An Empirical Comparison of Ensemble Methods Based on Classification Trees

Mounir Hamza and Denis Larocque

Department of Quantitative Methods

HEC Montreal
Canada

Outline

- Introduction
- Classification Trees
- Ensemble Methods
- Simulation
- Results
- Conclusion

Introduction

Supervised classification

Y : target variable
(categorical for this talk)

$X = (X_1, \ldots, X_p)$: vector of predictors

Goal: Based on a sample of (X, Y), develop a model to predict future Y values given X.

Examples:

- Discriminant analysis
- Logistic regression
- Neural networks
- Classification trees
- Support vector machines.

In this talk:

- Empirical comparison of several ensemble methods based on classification trees
- Comparison with the results of Lim, Loh and Shih (*Machine Learning*, 2000)

Classification trees

Idea: create a partition of the predictor space into a set of rectangles and assign a class to each of them.

CART, Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984), work with the notion of nodes impurity.

If Y is binary (0,1), Gini's impurity at a given node N is:

$$i(N) = P_N(0)P_N(1)$$

where $\mathrm{P_N(i)}$ = proportion of observations of node N such that $Y = i$.

This function attains its minimum 0 when the node is pure (i.e. contains only 0's or only 1's). It attains its maximum ¼ when there are as many 0's as 1's in the node.

At a given node N, the algorithm seeks the best split. A split is a separation of the observations of the node into two parts with respect to a rule defined by one of the predictor variable.

If X if continuous or ordinal: $X < c$

If X is nominal: $X \in \{c_1, \ldots, c_k\}$

The algorithm seeks the best split among all possible splits. For a given split, we can calculate the decrease in impurity

$$\Delta i(N) = i(N) - P_L\, i(N_L) - P_R\, i(N_R)$$

where N_L et N_R are the left and right nodes created after the split and where P_L et P_R are the proportions of observations of node N that end up in the left and right nodes after the split. The best split is the one that maximizes the decrease in impurity.

In the end, we get a tree $G(x)$ which, for a given x, returns an estimate of $P(Y=1|x)$. This estimate is simply the proportion of 1 in the terminal node that is reached when we let x fall into the tree. We can use a cut-point to classify an observation, for example, we could assign class 1 if $G(x) > ½$ and 0 otherwise.

Advantages of classification trees:

- Easy to understand (if there are not too many terminal nodes)
- Can handle missing data
- Nonparametric

Disadvantages:

- Not always the best in terms of performance
- Unstable

The trees are considered unstable in the sense that, if we slightly modify the data, the tree constructed with the modified data can be very different from the one constructed with the original observations.

Ensemble methods

"Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a (weighted) vote of their predictions"

Dietterich (2000).

Examples:

- Bagging
- Boosting
- Random Forests
- Randomization

Bagging

Bagging: bootstrap aggregating. Breiman (1996).

Idea: generate bootstrap samples, construct a tree with each of them and combine the results.

Typical algorithm:

For $b = 1, \ldots, B$,

1) Choose n observations at random and with replacement form the original data.
2) Construct a tree with these observations.

To classify an observation, i) use the majority class among the B trees or, since a tree returns an estimate of $P(Y = k \mid X)$, ii) we can average those estimates over the B trees and assign the class with the highest estimated probability.

Boosting (arcing)

Freund and Schapire (1996): AdaBoost.M1

Idea: construct a tree with the original data, give more weights to the data points that were misclassified, construct a new tree with the weighted points. Modify the weights again and so on… In the end, combine the results.

Breiman (1998): arcing (Adaptively Resample and Combine)

Algorithm arc-x4: Breiman (1998).

Start with the weights

$$p_i^{(0)} = 1/n \ ; \ i = 1,\ldots,n.$$

For $k = 1,\ldots,M$

1) By sampling from the original data with replacement and with weights $(p_1^{(k)},\ldots, p_n^{(k)})$, obtain a sample of size n.

2) Construct a tree, G_k, with this sample.

3) Let m_i be the number of times that the ith observation of the original data is misclassified by $G_1,\ldots,G_k$.

4) Update the weights:

$$p_i^{(k+1)} = \frac{(1+m_i^4)}{\sum(1+m_i^4)}$$

To classify an observation, use the majority class among the M trees.

Random forests

Breiman (2001).

