Large Datasets Lead to Overly Complex Models: an Explanation and a Solution

Tim Oates and David Jensen
Experimental Knowledge Systems Laboratory
Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610
{oates, jensen}@cs.umass.edu

Abstract

This paper explores unexpected results that lie at the intersection of two common themes in the KDD community: large datasets and the goal of building compact models. Experiments with many different datasets and several model construction algorithms (including tree learning algorithms such as C4.5 with three different pruning methods, and rule learning algorithms such as C4.5RULES and RIPPER) show that increasing the amount of data used to build a model often results in a linear increase in model size, even when that additional complexity results in no significant increase in model accuracy. Despite the promise of better parameter estimation held out by large datasets, as a practical matter, models built with large amounts of data are often needlessly complex and cumbersome. In the case of decision trees, the cause of this pathology is identified as a bias inherent in several common pruning techniques. Pruning errors made low in the tree, where there is insufficient data to make accurate parameter estimates, are propagated and magnified higher in the tree, working against the accurate parameter estimates that are made possible there by abundant data. We propose a general solution to this problem based on a statistical technique known as randomization testing, and empirically evaluate its utility.

Introduction

This paper presents striking empirical evidence that, with several popular model construction algorithms, more data is not always better. In particular, we show that increasing the amount of data used to build a model often results in a linear increase in model size, even when that additional complexity results in no significant increase in model accuracy. That is, large datasets may or may not yield more accurate models than smaller datasets, but they often result in models that are needlessly complex. The scope of this effect is explored with tree learning algorithms such as C4.5 with three different pruning methods and rule learning algorithms such as C4.5RULES and RIPPER. Equally surprising are empirical results showing that some of these algorithms build large models on datasets for which the class labels have been randomized. When there is absolutely no structure to be found in the data that would allow one to predict class labels, large models are still constructed, and the size of the model is still strongly dependent on the size of the dataset. Adding more data devoid of structure results in larger models; removing some of those data results in smaller models.

Given the empirical results mentioned above, this paper takes up two challenges: explaining the pathological relationship between dataset size and model size, and finding ways of building models that are not too large while retaining the benefits of large datasets (e.g. accurate parameter estimates). In the case of decision trees, the cause of the relationship between dataset size and tree size is identified as a bias inherent in several common pruning techniques. Pruning errors made low in the tree, where there is insufficient data to make accurate parameter estimates, are propagated and magnified higher in the tree, working against the accurate parameter estimates that are made possible there by abundant data. We propose a general solution to this problem based on a statistical technique known as randomization testing, and empirically evaluate its utility.

The Problem: Large Datasets and Excess Structure

In what way is model size dependent on dataset size? In particular, when adding more data fails to improve model accuracy, as we expect will happen with moderately large datasets, what does adding more data do to model size? We explored the answer to this question using three large datasets taken from the UC Irvine repository and three common model construction algorithms. The datasets are census-income (32,561 instances), led-24 (30,000 instances) and letter-recognition (20,000 instances). The algorithms include one decision tree learner, C4.5 (Quinlan 1993), and two rule learners, C4.5RULES (Quinlan 1993) and RIPPER (Cohen 1995b). Various decision tree pruning techniques have been developed with the explicit goal of eliminating excess structure in trees. Therefore, we ran C4.5 with three different pruning algorithms: error-based pruning (EBP – the C4.5 default) (Quinlan 1993), reduced error pruning (REP) (Quinlan 1987), and minimum description

length pruning (MDL) (Quinlan & Rivest 1989).

Plots of model size and accuracy for the led-24 dataset and C4.5 with REP, C4.5RULES (which takes as input unpruned trees) and RIPPER are shown below in Figure 1.[1] Model sizes reported for C4.5 are numbers of tree nodes, and sizes reported for C4.5RULES and RIPPER are numbers of conditions in rules. The number of instances used with C4.5 and RIPPER ranged from 0, in which case the best one can do is guess class labels randomly, to 20,000. The number of instances used with C4.5RULES ranged from 0 to 5,000 because larger datasets led to prohibitively long running times.

