ISSN 2186-5647

2-5

教師あり SOINN における高次元耐性の向上

日大生産工　○土田　悠介　　日大生産工　山内　ゆかり

1. まえがき

近年、ビッグデータに対する研究が発達している。そのような実世界のデータはノイズが多く、分布の形状が変化する、データの量が莫大に増え続けていく等扱いにくい性質がある。より実世界のデータに即した解析手法のなかに、長谷川らによるSOINN(Self-Organizing Incremental Neural Network)[1]という教師なし学習アルゴリズムがある。これはノイズに強く、事前に分布の形状を決定する必要がなく、追加学習が可能という特徴がある。

SOINNは教師なしの過程を経て形成されることが一般的ではあるが、もし学習段階でクラス情報が考慮されるのであれば、データの分類精度の著しい改良が可能なことがわかっている。須藤らによって教師データを入力次元に加えてSOINNと同じアルゴリズムで学習するSOINN-AM[2]を提案されている。しかしこれは重みベクトルの次元数に応じて教師ラベルの学習への影響が変わってしまう問題がある。ところで、同様の問題が自己組織化マップ(Self-Organizing Map:SOM)[3]でも起きており、佐藤らによって属性値SOM[4]という重みベクトルと属性値情報の競合学習を可能にした手法が提案されている。

本研究ではSOINNの高次元データにおける認識精度向上を目的とし、参照ベクトルと属性値情報の学習を可能にした教師ありSOINNを提案する。実験では手書き文字データセットMNIST[5]を使用し、認識率を従来のSOINNと比較し、提案手法の有効性を示す。

2. 従来研究

2.1.属性値SOM

SOM[3]とは、Kohonenが提案した教師ナシ学習アルゴリズムである。高次元データをベクトル量子化により特徴空間に写像することでクラスタリングや可視化することができる。

佐藤らによって提案された属性値SOINN[4]は、属性値を用いて学習可能なSOMである。属性値とはここでは、SOMのノードが各クラスにどの程度属しているのかを数値化したものである。属性値情報は学習率に重みを与える際やノードのクラス付けに用いる。属性値SOMのアルゴリズムを以下に示す。

まず、すべての出力層のノード$\boldsymbol{n}$の重みベクトル$\boldsymbol{W_n}$をランダムな値で初期化する。次に各ノードに対応する属性値$\boldsymbol{L}$を作成し、1で初期化する。$\boldsymbol{I}$個の入力データからランダムに選んだものを入力ベクトル$\boldsymbol{x}$とし、式(1)より勝者ノード$\boldsymbol{s}$を求める。

$$\boldsymbol{s} = \underset{i \in N}{\mathbf{argmin}} \|\boldsymbol{x} - \boldsymbol{W_i}\| \quad (1)$$

ここでNはSOMの持つノード全体である。
そして属性値$\boldsymbol{L}$の更新を行う。更新には式(2)を用いる。

$$\boldsymbol{L^i_{num}} = \boldsymbol{L^i_{num}} + \exp(\|\boldsymbol{r_s} - \boldsymbol{r_i}\|) \quad (2)$$

ここで$\boldsymbol{r_i}$はノード$\boldsymbol{i}$の位置ベクトル、$\boldsymbol{L^i_{num}}$はノード$\boldsymbol{i}$のもつ、入力ラベル$\boldsymbol{num}$の属性値である。
式(3)を用いて近傍関数$\boldsymbol{h_{si}}$を求める。

$$\boldsymbol{h_{si}} = \boldsymbol{\alpha_t} \exp\left(\frac{\|\boldsymbol{r_s} - \boldsymbol{r_i}\|}{2\boldsymbol{\sigma_t^2}}\right) \cdot \boldsymbol{\eta} \quad (3)$$

$$\boldsymbol{\sigma_t} = \boldsymbol{\sigma_0}\left(1 - \frac{\boldsymbol{t}}{\boldsymbol{T}}\right) \quad (4)$$

$$\boldsymbol{\alpha_t} = \boldsymbol{\alpha_0}\left(1 - \frac{\boldsymbol{t}}{\boldsymbol{T}}\right) \quad (5)$$

$$\boldsymbol{\eta} = \frac{\boldsymbol{L^i_{num}}}{\boldsymbol{max}\, \boldsymbol{L^i_j}} \quad (6)$$

