An Empirical Study of Fitting Learning Curves

Baohua Gu[1], Feifang Hu[2], and Huan Liu[3]

[1] Department of Computer Science, National University of Singapore, Singapore
gubh@comp.nus.edu.sg
[2] Department of Statistics & Applied Probability, National University of Singapore
stahuff@nus.edu.sg
[3] Department of Computer Science & Engineering, Arizona State University, USA
hliu@asu.edu

Abstract. It is well known that many learning algorithms have diminishing returns for increased training data size. This paper empirically studies fitting learning curves of large data sets in search of a principled stopping criterion. Such a criterion is particularly useful when the data size is huge as in most data mining applications. Learning curves are obtained by running decision tree algorithm C4.5 and logistic discrimination algorithm LOG on eight large UCI data sets, then fitted with six competing models, which are compared and ranked in terms of their performance on fitting full-size learning curves and on predicting late portion with curves fitted from early portions of learning curves with small data sizes. The three-parameters power law is found in the experiments overall close to the best in fitting and the best in predicting. It is also found that although the fit ranking of these fitting models is almost consistent for all the eight data sets using the two algorithms, their prediction ranking varies more for LOG than for C4.5 over the eight data sets and the amount of data used in fitting. The findings here can be used in effective data mining with large data.

1 Introduction

1.1 Motivation

When we run an induction algorithm (e.g., learning a classifier) on a data set, it 'learns' from the data and produces a generalized model (e.g., a decision tree, or a set of classification rules). Generally speaking, the more instances the learner sees during the learning, the higher the model accuracy. Such a relation is depicted by a so-called *learning curve*. On a typical learning curve, the horizontal axis represents $Size$, the number of instances used for training (so called training data), which can vary between zero and N, the total number of available instances, while the vertical axis represents $Accuracy$, the accuracy rate of the model produced on the training data and tested against a set of instances unseen during the training (so called testing data). Therefore, a learning curve consists of a group points of $(Size, Accuracy)$ pairs, which we call thereafter in this paper *learning points*.

Theoretical and empirical studies have suggested that learning curves typically have a fast increasing portion early in the curve, following with a relatively slow increasing portion and finally a plateau portion when the learning accuracy no longer increases with more training data [12]. Basically, the learning curve indicates diminishing returns in terms of accuracy as training set size increases. The exact behavior of a learning curve is founded differing with the used learning algorithm as well as the used data set [1, 6]. However, a learning curve, if available, can be very useful: (1) to predict what is a probable learning accuracy for a given amount of data, or (2) to estimate how many data should be used to achieve a desired learning accuracy [5]. Such uses can find many practical applications by data mining practitioners who frequently deal with large data sets.

In practice, due to the diminishing returns in terms of accuracy as training data size increases, researchers attempt to model the learning curve by fitting a few learning points in order to avoid the redundant and expensive work of learning every size of a data set [9]. For large data sets, the work becomes more significant if these learning points only covering sizes that are far less than the full size (N) of all available data. Mathematically, such a fitting can be implemented by interpolating the front part of a learning curve then extrapolating to the full size. The result of the fitting depends on the number of points to fit and the used model. The choice of a model is actually more dominant in deciding the result of extrapolating than the number of fitting points. Besides, we are restricted to use too many learning points or covering too large size of data due to the costly computation to obtain these points from large data sets. Therefore, the problem can be stated as follow: given a fixed number of fitting points, which model fits well learning curves.

Theoretically, one can fit a learning curve with as many models as he will, as long as he tries more functions to fit or uses more parameters to tune and can afford the subsequent complexity. Practically, however, models with simpler expression and less parameters are more preferable because of resource constraints. This study empirically compares six models found in the mathematical community that are potential candidates for fitting learning curves, investigates which model is the most economical and robust in learning curve fitting with various data sets and different learning algorithms, and provides some guidelines in applying the models in learning curve fitting.

These models, namely two-parameters power law, three-parameters power law, logarithm model, vapor pressure model, Morgan-Mercer-Flodin (MMF) model, and weibull model [14], come from three function families with two to four parameters. All these models take shapes similar to that of a typical learning curve. They are compared and ranked with respect to their performance on two aspects: (1) fitting full-size learning curves, and (2) predicting learning accuracy at full size by means of curves fitted from small early portions of full-size learning curves. Learning curves are obtained from running two classifiers, namely C4.5 (decision tree approach) and LOG (logistic discrimination approach) on eight large UCI data sets.

1.2 Related Research

The common feature in shapes of learning curves suggests a possibility of fitting learning curves with a common model, of which the parameters are tuned to fit the exact shapes. The famous power law seems more often suggested to take this job [4, 5, 8].

