

データドリフト対処のための Adversarial Validation を用いたデータ選択手法の評価

今野 由麻[†] 中野美由紀^{††} 小口 正人[†]

[†] お茶の水女子大学人間文化創成科学研究科 〒112-8610 東京都文京区大塚 2-1-1

^{††} 津田塾大学学芸学部 〒187-8577 東京都小平市津田町 2-1-1

E-mail: [†]{konno.yuma,oguchi}@is.ocha.ac.jp, ^{††}miyuki@tsuda.ac.jp

あらまし Adversarial Validation は、機械学習において学習データとテストデータの違いを検出する手法であり、より性能の良いモデルを学習するために用いられる。先行研究として、Adversarial Validation をデータドリフト対処のために利用し、教師あり機械学習モデルを更新して時系列データの予測をバッチ単位で行う枠組みが提案されている。その特徴量選択に注目した先行研究の枠組みを応用して、我々はこれまでにデータ選択を行う枠組みである、データ選択手法を提案してきた。本研究では、ドリフトが起こる人工データセットを用いて、データ選択手法の評価を行う。

キーワード 汎用機械学習技術, ストリームデータ処理, データドリフト, Adversarial Validation

1 はじめに

機械学習モデルの運用上の課題として、model decay という予測性能が低下する現象が知られており、その原因として学習データと予測を行うデータに差が生じるデータドリフトが存在する。そのようなドリフト環境下を想定して、我々は適切な学習データを過去のデータプールから選択してモデルの更新を行う枠組みであるデータ選択手法を提案してきた[1], [2]。人工ドリフトデータセットとして用いられてきた実績のある CIRCLES データセットを実験に利用し、直前のデータをもちいてモデルの更新を行う場合と比較すると、データ選択手法を用いたほうが性能のよいモデルを得られることを確認する。

2 先行研究

ここでは、Adversarial Validation を利用した2つの研究[3], [4]を紹介する。Adversarial Validation とは、学習データとテストデータを見分ける二値分類を行う敵対的分類器を用いて、二つの集団の分布が異なっていることを検出する手法である。この手法は、機械学習コンテストでよく用いられる手法であるが、論文としての発表は数少ない。論文において Adversarial Validation を用いた先行研究を二件示す。一つ目は、J. Pan ら[3]による Adversarial Validation を用いて特徴量選択を行う方法を含む3通りのコンセプトドリフト適応手法に適用し、比較・検討した、自動でコンセプトドリフトに適応可能なユーザーゲーティングシステムの事例である。我々が提案するデータ選択手法は、この研究の内容を拡張したものとなっている。二つ目は、Adversarial Validation を用いてバリデーションデータ選択を行った事例[4]である。この例では、Adversarial Validation をバリデーションデータ選択に利用することで、E コマースの購入意図予測タスクにおけるデータの不均衡さに対処している。

3 人工ドリフトデータセット CIRCLES

CIRCLES データセットは M.Kubat ら[5]によって提案されたものであり、徐々にドリフトが進行するタイプの人工ドリフトデータセットである。[6]で紹介されているように、これまでいくつかのドリフトに関係する研究で使用されてきたものである。本研究では A. Pesaranghader によって公開されている実装[7]をほとんどそのまま利用したが、クラスを n と p (negative と positive) の二値からそれぞれ 0 と 1 の二値に置き換えている。

3.1 CIRCLES データセットの生成方法

このデータセットの特徴量は 0 から 1 までの範囲の二次元空間上の座標である。また、クラスはその座標 (特徴量) がある円の外側にある場合を 1、内側にある場合を 0 とするような値である。0 と 1 の二つのクラスのデータは約半数ずつ生成される。データセット生成時に利用する円を切り替えることで、既存のコンセプトから新しいコンセプトへの移行を再現している。各データセットにつき、バッチの数と同数の円を使ってデータセットを生成する。終端バッチはドリフトの発生がなく、他と比べて特殊なバッチになっている。

3.2 CIRCLES データセットで設定可能な要素について

CIRCLES データセットを生成する際にいくつかの設定を変更して本実験で利用するデータセットを作成した。以下に CIRCLES データセット生成の際に設定した項目を示す。

- Batch Size: 各バッチに含まれるデータの数
- Transition Length(tl): 既存のコンセプトが次のコンセプトに移り変わり始めてから終わるまでの移行期間の長さ
- ドリフトパターン: データセット生成に利用する4つの円(円1から円4)であり、円の半径 r と中心座標 (x,y) の組み合わせの配列