$\boldsymbol{\alpha}$は学習率、$\boldsymbol{\sigma}$は学習範囲、 $\boldsymbol{h_{si}}$は近傍関数、$\boldsymbol{t}$は学習回数、$\boldsymbol{T}$は最大学習回数、$\boldsymbol{\eta}$は学習率に対する重み、$\boldsymbol{j}$はクラス番号を表す。また$\boldsymbol{\alpha_0}$、$\boldsymbol{\sigma_0}$は共に事前定義パラメータである。式(7)を用いて出力層の各ノードの重みベクトル$\boldsymbol{W_i}$を更新する。

$$\Delta \boldsymbol{W_i} = \boldsymbol{h_{si}}(\boldsymbol{x} - \boldsymbol{W_i}) \quad (7)$$

この操作を$\boldsymbol{T}$回繰り返すことで入力データを表現したマップが構築される。

2.2.SOINN

SOINN[1]は、一つの重みベクトルを保持するノードが自律的に増殖・消滅し、仮想的なノード同士のつながり(エッジ)を用いて構造化されるニューラルネットワークモデルである。現在SOINNには複数のアルゴリズムが存在し、その中でもAdjusted SOINNは、他のSOINNと比べてパラメータ数が少なく、計算時間が短いなどの優位点がある。そのため本研究では、Adjusted SOINNについて説明する。

まず入力$\boldsymbol{x}$に対して、既存のノードと類似度が最も近い2つのノードを近い方から勝者ノード$\boldsymbol{s_1}$、 $\boldsymbol{s_2}$として探索する。類似度は一般的にユークリッド距離が使われる。勝者ノード$\boldsymbol{s_1}$、 $\boldsymbol{s_2}$

Supervised SOINN with Improved High-Dimensional Robustness

Yusuke TSUCHIDA and Yukari YAMAUCHI

はそれぞれ式(8)および式(9)によって計算される閾値T_{s_1}、T_{s_2}を持ち、類似度と閾値の関係により挙動が変化する。

$$if\ N_i \neq \varnothing\ then$$

$$T_i = \max_{c \in N_i} \|W_i - W_c\| \quad (8)$$

$$else$$

$$T_i = \min_{c \in A \setminus \{i\}} \|W_i - W_c\| \quad (9)$$

ここで、N_iはノードiと直接エッジで繋がるノード集合、W_iはノードiの重みベクトルを表す。閾値を求めたら以下の図1のフローチャートに従ってノード挿入および重み更新を行う。条件分岐は式(10)を用いて行う。また、図2では条件分岐の視覚化を表す。

