

決定木と集団学習

研究ノート

1. 決定木

樹木モデルによる分類木(決定木)は、計算の速さ、結果の読みやすさ、説明のしやすさなどから多くの分野で応用されている。

決定木は多くのアルゴリズムが提案されている。WEKA には、10 種類の樹木に関するアルゴリズムが実装されている。しかし、その中ではデータ形式に特化したものもある。データのへの制約が少ないのは J48、NBTree、RandomForest、RandomTree、REPTree などである。

先月号で、すでに説明したとおり、J48 は C4.5 の WEKA バージョンである。

NBTree は、ナイーブベイズの分類器(naive Bayes classifiers)のアプローチで決定木を生成するアルゴリズムで、Ron Kohavi によって提案された(参考文献[4])。

RandomForest は、ブートストラップというリサンプリングの方法でサブデータを作成し、それぞれのサブデータセットの決定木を組み合わせる方法で、Breiman により提案されている[2]。WEKA では tree のカテゴリーに分類しているが、C4.5 や CART のような樹木モデルとは性質が異なる。RandomForest は、後述の集団学習アルゴリズムの一種であると考えるのが適切であろう。

RandomTree は、説明変数をランダムに、分岐に用いるアルゴリズムである。よって、分岐に用いる変数は重複して用いられ、C4.5 や CART などより茂る木を生成するのが特徴である。

REPTree は、ゲインと分散(gain/variance)の情報を用いて決定・回帰木を構築するアルゴリズムである。C4.5 との大きい違いは木の剪定方法で、精度より計算の速さが取り柄である。

多くの決定木の中で、どの決定木の精度が良いかに関しては、幾つかの実証方法があるが、最も理解しやすいのは識別・判別率(あるいは誤り率)である。

本稿では、データマイニングのベンチマークとして使用されている UCI データを用いて、識別の誤り率で精度を評価することにする。

UCI は、カリフォルニア大学アーバイン校(University of California, Irvine)の知識発見の研究者らが蓄積・公開しているデータアーカイブである。WEKA 用の UCI データは次のサイトからダウンロードすることができる。

<http://prdownloads.sourceforge.net/weka/datasets-UCI.jar>

WEKA 用の UCI には 37 セットのデータがある。その中にはデータのサイズが大きいものもあり、分類器によっては、メモリが 500MB 以下では計算ができないものもあるので、ここで用いるデータは表 1 に示すものに限定した。

また、分類器の精度の評価は、多重交差確認法(n=10)の誤り率を用いることにする。

表 1 に J48、NBTree、RandomForest、RandomTree、REPTree の誤り率と比率を示す。

誤り率の平均値が小さいからと言って、その分類器の精度が良いとは限らない。と言うのは、異なるデータセットにおける、誤りの率の差が大きいためである。分類器の精度の比較には、ある分類器を基準とした比率もひとつの指標となる。表1の比率は、J48を基準としている。この比率の値が1より小さいと、誤り率がJ48より低く、その分類器の精度がJ48より良いと評価できる。

表1から分かるように、平均の誤り率が最も低いのは、RandomForest、NBTree、J48の順であり、J48を基準とした比率では、RandomForest、J48、NBTreeの順である。

よって、精度が最も良いのは RandomForest で、J48 と NBTree は大きい差が見られない。参考のため、各データセットの中で誤り率が少ない順のランキングをまとめたものを表2に示す。例えば、表2の2行目2列目の10はJ48の誤り率が最も低いケースが10回であることである。表2からも分かるように、RandomForestの誤り率が最も低いケースが15で、最も多い。

2. 集団学習

決定木は、高精度の分類器ではないが、計算の速さやその結果の読みやすさに長所を持っている。

日本語では「三人寄れば文殊の知恵」、中国語では「三個臭皮匠、賽過一個諸葛亮」、英語では“Two heads are better than one.”という熟語がある。これらは、凡人でも多数集まって考えをまとめれば、何とか良い知恵が浮かぶということの意味で用いられている。

このような人間の日常の知恵がデータ解析のアルゴリズムとして提案されている集団学習という方法がある。

集団学習(ensemble learning)は、決して精度が高くない複数の結果を統合・組み合わせることで精度を向上させるために提案された機会学習方法である。複数の結果の統合・組み合わせの方法としては、分類の問題では多数決、数値の予測の問題では平均が多く用いられている。

