Modeling Performance of Different Classification Methods: Deviation from the Power Law

Sameer Singh
CS 362: Machine Learning
Department of Computer Science
Vanderbilt University, USA

***Abstract** – This project studied the effect of varying the training size for different classification techniques. The learning curves were then regressed using four common equations. In the restricted domain which was studied, the logarithmic equation was the best fit. This contradicts the earlier work carried out on Decision Trees in which the performance was best modeled by the Power Law. The other classification techniques studied in this project were K-Nearest Neighbors, Support Vector Machines, and Artificial Neural Networks, which have not yet been included in such a study. A preliminary study of how the modeling can be used for predicting the performance of the project was also undertaken. The equations which best predicted the performance were not same as the ones which best fit the final curve, and depended on the classification method more than the dataset.*

1. Introduction

This project attempts to model the performance of different kinds of classification techniques in terms of the training set size. For all classification techniques, the performance on a test set improves as the set it is trained on increases. Also, this improvement in accuracy, or conversely, the decrease in the error on the test set, is known to have diminishing returns. This project studies these error curves, and examines if they can effectively be modeled using simple equations for various classification techniques. It also attempts to study the utility of the modeling, and discover if it can be used for prediction of performance.

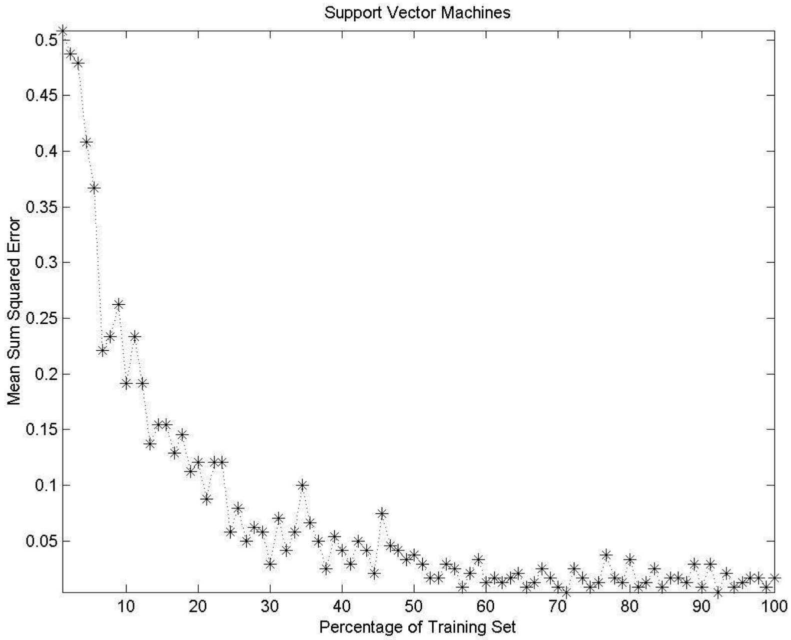

***Figure 1:** A typical learning curve for a classification technique. The SVM classification was used on the MUSK Clean1 dataset. This curve demonstrates the diminishing returns as the training size is increased.*

A typical error curve of a learning system is shown in Figure 1. This project attempted to fit this curve using four different type of parameterized equations, namely linear ($y=ax+b$), logarithmic ($y=a\log x+b$), exponential ($y=ba^x$) and power law ($y=bx^a$). Since most work on modeling decision trees considers power law as the best fit (See Section 6), it was expected the other classification techniques would follow power law too.

The following section talks about the motivation behind the project. Section 3 describes the method of the project. Section 4 gives the details of the classification methods and the datasets which were used. Section 5 describes the results of the project, followed by a discussion in Section 6. Section 6 shall also review the work in these field, and compare out results to them. The future plans are listed in Section 7.

2. Motivation

This project is most applicable in scenarios where either procuring data is very expensive, or the dataset size is very large. In these circumstances, the major problem is to determine how well a learning system would perform on the complete dataset, by examining only a small subset. If only the accuracy on the dataset is observed, it could lead to wrong decision since a different learning system may asymptote to a smaller value. This could be the main source of error in landmarking techniques of meta-learning, which chooses the learning method based on the performance of a number of basic learning techniques on the complete dataset. Landmarking also fails when only a subset of the complete dataset is used for the meta-learning process.

A secondary application of modeling the classification methods could be in predicting the amount of data needed for training. Based on a given subset and an error tolerance, the minimum amount of patterns needed to achieve the tolerable error may be predicted. This can be used for situations where the training and the testing processes are disjoint.

