Modelling Classification Performance for Large Data Sets
An Empirical Study

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

[1] Department of Computer Science, National University of 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. For many learning algorithms, their learning accuracy will increase as the size of training data increases, forming the well-known *learning curve*. Usually a learning curve can be fitted by interpolating or extrapolating some points on it with a specified model. The obtained learning curve can then be used to predict the maximum achievable learning accuracy or to estimate the amount of data needed to achieve an expected learning accuracy, both of which will be especially meaningful to data mining on large data sets. Although some models have been proposed to model learning curves, most of them do not test their applicability to large data sets. In this paper, we focus on this issue. We empirically compare six potentially useful models by fitting learning curves of two typical classification algorithms—C4.5 (decision tree) and LOG (logistic discrimination) on eight large UCI benchmark data sets. By using all available data for learning, we fit a *full-length* learning curve; by using a small portion of the data, we fit a *part-length* learning curve. The models are then compared in terms of two performances: (1) how well they fit a *full-length* learning curve, and (2) how well a fitted *part-length* learning curve can predict learning accuracy at the full length. Experimental results show that the power law ($y = a - b * x^{-c}$) is the best among the six models in both the performances for the two algorithms and all the data sets. These results support the applicability of learning curves to data mining.

1 Introduction

1.1 Motivation

When an induction algorithm (e.g., a classification algorithm) is run 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), while the vertical axis represents *Acc*, 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). 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 [13].

In practice, the learning curve can be modelled by fitting a group of $(Size, Acc)$ points (called *learning points* hereafter in this paper), in order to avoid the redundant and expensive work of learning every size of a data set [9]. The learning curve, if available, can be very useful: (1) to predict what is a reachable learning accuracy for a given amount of data, (2) to estimate how many data should be used to achieve a desired learning accuracy, and (3) when to stop a learning process according the predictable gain in learning accuracy [5]. Such uses can find many practical applications by data mining practitioners who frequently deal with large data sets.

For large data sets, it will be more preferable to fit learning points that only cover a small amount of all available data. Mathematically, such a fitting can be implemented by interpolating the front portion of a learning curve then extrapolating to its later portion until the full length. The result of the fitting depends on how many learning points to fit and which model to use. As we are restricted to use too many learning points that may involve costly computation of learning a large amount of data, the choice of the model is actually more important.

Theoretically, given a set of points, one can fit the 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 easier operations. Another, the exact behavior of a learning curve is founded differing with the used learning algorithm as well as the used data set [1, 7]. A natural question is thus that if there exists a common model that can well fit various learning curves with respect to different learning algorithms and various data sets.

Hopefully, for its application to data mining, we need a model that satisfies the following properties: (1) be simple (with less parameters), (2) to be able to well fit a full-length learning curve, (3) to be able to well extrapolate to the full length by interpolating the early part of the learning curve, and (4) be applicable to different learning algorithms and different data sets. When models are similar in the former two properties, the last two properties are more significant to data mining. Then which model is the desired?

1.2 Related Research

The common feature in the shape of learning curves suggests a possibility of fitting 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 two-parameter power law with other three two-parameters models, namely linear model, logarithm model and exponential model, they conclude that it is the best among the four models and usable to predict error rate of unseen larger size. Note that modelling learning error is equivalent to modelling 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 them only have hundreds of instances and the largest only has about three thousand instances. Moreover, the compared linear model ($y = a * x + b$)and exponential model ($y = a * exp(-b * x)$) are inherently not suitable in fitting learning curves due to their too fast increase 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 for 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 some 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, these researches do not consider the following questions: whether there is another model better than the power law, and whether a model such as the power law can apply to large data sets as well as other learning algorithms. In this work, we address our attention to these unsolved issues.

1.3 Our Contribution

In this study, we empirically compare six models found in the mathematical community and potentially usable for fitting learning curves, investigate which model performs best (in the sense of the above mentioned properties) for eight large UCI benchmark data sets and two typical classification algorithms, namely C4.5 (decision tree approach) and LOG (logistic discrimination approach). The six models, namely two-parameters power law, three-parameters power law, logarithm model, vapor pressure model, Morgan-Mercer-Flodin (MMF) model, and weibull model [16], come from three function families with two to four parameters. All of them have shapes similar to that of a typical learning curve.

Learning points are obtained by running the two classification algorithms on eight UCI data sets. By using all available data of a data set for learning, we fit a *full-length* learning curve; by using a small portion of the data, we fit a *part-length* learning curve. The models are then compared in terms of two performances: (1) how well they fit a *full-length* learning curve (called *fitting performance* hereafter in this paper), and (2) how well a fitted *part-length* learning curve can predict (extrapolate) the learning accuracy at the full length (called *predicting performance* hereafter in this paper). Our experiments show that the three-parameter power law ($y = a - b * x^{-c}$) is overall the best model in both the performances for the two algorithms and all the data sets. This result provides empirical support for applying the power law to fitting learning curves for large data sets, thus paves the way for applying learning curves to data mining.