Frey and Fisher [5] use $y = a * x^{-b}$ to model C4.5 decision tree performance, where y is error rate of learning, x is training data size, $a > 0$ and $b > 0$ are parameters to be fitted. After comparing the 2-parameter power law with other three 2-parameters models, namely linear model, logarithm model and exponential model, they conclude that it is the best of the 4 models and usable to predict error rate of large size. Note that modeling learning error is equivalent to modeling learning accuracy, their work does show that the power law is potentially suitable in fitting learning curves. However, all data sets they use are relatively small, most of which have hundreds of instances and the largest has less than 4000 instances. Moreover, the compared linear model and exponential model are inherently not suitable in fitting learning curves due to their too fast increasing with x, and this will become even worse for large data sets. John and Langley[8] use $y = a - b * x^{-c}$ to fit learning curves of a naive Bayesian classifier, where y is learning accuracy, x is data size, $a > 0, b > 0, c > 0$ are parameters to estimate. They apply it to large data sets, however, they do not consider other fitting models. Moreover, in each of the above works, only one learning algorithm is considered.

Therefore, it is reasonable to ask whether there is another model better than the power law and whether a specified model such as the power law can apply to large data sets as well as other learning algorithms. In this work, we empirically compare six models in two aspects of performance: fitting and predicting, by taking into account the factors of large data sets and different learning algorithms. We hope to make it as an effort toward answering the above questions.

2 Candidate Models, Classifiers and Data Sets

The six candidate models used in this study are briefly described in Table 1. All of them are non-linear and respectively belong to three different major function families: *Power Law* family, *Exponential* family and *Sigmoidal* family. Figure 1 roughly illustrates their shapes, compared to the shape of a typical learning curve. From Figure 1, we can see that all the six models have the potential to fit a learning curve properly by tuning their parameters.

To facilitate the fitting process, we use a curve-fitting software *Curve Expert* [7], in which the *Levenberg-Marquardt* method [2, 15] is used to solve nonlinear regression. This method combines the steepest-descent method and a Taylor series based method to obtain a fast, reliable technique for nonlinear optimization. We use the following setting in fitting with all models: auto-guessed initial values for regression, $1.0e-06$ for the tolerance value of stopping iterations, the maximum iterations 100, and the maximum diverging iterations 100.

Table 1. The 6 Candidate Fitting Models

Model Name	Notion	Formula	Remarks
2-parameters Power Law	Pow1	$y = a * x^b$	$a > 0, b > 0,$ Power Law family
Logarithm	Log	$y = a + b * log(x)$	Exponential family
3-parameters Power Law	Pow2	$y = a - b * x^{-c}$	$a > 0, b > 0, c > 0,$ Power Law family
Vapor Pressure	Vap	$y = exp(a + b/x + c * log(x))$	Exponential family
MMF model	MMF	$y = (a * b + c * x^d)/(b + x^d)$	Sigmoidal family
Weibull	Wbl	$y = a - b * exp(-c * x^d)$	$a > 0, b > 0, c > 0,$ Sigmoidal family

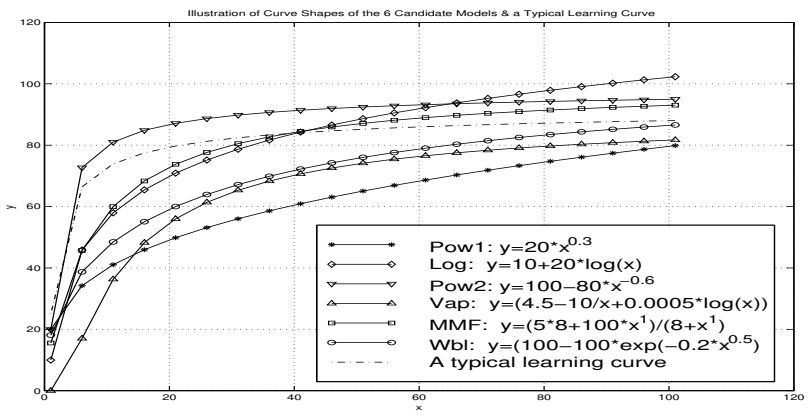

Fig. 1. Illustration of Curve Shapes of the 6 Candidate Models and a Typical Learning Curve

We choose C4.5 [13] and LOG [11] as representatives of two categories of classification algorithms: tree method and statistical method. The C4.5 is a popular decision tree induction system, and the **LOG** is a FORTRAN 90 implementation of logistic discriminant analysis. A comprehensive comparison of many classifiers [10] has reported that both algorithms tend to produce high classification accuracy. The eight large UCI data sets [3] used in this study are summarized in Table 2.

