Functional Models for Regression Tree Leaves

Luís Torgo
LIACC - University of Porto
R. Campo Alegre, 823 - 4150 Porto - PORTUGAL
email : ltorgo@ncc.up.pt
WWW : http://www.ncc.up.pt/~ltorgo

Abstract

This paper presents a study about functional models for regression tree leaves. We evaluate experimentally several alternatives to the averages commonly used in regression trees. We have implemented a regression tree learner (HTL) that is able to use several alternative models in the tree leaves. We study the effect on accuracy and the computational cost of these alternatives. The experiments carried out on 11 data sets revealed that it is possible to significantly outperform the "naive" averages of regression trees. Among the four alternative models that we evaluated, kernel regressors were usually the best in terms of accuracy. Our study also indicates that by integrating regression trees with other regression approaches we are able to overcome the limitations of individual methods both in terms of accuracy as well as in computational efficiency.

1 INTRODUCTION

In this paper we present an empirical evaluation of alternative regression models for the leaves of decision trees that deal with a continuous goal variable (i.e. regression trees). This study aims at providing a better insight on the trade-off between accuracy and computational efficiency of several regression models in the context of regression tree leaves.

Regression trees provide quite simple and interpretable regression models with reasonable accuracy. However, these methods are known for their instability (Breiman, 1996). Several authors have tried to improve partition-based methods approximations by fitting more complex models within each partition. Weiss & Indurkhya (1995) used a nearest neighbor approximation in their regression rules inductive system. Quinlan (1992) and Karalic (1992) have used linear models in regression tree leaves. In our study we add to these alternatives kernel regression models (Watson, 1964; Nadaraya, 1964). We compare all these alternatives in the context of regression trees. We have developed an inductive system (HTL), that learns regression trees and is able to use any of these alternative regression models in the tree leaves. We empirically compare the approximations provided by these alternatives on several data sets. Our experiments revealed significant differences among them.

HTL is a hybrid system that integrates several approaches to the regression problem. Experimental comparisons of different learning methods on various real world problems have shown the impossibility of selecting a method that performs better in all domains (Michie et al., 1994). This is sometimes called the selective superiority problem (Broadley , 1995). Several authors have tried to automatically identify the applicability of the different methods (Aha, 1992; Brazdil et al., 1994). Other lines of research have tried to take advantage of the different biases of the methods and obtain combined predictions (Breiman, 1996; Freund & Schapire, 1995; Wolpert, 1992). The main problem with these approaches is that the resulting predictions are no longer easily interpretable by the user. In effect, several different models are contributing for the final prediction[1]. Comprehensibility has always been considered a key advantage of Machine Learning (ML) approaches to the inductive task. HTL divides the input space into a set of

[1] A noticeable exception is an early work by Brazdil and Torgo (1990) where an integrated theory was built from different individually learned theories.

mutually exclusive partitions described by propositional assertions on the input variables. These assertions provide an interpretable view of the unknown regression function. Within these partitions HTL is able to fit several alternative regression models to improve predictive accuracy. By proceeding this way we intend to take advantage of the different biases of the methods. We were able to confirm our expectations in the experiments we have carried out.

In the next section we briefly describe the regression problem and present the approaches we propose to integrate. We then present HTL in section 3. The experiments carried out are described in section 4. Section 5 outlines future research plans. We them summarize with the conclusions of this work.

2 REGRESSION PROBLEMS

The problem of regression consists of obtaining a functional model that relates the value of a target continuous variable Y with the values of variables X_1, X_2, ..., X_v (the predictors). This model is usually obtained using samples of the unknown regression function.

Traditional statistical approaches to this problem assume a particular parametric function and use all samples to obtain the values of the function parameters that are optimal according to some fitting criterion. These global parametric approaches have been widely used and give good predictive results when the assumed model correctly fits the data. Modern statistical approaches to regression include k-nearest neighbors (Fix & Hodges, 1951), kernel regression (Watson, 1964; Nadaraya, 1964), local polynomial regression (Stone, 1977; Cleveland, 1979), radial basis functions, neural networks, projection pursuit regression, adaptive splines, and others.

ML researchers have always been mainly concerned with classification problems. The few existing systems dealing with regression usually build axis orthogonal partitions of the input space and fit a parametric model within each of these partitions. These models are usually very simple (like the average, the median or linear models). The non-linearities of the domain are captured by the axis-orthogonal partitions rather than by the models fitted within those partitions.