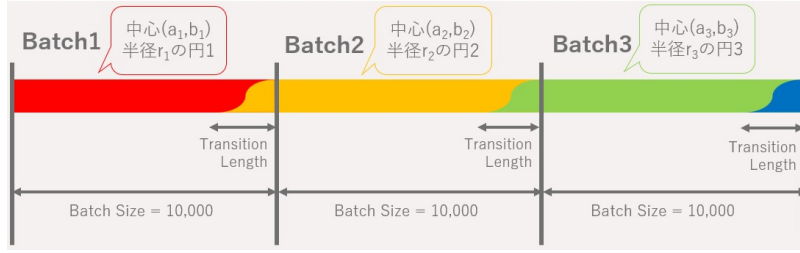


図 1: 短い Transition Length で生成した CIRCLES データセットの様子



図 2: 長い Transition Length で生成した CIRCLES データセットの様子

図 1 には Transition Length を短く設定した場合の CIRCLES データセットのイメージを、図 2 には長く設定した場合の様子を示した。この二つの図では、バーの色とデータ生成時に利用される円が一对一に対応している。各バッチの終端から Transition Length 分のバーの色の切り替わりは、シグモイド関数を使って次の円で生成されるデータの量が増えていく様子を表現しており、この部分がコンセプトの移行期間に該当する。バッチ 1 を例にすると、初めは赤いバーに対応する円 1 を利用してデータを生成し、徐々に黄色いバーに対応する円 2 を利用するように移行する。

3.3 実験で利用した 4 つのバッチで構成された 7 つの CIRCLES データセットの設定

本実験では、ドリフトパターンや Transition Length を変え、7 つのデータセットを生成して実験を行った。またその際、すべての場合において Batch Size は 10,000 とした。用意したドリフトパターンは 5 種類であり、それぞれ original/heavily-overlapping(h-overlapping)/overlapping/adjacent/separate と呼称することにする。それぞれのドリフトパターンの様子を図 3 に示す。

5 つのドリフトパターンのうちの 1 つである original は CIRCLES データセット提案者 [5] が利用していたものである。ドリフトパターン original を利用して生成したデータセットは、主に Transition Length を変化させた場合の結果を確認するために用いる。それ以外の h-overlapping/overlapping/adjacent/separate の 4 つは、ドリフトパターンに含まれるすべての円のサイズが均一 (半径 r が 0.1) であり、円の中心間の距離が一定間隔になるよう設計したデータセットである。半径 r に対してそれぞれのドリフトパターンでは、中心間の距離が r , $1.5r$, $2r$ (外接円), $2.5r$ となっ

ている。これらのドリフトパターンを利用して生成したデータセットは、Transition Length を均一に設定し、ドリフトが起これるデータの変化度合いが異なる場合の結果を比較するために用いる。表 1 は、実験に利用する 7 つのデータセットを生成するために利用したドリフトパターンと Transition Length を組み合わせの表である。以下では original.tl2000 のように、データセット生成に利用したドリフトパターンと Transition Length を「_」で繋げて、各データセットを呼称する。

4 ドリフトデータを見分けるデータ選択指標の検討

本章では、3 章で紹介した CIRCLES データセットを用いて、データ選択手法の根幹を担うデータ選択の指標について検討する。選択指標には、学習データの特徴量にクラス 0、予測を行いたいデータの特徴量にクラス 1 を新しく割り当てて、その二つを見分けるように学習した敵対的分類器から得られるクラス予測確率を用いる。ここでは敵対的分類器のクラス予測確率を利用した二つのデータ選択指標を用意し、生成した 4 つのバッチを持つ CIRCLES データセットのバッチ 2 と 3 を用いて、それぞれの指標に基づくデータ選択の効果を比較するための実験を行った。以下に実験の流れを示す。

表 1: 生成した 7 つのデータセットのドリフトパターンと Transition Length の組み合わせ

	tl2000	tl6000	tl10000
original	✓	✓	✓
h-overlapping			✓
overlapping			✓
adjacent			✓
separate			✓

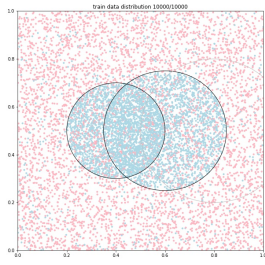
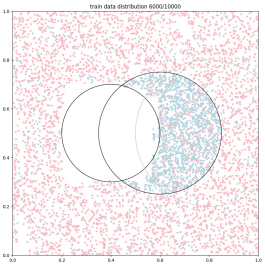
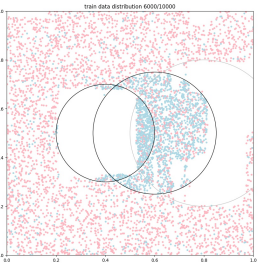
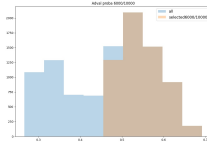
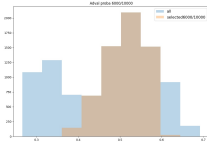
表 2: データ選択指標 1,2 に関するデータ選択時の性能比較 (Transition Length が異なる場合)

