Predicting the Density of Algae Communities using Local Regression Trees

Luís Torgo
LIACC – Machine Learning Group / FEP
University of Porto
R. Campo Alegre, 823
Phone: +351-2-6078830, Fax: +351-2-6003654
email: ltorgo@ncc.up.pt
WWW: http://www.ncc.up.pt/~ltorgo

ABSTRACT: This paper describes the application of local regression trees to an environmental regression task. This task was part of the 3rd Internation ERUDIT Competition. The technique described in this paper was announced as one of the three runner-up methods by the jury of the competition. We briefly describe RT, a system that is able to generate local regression trees. We emphasise the multi-strategy features of RT that we claim as being one of the causes for the obtained performance. We described the pre-processing steps that were taken in order to apply RT to the competition data, highlighting the weighed schema that was used to combine the predictions of the best RT variants. We conclude by reinforcing the idea that the combination of features of different data analysis techniques can be useful to obtain higher predictive accuracy.

KEYWORDS: Regression trees, local modelling, hybrid methods, local regression trees.

INTRODUCTION

This paper describes an application of local regression trees (Torgo, 1997a, 1997b, 1999) to an environmental data analysis task. This application was carried out in the context of the 3rd International Competition organised by ERUDIT in conjunction with the new Computational Intelligence and Learning Cluster. This cluster is a cooperation between four EC-funded Networks of Excellence : ERUDIT, EvoNet, MLnet and NEuroNet. This paper describes the use of system RT (Torgo, 1999) in the context of this competition. RT is a computer program that is able to obtain local regression trees. In this paper we briefly describe the main ideas behind local regression trees, and then focus on the steps taken to obtain a solution to this data analysis task.

The data analysis task concerns the environmental problem of determining the state of rivers and streams by monitoring and analysing certain measurable chemical concentrations with the goal of inferring the biological state of the river, namely the density of algae communities. This study is motivated by the increasing concern with the impact human activities are having on the environment. Identifying the key chemical control variables that influence the biological process associated with these algae has become a crucial task in order to reduce the impact of man activities. The data used in this 3rd ERUDIT international competition comes from such a study. Water quality samples were collected from various European rivers during one year. These samples were analysed for various chemical substances including: nitrogen in the form of nitrates, nitrites and ammonia, phosphate, pH, oxygen and chloride. At the same time, algae samples were collected to determine the distributions of the algae populations. The dynamics of algae communities is strongly influenced by the external chemical environment. Determining which chemical factors are influencing more this dynamics is important knowledge that can be used to control these populations. At the same time there is an economical factor motivating even more this analysis. In effect, the chemical analysis is cheap and easily automated. On the contrary, the biological part involves microscopic examination, requires trained manpower and is therefore both expensive and slow. The competition task consisted of predicting the frequency distribution of seven different algae on the basis of eight measured concentrations of chemical substances plus some additional information characterising the environment from which the sample was taken (season, river size and flow velocity).

RT is a regression analysis tool that can be seen as a kind of multi-strategy data analysis system. We will see that this characteristic is a consequence of the theory behind local regression trees. In effect, these models integrate regression trees (*e.g.* Breiman *et al.*,1984) with local modelling (*e.g.* Cleveland and Loader, 1995). The integration schema used in

RT allows several variants to be tried out in a given data set. Due to this flexibility, RT can cope with problems having different characteristics due to its ability of emulating techniques with different approximation biases. In this competition we have taken advantage of this facet of RT. We have treated the prediction of each of the seven algae frequencies as a different regression task. For each of the seven regression problems we have carried out a selection process with the aim of estimating which RT variant provided the best accuracy.

The following section provides a brief description of local regression trees. We describe the main components integrating these hybrid regression models and refer the method that was used to integrate them within RT. We them focus on the technical details concerning the application of RT to the task of predicting the density of algae communities.

