Boosting the Accuracy of Hedonic Pricing Models

Michiel van Wezel * Martijn Kagie Rob Potharst

Econometric Institute, Faculty of Economics, Erasmus University
P.O. Box 1738, 3000 DR, Rotterdam, The Netherlands,

Econometric Institute Report EI 2005-50

December 2, 2005

Abstract

Hedonic pricing models attempt to model a relationship between object attributes and the object's price. Traditional hedonic pricing models are often parametric models that suffer from misspecification. In this paper we create these models by means of boosted CART models. The method is explained in detail and applied to various datasets. Empirically, we find substantial reduction of errors on out-of-sample data for two out of three datasets compared with a stepwise linear regression model. We interpret the boosted models by partial dependence plots and relative importance plots. This reveals some interesting nonlinearities and differences in attribute importance across the model types.

Keywords: Conjoint Analysis, Data Mining, Gradient Boosting, Ensemble Learning, Hedonic Pricing, Marketing, Pricing.

1 Introduction

Hedonic pricing hypothesizes that each good is can be looked upon as a bundle of attributes $\mathbf{x}$ and that a functional relationship $p = F^*(\mathbf{x})$ exists between these attributes and the price p of a good. Hedonic pricing models are useful to assess the market value of goods before they are traded or prices of goods that are not explicitly traded. (An example of the latter: Property tax is often based upon the estimated market value of a property.) The development of hedonic pricing theory is generally attributed to Lancaster (1966), Griliches (1971b) (1971a) and Rosen (1974).

In practice the hedonic price function $F^*(\mathbf{x})$ is estimated by a model $F(\mathbf{x})$ based on a set $\{(\mathbf{x}_i, p_i)\}, i = 1, \ldots N$ of historical data observations of attribute vectors and prices. Traditional hedonic pricing models are typically linear or box-cox type models. These parametric models have the appealing properties that their parameters are well-interpretable and easy to estimate. Unfortunately, these models often suffer from misspecification: They impose a functional form on the hedonic model that does not allow a good fit to the data. Several authors, e.g., Anglin and Gençay (1996) Pace (1998) Gençay and Yang (1996) Bin (2004), have compared parametric hedonic price models with more flexible semi- and non-parametric hedonic price models, usually with substantial improvement in out-of-sample prediction performance.

In this paper we use a relatively new non-parametric technique, gradient boosting in combination with regression trees (Friedman 2001), for the hedonic pricing problem. The performance of the boosted tree models is compared with the performance of traditional parametric models using three benchmark datasets. Special attention is paid to interpretation of the non-parametric models by means of partial dependence and relative importance plots.

An advantage of (boosted) regression trees (and many other non-parametric model classes) over parametric models is that they allow for arbitrary interactions between product attributes, whereas

*Corresponding author. Phone: +31 10 4081341. Fax: +31 10 4089167. Email: mvanwezel@few.eur.nl

the parametric models allow no interactions at all, or limited interactions that are explicitly input to the model in the best case. An example of such an interaction effect is the following. Suppose we model car prices, and two of the attributes are *body shape* and *color*. For a *sports car*, a *red* color may add to its value and be preferred over *green*, whereas, for an *off-road* vehicle, *green* will probably give a higher market value than *red*. Thus, there is an interaction between *body shape* and *color*. Interactions between more than two attributes are also possible. Decision trees are good at discovering interactions and utilizing them for predictions.

Another advantage of boosted regression trees is their ability to model nonlinearities in the data. (Of course, interactions effects are non-linear as well.) An example of a nonlinearity in the case of car prices may be the dependence of *price* on the *trunk size*. For small trunks, an increase in *trunk size* may yield a lot of utility, but for large trunks a saturation effect occurs.

Many applications of hedonic pricing models have been in the realm of real estate appraisal (Harrison and Rubinfeld 1978; Gilley and Pace 1995; Gençay and Yang 1996; Anglin and Gençay 1996; Pace 1998; Bin 2004), but the method has also been applied to a number of other product categories, e.g., ball-points (Tomkovick and Dobie 1995) and digital cameras (Miyamoto and Tsubaki 2002).

Hedonic pricing models can also be put to fruitful use in conjoint analysis, one of the most widely applied quantitative marketing methods (see e.g., Lilien and Rangaswamy 2002; Green, Krieger, and Wind 2001). The aim of conjoint analysis is to predict the utility of a new product. Similar to hedonic pricing, a product is viewed as a bundle of attributes and a mapping between these attributes and the utility represented by the product is constructed. This mapping is called the part-worth function in conjoint analysis. The main difference between conjoint analysis and traditional hedonic pricing is that in conjoint analysis the analyst decides what products (i.e., which combinations of attribute values) to include in the study and these products are not actually traded, but respondents are asked to give their utilities. In summary, a part-worth function is constructed on specially collected respondent data, whereas a hedonic price model is constructed on data obtained from a real market, but both mappings can be used to obtain an estimate of a product that is not explicitly traded or rated. There have been sporadic applications of hedonic pricing in new product development, an example is described in Tomkovick and Dobie (1995). Another related article in the marketing area is given by Ofek and Srinivasan (2002).

The remainder of this paper is structured as follows. The next section (Section 2) describes the model types we use in this paper: stepwise linear regression, regression trees and boosted regression trees. Section 3 describes the datasets we used. Experiments and results are described in Section 4. Section 5 shows how to interpret boosted decision trees and gives the most striking insight that result from this interpretation. Finally, Section 6 gives a summary, some conclusions and an outlook.

2 Models

This section describes the model types we use in this paper. The first model is stepwise linear regression, which is included as a benchmark model.

2.1 Stepwise linear regression

Linear regression is a frequently used model in statistics. SLR is an extension of ordinary linear regression where attributes are added to the model in a greedy fashion – The model starts with one variable and after each step the algorithm selects from the remaining attributes the one which yields the largest reduction in the residual variance of the dependent variable, unless its contribution to the total error reduction is smaller than a specified threshold. Similarly, the algorithm evaluates after each step whether the contribution of any variable already included falls below a specified threshold, in which case it is dropped from the regression. SLR has the advantage that it is less prone to over-fitting than ordinary LR in data-sparse situations because fewer predictor variables are used.

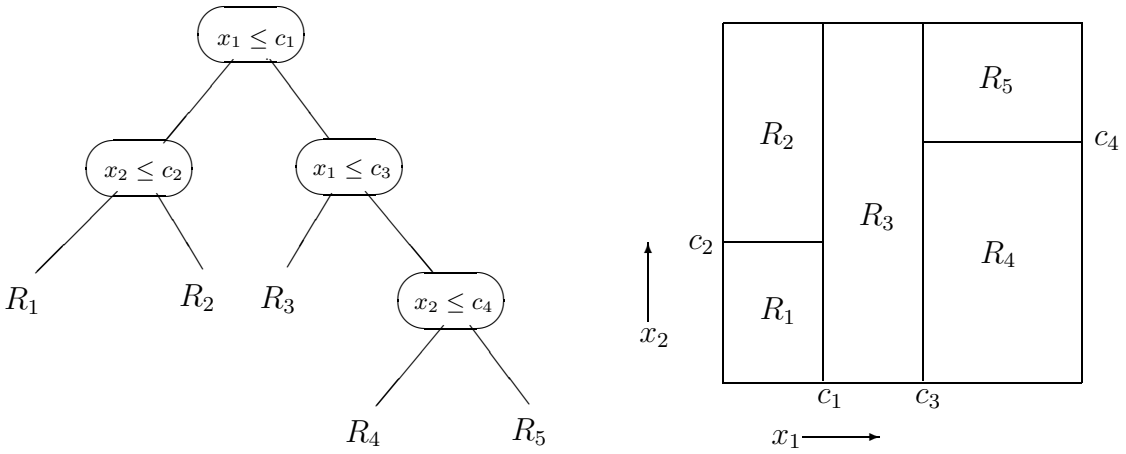

Figure 1: A regression tree (l), and a partitioning of the feature space. $F(\mathbf{x})$ is modeled by a different constant in each region $R_1, \ldots, R_5$. For instance, $F(\mathbf{x}|\mathbf{x} \in R_1) = 0.1, F(\mathbf{x}|\mathbf{x} \in R_2) = 0.4$, and so on.