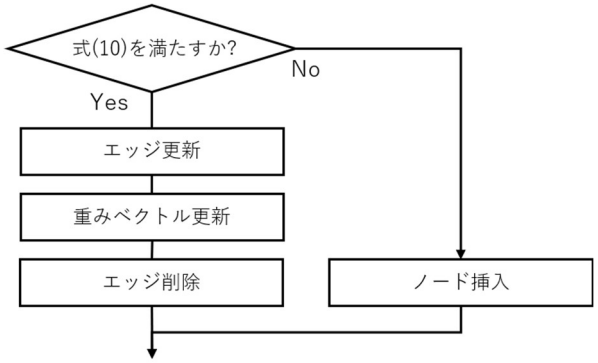

図 1. SOINN 空間更新アルゴリズム

$$T_{s_1} > \|W_{s_1} - W_x\| \wedge T_{s_2} > \|W_{s_2} - W_x\| \quad (10)$$

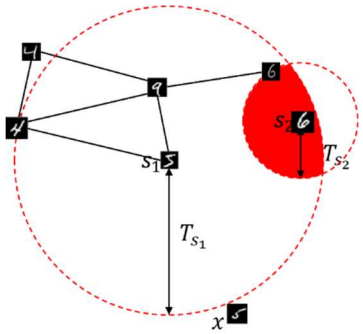

図 2. 閾値の可視化

類似度が閾値を片方でも超えた場合、分布の形成が不十分であると認識し、入力値を新たなノードとして挿入する。図2で色が塗られていない範囲である。類似度が両者の閾値の内側にあった場合、勝者ノード間のエッジと重みベクトルを更新する。図2で赤く色が塗られている範囲である。エッジの更新は、s_1、s_2間にエッジがついていない場合、エッジを年齢0として挿入する。すでにエッジがある場合はエッジの年齢を0にリセットする。その後s_1につながるエッジの年齢を1増加させる。そして、式(11)、式(12)に従い勝者ノードとその近傍ノードの重みベクトルを更新する。

$$\Delta W_{s_1} = \frac{1}{t_{s_1}}(x - W_{s_1}) \quad (11)$$

$$\Delta W_i = \frac{1}{100 t_i}(x - W_i) \quad (12)$$

$$(\forall i \in N_i)$$

ここでt_iはノードiが勝者に選択された回数である。その後、状態更新によりエッジ年齢が事前定義パラメータage_{max}を超えたエッジを削除し削除されたエッジに連結されていたノードのうちエッジを持たなくなったノードを削除する。最後に、入力パターンがノード削除周期λの倍数ごとに連結ノード数が1以下のノードを削除する。これらの操作を繰り返すことで、入力の分布をノードとエッジの集合で近似的に構成することができる。

3. 提案手法

SOM 系アルゴリズムの問題点としては、追加学習ができないことが挙げられる。SOINN は高次元データにおいてノイズ耐性を維持しつつ多様な特徴を保持することが困難であるという欠点がある。そこで本研究では高次元データに耐性のある追加学習可能な学習方法として教師あり SOINN を提案する。

まず教師ラベルとはデータに付加された所属クラスを表す。例えば図のようなデータだった場合、教師ラベルは 5 である。ここでC_{data}を画像$data$の持つクラスとし、図 3 の例では$C_{data} = 5$のように表すこととする。

図 3. 画像データと教師ラベル

ラベルを利用するための手法として、通常の SOINN に加えて各ノードに属性値Lを付与する。この属性値Lは、現在のノードが各クラスにどの程度属しているのかを数値化したものである。また、ノードのクラスを式(13)のように属性値のうち最も高い値のクラスと定める。

$$C_{node} = \underset{j}{argmax}\, L_{num}^{node} \quad (13)$$

新規ノードが挿入された際、新規ノードの持つ属性値は入力教師ラベルの値を 1、それ以外の値を 0 として初期化する。この属性値は、あるノードが第一勝者となった時に式(14)で更新する。

$$L_{num} = 1 - \frac{\|x - W_{s_1}\|^2}{D} \quad (24)$$

ここで num は入力データの教師ラベル、L_{num} はクラス num の属性値、xは入力データ、W_{s_1} は第一勝者の重みベクトルを表す。重みベクトルを更新する場合、式(15)で第一勝者ノードのもつ重みベクトルを更新する。

$$\Delta W_{s_1} = \frac{1}{t_{s_1}}(x - W_{s_1}) \cdot \eta \quad (15)$$

$$\eta = \frac{L_{num}^{s_1}}{max\, L_j^{s_1}} \quad (16)$$

ηは学習量に対する重み、jはクラス番号を表す。

SOINN は距離ベースのアルゴリズムであるため、次元が増えるほど共通部分の体積が相対的に減り、ノードが挿入される確率は高くなる。この時、第一勝者の近くに挿入されることがある。類似度の高いノードが複数できると、片方のノードがs_1となったときもう一方がs_2となりやすくなる。すると新しくエッジが作成されずノイズと判断され削除されてしまう。本研究ではこれに対応するため、二つの提案を行う。二つの提案手法を組み込んだフローチャートを図 4 に示す。

