A Method to Build Classification and Regression Trees

Emilio Unda-Trillas and Jorge Rivera-Rovelo

Universidad Anahuac Mayab
unda91@gmail.com, jorge.rivera@anahuac.mx

Abstract. An adaptable structure to build a classification tree is presented. From such structure different existing classification trees can be obtained, but also we can build new ones, and compare the results of different trees (classification error, tree size, number of levels or other defined criteria). We use the adaptable scheme to emulate ID3, C4.5 and M5 trees, but also create a new tree (called general tree), and results obtained shows that we can obtain the same results with the original trees, and for the case of the general tree, its results are very close to the better classifier tree of the three studied.

1 Introduction

Quinlan [1] seems to be the father of modern classification trees due to his *Iterative Dichotomiser 3* from 1979. Quinlan cites to Hunt Marin and Stone as creators of the CLS (Concept Learning System), which can be considered as the precursor for classification trees. However, in moder classification trees, ID3 is taken as the first method. Such a method can only deal with classification using discrete attributes, but not continuous.

In 1993, Quinlan [3] presented a new tree: the C4.5; which deals with classification problems using not only discrete attributes, but also continuous. Later on, he improves the method and presented C5.0, which is commercially available today.

In [3], the same author proposed M5, which is a method to build a tree which can deal not only with classification issues, but also with regression problems. In [4], Wang presented an improvement for M5, called M5P.

Classification trees have been used successfully for consumer credit behavior, medical diagnosis, effectiveness of mailing campaigns, among others. Some applications have important economic impact; for example, the credit behavior is important for banks because they move several millions (billions) in consumer credit. According to the Federal Reserve, in 2013, USA moves around 147 billion of dollars in consumer credits, and slow payers represents between 2% and 4% (between 3 and 5 billion dollars). So, methods which can help making decisions like to give or not a consumer credit, according to customer characteristics, are very useful.

Our objective was to obtain a method which, as M5P, can deal with classification and regression problems, which can take advantages of each one of the

E. Bayro-Corrochano and E. Hancock (Eds.): CIARP 2014, LNCS 8827, pp. 448–453, 2014.

mentioned classification trees, but not limited to only one stop criteria, or error measurement criteria or heuristic.

2 Method and Results

As it is known, classification trees can be categorized as methods based on supervised learning. Such trees can capture complex functions in a relatively simple form, and the classification function they generates, is graphically represented as a tree and can be easily interpreted by humans. Usually, we train such classifiers in a two steps scheme. First, growing the tree by dividing the samples and assigning them to the child nodes according to certain criteria. Then, trying to find some node that can be pruned to improve the classification effectiveness.

The General Tree is based on M5, which contains a final regressor at the leaf nodes. General Tree is a kind of meta-algorithm which separates the samples in order to do regression. However, the General Tree is easily modified to adapt itself to problem to be solved.

General Tree will take into account next criteria:

- Discrete attributes are divided into a node with so many nodes as values can take the attribute.
- Continuous attributes are binary divided,testing different thresholds.
- Heuristic is optional. If we have an heuristic, we can put it in a way to measure the error. Otherwise, heuristic error will be a root mean square error (RMSE).
- If given any node, there is no way to divide samples to reach a lower heuristic error or RMSE, such a node is marked as a leaf.
- Pruning phase is like in M5 method.