2 Candidate Models, Algorithms 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 probably by tuning their parameters. Note that their parameters vary from two to four. From the view of mathematics, a model with too many parameters is either hard to converge or subject to over-fitting [16]. From the view of practice, such a model is also difficult to use. Therefore, we do not consider models with more than four parameters.

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

To facilitate the fitting process, we use the *Levenberg-Marquardt* method [2, 17] to solve nonlinear regression. This method combines the steepest-descent method and a Taylor series based method and has been proven to be a fast reliable technique for nonlinear optimization. We use the following setting in fitting with all models: random-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.

We choose C4.5 [15] and LOG [11] as representatives of two main categories of classification algorithms: decision tree method and statistical method [12]. 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

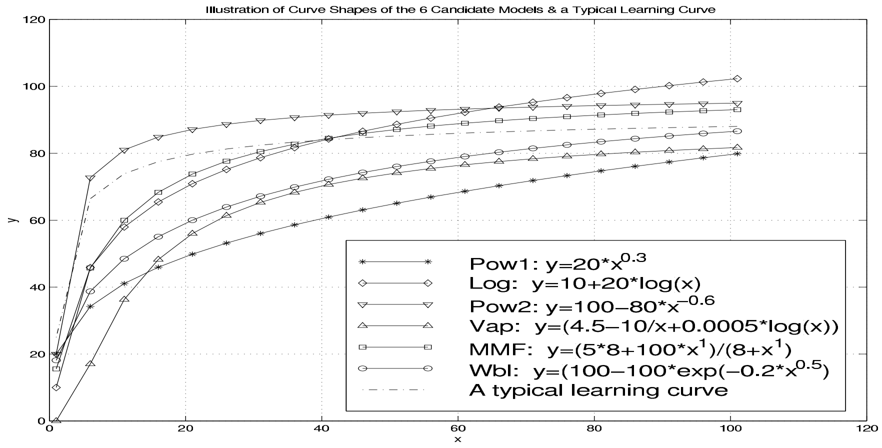

Fig. 1. Illustration of Curve Shapes of the 6 Candidate Models and a Typical Learning Curve

of many classifiers [10] has reported that both algorithms tend to produce high classification accuracy. The eight large UCI benchmark data sets [3] used in this study are summarized in Table 2. Note that the number of instances of these data sets range from three thousands to eighty thousands. Although they are not "very large" from the view of many real world applications of data mining [14], they are fairly "large" from the view of research.

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

3 Fitting Performance

3.1 Criterion of Comparison

We use standard error of the estimates S to measure the performance of non-linear 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 $f(x)$ 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

For each data set, we fit its C4.5 learning curve and LOG learning curve respectively. To get learning points for fitting a full-length learning curve, we use an uniform sampling size scheme: 1%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100% of the all available data. For each size, a random sample is drawn from the total training data and used to train a 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. Then each of the six candidate models is used to fit these learning points. Due to the space limit, we omit the results of fitting C4.5 and LOG learning curves. Readers can refer to [6] for detailed results. The overall ranking of the six models in terms of S value on the eight data sets are shown in Table 3 and Table 4. From the four tables, we observe that:

Table 3. Ranks of C4.5 Fitting

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

(1) In most cases, the models with more parameters tend to have smaller S values, thus are ranked in front, suggesting their higher performance in fitting. This is not surprising from the view of mathematics.

Table 4. Ranks of LOG Fitting

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

(2) In most cases, the models with the same number of parameters tend to have close S values, suggesting their similar performance in fitting. This is also not surprising from the view of mathematics.
(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.

Note that using one more parameter does increase the complexity of the fitting process. But, if it did not significantly increase the fitting accuracy, we would rather not use it. The above observations suggest that models with two parameters seems not enough to fit well a full-length learning curve, while using four parameters seems redundant. Therefore, in fitting learning curves, a model with three parameters seems a trade-off between model complexity and model accuracy. In this sense, the three-parameters power law should be the best in fitting the full-length learning curves.

However, this result is less attractive to data mining, because obtaining a full-length learning curve requires the fitted learning points cover the x-axis range from a small size up to the full size of the data set. And to obtain the learning accuracy at the large or full size, we must run the learning algorithm on a large amount of the available data thus involve high computational cost. This is obviously opposite to our motivation of applying learning curve to data mining. Therefore, it is necessary to examine the predicting performance of the models as we describe in the next section.

