小口研究室　研究紹介　(2021年度)
(お茶の水女子大学理学部情報科学科)

コンセプトドリフト対処のためのAdversarial Validationを用いたデータ選択に関する考察
（研究担当:今野 由麻）

研究背景

- 大量のデータが生成され、機械学習が広く利用されるようになった
- 一度学習したモデルでも、使い続けると精度が低下することが知られている
  → このような現象の原因として、コンセプトドリフトがある
- 精度が低下するたび、新しいデータでモデルを学習し直す等の簡単な手法(図1)
  → 過去の有用なデータも捨ててしまう

大量のストリームデータが生成される場面において、
学習データを適切に選んだモデルを継続的に入手する方法を考えたい

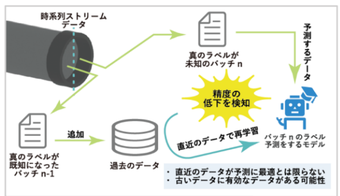


図1. デメリットのある簡単な手法の例

提案手法

- 本手法は、Panらによる先行研究[1]を拡張したものである
- 本実験で利用したデータセットRL(https://automl.chalearn.org/data)のように、時系列の複数のバッチを持ち、真のラベルが未知のバッチnが入手された時点でバッチn-1の真のラベルが得られるような状況を想定している
- 真のラベルが未知のバッチnの真のラベルを予測する機械学習モデルを得ることを今回の目的のタスクとしている

- 提案手法の流れ(図2)
  1. バッチnの真のラベルの予測を行うために利用できるデータとして、バッチ1からバッチn-1が保管されている
  2. 真のラベルが未知のバッチnに敵対的分類器のための新しいラベルy=1を付与し、過去のデータに対してy=0を付与する
  3. 真のラベルが未知のバッチnと過去のデータを見分けるような二値分類器である敵対的分類器を学習する
  4. 敵対的分類器の精度(AUCスコア)が0.8以下になるまで、過去のデータから分類精度の良いデータを省くことと敵対的分類器の再学習を繰り返す
     → 敵対的分類器の精度が悪くなるようなデータを選択することで、予測したいバッチと似たデータを抽出することができる
  5. 過去のデータのうち省かれずに残ったものを学習データとして、真のラベルが未知のバッチnのラベルを予測するモデルを学習する

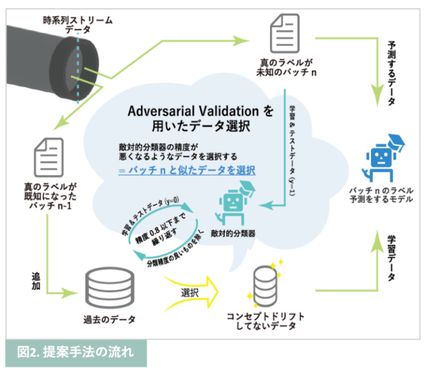


図2. 提案手法の流れ

[1] Jing Pan et al, “Adversarial validation approach to concept drift problem in automated machine learning systems”, CoRR, Vol. abs/2004.03045, , 2020.

データ選択のアルゴリズムと閾値

- データ選択の流れを以下の図3に示す
- アルゴリズム中で利用する分類確率として0と1のラベルに分類される確率のうち、大きい方(≥ 0.5)を採用
  → 敵対的分類器の精度が低くなるようなデータを選ぶことを優先したアプローチ
- このアルゴリズムには閾値が2つある
  - lim：過剰なデータ選択を予防する閾値
  - red：一度に削減するデータの数を決める閾値

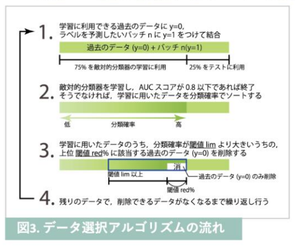


図3. データ選択アルゴリズムの流れ

- データセットRLにおいて、バッチ1からバッチn-1を用いて次のバッチnの予測を行う場合の、最適な閾値を調べる実験を行った(図4)
  - lim=0.85, red=0.05で最良の結果
  - redは小さい方が精度が良い傾向となった