3 Fitting Performance

3.1 Criterion of Comparison

We use standard error of the estimates S to measure the performance of nonlinear curve fitting and report correlation coefficient r for reference. The standard error of the estimate is defined as $S = \sqrt{(\sum_{i=1}^{n}(y_i - f(x_i))^2)/(n-m)}$, where n is number of points, m is number of parameters in the model, $n - m$ is actually the number of degrees of freedom, (x_i, y_i) is the ith point to be fitted, and $f(x)$

Table 2. Descriptions of 8 UCI data sets

Data Set Name	Notion	Nominal Attribute	Numeric Attribute	Instances # for Training	Instances # for Testing	C4.5 max accuracy	LOG max accuracy
abalone	aba	1	7	3000	1177	45.0	25.6
adult	adu	8	6	36000	12842	85.8	83.1
letter	let	0	16	15000	5000	87.7	78.1
mushroom	mus	22	0	8000	2992	99.6	99.9
pendigits	pen	0	16	8000	2992	96.4	95.3
satellite	sat	0	36	4435	2000	91.5	83.8
thyroid	thy	15	6	3772	3428	99.6	95.9
waveform	wav	0	21	80000	40000	99.0	87.5

is the fitted model. The correlation coefficient is defined as $r = \sqrt{(S_t - S_r)/S_t}$, where S_t is the deviation of the points from their mean value $\bar{y} = (\sum_{i=1}^{n} y_i)/n$, $S_t = \sum_{i=1}^{n}(y_i - \bar{y})^2$, S_r is the deviation of the points from the fitted curve, $S_r = \sum_{i=1}^{n}(y_i - f(x_i))^2$. For a perfect fit, there will be $S_r = 0$, thus $r = 1$.

3.2 Results Of Fitting

To get a full-size learning curve of each data set, an uniform sampling size scheme with totally 12 size points is used, i.e., 1%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100% of the total size. At each size, a random sample is drawn from the total training data and used to train each classifier, which is then tested against the testing data. This procedure is repeated 10 times to get an average learning accuracy for the size. Finally, twelve learning points are obtained for each data set and each classifier. Then the six candidate models are used to fit. The results of fitting C4.5 and LOG learning curves are respectively listed in Table 3and Table 4. The overall ranking of the six models in terms of S value on the eight data sets are shown in the last seven rows. From the two tables, we observe that:

1. In most cases, the models with more parameters tend to have smaller S values, thus are ranked in the front. This is consistent with the general mathematical rule.
2. In most cases, the models with the same number of parameters tend to have close S values, suggesting their similar performance in fitting.
3. In most cases, the models with three and four parameters are significantly better than the two models with two parameters, while the S values between three-parameters models and four-parameters models are not significantly different.

It is understood that using one more parameter does increase the complexity of a model and the fitting process. But, if it did not significantly increase fitting performance, we would rather not use it. The above result suggests that models