3. Method

3.1 Classification Methods

Four different learning techniques were examined in this project. A lot of work has been done on decision trees in this respect ([Frey], [Gu], and Section 5) which are modeled using the power law. Along with decision trees, three other classification techniques were chosen. K-Nearest neighbor is one of the simplest classification techniques which decides the class based on the average class of the k nearest neighbors of the current input. It was chosen to represent a naïve method of classification. Artificial Neural Networks were chosen since they form the basic connectionist classification technique, and a similar trend in them would signify a generic characteristic of classification techniques, instead of that only in ones which are tree based. Finally, Support Vector Machines are a relatively newer method of classification, and one which has been very successful is dealing with some problems very difficult for traditional classification techniques. Modeling them would give a better insight into their working [Forman].

3.2 Approach

The method for this project is similar to the one presented in [Frey]. The complete data set is represented by D. For each of the datasets, we start with a fixed testing set size S_{test}, set to 10% of the size of D. These S_{test} number of patterns are copied from D to a flexible dataset D^i. The training set size ($S_{train}^{(i)}$) increases incrementally, in steps of 1% of the data set size. For each increment, respective number of patterns are added to D^i from $D-D^i$. Thus the number of patterns in D^i at step *i* is equal to $S_{test}+S_{train}^{(i)}$.

Cross Validation Error is then performed by calculating the Mean Squared Error on D^i at each step by training on randomly chosen $S_{train}^{(i)}$ patterns, and testing on the rest of the S_{test} patterns. This is repeated 10 times for each step *i*. Mean of these MSE errors gives an indication of the performance of the learning technique at this iteration. It should be noted that this method is not a complete cross validation method since the number of times training and testing takes place is fixed to 10 and is independent of the training and testing size, thus does not cross validate the whole dataset.

3.3 Regression and Prediction

A linear regression method was used to regress this error for all the different types of equations, by taking logarithm of the respective variable. For example to regress through the logarithmic equation (y=*a*logx+*b*), the logarithm of all the x values can be taken, and linearly regressed to obtain the parameters a and b. The four equation which were studied were linear, logarithmic, exponential and the power law.

An objective of this project was also to see how well the modeling equations can predict the performance of the learning technique. For this effect, as we obtained the error at each iteration, we fitted the equations to the current available points. Thus we got the parameter values at each iteration, using which we predicted the final error on the complete dataset. Along with the predicted error, the values of the parameter and its variance, and the errors in the prediction were also observed. The results and their implications shall be studied in later sections.

The same method was used for all the various classification systems which were studied, for all datasets. Even though the training and the testing times differ from one learning system to the other, and also depend on the size of the dataset, the method was kept exactly the same. This was done just to ensure different parameters do not lead to contradicting results.

4. Experiment Design

The project was completely programmed in MATLAB. MATLAB provided a common platform on which all the various classification techniques were available, along with all the required functions for data preprocessing, calculation of error and regression. All experiments were taken on a Pentium 2.8GHz machine, and each data set took approximately 10-15 hours for completion.

4.1 Classification Techniques

All the classification techniques which we used were taken off-shelf, that is, were implementations which are currently being used for instructional, research and industrial purposes. Thus a source of error in the classification techniques would not be by the authors. Since most of these implementations have been studied in detail, they were assumed to be stable and correct.

The implementation of Decision Tree was ID3. A MATLAB implementation written by Frank Dellaert (Georgia Institute of Technology) was used. This implementation is the least tested compared to the rest of the methods. Future work shall use a more commonly used implementation of Decision Tree. None of the parameters of the Decision Trees were fixed since that could interfere with the results.

K-Nearest Neighbor was used as part of the NetLab [Nabney] toolbox for pattern recognition. This toolbox is one of the commonly used toolboxes for MATLAB. It offered the required flexibility. A fixed value of k=10 was used for the experiments, unless the training set size was less than 10, in which case k was set to the number of training patterns.

The most common implementation of Support Vector Machines being used and studied is LIBSVM [Chang]. A MATLAB version from Ohio State University (OSU SVM [Ma]) uses the pre-compiled code from LIBSVM. The RBF (Radial Basis Function) Kernel was used since it is more flexible when compared to linear or polynomial kernel functions, the values for all parameters set to the default (γ=1, C=1, ε=0.001).