	Pattern	Metrics	2,000 件	4,000 件	6,000 件	8,000 件	10,000 件
指標 1	original_tl2000	AUC	81.87	84.65	87.49	84.77	82.98
		Accuracy	51.88	52.91	63.46	65.61	64.03
	original_tl6000	AUC	73.80	81.77	82.68	83.06	80.21
		Accuracy	63.01	64.37	67.33	78.70	77.80
	original_tl10000	AUC	71.43	78.43	79.25	78.31	77.20
		Accuracy	69.76	70.00	74.50	77.14	74.89
指標 2	original_tl2000	AUC	72.70	77.41	81.68	83.88	82.98
		Accuracy	53.82	57.01	64.10	64.55	64.03
	original_tl6000	AUC	79.46	80.38	83.27	83.01	80.21
		Accuracy	55.75	58.54	67.14	78.38	77.80
	original_tl10000	AUC	81.64	84.43	85.58	80.10	77.20
		Accuracy	58.43	69.54	78.83	77.27	74.89

表 3: データ選択指標 1,2 に関するデータ選択時の性能比較 (円のサイズが均一なドリフトパターンの場合)

	Pattern	Metrics	2,000 件	4,000 件	6,000 件	8,000 件	10,000 件
指標 1	h-overlapping_tl10000	AUC	75.35	76.01	78.73	84.46	85.41
		Accuracy	67.02	63.44	65.27	75.10	78.76
	overlapping_tl10000	AUC	69.55	72.16	72.72	79.53	78.89
		Accuracy	57.57	59.75	62.24	68.58	71.51
	adjacent_tl10000	AUC	67.40	69.51	69.21	72.92	75.79
		Accuracy	55.27	58.29	66.35	66.33	66.95
指標 2	separate_tl10000	AUC	65.03	66.38	65.79	78.93	74.28
		Accuracy	53.26	60.57	65.16	65.43	67.03
	h-overlapping_tl10000	AUC	75.26	82.57	86.64	87.40	85.41
		Accuracy	51.72	53.03	60.22	74.94	78.76
	overlapping_tl10000	AUC	77.40	72.09	71.16	84.86	78.89
		Accuracy	60.83	56.92	57.64	71.96	71.51
	adjacent_tl10000	AUC	81.63	82.89	81.62	75.35	75.79
		Accuracy	67.17	70.46	70.35	68.09	66.95
	separate_tl10000	AUC	85.00	85.68	80.46	79.74	74.28
		Accuracy	70.87	68.49	66.41	63.15	67.03

表 4: データ選択指標ごとの予測性能 (AUC, Accuracy) と選択されたデータの分布 (original_tl10000, 6,000 件選択の場合)

	データ選択なし	指標 1	指標 2
AUC	77.20	79.25	85.58
Accuracy	74.89	74.50	78.83
データ分布			
クラス 1 への 予測確率			