2.1 REGRESSION TREES

Regression trees are a special type of decision trees that deal with a continuous goal variable. These methods perform induction by means of an efficient recursive partitioning algorithm. The choice of the test at each node of the tree is usually guided by a least squares error criterion. Binary trees consider two-way splits at each tree node. The best split at each node t is the split s that maximizes

$$\Delta Err(s,t) = Err(t) - P_L \times Err(t_L) - P_R \times Err(t_R) \qquad \textbf{(1)}$$

where P_L and P_R are the proportions of instances that fall respectively to the left and right branch of the node t; $Err\ (t_L)$ and $Err\ (t_R)$ are the errors of the left and right branches; and $Err\ (t)$ is the mean squared error at node t given by $\frac{1}{N_t}\sum_i^{N_t}\left(y_i - \bar{y}_t\right)^2$

Other important issues of this approach to regression are: the decision of when to stop tree growth; the related issue of how to post-prune the tree; and the model assignment rule used on the leaves. Some of the existent approaches to regression trees differ on this later issue. CART (Breiman et al., 1984) assigns a constant to the leaves (the average Y value). RETIS (Karalic, 1992) and M5 (Quinlan, 1992) are able to use linear models of the predictor variables. Weiss & Indurkhya's system uses nearest neighbor models on the right side of propositional rules. In another work, Indurkhya & Weiss (1995) suggest that the same approach could be followed on regression trees. We will see how our system HTL adds a further degree of smoothness by allowing other models at the tree leaves.

Regression trees obtain good predictive accuracy on many domains. However, the simple models used in their leaves have some limitations regarding the kind of functions they are able to approximate. Nevertheless, they provide interpretable models of the data and have low computational demands (both running time and storage requirements).

2.2 LINEAR REGRESSION

Linear regression is one of the most used statistical techniques. A linear form is assumed for the unknown regression function and the parameters of the model are estimated using a least squares criterion (Draper & Smith, 1981). Parameter estimation is done solving a set of equations on these parameters. To avoid numerical difficulties several techniques have been proposed to solve these equations. Among the most successful is Singular Value Decomposition (SVD) which we have adopted in our implementation. This technique has an interesting side effect. By zeroing the small singular values we are able to eliminate irrelevant variables from the linear model (Press et al., 1992).

2.3 KERNEL REGRESSION

Kernel regression is a non-parametric statistical method that belongs to a research field usually called *local modeling*[2]. These methods do not perform any kind of generalization of the given samples[3] and "delay learning" till prediction time. Given a query point **q** a prediction is obtained using the training samples that are "most similar" to **q**. Similarity is measured by means of a distance metric defined in the hyper-space of V predictor variables. Kernel regressors obtain the prediction for a query point **q**, by a weighted average of the Y values of its neighbors. The weight of each neighbor is calculated by a function of its distance to **q** (called the kernel function). These kernel functions give more weight to neighbors that are nearer to **q**. The notion of neighborhood (or bandwidth) is defined in terms of distance from **q**. The prediction for query point **q** is obtained by

$$y'(\mathbf{q}) = \frac{1}{SKs} \sum_{i}^{N} K\left(\frac{D(\mathbf{x_i}, \mathbf{q})}{h}\right) \times y_i \qquad \textbf{(2)}$$

where $D(.)$ is the distance function between two instances; $K(.)$ is a kernel function; h is a bandwidth value; $<\mathbf{x}_i, y_i>$ are training samples; and

$$SKs = \sum_{i}^{N} K\left(\frac{D(\mathbf{x_i}, \mathbf{q})}{h}\right)$$

One important design decision when applying local modeling is the choice of the bandwidth. Many alternative methods exist (Cleveland & Loader, 1995). One possibility is to use a nearest neighbor bandwidth. This technique sets the bandwidth as the distance to the kth nearest neighbor of **q**. This means that only the k nearest neighbors will influence the prediction of **q**. This is one of the bandwidth selection methods that HTL uses.

Other design issues of kernel regression include the choice of the kernel function and the definition of a distance function. Again many alternatives exist in the literature. In the next section we will discuss the choices we made.

[2] Related methods exist within the ML community that are usually called lazy learners or instance-based algorithms.

[3] This is not completely true for some kind of instance-based learners (Aha, 1990; Salzsberg, 1991).

Local modeling can be very sensitive to the presence of irrelevant features. The computational complexity of these methods is high when a large number of samples is available (Deng & Moore, 1995). Several techniques exist that try to overcome these limitations. Feature weighing can help to reduce the influence of irrelevant features by better "tuning" the distance function (Wettshereck et al. *in press*). Sampling techniques, indexing schemes and instance prototypes are some of the methods used with large samples (Aha, 1990). Still, local regression methods have a strong limitation when it comes to getting a better insight of the structure of the domain. The resulting regression models have low interpretability.