A random forest is a collection of trees $\{G(x,\Theta_k), k=1,...\}$ where $\Theta_1, \Theta_2, \ldots$ are training sets that are independent and identically distributed.

Note: Bagging is an example of a random forest.

Important result: we can't over-fit when the number of tree increases.

Two algorithms are used in Breiman (2001). Here is the description of one of them. It combines Bagging with a particular way of constructing trees.

The basic algorithm is (as for Bagging):

For $b = 1, \ldots, B$,

1) Choose n observations at random and with replacement form the original data.
2) Construct a tree with these observations.

The difference here is that we inject randomness into the tree construction algorithm itself. At a given node, we will select at random a subset of predictors (of size F) and let the algorithm find the best split among the selected predictors. The selected predictors can be different at each node.

Simulation

Data sets:

Wisconsin Breast Cancer (BCW)
BUPA Liver Disorders (BLD)
Boston Housing (BOS)
Contraceptive Method Choice (CMC)
Statlog DNA (DNA)
StatLog Heart Disease (HEA)
LED Display (LED)
PIMA Indian Diabetes (PID)
Image Segmentation (SEG)
Attitude toward Smoking Restrictions (SMO)
Thyroid Disease (THY)
StatLog Vehicle Silhouette (VEH)
Congressional Voting Records (VOT)
Waveform (WAV)

All of the data sets can be obtained from UCI Machine learning repository (http://www.ics.uci.edu/~mlearn/MLSummary.html), with the exception of SMO which can be obtained from http://lib.stat.cmu.edu/datasets/csb/.

Table 1: Characteristics of the data sets

Data set	Size	Number of values of the dependent variable	Number of predictors					
			Numerical	Number of values of the categorical predictors				Total number of predictors
				2	3	4	5	
BCW	683	2	9					9
BLD	345	2	6					6
BOS	506	3	12	1				13
CMC	1473	3	2	3		4		9
DNA	3186	3	0			60		60
HEA	270	2	7	3	2	1		13
LED	6000	10	0	7				7
PID	532	2	7					7
SEG	2310	7	19					19
SMO	2855	3	3	3	1		1	8
THY	7200	3	6	15				21
VEH	846	4	18					18
VOT	435	3	0		16			16
WAV	3600	3	21					21

Methods:

- **Single tree** with pruning (CART)
- **Bagging** with 100 trees (CART)
- **Arcing** (arc-x4) with 100 and 250 trees (CART)
- **Random forest** with 100 trees (R). The number of predictors chosen at random at a given node is $\log_2(M+1)$ where M is the total number of predictors.

3 split criteria: Gini, entropy and twoing.

With and without **linear combinations** of predictors.

Comparison criterion: classification error estimated with 10-fold cross-validation.

For another part of the study**, noise was added** to data.

Lim, Loh and Shih (2000): compared several classification methods with the same data sets.

Trees: CART, Splus tree, C4.5, FACT, QUEST, IND, OC1, LMDT, CAL5, T1.

Statistical algorithms: linear and quadratic discriminant analysis, nearest-neighbor, logistic discriminant analysis, Flexible discriminant analysis, penalized discriminant analysis, mixture discriminant analysis, POLYCLASS.

Neural networks: learning vector quantization (Splus), Radial Basis Function.

Results

Table 4: Overall ranking of the methods according to the mean classification error and according to the mean rank

Sorted according to the mean rank		Sorted according to the mean classification error	
Method	**Mean rank**	**Method**	**Mean classification error**
RF	2.38	RF	17.79
BO(100)-NLC-E	3.38	BA-NLC-E	18.61
BO(250)-NLC-E	4.17	BO(250)-NLC-G	18.64
BO(250)-NLC-G	4.56	BO(100)-NLC-E	18.72
BO(100)-NLC-G	4.91	BO(250)-NLC-E	18.76
BA-NLC-E	5.20	BO(100)-NLC-G	18.87
BA-NLC-G	5.67	BA-NLC-G	18.89
ST-NLC-E	7.64	ST-NLC-E	22.71
ST-NLC-G	7.92	ST-NLC-G	23.25

Table 5: Comparison with the results of Lim, Loh and Shih (2000)

