ISSN 2186-5647

2-38

Multi-Scale Batch-Leanring によるバランスの取れた自己増殖型クラスタリングの高速化

日大生産工　○嶋崎　愛也　　日大生産工　山内　ゆかり

1. まえがき

クラス不均衡学習はクラス分布が非常に偏っているデータを扱い、実世界のアプリケーションによく見られる。増加学習は、新しいデータを使って継続的にモデルを訓練する機能があり、古い情報を忘れることなく、新しい情報を学習する必要がある。これらの2つの問題は議論されてきたが、それらは十分に研究されていない。

そこで,Yue Shaoらはバランスの取れた自己増殖型ニューラルネットワーク（Balanced SOINN）[1]を提案している。SOINNにオーバーサンプリング法[2]を追加することでバランスの取れたSOINNにしている。Balanced SOINNを他の方法と比較し、非増分シナリオの人工データセットでは競争力があり、増分シナリオの実世界のデータセットでは最高のパフォーマンスを実現すると報告されている。

Fernando Ardillaらは複数のタスクを同時に学習できるモデルMulti-Scale Batch-Leaning Growing Neural Gas[3]を提案した。MSBL-GNGは各タスクのデータを共有することで関連するタスク間の情報を相互に利用して学習する。これにより、各タスクのパフォーマンスが向上し、全体として効率的な学習が可能になる。

本研究では、BSOINNはクラスタのバランスをとるために追加で計算をするため、SOINNに比べ処理速度が落ちる。特に大量のデータや高次元のデータを扱う場合計算負荷が大きくなる。そこでBalanced SOINNにMulti-Scale Batch-Leaningを導入することで処理速度の削減するMSBL-BSOINNを提案する。

2. 従来研究

Balanced SOINNは元のSOINNに基づいて開発され、3つの主要な改善が行われた。1つはラベルが導入され、アルゴリズムが教師あり学習に適応できるようになった。ノードの追加と更新は距離と閾値だけでなくラベルにも依存する。ノイズ削除部分も調整され、少数クラスのノードが削除される可能性が低くなった。2つ目はクラス内挿入は、新しいノードの挿入にほとんど貢献せず、必要なパラメータが多すぎるため削除された。3つ目はノードの重要度に基づいてオーバーサンプリング方式が採用された。それにより、データは適切に再バランスされる。

2.1 Balanced SOINNアルゴリズム

Balanced SOINNアルゴリズムのフローチャートをFigure 1に示す。

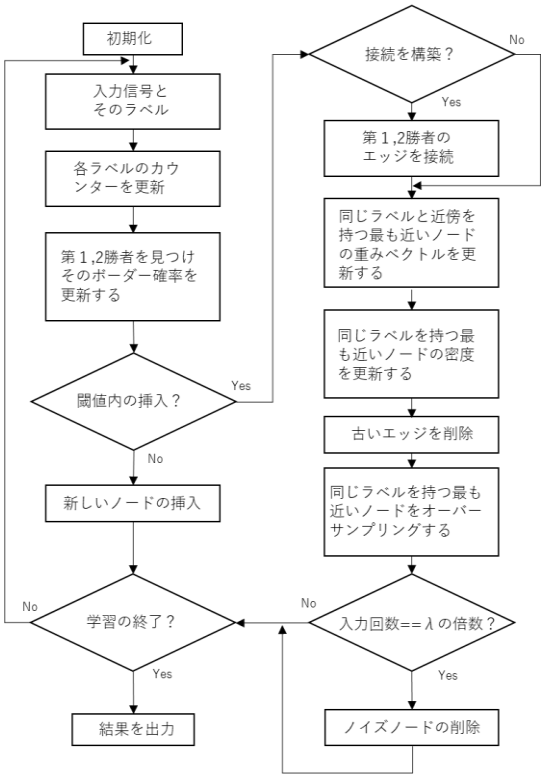

Figure 1　Balanced SOINN のフローチャート

・Step1

ノード数、エッジ、カウンターマップの初期化、入力信号とそのラベルをSOINN空間に与える。

・Step2

アルゴリズムはラベルのカウンターを更新する。最初の2つの入力信号であれば、以下の式に従って直接加算し、Step2に進み学習を続ける。

$$N = N \cup \{< \xi, \text{label} >\}$$

・Step3

Acceleration of Balanced SOINN by Multi-Scale Batch-Leaning

Manaya SHIMAZAKI and Yukari YAMAUCHI

入力からのユークリッド距離を用いて第１勝者、第２勝者の探索する。

$$S_1 = argmin\|\xi - W_t\| \quad (1)$$

$$S_2 = argmin\|\xi - W_t\| \quad (2)$$

S1、S2のボーダーノードである確率の更新する。

・Step4

入力値が類似度閾値に従って第１、２勝者の異なるクラスターに属している場合、または第１、２勝者のラベルのいずれも信号と同じでない場合は、新しいノードとして挿入が行われる。もしノードsが近傍ノードがある場合(3)式の上記の式を使用する。近傍ノードを持っていない場合(3)式の下記の式を使用する。

$$T_s = \begin{cases} argmax\|W_s - W_t\| \\ argmix\|W_s - W_t\| \end{cases} \quad (3)$$