Table 3. Fitting C4.5 Learning Curves

Data	Model	Estimated Parameters	S	S Rank	r
aba	Pow1	a=30.66 b=0.05219	1.935	6	0.859
	Log	a=28.51 b=2.251	1.835	5	0.874
	Pow2	a=45.04 b=323.3 c=0.9649	0.610	1	0.988
	Vap	a=3.790 b=-9.072 c=2.470e-3	0.619	2	0.988
	MMF	a=-83.69 b=3.408 c=45.00 d=1.027	0.660	3	0.988
	Wbl	a=44.89 b=1.014e+4 c=4.324 d=0.1306	0.689	4	0.987
adu	Pow1	a=74.36 b=0.01423	0.552	6	0.969
	Log	a=73.82 b=1.191	0.530	5	0.971
	Pow2	a=87.08 b=50.17 c=0.3532	0.095	3	0.999
	Vap	a=4.368 b=-7.527 c=8.216e-3	0.208	4	0.996
	MMF	a=72.31 b=26.58 c=86.30 d=0.6233	0.064	1	0.999
	Wbl	a=86.10 b=44.97 c=0.7159 d=0.1843	0.078	2	0.999
let	Pow1	a=31.42 b=0.1096	2.653	6	0.982
	Log	a=14.17 b=7.811	1.570	5	0.994
	Pow2	a=120.0 b=148.2 c=0.1586	0.165	2	0.999
	Vap	a=3.755 b=-21.34 c=0.07562	0.719	4	0.999
	MMF	a=9.482 b=9.778 c=98.14 d=0.4434	0.269	3	0.999
	Wbl	a=96.45 b=180.5 c=0.6263 d=0.1629	0.158	1	0.999
mus	Pow1	a=31.42 b=0.1096	2.653	6	0.982
	Log	a=14.17 b=7.811	1.571	5	0.994
	Pow2	a=100.4 b=144.7 c=0.6573	0.171	2	0.999
	Vap	a=3.755 b=-21.34 c=0.07562	0.719	4	0.999
	MMF	a=9.482 b=9.778 c=98.14 d=0.4434	0.269	3	0.999
	Wbl	a=96.45 b=180.5 c=0.6263 d=0.1629	0.158	1	0.999
pen	Pow1	a=57.40 b=0.06080	3.165	6	0.949
	Log	a=50.93 b=5.339	2.759	5	0.962
	Pow2	a=100.0 b=167.3 c=0.4277	0.089	2	0.999
	Vap	a=4.339 b=-10.52 c=0.02624	0.570	4	0.998
	MMF	a=-73.01 b=1.581 c=99.15 d=0.5090	0.055	1	0.999
	Wbl	a=97.58 b=614.8 c=1.780 d=0.1380	0.158	3	0.999
sat	Pow1	a=72.20 b=0.02904	0.684	6	0.983
	Log	a=70.81 b=2.533	0.619	5	0.986
	Pow2	a=99.41 b=43.15 c=0.2032	0.330	3	0.997
	Vap	a=4.335 b=-1.893 c=0.02196	0.425	4	0.994
	MMF	a=76.07 b=96.75 c=92.39 d=0.8605	0.242	1	0.998
	Wbl	a=92.26 b=27.56 c=0.2236 d=0.3257	0.277	2	0.998
thy	Pow1	a=88.93 b=0.01462	0.850	6	0.926
	Log	a=88.50 b=1.433	0.830	5	0.930
	Pow2	a=100.2 b=66.53 c=0.5832	0.042	3	0.999
	Vap	a=4.562 b=-2.134 c=4.922e-3	0.117	4	0.999
	MMF	a=77.14 b=8.602 c=99.92 d=0.7746	0.025	2	0.999
	Wbl	a=99.79 b=329.0 c=2.217 d=0.1458	0.023	1	0.999
wav	Pow1	a=72.74 b=0.02793	0.445	5	0.993
	Log	a=69.93 b=2.631	0.410	4	0.994
	Pow2	a=128.9 b=69.19 c=0.07553	0.404	3	0.995
	Vap	a=4.304 b=-7.651 c=0.02635	0.448	6	0.994
	MMF	a=86.12 b=7115.0 c=100.3 d=0.9969	0.075	1	0.999
	Wbl	a=101.4 b=23.90 c=0.05664 d=0.3332	0.295	2	0.998

Models	Sum of S ranks over 8 data sets	Average	Overall Rank
Pow1	$6+6+6+6+6+6+6+5=47$	5.9	6
Log	$5+5+5+5+5+5+5+4=39$	4.9	5
Pow2	$1+3+2+2+2+3+3+3=19$	2.4	3
Vap	$2+4+4+4+4+4+4+6=32$	4.0	4
MMF	$3+1+3+3+1+1+2+1=15$	1.9	1
Wbl	$4+2+1+1+3+2+1+2=16$	2.0	2