Data set	Best classification error obtained in Lim et al. (2000)	Method that obtained the best classification error in Lim et al. (2000)	Best classification error in the present study	Method that obtained the best classification error in the present study	Percent difference between the two
BCW	2.78	Neural networks	2.64	RF	-5.20
BLD	27.9	Classification tree (0C1)	25.80	RF	-7.54
BOS	22.5	Neural networks	19.96	BO(250)-WLC-G	-11.29
CMC	43.4	POLYCLASS	47.39	RF	9.19
DNA	4.72	POLYCLASS	3.23	RF	-31.51
HEA	14.1	Linear discriminant analysis	17.78	RF	26.08
LED	27.1	Linear discriminant analysis	26.27	RF	-3.07
PID	22.1	Linear discriminant analysis	22.93	RF	3.77
SEG	2.21	Nearest Neighbor	1.60	BO(100)-WLC-E	-27.52
SMO	30.5	Linear discriminant analysis	34.05	RF	11.62
THY	0.642	Classification tree (CART)	0.28	BO(100)-WLC-E	-56.73
VEH	14.5	Quadratic discriminant analysis	23.40	BO(250)-WLC-G	61.41
VOT	4.32	Flexible discriminant analysis	3.91	RF	-9.54
WAV	15.1	Neural networks	14.72	RF	-2.50
Mean	16.56		17.42		-3.06
Median	14.8		18.87		-4.14
Minimum	0.642		0.28		
Maximum	43.4		47.39		

Table 6: Percent increase in classification error between the original data and the data with noise

Data set	Classification method				
	ST-NLC-G	BA-NLC-G	BO(100)-NLC-G	BO(250)-NLC-G	RF
BCW	54.84	113.04	104.55	91.30	72.22
BLD	18.18	16.98	20.00	29.17	28.09
BOS	10.85	-1.69	15.53	10.89	1.87
CMC	-6.23	2.28	2.18	1.37	3.30
DNA	2.73	41.91	53.38	38.69	43.69
HEA	40.68	17.31	1.79	15.38	-2.08
LED	-0.68	-0.87	-4.21	-3.99	1.65
PID	19.23	11.81	8.40	6.15	5.74
SEG	120.00	148.00	174.36	128.26	115.56
SMO	8.78	3.07	4.25	5.00	5.56
THY	2156.82	176.00	333.33	292.00	23.08
VEH	-0.76	1.82	2.51	3.03	-5.26
VOT	-9.52	72.73	60.87	68.18	0
WAV	1.32	2.57	5.08	1.97	4.53
Mean	172.59	43.21	55.86	49.10	21.28
Median	9.82	14.40	11.97	13.14	5.04
Minimum	-9.52	-1.69	-4.21	-3.99	-5.26
Maximum	2156.82	176.00	333.33	292.00	115.56

Table 7: Classification error (in %) for random forests with F variables and with the optimal number of variables

	Classification error with F variables	Value of F	Classification error with the optimal number of variables	Optimal number of variables	Percent increase relative to optimal number of variables
BCW	2.64	3	02.34	1	12.50
BLD	25.80	2	25.80	2	0
BOS	21.15	3	19.17	5	10.31
CMC	47.39	2	45.96	3	3.10
DNA	3.23	5	3.11	9	4.04
HEA	17.78	3	15.19	4	17.07
LED	26.27	3	26.02	2	0.96
PID	22.93	3	22.56	1	1.67
SEG	1.95	4	1.69	8	15.38
SMO	34.05	3	30.47	1	11.72
THY	0.36	4	0.25	12	44.44
VEH	26.95	4	24.23	12	11.22
VOT	3.91	4	3.45	5	13.33
WAV	14.72	4	14.72	4	0
Mean	17.79	3.36	16.78	4.93	10.41
Median	19.46	3	17.18	4	10.76
Minimum	0.36	2	0.25	1	0
Maximum	47.39	5	45.96	12	44.44

Conclusion

In this study:

1) Random forests were significantly better than Bagging, Boosting and a single tree.

2) The error rate was smaller than the best one obtained by the methods used in Lim, Loh and Shih (2000) for 9 out of 14 data sets.

3) More robust to noise than the other methods.

Consequently, random forest is a very good « off-the-shelf » classification method:

- Easy to use

- No models, no parameters to select except for the number of predictors to choose at random at each node

- Relatively robust to noise