LOCAL REGRESSION TREES

Local Regression Trees (Torgo, 1997a, 1997b, 1999) explore the possibility of improving the accuracy of regression trees by using smoother models at the tree leaves. These models can be regarded as a hybrid approach to multivariate regression integrating different solutions to this data analysis problem. Local regression (*e.g.* Cleveland and Loader, 1995) is a non-parametric statistical methodology that provides smooth modelling by not assuming any particular global form of the unknown regression function. On the contrary these models fit a functional form within the neighbourhood of the query points. These models are known to provide highly accurate predictions over a wide range of problems due to the absence of a "pre-defined" functional form. However, local regression techniques are also known by their computational cost, low interpretability and storage requirements. Regression trees (*e.g.* Breiman *et al.*,1984), on the other hand, obtain models through a recursive partitioning algorithm that insures high computational efficiency. Moreover, the resulting models are usually considered highly interpretable. By integrating regression trees with local modelling, not only we improve the accuracy of the trees, but also increase the computational efficiency and comprehensibility of local models.

In this section we provide a brief description of local regression trees. We start by describing both "standard" regression trees and local modelling. We then address the issue of how these two methodologies are integrated in our RT regression tool.

"STANDARD" REGRESSION TREES

A regression tree can be seen as a kind of additive model (Hastie & Tibshirani, 1990) of the form

$$m(\mathbf{x}) = \sum_{i=1}^{l} k_i \times I(\mathbf{x} \in D_i) \qquad (1)$$

where,

k_i are constants;

$I(.)$ is an indicator function returning 1 if its argument is true and 0 otherwise;

and D_i are disjoint partitions of the training data D such that $\bigcup_{i=1}^{l} D_i = D$ and $\bigcap_{i=1}^{l} D_i = \phi$.

Models of this type are sometimes called *piecewise constant regression models* as they partition the predictor space[1] χ in a set of mutually exclusive "regions" and fit a constant value within each region. An important aspect of tree-based regression models is that they provide a propositional logic representation of these regions in the form of a tree. Each path from the root of the tree to a leaf corresponds to such a region. Each inner node[2] of the tree is a logical test on a predictor variable[3]. In the particular case of binary trees there are two possible outcomes of the test, true or false. This means that associated to each partition D_i we have a path P_i consisting of a conjunction of logical tests on the predictor

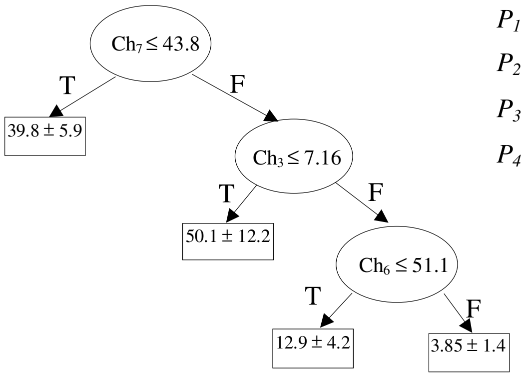
1 The multidimensional space formed by all input (or predictor) variables of a multivariate regression problem.

2 All nodes except the leaves.

3 Although work exists on multivariate tests (*e.g.* Breiman *et. al.* 1984; Murthy *et. al.*, 1994; Broadley & Utgoff, 1995; Gama, 1997).

variables. This symbolic representation of the regression surface is an important issue when one wants to have a better understanding of the problem under consideration.

For instance, consider the following small example of a regression tree obtained by applying RT to the competition data to obtain a model for one of the algae:



$P_1 \equiv \mathrm{Ch}_7 \leq 43.8$, with $k_1 = 39.8$

$P_2 \equiv \mathrm{Ch}_7 > 43.8 \wedge \mathrm{Ch}_3 \leq 7.16$, with $k_2 = 50.1$

$P_3 \equiv \mathrm{Ch}_7 > 43.8 \wedge \mathrm{Ch}_3 > 7.16 \wedge \mathrm{Ch}_6 \leq 51.1$, with $k_3 = 12.9$

$P_4 \equiv \mathrm{Ch}_7 > 43.8 \wedge \mathrm{Ch}_3 > 7.16 \wedge \mathrm{Ch}_6 > 51.1$, with $k_4 = 3.85$

Figure 1: An Example of a Regression Tree.

As there are four distinct paths from the root node to the leaves, this tree divides the input space in four different regions. The conjunction of the tests in each path can be regarded as a logical description of such regions, as shown above. This tree can be used both to make predictions of the density of this alga for future water samples. Moreover, it provides information on which variables influence more this density.

