Benchmarking Open-Source Tree Learners in R/RWeka

Michael Schauerhuber[1], Achim Zeileis[1], David Meyer[2], Kurt Hornik[1]

Department of Statistics and Mathematics[1]
Institute for Management Information Systems[2]
Wirtschaftsuniversität Wien
A-1090 Wien, Austria
Firstname.Lastname@wu-wien.ac.at

Abstract. The two most popular classification tree algorithms in machine learning and statistics—C4.5 and CART—are compared in a benchmark experiment together with two other more recent constant-fit tree learners from the statistics literature (QUEST, conditional inference trees). The study assesses both misclassification error and model complexity on bootstrap replications of 18 different benchmark datasets. It is carried out in the R system for statistical computing, made possible by means of the **RWeka** package which interfaces R to the open-source machine learning toolbox **Weka**. Both algorithms are found to be competitive in terms of misclassification error—with the performance difference clearly varying across data sets. However, C4.5 tends to grow larger and thus more complex trees.

1 Introduction

Due to their intuitive interpretability, tree-based learners are a popular tool in data mining for solving classification and regression problems. Traditionally, practitioners with a machine learning background use the C4.5 algorithm (Quinlan, 1993) while statisticians prefer CART (Breiman, Friedman, Olshen and Stone, 1984). One important reason for this is that free reference implementations have not been easily available within an integrated computing environment. RPart, an open-source implementation of CART, has been available for some time in the S/R package **rpart** (Therneau and Atkinson, 1997) while the open-source implementation J4.8 for C4.5 became available more recently in the **Weka** machine learning package (Witten and Frank, 2005) and is now accessible from within R by means of the **RWeka** package (Hornik, Zeileis, Hothorn and Buchta, 2007). With these software tools available, the algorithms can be easily compared and benchmarked on the same computing platform: the R system for statistical computing (R Development Core Team 2006). The principal concern of this contribution is to provide a neutral and unprejudiced review, especially taking into account classical beliefs (or

This is a preprint of an article published in *Data Analysis, Machine Learning, and Applications (Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation e.V., Albert-Ludwigs-Universität Freiburg, March 7–9, 2007).*

preconceptions) about performance differences between C4.5 and CART and heuristics for the choice of hyper-parameters. With this in mind, we carry out a benchmark comparison, including different strategies for hyper-parameter tuning as well as two further constant-fit tree models—QUEST (Loh and Shih, 1997) and conditional inference trees (Hothorn, Hornik and Zeileis, 2006). The learners are compared with respect to misclassification error and model complexity on each of 18 different benchmarking data sets by means of simultaneous confidence intervals (adjusted for multiple testing). Across data sets, the performance is aggregated by consensus rankings.

2 Design of the Benchmark Experiment

The simulation study includes a total of six tree-based methods for classification. All learners were trained and tested in the framework of Hothorn, Leisch, Zeileis and Hornik (2005) based on 500 bootstrap samples for each of 18 data sets. All algorithms are trained on each bootstrap sample and evaluated on the remaining out-of-bag observations. Misclassification rates are used as predictive performance measures, while model complexity requirements of the algorithms under study are measured by the number of estimated parameters (number of splits plus number of leafs). Performance and model complexity distributions are assessed for each algorithm on each of the datasets. In our setting, this results in 108 performance distributions (6 algorithms on 18 data sets), each of size 500. For comparison on each individual data set, simultaneous pairwise confidence intervals (Tukey all-pair comparisons) are used. For aggregating the pairwise dominance relations across data sets, median linear order consensus rankings are employed following Hornik and Meyer (2007). A brief description of the algorithms and their corresponding implementation is given below.

CART/RPart: *Classification and regression trees* (CART, Breiman et al., 1984) is the classical recursive partitioning algorithm which is still the most widely used in the statistics community. Here, we employ the open-source reference implementation of Therneau and Atkinson (1997) provided in the R package **rpart**. For determining the tree size, cost-complexity pruning is typically adopted: either by using a 0- or 1-standard-errors rule. The former chooses the complexity parameter associated with the smallest prediction error in cross-validation (RPart0), whereas the latter chooses the highest complexity parameter which is within 1 standard error of the best solution (RPart1).