MATLAB provides a very powerful toolbox for Neural Networks [Matlab]. Even though Netlab also has neural network implemented in it, the MATLAB toolbox is much more commonly used as experimental base. A feed-forward back propagation neural network with a single hidden layer with 10 hidden units was used. The number of hidden units was fixed to ensure uniformity across all the different data sets. 10 hidden units are usually large enough to successfully classify most datasets. Since the objective of this project is to observe the trend, the exact number of hidden units does not matter, as long it remains constant. The Levenberg-Marquadt training algorithm was used since it provides a balance of optimality and speed.

4.2 Data Sets

Four data sets were studied. An attempt was made to choose data sets which were very different from one another so that the results are not particular to a type of datasets. All the data sets used were from the UCI Machine Learning Repository [Hettich].

The Car Evaluation Dataset (CAR) consists of 1728 instances with 6 nominal attributes. This dataset was derived from a hierarchical decision model. As an example of a relatively different dataset from CAR, Clean1 from the MUSK dataset was chosen. Clean1 consists of one nominal and 166 continuous attributes, with 476 instances. The objective is to classify unseen molecules as musk or non-musk based on the various attributes of musk and non-musk molecules. The LED display dataset presents a noisy dataset. A dataset was generated with 24 boolean attributes, 1000 instances, and a 5% noise level. This represents a pseudo artificial dataset, with a high level of noise, making it similar to real-datasets. Finally, a fairly small dataset was chosen, namely the ZOO dataset. This dataset consisted 101 animals each with one nominal and 15 boolean attributes. The target was to classify each animal into one of seven types.

5. Results

The experiment results are presented in this section.

Table 1 shows the R^2 fitness values for all the classification method and Dataset combinations. Most of these combinations are best fit by logarithmic regression (81.25%), followed by power law (12.5%). The second highest value is highest for power law (50%), thus making it the second best equation to fit these classification methods and the datasets.

	Regression Method
Classification Method	CAR Evaluation
	MUSK - Clean1
	Zoo
	LED Display

(a)

	Linear	***Logarithmic***	***Exponential***	***Power Law***
Decision Tree (ID3)	0.7336	0.9509	0.8850	0.8943
	0.7600	0.9828	0.8722	0.6787
	0.6851	0.8258	0.3940	0.3020
	0.4660	0.6103	0.5644	0.6181
K-Nearest Neighbors	0.7326	0.9756	0.8649	0.9596
	0.7424	0.9283	0.8500	0.8911
	0.6619	0.5087	0.6956	0.4602
	0.6037	0.9381	0.6800	0.9558
Support Vector Machines	0.7510	0.9633	0.9451	0.8075
	0.5185	0.8815	0.7710	0.8208
	0.4553	0.7718	0.0061	0.0240
	0.7038	0.9474	0.8435	0.9169
Artificial Neural Networks	0.6215	0.9395	0.8598	0.8615
	0.5128	0.8501	0.1374	0.1433
	0.3141	0.4462	0.3191	0.3322
	0.6215	0.9221	0.7670	0.9210

(b)

Table 1: *Table (b) shows the R2 values for each of the classification methods, regression equations and datasets, formatted as shown in (a). The higher the value, the better the fit. XXX denotes the best fit, while XXX denotes the second best fit, for each dataset-classification method pair.*

An example of the fitness for the Car dataset is shown in figure 2.