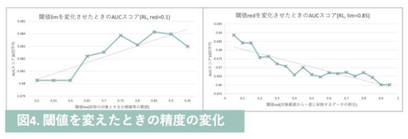


図4. 閾値を変えたときの精度の変化

実験結果

- 4つのバッチを持つデータセットRLを利用した実験の概要
  - ここでは表1に示すように(a)-(c)の3通りのデータ選択を行うバッチの選び方を用意して実験を行った
  - データ選択の閾値は、図4の実験で最も良い精度を出したlim=0.85, red=0.05を利用した
  - 提案手法の結果(実線)と、データ選択を行わずに予測を行ったベースラインの結果(点線)を図5に示した

表1. データ選択を行うバッチの選び方

名称	説明	バッチ n の予測に用いる過去のデータ
(a)	入手可能な全データを利用	バッチ 1 … バッチ n-1
(b)	直前のバッチのみ利用	バッチ n-1
(c)	一番古いバッチのみ利用	バッチ 1

- 実験結果から読み取れること
  - 一部の予測では、データ選択を行うことでベースラインと比較して精度を向上させることができたが、その精度の改善幅は小さい
  - バッチの選び方(a)のバッチ3と4の予測結果はベースラインより精度が悪くなってしまった
    → 閾値が不適切な可能性や過剰なデータ選択が発生している可能性が考えられる
  - 選び方(c)では、予測精度が一度下がった後に上向いていることから、古いデータが未来の予測に有用である実例を確認できた

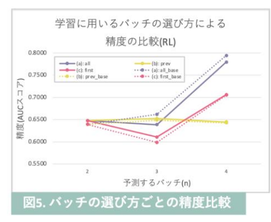


図5. バッチの選び方ごとの精度比較

まとめと今後の課題

- 先行研究の手法をベースとして、Adversarial Validationを用いてデータ選択を行うことで自動でドリフトに適応可能なシステムを提案した
- 学習に利用するデータセットごとに適切な閾値が変わる可能性がある
  → 適切な閾値を検討するための調査が必要
- 過剰なデータ選択が発生しているケースが散見された
  → 閾値の調整によって、どのくらい過剰なデータ選択が抑え込めるのか調査が必要
  → 閾値の調整だけで解決しない場合は、アルゴリズムの改善が必要

サーバシステムの性能データ収集および転送における効率化手法の検討
（研究担当:飯山 知香）

研究概要

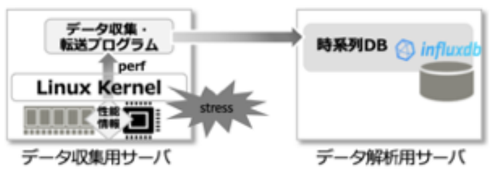
◆ 背景
- □ クラウド環境をはじめとした、多数台サーバの共有利用や分散処理利用に関する需要の増加
  - サーバの負荷分散や、システムやアプリケーションのチューニングを行うには、各サーバの低レイヤを含めた性能データを低オーバヘッドで収集してリアルタイムに分析・提示する手法が必要
- □ 多数台のサーバのデータを一元的にリアルタイムで分析する際、分析対象のサーバとその分析を行うサーバは分割されることが多い
  - Linuxの性能データなどの時系列データはデータサイズが比較的大きく、扱う際のオーバヘッドが大きくなる可能性がある

◆ 目的
効率的に性能データを収集・転送する手法の実現

実験概要

◆ 実験方針
データ収集用サーバとデータ解析用サーバが分離した環境で、データ収集/転送/分析がボトルネックとなるか調査
1. CPUやOSから性能データを収集＆時系列化
2. 時系列性能データをデータ解析用サーバに転送
3. 時系列性能データをDBに格納＆分析