2.2 Regression trees

A regression tree partitions the attribute space into a number of non-overlapping rectangles, and then fits a constant to each of them. For instance, if we have two attributes x_1 and x_2, the tree of Figure 1 splits the attribute space into five non-overlapping rectangular regions $R_1, \ldots, R_5$. Here, a binary split of the form $x_i \leq c$ splits a region into two subregions, one with $x_i \leq c$ (go left in the tree) and one with $x_i > c$ (go right in the tree). Thus, a regression tree θ splits the attribute space recursively by binary splits into a number of rectangular regions $R_1, \ldots, R_m$. Well-known algorithms for generating decision trees on the basis of a data set are C4.5 (Quinlan 1993) and CART (Breiman, Friedman, Olshen, and Stone 1984). Both algorithms can also be used for classification, where a categorical variable is modeled. In this paper we use the regression version of CART in our experiments.

Decision trees are usually built in two phases. The first phase is a *growing* phase, the second phase is a *pruning* phase. In the growing phase, the tree is grown until error reduction on the training set is no longer possible or a predetermined threshold has been reached. The resulting model usually overfits the data, and this is countered in a pruning phase, where the tree is shrunk until the error on a hold-out sample, the *pruning set*, is minimal.

As stated in the introduction, decision trees are well suited for modeling interactions and non-linearities. A drawback of decision trees is their instability – The implemented model depends heavily on the dataset used for model creation, and a small change in the data may have large consequences for the model. Ensemble methods, such as bagging (Breiman 1996) and boosting, have a stabilizing effect by averaging over a number of decision trees. We consider boosting below.

2.3 Boosting & Boosted regression trees

Boosting (Freund and Schapire 1996) is a relatively new 'learning method', which has received great praise in the machine learning, computer science and statistics communities. The method was referred to as 'one of the most powerful learning methods introduced in the last ten years' in Hastie, Tibshirani, and Friedman (2001). Boosting is sometimes called arcing in the literature (Breiman 1998). Boosting has shown to work well on a plethora of test problems and is often seemingly resistant to over-fitting.

The general idea of boosting is to create a sequence of models, where each model focuses on the patterns for which a large error was made by its predecessor(s). The predictions of these models are subsequently combined to form the final predictions, either by weighted (as in (Freund

and Schapire 1996)) or unweighted (as in Breiman (1998)) combination. In the case of regression problems this combination takes place by simple averaging, in the case of classification problems it is often done by 'voting'.

The original boosting algorithm, AdaBoost, is limited to data modeling problems with a categorical target (classification problems) and zero-one loss – an example is either modeled correctly or not. Various authors have suggested extensions to other loss functions. One of these extensions is the 'LS_Boost' algorithm due to Friedman (2001), a boosting algorithm for regressors with squared error loss, which we use in this paper. LS_Boost is an instantiation of Gradient_Boost, a boosting algorithm for general loss functions. We briefly describe LS_Boost below. For a more elaborate description of LS_Boost and Gradient_Boost the reader is referred to Friedman's paper (Friedman 2001).

Traditional (parametric) statistical models $F(\mathbf{x}|\theta)$ for relating a target y to inputs $\mathbf{x}$ are fit by minimizing an objective function as a function of model parameters θ:

$$F_{\text{best}}(\mathbf{x}) = F(\mathbf{x}|\ \theta_{\text{best}}), \qquad \theta_{\text{best}} = \arg\min_{\theta} E_{\mathbf{x},y}[L(F(\mathbf{x}|\theta), y)]$$

The appropriate loss criterion L is determined by the data modeling problem – It is often the negative logarithm of a likelihood function. The best parameter vector is often found in a number of steps using an optimization algorithm.

Boosting also finds an approximation of the target function in a number of steps, not by improving model parameters, but by adding a new model to the existing model in each step. Starting of with initial guess F_0, the estimated function is refined in the course of M steps by adding functions $s_1, \ldots, s_m$:

$$F_{\text{best}}(\mathbf{x}) = F_0(\mathbf{x}) + \sum_{m=1}^{M} s_m(\mathbf{x}).$$

Of course, we want to end up with F such that

$$F_{\text{best}} = \arg\min_{F} E_{\mathbf{x},y}[L(F(\mathbf{x}), y)]. \tag{1}$$

Gradient_Boost finds this minimum by taking M steps along the negative gradient direction of the objective function $E_{\mathbf{x},y}[L(F(\mathbf{x}), y)]$. A large enough number of such steps would certainly take us to a local minimum. At the m-th step the gradient at $\mathbf{x}$ is given by

$$g_m(\mathbf{x}) = \left[\frac{\partial E_y[L(F(\mathbf{x}), y)|\mathbf{x}]}{\partial F(\mathbf{x})}\right]_{F=F_{m-1}} = E_y\left[\frac{\partial L(F(\mathbf{x}), y)}{\partial F(\mathbf{x})}|\mathbf{x}\right]_{F=F_{m-1}} \tag{2}$$

where $F_{m-1}(\mathbf{x}) = F_0(\mathbf{x}) + \sum_{i=1}^{m-1} s_i(\mathbf{x})$. In practice when computing this gradient at data point $(\mathbf{x}_i, y_i)$ the expectation over y is dropped:

$$g_m(\mathbf{x}_i) = \left[\frac{\partial L(F(\mathbf{x}_i), y_i)}{\partial F(\mathbf{x}_i)}|\mathbf{x}_i\right]_{F=F_{m-1}}.$$

Ideally, we would like to take steps $-\alpha g_m(\mathbf{x})$ along the negative direction of this gradient, where α is a step size parameter. However, when using a finite dataset this gradient is only defined at the data points. Therefore Gradient_Boost approximates the gradient step as well as possible using a certain model class f. That is, the gradient is approximated using the closest possible model $f_m(\mathbf{x}) = f(\mathbf{x}|\mathbf{a}_m)$ from a given family (e.g., trees, neural networks) with parameters $\mathbf{a}$ (e.g., split points, weights):

$$\mathbf{a}_m = \arg\min_{\mathbf{a}} \sum_{i=1}^{N} \{g_m(\mathbf{x}_i) - f(\mathbf{x}_i|\mathbf{a})\}^2, \tag{3}$$

and this approximation is used in the m-th step of the algorithm:

$$s_m(\mathbf{x}) = -\alpha f_m(\mathbf{x}).$$

The models f are often called the 'base learners' in the machine learning literature. In principle, the gradient boosting algorithm can be used with any differentiable loss function L and any base learner f for which least squares minimization is possible. A linear base model is useless however, because a linear combination of linear models is still a linear model. In this paper we use CART as the base learner and squared error as the loss function.