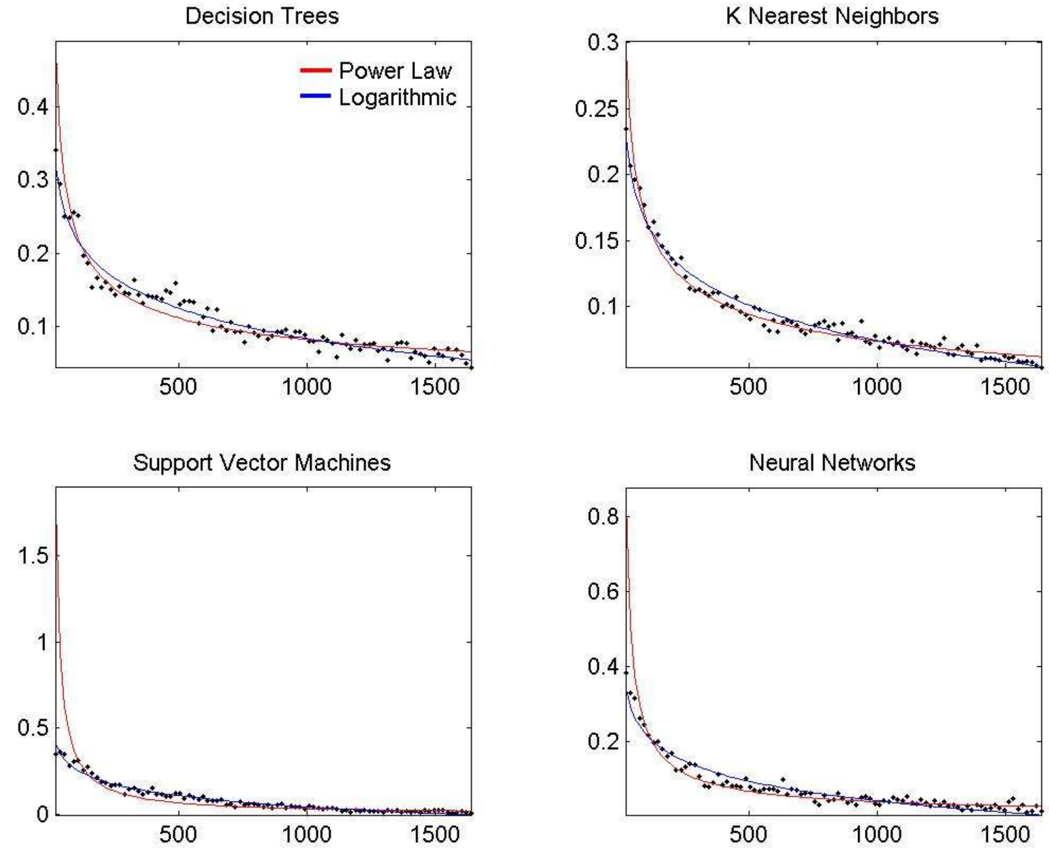

Figure 2: *Shows sample error curves. The X-axis consists of the training set size, and the Y-axis the Mean of Cross Validation error. The actual means are plotted as black dots. This graph has been plotted for the CAR dataset. The regression was obtained using the complete set of errors.*

A final fitness of logarithmic regression does not ensure that the error shall be better predicted by a logarithmic equation. Thus, partial data points were regressed to calculate the parameters of logarithm and power law, shown in Figure 3. It showed that the overall change in value of the parameter is much larger for power law than for the logarithm. Figure 4 shows the variance of this value. After initial jumps, both the variances follow similar trend. However, the value of the variance for logarithm is a few orders of magnitude smaller than that of power law.

These graphs on the CAR dataset show a preference towards logarithm even for prediction. But this is not conclusive. Using the values of the parameters calculated based on the partial datasets, the error was predicted at the final point (100% of training dataset). The error in prediction was calculated by the actual value for each increment in the training set size. The graph is plotted in Figure 5.

This method was repeated for all the datasets and the classification methods, and an attempt was made to identify the point at which the prediction of error stops changing. The actual value of this prediction is of less consequence when considering prediction. Table 2 shows are these values. The grayed cells correspond to the points which were within a ±50% range of the final prediction error. Even though this tolerance is very high, it gives an indication of the position at which the predicted error stop changing.

A thorough discussion of these results and their implications is given in the next section.