Regression trees are constructed using a recursive partitioning (RP) algorithm. This algorithm builds a tree by recursively splitting the training sample into smaller subsets. We give below a high level description of the algorithm. The RP algorithm receives as input a set of n data points, $D_t = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^{n_t}$, and if certain termination criteria are not met it generates a test node t, whose branches are obtained by applying the same algorithm with two subsets of the input data points. These subsets consist of the cases that logically entail the test in the node, $D_{t_L} = \{\langle \mathbf{x}_i, y_i \rangle \in D_t : \mathbf{x}_i \rightarrow s^*\}$, and the remaining cases, $D_{t_R} = \{\langle \mathbf{x}_i, y_i \rangle \in D_t : \mathbf{x}_i \nrightarrow s^*\}$. At each node the best split test is chosen according to some local criterion, which means that this is a greedy hill-climbing algorithm.

```
The Recursive Partitioning Algorithm.

Input  : A set of n data points, { < xi , yi > }, i = 1,...,n
Output : A regression tree

IF termination criterion THEN
       Create Leaf Node and assign it a Constant Value
       Return Leaf Node
ELSE
       Find Best Splitting Test s*
       Create Node t with s*
       Left_branch(t) = RegressionPartitioningAlgortihm({ <xi , yi> : xi → s* })
       Right_branch(t) = RegressionPartitioningAlgortihm({ <xi , yi> : xi ↛ s* })
       Return Node t
ENDIF
```

The algorithm has three main components:

- A way to select a split test (the splitting rule).
- A rule to determine when a tree node is terminal (termination criterion).
- A rule for assigning a value to each terminal node.

The answer to these three problems is related to the error criterion that is selected to guide the growth of the tree. The most common choice is the minimisation of the mean squared error (MSE). Using this criterion the constant that should be used in the leaves of the trees (terminal nodes) is the average goal variable value of the cases that "fall" in each leaf. Moreover, this criterion leads to the following rule for determining the best split of each test node (Torgo, 1999):

The best split s^* is the split that maximises the expression,

$$\frac{S_L^2}{n_{t_L}} + \frac{S_R^2}{n_{t_R}} \tag{2}$$

where, $S_L = \sum_{D_{t_L}} y_i$ and $S_R = \sum_{D_{t_R}} y_i$.

This rule leads to fast incremental updating algorithms that allow efficient search for the best test of each node (Torgo, 1999). Finally, the termination criterion is related to the issue of reliable error estimation. Methodologies like regression trees that have a "rich" hypothesis (model) language, incur the danger of overfitting the training data. This in turn leads to unnecessarily large models that usually have poor generalisation performance (*i.e.* lower predictive accuracy on new samples of the same problem). Overfitting avoidance in regression trees is usually achieved through a process of pruning unreliable branches of an overly large tree (*e.g.* Breiman *et al.*,1984). Many pruning techniques exist (see Torgo, 1999 for an overview), most of them relying on reliable estimation of predictive error of the trees (Torgo, 1998).

Regression trees achieve competitive predictive accuracy in a wide range of applications. Moreover, their computational efficiency allows dealing with problems with hundreds of variables and hundreds of thousands of cases. Still, regression trees provide a highly non-smooth function approximation with strong discontinuities[4].

LOCAL MODELLING

Local modelling[5] (Fan, 1995) belongs to a data analytic methodology whose basic idea behind consists of obtaining the prediction for a data point **x** by fitting a parametric function in the *neighbourhood* of **x**. This means that these methods are "locally parametric" as opposed to, for instance, least squares linear regression. Moreover, these methods do not produce a "visible" model of the data. Instead they make predictions based on local models generated on a query point basis.

According to Cleveland and Loader (1995) local regression traces back to the 19th century. These authors provide a historical survey of the work done since then. The modern work on local modelling starts in the 1950's with the *kernel methods* introduced within the probability density estimation setting (Rosenblatt,1956; Parzen,1962) and within the regression setting (Nadaraya,1964; Watson,1964). *Local polynomial regression* (Stone,1977; Cleveland,1979; Katkovnik,1979) is a generalisation of this early work on kernel regression. In effect, kernel regression amounts to fitting a polynomial of degree zero (a constant) in a neighbourhood. Summarising, we can state the general goal of local regression as trying to fit a polynomial of degree p around a query point (or test case) **q** using the training data in its neighbourhood. This includes the various available settings like kernel regression (p=0), local linear regression (p=1), *etc*[6].