Note that all of the accuracy curves in Figure 1 reach their maximum value at fewer than 1,000 instances and thereafter remain almost constant. Shockingly, despite the fact that additional data results in no increase in accuracy, model sizes continue to grow dramatically. The ratio of the size of the tree built by C4.5 on the full dataset to the size of the tree built on 1,000 instances is 14.2. That ratio for RIPPER is 4.6, and for C4.5RULES it is 3.9. It is important to remember that C4.5RULES used a much smaller dataset than RIPPER, and had it been computationally feasible to run C4.5RULES on the full dataset there is every indication that its ratio would have been significantly larger. In no way do these results indicate that one of the rule learning algorithms is better than the other.

Another noteworthy feature of the graphs in Figure 1 is the apparently linear nature of the relationship between model size and dataset size beyond the point at which accuracy ceases to increase. For each combination of dataset and algorithm, that point was found by scanning the accuracy curve from left to right, stopping when the mean of three adjacent accuracy estimates was no more than 1% less than the accuracy of the model based on the full dataset. Running linear regression on the model size curves to the right of this point reveals that the relationship between model size and dataset size is often highly linear. The upper portion of Table 1 shows r^2, which can be interpreted as the fraction of variance in model size attributable to dataset size. In the majority of cases, more than 90% of the variance in model size is due to changes in dataset size. The lower portion of Table 1 shows the ratio of the size of the model built on the full dataset to the size of the model built on the smallest amount of data needed to achieve approximately maximum classification accuracy.

One possible explanation for the apparently pathological relationship between dataset size and model size is that as the amount of available data increases the algorithms are able to discern increasingly weak structure in the dataset. An alternative is that the algorithms are simply susceptible to fitting noise, and this problem is exacerbated by large datasets. These two views were explored by building models on datasets for which the class labels had been randomized, destroying any structure in the data that would allow one to predict class labels.

[1]Details on the experimental method and results for 19 datasets and 4 different pruning techniques can be found in (Oates & Jensen 1997).

r^2			
	census	led-24	letter
C4.5/EBP	0.98	1.00	1.00
C4.5/REP	0.98	0.98	0.99
C4.5/MDL	0.99	1.00	0.97
C4.5RULES	0.91	0.97	0.98
RIPPER	0.86	0.92	0.83
Size Factor			
	census	led-24	letter
C4.5/EBP	3.8	1.2	13.3
C4.5/REP	7.4	14.2	1.9
C4.5/MDL	4.2	19.7	1.3
C4.5RULES	2.1	3.9	1.3
RIPPER	3.5	4.6	1.1

Table 1: Summaries of the behavior of model size as a function of dataset size over the range of dataset sizes for which accuracy no longer increases.

Figure 2 shows plots of model size as a function of randomized dataset size for each of the algorithms and datasets. Despite the fact that the data are completely devoid of useful structure, EBP and MDL build trees with literally thousands of nodes on the led-24 and letter datasets after as few as 5000 instances. EBP does quite well on the census dataset, constructing very small trees, but MDL and REP perform poorly. Although a direct comparison of the sizes of trees and rule sets is not possible, the rule learning algorithms build very small models on all three datasets, and they appear to remain small as the size of the dataset increases. In fact, RIPPER produces empty rule sets, as it should, for the letter and led-24 datasets, and rule sets with fewer than ten conditions for the census dataset. RIPPER is the only algorithm that behaves as expected when presented with structureless data.

The Cause: Bias Propagation

Why would any of the algorithms explored earlier exhibit a pathological relationship between model size and dataset size? This section answers that question for C4.5 by identifying a bias inherent in all three pruning techniques that leads to a phenomenon that we call *bias propagation*. Developing an answer for the rule learning algorithms is left as future work. The discussion begins with an analysis of REP for concreteness and clarity, and is then generalized to include EBP and MDL.

REP builds a tree with a set of growing instances, and then prunes the tree bottom-up with a disjoint set of pruning instances. The number of classification errors that a subtree rooted at node N makes on the pruning set, $E_T(N)$, is compared to the number of errors made when the subtree is collapsed to a leaf, $E_L(N)$. If $E_T(N) \geq E_L(N)$, then N is turned into a leaf.