※性能データには、perf recordによって取得したプロファイルデータを使用



◆ 転送データの作成～転送
1. 収集した性能データから必要なデータ(時刻、プロセスID、スレッドID、実行アドレス)のみ抽出
   - 1CPUコアのデータを1ms(100μs)単位で1秒間収集した場合は1000レコード(10000レコード)
   - 64bit×4項目×1000レコード(10000レコード)=32KB(320KB)
2. 抽出したデータをDataFrame形式に格納
3. InfluxDBのPythonモジュールを使用してデータ解析用サーバへ転送

時刻(ns単位)	Process ID	Thread ID	EIP(実行アドレス)
2021-12-20T13:18:36.159557443	698085	698085	0x55e3e9dd1151
2021-12-20T13:18:36.160568219	698085	698085	0x7faa24bd7e02

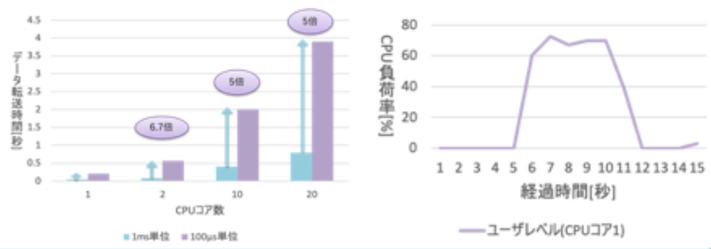
実験

◆ データ転送時間(1ms単位)
- ・収集対象のCPUコア数増加によるデータ転送時間の変化を比較した
- ・25CPUコア以上の多数コアCPUでは1秒間相当のデータを転送するのに1秒以上かかる
  →転送効率化が必要(データ転送プログラムの高速化、転送データの簡易分析、転送データの圧縮etc)

◆ データ転送時間(100μs単位)
- ・1CPUコア分のデータサイズ増加によるデータ転送時間の変化を比較した
- ・転送データサイズの増加率(10倍)に比例せず、転送時間はそれぞれ元の4～7倍程度増加した
  →データ本体の転送以外の、InfluxDBサーバとのネゴシエーション等のコストが大きく、時間がかかっていると推察される

◆ CPU負荷率
- ・データ収集用サーバ上で実行されるデータ転送プログラム、データ解析用サーバ上で実行されるInfluxDBのCPU負荷率を計測した
- ・データ転送プログラムのCPU負荷率が60%以上と高い
  →CPU負荷改善が必要(転送専用のCPUコアの用意、プログラム改善、転送処理の並列化etc)

◆ DB増加量
- ・格納されたデータの圧縮率を見るために、データ解析用サーバ上のDB増加量を確認した
- ・転送前後でのDB増加量は転送データの半分以下だった
  →今回利用したプロファイルデータは圧縮効果が高い



今後の課題

・検討した転送効率化/CPU負荷改善手法を試行
・InfluxDBサーバとのネゴシエーション等の詳細、プロファイルデータによる圧縮率の差異について調査

ユーザの位置情報プライバシを考慮したSNSデータからのイベント情報検索手法
（研究担当:石神 京佳）

研究概要

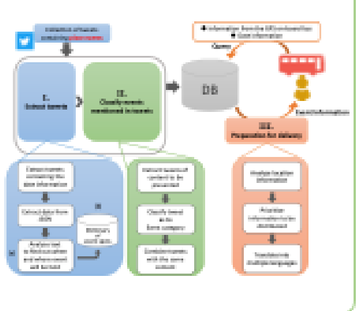
◆ 背景
- ● ソーシャルネットワーキングサービス（以降SNSとする）利用者の増加
  …SNS上には固定的なメディアに載っていないような大小様々な有益なイベント情報が存在
- ● 近年におけるデータの収集、分析技術の飛躍的な進歩
  …大量のデータとユーザの情報を組み合わせ、ユーザの意思決定をサポートする推薦サービスに応用する動きが盛ん
- ● サーバにて情報分析にユーザの情報を利用する際のプライバシリスク
  …データ分析/情報推薦の精度の保持とユーザのプライバシの保護の両立が課題

