Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics, Ft. Lauderdale, FL: Morgan Kaufmann (1999).

Modeling Decision Tree Performance with the Power Law

Lewis J. Frey & Douglas H. Fisher, Jr.
Computer Science Department
Vanderbilt University
Village at Vanderbilt
Nashville, TN 37235

Abstract

This paper discusses the use of a power law to predict decision tree performance. Power laws are fit to learning curves of decision trees trained on data sets from the UCI repository. The learning curves are generated by training C4.5 on different size training sets. The power law predicts diminishing returns in terms of error rate as training set size increase. By characterizing the learning curve with a power law, the error rate for a given size training set can be projected. This projection can be used in estimating the amount of data needed to achieve an acceptable error rate, and the cost effectiveness of further data collection.

1 INTRODUCTION

This paper examines the idea of projecting the error rate of decision trees from a portion of a data set. To do this a power function ($P=AS^{-b}$) is fit to the error rate performance of decision trees generated with C4.5 (Quinlan, 1993) from different sized training sets. In the power function, S is the size of the training set and P is the error rate performance. A and b are best fitting parameters. The power function predicts diminishing returns in error rate for increasing training set size. By characterizing the learning curve with a power function, the error rate for a given size training set can be projected. This projection can be used to estimate the amount of data that needs to be collected to achieve an acceptable level of error rate.

The motivation to use a power law to examine decision tree performance was in part due to the ubiquity of the power law. The power law occurs frequently in learning, including human learning performance. Anderson and Schooler (1991) review a number of paradigms in which a power law appears to describe human performance. For example, the power law of practice is a phenomenon in which the performance measure (e.g., response time, error rate) on a task is related by a power function to the amount of time spent practicing. Thus, the amount of time it takes to memorize a list, for example, can be predicted by the function $P=AS^{-b}$ where S is the amount of time spent in practice and P is a performance measure.

In machine learning the power law has been used to estimate performance of single and multi-layer networks (Cortes, Jackel, Solla, Vapnik, and Denker, 1995; Cortes, Jackel, and Chiang, 1995). If the power law holds for decision trees in practice, it can be used to predict the number of examples needed to achieve a particular error rate.

Coupled with a result of Oates and Jensen (1997), the power law can also be used to predict the size of the tree necessary to give a particular error rate. Oates and Jensen tested the hypothesis that the size of trees generated with C4.5 using an error-based pruning method (i.e., default pruning) grows linearly with training set size. This occurs even though error rate levels off. Oates and Jensen suggest reducing the training set to the size where error rates level off. Using a power law projection, a smaller decision tree can be generated with equivalent error rates to larger trees, based on the power law estimate of the error rate for a given training set size.

2 METHODS

The hypothesis being examined is whether the error rate of a pruned decision tree generated by C4.5 can be predicted by a power law defined on the size of the training set. The error rate is the percentage of instances incorrectly predicted by the tree. The pruning method used is error based pruning which is the default pruning method for C4.5. The error rates for the pruned trees are determined with different sizes of training sets using incremental k-fold cross-validation (Cohen, 1995).

In k-fold cross-validation, a data set, D, of size n is partitioned into k sets. Each of these D_i subsets (i.e., $1<= i <= k$) is the same size n/k. Before the data set is segmented into the k-folds, it is randomized. The ith tree is built on the training set (D - D_i) which is the difference between the entire set and the subset D_i. The tree is then tested on the subset D_i. This is done for each i. The resulting k trees are averaged to give the average size and error rate of the trees over all k-folds.

