

Diagnosing Extrapolation: Tree-Based Density Estimation

Giles Hooker
Department of Statistics
Stanford University
Stanford, CA, USA
gilesh@stanford.edu

ABSTRACT

There has historically been very little concern with extrapolation in Machine Learning, yet extrapolation can be critical to diagnose. Predictor functions are almost always learned on a set of highly correlated data comprising a very small segment of predictor space. Moreover, flexible predictors, by their very nature, are not controlled at points of extrapolation. This becomes a problem for diagnostic tools that require evaluation on a product distribution. It is also an issue when we are trying to optimize a response over some variable in the input space. Finally, it can be a problem in non-static systems in which the underlying predictor distribution gradually drifts with time or when typographical errors misrecord the values of some predictors.

We present a diagnosis for extrapolation as a statistical test for a point originating from the data distribution as opposed to a null hypothesis uniform distribution. This allows us to employ general classification methods for estimating such a test statistic. Further, we observe that CART can be modified to accept an exact distribution as an argument, providing a better classification tool which becomes our extrapolation-detection procedure. We explore some of the advantages of this approach and present examples of its practical application.

Categories and Subject Descriptors

G.3 [Probability and Statistics]: Distribution Functions; I.5.3 [Pattern Recognition]: Clustering—*Algorithms*; H.2.8 [Database Management]: Database Applications—*Data Mining*

General Terms

Algorithms, Measurement, Documentation, Verification

Keywords

Visualization, Density Estimation, Extrapolation, Diagnostics, Interpretation, CART, C4.5, Trees-based models, Modeling methodologies, Clustering

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD'04, August 22–25, 2004, Seattle, Washington, USA.
Copyright 2004 ACM 1-58113-888-1/04/0008 ...\$5.00.

1. INTRODUCTION

Most Machine Learning algorithms are trained on real-world data which exhibit strong and complex dependence structures among predictor variables. These properties are particularly notable in the common scenario that the predictor space is very high-dimensional. This situation means that most learning algorithms only see data in a very small, often complicated region of the hyper-rectangle bounding the ranges of the data.

Most learners also do not specify the behavior of their resulting functions away from a training sample. They are usually designed to be universal approximators – or as close as is practical – and have minimal modeling restrictions. This, in turn, provides very little prior control of the function in regions of little or no data. As a result, in most Machine Learning settings, we can neither control the behavior of the prediction function at points of extrapolation, nor determine where this is a problem.

Nominally, extrapolation should not be an issue; assuming a representative training sample in a static system, the probability of being required to predict a point of extrapolation is low, almost by definition. However, most training sets are not representative, do not come from static systems and we may well be required to extrapolate. In high dimensions, even empirical data generated from a product distribution can appear to have strong correlation structure. Since functions are learned conditional on an empirical sample, they may still be effectively extrapolating even in regions of theoretically large density.

Even when extrapolation is not an issue for prediction, it does become a problem in two situations. Firstly, when trying to understand the behavior of learned prediction functions, many diagnostic tools implicitly require the evaluation of the function on a measure in which the set of variables of interest are independent of its complement. Such an assumption can move a large amount of probability mass into regions of extrapolation, giving these regions undue influence over a representation of functional behavior.

The second situation in which extrapolation is a direct issue is when one or more predictor variables can be controlled. In this case an agency might want to search over the values of those variables in order to optimize a response. If there is strong historical correlation between these variables and the rest of the predictors (which remain fixed), such a search will involve extrapolation.

We present a new tool to address the first issue; how do we determine whether a given point in predictor space is a point of extrapolation? Such a determination has multiple uses. It represents a new diagnostic for outliers, both in the training data set and in unseen data. It provides a diagnostic for shifts in system dynamics; we can quantify how much extrapolation we are seeing and whether

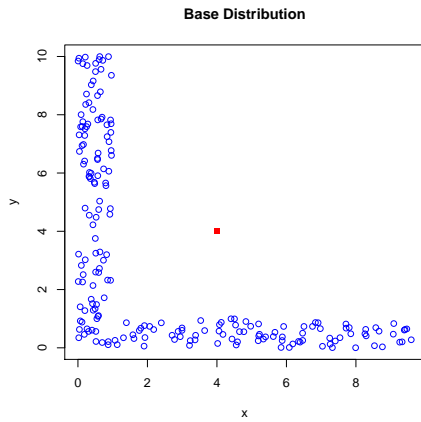


Figure 1: An example distribution with strong extrapolation. The square at (4, 4) represents a point of extrapolation which is difficult to diagnose.