LS_Boost starts by initializing $F_0(\mathbf{x}) = \bar{y}$. Subsequently the algorithm performs M steps in each of which a new base learner, in our case a CART model, is fitted. In each iteration Equation (2) is used to to find 'targets' or 'working responses' $\tilde{y}_i = g_m(\mathbf{x}_i)$ for minimization problem (3). Since the loss function is quadratic

$$\tilde{y}_i = -\frac{\partial((y_i - F_{m-1}(\mathbf{x}_i))^2/2)}{\partial F_{m-1}(\mathbf{x}_i)} = y_i - F_{m-1}(\mathbf{x}_i).$$

Next, a new model f_m is fitted and the combined model F_{m-1} is updated with s_m. The algorithm is summarized in Algorithm 1.

Input : Dataset with instances $\{\mathbf{x}_i; y_i\}_1^N$
Number of cycles M
Step size parameter α
Output: Model $F(\mathbf{x})$
$F_0(\mathbf{x}) = \bar{y}$
for $m = 1$ to M **do**
  $\tilde{y}_i = y_i - F_{m-1}(\mathbf{x}_i), i = 1, \ldots, N$
  train f_m using $\{\mathbf{x}_i; \tilde{y}_i\}_1^N$
  $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \alpha f_m(\mathbf{x})$
end

Algorithm 1: LS_Boost

3 Data

We used three datasets in our experiments: *Automobile, BostonHousing* and *WindsorEssexHousing*. The first two datasets are donwloadable from the UCI Machine Learning Repository (Hettich 2004), the third one was also used in the paper by Anglin and Gençay (1996) and can be obtained from the ftp-archive of the Journal of Applied Econometrics (`http://qed.econ.queensu.ca/jae/`). Below, we briefly describe these datasets:

Automobile: The automobile dataset describes 205 cars models imported into the U.S. in 1985. The cars are described in terms of specifications of various characteristics, summarized in Table 1, and, of course the aim of our analysis is to find a relationship between the target attribute, *Price* and these characteristics. Instances with a missing target attribute were removed from the dataset, leading to a data set size of 201.

We made some minor modifications to the original dataset: two attributes related to insurance insurance risk (i.e. *Symbolizing* and *Normalized-Losses*) were removed from the dataset, and *Make* was replaced by a categorical value indicating the country of origin. This operation maps the 22 original values of *Make* onto 7 countries of origin, preventing the models from using *Make* as an important indicator.

BostonHousing: The Boston housing dataset was created in 1978 by Harrison to study the effects of air pollution on housing prices (Harrison and Rubinfeld 1978), and has become a well-known dataset in the data analysis community. The 506 examples in the dataset describe suburbs of Boston, the target attribute is the median value of owner-occupied houses in the suburb. The attributes in the dataset are described in Table 2.

Categorical	
Name	Levels
Country	France (13) Germany (40) Italy (3) Japan (99) USA (20) UK (3) Sweden (17)
FuelType	diesel (20) gas (185)
Aspiration	standard (168) turbo (37)
BodyStyle	convertible (6) hardtop (8) hatchback (70) sedan (96) wagon (25)
Drive-wheels	4wd (9) fwd (120) rwd (76)
EngineLocation	front (202) rear (3)
EngineType	dohc (12) dohcv (1) l (12) ohc (148) ohcf (15) ohcv (13) rotor (4)
FuelSystem	1bbl (11) 2bbl (66) 4bbl (3) idi (20) mfi (1) mpfi (94) spdi (9) spfi (1)
Numerical	
Name	Avg Stdv Min Max (Missing)
Bore	3.33 0.27 2.54 3.94 (4)
Stroke	3.26 0.32 2.07 4.17 (4)
NumOfCylinders	4.36 1.06 2 12
EngineSize	126.88 41.55 61 326
Lenght	174.2 12.32 141.1 208.1
Width	65.89 2.1 60.3 72
Height	53.77 2.45 47.8 59.8
CurbWeight	2555.67 517.3 1488 4066
WheelBase	98.8 6.07 86.6 120.9
NumOfDoors	3.14 0.99 2 4 (2)
CompressionRatio	10.16 4 7 23
Horsepower	103.4 37.55 48 262 (2)
Peak-rpm	5117.59 480.52 4150 6600 (2)
City-mpg	25.18 6.42 13 49
Highway-mpg	30.69 6.82 16 54
Price (target)	13207.13 7947.07 5118 45400 (4)

Table 1: Attributes in the Automobile dataset

WindsorEssexHousing: The third dataset, *WindsorEssexHousing*, also concerns real-estate price estimation. It contains the characteristics of 546 houses in the Windsor and Essex area in Canada, sold in the period July-September 1987. The target attribute is the sale price. The attributes are described in Table 3.

For all three dataset we applied the LS_Boost algorithm with three different versions of CART as 'base learners'. Besides the regular CART models, we also used two extreme variants of CART, i.e., unpruned CART and decision stumps in combination with LS_Boost. In unpruned CART, no pruning takes place, only tree growing. This leads to a large tree which is prone to over-fitting. On the other extreme are decision stumps, which are small trees consisting of only one interior node (the root) and two leafs. Decision stumps are extremely easy and fast to create: Only one split and no pruning is required. Individual decision stumps are weak learners with a high error rate, but when combined in an ensemble they are much more powerful.

As benchmark models we used stepwise linear regression and CART without boosting. For each model/dataset combination we ran 100 experiments. In each experiment, the total dataset was randomly divided into a training set containing 90% of the total set, and a test set. (The i-th training set was equal for each model.) In each experiment, the training set was used to fit the model with, and the test set was used to obtain an estimate of the model's prediction error. Our primary evaluation criterion was the mean squared error on the test set. Below, we briefly describe the models as they were used in the experiments.

Name	Description	Avg Stdv Min Max
CHAS	Boolean indicating if tract bounds river	0.07 0.25 0 1
CRIM	Per capita crime rate by town	3.61 8.6 0.01 88.98
ZN	Proportion of residential land zoned for lots over 25,000 sq.ft.	11.36 23.32 0 100
INDUS	Proportion of non-retail business acres per town	11.14 6.86 0.46 27.74
NOX	Nitric oxides concentration (parts per 10 million)	0.55 0.12 0.39 0.87
RM	Average number of rooms per dwelling	6.28 0.7 3.56 8.78
AGE	Proportion of owner-occupied units built prior to 1940	68.57 28.15 2.9 100
DIS	Weighted distances to five Boston employment centres	3.8 2.11 1.13 12.13
RAD	Index of accessibility to radial highways	9.55 8.71 1 24
TAX	Full-value property-tax rate per $10,000	408.24 168.54 187 711
PTRATIO	Pupil-teacher ratio by town	18.46 2.16 12.6 22
B	$1000(Bk - 0.63)^2$ where Bk is the proportion of afro-americans by town	356.67 91.29 0.32 396.9
LSTAT	Percentage lower status of the population	12.65 7.14 1.73 37.97
MEDV (target)	Median value of owner-occupied homes in $1000's	22.53 9.2 5 50

