F-038

自己組織化ニューラル木立における AdaBoost の有用性に関する研究

Study on effect of AdaBoost in the Self-Organization Neural Grove

杉山 享志朗† 井上 浩孝‡
Kyoshiro Sugiyama Hirotaka Inoue

1．緒言

近年，パターン認識の技術は文字認識や画像処理などの様々な分野で活用されている。またその高度化に対する要求や期待も高まりつつあり，パターン認識の研究は活発になってきている。

パターン認識の代表的な手法の一つにニューラルネットワークが挙げられる[1)2)]。その歴史は古く，それを用いたパターン認識の方法も多種多様であるが，本研究では T.Kohonen[3)]によって提案された自己組織化マップ(SOM)を応用し，Wen らによって提案された木構造の自己生成ニューラルネットワーク(Self-Generating Neural Networks ： SGNN)[4)]を複数用いてアンサンブル学習によって認識精度を高めた自己組織化ニューラル木立(Self-Organizing Neural Grove ： SONG)を扱う。先行研究[5)]では，SGNN の並列化に際し，教師データの抽出に Bagging 法を用いているが，本研究ではブースト法として一般に知られている AdaBoost[6)]を用い，SONG の更なる性能向上を追求した。

2．理論

SGNN は与えられた訓練データセットにより，自動的に自己生成ニューラル木(Self-Generating Neural Tree ： SGNT)を構築することで訓練データの特徴を木構造内に写像し，高速な識別特性を持つ識別器である。
従来のニューラルネットワークを用いた識別機の多くは，訓練データセットを何万回と提示しなければならないため計算時間がかかることや，入力層，中間層，出力層のユニット数やそれぞれの重みなどのパラメータを各研究者がその経験と勘を元に設定しなければならないなどの欠点があった。しかしこの SGNN は，その各パラメータを自動的に木構造で決定し，さらに訓練データの提示も 1 回で済むので，計算時間が早く，高速な識別を可能にした。しかし，SGNN は入力信号の性質のみに基づく教師なし学習則を用いているため，識別結果には，望ましい出力が外部から与えられる教師あり学習則を用いた他の識別器と比較すると多くの汎化誤差が含まれるという欠点がある。そこで，仏圜・井上らはアンサンブル学習を用いて SGNN を複数生成することで汎化誤差を改善した SONG を提案し，その有効性を示した[5)]。

この先行研究では，各 SGNT に与えられる教師データの選定に Bagging 法を用いている。これは，教師データ群の中から復元抽出を繰り返すというもので，全ての木が並列の関係にあり，分散処理に適した手法である。本研究で用いたのは AdaBoost と呼ばれるブースト法の一種である。以下にそのアルゴリズムを簡略に示す。

1. 教師データそれぞれに重みというパラメータを用意し，$1/N$で初期化する。
2. 重みが大きいデータほど出現確率が高くなるような方法でデータを抽出する。
3. 抽出したデータで木を生成し，教師データ全てをその木でテストする。正解率が高いほど，その木の信頼度を大きく設定する。
4. 信頼度に応じて，木が正解したデータは重みを小さく，間違ったデータは重みを大きくする。
5. 2-4 を繰り返し，望みの本数の木を生成する。
6. 最終的な出力は，信頼度の高い木ほど影響力を持つよう設定された投票で決定する。

AdaBoost 法は基本的には 2 値分類のためのアルゴリズムであるので，次に示す実験では 2 値分類問題のみを扱う。

3. 実験方法

今回の実験で用いた教師データは，UCI 機械学習リポジトリ[7)]の breast-cancer-wisconsin，ionosphere，liver-disorders，pima-diabetes の 4 つの 2 値分類問題である。各データにおいて，以下の手順で実験を行った。

まずデータの順番をランダムに入れ替え，データ数の 1 割に当たる数のデータをテスト入力用データとして隔離する。残りのデータの中から，先ほどと同じ数のデータを，Bagging 法を用いて訓練に使用する教師データとして抽出し，木を生成し，テスト入力データを識別させ，正解率等を記録した。

以上の手順を，木の数を 1 から 25 まで変化させながら 100 回ずつ実行し，各記録の平均を取った。AdaBoost 法についても同様の実験を行った。

実験に使用したコンピュータのスペックは以下のとおりである。

・CPU ： Intel Core i7 920 @2.67GHz
・RAM ： 3.00GB RAM
・OS ： Microsoft Windows7 Ultimate 64bit