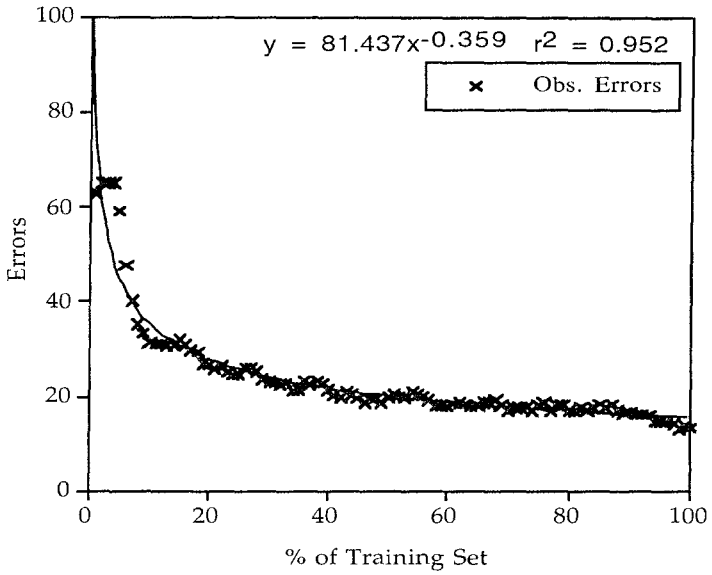

Figure 1. Changes in the percentage of errors for the *tic-tac-toe* data set across 1% increments in training set size. The average error rates of the 10-fold cross validation are fit with the power law function displayed in the upper right corner. The power function (thin line) is plotted along with the average observed error rates (x).

In the incremental cross-validation method (Cohen, 1995), a range of data set sizes are used for each *k*-fold to build the decision tree. In this paper, *k* equals 10. For each 10-fold, training set sizes increase in increments of 1%. Consequently, training sets increasing from 1% to 100% are used to build the trees. The different training set sizes are sequentially sampled from the training set (i.e., D - D_i). The test set of the previous run is placed at the beginning of D - D_i, causing a different sample ordering each run. The results for each different training set size are averaged across the 10-folds.

Although this paper is mainly focused on the power law function because of its ubiquity across diverse learning environments, three other types of functions (linear, logarithmic, and exponential) are compared. Oates and Jensen (1997) have shown that under some circumstances (i.e., decision tree size) a linear model provides a reasonable account. However, diminishing returns are typically observed in error rate performance so, the linear model is not expected to perform well. The other two models (logarithmic and exponential) should provide reasonable alternatives to the power law. Each of the four model classes has two free parameters. The four functions, a power, a linear, a logarithmic, and an exponential, are fit to the change in error rate across training set size.

The error rate fit is measured in terms of r^2 values and chi-squared (χ^2), a goodness of fit measure. Fourteen data sets, thirteen from the UCI repository and one generated (i.e., *parity*), are examined (for list see Table 1). The *parity* function is defined on seven binary attributes and is true if and only if an odd number of the attributes have the value true. A greedy decision tree learner is not going to capture the regularity in the data set so a decreasing error rate curve is not expected. The intent of including this data set was to show a data set that was difficult for the power law to model.

3 RESULTS

The results suggest that the error rate of a pruned decision tree generated by C4.5 can be predicted by a power law. Figure 1 displays changes in error rates across training set size for the *tic-tac-toe* data set. Each point is the average error rate of the 10-fold cross validation. Error rates are plotted across training set size (depicted by percentage of training set size). Since the training set size was incremented by 1%, going from 1% to 100% of the full training set (D-D_i), there are a hundred points. The power law (thin line) is plotted along with the learning curve of the observed data (i.e., x). The best fitting power law ($y = 81.437\ x^{-0.359}$) and its r^2 of 0.952 are displayed in Figure 1.

The error rates for the decision trees result from different sizes of training sets. These error rates show a power law behavior (see Table 1). The power law is able to account for the largest amount of the data variability (i.e., r^2) compared to a linear, logarithmic, or exponential fit. Of the four models, the power law is the best fit for the data sets in Table 1, except for the *heart* data set in which the fit is equal to a logarithmic fit, and the *parity* data set which both the linear and the exponential fits are better than the power fit.

Table 1. Power, linear, logarithmic, and exponential functions are fit to the **percentage of errors** across **training set size** for the average of the 10-fold cross validation. The equation and measure of fit (i.e., r^2) for each data set are presented for each function. The largest r^2 for each data set is indicated by a bold font.