or not the rate of extrapolation is increasing. Such an observation suggests a shift in the distribution of the predictor variables and a need to retrain the model. We will show that this tool provides an interpretable density estimate in high dimensions, giving a rough diagnostic for non-linear covariance structures. Finally, in producing a rough estimate of the distribution of the training data, it provides an important element in building diagnostic techniques that are resistant to bad extrapolation.

2. PROBLEMS OF EXTRAPOLATION

2.1 Extrapolation and Machine Learning

Diagnosing extrapolation is of interest in itself in terms of enabling an understanding of the distribution of underlying predictor variables. The ideas presented in this paper represent one of the few interpretable estimates of density of which the author is aware. In the context of Machine Learning, such a diagnostic is of added importance. Few Machine Learning procedures are built with an eye to extrapolation. This approach has two main consequences:

- Uncontrolled extrapolation can create obviously unrealistic predictions, even in superficially reasonable points of predictor space.
- Prediction at points of extrapolation exhibits high variance under perturbations of the training data, even when the predicted values appear reasonable.

We will illustrate this issue with the aid of a bivariate distribution given in Figure 1. The association structure given there consists of an “arm” along each axis and is similar to that found in the Boston Housing Data, [3], which will be used as an example below. Our canonical point of extrapolation for the purpose of this exposition is (4, 4): clearly different from the remainder of the distribution, but within its convex hull.

To demonstrate that unreasonable predictions can occur, consider the linear model $f(x_1, x_2) = x_1 + x_2$; the response at this point is 8. However, if we extend this structure to 5 dimensions – placing an “arm” on each axis – the response at the equivalent point is 20: already double the response for any point within the real distribution. This may not be an issue if we genuinely believe in the

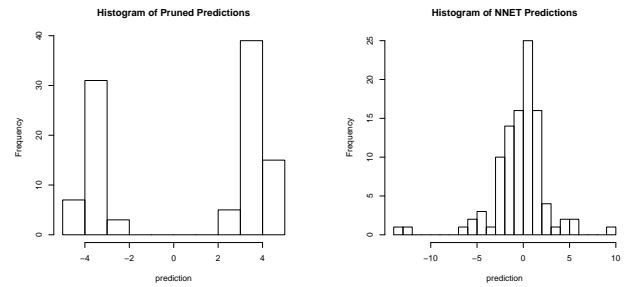


Figure 2: A histogram of CART predictions (left) at (4, 4) point for 100 samples from the data given in Figure 1 and response $x_1 - x_2 + \epsilon$. The right hand plot gives a histogram of predictions from a 2-20-1 neural network. Both are highly variable.

linear model on the whole space. A more flexible model that does not incorporate a strong prior belief can provide worse extrapolation without reason to believe its predictions far away from training data.

In many instances, extreme predictions are obviously nonsensical and are therefore easy to diagnose. However, that diagnosis does not account for the variance of predictive functions when retrained with different data. This is illustrated in Figure 2 using trees and neural networks, both of which have finite bounds on the predictions they give. We used 200 points with the same distribution as above, generating response $x_1 - x_2 + \epsilon$ with $\epsilon \sim N(0, 0.04)$ to train a tree and a 2-20-1 neural network 100 times using different samples each time. Figure 2 provides histograms for the predictions of each model at the point (4, 4). These demonstrate very high variance. Trees produce strongly bimodal predictions, depending on what “arm” is split first. Neural networks have many parameter values that produce similar predictions on the training data but which diverge sharply away from it.

2.2 Deliberate Evaluation at Extrapolation

As noted, extrapolation is not nominally a concern for a static system with a representative training sample that is large with respect to its dimensionality. This is not common in data mining situations and even in this case, there are at least two situations in which prediction functions may be deliberately evaluated at points of extrapolation.