3 HYBRID REGRESSION TREES

We will now briefly describe the system HTL that we will use for a comparative evaluation of alternative regression tree leaf models. The goal of this system is to try to keep the efficiency and interpretability of regression trees while improving their accuracy by increasing the non-linearity of the functions used in the leaves.

3.1 THE LEARNING STAGE

The learning stage of our method consists of developing a regression tree. The algorithm we use is similar to the one described by Breiman et al. (1984) employed in CART. We grow a binary tree with the goal of minimizing the mean squared error (MSE) obtained by the average Y value using a formulation similar to the one given in Equation 1. This corresponds to dividing the input space into regions with similar Y values.

Splits on continuous variables are done by choosing the cut-point value that maximizes the gain of MSE. Splits on discrete variables consist of finding the subset of values that maximizes the referred gain.

With respect to the stopping criterion for tree growth we use two thresholds. The first is a minimum number of examples in a node. The default value used in our experiments is 20 examples. The second stopping criterion is a minimum value of a statistic called *Coefficient of Variation* (Chatfield, 1983) that captures the relative spread of a set of values. This coefficient is given by

$$CV_Y = \frac{s_y}{\bar{y}} \qquad \textbf{(3)}$$

where s_y is the standard deviation of the Y values.

By default HTL stops growing a tree when the value of *CV* reaches 0.15. These stopping criteria lead to overly large trees so a post pruning phase is necessary.

HTL uses a pruning algorithm that estimates the true error of a node based on the resubstitution estimate obtained with the training data. The resubstitution estimate gives an over-optimistic estimate of the node error. This estimate uses the average *Y* value to obtain the mean squared error in a node (see Equation 1). HTL tries to improve this estimate by using a confidence interval for the average value of *Y*. Using standard statistical estimation techniques we obtain a confidence interval of the form $\bar{y} \pm t_{(1-\alpha/2)} \frac{s_y}{\sqrt{N}}$, where $t_{(1-\alpha/2)}$ is the given from the *t* distribution tables corresponding to a 100(1-α) percent confidence interval for the mean *Y* value. Using the two extreme values defined by this interval HTL uses them to obtain two values of the node error. The higher of these values is chosen as a kind of most pessimistic estimate of the node error. This is an heuristic estimation technique. It has the advantage of avoiding the growth of several trees to obtain better estimates as it is done by CART for instance that uses *N*-fold Cross Validation.

3.2 MAKING PREDICTIONS

It is on prediction tasks that the hybrid nature of HTL becomes evident. Given a query point **q**, we run it through the tree until a leaf node is reached. At this stage instead of using the average *Y* value of the training samples on the leaf, we may use more sophisticated regression models. These models are obtained using only the training samples within the leaf.

The reason why we call HTL a hybrid system relies on the fact that while the trees are grown with the goal of minimizing the MSE using the average *Y* value as target model, HTL may then use other models during prediction. The obvious question is why not using the target models already during tree growth. This would correspond to changing the calculation of MSE in Equation 1 to something like $\frac{1}{N_t}\sum_i^{N_t}\left(y_i - g(x)\right)^2$, where g(x) is any target model built using the samples within the node in question. The problem with this approach is computational complexity. While for calculating averages there are fast incremental algorithms that enable an efficient evaluation of all candidate splits for each node, the same is not true for some of the models we use. However, it remains an open research question how these two approaches compare both in terms of accuracy and computational complexity.

3.2.1 Linear Models

One of the alternatives to averages provided by HTL is to build a linear regression model using the training data in the leaf. This model is obtained using the already mentioned SVD technique. Predictions of testing instances that fall on this leaf are then obtained by evaluating the linear model. This turns out to be similar to the strategy followed by RETIS (Karalic, 1992) or M5 (Quinlan, 1992).

3.2.2 Local Models

Local models are computationally more expensive as they need to be re-evaluated for each query. HTL is able to use k-nearest neighbors models and kernel regressors. Although they are usually called non-parametric approaches they strongly depend on several design issues. One of the key points is the distance function. HTL uses a diagonally weighted Euclidean distance (Atkeson et al., *in press*) defined as

$$D(\mathbf{x_1}, \mathbf{x_2}) = \sqrt{\sum_v FW_v \times \delta(\mathbf{x_{1,v}}, \mathbf{x_{2,v}})^2} \qquad \textbf{(4)}$$

where $\mathbf{x}_i$ is an instance vector; $\mathbf{x}_{i,j}$ is the value of variable *j* on instance *i*; FW_i is the weight of variable *i*; and $\delta(v_1, v_2)$ is the distance between two variable values.