Most studies on local modelling are done for the case of one univariate problems. Still, the framework is applicable to the multivariate case and has been used with success in some domains (Atkeson *et al.*,1997; Moore *et al.*,1997). However, several authors alert for the "danger" of applying these methods in higher input space dimensions[7] (Hardle, 1990; Hastie & Tibshirani,1990). The problem is that with high number of variables the training cases are so sparse that the notion of local neighbourhood can hardly be seen as local. Another drawback of local modelling is the complete lack of interpretability of the models. No interpretable model of the training data is obtained. With some simulation work one may obtain a graphical picture of the approximation provided by local regression models, but this is only possible with low number of input variables.

---

[4] This later characteristics can both be seen as advantageous or disadvantageous, depending on the application.

[5] Also known as non-parametric smoothing and local regression.

[6] A further generalisation of this set-up consists of using polynomial mixing (Cleveland & Loader,1995), where p can take non-integer values.

[7] The so called "curse of dimensionality" (Bellman, 1961).

In spite of being considered a non-parametric regression technique, local modelling does have several "parameters" that must be tuned in order to obtain good predictive results. One of the most important is the notion of neighbourhood. Given a query point **q** we need to decide which training cases will be used to fit a local polynomial around the query point. This involves defining a distance metric over the multidimensional space defined by the input variables. With this metric we can specify a distance function that allows finding the nearest training cases of any query point. Still, many open issues remain unspecified. Namely, weighing of the variables within the distance calculation can be crucial in domains with less relevant variables. Moreover, we need to specify how many training cases will enter the local fit (usually known as the bandwidth selection problem). Even after having a bandwidth size specification, we need to weigh the contributions of the training cases within the bandwidth. Nearer points should contribute more into the local fit. This is usually accomplished through a weighing function that takes the distance to the query point into account (known as the kernel function). The correct tuning of all these modelling "parameters" can be crucial for a successful use of local modelling.

Local modelling provides smooth function approximation with wide applicability through correct tuning of the distance function parameters. Still, this is a computational intensive method that does not produce a comprehensible model of the data. Moreover, these methods have difficulties in dealing with domains with strong discontinuities in the function being approximated.

INTEGRATING REGRESSION TREES WITH LOCAL MODELLING

In this section we describe how we propose to overcome some of the limitations of both regression trees and local modelling through an integration of both methods that result in what we refer to as local regression trees. The main goals of this integration can be stated as follows:

- Improve standard regression trees smoothness, leading to superior predictive accuracy.
- Improve local modelling in the following aspects:
  - Computational efficiency.
  - Generation of models that are comprehensible to human users.
  - Capability to deal with domains with strong discontinuities.

In our study of local regression trees we have considered the integration of the following local modelling techniques:

- Kernel models (Watson, 1964; Nadaraya, 1964).
  These models amount to fitting a polynomial of degree zero (*i.e.* a constant) in the local neighbourhood.
- Local linear polynomial (Stone,1977; Cleveland,1979; Katkovnik,1979).
  Where we fit a linear polynomial within the neighbourhood.
- Semi-parametric models or partial linear models (Spiegelman, 1976; Hardle, 1990).
  Consisting of fitting a standard least squares linear model on all data plus a kernel model component on the residuals of the linear polynomial that acts as a local "correction" of the global linearity assumption of the polynomial.

The main decisions concerning the integration of local models with regression trees are how, and when to perform it. Three main alternatives exist:

- Assume the use of local models in the leaves during all tree induction (*i.e.* growth and pruning phases).
- Grow a standard regression tree and use local models only during the pruning stage.
- Grow and prune a standard regression tree. Use local models only in prediction tasks.