The first of these is when one of the predictor variables can be controlled by an agency. This might be the case, for example, for a credit agency dealing with customers who default on payments. One predictor of payment is the action that the agency takes to remind a customer that the payment is due: the frequency of letters, telephone calls, personal visits etc. Historically, the more severe the problem, the more extreme the action taken. If the agency then wants to build a predictor system for the probability of default so that they can choose a most cost-effective action, searching over the action space will evaluate the function at points of extrapolation. In Figure 1, x_1 might represent an amount paid with x_2 being the reminder action taken. Then a search over actions x_2 at $x_1 = 4$ will evaluate at $(4, k)$ for $k \in [1, 10]$. A diagnostic tool for extrapolation will at least allow such a procedure to flag predicted values that are untrustworthy. It also provides a diagnostic for areas of the predictor space in which the agency can conduct experiments to better gauge a response.

The second scenario in which functions are deliberately evalu-

ated at points of extrapolation is in the use of diagnostic tools. A popular tool for understanding prediction functions is the Partial Dependence Plot, developed in the context of ensemble methods using trees. [2] defines the partial dependence of a function $F(z)$ on z_l – a subset of the variables – to be $F(z_l, z_{\setminus l})$, averaged over the marginal distribution of $z_{\setminus l}$, the remaining predictors. There is a natural data-driven approximation for this quantity of F on z_l given by

$$\frac{1}{N} \sum_{i=1}^N F(z_l, z_{i,\setminus l}) \tag{1}$$

which makes the shift in probability mass clear. Even if the distribution of $z_{\setminus l}$ is highly concentrated at z_l , we still measure F on all the points $\{z_l, z_{i,\setminus l}\}_{i=1}^N$.

As an empirical demonstration of the dangers of extrapolating, consider the following example. Take the distribution in Figure 1 and let us suppose that we have an estimate

$$g(x_1, x_2) = x_1 + x_2 + (x_1 - 1)_+^2 (x_2 - 1)_+^2.$$

for an expected response $f(x_1, x_2) = x_1 + x_2$. f has been recovered exactly on the training data distribution, but has a spurious term which only appears in a region of zero probability. This sort of term can be found, for example in MARS [4], and might arise due to a single outlying data point. Despite not wishing to capture this spurious effect, the partial dependence plot moves a large fraction of probability mass into the empty region where it occurs. The partial dependence of g on x_1 is plotted in Figure 3. It is clear in this case that the partial dependence plot is unrepresentative of the true behavior of the function as we change x_1 . At the point $x_1 = 9$, say, we are still averaging $\{F(9, x_{i,2})\}_{i=1}^{200}$ even though these points can be very far away from any of the original data points.

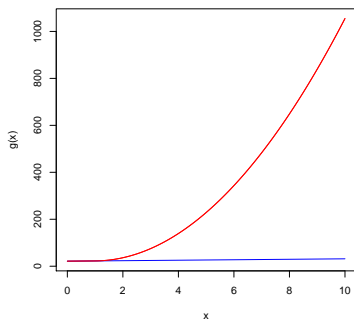


Figure 3: The partial dependence plot of g on x ; the lower line represents the target function $y = x$, here almost horizontal due to the size of the spurious effect.

3. A MEASURE OF EXTRAPOLATION

We propose to measure an extrapolation quotient as being the Neymann-Pearson test statistic at the point of interest for distinguishing the data distribution from a uniform distribution null hypothesis on the same range. Given the true data distribution and the uniform, this is the test with minimum theoretical misclassification rate. It is equivalent to the classification probability of a point being generated by the process generating the empirical data as opposed to the uniform distribution with each distribution is given

prior probability of one half. Formally:

$$\text{Extrap}(x) = \frac{\hat{P}(x)}{\hat{P}(x) + U(x)}. \tag{2}$$

Here, \hat{P} is an approximation to the data distribution P , which has compact support. This would normally be an empirical approximation given a data set. More formally, however, a distribution with non-bounded support can be truncated so as to leave some small probability mass outside the support. U is then a uniform distribution on the range of the support of \hat{P} .

This approach has a number of advantages. We formalize the question, “*Is this a point of extrapolation?*” to a test: “*Would we choose to believe this point to be generated from the same distribution as the training data or from a uniform distribution?*” Here we wish to determine if a new data point indicates a move away from the processes generating the training data and, if so, assess how much confidence can be placed in the predictions that our learned function produces.