Table 2: Attributes in the Boston Housing dataset

Name	Description	Avg Stdv Min Max
LOT	Lot size in square ft.	5150 2169 1650 16200
BDMS	Number of bedrooms	2.965 0.737 1 6
REC	1 If house has recreational room	0.178 0.838 0 1
STY	Number of stories	1.808 0.868 1 4
FFIN	1 If house has full basement	0.350 0.477 0 1
GHW	1 If house uses gas for hot water heating	0.046 0.209 0 1
CA	1 If house has central air conditioning	0.317 0.466 0 1
GAR	Number of garage places	0.629 0.861 0 3
DRV	1 If house has a driveway	0.859 0.348 0 1
REG	1 If house is in popular neighborhood	0.234 0.424 0 1
FB	Number of full bathrooms (incl. toilet, sink, bath tub)	1.286 0.502 1 4
P (target)	Sale price	6812 26703 25000 190000

Table 3: Attributes in the Windsor Essex Housing dataset

	MSE ($\times 10^6$)				
Model	Mean	Sd	Min	Max	Improvement
SLR	9.9294	5.7549	2.22300	32.4510	0 %
CART	8.6494	4.0573	2.26310	23.3230	12.9 %
LS_Boost CART	5.5086	3.2658	1.3968	17.072	44.5 %
LS_Boost Unpruned CART	6.5590	3.2210	1.8461	17.0650	33.0 %
LS_Boost Decision Stumps	4.4159	2.0413	0.6457	10.597	55.5 %

Table 4: Results for *Automobile*. The column 'Mean' displays the MSE on *out of sample* data over all 100 runs with bootstrapped datasets. 'Sd' gives the standard deviation of these MSE values over the 100 runs, Min and Max give the smallest and largest MSE values encountered during the bootstrapped runs.

Model	p-value	Improvement$_{p=0.05}$	
		abs.	rel.
CART	0.09539	-0.184	-1.08%
LS_Boost CART	3.052×10^{-13}	2.86	28.83%
LS_Boost Unpruned CART	1.688×10^{-9}	2.14	21.64%
LS_Boost Decision Stumps	$< 2.2 \times 10^{-16}$	3.91	39.45%

Table 5: Wilcoxon signed rank test outcomes for *Automobile*. The column p-value in the row for model X gives the p-value for the Wilcoxon signed rank test of the hypothesis *mean of MSE improvement model* $X = 0$ versus the alternative hypothesis *mean of MSE improvement model* $X > 0$. The column Improvement$_p = 0.05$ gives the improvement (absolute and relative) for which the Wilcoxon test would yield a p-value of 0.05. This means that according to the Wilcoxon test, the true improvement is greater than the number given with a probability of 95%.

4 Experiments and Results

Since stepwise linear regression is unable to cope directly with categorical data, it could not be applied to the *Automobile* dataset. Therefore, we transformed each categorical value in this dataset into an indicator vector where each position represents a level of the categorical variable. This vector has a 1 on the appropriate position, 0 in the other positions. SLR is the only model using these dummy variables, the other models use the categorical values.

Our CART benchmark model uses 20% of the training set as a pruning set (see Section 2.2). The remaining patterns in the training-set are used for the the tree growing phase.

The step size parameter α in the LS_Boost algorithm was set to $0.1.\nu_m$ in iteration m, where ν_m is the optimal α for iteration m found by line search. The number of iterations was set to 40 for each dataset. The members in the ensemble of CART models that is created in LS_Boost all use the same pruning set, consisting of 20% of the training set.

Below we present the results of the analyses we performed per dataset. We only pay attention to model performance – Model interpretation is deferred to Section 5.

Table 4 shows out-of-sample errors obtained for all models on the *Automobile dataset*. Note that the boosted models clearly outperform the benchmark models in terms of out-of-sample MSE. The largest improvement with respect to SLR is obtained by boosted decision stumps: 55%. This is more or less surprising because decision stumps nor combinations of decision stumps as created in boosting are able to model interactions between attributes. The latter are, however, able to model complex nonlinearities in one variable. The fact that boosted stumps outperform boosted trees indicates the absence of interaction effects.

We performed the Wilcoxon signed rank test for paired data to test whether obtained improvements were statistically significant. The results of these tests are shown in Table 5. From this table we conclude that the improvements are significant.

	MSE				
Model	Mean	Sd	Min	Max	Improvement
SLR	23.56	8.84	10.42	48.4	0 %
CART	19.06	10.56	5.03	62.8	19.1 %
LS_Boost CART	11.77	6.23	4.27	49.21	50.0 %
LS_Boost Unpruned CART	15.91	10.23	5.26	60.75	32.5 %
LS_Boost Decision Stumps	14.51	6.52	5.28	41.8	38.4 %

Table 6: Results for *BostonHousing*. See the caption of Table 4 for an explanation.

Model	p-value	Improvement$_{p=0.05}$	
		abs.	rel.
CART	7.104×10^{-5}	2.285	9.67%
LS_Boost CART	$< 2.2 \times 10^{-16}$	10.061	42.69%
LS_Boost Unpruned CART	1.768×10^{-10}	6.624	28.09%
LS_Boost Decision Stumps	$< 2.2 \times 10^{-16}$	8.152	34.59%

Table 7: Wilcoxon signed rank test outcomes for *BostonHousing*. Please refer to the caption of Table 5 for an explanation.

The results of the experiments on the *BostonHousing* dataset are reported in Table 6. A similarity with the previous dataset is that the boosted regression trees clearly outperform the benchmark models. In this case, the highest improvement is 50%. The outcomes of the Wilcoxon signed rank tests are shown in Table 7.

The results for the *WindsorEssexHousing*, shown in Table 8, are not as favorable as for the other two datasets. Here, SLR is the best model closely followed by boosted decision stumps. This may indicate the absence of interaction effects. Anglin and Gençay (1996) found no interaction effects either using this dataset and a semi-parametric model with a linear and a nonlinear part that is additive in the original attributes. They do obtain some improvement over a log-linear model w.r.t. out of sample performance. We were unable to perform the Wilcoxon tests for this dataset because we only saved the aggregated results. However, nobody will dispute that boosting does not lead to a significant performance improvement.

5 Interpretation

Parametric techniques often have the advantage that a useful interpretation can be given to the model parameters, e.g., in SLR the model parameters can be interpreted as price elasticities. Although not parametric, regression trees are also highly interpretable and can be written as an equivalent set of if-then rules. Boosted trees lack both these appealing properties. Not all is lost, however, because two useful tools exist that can be used for the interpretation of these ensemble

	MSE ($\times 10^8$)				
Model	Mean	Sd	Min	Max	Improvement
SLR	2.53	0.66	1.09	4.96	0 %
CART	4.17	1.11	2.19	7.64	-64.8 %
LS_Boost CART	3.25	0.96	1.54	6.70	-28.5 %
LS_Boost Unpruned CART	3.70	0.91	1.63	6.58	-46.2 %
LS_Boost Decision Stumps	2.62	0.71	1.17	4.94	-3.6 %

Table 8: Results for *WindsorEssexHousing*. See the caption of Table 4 for an explanation.

models:

Relative importance plots, that visualize how important the various independent variables are relative to one another in predicting the dependent variable. In regression trees, the relative importance of a variable is measured examining the effect of each split on that variable on the model outcome. Roughly speaking, this effect is high if the split results in a large difference in model outcome for the right and the left subtree and/or the split is likely to occur. (The probability that a split occurs is indicated by the number of patterns that travel through the corresponding node relative to the number of patterns in the dataset. Generally this probability is higher for nodes close to the root.) In an ensemble of regression trees such as built by boosting, the relative importances are simply averaged over all trees in the ensemble.