In order to avoid overweighing of discrete variables with respect to continuous variables we use a ramping function (Hong, 1994) for the later.

Another important design issue relates to feature weights. Several different approaches exist in the literature (see Wettscherek et al. (*in press*) for an excellent review in the context of k-NN). Recently, Robnik-Sikonja & Kononenko (1996) presented a new method based on a previous work on *Relief* (Kira & Rendell, 1992; Kononenko, 1994) for estimating variable relevance in the context of regression. They describe an algorithm called *RReliefF* and successfully applied it in the context of regression trees. HTL uses this same method for estimating the weights of the variables that are used in distance calculations.

HTL uses a gaussian kernel function with the form

$$K(d) = e^{-d^2} \qquad \textbf{(5)}$$

Atkeson et al. (*in press*) claim that the choice of the kernel function is not a critical design issue as long as the function is reasonably smooth.

Finally, HTL provides several alternative ways of specifying the bandwidth that can be applied globally or locally. In the global setting all of the given training instances are used for estimation of the bandwidth. The local approach estimates a bandwidth for each leaf using only the respective training instances.

3.3 RELATIONS TO OTHER WORK

Integrating partition-based methods with instance-based models is not a new idea. Several authors have followed a similar integration schema for classification tasks. Smyth et al. (1995) integrate class probability trees with kernel density estimators to obtain non-parametric estimates of class probabilities. The authors report significant improvements over decision trees and kernel density alone for certain class of problems. Their approach deals with discrete target variables as opposed to ours. Other approaches within the classification scenario are EACH (Salzberg, 1991) and RISE (Domingos, 1996) that try to generalize instances producing exemplars that can be seen as axis-parallel partitions of the input space.

Deng & Moore (1995) describe a similar approach but for regression tasks. Their *multires* system integrates *k*d-trees with kernel regression producing what they call kernel regression trees. *K*d-trees (Bentley, 1975) are a method of structuring a set of records each containing a set of measured variables. They are binary trees built in a similar fashion as decision trees. However, while regression trees are built with the goal of grouping data points with similar Y values, *k*d-trees try to optimize the storage of these data points in order to achieve faster access time. The consequence is that the splitting criterion of the two approaches is completely different. Moreover, while HTL obtains the prediction using one leaf of the regression tree, *multires* obtains the prediction by a combination of contributions of several nodes (not necessarily leaves). The integration in HTL is done with interpretability as one of the main goals, while in *multires* the main issue is obtaining an efficient way of structuring all the training set to make kernel regression computationally more efficient.

The work of Weiss & Indurkhya (1995) integrates a rule-based partitioning method with k-nearest neighbors. However, these authors deal with regression by mapping it into a classification problem. The original Y values are transformed into a set of I intervals. One problem of this approach is that the number I needs to estimated which can be computationally demanding (Torgo & Gama, 1997). Moreover, the search space explored by rule learners is larger than the one of trees. This means that rule learning systems may find solutions that tree learners can not but at the cost of computational complexity (Indurkhya & Weiss, 1995). These two later observations indicate that HTL should be able to cope with larger problems than Weiss & Indurkhya's system. Another important difference to our work is on the type of local models used. Weiss & Indurkhya's system uses k-NN while HTL is able to use kernel regressors. As we will see in the next section this makes a significant difference in accuracy on the data sets we have tried.

M5 system (Quinlan, 1993) also uses regression trees and k-NN. However, this system performs prediction combination instead of integrating the two methodologies like in HTL. This means that two models are independently obtained and their predictions combined. This has strong implications in terms of comprehensibility as mentioned before.

4 EXPERIMENTS

In this section we experimentally evaluate the behavior of HTL. We compare the different alternative regression models that HTL is able to use in the tree leaves. We also compare HTL to the individual methods it integrates.

The data sets we have chosen to carry out our experiments were mainly obtained from the UCI Machine Learning Repository (Merz & Murphy, 1996) :

Housing - this data set contains 506 instances described by 13 continuous input variables. The goal consists of predicting the housing values in suburbs of Boston.

Auto-Mpg - 398 instances described by 3 nominal and 4 continuous variables. The target variable is the fuel consumption (miles per gallon).