4 Predicting Performance

A model may become less attractive to learning on large data sets if it is strong in *fitting performance* but weak in *predicting performance*. Given two models with similar fitting performance, the one with higher predicting performance is more desirable to data mining. In this section, we compare the six candidate models with respect to this performance. Instead of fitting a *full-length* learning curve for each data set, we intentionally fit a *part-length* one on a small portion of the whole available data, then we extrapolate it to the full length to see whether it can predict the learning accuracy at that size accurately. The predicted accuracy is 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 the better.

However, this implies an essential but still open issue: how many instances are adequate to support a good extrapolation. Although we hope to use as fewer instances as possible to obtain enough learning points (the pairs of $(Size, Acc)$), the learning points corresponding to too small amount of instances will exhibit too little information about the shape of the whole learning curve so that no model can properly "guess" the latter trend of the curve. This is obvious in mathematics: given a segment of a curve, one can arbitrarily extend it to any shape. Frey and Fisher [5] mention this issue, however, they only test a 15% portion of a data set. Intuitively, the larger the portion, the more accurate the extrapolation. To find out how the predicting performance will be affected by size of the portions, we intentionally test four portions for each data set, covering 5%, 10%, 15%, and 20% of the total data. For each portion, A part-length learning curve is fitted from 10 learning points that equally span the interval of the portion. 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. From the two tables, we have the following observations:

(1) 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% portion and 10% portion).

(2) 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 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.

(3) There is no model that consistently makes the best prediction for all portions on all data sets.

(4) The averaged ranking show that the three-parameters power law tends to have the best predicting performance.

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
Average	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

The first two observations are not out of our expectations, because similar observations have been shown by previous studies (but typically for small data sets). And the two observations may also account much for the third observation. In the presence of the three observations, the fourth observation shows that the three-parameters power law is more robust than the others with respect to the two classification algorithms as well as the eight data sets, and also more robust with respect to the amount of data used in fitting the *part-length* learning curves. Although the MMF and Weibull are better than the three-parameter power law in accuracy when fitting *full-length* learning curves, they lose to the latter in the predicting performance. This is probably because given only a segment of a whole learning curve, models with four parameters tend to overfit it, while models with two parameters tend to underfit it, thus both negatively affect their extrapolation. Combined with the results of fitting performance as shown in the previous section, these results show that the three-parameter power law is the best in both performance thus should be the first considered model to fit learning curve for the large data sets in our experiments.

5 Conclusion and Future Research

The benefit of obtaining an accurate learning curve of large data sets lies in that we are able to use it to predict: (1) what is the achievable learning accuracy for a given amount of training data, (2) how many training data is needed to reach a desirable learning accuracy, and (3) when to stop the learning process according the predictable gain in learning accuracy. The benefit of choosing a good model for fitting learning curves lies in that we are able to use it to obtain the desired learning curve without too much learning effort that may cover a large amount of data, thus can save much computation cost.

In this empirical study, aiming at finding a model that fits well learning curves for large data sets, we compare six candidate models that come from three functional families with two to four parameters. We compare their fitting performance (how well they fit a *full-length* learning curves) and predicting performance (how well they extrapolate to the full length by a fitted *part-length* learning curve), for eight large UCI benchmark data sets and two typical classification algorithms. Our experimental results show that the three-parameters power law ($y = a - b * x^{-c}$) is the best among the six models in both the performances. It also exhibits another salient advantage: it is less susceptible to the two classification algorithms and the eight benchmark data sets than the other models. This suggests its possible applicability to other learning algorithms and various data sets. Therefore our results should be useful in facilitating various data mining tasks 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 with 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 this amount 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. B.H. Gu, F.F. Hu, and H. Liu. An empirical study of fitting learning curves. Technical report, Department of Computer Science, National University of Singapore, 2001.
7. H. Gu and H. Takahashi. Exponential or polynomial learning curves?—-case-based studies. *Neural Computation*, 12(4):795–809, 2000.
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. D. Michie, D.J. Spiegelhalter, and C.C. Taylor. *Machine Learning, Neural and Statistical Classification.* Ellis Horwood Limited, Campus 400, Maylands Avenue, Hemel Hempstead, Hertfordshire, HP2 7EZ, England, 1994.
13. 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.
14. F. Provost and V. Kolluri. A survey of methods for scaling up inductive algorithms. *Machine Learning*, pages 1–42, 1999.
15. J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993. (http://www.cse.unsw.edu.au/ quinlan/).
16. D. A. Ratkowsky. *Handbook of Nonlinear Regression Models.* Marcel Dekker, INC., 1990.
17. G.A.F. Seber and C.J. Wild. *Nonlinear regression.* Wiley & Sons, 1989.