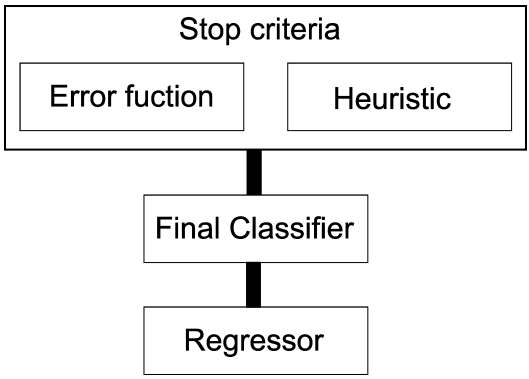

Fig. 1. General scheme of the General Tree generator

A general scheme of the General Tree is shown in Fig. 1. The pseudocode of the General Tree is as follows:

```
General Tree(data):{
  InputData <- divide(data, 70\%)
  PruneData <- divide(data, 30\%)
  root <- new node(InputData)
  root.train()
  root.prune(PruneData)
}

node.train(): {
    if stopCriteria.stop?(node.data)
          finish
    else
          node.classifier <- finalClassifier.newInstance(node.data)
          For each subdivision of node.data:
                Choose the best according to heuristicError or errorFunction
                node.division <- best subdivision
                dataDiv[] <- divide(node.data, node.division)
                For each Di in dataDiv:
                      node.child[i] <- new node(Di)
                      node.child[i].train()
}

node.prune(data): {
    if node.type is leaf
          finish
    else
          For each child in node.child:
                child.prune(divide(data,child))
          if errorFunction(node.classifier) < errorFunction(node.subtree)
                delete(node.child)
                node.type <- leaf
}
```

In General Tree, one can choose between different error measurements in order to evolve the tree. For the heuristic error one can choose between entropy, standard deviation or absolute error; for error function (used when no heuristic is available), one can choose between nominal error, root mean square error, or absolute error. The stop criteria is when the error is less than a threshold. The final classifier can be based on the majority criteria (the simple ZeroR), median criteria, or can have a linear regression at leaf nodes.

As a measure for uncertainty of data with respect to a class, we use entropy (eq. 1), then we compute gain of information (eq. 2), proportion of gain (eq. 3) and a measurement of entropy affected by samples' split (eq. 4).

$$E(X) = -\sum_{X_i}^{c} \frac{|X_i|}{|X|} * \log_n \frac{|X_i|}{|X|} \tag{1}$$

$$G(X) = E(X) - \sum_{X_i}^{n} \frac{|X_i|}{|X|} * E(X_i) \tag{2}$$

$$Gp(X) = \frac{G(X)}{ES(X)} \tag{3}$$

$$ES(X) = -\sum_{k}^{X_i} \frac{|X_i|}{|X|} * \log_2 \frac{|X_i|}{|X|} \tag{4}$$

3 Discussion

The computational complexity of the General Tree for classification can be expressed as in eq. 5, where n is the number of samples, m is the number of attributes, and p is the percentage of continuous attributes. As can be seen in the middle term, the number of continuous attributes plays an important role in complexity. If a regression step is added, the complexity is modified according the complexity of the regressor at the leaves of the tree.

$$O(log(n) * m(1 + pn) * i) \tag{5}$$

For the first batch of experiments we used C4.5 Naive Bayes, MLP and the General Tree implementation of ID3 for three classification problems. We collected three datasets associated with a classification task, one for Acute Bacterial Meningitis, the second related to Cardiac Diseases, and lastly about Muscular Distrophy. Table 1 shows the percentage of correctly classified data for C4.5 (marked as (1)), General Tree emulating ID3 (marked as (2)), Naive Bayes (marked as (3)), and MLP (marked as (4)). Each row of the table corresponds to a diferent dataset: *abm* for Acute Bacterial Meningitis dataset; *acath* for Duke Cardiac Catheterization Coronary Artery Disease Diagnostic Dataset; and *dmd* for Duchenne Muscular Dystrophy Dataset. The blank dot close to the results means a statistically significant improvement, while the black dot means statistically significant degradation. As can be seen, for *abm*, the General Tree is close to the best result, obtained with C4.5, and is slightly better than the other two. However, for the last two datasets the General Tree is not so close. So, for classification tasks, the General Tree is not the best method, although it can reach results comparable to other methods.

