[4] 龍一石原, 宏治岩沼, 英知鍋島. Lf-002 大規模データ系列中に頻出する部分系列のオンライン抽出アルゴリズム (f 分野:人工知能・ゲーム). 情報科学技術レターズ, No. 4, pp. 89–92, aug 2005.

[5] 直弥鳥谷部, 拓也喜田. データストリームに対する効率良い頻出アイテム発見アルゴリズム. DEIM Forum 2019 D4-1 6 ページ, 2019.

[6] Shai Shalev-shwartz, Koby Crammer, Ofer Dekel, and Yoram Singer. Online passive-aggressive algorithms. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pp. 1229–1236. MIT Press, 2004.