Data Sets	Power	Linear	Logarithmic	Exponential
breast-cancer n = 286	$y=52.19x^{-0.145}$ **$r^2=0.64$**	$y=-0.148x+38.84$ $r^2=0.41$	$y=-12.48\log(x)+51.06$ $r^2=0.56$	$y=38.15*10^{-0.002x}$ $r^2=0.53$
glass n = 214	$y=90.79x^{-0.226}$ **$r^2=0.90$**	$y=-0.290x+55.62$ $r^2=0.57$	$y=-25.92\log(x)+81.91$ **$r^2=0.87$**	$y=55.12*10^{-0.003x}$ $r^2=0.71$
heart n = 155	$y=47.56x^{-0.154}$ **$r^2=0.82$**	$y=-0.127x+33.91$ $r^2=0.54$	$y=-11.24\log(x)+45.28$ **$r^2=0.82$**	$y=33.67*10^{-0.002x}$ $r^2=0.61$
hypothyroid n = 3163	$y=6.29x^{-0.463}$ **$r^2=0.83$**	$y=-0.022x+2.49$ $r^2=0.45$	$y=-2.03\log(x)+4.56$ $r^2=0.71$	$y=2.234*10^{-0.006x}$ $r^2=0.62$
iris n = 150	$y=80.94x^{-0.632}$ **$r^2=0.91$**	$y=-0.257x+23.75$ $r^2=0.35$	$y=-27.04\log(x)+53.51$ $r^2=0.76$	$y=18.57*10^{-0.007x}$ $r^2=0.59$
kr-vs-kp n = 3196	$y=85.86x^{-1.117}$ **$r^2=0.94$**	$y=-0.175x+12.88$ $r^2=0.22$	$y=-19.98\log(x)+35.62$ $r^2=0.56$	$y=7.41*10^{-0.014x}$ $r^2=0.75$
labor-neg n = 57	$y=63.94x^{-0.350}$ **$r^2=0.69$**	$y=-0.406x+33.35$ $r^2=0.38$	$y=-24.20\log(x)+55.36$ **$r^2=0.67$**	$y=31.20*10^{-0.006x}$ $r^2=0.43$
lymphography n = 141	$y=54.82x^{-0.201}$ **$r^2=0.78$**	$y=-0.163x+35.29$ $r^2=0.43$	$y=-15.57\log(x)+51.66$ **$r^2=0.75$**	$y=34.17*10^{-0.002x}$ $r^2=0.49$
parity n = 128	$y=39.00x^{0.118}$ $r^2=0.71$	$y=0.242x+48.26$ **$r^2=0.83$**	$y=15.94\log(x)+35.32$ $r^2=0.70$	$y=48.83*10^{0.002x}$ **$r^2=0.83$**
sickeuthyroid n = 3163	$y=10.20x^{-0.350}$ **$r^2=0.65$**	$y=-0.039x+5.19$ $r^2=0.30$	$y=-4.12\log(x)+9.71$ **$r^2=0.62$**	$y=4.26*10^{-0.003x}$ $r^2=0.32$
soybean,large n = 683	$y=150.54x^{-0.640}$ **$r^2=0.98$**	$y=-0.422x+39.88$ $r^2=0.49$	$y=-40.57\log(x)+82.65$ $r^2=0.88$	$y=37.14*10^{-0.008x}$ $r^2=0.79$
tic-tac-toe n = 958	$y=81.437x^{-0.359}$ **$r^2=0.95$**	$y=-0.281x+37.88$ $r^2=0.56$	$y=-25.43\log(x)+63.84$ $r^2=0.89$	$y=37.05*10^{-0.004x}$ $r^2=0.76$
vote n = 435	$y=26.99x^{-0.458}$ **$r^2=0.86$**	$y=-0.103x+11.23$ $r^2=0.24$	$y=-11.48\log(x)+24.14$ $r^2=0.58$	$y=9.24*10^{-0.005x}$ $r^2=0.56$
vote 1 n = 435	$y=33.88x^{-0.275}$ **$r^2=0.83$**	$y=-0.118x+19.01$ $r^2=0.37$	$y=-11.56\log(x)+31.33$ $r^2=0.69$	$y=18.20*10^{-0.003x}$ $r^2=0.61$