閾値外の入力→Step5

閾値内の入力→Step6

・Step5

新しいノードとしてSOINN空間に挿入する。

→Step12

・Step6

S_1とS_2間にエッジが存在するか判断する。

・*Step7*

S1とS2の間にエッジが存在せず、$label_{s1}$と$label_{s2}$が同じ場合エッジ作成し、S1,S2間のエッジの年齢を0にする

・Step8

入力値と同じラベルを持つ最も近いノードsを見つける。sとつながっているすべてのエッジの年齢を1増加させる。sとsとつながっているノードの重みベクトルを更新する。

ノード s の重み更新式

$$\Delta W_s = \epsilon_1(t)(\xi - W_s) \quad (4)$$

ノード s とエッジが接続されているノードの重み更新式

$$\Delta W_i = \epsilon_2(t)(\xi - W_i) \quad (5)$$

次のような式で学習率を調整する。

$$\epsilon_1(t) = \frac{1}{t} \quad (6)$$

$$\epsilon_2(t) = \frac{1}{100t} \quad (7)$$

・Step9

ノードsの密度ベクトルを更新する。

・Step10

あらかじめ設定されたAGEMAXより大きい年齢のエッジを削除する。その結果、エッジがなくなったノードも削除する。

・Step11

ノードの重要度の確率でノードsをオーバーサンプリングする。ノードの重要度の確率が以下の図２のフローチャートで計算される。

・Step12

学習回数がλ倍になったときノイズノードを削除する。

・Step13

Step2に進み、学習回数を満たすまで学習を続ける。

2.2.1サンプリング法

オーバーサプリング法を使用してデータ分布のバランスを再調整し、クラスの不均衡の問題を克服する。サンプリング法は、入力信号が来たときに評価されるノードの重要度に従い手動で設定することなく、クラス分布のバランスを再調整する。信号が少数派クラスから来た場合、オーバーサンプリングされる可能性が高くなり、ノードの重要度が高くなる。ノードの重要度が高いということは、信号が少数派のクラスからのものであるか、高密度領域にあるか、異なるクラスの境界上にあることを意味する。

2.2.2 サンプリングアルゴリズム

信号iが来て、アルゴリズムが同じラベルを持つ最も近いノードsをオーバーサンプリングする場合、アルゴリズムを使用してノード重要度を計算する。不均衡指数は、すべての信号の数に対する1つのクラスのサンプル割合であるため、少数派クラスのノードは不均衡指数が小さいため、これを使用してオーバーサンプリングするかどうか決定する。ノードの重要度が高いということは、信号が少数派クラスからのものであり、高密度領域にあるか境界に近いことを意味する。ノードの重要度を使用して、オーバーサンプリングによって生成されるサンプルの数を決定する。ノードの重要度は不均衡指数に反比例するため、アルゴリズムはさまざまなレベルの不均衡データに適応できる。不均衡指数が低いノード$\frac{1}{class\ number} - 0.1$未満の場合、少数派クラスからのノードであると想定し、さらにオーバーサンプリングする。不均衡指数が高いノード$\frac{1}{class\ number} - 0.1$より大きいが$\frac{1}{class\ number}$未満の場合、境界ノードである確率と密度に応じてオーバーサンプリングする。ノードのBorderProbと密度は、式(9)で定義されている。不均衡指数が$\frac{1}{class\ number}$より大きいノードは、

多数派クラスである可能性が高いため、オーバーサンプリングを行わない。ノードaをオーバーサンプリングをすると決めると、SMOTEアルゴリズムはk個の最も近い少数派クラス近傍ノードbの1つをランダムに選択し、aとbを接続して特徴空間に線を形成する。aとbの線から新しいノードがランダムに選択される。ノードの重要度が1より大きい場合、オーバーサンプリングプロセスは複数回繰りかえされる。

2.3 ノードの重要度評価

少数派クラスからの信号は高密度領域にあるか、中心から遠く離れているため、ノードの重要度が高くなる。新しい信号＜ξ,label＞が来ると、それが少数派クラスからのものであるかどうかは、アルゴリズム１のカウンターマップCによって判断される。まず、式(8)でノードiの平均距離d_iが近傍ノードから計算される。mはノードiの近傍ノードの数、W_iはノードiの重みベクトルである。

$$\bar{d}_i = \frac{1}{m}\sum_{j=1}^{m}\|W_i - W_j\| \qquad (8)$$

ノードiのポイントは次のように計算される。

$$P_i = \begin{cases} \dfrac{1}{(1+\bar{d}_i)^2} & if node\ i\ is\ winner \\ 0, & if\ node\ i\ is\ not\ winner \end{cases} \qquad (9)$$