Partial dependence plots, that visualize the partial dependence of the implemented function on a subset of the independent variables. For a grid of values for this variable subset, the 'expected' output of the model at that point is computed. Ideally, this expectation at a grid point should be computed with respect to the conditional distribution (given the grid point) of the variables *not* in the subset. In practice it is obtained by averaging the model outputs over all instances in the dataset, keeping the values in the selected variable subset fixed to the grid point, thus using the dataset to approximate the distribution.

In terms of prediction performance, LS_Boost with regression trees seems superior to the benchmark models for hedonic pricing problems for two out of the three datasets. However, interpretability is also a very important evaluation criterion for model quality. This is especially true within the hedonic pricing context, where the data analyst may be more interested in answering the questions which product features have the most profound influence on product price and what the form is this influence is, than obtaining a somewhat more accurate price estimate. Below, we will compare the various models in terms of attribute importance and functional form of the attribute influence per dataset. We will highlight the most important points of disagreement among the models.

Automobile

The SLR model for the *Automobile* dataset is shown in Table 9. The CART model is shown in Figure 2. The boosted models are interpreted by means of relative importance plots and partial dependence plots – these are given in Figures 3 and 4. For sake of comparison, these figures also show the plots for SLR and CART.

As can be seen from Figure 3, all models designate *EngineSize* and *CurbWeight* (weight of the empty vehicle) as the most important attributes influencing the price. However, the 3-rd, 4-th and 5-th most important attributes differ across the models. Most notably, SLR thinks that the fact that a car is built in Germany is important (3-rd position) whereas *CountryOfOrigin* hardly has any influence in the (boosted) tree models. The (boosted) tree models place (*HorsePower*, *Width* and *Length* on positions 3 to 5 respectively, while these are absent in the SLR model.

Partial depence plots for the three most important variables according to boosted stumps (*EngSz, CurbW, HP*) are shown in Figure 4 together with cumulative histograms of these variables. These graphs show the partial dependencies for boosted stumps, CART and SLR. There are several things noticeable about these graphs. First of all the tree based models found clear nonlinearities for all three variables. For *EngSz*, the range $180 - 200$ seems to be critical for price-effects. This is not surprising, as this area is occupied by models of brands of increasing exclusivity: *Nissan*, *Mercedes-Benz* and *Porsche*. The area above 200 is the domain of exclusive car models, the area below 180 contains primarily all-day car models. For *CurbW* there is also a clear nonlinearity – the most important price effect of this attribute is between 2250 and 2750. The last attribute, *HP*, also has a nonlinear influence. These nonlinearities may be the cause for the difference in relative importance found between the SLR and tree-based models.

Further note that the graphs produced by the boosted stumps are somewhat smoother than the graphs produced by the single trees. This can be attributed to the fact that boosting averages

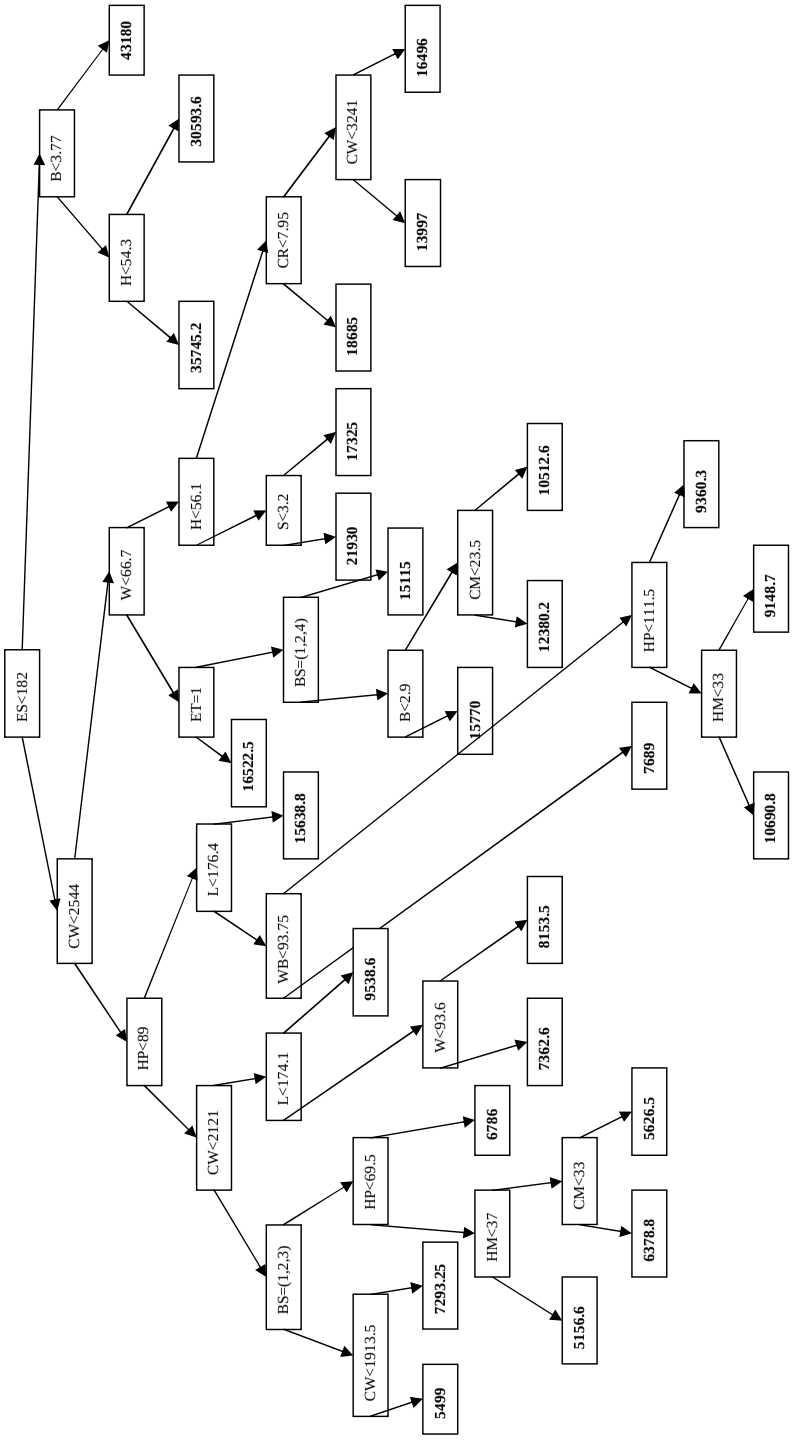

Figure 2: Regression Tree Automobile Dataset

Attribute	Coefficient	Std. Error	P-value
Intercept	-4820.3	4831.4337	0.319763
Germany (Country)	5095.5	507.8983	$< 2 \times 10^{-16}$
Sweden (Country)	2206.5	689.2607	0.001617
Aspiration	1742.5	523.1255	0.001050
Wagon (BodyStyle)	-1297.1	579.1481	0.026331
Rwd (Drive-wheels)	1418.2	515.9862	0.006596
EngineLocation	7051.9	1604.5853	1.89×10^{-5}
CurbWeight	4.8092	0.9845	2.28×10^{-6}
Ohcv (EngineType)	3397.8	952.6319	0.000463
EngineSize	126.7	12.1772	$< 2 \times 10^{-16}$
1bbl (FuelSystem)	2192.2	871.8692	0.012801
2bbl (FuelSystem)	1683.7	536.1671	0.001974
Bore	-2610.1	939.8130	0.006062
Stroke	-2877.6	649.8477	1.65×10^{-5}
Peak-Rpm	1.0144	0.4292	0.019182