Servo - 167 instances; 4 nominal attributes.

Machine-cpu - 209 instances; 6 continuous attributes. The goal is to predict the cpu relative performance based on other computer characteristics.

Auto-price (Automobile database) - 159 cases; 16 continuous attributes. This data set is built from the *Automobile* database by removing all instances with unknown values from the original 205 cases. Nominal attributes were also removed. The goal is to predict the car prices based on other characteristics.

Auto-losses (Automobile database) - based on the same database we have built a different data set consisting of 164 instances described by 11 nominal attributes and 14 continuous variables. From the original data we only removed the cases with unknown value on the attribute "normalized-losses". This attribute describes the car insurance normalized losses. This variable was taken as the predicting goal.

Abalone - 4177 instances; 1 nominal and 7 continuous attributes. The goal consists of predicting the age of abalone using some of its physical measurements.

Wisconsin Prognostic Breast Cancer - predicting recurrence time in 194 breast cancer cases (4 instances with unknowns removed); 32 continuous attributes.

Gate (non-UCI data set) - 300 instances; 10 continuous variables. The problem consists of predicting the time to collapse of an electrical network based on some monitoring variable values.

Artificial1 (non-UCI data set) - 5000 instances of an artificial data set used by Breiman et al. (1984); 10 continuous variables. The generating function depends on the value of one predictor variable. Depending on this value the target is generated from two alternative linear functions of other variables.

Artificial2 (non-UCI data set) - another artificial data set with similar characteristics to the previous one. The difference is that the two used functions are highly non-linear.

All experiments were carried out using the following methodology. Each original data set was permuted to eliminate any ordering effects. A 10-fold cross validation evaluation was done on the permuted data. All methods were compared on the same partitions of the data. For each iteration we have recorded the Mean Average Deviation (MAD) of the method predictions and the cpu time spent on the complete iteration (learning plus testing). Averages across the 10 folds were collected and *t*-Student paired tests for testing significant differences were done for each pair of methods.

The best result is presented in italics in the tables of results. Statistical significance results are relative to the method presented in the first column of the table. The results of the methods in the other columns may be underlined or double underlined. Underlined results indicate a significant difference with 90% confidence, while double underline represents 95% confidence. The other differences are statistically insignificant.

Table 1 shows the MAD results of the same regression tree with different regression models at the leaves. In these experiments local models (kernel regression and k-nearest neighbors) were obtained using a 3-nearest neighbor bandwidth selection method.

These results give a clear indication of the advantage of using non-parametric models in the tree leaves. In effect, both KR models and kNN models are frequently significantly better than the other models and are never significantly worse (with the exception of *Artificial1* that is clearly biased towards linear models). On the other hand, parametric models obtain very bad accuracy on certain data sets indicating the inadequacy of their assumptions regarding the regression surface. On some data sets KR models achieve worse results than kNN models, but always without statistical significance. However, the opposite occurred on 3 data sets, leading us to conclude that KR models proved to be superior on these data sets (which can also be seen from the average rank position).

Table 1- MAD Comparisons Between Different Models at Tree Leaves.

	KR Trees	**kNN Trees**	**Linear Trees**	**Avg. Trees**
Housing	*2.8±0.5*	2.9±0.4	3.9±2.7	3.4±0.6
Mpg	2.4±0.4	*2.3±0.4*	18.0±5.6	3.2±0.6
Cpu	*31.2±15.1*	31.5±14.7	35.7±11.7	42.4±12.9
Servo	*0.4±0.2*	*0.4±0.2*	0.9±0.2	0.5±0.2
Price	*1637±570*	1662±581	2463±518	1682±443
Losses	*12.3±4.5*	13.7±4.4	105.2±34.9	27.4±6.5
Abalone	*1.7±0.1*	*1.7±0.1*	1.8±0.1	1.9±0.1
Breast	28.5±5.6	28.6±5.6	*28.2±5.6*	29.8±4.9
Gate	*0.4±0.1*	0.5±0.1	0.8±0.2	1.3±0.2
Artificial1	0.9±0.0	0.9±0.0	0.8±0.0	0.9±0.0
Artificial2	*1.1±0.0*	*1.1±0.0*	1.3±0.1	1.4±0.1
Avg.Rank	1.6	1.7	3.1	3.4

Regarding execution time, average models significantly outperform (99% level) the other methods in all data sets. Linear models significantly lose over local models in smaller data sets. In the larger data sets this tendency is somehow inverted. Instance-based methods execution time is related to the size of both training and testing sets. Their complexity is O(#Train×#Test×#Vars). Parametric models, on the other hand, depend only on the training set. It is thus natural that in larger data sets local models start to lose efficiency. This tendency is however alleviated by the hybrid trees as the local models are applied on a restricted set of examples instead of all training set.