集団学習では、異なる重み、あるいは異なるサンプルから単純なモデルを複数作成し、これらを何らかの方法で組み合わせることで、精度と汎化力を両立するモデルを構築する。決定木と関わる最も知られている集団学習アルゴリズムはバグギング(bagging)、ブースティング(boosting)、ランダム森(random forest)である。

(1) バグギング

バグギング(bagging)の bagging は、bootstrap aggregating の頭の文字列を組み合わせた造語である。バグギングは、与えられたデータセットから、ブートストラップ法による複数の学習データセットを作成し、そのデータを用いて作成した分類器を統合・組み合わせる。ブートストラップサンプルはそれぞれ独立で、学習は並列に行うことができる(参考文献[1])。

ブートストラップサンプルは、与えられたデータの経験分布とその推定量に基づいたリサンプリングにより得られたサンプルである。

(2) ブースティング

ブースティング(boosting)は、与えた学習データを用いて学習を行い、その学習結果を踏まえて逐次に重みの調整を繰り返すことで複数の学習結果を求め、その結果を統合・組み合わせ、精度を向上させる方法である(参考文献[3])。ブースティングの中で最も広く知られているのはAdaBoost というアルゴリズムである。

(3) ランダム森

ランダム森(random forest: RF)は、Bagging の提案者 Breiman により、今世紀の初頭に提案された新しいアルゴリズムである(参考文献[2])。

RF のアルゴリズムを次に示す。

1. 与えられたデータセットから N 組のブートストラップサンプルを作成する。
2. 各々のブートストラップサンプルデータを用いて未剪定の最大の決定・回帰木を作成する。ただし、分岐のノードはランダムサンプリングされた変数の中の最善のものを用いる。
3. 全ての結果を統合・組み合わせ(回帰の問題では平均、分類の問題では多数決)、新しい予測・分類器を構築する。

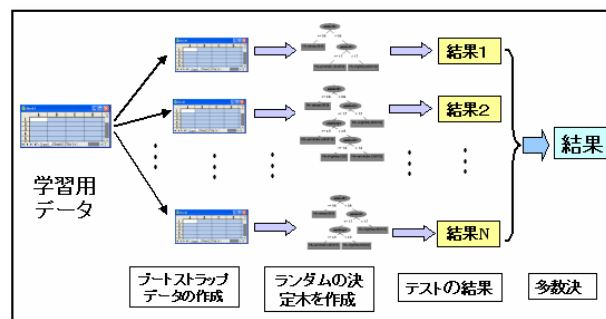


図 1 RF のアルゴリズムイメージ

Bagging と RF の大きい違いは、Bagging は全ての変数を用いるが、RF では変数をランダムサンプリングしたサブセットを用いることができるので、高次元データの計算が容易である。

ランダムサンプリングする変数の数 M はユーザが自由に設定することができる。Breiman は、 M は変数の数の正の平方根をとることを勧めている。

RF は、多くの工夫が施され、次のような長所を持っていると言われている。

主な長所：

- ◇ 精度が高い。
- ◇ 大きいデータに効率的に作動する。何百、何千の変数でもよい。

- ◇ 分類に用いる変数の重要度を推定する。
- ◇ 欠損値の推測および多くの欠損値を持つデータの正確さが維持するのに有効である。
- ◇ 分類問題における各群の個体数がアンバランスであるデータにおいてもエラーのバランスが保たれる。
- ◇ 学習データから生成された RF は保存して、新たなデータに適応することができる。
- ◇ 分類と変数の関係に関する情報を計算する。
- ◇ 群間の近似の程度が計算できる。
- ◇ 群の情報がないデータにも適応できる。

3. 集団学習の精度比較

本項では、決定木に集団学習方法を組み合わせた精度について比較を行うことにする。ここでも前項と同じく UCI のデータセットを用いることにする。ただし、表 1 に用いたデータの中で、メモリ不足で計算不能なものは除いた。

WEKA では、Bagging、Boosting は Classify タブの中の meta フォルダに置かれている。WEKA に実装されている Boosting は AdaBoostM1 である。Bagging、Boosting は次の手順で取り込む。

- ◇ Classify パネルの [Choose] ボタンを押す。
- ◇ meta フォルダを開く。
- ◇ Bagging、あるいは AdaBoostM1 を右クリックする。

