D-012

コンセプトドリフトが想定される環境でのオンライン K-medoids クラスタリングの代表データ選択

Medoids Selection for online K-medoids clustering under expected concept drifts

田山 凜太郎 *
Rintaro Tayama
t2331098@gl.cc.uec.ac.jp

古賀 久志 *
Hisashi Koga
koga@sd.is.uec.ac.jp

1 はじめに

IoT 技術の普及に伴いセンサーデータやネットワークトラフィックデータなどストリームデータの解析が重要となっている．クラスタリングは非教示でデータ特徴を捉えられるデータマイニング手法である．ストリームデータを対象とする際には，新しいデータが時間経過とともに到達するためオンラインでクラスタリングすることが求められる．とくに，データの到着スピードに追従するため，オンライン処理ではデータの集約処理のみを高速に行い，最終的なクラスタリング結果はオフラインで決定する手法が多い [1].

またストリームデータにおいて，連続して到着する複数個のデータで構成されるデータ系列をシーケンスデータと呼ぶ．シーケンスデータはオンライン不正検知や人間の行動分析において特に重要である．シーケンスデータ間の (非) 類似度には DTW (Dynamic Time Warping) などの非計量な距離尺度が使われることも多い．このため，シーケンスデータのオフラインクラスタリングでは，K-means 法よりも PAM(Partitioning Around Medoids)[2] のような K-medoids 法が適している．

近年，Nadeem らは K-medoids 法をオンライン環境に拡張し，ストリームデータに含まれるシーケンスデータをオンラインクラスタリングするアルゴリズム SECLEDS[3] を提案した．SECLEDS の主要な特徴は以下の 2 つである．

- 計算量が小さく，リアルタイムでシーケンスデータをクラスタに割り当てる完全なオンラインアルゴリズムである．
- データ分布が時間経過に伴って変化するコンセプトドリフトに適応してクラスタ形状を変化させられる．

より具体的には，SECLEDS はクラスタごとに複数の代表データを持ち，到着したシーケンスデータからの投票によって代表データの取捨選択を行う．

本研究では，SECLEDS の代表データ選択方法がコンセプトドリフトに適応するためにノイズに弱くなっていることを指摘し，コンセプトドリフトに適応しつつノイズにもロバストな代表データ選択方法を 2 個提案する．1 つ目は既存の代表データの得票数が指定された下限値を越えていれば，代表データから除去しない MVT(Minimum Vote Threshold) という手法である．2 つ目は，新データを代表データにするか否かを到着時には判定しない TSND(Temporary Storage for New Data) という手法である．ドリフトが含まれる通信データセットである KDD Cup 1999 Data[4] を用いた実験的に評価した結果，提案手法は SECLEDS に対して 20%程度の精度向上を実現した．

2 シーケンスデータ

ストリームデータ $X = x_1, x_2, \ldots, x_i, \ldots$ は無限に続く d 次元ベクトルの集合である．スライディングウィンドウによって決定される X に含まれる w 個の連続したデータ系列をシーケンスデータという．w はウィンドウサイズであり，ウィンドウは 1 度に $step\_size$ だけシフトする．以降では，t 番目のシーケンスデータを「時刻 t に出現したシーケンスデータ」と呼び S_t と記載する．S_t は $x_{1+(t-1)step\_size}$ から $x_{w+(t-1)step\_size}$ までの w 個のデータで構成される例えば $w = 5, step\_size = 2$ の場合，S_i は図 1 のように作成される．