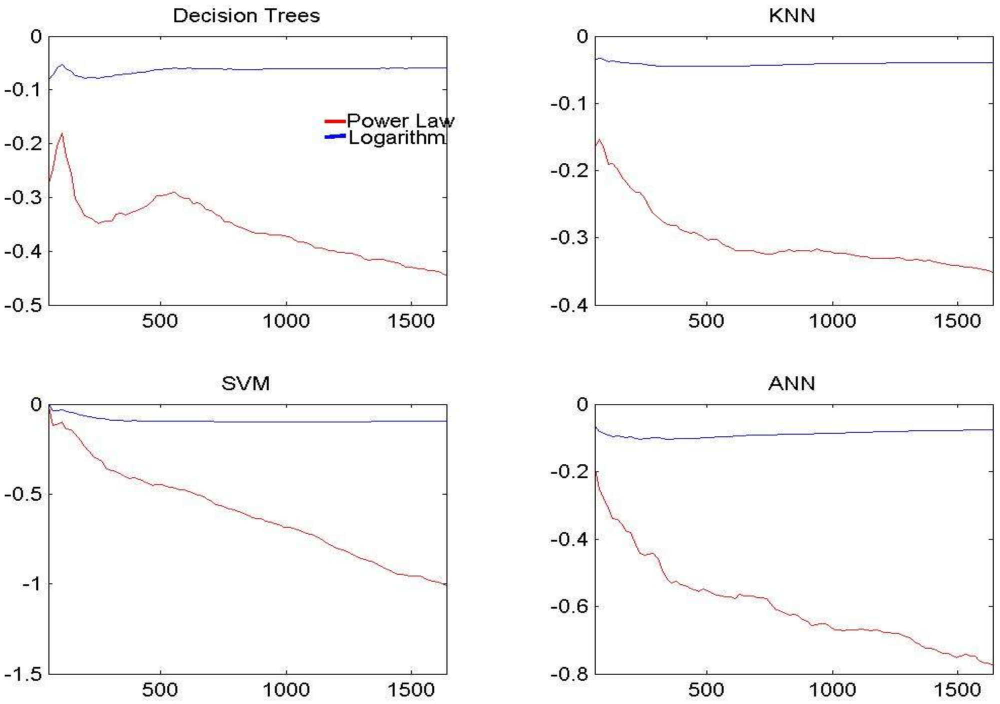

Figure 3: *Plots the values of one of the parameters of the regression, "a", for both logarithmic (y=alogx+b) and Power Law ($y=bx^a$). These values were calculated by regressing the partial data obtained. Thus the X-axis denotes the training set size using which data points till which were used for regression. This plot also uses the CAR dataset.*

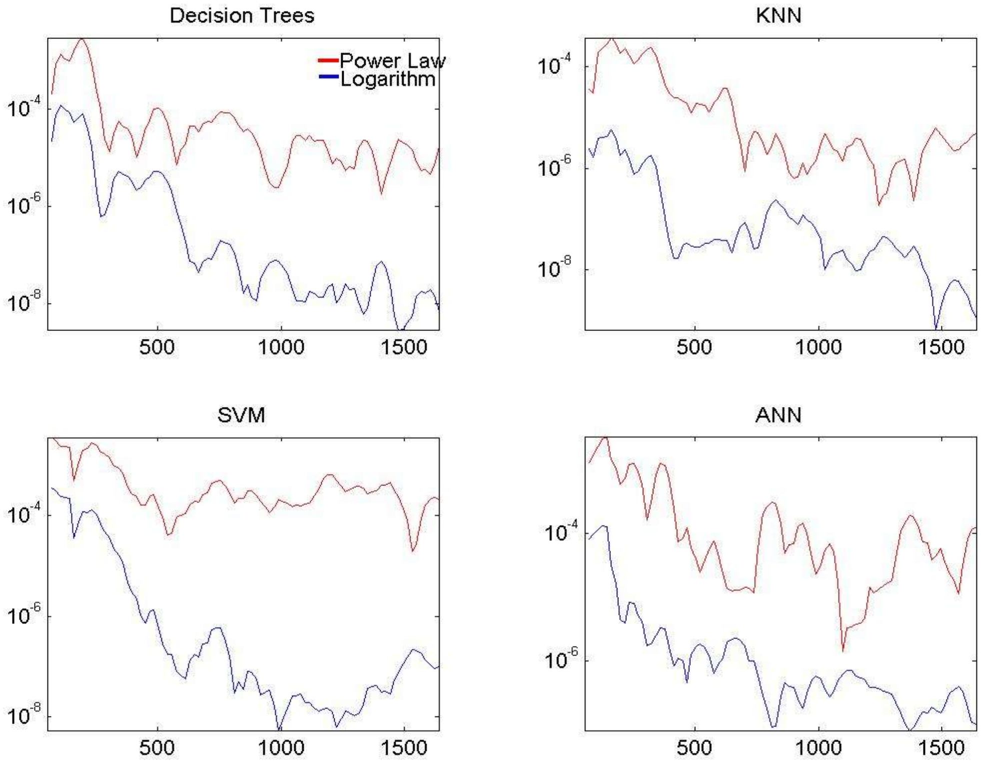

Figure 4: *The Variance of the parameter "a", as the training set size is varied for the CAR dataset. It should be noted that the Y-axis is set to a log scale, and thus the difference between the values of power law and logarithm is in orders of 10. It seems the value of "a" for logarithm stops changing fairly early ($<10^{-6}$), but the value of "a" for power never stabilizes.*