図 2 に meta フォルダを開いた画面、図 3 に AdaBoostM1 を取り込んだ画面を示す。

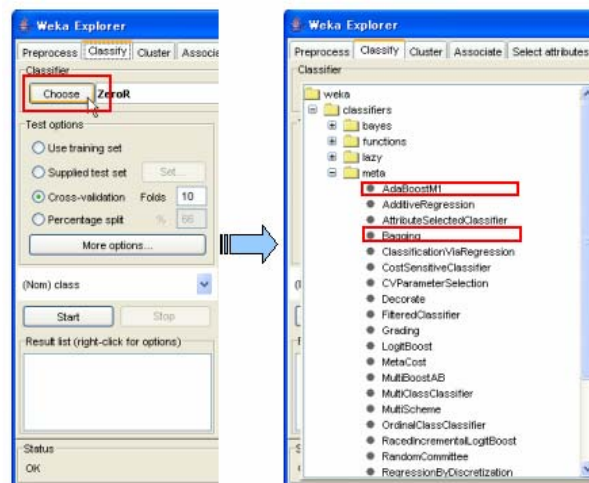


図 2 meta のリスト

WEKA における AdaBoostM1、Bagging は決定木のような分類器を組み合わせて用いるように実装されている。RF は、集団学習の方法であるが、AdaBoostM1 及び Bagging とは異なり、RF 自体がひとつのデータ解析器として tree フォルダに置かれている。

組み合わせるアルゴリズムの選択は、まず AdaBoostM1 (あるいは Bagging) が取り込まれてい

るパネルにおける AdaBoostM1 の文字列の部分をクリックし、weka.classifiers.meta.AdaBoostM1 パネルを開き、そのパネルの中の[Choose]ボタンを押し、組み合わせるアルゴリズムを取り込む。

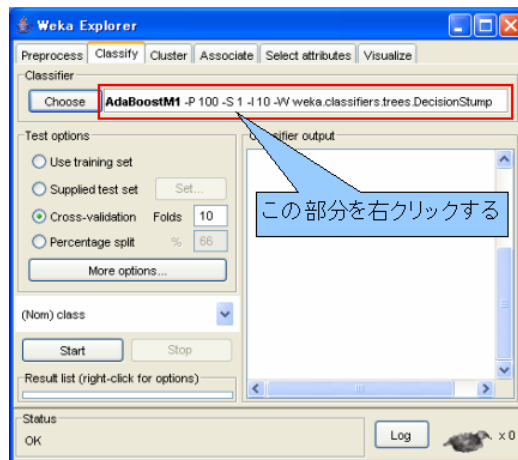


図3 AdaBoostM1 が取り込まれている画面

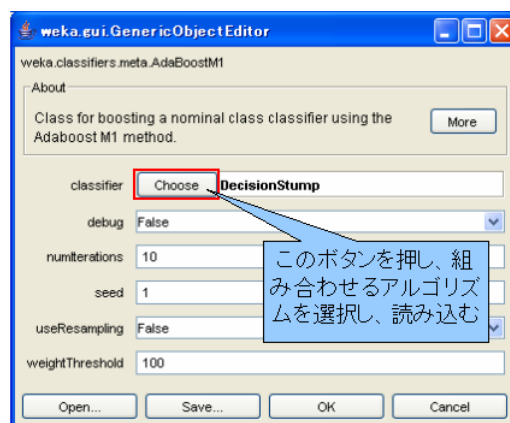


図4 weka.classifiers.meta.Bagging 画面

このような手順で集団学習アルゴリズム AdaBoostM1、Bagging と樹木アルゴリズム J48、NBTree、RandomForest を組み合わせる解析を行った誤り率を表3に示す。RFも集団学習の方法であるが、ここではひとつの試みとして、さらに集団学習アルゴリズム AdaBoostM1、Bagging を組み合わせる解析も行った。

今回の比較分析では、次のことが分かった。

- ◇ 用いた分類器は、Bagging、Boosting との組み合わせにより誤り率は減少し、精度が向上している。
- ◇ Bagging と NBTree の組み合わせによる誤り率が最も低い。ただし、この組み合わせはメモリを多く占有し、計算にも時間が掛かる。
- ◇ RF と Bagging、Boosting と組み合わせると誤り率がさらに減少する。特に Bagging との組み合わせによる誤り率の減少が顕著である。ただし、非常にまれであるが、誤り率が若干増加するケースもある。