ノードの累積ポイントの平均は、そのノードの密度を表す。さらにノードが異なるクラスの境界にある場合は境界ノードと定義する。境界ノードは、クラス分布の中心にあるノードよりも異なるラベルの隣接ノードが多いため、異なるラベルのノードの数を使用して、ノードが境界ノードである確率を表す。新しい信号が来たときに、勝者のラベルが異なる場合、勝者ノードの異なるラベルの数は1増加する。ノードが境界ノードである確率は次のように定義される。WinnigTimeはノードが勝者になった回数をカウントしている。

$$BorderProb_i = BorderCount_i / WinnigTime_i \qquad (10)$$

2.4.1 ノイズ削除

実際のデータセットにはノイズが存在する、BalancedSOINNは、トレーニング中にノイズからノードを生成する。そこでノイズ低減アルゴリズムを採用し、少数派クラスからのノードの削除を減らすための正当化を行う。ノードが多数派クラスに属するかどうかを不均衡指数によって判断される。不均衡指数は、1つのクラスのサンプルがすべてのクラスの数占める割合であり、不均衡指数が高いということは、ノードが多数派クラスに属することを意味する。ラベルnを持つノードiの不均衡指数は次のように定義される。

$$ImbalanceIndex_i = \frac{counter\ mapC[label_i]}{sum(counter\ map\ C)} \qquad (11)$$

密度の低い領域にあるノードは他のノードに接続されたエッジが少なく、ノイズの多いデータから生成される可能性が高くなる。ただし、少数派クラスのノードの場合、その数が少ないため、他のノードよりもエッジが少なくなることがよくある。

2.4.2 ノイズ削除アルゴリズム

・Step1:ニューロン集合ノードiについて、ノードiに最も近いk個のノードを見つける。このときのkは初期パラメータで決定する。

・Step2: k 個の最近傍ノードがノードiと異なるクラスラベルを持つか、または以下の不等式を満たす場合、ノードiはminedge未満であり、ニューロン集合Nから削除される。

$$ImbalanceIndex_i = \frac{1}{C} - 0.1 \qquad (12)$$

・Step3:ノードiがminedge以上持っている場合、最も近いk個のノードの中にノードiと同じクラスラベルを持つノードがなければ、それをニューロン集合Nから削除する。

・Step4:すべてのノードの処理が終わるまでこのプロセスを繰りかえす。

3 提案手法

BalancedSOINNはクラスタのバランスをとるために追加で計算をするため、SOINNに比べ処理速度が落ちる。特に大量のデータや高次元のデータを扱う場合計算負荷が大きくなる。そこでBalancedSOINNにMulti-Scale Batch-Leaningを導入することで処理速度の削減するMSBL-BSOINNを提案する。

MSBL-BSOINNは、従来のBSOINNに比べ変更点は大きく分けて2つある。1つはバッチ学習を導入する。3つのPhaseに分けられPhase1ではデータ数DをD/8ごとに式(13)を使用してミニバッチ学習を行う。Phase2からPhase3はデータ数の分割をD/4、D/2にしてフルバッチ学習を行う。もう1つの変更点は成長段階に応じて探索する勝利ノードの数がPhase1では第1勝者のみであり、Phase2では第1、第2勝者の探索になり、Phase3では第3勝者までの探索を式(1)と式(2)と同じように行う。新規ノード追加にも閾値を使用していたが、式(14)の確率$p_k(v)$を用

いて新規ノードをネットワークに追加するか判断する。式(15)にあるεは少量の値である。

$$w_i \leftarrow w_i + \frac{\Delta w_i}{x_i^1 + x_i^2}, if\left(x_i^1 + x_i^2\right) > 0 \qquad (13)$$

$$p_k(v) = \max\left(0, \tanh\left(\frac{z_k(v)}{l_{range}}\right)\right) \qquad (14)$$

$$z_k(v) = \sum_{i=1}^{k} \gamma_{k,i}\, d_{si}, l_{range} = l_{max} - l_{min} + \varepsilon \qquad (15)$$

ミニバッチ学習の導入により、新規ノードの追加戦略をノイズノード削除の前で行うと追加後に即削除になるため削除戦略後に行う。バッチサイズは可変であり、λの倍数に設定する。

4　実験および検討

Optdigitのデータを用いて正解度の比較を行おうと思う。データ数3823個、ラベルの数は「0」~「9」の数字の10種類のデータセットを用いて提案手法と他の手法SOINNとBSOINNと比較する。

5. まとめ

本研究では、Balanced SOINNの欠点である計算速度の低下に焦点をあて、Multi-Scale Batch-Leaningを導入したMSBL-BSOINNを提案することで処理速度の削減されることが考えられる。ベイズ最適化などの自動化されたハイパーパラメータ最適手法を使用することで最適なパラメータセットを見つけることが可能になると思われる。

参考文献

1) Yue Shao, A self-organizing incremental neural network for imbalance learning(2023)

2) N.V.Chawla, SMOTE: Synthetic Minority Over-sampling Texhnique(2002)

3) Fernando Ardilla, Batch Learning Growing Neural Gas for Sequential Point Cloud Processing(2022)