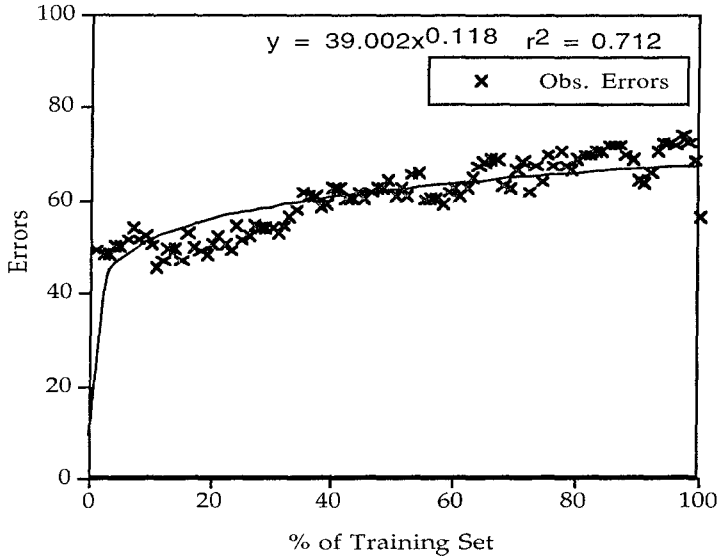

Figure 2. Changes in the percentage of errors for the *parity* data set across training set size. The average error rates of the 10-fold cross validation are fit with the power law function displayed in the upper right corner. The power function (thin line) is plotted along with the average observed error rates (x).

The *parity* function was fit equally well by the linear and exponential functions. The observed error rates are plotted in Figure 2. The error rate grows with increased training set size. The power law is also plotted (thin line), and it projects a higher error rate for larger training set size.

Although we do not present the results, these experiments also verify Oates and Jensen's finding that tree size grows linearly with training set size, regardless of error rate.

4 PROJECTION

Knowing that decision tree learning curves can be modeled by a power law suggests that a power law fit to a small portion of data can be used to estimate the error rate for decision trees learned on a larger amount of data. This can be of value if there is a cost associated with collecting data. A projection can be made using a small portion of data, and the projected learning curve can be used to decide what would be gained from collecting more data. The projection can also be used to estimate the smallest training set size to achieve a desired level of error rate that gives the additional benefit of smaller decision trees.

Figure 3 depicts the power, linear, logarithmic, and exponential fits obtained from only 15% of the training set of the *tic-tac-toe* database. The projected power law values estimate the error rates of the rest of the observed values (i.e., 85% of the remaining data) within 3%. The other three functions can be seen to diverge.

This procedure is used for all the goodness of fit significance tests, χ^2, presented in Table 2. The functions are fit to the first fifteen points (i.e., 15% of the data). The tails (i.e., the last 85%) of the data sets are compared so that the first 15% used to estimate the functions are excluded from the comparisons. The χ^2 results are displayed in Table 2. The fits are calculated relative to the same observed learning curves used for Table 1. The four functions are the best fitting curves to the first 15% of the data. The derived functions are then used to estimate the remaining 85% of the data.