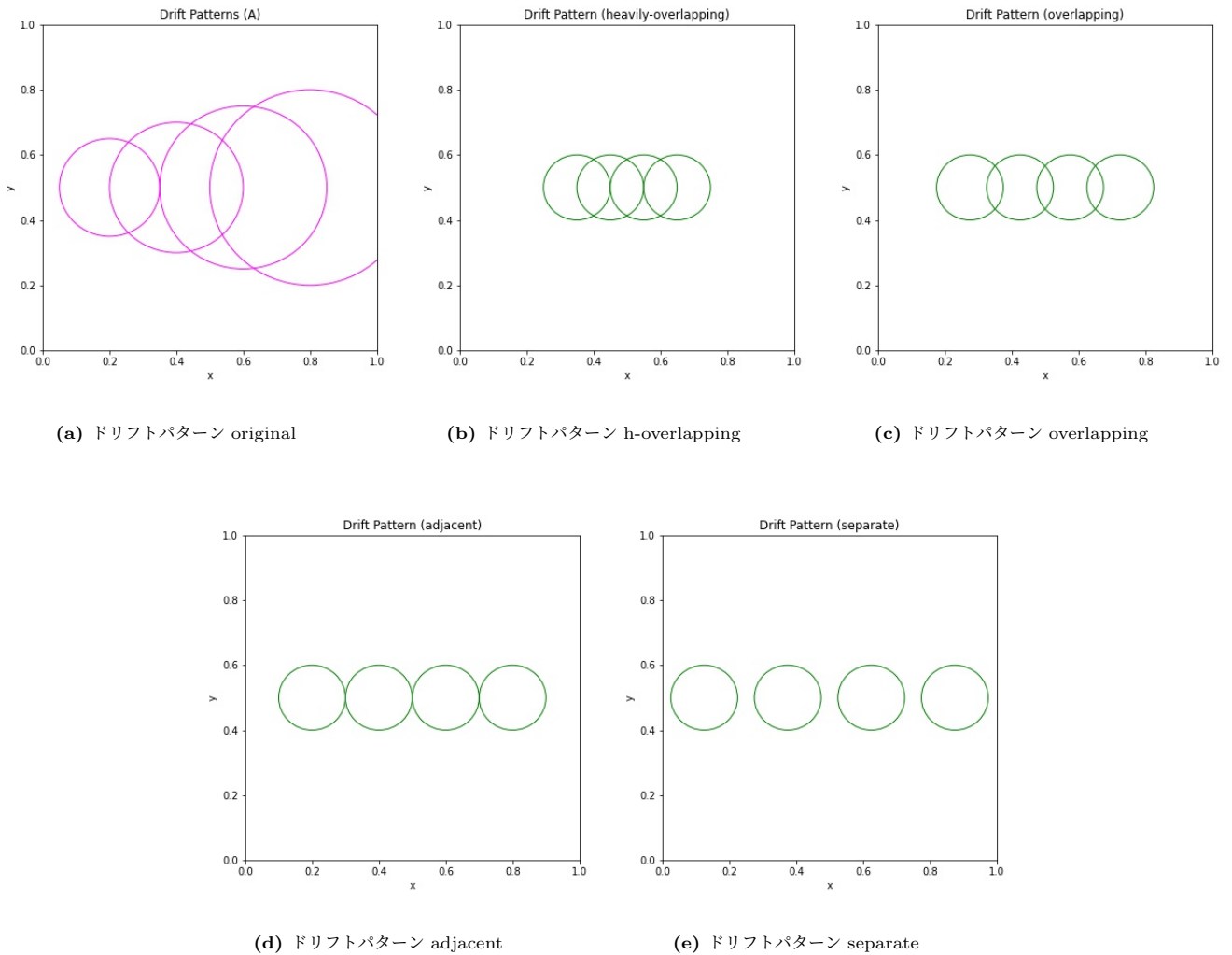


図 3: ドリフトパターン

(1) 学習に利用可能なデータとして、バッチ 2 の特徴量に敵対的分類器のための新しいクラス 0 を付ける。また予測を行いたいデータとして、バッチ 3 にクラス 1 を付ける。

(2) (1) で付与した新しいクラスを見分けるような敵対的分類器 (GBDT : Gradient Boosting Decision Tree) を学習する。

(3) 学習した敵対的分類器を用いて、バッチ 2 のデータのバッチ 3 への予測確率を得る。

(4) クラス予測確率を利用したデータ選択指標に基づいて、バッチ 2 のデータから予測に適したデータを選択する。

(5) 選択されたデータを用いて、バッチ 3 のデータのクラスを予測するモデルを学習し、その予測性能や選択されたデータの分布を確認する。

クラス予測確率を用いる二つの指標として、以下を用意した。

- 指標 1: 学習に利用可能なデータ (クラス 0) のうち、予測を行いたいデータ (クラス 1) への予測確率が高いものを学習に利用する
- 指標 2: 学習に利用可能なデータ (クラス 0) のうち、予測を行いたいデータ (クラス 1) への予測確率が 50% 付近のものを学習に利用する

表 2 には Transition Length が異なる 3 つのデータセットの

結果を、表 3 にはドリフトパターンが異なる (中心座標が異なる均一な大きさの円のドリフトパターン) 4 つのデータセットの結果を示した。それぞれの表には、データ選択指標 1 と 2 を用いてバッチ 2 から選択された 2,000 件から 10,000 件 (全件) を学習に利用してバッチ 3 のクラス予測を行った場合の AUC (Area Under the Curve) と Accuracy を示している。この結果はシードを変えて 30 回実験を行った結果の平均であり、各行の最良の結果が太字でハイライトされている。データ選択を行わなかったとき (10,000 件のデータを持ちたとき) と比較して、データ選択を行うことで性能の改善が見られるかどうかを確認する。

表 2 の結果では、データ選択を行わなかった場合 (10,000 件) よりもすべての場合において、データ選択を行ったほうが性能が良くなる結果となった。AUC に注目すると、original_tl10000 で指標 2 を利用した場合に最も性能を大きく改善できている。Transition Length が小さい場合では、指標 2 よりも指標 1 がよりよい結果を出しているが、基本的には指標 2 のほうがよりよい指標であると考えられる。今回実験で用いたバッチ 2 の生成には円 2 (ドリフトパターンの 4 つの円のうち左から 2 つ目の円) と円 3 が利用されており、バッチ 3 の生成には円 3 と円 4 が利用されている。ここでのデータ選択では、バッチ 2 の

