2A2-3

アンサンブル学習を用いたConcept Driftへの適応手法

Adaptive Ensemble Learning Method for Drifting Concepts

木谷 奈穂 Naho Kitani　安村 禎明 Yoshiaki Yasumura　上原 邦昭 Kuniaki Uehara

神戸大学大学院自然科学研究科
Graduate School of Science and Technology, Kobe University

In this paper, we present an algorithm to detect concept drift with ensemble learning. The data is assumed to arrive in chunks large enough to train base classifiers. This algorithm builds an ensemble of classifiers by AdaBoost per chunk. Concept drift is detected based on the difference between the prediction error of the latest ensemble and the previous ensembles. The experimental results using the moving hyperplane detasets show that this algorithm can detect smooth drifts and outperforms learning algorithms that ignore concept drift.

1. はじめに

近年，concept drift を含むデータストリームに対応したマイニング技術への関心が高まってきている．concept drift とは，データから学習しようとする concept が時間的に変化することを指す．たとえば，情報フィルタリング問題においては，ユーザの嗜好は時間が経つにつれて変わっていくため，その変化を素早く察知し，対応することが必要となる．

このような concept drift への対応を試みた手法は数多く提案されており，大別すると以下の 3 種類となる．time window を適応的に調節する手法，現在の concept を基に事例に重み付けを行う手法， そしてアンサンブル学習を用いる手法 [Wang 03][Scholz 05] である．アンサンブル分類器を用いたアプローチは，concept drift への柔軟な対応が期待できるため，注目を集めている．

本研究では，代表的なアンサンブル学習手法である AdaBoost[Freud 95] に着目し，AdaBoost を用いることで concept dirft の検出と分類精度の向上を目指す．AdaBoost の分類境界付近の事例に対して敏感であるという性質を活かし，発見が困難となる緩やかな drift の検出を試みる．

2. 問題定義

本研究では，訓練事例 $\{\boldsymbol{x} \in X, y \in Y\}$ が複数個集まった塊（chunk）として逐次的に与えられる状況を考える．一つの chunk に含まれる事例数（*chunksize*）は一定であると仮定する．

$$\underbrace{z_{(1,1)}, \cdots, z_{(1,m)},}_{\text{chunk 1}} \underbrace{z_{(2,1)}, \cdots, z_{(2,m)},}_{\text{chunk 2}} \cdots \underbrace{z_{(t,1)}, \cdots, z_{(t,m)},}_{\text{chunk t}} \cdots$$

3. AEL アルゴリズム

本節では，提案手法である AEL（Adaptive Ensemble Learning）アルゴリズムについて述べる．この手法の特徴は，AdaBoost アルゴリズムを用いることで concept drift の検出，また drift が発生していない区間での分類精度の向上を試みている点にある．アルゴリズムの流れを以下に示す（図 1）．

1. 訓練集合として新しく chunk t が与えられると，AdaBoost アルゴリズムにより height 個の分類器を生成する．

連絡先: 木谷 奈穂，神戸大学大学院自然科学研究科情報知能工学専攻，E-mail: naho@ai.cs.scitec.kobe-u.ac.jp

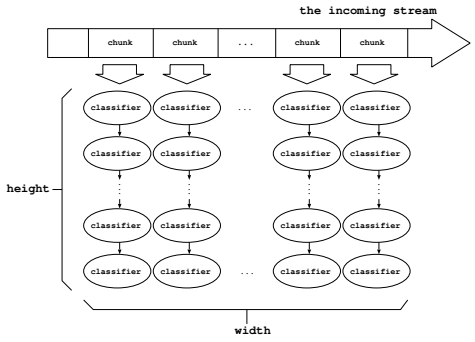

図 1: AEL アルゴリズムの概要

2. width 個のアンサンブルが構築済の場合，最古のアンサンブルを削除する．

3. 前回までに生成したアンサンブルと最新アンサンブルの chunk t に対するエラーを比較し，閾値 α より大きければ concept drift が検出されたとし，アンサンブルの再構築を行う．閾値以下であれば，chunk t を基に以前の全分類器の重みを更新する．ただし，アンサンブル数が width 個未満の場合は α の値に関係なく再構築は行わないものとする．データストリームが終了するまで，以上の操作を繰り返す．

AEL アルゴリズムをまとめると図 2 のようになる．各時点での分類予測には以下の式を用いる．

$$h_{fin}(x) = \arg\max_{y \in Y} \sum_{t,k:h_{tk}(x)=y} \log \frac{1}{\beta_{tk}} \qquad (1)$$

4. 性能評価

4.1 実験設定

提案手法の性能を評価するため，moving hyperplane を基にした人工データを用いて実験を行った．d 次元空間の超平面は $\sum_{i=1}^{d} a_i x_i = a_0$ のように表され，各事例へのラベル付けは $\sum_{i=1}^{d} a_i x_i \geq a_0$ のとき positive，それ以外を negative とする．多次元空間 $[0,1]^d$ に一様分布するランダム事例を生成し，

