Generating Well-Behaved Learning Curves: An Empirical Study

Gary M. Weiss and Alexander Battistin
Department of Computer & Information Science, Fordham University, Bronx NY, USA

Abstract**—***Data mining is an important discipline that helps extract useful knowledge from data in business, science, health, and engineering domains. Classification is one of the most common and important data mining tasks. Achieving good classification performance is critical and performance is known to be linked to the amount of available training data. Learning curves, which describe the relationship between training set size and classifier performance, can be used to help determine the optimal amount of training data to use when there are costs associated with procuring labeled data. For learning curves to be helpful, they should be good predictors of future performance, which means that they should be "well-behaved" (i.e., smooth and monotonically non-decreasing). This paper describes how various factors, such as the classification algorithm and experiment methodology (e.g., random sampling vs. cross validation), affect the behavior of learning curves.*

Keywords: classification, learning curves, methodology

1. Introduction

Classification is one of the most important and common data mining tasks. It is well known that classification performance improves with increasing amounts of training data. This can be visually demonstrated via a learning curve, which plots training set size on the x-axis and classifier performance (e.g., accuracy) on the y-axis. The prototypical learning curve is described as follows: performance improves quickly at the start when there is not sufficient data to properly learn the underlying concept well, then the learning curve's slope begins to decrease as an adequate amount training data becomes available, then in the last phase the curve begins to flatten out and the slope approaches 0, as additional training data provides little additional information. However, as was shown in prior research, even in this last phase, small improvements in classifier performance can persist for quite a long time [3].

Data mining work often assumes that there is a fixed amount of training data, available at no cost, and that additional data cannot be procured. This situation undoubtedly fits some real-world situations, but not all. In reality, it is often possible to procure additional training data, but at a cost. This cost could be related to the cost of procuring the data itself, labeling the data (requiring a human and often a human domain expert), or both. In such a situation, a learning curve can be utilized to assess the costs and benefits of obtaining additional training data, and then make the optimal decision. Of course, in practice one cannot form a learning curve without first obtaining training data, so one will always have to predict future learning curve behavior based on the current available training data. In order to make such predictions accurate, it is best if the learning curve is well-behaved—smooth and monotonically non-decreasing.

There has been relatively little related work on utilizing learning curves or generating well-behaved learning curves. Provost, Jensen and Oates [2] evaluated progressive sampling strategies in order to efficiently identify the point where learning curve performance begins to plateau. Weiss and Tian [3] looked at how learning curves can be used to optimize classifier utility when the utility includes classifier performance, CPU time, and data acquisition costs. Both of these research studies would benefit from well-behaved learning curves. Weiss and Tian [3] specifically acknowledged this in their paper when they said, "Because the analyses are all driven by the learning curves, any method for improving the quality of the learning curves (i.e., smoothness, monotonicity) would improve the quality of our results, especially the effectiveness of the progressive sampling strategies." The work in this paper can be viewed as addressing this prior research challenge.

In this paper, we will generate learning curves for six data sets, using four different classification algorithms, and two methodologies for partitioning the data and running the experiments (random sampling and cross-validation). We will visually inspect several of the learning curves to check for monotonicity, but will also look at the variance of the classification performance results. Our hope is that this work will bring attention to the importance of learning curves and will show which factors tend to produce good learning curves and consistent results with low variance. Classification algorithms are currently judged on a number of factors: quality of results, speed of model generation, speed of model application, scalability, and the understandability of the induced model. We would like the consistency (i.e., variance) of the results, which impacts the quality of the learning curves, to be considered as an additional characteristic when evaluating learning methods and learning methodologies.

2. Experiment Methodology

The experiments in this paper are used to assess how various factors impact the quality of generated learning curves. Learning curves are generated by varying the training set sizes for the data sets listed in Table I. Most of these data sets are fairly large, which enables us to generate learning curves that span a large range of training set sizes—which will help with the evaluation of the quality of the learning curves. Training set sizes are sampled at regular 2% intervals, based on the total amount of data available for training. For 10-fold cross validation, the total amount of data available for training is 90% of the total in Table I, while for random sampling 75% of the total is available, since for all of our experiments random sampling initially allocates 75% of the data for training and 25% for testing.

The Adult, Kr-vs-kp, German, and Arrhythmia datasets are from the UCI Machine Learning Repository [1] while the Coding, Blackjack, \Boa1, Network1, and Move data sets were obtained from researchers at AT&T and can be obtained from the authors.

TABLE I
DESCRIPTION OF DATA SETS

Dataset	# Examples	# Classes	# Attributes
Adult	32,561	2	14
Coding	20,000	2	15
Blackjack	15,000	2	4
Boa1	11,000	2	68
Network1	3,577	2	30
Kr-vs-Kp	3,196	2	36
Move	3,029	2	10
German	1,000	2	20
Arrhythmia	452	2	279

Learning curves are generated using three classification algorithms from the WEKA data mining suite [4]: J48, Random Forest (RF), and Naïve Bayes (NB). J48 is a WEKA implementation of the C4.5 decision tree algorithm. Weka's experimenter mode, as described in an online tutorial [4], was utilized to facilitate the generation of the learning curves. Unless otherwise specified, all results in this paper are based on 10 runs.

3. Results

In this section we evaluate the quality of the learning curves with respect to learning algorithm and experiment methodology (i.e., partitioning strategy). However, well-behaved learning curves are not very useful if classifier performance is not good. Thus, it is important to also know how well the learning methods perform. While the learning curves encode classifier performance, because we use a different graph for each classification algorithm, the relative performance of each learning method may not be apparent from the learning curves. Therefore we display the classifier accuracy for each learning algorithm, at the maximum training set size, in Table II (for 10-fold cross validation). The results show that Random Forest and J48 have the highest average accuracies and significantly outperform Naïve Bayes (Random Forest has a slight overall advantage over J48 even though J48 performs best on 4 of 9 data sets).