Table 9: Stepwise LR Model Automobile

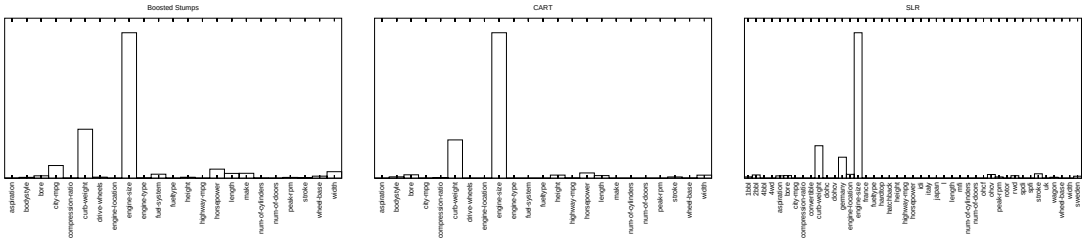

Figure 3: Relative importance plots *Automobile* dataset

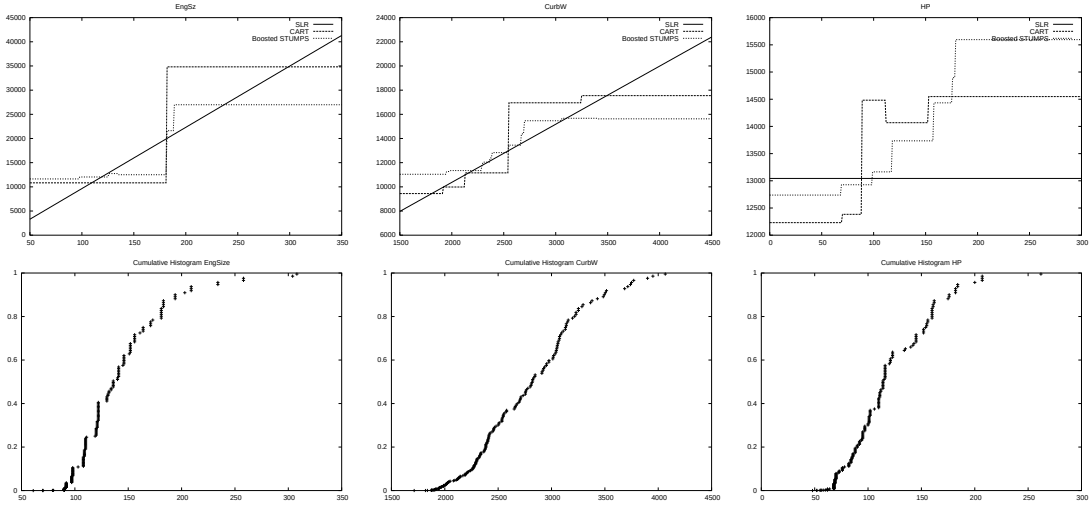

Figure 4: Partial dependence plots (top) and cumulative histograms *Automobile* dataset

Attribute	Coefficient	Std. Error	P-value
Intercept	36.341145	5.067492	2.73×10^{-12}
CRIM	-0.108413	0.032779	0.001010
ZN	0.045845	0.013523	0.000754
CHAS	2.718716	0.854240	0.001551
NOX	-17.376023	3.535243	1.21×10^{-6}
RM	3.801579	0.406316	$< 2 \times 10^{-16}$
DIS	-1.492711	0.185731	6.84×10^{-15}
RAD	0.299608	0.063402	3.00×10^{-6}
TAX	-0.011778	0.003372	0.000521
PTRATIO	-0.946525	0.129066	9.24×10^{-13}
B	0.009291	0.002674	0.000557
LSTAT	-0.522553	0.047424	$< 2 \times 10^{-16}$

Table 10: Stepwise LR model *BostonHousing* dataset

over a large number of trees, whereas a single CART model contains a very limited number of splits on a certain variable (especially when using limited data) and this results in a 'step function' like approximation with large discontinuities. In a boosted model these large discontinuities are replaced by a large number of smaller discontinuities, yielding a 'smoother' function.

Boston Housing

The obtained parameters for the SLR model and the CART model obtained are shown in Table 10 and Figure 5. Both models displayed here were fitted on the whole dataset.

The relative importance plots for SLR, CART and boosted CART are shown in Figure 6. The partial dependence plots of the most important attributes are shown in Figure 7, together with their cumulative histograms.

Considering the relative importance plots, it is clear that there is a large difference in attribute preference across the models. The most striking differences concern the variables *RAD* and *CRIM*. *RAD*, the accesibility of highways, the 4-th most important attribute according to SLR, but in the boosted CART model it occurs on the 11-th position with hardly any influence. Furthermore, boosted CART assigns a much greater weight to the attribute *CRIM*, the criminality ratio, than SLR. This attribute takes the 3-rd place in boosted CART and the 9-th place in SLR. The tree based models seem to rely on fewer variables than the SLR model. CART uses only 4 attributes in its model and these are also the most important ones in boosted CART. In contrast, SLR uses 11 attributes, with 7 significant ones.

The partial dependence plots in Figure 7 clearly show that CART and boosted CART find nonlinearities in the data: the most important price effect of the number of rooms is between 7 and 8 rooms, and there seems to be a saturation effect in the negative price effect of *LSTAT* and *CRIM*.

Windsor Essex Housing

SLR model and CART model for this dataset are shown in Table 11 and Figure 8. Relative importance plots and partial dependence plots for this dataset are shown in Figures 9 and 10.

For the *WindsorEssexHousing* dataset all models assign the highest relative importance to *LOT*, the lot size, and *FB*, the number of bathrooms. The SLR models assigns the 3-rd rank to *STY*, the number of stories. This attribute plays no role in the CART model and a limited role in the boosted stumps model. The 3-rd place in these model is taken by *CA*, a boolean indicating the presence of air conditioning. This attribute comes 4-th in the SLR model. Strikingly, the number of bedrooms plays no role in SLR and CART, and a very limited role in boosted CART. This contradicts the results on the *BostonHousing* dataset, where *RM* was unanimously found to