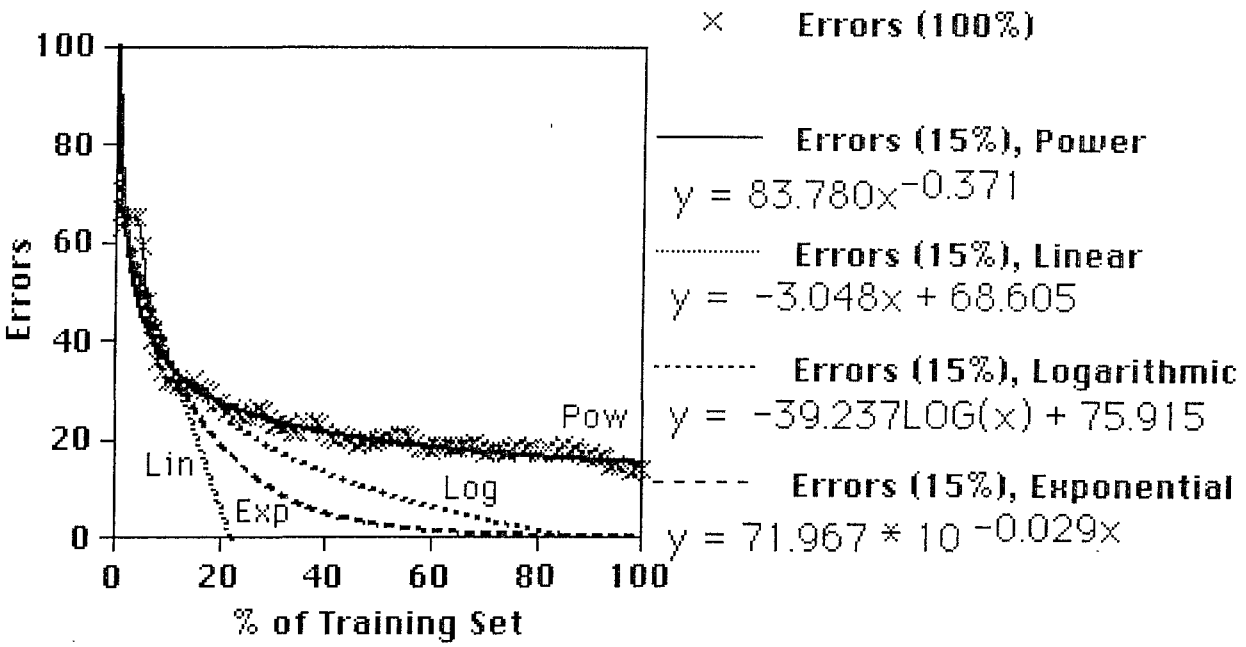

Figure 3. Changes in the percentage of errors for the *tic-tac-toe* data set across training set size. The average observed error rates (x) of the 10-fold cross validation for 15% of the data are fit with the power (Pow, solid line), linear (Lin, dotted line), logarithmic (Log, medium dashed line), and exponential (Exp, large dashed line) functions. The functions for 15% of the data are fit using the first fifteen points.

Table 2. Chi-squared (χ^2) values for the last 85% of the error rate data for each data set across the four types of fits are presented. The first 15% of the error rate data are fit to the four functions. The tails of 85% of the remaining unfit data are used to compare the projections made by each of the functions. Bold font and an '*' indicate a significant χ^2. A significant χ^2 indicates that the data were **not** generated by the function with 99.9% confidence. The critical value of χ^2 is 125.17 with an alpha of 0.001. There are 83 degrees of freedom (df). df = N-M degrees of freedom, where N = 85 = number of data points (note N = 43 for labor-neg) and M = 2 = number of parameters.

Data Sets	Power	Linear	Logarithmic	Exponential
breast-cancer	12.65	**26088.50***	53.45	**1464.15***
glass	53.26	**31303.86***	**467.03***	**912.39***
heart	43.85	**11739.47***	**195.39***	**876.48***
hypothyroid	56.68	**23630.78***	91.65	**163.27 ***
iris	7.65	**250652.81***	**3951.32***	**152.10 ***
kr-vs-kp	63.14	**25582478.00***	**557237.56***	**269.95***
labor-neg (df = 41, cv = 74.23)	1.87	**2668.61***	15.39	55.00
lymphography	11.80	**8306.75***	66.68	**300.79***
parity	**191.44 ***	**390.71***	**187.70***	**339.45***
sickeuthyroid	34.57	**73760.48***	**291.24***	**459.07***
soybean large	35.36	**93469.86***	**1573.49***	**742.11***
tic-tac-toe	7.92	**146103.91***	**1095.61***	**1877.59***
vote	43.22	**119159.34***	**4040.04***	**156.76 ***
vote1	67.29	**46611.88***	**1302.54***	**507.88***

Table 3. The average absolute difference between the observed and the projected error rates (|Observed - predicted|) for the last 85% of the error rate data for each data set across the four types of fits are presented. The standard deviations of these absolute differences are reported within the parentheses. For each data set, the smallest average absolutedifference and standard deviation are indicated by a bold font. The first 15% of the error rate data are fit to the four functions. The tails of 85% of the remaining unfit data are used to compare the projections made by each of the functions.