表1 データセットにおける誤り率(%)
(比率は、J48の誤り率を基準とした誤り率)

データ名	個体数	変数	群数	J48	NBTree		RandomTree		RandomForest		REPTree	
				誤り率	誤り率	比率	誤り率	比率	誤り率	比率	誤り率	比率
anneal	898	38	6	1.56	1.78	1.14	3.56	2.28	0.67	0.43	2.01	1.29
anneal-ORGI	898	38	6	9.02	3.34	0.37	7.02	0.78	5.23	0.58	9.24	1.02
audiology	226	69	24	22.12	21.68	0.98	40.71	1.84	22.57	1.02	27.88	1.26
auto	205	25	6	18.05	20.49	1.14	27.32	1.51	16.59	0.92	37.56	2.08
balance-scale	625	4	3	23.37	22.88	0.98	22.24	0.95	19.52	0.84	23.36	1.00
breast-cancer	286	9	2	24.48	29.02	1.19	28.32	1.16	30.77	1.26	29.02	1.19
breast-w	699	9	2	5.44	3.43	0.63	5.58	1.03	3.86	0.71	6.15	1.13
colic-ORIGI	368	22	2	33.70	35.33	1.05	29.62	0.88	31.52	0.94	32.34	0.96
colic	368	27	2	14.67	16.58	1.13	29.35	2.00	13.86	0.94	15.49	1.06
credit-a	690	15	2	13.91	14.49	1.04	25.07	1.80	14.93	1.07	14.20	1.02
credit-g	1000	20	2	29.50	26.20	0.89	32.60	1.11	27.20	0.92	27.40	0.93
diabetes	768	8	2	26.17	25.52	0.98	33.07	1.26	26.04	1.00	24.87	0.95
glass	214	9	6	33.18	29.44	0.89	42.06	1.27	28.04	0.85	33.65	1.01
heart-c	303	13	2	22.44	19.08	0.85	27.39	1.22	18.81	0.84	23.43	1.04
heart-h	294	13	2	19.05	19.73	1.04	21.77	1.14	21.77	1.14	23.13	1.21
heart-statlog	270	13	2	23.33	21.11	0.90	26.30	1.13	21.85	0.94	23.33	1.00
hepatitis	155	19	2	16.13	19.36	1.20	21.94	1.36	17.42	1.08	21.29	1.32
hypothyroid	3772	29	3	0.42	0.53	1.26	5.09	12.12	0.93	2.21	0.42	1.00
ionosphere	351	34	2	8.55	10.26	1.20	15.10	1.77	7.12	0.83	10.26	1.20
iris	150	4	3	4.00	7.33	1.83	9.33	2.33	5.33	1.33	6.00	1.50
kr-vs-kp	3196	36	2	0.56	2.91	5.20	11.55	20.63	1.16	2.07	1.00	1.79
labor	57	16	2	26.32	12.28	0.47	19.30	0.73	12.28	0.47	22.81	0.87
lymphography	148	18	4	22.97	18.92	0.82	29.05	1.26	19.60	0.85	27.70	1.21
primary-tumor	339	20	18	60.18	53.98	0.90	65.49	1.09	56.64	0.94	60.47	1.00
segment	2310	19	7	3.07	5.02	1.64	10.69	3.48	2.51	0.82	4.94	1.61
sick	3772	29	2	1.19	2.33	1.96	4.19	3.52	1.62	1.36	1.30	1.09
sonar	208	60	2	28.85	23.08	0.80	31.73	1.10	19.23	0.67	24.04	0.83
soybean	683	35	19	8.49	8.49	1.00	24.89	2.93	8.49	1.00	15.67	1.85
vehicle	846	18	4	27.54	27.90	1.01	35.70	1.30	22.58	0.82	27.90	1.01
vote	435	16	2	3.68	4.37	1.19	7.36	2.00	4.14	1.13	4.60	1.25
vowel	990	13	11	18.49	6.47	0.35	26.77	1.45	4.04	0.22	32.32	1.75
waveform	5000	40	3	24.92	20.34	0.82	37.46	1.50	18.20	0.73	23.12	0.93
平均				17.98	16.68	1.15	23.68	2.50	15.77	0.97	19.90	1.20