Note that $E_L(N)$ is independent of the structure of

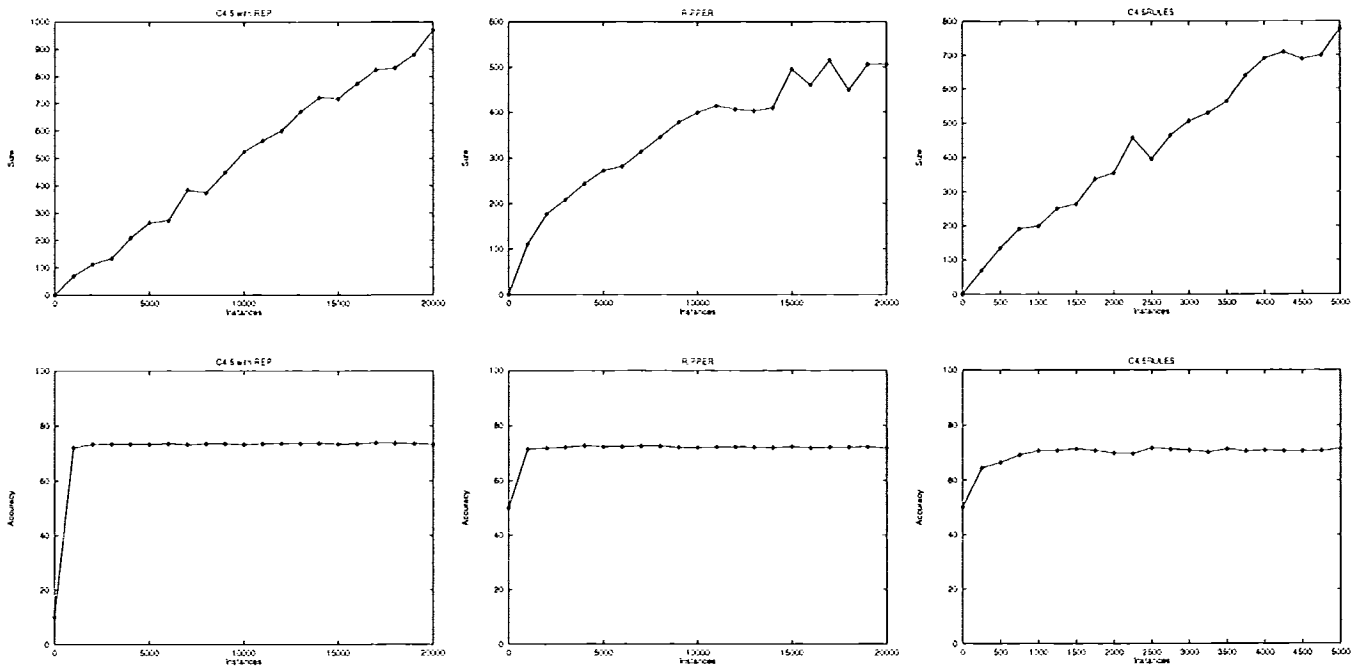

Figure 1: Plots of model size (upper row) and accuracy (lower row) on the led-24 dataset for C4.5 with REP (leftmost column), RIPPER (middle column) and C4.5RULES (rightmost column). Note that the scales of the model size plots are different.

the subtree rooted at N. To compute $E_L(N)$, all of the instances in the pruning set that match the attribute tests on the path from the root of the tree to N are treated as a set. The number of instances in this set that do not belong to the majority class of the set is the number of errors that the subtree would make as a leaf. For a given pruning set, $E_L(N)$ depends only on the structure of the tree above N, and therefore does not depend on how pruning set instances are partitioned by additional tests below N. As a consequence, $E_L(N)$ remains constant as the structure beneath N changes due to the effects of bottom-up pruning.

In contrast to E_L, $E_T(N)$ is highly dependent on the structure of the subtree rooted at N. $E_T(N)$ is defined to be the number of errors made by that subtree on the pruning set, and its value can change as pruning takes place beneath N. Consider a subtree rooted at N', where N' is a descendant of N. If $E_T(N') < E_L(N')$ then N' is not pruned, and because the structure beneath N remains unchanged, $E_T(N)$ also remains unchanged. The alternative is that $E_T(N') \geq E_L(N')$, in which case N' is turned into a leaf. This structural change either causes $E_T(N)$ to remain unchanged (when $E_T(N') = E_L(N')$) or to decrease (when $E_T(N') > E_L(N')$).