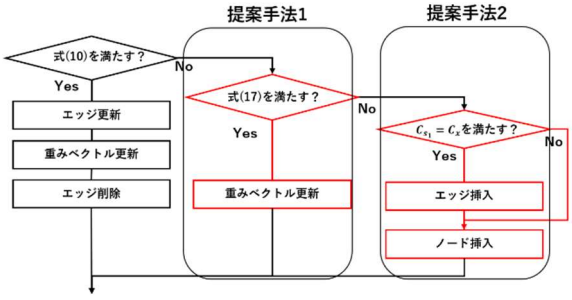

図 4. 提案手法の SOINN 空間更新アルゴリズム

一つ目は、入力教師データと第一勝者のクラスが等しかった場合、第一勝者の閾値内に入っていれば更新のみを行うことにする。条件式は式(17)である。

$$T_{s_1} > \|W_{s_1} - W_x\| \wedge C_{s_1} = C_x \quad (17)$$

従来手法は図 2 のように、共通部分に入力データが入った場合にエッジ作成および重み更新を行い、それ以外の場所に入力データがある場合はノードの挿入を行う。提案手法では共通部分の処理は変わらないが、入力教師データと第一勝者のクラスが等しい時かつ、図 5 の薄い色が付いた領域に入力データがあった場合に重みの更新のみを行う。

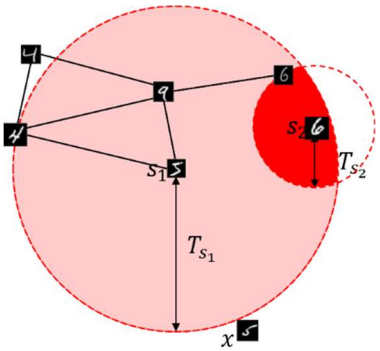

図 5. 提案手法 1 の閾値

この領域とは第一勝者の閾値内であり第二勝者の閾値外となる領域である。入力データのラベルと第一勝者のラベルが等しい場合、入力は第一勝者と類似性の高いノードであるため閾値を広げる。

二つ目はノードを挿入することになった場合、入力データのクラスと第一勝者のクラスが等しい時に入力データを第一勝者とエッジでつなげて挿入する。

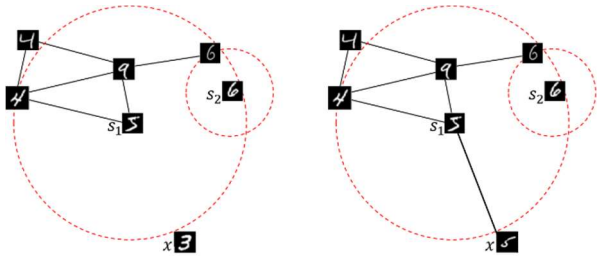

図 6. 提案手法 2 の様子

入力クラスがs_1と異なるクラスであれば図 6 左、通常の SOINN と同様に挿入する。入力クラスがs_1と同じクラスであれば図 6 右のようにs_1とエッジを繋いだまま挿入する。高次元だとエッジが付きにくいという問題を解消するために挿入時に、エッジを第一勝者とつなげて挿入することを提案する。

4. 実験環境

本実験では手書き数字画像データセット MNISTを用いて、従来手法および提案手法を学習させる。MNISTは学習データ数60000、テストデータ数10000の手書き数字データセットである。784次元の入力で0~9の10つの数字を分類する。また、事前定義パラメータはage_{max} = 100、 λは200および1000の場合で学習を行った。

認識率の測定方法は以下のとおりである。まずモデルの学習完了後、全ての学習データに最も近いノードを選び記録する。あるノードに最も近い学習データ集合において、もっとも多いクラスをそのノードのクラスと定義する。図7のように、学習データの4のクラスに似ている

と選ばれた回数が最も多いノードのクラスは4とする。

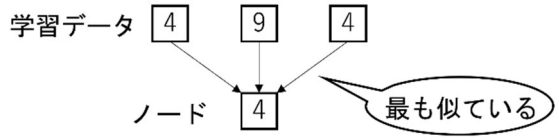

図 7. ノードのラベル付け方法

次にテストデータに最も近いノードを選ぶ。テストデータのクラスと選ばれたクラスが一致した割合が識別率となる。教師あり学習では各ノードに属性値が付くため、その属性値が最も大きいクラスをそのノードのクラスとして、属性値を用いた精度も測った。こちらは推論時に全学習データとノード間との距離を測る必要がないため高速であるという利点がある。