本研究では取り扱うクラスタリングでは，ストリーム X 内のシーケンスデータを k 個のクラスタに分類することを目標とする．

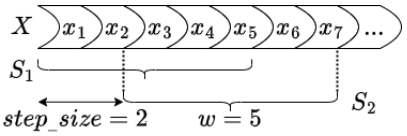

図 1: シーケンスデータ

ストリームデータに対するクラスタリング特有

*電気通信大学大学院　情報理工学研究科　情報・ネットワーク工学専攻

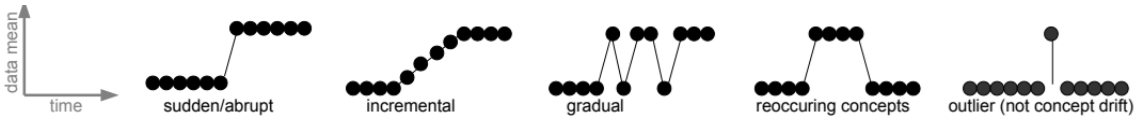

図 2: コンセプトドリフトの種類

の課題としてデータ量の増加とコンセプトドリフトの発生の 2 点が挙げられる．ストリームデータではデータが逐次到着して増加する環境で，リアルタイム処理が求められる．このため，アルゴリズム設計では効率的なデータ構造が求められる．加えて，オンライン環境下ではコンセプトドリフトが発生する．これは，時間の経過とともにデータの分布が変化していく現象を指す．図 2 はコンセプトドリフトの種類を指している [5]．図に示すようにドリフトにはいくつかの種類が存在し，その発生様式には突発的なものもあれば徐々に変化するものも含まれる．このようなドリフトと一時的な異常である外れ値やノイズを識別することが重要となる．

3 SECLEDS

本節以降では，シーケンスデータを単にデータと記述する．SECLEDS はオンラインの K-medoids クラスタリングアルゴリズムであり，常に k 個のクラスタ $C_1, C_2, \cdots, C_k$ を保持する．一方で，通常の K-medoids と異なり 1 クラスタが複数 p 個の代表データを持ち，1 クラスタが複雑なクラスタ形状を保てる．つまり，代表データの総数は $k \cdot p$ 個である．

初期代表データはストリームの先頭の $1.5kp$ 個のデータの中から kp 個の代表データを選択する．K-means++と同様なやり方で，お互いに離れた k 個の代表データを選び各クラスタの 1 つ目の代表データ $m_{1,1}, m_{2,1}, \cdots, m_{k,1}$ とする．ここで $m_{i,j}$ はクラスタ C_i の j 番目の代表データを表す．各クラスタ C_i の他の $p-1$ 個の代表データは，残っているデータの中で $m_{i,1}^1$ に最も近い $p-1$ 個のデータをインクリメンタルに選択する．

次に時刻 t にストリームにシーケンスデータ S_t が来たときの処理を説明する．SECLEDS は完全なオンラインアルゴリズムであるので，S_t を割り当てるクラスタをその場で決定する．ここでは，S_t を最近クラスタに割り当てるが，クラスタ C_i と S_t との距離 $D(S_t, C_i)$ は式 (1) のように p 個の代表データと S_t との平均距離で定義される．$d(.,.)$ は任意の距離関数を指定してよい．

$$D(S_t, C_i) = \frac{\sum_{j=1}^{p} d(S_t, m_{i,j})}{p} \tag{1}$$

クラスタ代表データの更新

S_t がクラスタ C_{cid} に割り当てられたとしよう．この時，SECLEDS はクラスタ C_{cid} の代表データを投票によって更新する．C_{cid} の代表データ $m_{cid,j}$ の得票数を $v_{cid,j}$ とする．S_t からの既存の代表データに投票した後，得票数最小の代表データを S_t に置き換える．

より具体的には，$v_{cid,j}$ の時刻 $t-1$ 直後の値を $v_{cid,j}^{t-1}$ とする．式 (2) のように時刻 t に X_t は最近代表データに 1 票投票し，それ以外の代表データは減衰係数 $\lambda < 1$ の値に基づき票数を減らされる．すなわち，

$$v_{cid,j}^{t} = \begin{cases} v_{cid,j}^{t-1} + 1 & \text{if } m_{cid,j} \text{ が最近代表データ} \\ \lambda v_{cid,j'}^{t-1}. & \text{最近代表データ以外．} j' \neq j \end{cases} \tag{2}$$