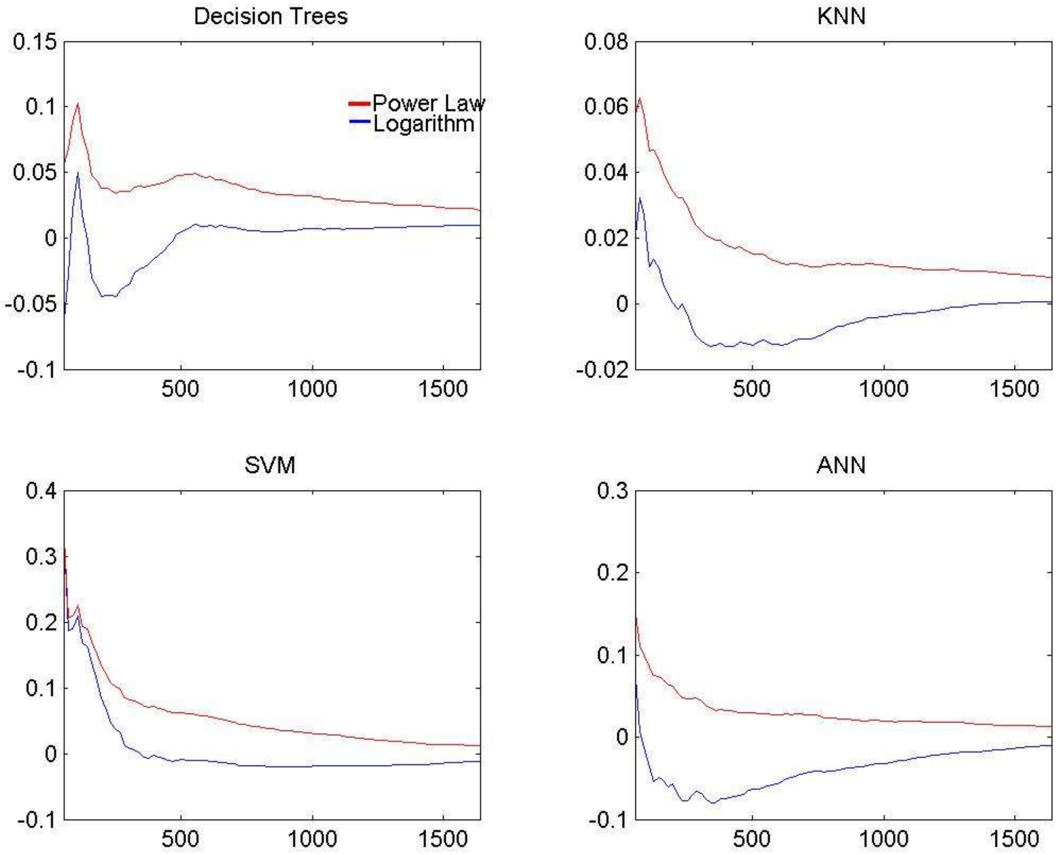

Figure 5: *Shows the error in prediction (Predicted-Actual) when partial curves for the CAR dataset are used to predict the error at the final position. The initial error is always very high in magnitude. The earlier the curves converge to their final value, the better. The error at the last point is of less consequence.*

	Dataset	
Classification Method	logarithm	Power Law
	10%	10%
	25%	25%
	50%	50%
	75%	75%
	100%	100%

(a)

	Car		***Zoo***		***LED Display***		***Musk***	
Decision Tree	-0.0310	0.0478	-0.2237	0.0734	-0.2860	-0.1052	-0.0217	0.0461
	-0.0165	0.0402	-0.0628	0.0739	-0.0579	-0.0140	-0.0128	0.0339
	0.0050	0.0351	0.0460	0.0957	0.0028	0.0178	-0.0103	0.0212
	0.0076	0.0269	0.0215	0.0366	0.0041	0.0124	-0.0109	0.0096
	0.0096	0.0211	0.0208	0.0202	0.0017	0.0062	-0.0081	0.0033