表2 誤り率の小さい順位表

順位	J48	NBTree	RandomTree	RandomForest	REPTree
1	10	9	1	15	2
2	7	8	3	9	5
3	7	7	4	6	12
4	7	7	5	2	10
5	1	1	19	0	3
平均順位	2.44	2.47	4.19	1.84	3.29

表 3 集団学習の結果
(J48&比率は、J48 を基準とした比率の平均)

データ名	誤り率(%)								
	J48	Bagged J48	Boosted J48	NBTree	Bagged NBTree	Boosted NBTree	RForest	Bagged RForest	Boosted Rforest
anneal	1.56	1.11	0.45	1.78	0.78	0.56	0.67	0.45	0.66
anneal-ORGI	9.02	6.46	4.23	3.34	2.79	1.00	5.23	4.90	5.46
auto	18.05	15.12	13.66	20.49	19.02	15.61	16.59	16.59	15.12
balance-scale	23.37	17.76	21.12	22.88	17.76	18.88	19.52	17.60	21.92
breast-cancer	24.48	26.57	30.42	29.02	26.57	31.21	30.77	29.72	33.22
breast-w	5.44	4.14	4.29	3.43	2.72	3.58	3.86	3.43	3.86
colic	14.67	14.40	16.58	16.58	15.76	16.58	13.86	13.86	17.12
colic-ORIGI	33.70	33.70	33.70	35.33	27.72	26.90	31.52	30.16	29.89
credit-a	13.91	14.64	15.80	14.49	13.91	13.04	14.93	13.33	14.64
credit-g	29.50	26.00	30.40	26.20	24.90	27.50	27.20	23.10	26.80
diabetes	26.17	25.91	27.60	25.52	24.22	25.91	26.04	23.18	25.65
glass	33.18	28.97	25.70	29.44	22.90	27.10	28.04	22.43	26.65
heart-c	22.44	20.79	17.82	19.08	15.51	20.13	18.81	16.17	18.48
heart-h	19.05	21.09	21.43	19.73	16.67	20.07	21.77	20.07	23.13
heart-statlog	23.33	20.00	19.63	21.11	17.78	21.85	21.85	18.15	21.85
hepatitis	16.13	16.77	14.19	19.36	16.13	16.13	17.42	16.77	16.77
hypothyroid	0.42	0.42	0.42	0.53	0.27	0.37	0.93	0.74	0.74
ionosphere	8.55	6.84	6.84	10.26	7.41	7.40	7.12	7.12	7.41
iris	4.00	4.67	6.67	7.33	6.67	6.00	5.33	5.33	5.33
kr-vs-kp	0.56	0.56	0.50	2.91	0.69	0.60	1.16	0.85	1.16
labor	26.32	15.79	10.53	12.28	10.53	10.53	12.28	12.28	12.28
lymphography	22.97	20.95	18.92	18.92	12.84	13.51	19.60	15.54	18.92
primary-tumor	60.18	57.82	59.88	53.98	53.98	53.98	56.64	56.05	56.93
sick	1.19	1.28	0.82	2.33	1.64	1.17	1.62	1.67	1.56
sonar	28.85	25.48	22.12	23.08	17.31	20.19	19.23	13.46	17.79
vehicle	27.54	23.40	23.76	27.90	26.12	23.64	22.58	23.52	22.93
vote	3.68	3.68	4.14	4.37	4.37	4.14	4.14	3.68	4.37
vowel	18.49	9.60	6.67	6.47	4.24	6.47	4.04	1.62	4.04
waveform	24.92	18.70	19.52	20.34	15.60	18.88	18.20	15.00	15.92
平均	18.68	16.64	16.48	17.19	14.72	15.62	16.24	14.72	16.23
J48&比率	1.00	0.90	0.87	1.15	0.83	0.85	0.97	0.86	0.96

参考文献

- [1] Breiman, L. (1996). Bagging Predictors, Machine Learning, 24, 123-140.
- [2] Breiman, L. (2001). Random Forests, Machine Learning, 45, 5-23.
- [3] Freund, Y. and Schapier, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119-139.
- [4] Ron Kohavi (1996). Scaling up the accuracy of naive-Bayes classifiers: a decision tree hybrid. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (<http://citeseer.ist.psu.edu/kohavi96scaling.html>)