Table 2 shows the accuracy of the individual regression models alone (i.e. without growing a tree). We can observe from this table that kernel regression models are clearly the best function approximators among the ones we have tested on these data sets. There is an evident superiority of non-parametric approaches which is natural taking into account the strong assumptions on the function form made by the others.

Table 2 - MAD of the Individual Models.

	KR	**kNN**	**LR**	**AVG**
Housing	*2.7±0.6*	2.9±0.6	3.4±0.7	13.3±2.4
Mpg	*2.3±0.3*	2.3±0.3	4.4±0.4	6.8±0.9
Cpu	*29.8±12.9*	31.8±14.8	41.2±8.5	200.6±39.2
Servo	*0.6±0.2*	*0.6±0.2*	3.5±0.4	1.0±0.4
Price	*1845±540*	1972±558	2011±382	4504±1104
Losses	*11.7±4.6*	13.1±4.5	129.1±22.1	31.3±8.7
Abalone	1.69±0.1	*1.68±0.1*	1.9±0.0	2.4±0.1
Breast	28.5±5.6	28.6±5.6	*28.2±5.6*	56.0±10.7
Gate	*0.40±0.1*	0.44±0.1	1.0±0.1	5.2±1.2
Artificial1	*1.15±0.0*	1.17±0.0	1.9±0.1	5.7±0.7
Artificial2	*1.27±0.0*	1.29±0.0	1.8±0.1	162.0±5.9
Avg. Rank	1.2	1.9	3	3.3

We now compare these results to the results of the respective hybrid trees (i.e. applying these models in the leaves of the tree). Each bar of the graph in Figure 1 shows the absolute percentage difference of accuracy between the hybrid tree and the respective individual model. Wins of the hybrid models are indicated by negative percentage values.

The results pictured in Figure 1 need to be analyzed carefully. First of all there is a clear difference between the hybrids with parametric models in the leaves (AVG. Trees and LR Trees) and the non-parametric hybrids. The former have a clear and significant advantage over the respective individual models. This was somehow expected as the individual models make very strong assumptions regarding the unknown regression surface. Two noticeable exceptions occur in the *Price* and in the *Mpg* data sets where the linear regression model has a statistically significantly (95% confidence) lower error than the respective hybrid tree. This seems to indicate that in these domains the regression surface has an approximately "planar" form. In the *Housing* data set a similar effect occurs but without statistical significance.

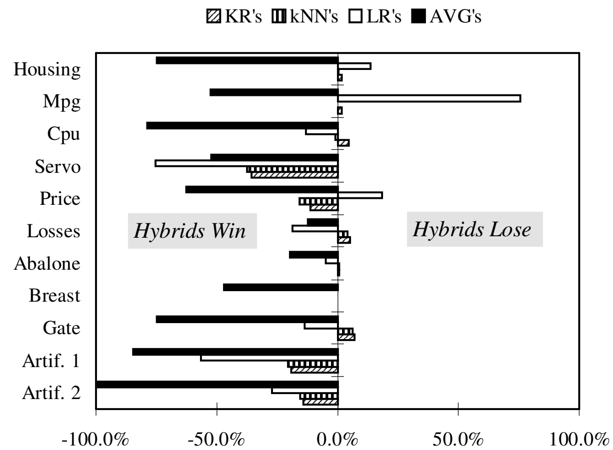

Figure 1 : Hybrids vs. Individual Models (absolute percentage differences).

With respect to the non-parametric models the scenario is different. In some data sets the non-parametric models alone are slightly superior to the hybrids although never with statistical significance. On the other hand hybrids have a statistically significant advantage in four data sets. A possible explanation for this observation has to do with the known difficulty of local modeling methods to deal with strong discontinuities of the regression surface (Cleveland & Loader, 1995). These methods provide a smoothing effect that is not adequate when strong discontinuities exist. Hybrid models on the contrary can easily cope with this type of situations due to the nature of the partitions provided by the tree. This explanation can be confirmed in the two artificial data sets where we know by their definition that the regression surface suffers one abrupt discontinuity. This is easily captured by the regression tree by a split in the root node that separates the two hyper-planes. It is possible that the same is happening both in the *Servo* and the *Price* data sets although we can not be sure about it as the true regression surface is unknown. This means that hybrid methods are able to compensate for a limitation of local models. Hybrid models never performed significantly worse than the individual models (with the exception of *Abalone*). Their results are never too bad which does not happen with the individual models on certain domains. This provides evidence for our expectations regarding taking advantage of the different biases of the models. Moreover, we should add that while local models provide no information on the structure of the domain, hybrid models build a symbolic representation of the domain which improves our knowledge of the unknown function.