The first of these alternatives is more consistent from a theoretical point of view as the choice of the best split depends on the models at the leaves. This is the approach followed in RETIS (Karalic, 1992), which integrates global least squares linear polynomials in the tree leaves. However, obtaining such models is a computationally demanding task. For each trial split the left and right child models need to be obtained and their error calculated. Even with a simple model like the average, without an efficient incremental algorithm the evaluation of all candidate splits is too heavy. This means that if more complex models are to be used, like linear polynomials, this task is practically unfeasible for large domains[8]. The experiments described by Karalic (1992) used data sets with few hundred cases. The author does not provide any results concerning the computation penalty of using linear models instead of averages. Still, we claim that

[8] Particularly with large number of continuous variables.

when using more complex models like kernel regression this approach is not feasible if we want to achieve a reasonable computation time.

The second alternative is to introduce the more complex models only during the pruning stage. The tree is grown (*i.e.* the splits are chosen) assuming averages in the leaves. Only during the pruning stage we consider that the leaves will contain more complex models, which entails obtaining them for each node of the grown tree. Notice that this is much more efficient than the first alternative mentioned before which involved obtaining the models for each trial split considered during the tree growth. This is the approach followed in M5 (Quinlan, 1992). This system also uses global least squares linear polynomials in the tree leaves but these models are only added during the pruning stage. We have access to a version of M5[9] and we have confirmed that this is a computationally feasible solution even for large problems.

In our integration of regression trees with local modelling we have followed the third alternative. In this approach the learning process is separated from prediction tasks. We generate the regression trees using the "standard" methodology described previously. If we want to use the learned tree to make predictions for a set of unseen test cases, we can choose which model should be used in the tree leaves. These can include complex models like kernels or local linear polynomials. Using this approach we only have to fit as many models as there are leaves in the final pruned tree. The main advantage of this approach is its computational efficiency. However, it also allows trying several alternative models without having to re-learn the tree. Using this approach the initial tree can be regarded as a kind of rough approximation of the regression surface which is comprehensible to the human user. On top of this rough surface we may fit smoother models for each data partition generated in the leaves of the regression tree so as to increase the predictive accuracy.

In Torgo (1999) a large experimental evaluation of local regression trees over a wide range of problems was carried out. This large set of experimental comparisons confirmed that local regression trees are significantly more accurate than the "standard" regression trees. However, these new regression models are computationally more demanding and less comprehensible than standard regression trees. We have also observed that local regression trees could overcome some of the limitations of local modelling techniques, particularly their lack of comprehensibility and rather high processing time. These experiments have shown that local regression trees are significantly faster than local modelling techniques. Moreover, through the integration within a tree-based structure we obtain a comprehensible insight of the approximation of these models. Finally, we have also observed that the modelling bias resulting from combining local models and partition-based approaches improves the accuracy of local models in domains where there are strong discontinuities in the regression surface.

PREDICTING THE DENSITY OF ALGAE COMUNITIES

In this section we describe the steps followed in the application of RT in the context of the 3rd International ERUDIT Competition. As we have mentioned this competition consisted of a data analysis task concerning the environmental problem of determining the state of rivers and streams by monitoring and analysing certain measurable chemical concentrations. The goal of the task was to obtain a model that should allow making predictions concerning the frequency distribution of seven algae. With this purpose the organisers have delivered two data files to each competitor. The first contained the training data consisting of 200 samples described by 11 input variable plus the observed algae frequency distributions. This training file was presented as a matrix with 200 rows and 18 (11+7) columns. From the 11 input variables (labelled as *A*,…,*K*) three were categorical, namely the season, the river size and the fluid velocity. All remaining input variables were numeric. Several river samples contained unknown input variable values (signalled by "XXXX"). The competitors were also given a second file containing the test data consisting of another 140 river samples. The competitors were only given the values of the 11 input variables of these test samples. There were also unknown variable values in the test cases. Thus the test data set corresponded to a matrix of 140 rows and 11 columns of values. The goal of the competition was to build a model based on the training data that could provide predictions of the frequency distributions of the 7 algae (labelled as *a*,…,*g*) for the 140 test samples, whose true values were only known to the organisers. The competitors should present their results as a 140×7 matrix of predictions. The solutions proposed by the competitors were evaluated by the sum of the total squared errors between their predictions and the true values.

---

[9] Version 5.1

PROBLEM SOLUTION USING RT

This section briefly describes the steps taken to produce a matrix of predictions based on models obtained with RT.

The first step that we carried out was to divide the given training data into seven different training files. The division was done for each of the 7 different algae. This means that we dealt with this problem as seven different regression tasks. For each of these seven training sets all input variable values were the same, and only the target variables changed.