C4.5/J4.8: C4.5 (Quinlan, 1993) is the predominantly used decision tree algorithm in the machine learning community. Although source code implementing C4.5 is available in Quinlan (1993), it is not published under an open-source license. Therefore, the Java implementation of C4.5 (revision 8), called J4.8, in **Weka** is the de-facto open-source reference implementation. For determining the tree size, a heuristic confidence threshold C

Table 1. Artificial [⋆] and non artificial benchmarking data sets

Data set	**# of obs.**	**# of cat. inputs**	**# of num. inputs**
breast cancer	699	9	-
chess	3196	36	-
circle ⋆	1000	-	2
credit	690	-	24
heart	303	8	5
hepatitis	155	13	6
house votes 84	435	16	-
ionosphere	351	1	32
liver	345	-	6
Pima Indians diabetes	768	-	8
promotergene	106	57	-
ringnorm ⋆	1000	-	20
sonar	208	-	60
spirals ⋆	1000	-	2
threenorm ⋆	1000	-	20
tictactoe	958	9	-
titanic	2201	3	-
twonorm ⋆	1000	-	20

is typically used which is by default set to $C = 0.25$ (as recommended in Witten and Frank, 2005). To evaluate the influence of this parameter, we compare the default J4.8 algorithm with a tuned version where C and the minimal leaf size M (default: $M = 2$) are chosen by cross-validation (J4.8(cv)). A full grid search for $C = 0.01, 0.05, 0.1, \ldots, 0.5$ and $M = 2, 3, \ldots, 10, 15, 20$ is used in the cross-validation.

QUEST: *Quick, unbiased and efficient statistical trees* are a class of decision trees suggested by Loh and Shih (1997) in the statistical literature. QUEST popularized the concept of unbiased recursive partitioning, i.e., avoiding the variable selection bias of exhaustive search algorithms (such as CART and C4.5). A binary implementation is available from `http://www.stat.wisc.edu/~loh/quest.html` and interfaced in the R package **LohTools** which is available from the authors upon request.

CTree: *Conditional inference trees* (Hothorn et al., 2006) are a framework of unbiased recursive partitioning based on permutation tests (i.e., conditional inference) and applicable to inputs and outputs measured at arbitrary scale. An open-source implementation is provided in the R package **party**.

The benchmarking datasets shown in Table 1 were taken from the popular UCI repository of machine learning databases (Newman, Hettich, Blake and Merz, 1998) as provided in the R package **mlbench**.

3 Results of the Benchmark Experiment

3.1 Results on Individual Datasets: Pairwise Confidence Intervals

Here, we exemplify—using the well-known Pima Indians diabetes and breast cancer data sets—how the tree algorithms are assessed on a single data set. Simultaneous confidence intervals are computed for all 15 pairwise comparisons of the 6 learners. The resulting dominance relations are used as the input for the aggregation analyses in Section 3.2.

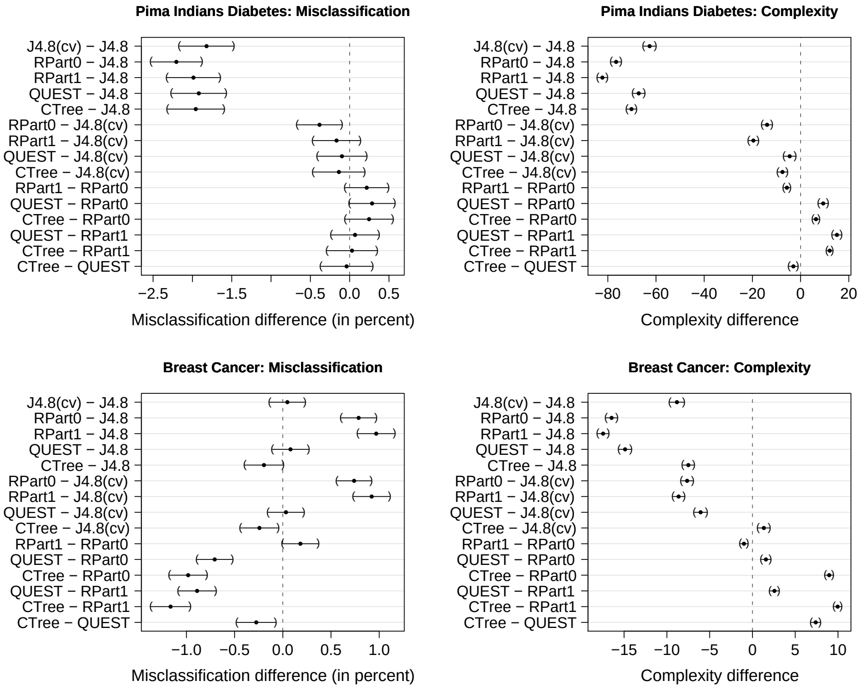

Fig. 1. Simultaneous confidence intervals of pairwise performance differences (left: misclassification, right: complexity) for Pima Indians diabetes (top) and breast cancer (bottom) data.