◆ 目的
クライアント側のユーザのプライバシに関わるような情報の保護処理と、サーバ側のSNSデータの分析結果を組み合わせユーザの位置情報プライバシを考慮したSNSからのイベント情報抽出、推薦システムの構築

先行研究

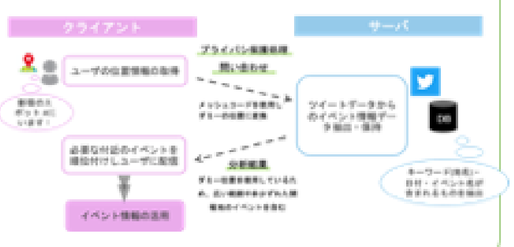
◆ SNS データを用いた場所と時間を考慮するイベント検索手法の提案と評価(お茶大・工藤ら[DICOMO2018])
- ● 時間移動する旅行者などにリアルタイムにSNS上の有益な情報を配信するシステム

1. Twitter上の地名に紐づいたツイートを収集→イベント情報の抽出
2. イベント情報を分析しイベントを分類
3. ユーザの位置情報をもとに有益度順にイベント情報を提示



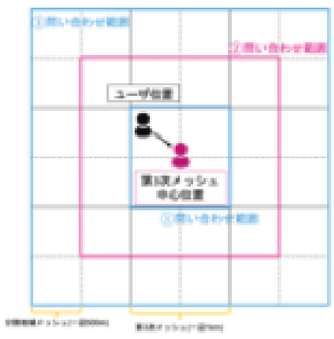
提案システム

◆ 概要
- ● プライバシに関わるユーザの位置情報は、プライバシ保護処置を施したものをサーバに問い合わせに使用
- ● サーバ側は先行研究の手法を用いてツイートデータからのイベント情報抽出、データベースへの格納を行う



実験

◆ 方法
- ● 地名キーワードを含むツイートの収集
  ① 期間：2021年11/20から11/24の計5日
  キーワード：「原宿、渋谷、新宿、赤坂、代々木、池袋、六本木」を含むツイート
  ② 期間：2021年12/10から12/14の計5日
  キーワード：「立川、足立、八王子」を含むツイート
- ● 正規表現を用いて日付とイベント名が含まれるツイートを収集
- ● イベント開催地にスポット名または住所が含まれるものに対しGeolocation APIを用いて緯度・経度を取得し保存→緯度・経度から地域メッシュコードを算出
- ● 第3次メッシュの中心位置をダミーの位置としてサーバへの問い合わせに使用、右図のように問い合わせ範囲を段階的に拡大



◆ 問い合わせ結果(右表)
- ● 7種のユーザ位置の第3次メッシュ中心位置から範囲を広げ問い合わせ
- ● ユーザ位置によって、範囲を広げた問い合わせをしてもイベント取得数が増加しない
  →イベント開催地がターミナル駅付近に集中しているため

ユーザ位置	問い合わせで得られたイベント数(個)		
	1回目問い合わせ	2回目問い合わせ	3回目問い合わせ
新宿三丁目駅	44	144	180
東京都庁	2	122	178
代々木駅	0	0	35
東京芸術劇場	29	39	45
東京ドーム	1	1	10
六本木ヒルズ	17	19	30
八王子駅	7	7	7

今後の課題

◆ イベントの密集状況に応じた、柔軟な範囲の拡大の検討
◆ ツイートから得られたイベント情報の精査
◆ ユーザへの情報推薦に位置情報以外のユーザデータの使用を検討（ユーザの趣味趣向、会話のログ等）

http://ogl.is.ocha.ac.jp/
oguchi@is.ocha.ac.jp