Data Sets	Power		Linear		Logarithmic		Exponential	
breast-cancer	**2.02**	**(1.14)**	78.85	(42.72)	4	(1.72)	21.5	(5.96)
glass	**6.79**	**(2.71)**	149.43	(81.19)	20.15	(6.85)	29.33	(6.94)
heart	**4.06**	**(1.84)**	61.93	(33.63)	8.62	(3.1)	18.45	(5.32)
hypothyroid	**0.41**	**(0.14)**	8.21	(4.97)	0.5	(0.25)	0.81	(0.22)
iris	**1.24**	**(0.79)**	195.38	(108.46)	25.69	(11.16)	5.92	(1.26)
kr-vs-kp	**0.54**	**(0.26)**	193.53	(104.08)	32.63	(11.54)	1.17	(0.62)
labor-neg	**2.3**	**(1.83)**	92.36	(55.55)	6.89	(4.55)	14.76	(4.86)
lymphography	**3.6**	**(2.08)**	97.7	(54.4)	8.87	(4.32)	19.62	(5.07)
parity	13.48	(6.76)	19.32	(9.63)	**13.34**	**(6.73)**	18	(8.96)
sickeuthyroid	**0.59**	**(0.27)**	25.7	(14.95)	1.66	(0.89)	2.22	(0.48)
soybean large	**3.72**	**(0.95)**	216.16	(122.73)	22.17	(10.66)	9.68	(2.09)
tic-tac-toe	**0.85**	**(0.54)**	128.23	(71.78)	11.46	(5.01)	15.56	(3.01)
vote	**2.12**	**(0.53)**	97.89	(52.65)	19.66	(6.42)	4.14	(0.63)
vote1	**3.38**	**(1.36)**	79.95	(43.33)	14.77	(4.98)	9.65	(2.4)

The chi-squared values are calculated using the observed data, the observed variances, and the predicted data. The null hypothesis is that the observed and predicted data do not differ. If the chi-squared value is greater than the critical value using an alpha of 0.001, the likelihood that the data were generated with the model is equal to or less than 0.001%. A chi-squared value less than the critical value does not ensure that the observed data set was generated by the model. It suggests that it is a possible model of the data set.

The results in Table 2 show that the power laws based on 15% of the observed data do not differ significantly from the predicted 85% of the error rate data for thirteen out of the fourteen data sets. The linear functions were significantly different for all the data sets. The observed and predicted logarithmic functions were with high confidence indistinguishable for four of the fourteen data sets. The exponential function was significantly different from the observed data for all but *labor-neg*, which had relative few data points and a large standard deviation.

The power law did fail at predicting *parity*. But *parity* is expected to have a poor fit based on a small subset of the data because its error increases with increasing data set size (see Figure 2).

In Table 3 the mean values of the absolute difference between the observed and projected error rates are shown for the four models. The standard deviations for these absolute differences are also displayed. As can be seen from Table 3, the power laws have smaller average absolute difference values for all the data sets except *parity*. The power laws also have small standard deviations. Excluding *parity*, the average absolute differences range between 0.41% and 6.79% for power law. Compare this to the linear which ranges from 8.21% to 216.16%, the logarithmic which ranges from 0.5% to 32.63%, and the exponential which ranges from 0.81% to 29.33%. Coupled with the chi squared goodness of fit test, the average absolute difference scores suggest that the power law is a better fit than the linear, logarithmic, and exponential models.

The predictive ability of the power law can also fail if an insufficient amount of data is used for the projection. This is demonstrated in Figure 4 when only 307 *soybean* instances are used. The power law over estimates the error given 15% of the data. If 25% of the data were used, the power law would have a much better fit. Undoubtedly this is a result of *soybean* having nineteen classes which requires more instances to properly learn the classification. With too small a training set, the classes may not be adequately represented.

In general the under sampling of the *soybean* data raises the question of when do you know that enough data have been used in making a projection for error rates of larger training set size. More work is needed to answer this question, but an initial direction would be to measure the stability of the power law being used for the projection. If increasing the size of the data set does not change the projection significantly this may be an indication that the fit of the power law will not greatly improve with additional data.