The second step of our proposed solution consisted of filling-in the unknown variable values. For each training case with an unknown variable value we have searched for the 10 most similar cases. This similarity was asserted using an Euclidean distance metric. The median value of the variable of these 10 cases was used to fill in the unknown variable value.

The methodology used in RT to integrate local models with regression trees enables this system to emulate a large number of regression techniques through simple parameter settings. In effect, as the predictions of the local models used in the leaves are obtained completely independently of the tree growth phase we can very easily try different variants of local regression. Moreover, we can even use these models on all training data that would correspond to the original local models. This later alternative is easily accomplished by pruning too much the initial tree until a single leaf is reached. This means that RT can emulate several regression techniques like standard regression trees, local regression trees and local modelling techniques. Moreover, several of the implemented local modelling techniques can be used to emulate other regression models. For instance, a local linear polynomial can behave like a standard least squares linear regression model through appropriate parameter tuning[10]. In resume, RT hybrid character can be used to obtain the following types of models:

- Standard least squares (LS) regression trees.
- Least absolute deviation (LAD) regression trees.
- Regression trees (LS or LAD) with least squares linear polynomials in the leaves.
- Regression trees (LS or LAD) with kernel models in the leaves.
- Regression trees (LS or LAD) with *k* nearest neighbors in the leaves.
- Regression trees (LS or LAD) with least squares local linear polynomials in the leaves.
- Regression trees (LS or LAD) with least squares partial linear polynomials (models that integrate both parametric and non-parametric components) in the leaves.
- Least squares linear polynomials (with or without backward elimination to simplify the models).
- Kernel models.
- *K* nearest neighbors.
- Local linear polynomials
- Partial linear polynomials.

Having so many variants in RT the next step of our analysis was to try to find out which were the most promising techniques for each of the seven regression tasks. With this purpose we have carried out the following experiment. We have selected a large set of candidate variants of RT. For each of the seven training sets we have carried out the following experiment with the goal of estimating the predictive accuracy (measured by the Mean Squared Error) of each variant. The MSE was estimated using a 10-fold Cross Validation experiment. Each training set was randomly divided into 10 approximately equally sized folds. For each fold, a model was built with the remaining nine and its prediction error calculated in the fold. This process was repeated for the 10 different folds and the prediction errors were averaged. This 10-fold Cross Validation process was repeated 10 times with 10 different random permutations of each training set. The final score of each variant was obtained by averaging over these 10 repetitions. For each of the seven algae the most promising variants of RT were collected together with their estimated prediction error. This model selection stage immediately revealed that the seven regression tasks posed quite different challenges. In effect, there was a large diversity of techniques that were estimated as the most promising depending of the alga in consideration.

For each of the seven algae training sets we have used the respective most promising RT variants to obtain a model. These models were them applied in the given 140 test cases and their predictions collected. The final predictions that

---

[10] In this example it is enough to use an infinite bandwidth (meaning that all training points contribute to obtain the model) and a uniform weighing function (which gives equal weights to all training points).

were submitted as our solutions were obtained by a weighed average of these predictions. To better illustrate this weighing schema, imagine that for alga *a* the most promising RT variants and their estimated predictive accuracy were:

$$M_a, Err_{M_a}; M_b, Err_{M_b}; M_c, Err_{M_c}; M_d, Err_{M_d}$$

The final prediction for a test case $\mathbf{x}_i$ would be calculated as:

$$y_i' = \frac{M_a'(\mathbf{x}_i) \times Err_{M_a} + M_b'(\mathbf{x}_i) \times Err_{M_b} + M_c'(\mathbf{x}_i) \times Err_{M_c} + M_d'(\mathbf{x}_i) \times Err_{M_d}}{Err_{M_a} + Err_{M_b} + Err_{M_c} + Err_{M_d}}$$

where,

$\mathbf{x}_i$ is a test case;

y_i' is the prediction for case $\mathbf{x}_i$;

and $M_a'(\mathbf{x}_i)$ is the prediction of model M_a for case $\mathbf{x}_i$.