E_L and E_T can be used to estimate the error rate of a subtree, as a leaf and as a tree respectively, on the population of instances from which the pruning set was drawn. Each time pruning occurs beneath N, $E_L(N)$ remains invariant and $E_T(N)$ usually decreases. This systematic deflation of E_T, a statistical bias inherent in REP, produces two effects: (1) pruning beneath N increases the probability that $E_T(N) < E_L(N)$ and that N will therefore not be pruned; (2) E_T for the final pruned tree tends to be an underestimate. These effects should be larger for large unpruned trees, because they afford many opportunities to prune and to deflate E_T. These effects should also be larger for small pruning sets because they increase the variance in estimates of E_L and E_T. Even when a node is more accurate as a tree than as a leaf on the population, highly variable estimates make it more likely that, by random chance, $E_T(N) \geq E_L(N)$ and the subtree rooted at N will be pruned, thereby lowering E_T for all parents of N. Likewise, even when a node is more accurate as a leaf than as a tree, it is more likely that, by random chance, $E_T(N) < E_L(N)$, resulting in no change in E_T for the parents of N and the retention of the structure beneath N. In either case, the net result is larger trees, either from the explicit retention of structure or systematic deflation of E_T which often leads to the retention of structure higher in the tree.

What are the features, at an abstract level, that lead to bias propagation in REP? First, each decision node in a tree is assigned two scores for the purpose of pruning: the score of the node as a tree, $S_T(N)$, and the score of the node as a leaf, $S_L(N)$. For REP, these scores correspond to errors on the pruning set. Second, the disposition of the node is chosen to either maximize or minimize its score. When scores are equiva-

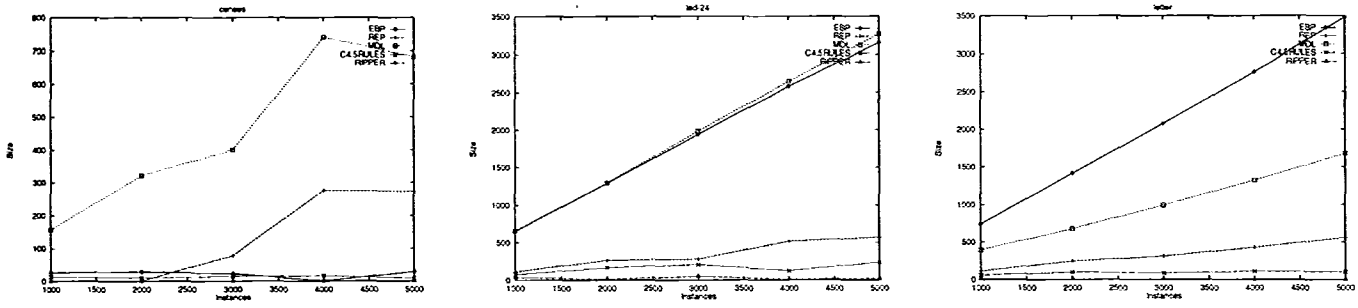

Figure 2: Model size as a function of dataset size for datasets with randomized class labels.

lent to errors, lower scores are better and pruning occurs when $S_L(N) \leq S_T(N)$. Other scoring functions may treat high scores as better and prune nodes when $S_L(N) \geq S_T(N)$. (The effects of consistently choosing the maximum or minimum of a set of values on the sampling distribution is discussed in detail in (Jensen & Cohen 1998).) Third, the score of a node as a tree (S_T) is directly dependent on the scores of all of that node's children. Finally, pruning proceeds bottom-up. The net effect is that pruning decisions made below a node in the tree serve to bias the score of that node as a tree, regardless of whether those pruning decisions were correct, making the node's subtree look increasingly attractive for retention.