We have chosen the uniform distribution as a natural null hypothesis in the sense that it is least informative about how a point of extrapolation might be generated. It also corresponds to a best-case underlying distribution from the point of view of extrapolation - the coverage of the space may be sparse, but it provides the same confidence in predicted values at all areas of the space. This gives us that $\text{Extrap}(x) \geq 1/2$ indicates that we are in an area of relatively dense points. If not, then some caution is warranted in using a predicted value¹. We will also see that the uniform distribution is computationally convenient to use in the procedures below.

A further advantage is that under this framework, $\text{Extrap}(x)$ may be coarsely estimated using generic classification tools. This avoids the need to find a good density estimation algorithm. Of course, the density can be recovered from the classifier using the approach in [4]:

$$\hat{P}(x) = \frac{\text{Extrap}(x)U(x)}{1 - \text{Extrap}(x)}.$$

The use of a score for extrapolation is advantageous over a simple classification for several reasons. To begin with, it provides a sense of the relative density of training points - a handful of very sparsely distributed points in some region may increase the confidence we place in a prediction, but not as much as having many training examples nearby. A confidence score can be used when searching over possible actions to weigh the potential gain from each action. It is also possible to design diagnostic tools that make use of a score to provide low dimensional plots which will capture more behavior than a simple classification of outliers [5]. Of course, when such a classification is required, a simple cutoff can be applied.

4. CONFIDENCE AND EXTRAPOLATION REPRESENTATION TREES (CERT)

We propose to estimate (2) directly via a classification tool. In particular, we feel that tree-based methods provide a useful framework in which to do this. Particular advantages of this approach include

- Interpretability, and in particular an interpretable set of regions in which to display summary diagnostics for functional behaviour. This is demonstrated in Figure 7

¹The confidence placed in predictions, of course, must be tempered with the overall ratio of the number of points to number of dimensions. This can be done directly for a uniform distribution, treated like a prior; the measurement here represents a refinement on that confidence.

Research Track Poster

- The resulting regions are hyper-rectangles. For more sophisticated diagnostic tools, the leaves of the tree can be turned into product measures, providing a mixture-of-products approximation to the data density.
- Trees are both computationally efficient and a known and accepted part of machine learning, making the understanding of a new diagnostic tool easier.
- Trees can take a known distribution as an argument. In particular, we are able to classify a data distribution against a theoretical distribution. This provides a marked improvement in the resulting estimation.

4.1 Monte-Carlo Data and the Curse of Dimensionality

An initial proposal might be to simulate a random sample from a uniform distribution and then use CART [1] to classify the two data sets. This approach, however, turns out to produce highly variable measures. Moreover, in high-dimensional situations in which real data occupies a space of small Lebesgue measure, a tree will often be able to exactly separate the real data from a Monte-Carlo uniform sample long before it captures the distribution of the real data. This situation has the potential to leave regions of extrapolation marked as regions of confidence.

By way of illustration, consider N points drawn from a multivariate “porcupine” distribution, P_1 , given as the natural extension of the example distribution in Figure 1: one arm extending along each axis. For simplicity, the arms will only extend two units. The support of P_1 has $U(0, 2)^n$ measure $\frac{n+1}{2^n}$ and requires $2n(n-1)$ splits to define. Define a new density P_2 by taking the first k predictors as independent given the rest - filling in their marginal distributions. Using Monte Carlo uniform data, in order to distinguish between these two distributions, a classifier needs to see a point in the support of P_2 but not P_1 . The probability of this is

$$1 - \left(1 - \frac{2^k(n-k+1) - n - 1}{2^n}\right)^N.$$

We require N to scale linearly with n for this probability to remain constant. However, there are $\binom{n}{n-k}$ alternative distributions that can be defined this way, so the probability of producing a tree that does not describe all the detail of P_1 is large. This means that a method based on a Monte Carlo uniform sample will not be as powerful as one that “knows” the uniform distribution exactly.

4.2 CART and Distributional Information

Trees are capable of taking an exact distribution as an argument instead of an empirical data set. For each split, we merely replace the number of Monte-Carlo uniform data points on each side of the split with the expected number. This both reduces sources of variance in the original data and allows a much finer approximation to the empirical data density.