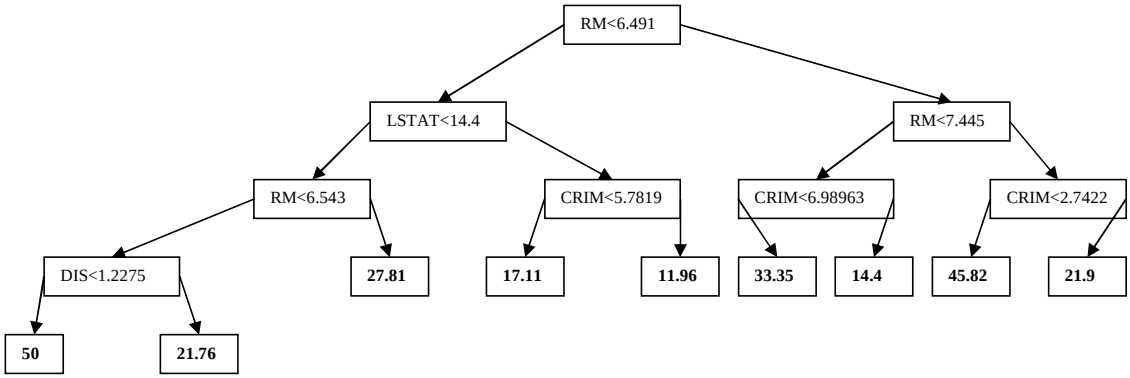

Figure 5: Regression tree *BostonHousing* dataset

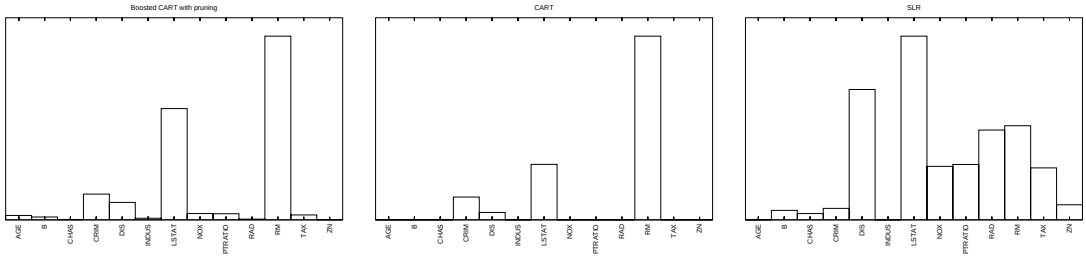

Figure 6: Relative importance plots *BostonHousing* dataset

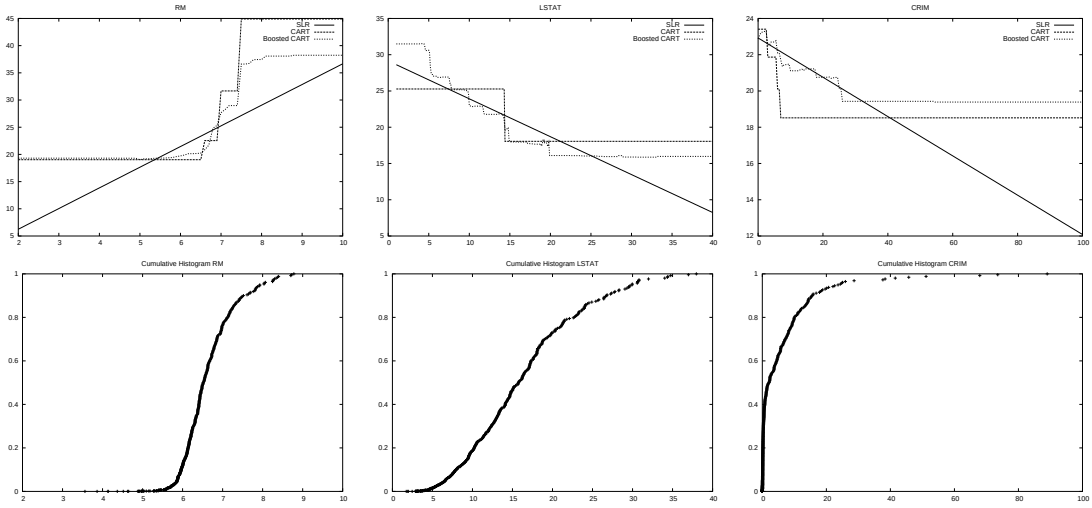

Figure 7: Partial dependence plots and cumulative histograms for the *BostonHousing* dataset

Attribute	Coefficient	Std. Error	P-value
Intercept	-497.3484	2749.2067	0.856510
LOT	3.5959	0.3498	$< 2 \times 10^{-16}$
FB	14924.1569	1454.2442	$< 2 \times 10^{-16}$
STY	7128.7990	867.3174	1.55×10^{-15}
DRV	6259.6177	2034.4638	0.002200
REC	4440.4102	1903.1824	0.020010
FFIN	5846.5080	1574.9946	0.000227
GHW	12949.4428	3223.0822	6.72×10^{-5}
CA	12605.9217	1557.9379	3.98×10^{-15}
GAR	4355.3216	839.7822	3.05×10^{-7}
REG	9431.7782	1671.9235	2.74×10^{-8}

Table 11: Stepwise LR model *WindsorEssexHousing*

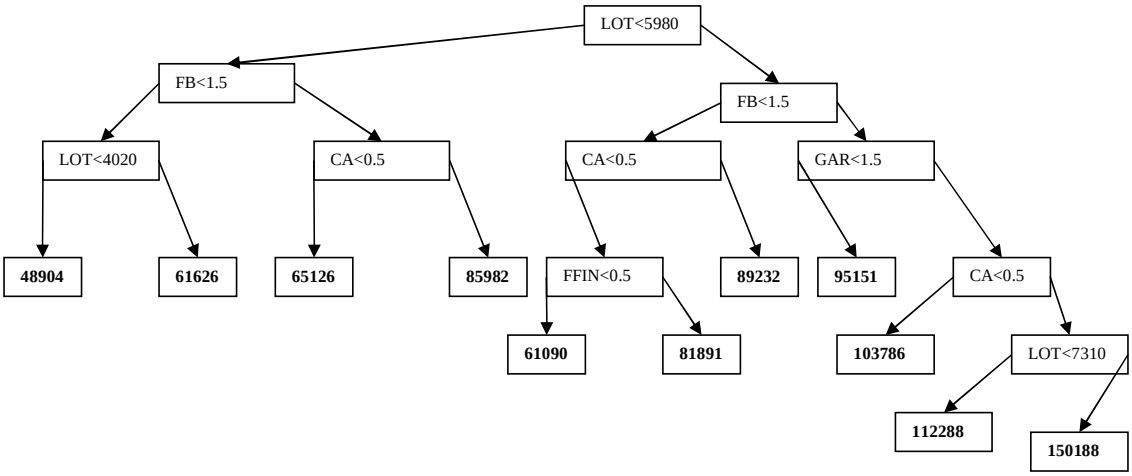

Figure 8: CART model *WindsorEssexHousing*

be the most important indicator. The most probable explanation for this observation is that both datasets come from a different domain where different preferences may govern the influence on price.

Considering the partial dependence plots, we see that the amount of nonlinearity if much lower than for the previous datasets. E.g., the partial dependency plot boosted decision stumps for attribute *LOT* fluctuates around the line found by SLR. It is a well-known property of tree-based models that they are not well able to approximate linear (non-constant) relationships – This requires a large number of splits. The linearity assumption may be appropriate for this dataset and this may explain the degraded performance of the tree-based models as opposed to the SLR model.

6 Summary, discussion and conclusions

We studied the application of boosted regression trees to three hedonic pricing problems. Three variants of regression trees were used as a base model: normal CART, CART without pruning and decision stumps consisting of one node. The out of sample performance of these models was compared with reference models CART and stepwise linear regression.