Both MDL and EBP, in addition to REP, satisfy all of the requirements for bias propagation. MDL is a bottom-up pruning technique that chooses to prune when the description length of a node is smaller as a leaf than as a tree, and the number of bits required to encode a tree is directly related to the number of bits required to encode the children of the tree's root. EBP is a bottom-up pruning technique that chooses to prune when the estimated number of classification errors that a node will commit as a leaf is smaller than as a tree, and the error estimate for a node is directly related to the error estimates for that node's children.

The Solution: Randomization Pruning

For each decision node, N, in a pruned decision tree, $S_T(N) > S_L(N)$ (assuming that we are maximizing S); otherwise, N would have been pruned back to a leaf. However, experiments with randomized data showed clearly that $S_T(N)$ can be greater than $S_L(N)$ even when the subtree rooted at N is fitting noise, and the previous section identified bias propagation as the cause of this problem. Ideally, we would like to retain decision nodes only when their scores are high because they root subtrees that represent structure in the data, not when their scores are artificially inflated due to bias propagation. Stated in terms of statistical hypothesis testing, our null hypothesis (H_0) is that the class label is independent of the attributes in the data that arraive at a node. We would like the probability of obtaining $S_T(N)$ under H_0 to be low for all of the decision nodes in the pruned tree.

Randomization testing is a statistical technique that constructs an empirical distribution of a statistic under H_0, making it possible to determine the probability of obtaining a value larger (or smaller) than any particular value for that statistic given that H_0 holds (Cohen 1995a; Edgington 1995; Jensen 1991; 1992). For example, consider the null hypothesis stated above. For any node, N, in a decision tree, we want to determine the probability of obtaining a score greater than or equal to $S_T(N)$ given H_0. If that probability is low, we can be confident that the subtree rooted at N has captured structure in the data. If that probability is high, the subtree might be fitting noise due to bias propagation.

To construct an empirical distribution of $S_T(N)$ under H_0, we repeat the following procedure K times. Collect all of the data that arrive at node N, randomize the class labels of that data, build and prune a tree on the randomized data, and record S_T for that tree. Randomizing the class labels enforces H_0. Given a sample of values of S_T of size K obtained in this manner, we can estimate the of obtaining a value at least as high as $S_T(N)$ based on the original (not randomized) data by counting the number of values in the sample that are greater than or equal to $S_T(N)$ and dividing by K.

This simple application of randomization testing was used as the basis for *randomization pruning*, an algorithm for eliminating excess structure from trees built on large datasets. Randomization pruning simply walks over a pruned tree and at each node, N, computes the probability that a score at least as high as $S_T(N)$ could have been obtained under H_0. If that probability is above some threshold, α, then N is pruned back to a leaf. That is, if N was retained simply because $S_T(N)$ was artificially inflated as a result of bias propagation, then we eliminate N from the tree.

Figure 3 shows the results of applying randomization pruning to EBP with $\alpha = 0.05$. Each of the three columns represents a dataset, the plots in the upper row are size, and the plots in the lower row are accuracy. Each plot shows results for standard EBP and for EBP enhanced with randomization pruning. All curves were generated by randomly selecting 5,000 instances from the dataset, retaining all of the remaining instances for testing accuracy, and building trees on subsets of vari-