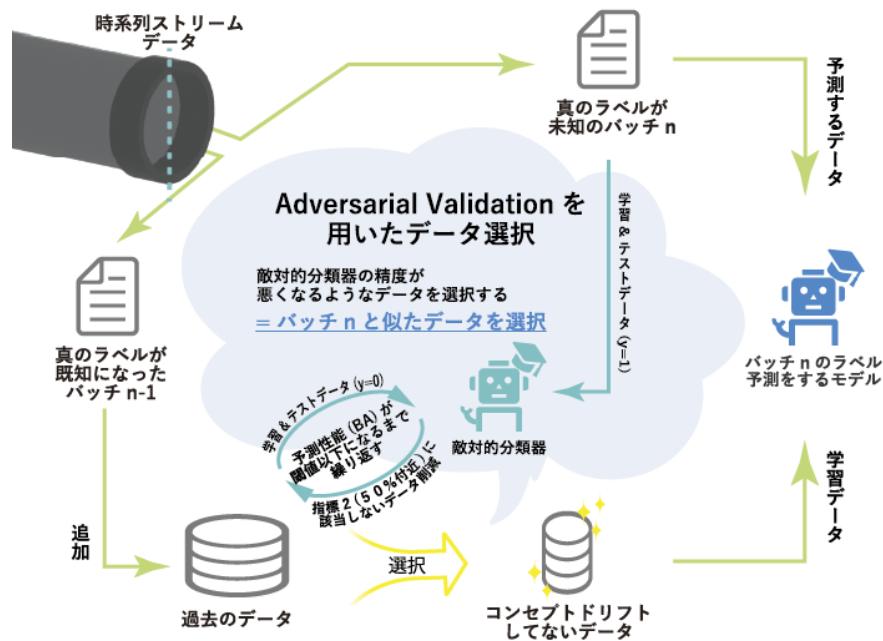


図 4: データ選択手法概要

データのうち、バッチ 2 と 3 に共通して用いられる円 3 で生成されるデータを選択することが性能改善につながると考えられる。しかし、指標 1 が有利になった Transition Length が短い場合には、バッチ 2 のうち性能改善につながる円 3 で生成されたデータの量が少ないため、最大限よいデータ選択を行っても Transition Length が大きいときほどの効果が見込めない。よって、Transition Length が短い場合には指標 1 に劣る可能性はあるものの、基本的には指標 2 のほうがよりよい指標であると考えられる。Accuracy では、指標 1 を利用した場合と指標 2 を利用した場合それぞれの最良の結果を比較すると、1% 程度の差しかないことがわかる。

表 3 の結果では、指標 2 を用いてデータ選択を行なった場合、ほとんどのケースでデータ選択を行わなかった場合よりも性能を改善できていることが確認できた。性能の改善幅をそれぞれのデータセットごとに確認してみると、ドリフトによるデータの変化量が大きいデータセットの方がよりデータ選択による性能改善の幅が大きいことがわかる。また、データセットによって、2,000 件から 10,000 件までの範囲の選択件数のうち、最適な結果となる件数に違いがあることが確認できた。これは、それぞれのドリフトパターンにおいて円と円が重複する部分の面積が異なることで、学習に有用なデータ数に違いがあることが関係していると考えられる。

続いて、データ選択を行なって選ばれたデータの分布を確認する。表 4 には、Transition Length を変えた実験 (表 2) のうち original_tl10000 のデータを利用した場合において、最も性能の良かった 6,000 選択した場合に選択されたデータの分布を示している。散布図とヒストグラムには、バッチ 2 のデータのうち実際に学習に利用したデータの分布を示した。CIRCLES データセットの特徴量は 0 から 1 までの範囲の二次元空間上の

座標であるため、散布図にそのままプロットすることができる。この散布図では、正のクラスをピンク色、負のクラスを水色のドットで表現している。ヒストグラムには、バッチ 2 のデータを水色で示し、そのうちの学習に利用した部分をオレンジ色で示した。ヒストグラムの横軸は敵対的分類器のクラス 1 (予測を行いたいバッチ 3) への予測確率である。

この結果では、散布図の様子からデータ選択指標 1 とデータ選択指標 2 の間では、異なるデータが選択されていることがわかる。指標 2 を用いて選択されたデータの分布では、右端の部分に空白ができています。この部分はちょうど予測を行うバッチでドリフトの影響を受ける範囲であり、この部分を学習データから省いたことが性能改善につながったと考えられる。