For two out of the three datasets, *Automobile* and *BostonHousing*, the boosted tree models

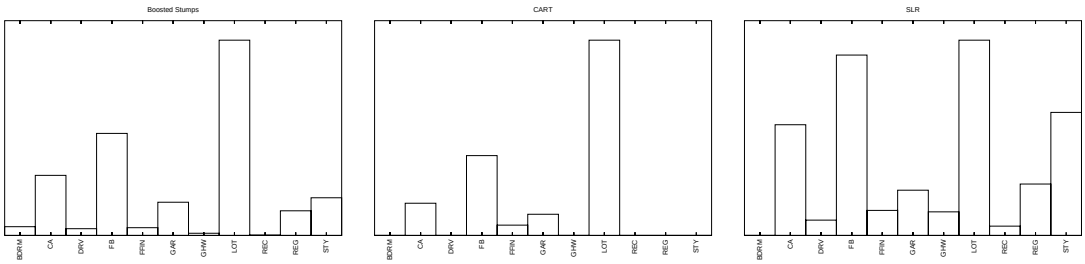

Figure 9: Relative importance plots *WindsorEssexHousing* dataset

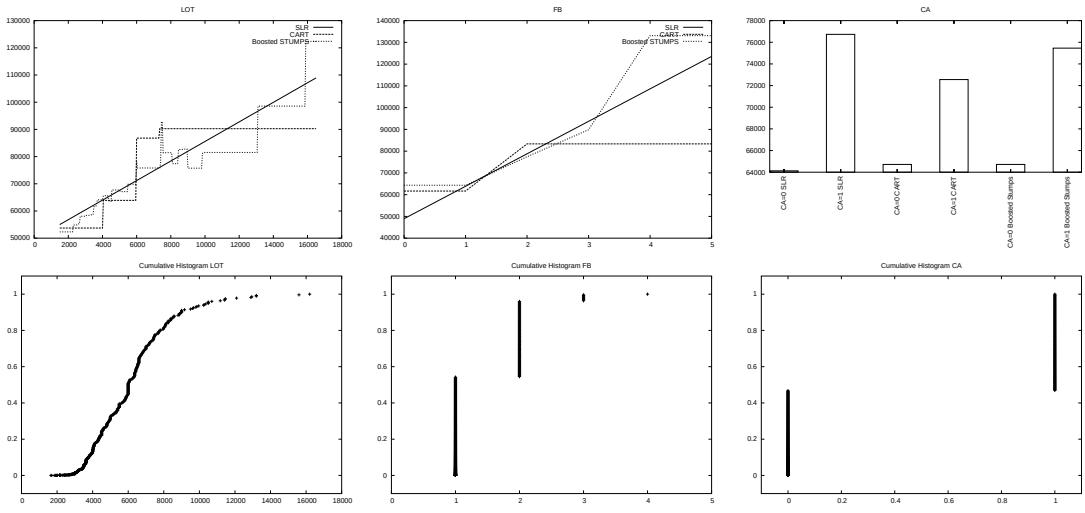

Figure 10: Partial dependence plots and cumulative histograms for the *WindsorEssexHousing* dataset

substantially improved the SLR reference models. For the third dataset, *WindsorEssexHousing*, the results were less encouraging. The best performing base learner for the *BostonHousing* dataset was a normal, pruned CART model. For the *Automobile* dataset decision stumps were more successful. The most likely explanation is the absence of interaction effects in this latter dataset.

We used two methods, partial dependence plots and relative importance plots, for interpretation of the boosted models. Differences in attribute importance among the models were highlighted. Several clear examples of nonlinearities were found. Moreover it turned out that the various model types assigned very different probabilities to some of the attributes.

There are several ways in which this research can be extended. First of all, we intend to use different base learners in combination with LS_Boost. Other base learners than CART, e.g., neural networks or multivariate regression splines, have the advantage that they yield smoother mappings than boosted decision trees and have no problems approximating linear relationships. This may lead to better quality predictions. Another extension could be the application of monotone decision trees (Potharst and Bioch 2000) as base learners. These models guarantee a monotone relationship between the input attributes and the price, which is often realistic in hedonic pricing problems, but they retain a large part of the flexibility of normal decision trees.

References

Anglin, P. M. and R. Gençay (1996). Semiparametric estimation of a hedonic price function. *Journal of Applied Econometrics 11*, 633–648.

Bin, O. (2004). A prediction comparison of housing sales prices by parametric versus semiparametric regression. *Journal of Housing Economics 13*, 68–84.

Breiman, L. (1996). Bagging predictors. *Machine Learning 24*(2), 123–140.

Breiman, L. (1998). Arcing classifiers. *Annals of Statistics 26*(3), 801–849.

Breiman, L., J. Friedman, R. Olshen, and C. Stone (1984). *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks/Cole. ISBN 0412048418.

Freund, Y. and R. Schapire (1996). Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*. Morgan Kaufmann.

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics 29*(5), 1189–1232.

Gençay, R. and X. Yang (1996). A forecast comparison of residential housing prices by parametric versus semiparametric conditional mean estimators. *Economics Letters 52*, 129–135.

Gilley, O. W. and R. K. Pace (1995). Improving hedonic estimation with an inequality restricted estimator. *The Review of Economics and Statistics 77*(4), 609–621.

Green, P. E., A. M. Krieger, and Y. Wind (2001). Thirty years of conjoint analysis: Reflections and prospects. *Interfaces 31*(3), S56–S73.

Griliches, Z. (1971a). Hedonic price indexes for automobiles: An econometric analysis of quality change. In Z. Griliches (Ed.), *Price Indexes and Quality Change: Studies in New Methods of Measurement*, pp. 55–87. Cambridge: Harvard University Press.

Griliches, Z. (1971b). Hedonic price indexes revisited. In Z. Griliches (Ed.), *Price Indexes and Quality Change: Studies in New Methods of Measurement*, pp. 3–15. Cambridge: Harvard University Press.

Harrison, D. and D. Rubinfeld (1978). Hedonic prices and the demand for clean air. *Journal of Environmental Economics and Management 5*, 81–102.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning*. Springer series in statistics. Springer. ISBN 0387952845.

Hettich, S. (2004). UCI machine learning repository. Web-Site. `http://www.ics.uci.edu/~mlearn/MLRepository.html`, School of Information and Computer Science, University of California, Irvine.

Lancaster, K. (1966). A new approach to consumer theory. *Journal of political economy 74*, 132–157.

Lilien, G. L. and A. Rangaswamy (2002). *Marketing Engineering: Computer-Assisted Marketing Analysis and Planning* (2nd ed.). Prentice Hall. ISBN: 0130355496.

Miyamoto, M. and H. Tsubaki (2002). A linear mixed model for the hedonic pricing model. *Applied Stochastic Models in Business and Industry 18*, 259–270.

Ofek, E. and V. Srinivasan (2002). How much does the market value an improvement in a product attribute? *Marketing Science 21*(4), 398–411.

Pace, R. K. (1998). Appraisal using generalized additive models. *Journal of Real Estate Research 15*, 77–99.

Potharst, R. and J. Bioch (2000). Decision trees for ordinal classification. *Intelligent Data Analysis 4*(2), 97–112.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann. ISBN 1558602380.

Rosen, S. (1974). Hedonic prices and implicit markets: Product differentiation in pure competition. *Journal of Political Economy 82*(1), 34–55.

Tomkovick, C. and K. E. Dobie (1995). Applying hedonic pricing models and factorial surveys at parker pen to enhance new product sales. *Journal of Product Innovation Management 12*, 334–345.