	Car		***Zoo***		***LED Display***		***Musk***	
K-Nearest Neighbors	0.0062	0.0399	0.7490	4.9871	-0.0710	-0.0136	0.1088	0.1201
	-0.0129	0.0181	0.2452	0.3135	-0.0622	-0.0245	0.0132	0.0416
	-0.0076	0.0119	0.0967	0.0999	-0.0463	-0.0256	-0.0238	0.0053
	-0.0018	0.0103	0.0575	0.0596	-0.0319	-0.0199	-0.0241	-0.0030
	0.0007	0.0079	0.0387	0.0382	-0.0214	-0.0145	-0.0180	-0.0049
Support Vector Machines	0.1378	0.1704	0.2444	0.2611	0.3061	0.3606	-0.1401	0.0660
	-0.0029	0.0714	-0.1539	-0.0087	0.0194	0.1390	-0.1642	0.0189
	-0.0189	0.0398	-0.1888	-0.0664	-0.1012	0.0181	-0.1109	0.0047
	-0.0184	0.0215	-0.1099	-0.0608	-0.0831	0.0002	-0.0771	-0.0043
	-0.0117	0.0119	-0.0719	-0.0511	-0.0519	-0.0014	-0.0476	-0.0041
Artificial Neural Networks	-0.0525	0.0696	-0.3214	-0.0275	-0.1235	0.2333	-0.1732	0.0173
	-0.0754	0.0320	-0.1778	-0.0454	-0.1476	0.0329	-0.1239	0.0041
	-0.0408	0.0228	-0.0618	-0.0184	-0.1186	-0.0008	-0.0784	-0.0140
	-0.0210	0.0183	0.0112	0.0240	-0.0784	-0.0047	-0.0488	-0.0106
	-0.0102	0.0126	-0.0060	-0.0081	-0.0483	-0.0051	-0.0332	-0.0156

(b)

Table 2: *Gives a complete list of error predictions for logarithmic and power law curves. (a) gives the format each cell of table in (b). For each dataset and classification method, prediction errors at 10%, 25%, 50%, 75% and 100% of the data is given. By examining these values, an approximate point of convergence can be calculated. The shaded regions show the values which were ± 50% of the error at the final point, indicating the region of relative stability.*

6. Discussion

Modeling decision tree performance has been studied for quite some time. [Frey] tested the performance on a large number of datasets from the UCI repository, and they were found to follow the power law better than they followed the other equations. Most of the datasets considered in this paper were fairly small. A later paper [Gu] applied similar approach to much larger datasets, and found that a biased power law provided the best fit to the learning curves.

In we examine the results in Table 1 for Decision Trees, it seems that the power law achieves good result only for one of the four datasets, and performed fairly poorly on the rest of the datasets. This deviation from the power law was observed in the other classification methods also, best of which was consistently logarithm. Not many experiments have been carried out on these other classification methods, though. [Forman] observes the effect of “little training” on SVMs, but does not model the curves using equations. A sample of the learning curve, and the logarithmic and power law fitness, obtained is shown in Fig 2.

The cause of the low values of the R^2 fitness value of power regression can be studied by examining the graphs. If we compare the logarithmic and power law regression, we see that power law in fact does asymptote to the value which the classification method asymptotes too. On the other hand, the logarithm function is still decreasing towards the end. The low value of R^2 value for power law regression is hence caused by the initial data points, which the power law does not fit at all. This results in the observation that the logarithm should be better at predicting the performance, even if the power law fits well to the final graph, since the logarithm parameters are “decided” by the initial points, while the power law requires a lot of data points towards the end to fit.

These modeling results lead to experiments which determine their use for prediction of performance. [Frey] and [Gu] mention a probable application of their work in prediction. [Brumen] attempted to assess the classification performance of C5.0 from initial small training size performance using a method which did not use a fixed equation to model the performance.

The prediction of performance in this project initially seems to favor the logarithm technique. Fig 3 shows the value of the parameter “a” when logarithm and power law are fit to the partial result obtained for one of the datasets. The results for the other datasets were similar. The value for logarithm fairly linearizes quite early, while it continues to be jagged for most of the curve for the power law. This difference is more clear when the variance of the value of “a” is observed (Fig 4). This variance for the power law is much higher than that of logarithm.

Even though this indicates that the logarithm might be predicting better, it does not say anything conclusively. Firstly, it could be the case that the logarithm is very inflexible towards the new data, and does not fit them well, and hence, does not change the value of its parameter. Secondly, the power law might be less sensitive to its parameter value, and thus large changes in “a” might not lead to large changes in the actual plot of the curve. Thus more methods were required for assessing prediction.

Figure 5 extrapolates the equations regressed from partial curves, and plots the error between the actual and the predicted value of the error when using whole of the training set size. This seems to show that the power law is actually stabilizing around the same time as the logarithm, thus being more in accordance with earlier work. Similar analysis was done for all the datasets, the results of which are shown in Table 2.