Finally, a word on execution times. In the smaller data sets hybrid methods suffer from the burden of having to learn a tree. However, the numbers are almost irrelevant as they are usually less than 1 cpu second. In the larger data sets the local model hybrids significantly outperform the non-hybrid approaches. The reason is that local models are being applied on subsets of the training data. Hybrid methods are thus able to improve the efficiency of local models without significant accuracy loss.

5 FUTURE IMPROVEMENTS

We are finishing the implementation of local linear polynomials within HTL. This will enhance the range of models available for tree leaves.

An obvious improvement is to let HTL estimate the best model for each tree leaf. This would probably bring some gains in accuracy. However, this kind of approaches have usually strong computational costs.

A problem we intend to address is the storage requirements of HTL. We will explore techniques for generating prototypes (or exemplars). This would significantly decrease the computational demands of HTL, but would also most probably carry some accuracy loss in some domains (Wettscherek, 1994). MDL coding schemes could bring some guidance for this tradeoff between compactness of the representation and accuracy. Another related issue is indexing schemes that improve the case retrieval efficiency which is so important for local models.

6 CONCLUSIONS

We have presented an empirical study of hybrid regression trees. We have compared several alternative models for regression tree leaves. Regarding existing work on regression we have added the possibility of using kernel models in regression tree leaves. Our experiments with 11 data sets have shown the advantage of these models over the existing approaches.

The experiments carried out revealed that hybrid models are able to overcome some limitations of the individual models. Hybrid models achieve high accuracy levels across all tested domains. Moreover, they provide a symbolic representation of the unknown regression function that improves our knowledge of the domain.

The use of local models has a significant computational cost on large data sets. Hybrid models relax this problem by applying the models on subsets of the input domain without a significant loss on accuracy and even with some gains when the unknown regression surface has strong discontinuities.

References

Aha, D. (1990) : A study of instance-based learning algorithms for supervised learning tasks: Mathematical, empirical, and psychological evaluations. PhD Thesis. Tech. Report 90-42. University of California at Irvine, Department of Information and Computer Science.

Aha, D. (1992) : Generalizing from case studies : A case study. In Proceedings of the 9th International Conference on Machine Learning. Sleeman,D. & Edwards,P. (eds.). Morgan Kaufmann.

Atkeson,C.G., Moore,A.W., Schaal,S. (in press) : Locally Weighted Learning. Special issue on lazy learning. Aha, D. (Ed.). *Artificial Intelligence Review*.

Bentley,J.L. (1975) : Multidimensional binary search trees used for associative searching. *Communications of the ACM*, **18**(9), 509-517.

Brazdil,P. , Gama,J., Henery,B. (1994) : Characterizing the applicability of classification algorithms using meta-level learning. In Proceedings ECML-94. Bergadano,F. & De Raedt,L. (eds.). Lecture Notes in Artificial Intelligence, 784. Springer-Verlag.

Brazdil,P. Torgo,L. (1990) : Knowledge Acquisition via Knowledge Integration. In *Current Trends in Knowledge Acquisition*. Wielinga,B. et al. (eds.). IOS Press.

Breiman,L. (1996) : Bagging Predictors. In *Machine Learning*, **24**, (p.123-140). Kluwer Academic Publishers.

Breiman,L. , Friedman,J.H., Olshen,R.A. & Stone,C.J. (1984): Classification and Regression Trees, Wadsworth Int. Group, Belmont, California, USA, 1984.

Broadley, C. E. (1995) : Recursive automatic bias selection for classifier construction. *Machine Learning*, **20**, 63-94. Kluwer Academic Publishers.

Chatfield, C. (1983) : *Statistics for technology* (third edition). Chapman and Hall, Ltd.

Cleveland, W.S. (1979) : Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, **74**, 829-836.

Cleveland,W.S., Loader,C.R. (1995) : Smoothing by Local Regression: Principles and Methods (with discussion). In *Computational Statistics*.

Deng,K., Moore,A.W. (1995) : Multiresolution Instance-based Learning. In Proceedings of IJCAI'95.