DISCUSSION

The solution obtained through the process described before was declared by the jury of the ERUDIT competition as one of the three runner-ups winners. After the announcement of the results the organisation provided the true values of the test samples and this enabled us to analyse in more detail the performance of our proposed data analysis technique. The total sum of squared error of our solution was 85279.97, which corresponds to a MSE of 87.020. At the time of writing the scores of the other competitors are not known so it is a bit hard to interpret the value of these numbers. Still, we have collected other statistics that could provide further insights on the quality of the solution. We have calculated the mean absolute deviation (MAD) of the predictions of RT[11]. The following graph shows the minimun MAD, maximum MAD and an interval (represented by a box) between MAD±Standard Error, for all seven algae.

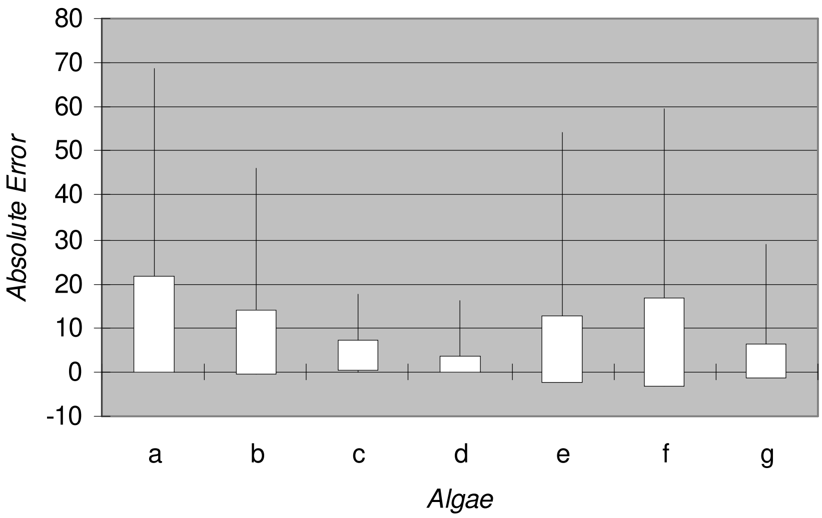


Figure 2: The Scores of our Solution in term of Absolute Error.

From this figure it is clear that the best results are obtained for predicting the frequency of algae *c*, *d* and *g*, while the worst predictions occur with algae *a* and *f*. Still, the overall Mean Absolute Error is 5.365 which is a relatively low value[12].

---

[11] This statistic is more meaningful as it is in the same scale as the goal variable.

[12] The values of the algae distributions range from 0 to 100 (although the maximum value in the training samples was 89.8 for alga *a*).

With the true goal variable values we were able to confirm that the ranking of RT variants obtained through repeated 10-fold Cross Validation was most of the times correct.

We have also carried out some initial experiments to try to identify the main causes for the good performance of our solution. These experiments clearly show that the availability of different variants was highly benefict because worse results are obtained when trying the same variant over the seven regression tasks. As an example, while our multi-strategy weighed solution achieved an overall MSE of 87.020 on all seven tasks, using the best RT variant (Torgo, 1999), which consists of using local regression trees with partial linear models in the leaves, we obtained a MSE of 93.184. Moreover, the weighing schema does also bring some accuracy gains when compared to the alternative of always selecting the prediction of the best estimated variant for each task[13]. We think that using more sophisticated model combination methodologies like boosting (Freund, 1995) or bagging (Breiman,1996) should provide even more interesting accuracy results.

CONCLUSIONS

We have described the application of a regression analysis tool named RT to an environmental data used in the 3rd International ERUDIT Competition. RT implements a new regression methodology called local regression trees. This technique can be regarded as a combination of two existing methodologies, regression trees and local modelling. We have described the integration schema used in RT to obtain local regression trees. We have shown how this schema is important in obtaining a tool that is able to approximate quite different regression surfaces. The results obtained by RT in the competition provide further confidence on the correctness of our integration schema.

The methodology used to obtain a solution for the competition was based on the assumption that using different variants of RT on each of the seven regression tasks of the competition provided better results than always using the same regression methodology. Our experiments and the results of the competition confirmed this hypothesis. Moreover, averaging over the best variants did also improve the accuracy of the resulting predictions. This application provides further evidence towards the advantages of systems incorporating multiple data analysis techniques, and also towards the accuracy advantages of combining predictions of different models.