The second batch of experiments is for regression problems. For this second batch of experiments we used M5, MLP, and two implementations of General Tree M5PP, and Median Tree, and each one was applied for three diferent regression problems. The three datasets were obtained from the UCI Machine Learning repository [8]. Table 2 shows the root relative squared error obtained for the M5P (marked as (1)), Multilayer Perceptron (marked as (2)), General Tree M5PP (marked as (3)), and General Tree Median Tree (marked as (4)). The datasets used (one in each row), are: Airfoil Self Noise Dataset (*airfoil*), Communities and Crime (*crimepredict*) and Energy Efficiency Dataset (*energy efficiency*), all from [8]. In this case, as can be seen, General Tree achieved a statistically significant improvement with respect the other methos in most cases.

For each batch of experiments we tested each algorithm with each dataset using a ten fold cross-validation, and running each experiment ten times to reduce statistical irregularities.

Table 1. Comparison of the General Tree Algorithm, with C4.5, Naive Bayes and a Multilayer Perceptron

Dataset	(1)	(2)	(3)	(4)
abm	99.96	98.82 •	98.44 •	96.65 •
acath	74.72	66.54 •	74.78	74.67
dmd	92.24	89.42	89.37	99.29 ○
Average	88.97	84.93	87.53	90.20

○, • statistically significant improvement or degradation.

Table 2. Comparison two instances of the General Tree Algorithm, with M5P and a Multilayer Perceptron

Dataset	(1)	(2)	(3)	(4)
airfoil	39.06	43.44	53.89 ○	63.88 ○
crimepredict	59.42	59.69	84.00 ○	95.64 ○
energy efficiency	9.10	6.18 •	10.18	11.95 ○

○, • statistically significant improvement or degradation.

4 Conclusion

Even though we have only tested and compared the results obtained with General Trees with few datasets, results suggest that the General Tree can deal better with regression problems than with classification problems. However, more experimentation is needed to deterimine if some unseen factor have altered results, or what is the reason for such behavior. At this moment we can conclude that:

- Using our General Tree scheme, we can resemble the ID3, C4.5, M5 or other classification trees.
- Using different error measurements criteria, we can obtain general trees with performance close to other classifiers.
- The General Tree can be constructed for classification or regression by suppressing or adding the regression step at the leaves.
- As in C4.5 or M5, we can deal with continuous values, but C4.5 can not deal with regression, and M5 gives in our experiments a slightly greater error.

The biggest drawback of the General Tree is its computational complexity because it could take minutes, hours or even days to complete a single tree, depending on the number of samples and attributes involved. We estimate that if we use more than 400 attributes for a regression problem, with around 50,000 samples, the algorithm may delay a week to obtain a tree.

For a second phase, we will compare General Tree with more sophisticated methods, like Support Vector Machines.

Acknowledgment. The authors would like to thank to Universidad Anahuac-Mayab for supporting this work.

References

1. Quinlan, J.R.: Induction of Decision Trees. New South Wales Institute of Technology. Centre for Advanced Computing Sciences. Boston-Klwer Academic Publishers (1986)
2. Quinlan, J.R.: Improved use of Continuous Attributes in C4.5. Journal of Artificial Intelligence Research, 77–90
3. Quinlan, J.R.: Learning with continuous classes. In: Proceedings of the Int. Conference on Artificial Intelligence, pp. 343–348. World Scientific, Singapore
4. Wang, Y., Witten, I.H.: Induction of Model Trees for Predicting Continuous Classes. (Working paper 96/23). University of Waikato, Department of Computer Science, Hamilton, New Zealand
5. Khandani, A.E., Kim, A.J., Lo, A.W.: Consumer Credit Risk Models via Machine Learning Algorithms. Journal of Banking and Finance 34, 2767–2787 (2010)
6. Federal Reserve. FRB: G.19 – Consumer Credit. Retrieved (January 2014), `http://www.federalreserve.gov/releases/g19/Current/`
7. Spanos, A., Harrell, F.E., Durack, D.T.: Differential diagnosis of acute meningitis: An analysisof the predictive value of initial observations. Duke University Medical Center
8. Bache, K., Lichman, M. (UCI) Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences (2013), `http://archive.ics.uci.edu/ml`