その後で，$1 \leq j \leq p$ に対して $j = \arg\min_j v_{cid,j}^t$ を満たす $m_{cid,j}^t$ を $C_{cid,j}^t$ から除外し,X_t と入れ替える．このとき，新代表データ X_t の初期得票数を 1 にセットする．以上のように，SECLEDS は新しいデータを必ずクラスタ代表に指定する．これは時間経過によってデータ分布が変化するコンセプトドリフトに対して，新代表データを準備することを目指している．

なお，X_t が割り当てられないクラスタ cid 以外のクラスタに関しては代表データの票数の増減を行わない．

4 提案手法

SECLEDS の欠点は，外れ値とコンセプトドリフトを区別する能力が欠如していることである．SECLEDS では各時刻 t で最新シーケンス S_t を必ず既存の代表データと入れ替えることで，時間経過に伴うデータ分布の変化，つまりコンセプトドリフトへの対応力を向上させている．しかし，S_t が本当にコンセプトドリフトなのか，それとも一時的な外れ値なのかを，S_t の到達直後に判断することは困難である．SECLEDS では S_t が外れ値である可能性を考慮していないため，外れ値への対応能力が十分ではない．そこで，本研究では代表データの票が少ない場合にのみ S_t を代表データとする MVT(Minimum Vote Threshold) 手法と，S_t を代表データとするか否かの判定を保留し，一時的にバッファに格納する TSND(Temporary Storage for New Data) 手法を

Algorithm 1 TSND Clustering Algorithm

1: **Input:** S_t
2: **Output:** $C^t_{cid}, m^t_{cid,1}, \ldots, m^t_{cid,p}$
3: $cid \leftarrow \arg\min_{1 \leq cid \leq k} \frac{\sum_{j=1}^{p} d(s, m^{t-1}_{cid,j})}{p}$
4: C_{cid} の中で S_t に最近のデータを a 個見つける．
5: $v^t_{cid,(i)} = v^{t-1}_{cid,(i)} + 1$ for all $1 \leq i \leq a$
6: $v^t_{cid,(i)} = \lambda^i v^{t+1}_{cid,(i)}$ for all $a < i \leq D$
7: Remove $\{c^t_{cid,i} | v^t_{cid,i} \leq min\_vote\}$ from C^t_{cid}
8: 得票数上位 p 個の要素を代表データ $m^t_{cid,1}, \ldots, m^t_{cid,p}$ と設定
9: Assign S_t to C^t_{cid}

提案する．

4.1 MVT

SECLEDS では S_t を得票数最小の代表データを必ず入れ替える．これでは，S_t がコンセプトドリフトでなく単なる外れ値だった場合に，外れ値により既存の代表データを追い出すことになるのでクラスタリング結果が不安定になる．

提案手法では MVT は得票数最小の代表データの得票数が閾値 min_vote 以上である場合に代表データの入れ替えを行わない．つまり，安定した代表データが外れ値によって追い出されないようにする．従って，コンセプトドリフトが起きない状況で SECLEDS よりも優れている．min_vote はユーザが指定するパラメータであり，$0 <$ min_vote < 1 を満たす値である．

4.2 TSND

TSND は MVT の外れ値への対応能力を保持したまま，コンセプトドリフトが発生している状況に対してもロバストである手法である．これまでのデータ分布に従わない S_t が到着した場合，新たなコンセプトドリフトの兆候を示しているか単なる外れ値であるかどうかは，S_t が到達してからしばらく時間が経過しなければ分からない．

SECLEDS や MVT は到達要素に対してその要素が到達した時間にのみ代表データに加える判定を行うため，正確な判定を行う事は困難だった．そこで本手法では到達要素を代表データ以外として一時的に格納する．コンセプトドリフトか外れ値かどうか判断できるまでクラスタ内に代表データ以外として保持しておくことで判別の精度をより向上させることを狙う．各クラスタ C_i は代表データと代表データ以外を持つことになるが，それらの和集合を C_i のメンバと呼ぶ．代表データ以外のデータ数は増えすぎないよう工夫している．また本手法では票の減衰モデルも更新する．従来のモデルでは票の減少が一定の割合で行われるため，急激なデータの変化に対応しきれない場合があった．本手法では時間の経過に応じてより票の減衰率を大きくし，古いデータの影響を素早く減少させ新しいデータを迅速に反映することが可能となる．これにより，さらなる精度の向上が期待される．