Table 4. Fitting LOG Learning Curves

Data	Model	Estimated Parameters	S	S Rank	r
aba	Pow1	a=13.99 b=0.07739	0.730	6	0.961
	Log	a=11.79 b=1.765	0.600	5	0.974
	Pow2	a=28.09 b=33.06 c=0.3141	0.207	3	0.997
	Vap	a=2.886 b=-6.658 c=0.04437	0.274	4	0.995
	MMF	a=12.50 b=33.54 c=25.95 d=0.8087	0.179	1	0.998
	Wbl	a=25.66 b=25.30 c=0.3734 d=0.3027	0.189	2	0.998
adu	Pow1	a=69.81 b=0.01749	1.153	6	0.874
	Log	a=68.86 b=1.431	1.130	5	0.879
	Pow2	a=83.35 b=666.2 c=0.7495	0.470	1	0.982
	Vap	a=4.391 b=-31.49 c=2.829e-3	0.484	3	0.981
	MMF	a=59.54 b=126.3 c=83.23 d=0.9380	0.489	4	0.983
	Wbl	a=83.00 b=37.87 c=0.1928 d=0.3582	0.477	2	0.984
let	Pow1	a=35.59 b=0.08677	5.283	6	0.862
	Log	a=20.68 b=6.371	4.747	5	0.890
	Pow2	a=79.54 b=1612.0 c=0.7584	0.751	4	0.998
	Vap	a=4.317 b=-86.16 c=5.235e-3	0.747	3	0.998
	MMF	a=2.586 b=166.6 c=78.81 d=1.051	0.588	2	0.999
	Wbl	a=77.84 b=79.66 c=0.05256 d=0.5541	0.333	1	0.999
mus	Pow1	a=92.77 b=9.098e-3	0.505	6	0.933
	Log	a=92.57 b=0.9003	0.497	5	0.935
	Pow2	a=100.4 b=49.49 c=0.5556	0.047	3	0.999
	Vap	a=4.577 b=-2.010 c=3.385e-3	0.085	4	0.998
	MMF	a=86.53 b=12.32 c=100.2 d=0.7559	0.043	1	0.999
	Wbl	a=100.1 b=197.7 c=2.077 d=0.1419	0.043	1	0.999
pen	Pow1	a=64.47 b=0.04599	1.961	6	0.950
	Log	a=60.05 b=4.154	1.776	5	0.960
	Pow2	a=99.87 b=142.9 c=0.3994	0.530	3	0.997
	Vap	a=4.359 b=-11.48 c=0.02321	0.709	4	0.994
	MMF	a=61.24 b=54.20 c=97.01 d=0.8082	0.484	2	0.998
	Wbl	a=96.00 b=57.92 c=0.2273 d=0.3433	0.446	1	0.998
sat	Pow1	a=52.17 b=0.05922	1.629	6	0.971
	Log	a=47.51 b=4.530	1.523	3	0.974
	Pow2	a=130.3 b=91.05 c=0.08354	1.543	4	0.976
	Vap	a=3.997 b=-1.523 c=0.05385	1.670	5	0.972
	MMF	a=64.44 b=4.635e+4 c=84.15 d=1.789	0.433	1	0.998
	Wbl	a=83.78 b=20.93 c=2.109e-3 d=0.9821	0.644	2	0.996
thy	Pow1	a=81.50 b=0.02158	1.472	6	0.895
	Log	a=80.58 b=2.036	1.430	5	0.901
	Pow2	a=97.18 b=96.36 c=0.5853	0.670	3	0.981
	Vap	a=4.518 b=-3.410 c=6.334e-3	0.768	4	0.975
	MMF	a=85.56 b=1.369e6 c=96.09 d=2.735	0.136	1	0.999
	Wbl	a=96.09 b=12.81 c=5.225e-3 d=1.011	0.208	2	0.998
wav	Pow1	a=83.93 b=3.879e-3	0.267	6	0.879
	Log	a=83.87 b=0.3383	0.265	5	0.881
	Pow2	a=87.54 b=488.4 c=0.8285	0.026	3	0.999
	Vap	a=4.467 b=-16.04 c=4.270e-4	0.032	4	0.998
	MMF	a=83.13 b=1212.0 c=87.59 d=1.105	0.021	1	0.999
	Wbl	a=87.50 b=464.5 c=2.204 d=0.1370	0.022	2	0.999

Models	Sum of S ranks over 8 data sets	Average	Overall Rank
Pow1	$6+6+6+6+6+6+6+6=48$	6.0	6
Log	$5+5+5+5+5+5+5+5=40$	5.0	5
Pow2	$3+1+4+3+3+4+3+3=24$	3.0	3
Vap	$4+3+3+4+4+5+4+4=31$	3.9	4
MMF	$1+4+2+1+2+1+1+1=13$	1.6	2
Wbl	$2+1+1+1+1+2+2+2=12$	1.5	1

with two parameters seems not enough to fit well a learning curve, but using four parameters seems redundant. Therefore, a model with three parameters seems a trade-off between model complexity and model accuracy.

These observations reinforce the previous claim that the power law is suitable for fitting learning curves, considering its simplicity and popularity[5].

4 Predicting Performance

A model may become less attractive if it is strong in fitting a full-size learning curve but weak in extrapolating a small portion of learning curve. Given two models with similar fitting performance, the one with more accurate extrapolation is more desirable to data mining. We call this extrapolating property the *predicting performance* of a model. In this section, we compare this performance of the six candidate models. For each data set and each algorithm, we first obtain learning points within a small early portion of a full-size learning curve to fit with the six models, then predicting learning accuracy at the full size, compared to the observed full-size learning accuracy given in Table 2. The model that gives the less difference between the predicted and the observed is considered better.

However, doing so implies an essential but still open issue: how many instances are adequate to support a good extrapolation. Frey and Fisher [5] mention this issue, however, they only test a 15% portion of whole data. Intuitively, when used to extrapolate, the larger the portion, the more accurate the prediction. To find out how the predicting performance will be affected by size of the portions, we intently set four portions that cover 5%, 10%, 15%, and 20% of the total size for each data set. Within each portion, 10 learning points equally spanning the interval of the portion are obtained to fit. Like in the previous section, learning accuracy at each size is averaged from 10 runs on random samples with that size.