5 データ選択手法

5.1 データ選択手法概要

データ選択手法は J.Pan らによる先行研究 [3] を拡張するものである。[3] と同様に、ストリームデータがバッチ単位で処理され、新バッチ処理 (バッチ n) が行われるタイミングで直前バッチ (バッチ n-1) のラベルが判明する。また、性能を評価する指標として AUC スコアを用いる。

図 4 にデータ選択手法の概要を示す。まず、新バッチ (バッチ n) の予測を行う時点でクラスが判明している過去のデータのうち、データ選択の対象にするデータの特徴量に対して、敵対的分類器の学習のために新しくクラス 0 を割り当てる。予測を行いたい新バッチ (バッチ n) に対しても同様にクラス 1 を割り当てる。そしてその新しく割り当てたクラス 0 と 1 を見分けるような二値分類器 (GBDT) を敵対的分類器として学習する。次に、学習した敵対的分類器の予測性能 (BA: Balanced Accuracy) が与えられた閾値より小さくなるまで、ドリフトの

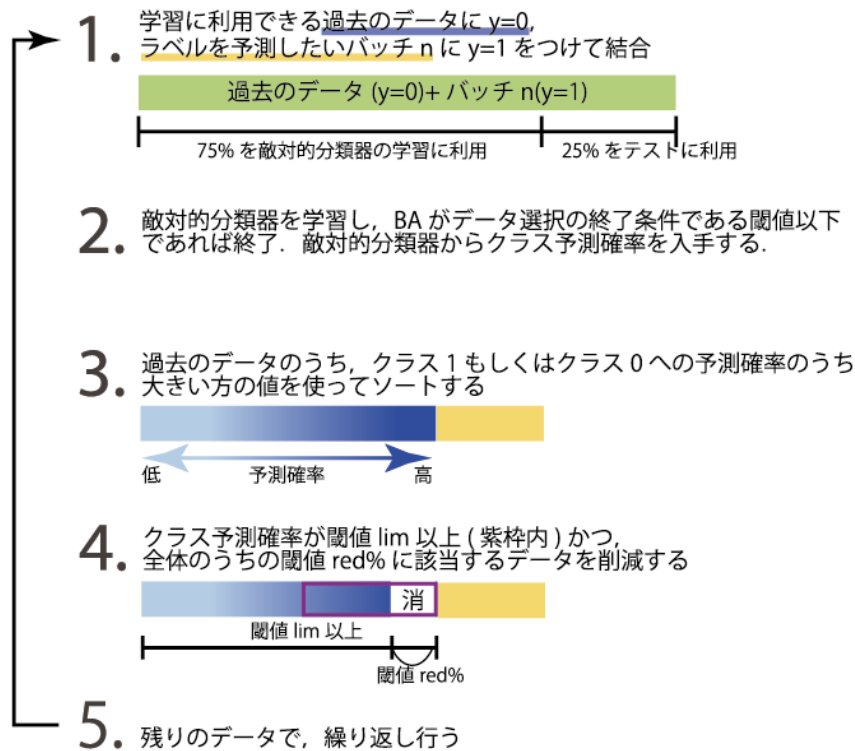


図 5: データ選択の流れ

影響を受けたデータの少量削減と敵対的分類器の再学習 (データ選択のループ) を繰り返し行う. 最終的に残ったデータを学習データとしてモデルの更新を行い, 予測を行いたいデータのラベル予測に用いる.

どのようなデータをドリフトの影響を受けたデータとみなすのかが本手法の一つの要になる. ここでは, 4 章の実験で効果を確認したデータ選択指標 2 (学習に利用可能なデータのうち, 敵対的分類器のクラス予測確率が 50% 付近のデータをドリフトの影響が少ないデータとみなす) を用いる. また, ここでは最終的な性能評価の指標として AUC を設定しているが, データ選択のループ終了条件である閾値には BA を使用している. その理由は, ループを繰り返すごとにクラス不均衡になっていく敵対的分類器の学習において, 不均衡データの予測で高い値を示すことがある AUC を用いることでループ終了条件がうまく機能せず, 過度にデータ選択が行われることを防ぐためである. この終了条件である閾値よりも, 敵対的分類器の性能が低くなるような学習データを選ぶことで, 敵対的分類器にとってクラス 1 のデータ (予測を行いたいデータ) と見分けがつかないようなクラス 0 のデータ (学習データ) を選択することができる.

次に生成するモデルは再利用されないため, 汎化性能を必要としない. また, 本手法を利用する状況では, ドリフトの影響により学習データが母集団全体の代表的な値であるという仮説が成立しなくなる. そのため本手法を用いてドリフト後のデータを説明可能な学習用データを選択することで, データ選択を行わない場合よりも適切な学習を行える可能性がある.