Domingos,P. (1996) : Unifying Instance-based and Rule-based Induction. In *Machine Learning*, **24**-2, 141-168. Kluwer Academic Publishers.

Draper,N.R., Smith,H. (1981) : *Applied Regression Analysis*, 2nd edition, John Wiley.

Fix, E., Hodges, J.L. (1951) : Discriminatory analysis, nonparametric discrimination consistency properties. Technical Report 4, Randolph Filed, TX: US Air Force, School of Aviation Medicine.

Freund,Y., Schapire,R.E. (1995) : A decision-theoretic generalization of on-line learning and an application to boosting. Technical Report . AT & T Bell Laboratories.

Hong,S.J. (1994) : Use of contextual information for feature ranking and discretization. Technical Report RC19664, IBM. To appear in IEEE Trans. on Knowledge and Data Engineering.

Indurkhya,N., Weiss,S. (1995) : Using Case Data to Improve on Rule-based Function Approximation. In proceedings of the 1st International Conference on Case-base Reasoning (ICCBR'95). Lecture Notes in AI, 1010, p.217-228. Springer-Verlag.

Karalic,A. (1992) : Employing Linear Regression in Regression Tree Leaves. In Proceedings of ECAI-92. Wiley & Sons.

Kira,K., Rendell,L.A. (1992) : The feature selection problem : traditional methods and new algorithm. In Proceedings of AAAI'92.

Kononenko,I. (1994) : Estimating attributes : analysis and extensions of Relief. In Proceedings ECML-94. Bergadano,F. & De Raedt,L. (eds.). Lecture Notes in Artificial Intelligence, 784. Springer-Verlag.

Merz,C.J., Murphy,P.M. (1996) : UCI repository of machine learning databases [http:// www.ics.uci.edu/ MLReposiroty.html]. Irvine, CA. University of California, Dept. of Information and Computer Science.

Michie,D., Spiegelhalter;D.J. & Taylor,C.C. (1994): *Machine Learning, Neural and Statistical Classification*, Ellis Horwood Series in Artificial Intelligence, 1994.

Nadaraya, E.A. (1964) : On estimating regression. *Theory of Probability and its Applications*, **9**:141-142.

Press,W., Teukolsky,S., Vetterling,W., Flannery,B. (1992) : *Numerical Recipes in C*, 2nd edition. Cambridge University Press.

Quinlan, J.R. (1992): Learning with Continuos Classes. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*. World Scientific, 1992.

Quinlan,J.R. (1993) : Combining Instance-based and Model-based Learning. Proceedings of the 10th ICML. Morgan Kaufmann.

Robnik-Sikonja,M., Kononenko,I. (1996) : Context-sensitive attribute estimation in regression. Proceedings of the ICML-96 Workshop on Learning in Context-Sensitive Domains.

Salzberg,S. (1991) : A nearest hyperrectangle learning method. *Machine Learning*, **6**-3, 251-276. Kluwer Academic Publishers.

Smyth,P., Gray,A., Fayyad,U.M. (1995) : Retrofitting Decision Tree Classifiers using Kernel Density Estimation. Proceedings of the 12th ICML. Prieditis,A., Russel,S. (Eds.). Morgan Kaufmann.

Stone, C.J. (1977) : Consistent nonparametric regression. *The Annals of Statistics*, **5**, 595-645.

Torgo,L. Gama,J. (1997) : Search-based Class Discretization. To appear in Proceedings of ECML'97. Springer-Verlag.

Utgoff,P. (1989) : Incremental induction of decision trees. *Machine Learning*, **4**-2, 161-186. Kluwer Academic Publishers.

Watson, G.S. (1964) : Smooth Regression Analysis. *Sankhya: The Indian Journal of Statistics*, Series A, **26** : 359-372.

Weiss, S. and Indurkhya, N. (1993) : Rule-base Regression. In Proc. of the 13th IJCAI, p.1072-1078.

Weiss, S., Indurkhya, N. (1995) : Rule-based Machine Learning Methods for Functional Prediction. In Journal Of Artificial Intelligence Research (JAIR), **3**, p.383-403.

Wettschereck,D. (1994) : A study of distance-based machine learning algorithms. PhD thesis. Oregon State University.

Wettschereck,D., Aha,D.W., Mohri,T. (in press) : A review and empirical evaluation of feature weighting methods for a class of lazzy learning algorithms. Special issue on lazy learning. Aha, D. (Ed.). *Artificial Intelligence Review*.

Wolpert,D.H. (1992) : Stacked Generalization. In *Neural Networks*, **5**, (p.241-259).