The differences (in percentage) between the prediction and the observation on the 100% size (i.e., $diff = y_{predicted} - y_{observed}$) for the two classification algorithms are reported respectively in Table 5 and Table 6, for all data sets and the four portions. In both tables, for each portion of a data set, the names of models that rank the first three in terms of $|diff|$ is reported in the rightmost three columns of that portion, while the bottom six rows report the ranking of the six models averaged on the eight data sets for each portion and the overall ranking for the four portions.

The results show that the three-parameters power law tends to have the best predicting performance. This is probably because given only a segment of a whole learning curve, models with two parameters tend to underfit it, while models with four parameters tend to overfit it thus negatively affect their extrapolation. However, some observations also distinguish themselves.

(1) The predicting performance of a model may greatly differ with algorithms. This can be seen from their ranking. For example, the Pow2 overall ranks best

Table 5. Predicting Performance Comparison of C4.5

Data	Portion	Pow1	Log	Pow2	Vap	MMF	Wbl	Rank 1,2,3
aba	5%	14.12	9.34	4.27	7.97	2.78	3.35	MMF, Wbl, Pow2
	10%	12.64	9.67	-0.04	-0.03	-1.85	-2.32	Vap, Pow2, MMF
	15%	3.09	2.67	-0.58	-0.17	-1.43	-1.67	Vap, Pow2, MMF
	20%	1.84	1.62	1.05	1.70	-0.59	0.62	MMF, Wbl, Pow2
adu	5%	7.28	6.56	0.39	2.38	-0.69	-1.65	Pow2, MMF, Wbl
	10%	2.04	1.91	1.03	1.72	0.19	0.99	MMF, Wbl, Pow2
	15%	1.59	1.51	1.00	1.83	0.52	1.70	MMF, Pow2, log
	20%	1.64	1.58	0.50	0.93	-0.17	0.26	MMF, Wbl, Pow2
let	5%	24.89	9.31	1.22	14.56	-8.85	-15.81	Pow2, MMF, Log
	10%	13.27	6.05	-0.11	5.58	-2.85	-5.27	Pow2, MMF, Wbl
	15%	9.73	5.02	0.62	5.48	-3.16	-7.44	Pow2, MMF, Vap
	20%	8.32	4.75	1.08	3.85	-2.60	-3.31	Pow2, MMF, Wbl
mus	5%	23.82	18.75	-2.46	-1.06	-2.48	-3.10	Vap, Pow2, MMF
	10%	10.81	9.40	0.07	2.12	-0.07	-0.81	Pow2, MMF, Wbl
	15%	6.76	6.11	-0.42	0.57	-0.48	-0.64	Pow2, MMF, Vap
	20%	4.69	4.34	-0.64	-0.14	-0.65	-0.74	Vap, Pow2, MMF
pen	5%	31.85	21.31	-0.69	6.13	-0.81	-1.53	Pow2, MMF, Wbl
	10%	15.20	11.75	0.80	4.67	0.65	0.36	Wbl, MMF, Pow2
	15%	9.24	7.60	0.57	2.93	-1.47	-2.35	Pow2, MMF, Wbl
	20%	6.04	5.15	-0.06	1.46	-0.97	-1.28	Pow2, Wbl, MMF
sat	5%	23.15	17.76	-2.34	-0.72	-2.49	-3.49	Vap, Pow2, MMF
	10%	7.78	6.68	-3.02	-3.11	-3.52	-3.62	Pow2, Vap, MMF
	15%	3.60	3.14	-1.40	-0.28	-1.62	-1.77	Vap, Pow2, MMF
	20%	1.95	1.71	-0.93	-0.11	-1.21	-1.07	Vap, Pow2, Wbl
thy	5%	12.02	10.52	-1.68	-0.62	-1.77	-2.07	Vap, Pow2, MMF
	10%	5.80	5.35	0.68	2.20	0.22	0.41	MMF, Wbl, Pow2
	15%	3.17	3.00	0.63	1.47	0.03	0.13	MMF, Wbl, Pow2
	20%	2.69	2.57	0.42	1.00	-0.73	-0.14	Wbl, Pow2, MMF
wav	5%	-2.02	-2.39	-3.40	2.14	5.40	7.83	Pow1, Vap, Log
	10%	0.77	0.32	-0.68	2.24	3.20	2.43	Log, Pow2, Pow1
	15%	1.69	1.28	-0.01	2.07	1.97	2.15	Pow2, Log, Pow1
	20%	1.66	1.32	0.38	2.79	2.20	2.87	Pow2, Log, Pow1