5.2 データ選択ループ内の流れについて

図 5 では, データ選択のループで削減するデータの選び方について, さらに詳しい流れを説明している. 以下にここで用いた 3 つの閾値を紹介する.

- データ選択のループ終了条件: BA であり, この値よりも敵対的分類器の予測性能 (BA) が低下したときに, データ選択のループを終了する. また, 削減できるデータがなくなった場合にもループは終了される.
- 閾値 $lim(limit)$: 過剰なデータ削減を予防する働きをする値
- 閾値 $red(reduce)$: 一度のループで削減するデータの割合を決める値

図 5 の 1. では, 新しくラベル 0 を割り当てた学習に利用できる過去のデータとラベル 1 を割り当てた予測したいデータ (バッチ n) を, シードを利用して敵対的分類器の学習用データとテスト用データをランダムに分割する. 2. では, 敵対的分類器を学習し, そのモデルから得たクラス予測確率がデータ選択のループ終了条件を下回っていないかを確認する. 3. では, ラベル 0 を割り当てた過去のデータをクラス 1 への予測確率が 50% 付近である (データ選択指標 2) もので並び替える. 4. では, 閾値 lim によって与えられる削減可能範囲 (紫枠内), かつデータ選択指標 2 に基づいてドリフトを起こしている可能性の高い上位 $red\%$ の過去のデータを削除している. このとき削減可能なデータがない場合も, データ選択のループを終了する. この 1 から 4 の流れを繰り返し行う.

表 5: 閾値 \lim をなしとし、データ選択の終了条件を変化させた場合のバッチ 2 でバッチ 3 の予測を行うデータ選択手法の結果

		ループ終了条件 (BA)			全件
		50%	60%	70%	
original_tl10000	AUC	81.36	77.25	77.20	77.20
	選択件数 (/10,000)	7026.17	9975.00	10000	10000
	ループ数	10.43	1.067	1.000	-
heavily-overlapping_tl10000	AUC	87.74	88.01	85.41	85.41
	選択件数 (/10,000)	6858.40	8533.93	10000	10000
	ループ数	11.00	5.167	1.000	-
overlapping_tl10000	AUC	86.97	85.92	78.89	78.89
	選択件数 (/10,000)	7032.47	8783.73	10000	10000
	ループ数	10.40	4.400	1.000	-
adjacent_tl10000	AUC	85.90	80.88	75.79	75.79
	選択件数 (/10,000)	7236.67	8818.53	10000	10000
	ループ数	9.633	4.300	1.000	-
separate_tl10000	AUC	84.12	79.55	74.28	74.28
	選択件数 (/10,000)	7218.50	8943.53	10000	10000
	ループ数	9.633	3.933	1.000	-

表 6: データ選択のループ終了条件を 50%とし、閾値 \lim を変化させた場合のバッチ 2 でバッチ 3 の予測を行うデータ選択手法の結果 (バッチ数 4 のデータセット)

		50 ± 15%	50 ± 10%	50 ± 5%	\lim なし
original_tl10000	AUC	78.30	79.62	81.36	81.36
	選択件数 (/10,000)	9537.57	8454.27	7042.57	7026.17
	ループ数	3.633	6.667	10.40	10.43
heavily-overlapping_tl10000	AUC	87.37	87.54	87.74	87.74
	選択件数 (/10,000)	8322.67	7392.37	6858.40	6858.40
	ループ数	9.533	10.10	11.00	11.00
overlapping_tl10000	AUC	85.37	86.09	86.11	86.97
	選択件数 (/10,000)	8802.30	8245.47	7197.63	7032.47
	ループ数	5.633	7.800	10.10	10.40
adjacent_tl10000	AUC	75.88	77.48	83.91	85.90
	選択件数 (/10,000)	9551.63	9209.40	7997.30	7236.67
	ループ数	2.467	2.000	7.700	9.633
separate_tl10000	AUC	77.85	79.49	82.60	84.12
	選択件数 (/10,000)	9346.70	8946.90	7547.93	7218.50
	ループ数	3.267	5.067	8.967	9.633

6 データ選択手法を用いた実験

4章で行った実験の結果と比較を行うため、バッチ2でバッチ3の予測を行う場合の実験を2つ行った。2つの実験に用いたデータセットは、用意した7つのデータセットのうち、Transition Lengthが10,000の5つである。一度のループにおけるデータ削減量を定める閾値redは、細やかな選択を行うために5%と設定した。また実験結果では、いずれの結果もシードを変えて30回実験を行った平均値を結果とした。