As can be seen from the performance plots for Pima Indian diabetes in Figure 1, standard J4.8 is outperformed (in terms of misclassification as well as model complexity) by the other tree learners. All other algorithm comparisons indicate equal predictive performances, except for the comparison of RPart0 and J4.8(cv), where the former learner performs slightly better than the latter. On this particular dataset tuning enhances the predictive performance of J4.8, while the misclassification rates of the differently tuned RPart versions are

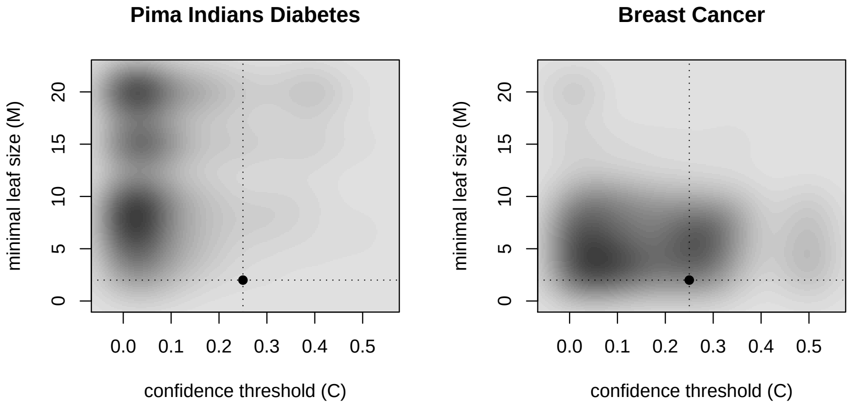

Fig. 2. Distribution of J4.8(cv) parameters obtained through cross validation on Pima Indians diabetes and breast cancer data sets.

not subject to significant changes. In terms of model complexity J4.8(cv) produces larger trees than the other learners. Looking at the breast cancer data yields a rather different picture: Both RPart versions are outperformed by J4.8 or its tuned alternative in terms of predictive accuracy. Similar to Pima Indians diabetes, J4.8 and J4.8(cv) tend to build significantly larger trees than RPart. On this dataset, CTree has a slight advantage over all other algorithms except J4.8 in terms of predictive accuracy. For J4.8 as well as RPart, tuning does not promise to increase predictive accuracy significantly. A closer look at the differing behavior of J4.8(cv) under cross validation for both data sets is provided in Figure 2. In contrast to the breast cancer example, the results based on the Pima Indians diabetes dataset (on which tuning of J4.8 caused a significant performance increase) show a considerable difference in choice of parameters. The multiple inference results gained from all datasets considered in this simulation experiment (just like the results derived from the two datasets above) form the basis on which further aggregation analyses of Section 3.2 are built upon.

3.2 Results Across Data Sets: Consensus Rankings

Having 18 × 6 = 108 performance distributions of the 6 different learners applied to 18 bootstrap data settings at hand, aggregation methods can do a great favor to allow for summarizing and comparing algorithmic performance. The underlying dominance relations derived from the multiple testing are summarized by simple sums in Table 2 and by the corresponding median linear order rankings in Table 3. In Table 2, rows refer to winners, while columns denote the losers. For example J4.8 managed to outperform QUEST on 11 datasets and 4 times vice versa, i.e., on the remaining 3 datasets, J4.8 and QUEST perform equally well.

Table 2. Summary of predictive performance dominance relations across all 18 datasets based on misclassification rates and model complexity (columns refer to losers, rows are winners).

Misclassification	**J4.8**	**J4.8(cv)**	**RPart0**	**RPart1**	**QUEST**	**CTree**	$\sum$
J4.8	0	2	9	10	11	8	40
J4.8(cv)	4	0	8	9	11	9	41
RPart0	5	6	0	7	10	7	35
RPart1	6	4	1	0	8	6	25
QUEST	4	2	2	5	0	7	20
CTree	7	6	7	8	9	0	37
$\sum$	26	20	27	39	49	37	

Complexity	**J4.8**	**J4.8(cv)**	**RPart0**	**RPart1**	**QUEST**	**CTree**	$\sum$
J4.8	0	1	0	0	2	0	3
J4.8(cv)	17	0	0	0	5	3	25
RPart0	18	18	0	0	13	15	64
RPart1	18	18	16	0	14	15	81
QUEST	15	13	5	4	0	10	47
CTree	18	14	3	2	8	0	45
$\sum$	86	64	24	6	42	43	

Table 3. Median linear order consensus rankings for algorithm performance

	Misclassification	**Complexity**
1	J4.8(cv)	RPart1
2	J4.8	RPart0
3	RPart0	QUEST
4	CTree	CTree
5	RPart1	J4.8(cv)
6	QUEST	J4.8