The table seems to suggest that the equation which predicts better is independent of the dataset. For example, the CAR dataset is better predicted when using power law for two of the four methods, while logarithm is better for one. Similar observations can be made about all of the datasets. On the other hand, if we follow the trend in classification techniques across the datasets, three seems to be a connection. For example, both neural network and support vector machines are better predicted when using the power law. On the other hand, decision trees seem to be better predicted by the logarithm. No such conclusion can be made about the K-Nearest Neighbor.

This project suggests that the logarithm function fits more datasets. But merely fitting the error curve better is not a good indicator of the performance of the classification method. The fitting of the equation to the complete dataset depends on the dataset, but prediction is depended more on the classification method used.

The next section of future work shall discuss some of the probable sources of error in this work, and plans on how to remove them so as to be more confident about the results presented.

7. Future Work

The results obtained for this project were very different from the ones found in earlier papers, even though the testing environment was pretty similar. More work is needed before this work can be generalized to all datasets and learning techniques.

Many more datasets need to be tested. Even though results from these four datasets are quite preliminary, they all demonstrate a deviant trend. This probably leads us to believe that if there is an error in our technique, it is not in the lack of datasets being considered. Thus we need to examine out method in detail.

Our intuitive seems to suggest that the problem lies in the regression method, in particular to the technique of finding linear regression after taking log. A different method to be considered for regression is going to be the first step in this direction, especially for exponential and power law regression.

The prediction of performance could be highly sensitive to the class distribution difference between the initial test set, and that of the complete dataset. Though since the patterns are being added to this set and thus the distribution approaches the final distribution ultimately, it could still cause problems in predicting the performance, which is only depended on the initial few samples.

If after these experiments the results obtained still similar to the current ones, the only issue left to handle is that of the learning techniques. More stable and commonly used implementations than the one used will be obtained. The initial parameters for learning system shall be decided more formally than using a empirical method as used in this project. It is only after these experiments can anything be ascertained about the logarithmic performance modeling of the classification techniques.

8. Conclusion

This project studied the performance of a variety of classification techniques on a number of various datasets, by varying the training size. The obtained learning curves were regressed by four simple equations, and the fitness was measured. It was found, for the restricted domain which was studied, that the logarithm function fits the curves better than power law for three out of the four datasets studied. This contradicts the belief of earlier work of the power law being a much better performance modeling for decision trees. By examining four completely different classification techniques, it was also observed that the equation thats models the performance depends more on the data than on the classification method. The performance of the modeling equations, when used for prediction, was found to be independent of the datasets. Decision Tree was found to be predicted well by the logarithmic function, but Support Vector Machines and Neural Network were better predicted by the power law. This selection of equation can be used to predict the performance on larger datasets without necessarily having all the patterns.

References

[Brumen] Brumen, B., Golob, I., Jaakkola, H., Welzer, T. and Rozman, I. *Early Assessment of Classification Performance.* Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation (2004)

[Chang] Chang, C. C., and Lin, C.J. *LIBSVM*: a library for support vector machines. (2001) [http://www.csie.ntu.edu.tw/~cjlin/libsvm/]

[Forman] Forman, G., and Cohen, I. *Learning from Little: Comparison of Classifiers Given Little Training.* 15th European Conference on Machine Learning and the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Italy (2004)

[Frey] Frey, L. J., and Fisher, D. H. *Modeling decision tree performance with the power law*. In Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics (1999)

[Gu] B.Gu, F.Hu, and H.Liu. Modelling Classification Performance for Large Data Sets -- An Empirical Study. In Proceedings of the Second International Conference of Web-Age Information Management (2001)

[Hettich] Hettich, S. and Blake, C. L. & Merz, C. J. *UCI Repository of machine learning databases.* Irvine, CA: University of California, Department of Information and Computer Science (1998) [http://www.ics.uci.edu/~mlearn/MLRepository.html]

[Ma] Ma, J., Zhao, Y., and Ahalt, S. OSUSVM: a Matlab SVM toolbox. [http://www.ece.osu.edu/~maj/osu_svm/]

[Matlab] Mathworks. *Matlab Neural Networks Toolbox* [http://www.mathworks.com/products/neuralnet/]

[Nabney] Nabney, I., and Bishop, C. *NetLab.* Neural Computing Research Group, Aston University, United Kingdom [http://www.ncrg.aston.ac.uk/netlab/]