6.1 閾値limを固定した実験

一つ目の実験では、閾値redを除いた残り二つのデータ選択手法の閾値のうち、閾値limは利用しない条件でデータ選択手法を行った。データ選択のループ終了条件は50%, 60%, 70%の3つを試した。

表5にこの実験の結果を示した。データ選択手法のループ終了条件である閾値を50%, 60%, 70%の三つすべてを利用した結果と、比較のためにデータ選択を行わなかった(全件を学習に利用した)ときの結果を掲載しており、それぞれの場合で最終的に選択されたデータを使ってバッチ3の予測を行った性能(AUC)と選択された件数とデータ選択ループの回数を示した。

この結果から、すべての場合において、データ選択を行って性能を改善できたことが確認できた。4章の実験結果と比較して、AUCはoriginal_tl10000では約4%ほど下回る結果となったが、それ以外の4つの場合では同等かわずかに上回る結果となった。一方選択件数に注目して比較すると、4章の表2や表3の結果において、最も良い結果が出たときの学習データ数は、h-overlapping_tl10000とoverlapping_tl10000で8,000件、adjacent_tl10000とseparate_tl10000で4,000件であったことに対して、この実験では7,000件程度のデータで学習を行っていることがわかる。十分な性能は得られているものの、閾値を下回るまで学習データの少量削減を繰り返すデータ選択手法では、最小限以上のデータが選択されている可能性がある。

6.2 データ選択のループ終了条件を固定した実験

二つ目の実験では、閾値redを除いた残り二つのデータ選択手法の閾値のうち、データ選択のループ終了条件の閾値を直前の実験6.1で最もよい結果を示した50%に固定してデータ選択手法を行った。データ選択指標に基づく一定範囲のデータを削減範囲から退避するエリアを決めることができる閾値limは、50%±15, 50%±10, 50%±5, 50%±0(閾値limなし)の4つを試した。

表6にこの実験の結果を示した。ここでは閾値limの値を変えて、表5と同様にデータ選択手法を行ったときのAUCと選択件数とデータ選択のループの回数を示した。

この結果から、閾値limを設定することは今回のデータセットにおいては性能改善に寄与しないことがわかった。6.1でも述べたように、選択件数7,000件程度と比較的多いことから、今回の実験環境ではデータ選択手法が過剰な環境ではなかったことが原因として予想される。

7 まとめと今後の課題

機械学習モデル利用の課題であるmodel decayに対して、ドリフトしていないデータを選択してモデルの更新を行うことで、性能の良いモデルを継続的に入手する枠組みの提案を行った。また、その枠組みの中で利用する、ドリフトの影響を受けていないデータを判別するためのデータ選択指標の検討を行った。

今回の実験では、つねに新しいコンセプトへ移動するタイプの人工データセットで検証を行ったため、バッチnの予測を行うために直前のバッチ(バッチn-1)よりも古いデータを学習に用いることの有効性を確認することができなかった。しかし、ドリフトに含まれるコンセプトは再帰的な変化を起こすことも考えられるため、データセットを変えてデータ選択手法の効果を確認していきたい。

文 献

- [1] 今野 由麻, 中野 美由紀, 小口 正人, “コンセプトドリフト対処のための、Adversarial Validation を用いた学習データ選択に関する検討”, 第14回データ工学と情報マネジメントに関するフォーラム (DEIM2022), D33-1.
- [2] Yuma Konno, Miyuki Nakano and Masato Oguchi, “Efficient Data Selection Indicators for Updating Models under Data Drifted Environment”, 2022 IEEE International Conference on Big Data (Big Data), 2022.
- [3] Jing Pan et al, “Adversarial validation approach to concept drift problem in automated machine learning systems”, CoRR, Vol. abs/2004.03045, , 2020.
- [4] Shotaro Ishihara et al, “Adversarial Validation to Select Validation Data for Evaluating performance in E-commerce Purchase Intent Prediction”, <https://sigirecom.github.io/ecom21DCPapers/paper3.pdf>, SIGIR eCOM'21, 2021.
- [5] Miroslav Kubat and Gerhard Widmer. “Adapting to Drift in Continuous Domains,” OFAI, Vienna, 1994.
- [6] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, “Learning under Concept Drift: A Review,” in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346-2363, 1 Dec. 2019, doi: 10.1109/TKDE.2018.2876857.
- [7] Ali Pesaraghader, “The Tornado framework” [github.com. https://github.com/alipsgh/tornado](https://github.com/alipsgh/tornado) (accessed Aug. 6, 2022).

謝 辞

本研究は一部をJST CREST JPMJCR22M2の助成、一部をJSPS 科研費18K11318の助成を受けたものです。