The median linear order for misclassification reported in Table 3 suggests that tuning of J4.8 instead of using the heuristic approach is worth the effort. A similar conclusion can be made for the RPart versions. Here, the median linear order suggests that the common one standard error rule performs worse. For both cases, the underlying dominance relation figures of Table 2 catch our attention. Regarding the first case, J4.8(cv) only dominates J4.8 in four of six data settings, in which a significant test decision for performance differences could be made. In addition the remaining 12 data settings yield equivalent performances. Therefore superiority of J4.8(cv) above J4.8 is questionable. In contrast the superiority of RPart0 vs. RPart1 seems to be more reliable but still the number of data settings producing tied results is high. A comparison of the figures of CTree and the RPart versions confirms previous findings (Hothorn et al., 2006) that CTree and RPart often perform equally well. The question concerning the dominance relation between J4.8 and RPart cannot be answered easily: Overall, the median linear order suggests that the J4.8 decision tree versions are superior to the RPart tree learners in terms of predictive performance. But still, looking at the underlying relations of the best performing versions of both algorithms (J4.8(cv) and RPart0) reveals that a confident decision concerning predictive superiority cannot be made. The number of differences in favor of J4.8(cv) is only two and no significant differences are reported on four data settings. A brief look at the complexity ranking (Table 3) and the underlying complexity dominance relations (Table 2, bottom) shows that J4.8 and its tuned version produce more complex trees than the RPart algorithms. While analogous analyses of comparing J4.8 versions to CTree do not indicate confident predictive performance differences, superiority of the J4.8 versions versus QUEST in terms of predictive accuracy is evident.

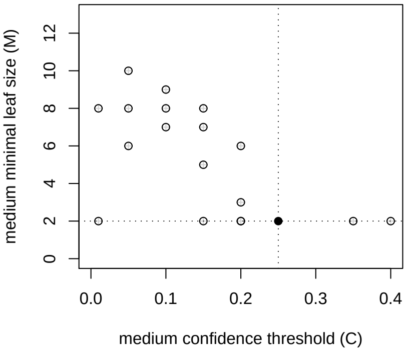

Fig. 3. Medians of the J4.8(cv) tuning parameter distributions for C and M

To aggregate the tuning results from J4.8(cv), Figure 3 depicts the median C and M parameters chosen for each of the 18 parameter distributions. It confirms the finding from the individual breast cancer and Pima Indians diabetes results (see Figure 2) that the parameter chosen by cross-validation can be far off the default values for C and M.

4 Discussion and Further Work

In this paper, we present results of a medium scale benchmark experiment with a focus on popular open-source tree-based learners available in R. With respect to our two main objectives—performance differences between C4.5 and CART, and heuristic choice of hyper-parameters—we can conclude: (1) The fully cross-validated J4.8(cv) and RPart0 perform better than their heuristic counterparts J4.8 (with fixed hyper-parameters) and RPart1 (employing a 1-standard-error rule). (2) In terms of predictive performance, no support for the claims of (clear) superiority of either algorithm can be found: J4.8(cv) and RPart0 lead to similar misclassification results, however J4.8(cv) tends to grow larger trees. Overall, this suggests that many beliefs or preconceptions about the classical tree algorithms should be (re-)assessed using benchmark studies. Our contribution is only a first step in this direction and further steps will require a larger study with additional datasets and learning algorithms.

References

BREIMAN, L., FRIEDMAN, J., OLSHEN, R. and STONE, C. (1984): *Classification and Regression Trees.* Wadsworth, Belmont, CA, 1984.

HORNIK, K. and MEYER, D. (2007): Deriving Consensus Rankings from Benchmarking Experiments In: *Advances in Data Analysis (Proceedings of the 30th Annual Conference of the Gesellschaft für Klassifikation e.V., March 8–10, 2006, Berlin)*, Decker, R., Lenz, H.-J. (Eds.), Springer-Verlag, 163–170.

HORNIK, K., ZEILEIS, A., HOTHORN, T. and BUCHTA, C. (2007): ***RWeka**: An R Interface to **Weka**.* R package version 0.3-2. `http://CRAN.R-project.org/`.

HOTHORN, T., HORNIK, K. and ZEILEIS, A. (2006): Unbiased Recursive Partitioning: A Conditional Inference Framework. *Journal of Computational and Graphical Statistics*, 15(3), 651–674.

HOTHORN, T., LEISCH, F., ZEILEIS, A. and HORNIK, K. (2005): The Design and Analysis of Benchmark Experiments. *Journal of Computational and Graphical Statistics*, 14(3), 675–699.

LOH, W. and SHIH, Y. (1997): Split Selection Methods for Classification Trees. *Statistica Sinica*, 7, 815–840.

NEWMAN, D., HETTICH, S., BLAKE, C. and MERZ C. (1998): UCI Repository of Machine Learning Databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

QUINLAN, J. (1993): *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, Inc., San Mateo, CA.

R DEVELOPMENT CORE TEAM (2006): *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, `http://www.R-project.org/`.

THERNEAU, T. and ATKINSON, E. (1997): An Introduction to Recursive Partitioning Using the **rpart** Routine. *Technical Report.* Section of Biostatistics, Mayo Clinic, Rochester, `http://www.mayo.edu/hsr/techrpt/61.pdf`.

WITTEN, I., and FRANK, E. (2005): *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, San Francisco, 2nd edition.