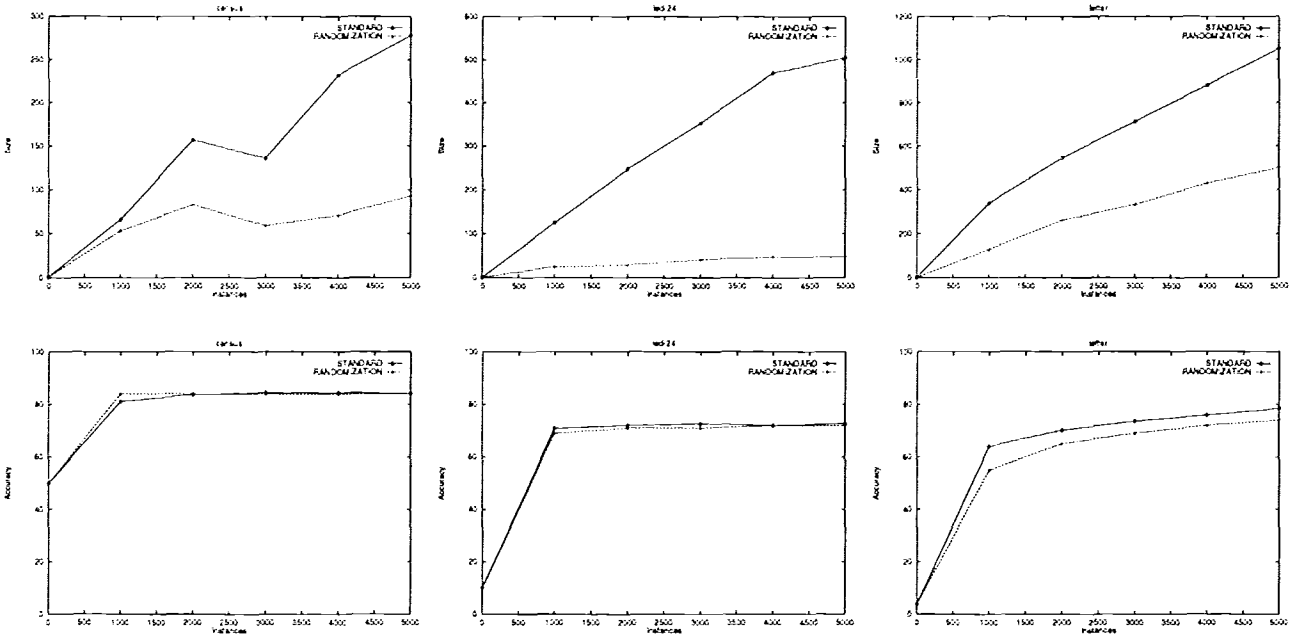

Figure 3: The effects of randomization pruning on tree size and accuracy.

ous size of the 5,000 instances. That is, the points in the curves are not means. Notice that randomization pruning produced dramatically smaller trees for all three datasets, with the size of the reduction increasing with the size of the dataset. There was no appreciable impact on accuracy for census and led-24, and there was a loss of accuracy of 5.4% on average for letter. Unlike the other two datasets, accuracy on letter increased over all dataset sizes in Figure 3.

Conclusion

Experiments with several different datasets and model construction algorithms showed that increasing the amount of data used to build models often results in a linear increase in model size, even when that additional complexity results in no significant increase in model accuracy. Additional experiments produced the surprising result that some algorithms (notably C4.5) construct huge models on datasets that are devoid of structure. Based on this observation, the cause of the pathological relationship between dataset size and tree size was identified as a bias inherent in all three decision tree pruning techniques. We proposed randomization pruning, an algorithm than can be wrapped around existing pruning mechanisms, as a mechanism for counteracting bias propagation, and evaluated its utility.

Future work will involve additional experimentation with randomization pruning, including extending the implementation to wrap around REP and MDL and running it on additional datasets. Also, we want to determine when and why rule learning algorithms fail to appropriately control rule set growth as the size of the dataset increases.

References

Cohen, P. R. 1995a. *Empirical Methods for Artificial Intelligence.* The MIT Press.

Cohen, W. W. 1995b. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning.*

Edgington, E. S. 1995. *Randomization Tests.* Marcel Dekker.

Jensen, D., and Cohen, P. R. 1998. Multiple comparisons in induction algorithms. To appear in Machine Learning.

Jensen, D. 1991. Knowledge discovery through induction with randomization testing. In *Proceedings of the 1991 Knowledge Discovery in Databases Workshop,* 148–159.

Jensen, D. 1992. *Induction with Randomization Testing: Decision-Oriented Analysis of Large Data Sets.* Ph.D. Dissertation, Washington University.

Oates, T., and Jensen, D. 1997. The effects of training set size on decision tree complexity. In *Proceedings of The Fourteenth International Conference on Machine Learning,* 254–262.

Quinlan, J. R., and Rivest, R. 1989. Inferring decision trees using the minimum description length principle. *Information and Computation* 80:227–248.

Quinlan, J. R. 1987. Simplifying decision trees. *International Journal of Man-Machine Studies* 27:221–234.

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