To illustrate this point, the results of a simulation are reproduced in Figures 4 and 5. We simulated 50 data sets of 1000 points with a Gaussian distribution in 5 dimensions. We then used CART to classify them away from another 1000 points, uniformly distributed on the empirical range of the data. A tree was also built for each of the data sets using the technique described above. Figures 4 and 5 compare their accuracy, stability, and resolution on a further set of 1000 normally and 1000 uniformly distributed points.

For these instances the trees were pruned in a standard manner - using the 1 standard error rule and cross validation for the standard CART trees. A separate test set of normally distributed points along with the expected uniform points was used to prune the trees

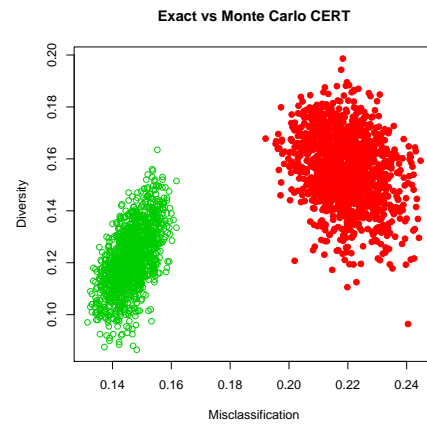


Figure 4: Performance for CART with Monte Carlo uniform samples (solid) and with uniform expectations (hollow). Average accuracy is plotted on the x axis for pairs of trees against diversity on the y axis, measured by the proportion of points classified differently.

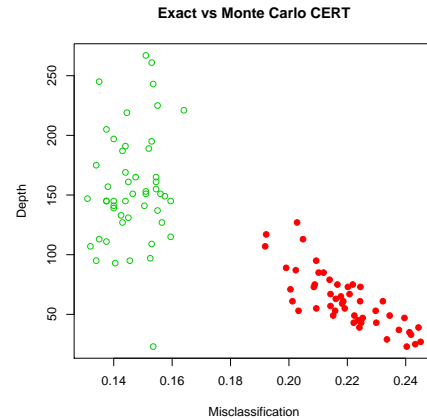


Figure 5: Performance for CART with Monte Carlo uniform samples (solid) and uniform expectations (hollow). Misclassification accuracy is plotted for individual trees on the x axis against size of tree given by number of nodes on the y axis.

incorporating distributional information. It can be seen from these experiments that the use of distributional information improves the trees in all three dimensions, providing improved classification performance, greater classification stability and greater resolution.

5. CERT DETAILS

There are several components to the CART algorithm, not all of which are necessarily appropriate in this setting. This section provides a list of specific criteria included in CERT.

5.1 Splitting Criteria

All tree-building algorithms use a greedy search strategy, picking the next split on some score. CART uses the Gini index, also employed in CERT. C4.5 [6] uses binomial entropy.

5.2 Pruning

The CART procedure builds a large tree and then “prunes” it - removing lower splits which do not increase predictive accuracy. In CART the amount of pruning done is chosen based on cross-validated misclassification error. The trees used here have been pruned based on test-set misclassification error.

Alternative scores can be employed for different purposes. If an estimate of density is required, then using binomial entropy calculated by a test set may provide more accuracy. Empirically, it leads to larger trees. Alternatively, a measure of independence between variables may be useful for developing a representation that can be used with diagnostic tools.

A final possibility for some applications is the C4.5 rule-pruning strategy which does not maintain the tree structure. The density model produced from this approach now takes the form of a mixture of overlapping uniform distributions, also indicating a potential fuzzy clustering.

5.3 Missing Values and Surrogate Splits

When missing values are encountered on new data which are required at some split, CART tries to determine which branch to follow based on correlations between predictor variables. This uses a technique called surrogate splits. CERT diverges from CART on this issue and we employ the strategy of C4.5. Suppose that we are missing x_i , then the natural measure of extrapolation should be

$$\text{Extrap}(x_{-i}) = \frac{P(x_{-i})}{P(x_{-i}) + U(x_{-i})}$$

This can be written alternatively as

$$P(D|x_{-i}) = \sum_{k=1}^K P(D|L_k)P(L_k|x_{-i}),$$

where D is an indicator that x was generated from the data distribution and $\{L\}_{k=1}^K$ represent the terminal nodes of the tree. Now we observe that $P(L_k|x_{-i})$ is exactly the weight given to each leaf if we traverse the tree according to the C4.5 strategy - going left and right at splits involving x_i with probability equal to the proportion of total (real and uniform) weight in that direction.