REFERENCES

Atkeson,C.G., Moore,A.W., Schaal,S., 1997, “Locally Weighted Learning”, *Artificial Intelligence Review*, **11**, 11-73. Special issue on lazy learning, Aha, D. (Ed.).

Bellman,R. , 1961, *Adaptive Control Processes*. Princeton University Press.

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

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

Broadley,C., Utgoff,P., 1995, “Multivariate decision trees”, *Machine Learning*, **19**, 45-77. Kluwer Academic Publishers.

Cleveland, W., 1979, “Robust locally weighted regression and smoothing scatterplots”, *Journal of the American Statistical Association*, **74**, 829-836.

Cleveland,W., Loader,C., 1995, “Smoothing by Local Regression: Principles and Methods (with discussion)”, in *Computational Statistics*.

Fan,J., 1995, “Local Modelling”, in *Encyclopidea of Statistical Science*.

Freund,Y., 1995, “Boosting a weak learning algorithm by majority”, in *Information and Computation*, **121** (2), 256-285.

Gama,J., 1997, “Probabilistic linear tree”, in *Proceedings of the 14th International Conference on Machine Learning*, Fisher,D. (ed.). Morgan Kaufmann.

Hardle,W. , 1990, *Applied Nonparametric Regression*. Cambridge University Press.

[13] While the weighed average obtained a total sum of squared error of 85279.97 (MSE=87.020), using the predictions of the best variant we get a total of 87510.21 (MSE=89.296).

Hastie,T., Tibshirani,R., 1990, *Generalized Additive Models*. Chapman & Hall.

Karalic,A., 1992, "Employing Linear Regression in Regression Tree Leaves", in Proceedings of ECAI-92. Wiley & Sons.

Katkovnik,V., 1979, "Linear an nonlinear methods of nonparametric regression analysis", *Soviet Automatic Control*, **5**, 25-34.

Moore,A. Schneider,J., Deng,K., 1997, "Efficient Locally Weighted Polynomial Regression Predictions", in *Proceedings of the 14th International Conference on Machine Learning (ICML-97)*, Fisher,D. (ed.). Morgan Kaufmann Publishers.

Murthy,S., Kasif,S., Salzberg,S., 1994, "A system for induction of oblique decision trees", in *Journal of Artificial Intelligence Research*, **2**, 1-33.

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

Parzen,E. , 1962, "On estimation of a probability density function and mode", *Annals Mathematical Statistics*, **33**, 1065-1076.

Quinlan, J., 1992, "Learning with Continuous Classes", in *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*. Singapore: World Scientific, 1992.

Rosenblatt,M., 1956, "Remarks on some nonparametric estimates of a density function", *Annals Mathematical Statistics*, **27**, 832-837.

Spiegelman,C., 1976, "Two techniques for estimating treatment effects in the presence of hidden variables: adaptive regression and a solution to Reiersol's problem". Ph.D. Thesis. Dept. of Mathematics, Northwestern University.

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

Torgo,L., 1997a, "Functional models for Regression Tree Leaves", in Proceedings of the International Conference on Machine Learning (ICML-97), Fisher,D. (ed.). Morgan Kaufmann Publishers. Also available in http://www.ncc.up.pt/~ltorgo/Papers/list_pub.html.

Torgo,L., 1997b, "Kernel Regression Trees", in Proceedings of the poster papers of the European Conference on Machine Learning (ECML-97), Internal Report of Faculty of Informatics and Statistics, University of Economics, Prague, ISBN:80-7079-368-6. Also available in http://www.ncc.up.pt/~ltorgo/Papers/list_pub.html.

Torgo,L., 1998, "A Comparative Study of Reliable Error Estimators for Pruning Regression Trees", in Proceeding of the Iberoamerican Conference on Artificial Intelligence, Coelho,H. (ed.), Springer-Verlag. Also available in http://www.ncc.up.pt/~ltorgo/Papers/list_pub.html.

Torgo,L., 1999, "Inductive Learning of Tree-based Regression Models", Ph.D. Thesis, to be presented at the Department of Computer Science, Faculty of Sciences, University of Porto. Soon available in http://www.ncc.up.pt/~ltorgo/thesis.html.

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