5. 結果および考察

SOINNは都度結果の変わるアルゴリズムであるため、1エポックを10回学習し、その時のノード数平均、精度平均を記録し従来手法と比較し、表1、表2にまとめた。SSは教師あり学習を取り入れたSOINN、1、2はそれぞれ提案手法1と提案手法2を取り入れたかどうかを表している。属性値精度とは、教師あり学習で取り入れた属性値のみを用いて未知データを予測したときの精度である。

表 1. 実験結果 $\lambda = 200$

モデル	ノード数平均	精度	属性値精度
SOINN	216.1	0.804	
SS	214.7	0.814	0.811
SS+1	142.9	0.820	0.819
SS+2	346.4	0.894	0.893
SS+1,2	194.3	0.889	0.888

表 2. 実験結果 $\lambda = 1000$

モデル	ノード数平均	精度	属性値精度
SOINN	807.8	0.894	
SS	787.3	0.898	0.897
SS+1	529.5	0.898	0.898
SS+2	984.1	0.925	0.924
SS+1,2	459.5	0.916	0.916

教師あり学習にした場合、SOINNと比べてノード数はほぼ変わらないが精度が上がる傾向がある。また、提案手法1を取り入れると教師あり学習と比べて精度を変えずにノード数が削減できた。これは閾値領域が従来手法と比べて広がったため、より未知であると判断されたデータのみネットワークの形成のために追加されていると考えられる。提案手法2を取り入れるとノード数が増えるが精度は上がることが読み取れる。エッジが構築されるため、ノードが削除されにくくなりノードが増える。しかし、同じクラス同士にエッジが付きやすくなるため共通した特徴が残りやすい。これが精度を上げる要因になっていると考えられる。それぞれの手法を組み合わせた場合だと、お互いの短所を打ち消しあい、ノード数を削減しつつ精度を上げていることが確認できる。従来手法のSOINNと比べて精度を$\boldsymbol{\lambda}$ =200の場合では8.5ポイント、$\boldsymbol{\lambda}$ =1000の場合では2.2ポイント向上させることができた。

属性値を用いた予測は全データを用いて予測した場合と比べて精度が若干落ちてしまう。しかし、計算量の大幅な削減と学習データを用いずに予測することができるため、オンライン学習に適した推論方法だと考える。

6. まとめ

本研究では、SOINNの高次元空間における認識精度の向上を目的とし、重みベクトルと属性値情報双方の学習を可能にした属性値SOINNを提案した。実験では手書き文字データMNISTを使用し、認識率で評価を行った。

従来手法のSOINNに比べて教師ありSOINNでは属性値情報$\boldsymbol{L}$によって平均ノード数を増やさず精度を上げることができた。また、属性値を用いて推論することでいつでも高速に推論することが可能となった。高次元耐性の提案はそれぞれメリットがあるものだが、組み合わせることでノード数平均を小さくしつつ精度を向上させることができた。そのため高次元データに対して耐性のあるSOINNを構築できたといえるだろう。ところでSOINNにはノイズに対してロバストであるという特徴がある。データセットにノイズが付与されていた場合でも本提案が有効に働くか実験を試みたいと考える。

参考文献

[1] F.Shen, 長谷川修, ”An incremental network for on-line unsupervised classification and topology learning”, Neural Networks, vol. 19, No. 1, 2005, 90-106
[2] Sudo Akihito; Sato Akihiro; Hasegawa Osamu, ”Associative Memory for Online Learning in Noisy Environments Using Self-organizing Incremental Neural Network”, IEEE Transactions on Neural Networks, Vol.20, No.6, pp.964-972, (2009)
[3] T.Kohonen, “The Self-Organizing Map”, Proceedings of the IEEE 78, 1990, pp.1464-1479
[4] 佐藤哲哉、山内ゆかり、” 属性値考慮学習を行う自己組織化マップの提案”、 信学技報, vol. 119, no. 453, NC2019-96, pp. 119-124, 2020年
[5] Y. LeCun, C. Cortes and C. J. C. Burges, “The MNIST Database of Handwritten Digits”, http://yann.lecun.com/exdb/mnist/, 1998.