6. OUTLIER DETECTION

An obvious first use for any measure of extrapolation is as an outlier detection device. We will simply call a point x an outlier if $\text{Extrap}(x)$ is less than some constant C . Figure 6 compares the performance of CERT with the common technique of excluding points based on their Mahalanobis distance (with the training covariance) from the mean of the training data. We have chosen two distributions to test this. The first is a 5-dimensional Gaussian with a tri-diagonal correlation matrix that takes 0.3 on the off-diagonals. Mahalanobis distance should be optimal for this situation. The second is a 5-dimensional extension of the distribution from §2 - an “arm” extending along each axis, which should be well-described by CERT. We used 200 training examples and outliers were generated by a uniform distribution. In order that the performance of the two methods be comparable, we have drawn new sample from the training distribution and plotted the percentage of misclassified “outliers” against the percentage of misclassified “real” examples.

From Figure 6 we can see that Mahalanobis does, as expected, outperform CERT on a multivariate Gaussian. This discrepancy becomes larger as the correlation between variables becomes stronger. CART allows splits to be made on linear combinations of variables - although that option is rarely used - and employing this with

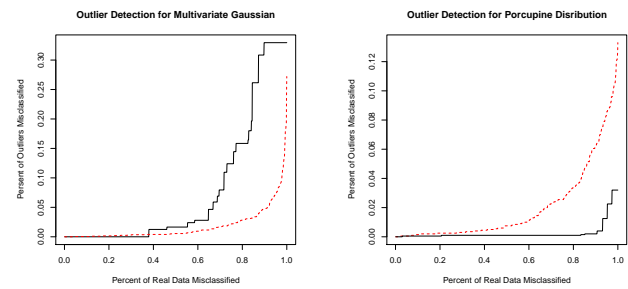


Figure 6: Performance of outlier detection techniques. CERT (solid) and Mahalanobis distance (dashed) on a multivariate Gaussian (left) and porcupine distribution (right). When the distribution of data is non-convex, CERT has a significant performance advantage.

CERT should improve CERT’s performance. For an independent Gaussian distributions the performance of the two methods is much closer. Having more training examples also improves the relative performance of CERT. In our second distribution, however, it is quite clear that CERT outperforms Mahalanobis distance; any ellipsoid that tries to span the training data in this case will inevitably contain large amounts of empty space.

Although we do not present results here, given a measure of extrapolation, concept drift can be measured by the amount of extrapolation occurring. If the sum of extrapolation scores taken over some period shows an increase beyond an acceptable level, there is good reason to retrain the model. CERT can also be used in this context to indicate into which regions the underlying predictor distribution is moving.

7. DESCRIPTIVE STATISTICS

Beyond the utility of being able to detect extrapolation well, diagnostic tools can also be helpful in understanding where the training data lies and the associations that exist between variables. A large factor in the choice of trees as a classification tool is that they provide an interpretable set of regions in which to examine the behavior of a learned function. These regions remain high-dimensional and do not admit any easier display of a function. However, they do allow us to provide a summary of the behavior of a function within that region. Descriptive statistics such as function means and variances on each region as well as the proportion of original data points in a region provide an assessment of how serious extrapolation may be. In particular, it diagnoses areas where large Gibbs effects occur and allow us to artificially smooth the function, or to flag predictions within those regions as suspect.

In Figure 7 below, we present the graphical representation of the underlying predictor space for the Boston Housing data. In particular, for each leaf l , we report four numbers in order:

- The number of real data points in the leaf, N_l .
- The expected number of uniform points in the leaf, U_l .
- The mean value of a prediction function evaluated on a uniform sample in that leaf, μ_l .
- The variance of the function evaluated on the same uniform sample, σ_l^2 .

Research Track Poster

We would then be concerned about predictions in leaves with a small number of real data points. This concern would be strengthened if we observe a large variance or an extreme mean in that leaf. Such concern incorporates a heuristic belief that predictions are likely to be highly variable with resampling of the training data in regions where \hat{F} changes a great deal without data supporting such variation.