到達データ S_t がクラスタ C_{cid} に割り当てられたとする．時刻 $t-1$ における C_{cid} のメンバ数を D とし，メンバとその得票数を $C^{t-1}_{cid} = \{(c^{t-1}_{cid,1}, v^{t-1}_{cid,1}), \ldots (c^{t-1}_{cid,D}, v^{t-1}_{cid,D})\}$ であるとする．

C^{t-1}_{cid} の中で S_t と距離が最近の上位 a 個の要素を見つける．これらの要素に対して 1 票投票し，それ以外の要素は λ の値に基づき票数を減らされる．票を減らす際は，SECLEDS や MVT とは異なる時間の経過に伴い票の減少率が大きくなるモデルを用いる．

要素 $c^t_{cid,j}$ がクラスタに到達した時刻を $t'(t > t')$ としたとき，時刻 t での票は次のように表される．

$$v^t_{cid,j} = \begin{cases} v^{t-1}_{cid,j} + 1 & \text{投票数上位 } a \text{ 個の要素} \\ \lambda^{t-t'} v^{t-1}_{cid,j'} & \text{それ以外の要素 } j' \neq j \end{cases}$$

票の更新によって $v^t_{cid,i} < min\_vote$ となった要素は C^t_{cid} に加えず廃棄する．その後，得票数が上位 p 個の要素を新たに代表データ $m^t_{cid,1}, \ldots, m^t_{cid,p}$ とし，S_t を初期得票数 1 にセットした上で C^t_{cid} に追加する．

TSND では得票数が min_vote 以下になったデータを廃棄することで，1 クラスタが管理するメンバ数が増えないようにしている．また，S_t は到着後にさらに票を獲得しないと代表データにならないので，一時的な外れ値は代表データにしない．

5 実験設定

実験を行う環境として，Intel(R)Core(TM)i7-6700CPU@3.40GHz,16GB メモリ，Windows10 の計算機を用意した．

5.1 使用データセット

実験では KDD Cup 1999 Data[4] を用いる．このデータセットは侵入検知システムの評価プロジェクトで収集されたネットワークトラフィックデータを基にしたデータセットである．次元数 $d = 41$ であり，総レコード数 $n = 4898431$ のうち, 攻撃通信が 3925650 レコード，通常通信が 972781 レコード存在する．これを，$w = 100, step\_size = 50$ とするシーケンスデータに変換し，クラス数 $k = 2$ のクラスタに分類することを目的とする．

5.2 評価指標

提案手法の評価には，ラベルベースの指標，距離ベースの指標，及び実行時間の 3 種類の指標を用い

る．ラベルベースの指標では Recall(R)，Precision (P)，F1 スコア (F) を採用する．クラスタリングにおける混同行列は，同一のラベルのデータを同じクラスタに割り当てる真陽性 (TP)，異なるラベルのデータを異なるクラスタに割り当てる真陰性 (TN)，異なるラベルのデータを同じクラスタに割り当てる偽陽性 (FP)，同一のラベルのデータを異なるクラスタに割り当てる偽陰性 (FN) と定義される．これらに対して各評価指標はそれぞれ次のように表される [6].

$$P = \frac{TP}{TP+FP}\ R = \frac{TP}{TP+FN}\ F = \frac{2PR}{P+R}$$

次に，距離ベースの指標としてシルエット係数 [7] を拡張した指標を用いる．この指標は代表データ間の距離を計算することで代表データの凝集度を測定し，ラベル指標とは異なる観点で適切に分離されたクラスタを取り出せているかを評価するために用いる．