Rank	Portion	Pow1	Log	Pow2	Vap	MMF	Wbl	Remarks
Aver -age	5%	6	5	1	3	3	4	average over 8 data sets for each portion
	10%	6	5	1	4	2	3	
	15%	6	5	1	3	2	5	
	20%	6	5	1	4	2	3	
Overall Rank		6	5	1	3	2	4	over all portions

Table 6. Predicting Performance Comparison of LOG

Data	Portion	Pow1	Log	Pow2	Vap	MMF	Wbl	Rank 1,2,3
aba	5%	-9.40	-14.98	-16.89	-3.80	-21.22	-23.67	Vap, Pow1, Log
	10%	-14.84	-16.85	-11.65	-12.70	-20.10	-21.17	Pow2, Vap, Pow1
	15%	-15.03	-16.59	-14.14	-14.30	-19.94	-21.26	Pow2, Vap, Pow1
	20%	-16.14	-17.05	-14.51	-17.45	-19.20	-19.59	Pow2, Pow1, Log
adu	5%	1.91	1.40	0.35	7.00	8.64	8.31	Pow2, Log, Pow1
	10%	1.29	1.04	-3.34	-2.87	-3.38	-3.51	Log, Pow1, Vap
	15%	-1.54	-1.61	-2.07	-1.32	-2.16	-1.38	Vap, Wbl, Pow1
	20%	0.60	0.45	-0.39	0.19	-1.66	-0.77	Vap, Pow2, Log
let	5%	68.47	20.24	8.17	31.88	10.82	5.50	Wbl, Pow2, MMF
	10%	44.62	20.68	13.97	38.73	-5.21	-11.12	MMF, Wbl, Pow2
	15%	23.63	13.42	4.52	9.53	-3.08	-9.74	MMF, Pow2, Vap
	20%	11.80	7.34	-5.49	-5.40	-7.01	-10.72	Vap, Pow2, MMF
mus	5%	5.04	4.73	0.04	1.39	-0.85	-1.65	Pow2, MMF, Vap
	10%	3.56	3.42	1.02	1.92	0.75	0.78	MMF, Wbl, Pow2
	15%	2.59	2.51	0.72	1.26	0.64	0.59	Wbl, MMF, Pow2
	20%	1.57	1.54	0.90	1.14	0.77	1.12	MMF, Pow2, Wbl
pen	5%	20.01	13.81	1.36	8.07	-1.34	-1.10	Wbl, MMF, Pow2
	10%	5.92	4.42	-5.35	-3.88	-6.48	-7.37	Vap, Log, Pow2
	15%	3.67	2.70	0.65	2.07	-0.11	0.72	MMF, Pow2, Wbl
	20%	4.60	3.71	1.96	3.85	2.23	3.10	Pow2, MMF, Wbl
sat	5%	10.36	3.92	-18.56	-22.93	-19.36	-19.54	Log, Pow1, Pow2
	10%	-0.25	-3.10	-5.29	16.63	48.74	16.57	Pow1, Log, Pow2
	15%	5.90	2.77	0.50	12.50	-0.95	2.39	Pow2, MMF, Wbl
	20%	-1.26	-2.85	-4.21	7.14	-3.11	3.01	Pow1, Log, Wbl
thy	5%	5.43	3.55	0.88	15.30	-4.65	8.43	Pow2, Log, MMF
	10%	7.50	5.89	3.33	11.87	-4.03	7.80	Pow2, MMF, Log
	15%	3.97	3.30	0.86	2.37	-3.63	-0.94	Pow2, Wbl, Vap
	20%	4.15	3.58	-0.82	0.38	-1.74	-3.05	Vap, Pow2, MMF
wav	5%	-8.19	-8.29	-11.72	-12.15	-11.80	-11.88	Pow1, Log, Pow2
	10%	-10.26	-10.28	-11.69	-11.95	-11.76	-11.44	Pow1, Log, Wbl
	15%	-10.97	-10.97	-11.48	-11.35	-11.60	-11.06	Pow1, Log, Wbl
	20%	-11.18	-11.18	-11.51	-11.43	-11.54	-11.20	Pow1, Log, Wbl

Rank	Portion	Pow1	Log	Pow2	Vap	MMF	Wbl	Remarks
Aver -age	5%	4	2	1	6	3	5	average over 8 data sets for each portion
	10%	3	2	1	5	4	6	
	15%	6	5	1	3	4	2	
	20%	6	3	1	2	4	5	
Overall Rank		6	2	1	4	3	5	over all portions

for both C4.5 and LOG, however, in its C4.5 predicting, for all data sets and all portions, it ranks in the first three, while in its LOG predicting, for some portions of some data sets, it ranks out of the first three. This can also be seen from the $|diff|$. For example, in the C4.5 predictions of Pow2, the $diff$ values for all data sets and all portions are less than 3%, except 4.27% for *abalone* and 3.4% for *waveform* on the 5% portion, while in its LOG predictions, the $diff$ becomes 16% for *abalone* and 11.72% for *waveform* on the 5% portion.