It is tempting to produce a pseudo-prediction interval:

$$F(x) \pm \Phi^{-1}(\alpha/2) \frac{\text{var}_{\Omega_x}(F)}{N_x}. \quad (3)$$

Here $\text{var}_{\Omega_x}(F)$ represents

$$\text{var}(F(x)|F, x \in \Omega_x)$$

where Ω_x is the leaf of the CERT model that contains x and N_x represents the number of training samples in that leaf. This variance is also scaled by the Lebesgue measure of Ω_x . This interval would formalize a belief that

$$\text{var}_{\{x_i\}_{i=1}^n} \left(\hat{F}(x; \{x_i\}_{i=1}^n) \right) \sim \frac{\| \frac{d}{dx} F(x) \|^2}{NP(x)},$$

which is assumed to be approximately constant in each leaf. (3) is then a δ -method estimate of this quantity.

Despite the intuitive appeal of this approach, the variance of $F(x)$ with respect to resampling the training data depends crucially on the model assumptions and fitting procedures involved in producing F and such an estimate could be highly misleading.

8. EXAMPLE: BOSTON HOUSING DATA

We used the Boston Housing Data to create a representation of the underlying predictor space, leaving out a test set of size 50 for pruning purposes. The resulting tree is reproduced in Figure 7. Reported in the leaves of the tree are the means and variances of a 12-10-1 neural network trained on the data using the default values of the R package `nnet` [7]. Here it can be seen that the fourth leaf from the left contains almost all the data mass. We might raise concerns about predictions in the fifth and last leaves as having no data mass and relatively low values. The first leaf, too, is problematic in having very low data mass, yet a very high variance. Comparing CERT and Mahalanobis distance for these data in the same manner as §6 found that the probability of “outlier” misclassification differed by at most 0.004 for any value of Type 2 error.

9. CONCLUSIONS

CERT provides a new tool in diagnosing unusual points. We have defined a natural measure of extrapolation as being the relative likelihood of the data distribution versus a uniform distribution. This measure may be efficiently estimated with a variant of CART, and we have demonstrated that the inclusion of exact distributional information aids both the accuracy and stability of the resulting estimate.

This tool can now be used in several settings. To begin with, it is a straight forward diagnostic tool for high-dimensional covariance structures in a set of predictor variables. It also aids our understanding of function dynamics in areas of low data density. The measure can be used as an outlier-detection device; we reject any data point that is in a region of low probability. It is also a diagnostic for shifts in system dynamics. The mixture of products representation of the data density implied by CERT can be used to create diagnostic tools that are resistant to extrapolation [5].

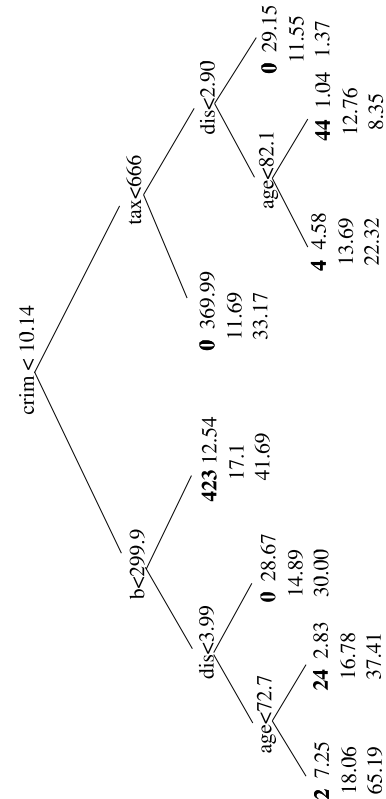


Figure 7: Tree-based density model for the Boston Housing Data with a 12-10-1 neural network.

Acknowledgements

The author would like to thank Jerome Friedman for comments and guidance and Bogdan Popescu for pointing out the credit agency example.

10. REFERENCES

- [1] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, 1984.
- [2] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.
- [3] D. Harrison and D. L. Rubinfeld. Hedonic prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5:81–102, 1978.
- [4] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Machine Learning: Data Mining, Inference and Prediction*. Springer, New York, 2001.
- [5] G. Hooker. Black box diagnostics and the problem of extrapolation: Extending the functional anova. Technical report, Stanford University, 2004.
- [6] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, 1993.
- [7] R-project. <http://www.r-project.org/>.