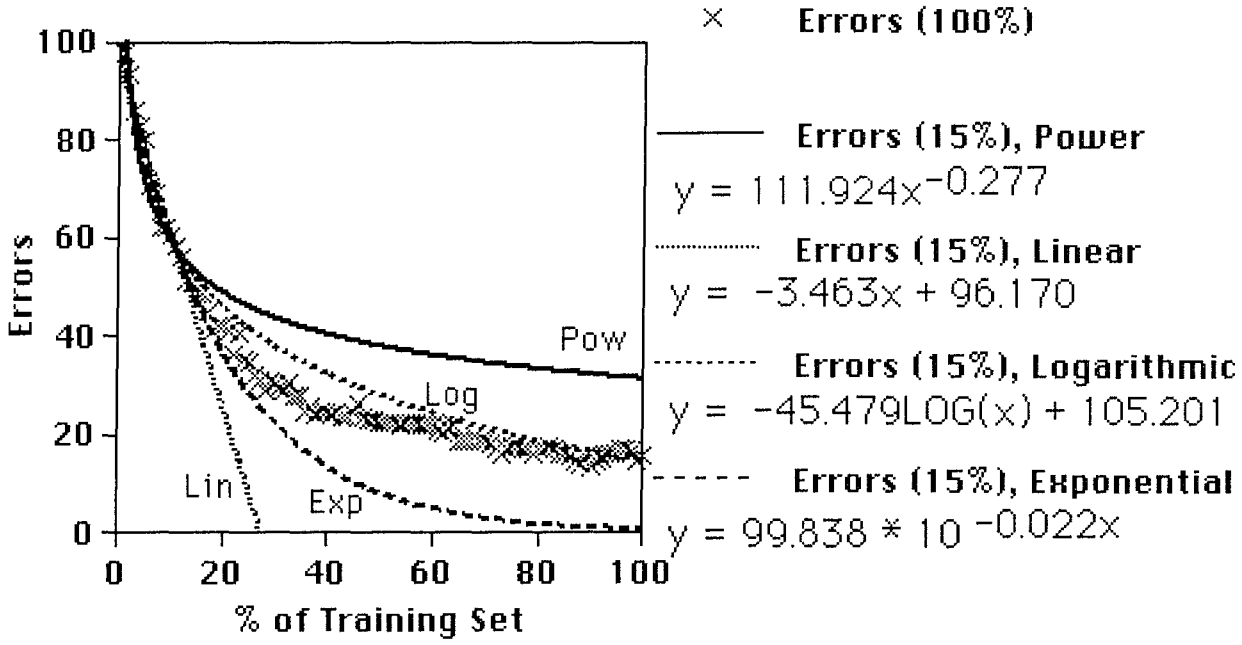

Figure 4. Changes in the percentage of errors for the *soybean* data set across training set size if only 307 instances are used. The average error rates (x) of the 10-fold cross validation for 15% of the data are fit with the power (Pow, solid line), linear (Lin, dotted line), logarithmic (Log, medium dashed line), and exponential (Exp, large dashed line) functions. The functions for 15% of the data are fit using the first fifteen points.

5 CONCLUSION

The power law provides the best fit for the error rates for thirteen out of fourteen data sets. This suggests that the decision tree error rates follow a power law with diminishing returns for increased training set size. The more training examples used the less of an improvement in error rate. Being able to model the error rate with a power law suggests the possibility of a principled stopping criterion. For example, stop increasing the training set size when the power law predicts insufficient gains in error rate.

Oates and Jensen suggest a criterion to restrict the size of the training set based on the mean of three adjacent error rates being within 1% of the error rate of the tree trained on all training data. Since they found that tree size increases linearly with training set size without regard to the error rate, their criterion produced smaller trees for many of the data sets they examined. A power law criterion could also be used to reduce the decision tree size by reducing the training set size. This could be done by training with a data set size for which the power law predicts an error rate acceptable to the user. More generally we can use a trade-off function of error rates projected by the power law, tree size, and the cost of collecting subsequent data, to decide the size of the training set.

Acknowledgments

The breast cancer and lymphography data sets were provided by M. Soklic and M. Zwitter, University Medical Center, Institute of Oncology, Ljubljana, Solvenia.

References

Anderson, J. R. and Schooler, L. J.. (1991). Reflections of the environment in memory. Psychological Science 2: 396-408.

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

Cortes, Jackel, Chiang (1995) Limits on learning machine accuracy imposed by data quality. KDD-95 and NIPS-7.

Cortes, Jackel, Solla, Vapnik, Denker (1994) Learning curves: Asymptotic values and rates of convergence. NIPS-6.

Oates, T. and Jensen, D. (1997). The Effects of Training Set Size on Decision Tree Complexity. In D. H. Fisher, Jr. (Ed.), Proceedings of the XIV International Conference (pp. 254-262). Morgan Kaufmann.

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