あるクラスタ C の代表データ集合 M と，C に最も近い代表データ集合 M_{near} が存在する．このとき，任意の代表データ $m \in M$ に対してクラスタ内代表データ距離 $a(m)$，クラスタ外代表データ距離 $b(m)$，代表データに対するシルエット係数 $MSC(m)$ をそれぞれ次のように定義する．

$$a(m) = \frac{1}{|M|-1} \sum_{s \in M \setminus m)} d(m,s)$$

$$b(m) = \frac{1}{|M_{near}|} \sum_{s \in M_{near}} d(m,s)$$

$$MSC(m) = \frac{b(m)-a(m)}{\max(a(m),b(m))}$$

これを毎時刻，全クラスタの全ての代表データに対して計算し，その平均値を指標として扱う．

5.3 実験結果

表 1 に実験の結果を示す．MVT，TSND はどちらも SECLEDS よりもシルエット係数が高く，SECLEDS よりもクラスタの分離性が優れている．一方で，$a(m)$ が小さいことから SELCEDS よりクラスタの代表データの凝集度が高いことが分かる．一般には同一クラスタの代表点が凝集するのは望ましいことである．しかしコンセプトドリフトが発生する状況では，データ分布の変化に追従するため，代表データが分散している方が優れていることもありえる．しかし，SECLEDS は MVT，TSND の両方と比較して F1 スコアが大きく劣っていることから，外れ値を代表データに採用したために $a(m)$ が大きくなった可能性が高い．TSND は MVT より $a(m)$ が大きいにも関わらず，F1 スコアも増加した．これは TSND が外れ値を代表データとせず，コンセプトドリフト発生時には代表データを更新できていることが原因と考えられる．実行時間は，MVT は SECLEDS より短くなった．これは，MVT 手法では min_vote を下回った場合代表点の更新処理を行わないのが理由として考えられる．

表 1: 各アルゴリズムの実験結果

	SECLEDS	**MVT**	**TSND**
$a(m)$	5.25	4.90	5.10
$b(m)$	27.1	29.6	27.5
MSC	0.786	0.819	0.809
Recall	0.627	0.807	0.853
Precision	0.696	0.740	0.770
F1 Score	0.660	0.771	0.809
Runtime(sec)	509	489	859

6 結論

本研究では SECLEDS の投票手法を拡張し，コンセプトドリフトへの対応力を伸ばすことで精度の向上を図った 2 種類のアルゴリズムを提案した．ドリフトを含むデータセットに対する実験の結果，既存手法に対する精度の向上が確認できた．今後の展望は, 異なるデータセットに対する実験，代表データ同士の距離がクラスタの精度に与える影響の分析，そしてクラスタごとに代表データの個数の上限値を定義しない手法を考えること等が挙げられる．

謝辞

本研究は JSPS 科研費 JP21K11901 の助成を受けたものである．

参考文献

[1] Marcel R Ackermann, Marcus Märtens, Christoph Raupach, Kamil Swierkot, Christiane Lammersen, and Christian Sohler. Streamkm++ a clustering algorithm for data streams. *Journal of Experimental Algorithmics (JEA)*, Vol. 17, pp. 2–1, 2012.

[2] Erich Schubert and Peter J Rousseeuw. Faster k-medoids clustering: improving the pam, clara, and clarans algorithms. In *Similarity Search and Applications: 12th International Conference, SISAP 2019, Newark, NJ, USA, October 2–4, 2019, Proceedings 12*, pp. 171–187, 2019.

[3] Azqa Nadeem and Sicco Verwer. Secleds: sequence clustering in evolving data streams via multiple medoids and medoid voting. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 157–173, 2022.

[4] S. Hettich and S. D. Bay. The uci kdd archive. Irvine, CA: University of California, Department of Information and Computer Science. `http://kdd.ics.uci.edu`, 1999.

[5] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, Vol. 46, No. 4, pp. 1–37, 2014.

[6] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Flat clustering*, pp. 321–345. Cambridge University Press, 2008.

[7] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, Vol. 20, pp. 53–65, 1987.