(2) The predicting performance of a model may differ greatly with data sets. For example, the logarithm model can predict LOG learning accuracy very well in *adult* ($|diff| < 1.5\%$ for all the portions) but very bad in *letter* ($|diff| > 20\%$ for the 5% and 10% portions).

(3) The expectation that a larger portion gives better predicting is satisfied better in C4.5 predicting than in LOG predicting. In C4.5 predicting, almost all models produce significantly less $|diff|$ in the 20% portion than in the 5% portion. In LOG predicting, even the rank-1 Pow2 shows remarkable drops of $|diff|$ values only in one data sets the *satellite*, while for *abalone*, *letter*, and *waveform*, the reduces become less remarkable, for the rest four data sets, even no reduce but increase.

(4) There are two data sets, *abalone* and *waveform*, all models predict bad the LOG learning accuracy for all the four portions (all $|diff|$ values larger than 10%).

These observations imply that even for the best model (the three-parameters power law), its predicting performance depends on learning algorithms as well as data sets.

5 Conclusion and Future Research

In this empirical study, we compare six candidate models with two to four parameters and coming from three functional families, in terms of their performance of fitting full-size learning curves and that of predicting full-size learning accuracy by extrapolating partial learning curves fitted from small portions of whole data. We use eight large UCI data sets and two different classification algorithms. The three-parameters power law model is found over close to the best in fitting performance and the best in predicting performance. If taking into account the trade-off between model complexity and model accuracy, it should be the best in terms of fitting and predicting of the six models.

With the aid of this model, we are able to predict for a large data set: (1) what is the achievable learning accuracy of a given size of data, and (2) how many data is needed to reach a desirable learning accuracy, by fitting a small portion of learning curve covering limited size of total data. This will save much computation cost, and therefore, is useful in effective data mining with large data sets.

Our experiments also reveal that even the best model also has varying predicting performance; sometimes the performance could vary significantly with learning algorithms as well as data sets. Therefore, we should not simply rule

out the other models. More detailed study should be conducted more learning algorithms and more data sets to investigate under which circumstances which models are more appropriate. Another more pressing issue on our immediate agenda is that as the predicting performance depends on the amount of data used in generating learning points for fitting, how we can confidently decide the proper number of learning points effectively.

References

1. S. Amari, N. Fujita, and S. Shinomoto. Four types of learning curves. *Neural Computation*, 4(4):605–618, 1992.
2. Yonathan Bard. *Nonlinear Parameter Estimation*. Academic Press, 1974.
3. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. (http://www.ics.uci.edu/∼mlearn/MLRepository.html).
4. C. Cortes. *Prediction of Generalization Ability in Learning Machines*. PhD thesis, Department of Computer Science, University of Rochester, New York, 1993.
5. L.J. Frey and D.H. Fisher. Modeling decision tree performance with the power law. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*. Morgan Kaufmann, 1999.
6. H. Gu and H. Takahashi. Exponential or polynomial learning curves?—-case-based studies. *Neural Computation*, 12(4):795–809, 2000.
7. Daniel Hyams. Curve expert: A curve fitting system for windows, 1997. (http://www.ebicom.net/ dhyams/cvxpt.htm).
8. G.H. John and P. Langley. Static versus dynamic sampling for data mining. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*. AAAI / MIT Press, 1996.
9. C.M. Kadie. *Seer: Maximum Likelihood Regression for Learning-Speed Curves*. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1995.
10. T. Lim, W. Loh, and Y. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 1999.
11. Tjen-Sien Lim. Users' guide for logdiscr version 2.0, 1999. (http://recursive-partitioning.com/logdiscr/).
12. F. Provost, D. Jensen, and T. Oates. Efficient progressive sampling. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining(KDD'99)*, pages 23–32. AAAI/MIT Press, 1999.
13. J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993. (http://www.cse.unsw.edu.au/ quinlan/).
14. D. A. Ratkowsky. *Handbook of Nonlinear Regression Models*. Marcel Dekker, INC., 1990.
15. G.A.F. Seber and C.J. Wild. *Nonlinear regression*. Wiley & Sons, 1989.