Algorithm AEL
Given: S: a dataset of *chunksize*
height: the number of iterations for AdaBoost
width: the number of ensembles
α: the threshold for reset
while not end stream, do
1. $h_{t1} = BaseLearn(S_{t1})$.
2. For $k = 1, \cdots, height$,
Calculate the error of h_{tk}:
$$\epsilon_{tk} = \frac{\sum_{i:h_{tk}(x_i) \neq y_i} weight(x_i)}{chunksize}.$$
set $\beta_{tk} = \epsilon_{tk}/(1 - \epsilon_{tk})$.
For each $x_i \in S_{tk}$,
if $h_{tk}(x_i) = y_i$
then $weight(x_i) = weight(x_i) \times \beta_{tk}$
standardize S_{tk} to *chunksize*.
3. Delete the oldest ensemble.
4. Calculate the error of the latest ensemble and the previous ensembles:
$$h(x) = \arg\max_{y \in Y} \sum_{t,k:h_{tk}(x)=y} \log \frac{1}{\beta_{tk}}.$$
If $\epsilon_{previous} - \epsilon_{latest} > \alpha$
reset all ensembles
5. Update β_{tk} with the latest chunk.

図 2: AEL アルゴリズム

等しい大きさの空間に分割する a_0 を $a_0 = \frac{1}{2}\sum_{i=1}^{d} a_i$ と設定する．重み $a_i (1 \leq i \leq d)$ を変化させることで concept drift 環境を実現する．本実験では全次元数 $d = 5$，変化させる次元数 5 とし，1000 事例毎に drift を加えた．テストデータは次回の chunk とする．比較アルゴリズムには，（1）シングル分類器，（2）AEL アルゴリズムから drift 検出による再構築操作を除いたもの，の 2 つを用いた．AEL アルゴリズムの各パラメータは $chunksize = 100$，$height = 10$，$width = 5$，$\alpha = 4.5$ と設定する．ベース分類器には SVM（Support Vector Machine）を用いた．

4.2 実験結果と考察

AEL アルゴリズムにおいて concept drift 検出の指標とした分類エラー差の有効性を示すため，時間的変化を調べる実験を行った（図 3）．図 3 をみると，10 chunk 毎に顕著な値の伸びを確認することができる．これはちょうど drift 発生点に相当するため，分類エラー差が drift 検出に有効であることがわかる．図 4 は（1）シングル分類器，（2）AEL アルゴリズムから drift 検出による再構築操作を除いたもの，（3）AEL アルゴリズム，の 3 手法の分類精度を時間軸に沿って比較したグラフである．シングル分類器は，その時点までに得られた全てのデータを訓練事例として用いているため，concept drift を含むデータでは時間が経つにつれて精度の低下が見られる．ところが AEL アルゴリズムでは，古いデータの情報を随時削除しているため，比較的優れた精度を維持している．また drift 検出機構により，drift 発生による精度の低下からの素早い回復が実現された．これは，AEL アルゴリズムから drift 検出による再構築操作を除いたものとの比較から確認することができる．$t = 60$ までの分類精度の平均は（1）79.7%，（2）93.7%，（3）95.9% となり，AEL アルゴリズムが最も優れた精度を達成した．

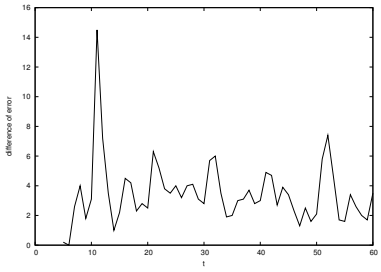

図 3: 分類エラー差の時間的変化

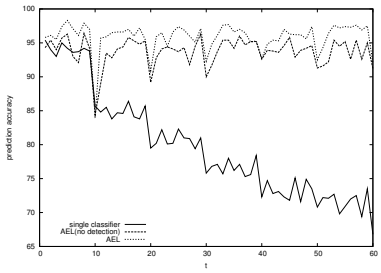

図 4: 分類精度比較

5. まとめ

本稿では，AdaBoost アルゴリズムによるアンサンブル分類器を用いた concept drift への適応手法を提案した．AdaBoost を用いることで，緩やかな drift の検出が可能となり，結果として全体的な分類精度の向上へと繋がった．drift 検出後のアンサンブル削除法には更なる検討の余地があるため，今後は drift の強弱によってアンサンブルの形態を変えていくようなシステムの構築を考えていくつもりである．また，*chunksize* への対応も考慮する必要がある．

参考文献

[Freud 95] Freud, Y. and Schapire, R. E. : A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, Vol. 55, pp. 119-139 (1995).

[Wang 03] Wang, H. , Fan, W. , Yu, P. S. and Han J. : Mining Concept-Drifting Data Stream using Ensemble Classifiers, Proc. of the 9th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining, pp. 226-235 (2003).

[Scholz 05] Scholz, M. and Klinkenberg, R. : An Ensemble Classifier for Drifting Concepts, Proc. of the 2nd Int. Workshop on Knowledge Discovery from Data Stream at ECML/PKDD 2005, pp. 53-64, Porto, Portugal (2005).