4. 結果と考察

表 1 に Bagging と Adaboost それぞれの識別率を，表 2 にそれぞれが計算に要した時間を，表 3 にそれぞれが計算に要したユニット数(木 1 本当り)を示す。表中の SGNT は木の数を 1 とした時，SONG は木の数を 25 とした時の結果である。また，同条件で抽出方法を変えた場合に優れている方を太字にしてある。

† 呉工業高等専門学校専攻科 機械電気工学専攻
‡ 呉工業高等専門学校電気情報工学科

表 1 識別率

Accuracy	SGNT		SONG	
Dataset	Bagging	AdaBoost	Bagging	Adaboost
breast-cancer-w	**0.949**	0.940	**0.974**	0.959
ionosphere	**0.847**	0.835	**0.893**	0.790
liver-disorders	**0.574**	0.572	**0.632**	0.581
pima-diabetes	**0.693**	0.685	**0.753**	0.725
Average	**0.766**	0.758	**0.813**	0.764

表 2 計算時間(s)

Time(sec)	SGNT		SONG	
Dataset	Bagging	AdaBoost	Bagging	Adaboost
breast-cancer-w	**0.172**	0.391	**4.334**	23.217
ionosphere	**0.284**	0.621	**6.971**	7.917
liver-disorders	**0.072**	0.162	**1.920**	2.776
pima-diabetes	**0.252**	0.568	**6.147**	9.241
Average	**0.195**	0.436	**4.843**	10.788

表 3 ユニット数(units)

Memory(Nodes)	SGNT		SONG	
Dataset	Bagging	AdaBoost	Bagging	Adaboost
breast-cancer-w	38.905	**37.957**	37.283	**7.846**
ionosphere	**93.151**	95.558	63.592	**23.132**
liver-disorders	182.252	**174.983**	170.360	**60.160**
pima-diabetes	**240.224**	243.296	245.869	**84.357**
Average	138.633	**137.949**	129.276	**43.874**

表 1 から，識別率では Bagging が有利なことがわかる。AdaBoost で識別率が向上しなかった理由は，間違えやすいデータが多く出現することで木の多様性が Bagging より劣り，汎化能力が減少した結果だと考えられる。

表 2 から，計算時間では Bagging が有利なことがわかる。Adaboost では，木を生成する際に，前の木に教師データを全て識別させ，間違い率や信頼度などを測る必要があり，計算時間の増加に起因したと考えられる。

表 3 から，ユニット数の面では AdaBoost が有利な傾向があることがわかる。SGNT では差は殆ど見られないが，SONG になるとブースティングが進み，枝刈りの際に多くのユニットが削除されていると推測できる。

5. 結言

本研究では，SONG における AdaBoost の有用性を検討するため，比較対象として Bagging を選び，4 つの問題に対して実験を行った。結果として，AdaBoost 法は使用メモリ削減という面では優れた結果が得られたが，識別精度・識別速度の面では Bagging 法に及ばなかった。また，Bagging 法が並列分散処理に適した手法であるのに対し，AdaBoost 法は言わば直列な処理を行うため，並列分散処理に向いているとは言いがたい。これらの理由から総合的に判断すると，Bagging 法を用いた SONG がより優れているという結論に達した。

参考文献

1) 甘利俊一, 麻生英樹, 津田宏治, 村田昇, ``パターン認識と学習の統計学," 第 I 部, 岩波書店, 2003
2) C.M.Bishop, ``Neural Networks for Pattern Recognition," Oxford University Press, 1995
3) T.コホネン(著), 徳高平蔵, 大藪又茂, 堀尾恵一, 藤村喜久朗, 大北正昭(監修), ``自己組織化マップ改訂版," シュプリンガー・ジャパン, 2005
4) W.X.Wen, A.Jennings, H.Lin, ``Learning a neural tree," In proc. of the International Joint Conference on Neural Networks, Beijing China, Nov.3-6, vol.2, pp.751-756.1992
5) 佛圓和之, 井上浩孝, ``自己生成ニューラルネットワークを用いたアンサンブル学習に関する研究," 平成 19 年度電気・情報関連学会中国支部連合大会講演論文集, pp.152-153, 2007
6) 村田昇, ``アンサンブル法,"電子情報通信学会『知識ベース』S3 群 4 編 1 章, pp.44-47
7) A.Frunk, A.Asuncion, ``UCI machine learning repository," 2010, http://archive.ics.uci.edu/ml