TABLE II
ACCURACY WITH LARGEST TRAINING SIZE

Dataset	J48	Random Forest	Naïve Bayes
Adult	86.3	84.3	83.4
Coding	72.2	79.3	71.2
Blackjack	72.3	71.7	67.8
Boa1	54.7	56.0	58.0
Network1	77.3	77.6	74.8
Kr-vs-kp	99.4	98.7	87.8
Move	76.0	80.3	65.2
German	71.1	74.1	75.2
Arrhythmia	65.4	65.2	62.0
Average	75.0	76.4	71.7

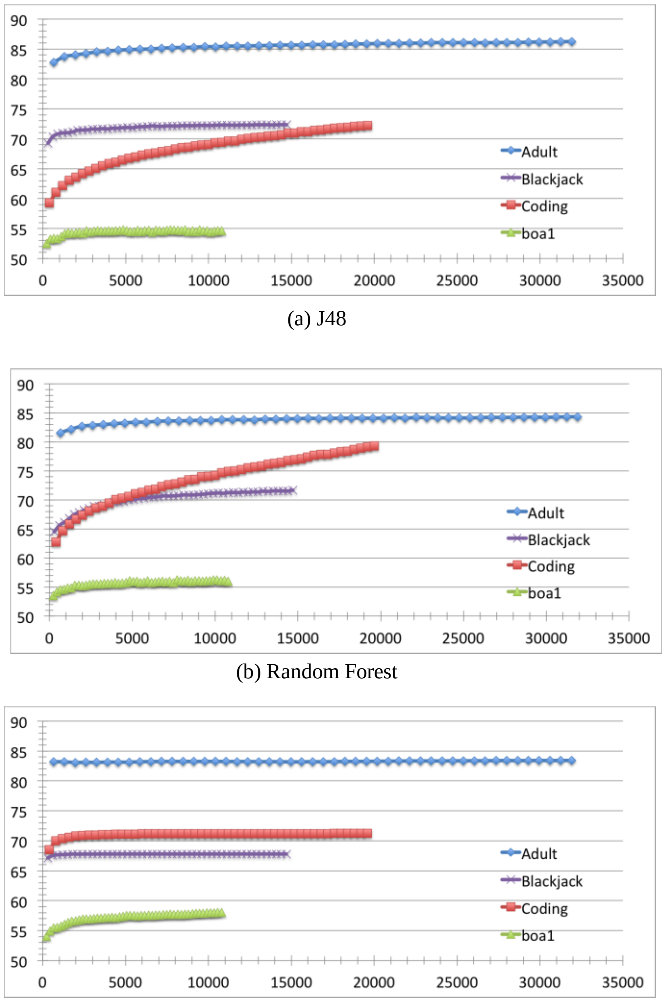
The quality, or monotonicity, of a learning curve can be assessed visually, but a more objective and easily summarized measure is the "variance" of the learning curve. The variance of a learning curve is computed by determining the variance in classifier performance for each evaluated training set size (based on multiple runs) and then averaging these individual variances. The results of the learning curve variances, using 10 runs of 10-fold cross validation, are displayed in Table III.

The results in Table III clearly show that Naïve Bayes generates the lowest variance overall, although for the Boa1 data set it actually has the highest variance (but all values are so low that this may not be too meaningful). Overall J48 and Random Forest seem to perform similarly. Given that the data in Table II showed that J48 and Random Forest produced the most accurate results, J48 would seem to be the best classifier when factoring in accuracy and consistency of results. It should be pointed out that because variance measures the consistency of results for a given training set size, it is theoretically possible to have low variance but have a curve that is not smooth. However, this is very unlikely given that consistent results should lead to the expected behavior—a learning curve that is monotonically non-decreasing.

TABLE III
VARIANCES FOR LEARNING CURVES USING 10-FOLD CV

Dataset	J48	Random Forest	Naïve Bayes
Adult	0.51	0.32	0.01
Coding	9.78	17.08	0.19
Blackjack	0.36	2.81	0.01
Boa1	0.20	0.31	0.73
Network1	2.37	2.19	0.10
Kr-vs-kp	3.54	12.08	4.34
Move	28.65	24.73	1.02
German	4.38	1.97	3.48
Arrhythmia	41.46	15.87	9.90

Our first set of learning curves, comprising the four largest data sets, is presented in Figure 1. The curves seem to be well-behaved in that they all appear to be monotonically non-decreasing. Although most of the curves seem quite smooth, the curves for Naïve Bayes appear to be smoother.



(a) J48

(b) Random Forest

(c) Naïve Bayes

Figure 1. Learning curves generated using 10-fold cross validation on the four large data sets. Each chart (a-c) shows the results for a different learning algorithm.

Since J48 and Random Forest are close in variance and accuracies, it is worth taking a more detailed look at each for a specific dataset. In Figure 2 we compare these two algorithms for the adult data set. The results clearly show that J48 generates more accurate results, but also a much better behaved learning curve—with far fewer "blips" where a larger training set size yields a decrease in accuracy.

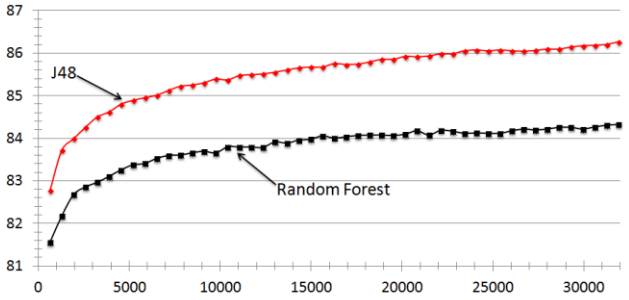
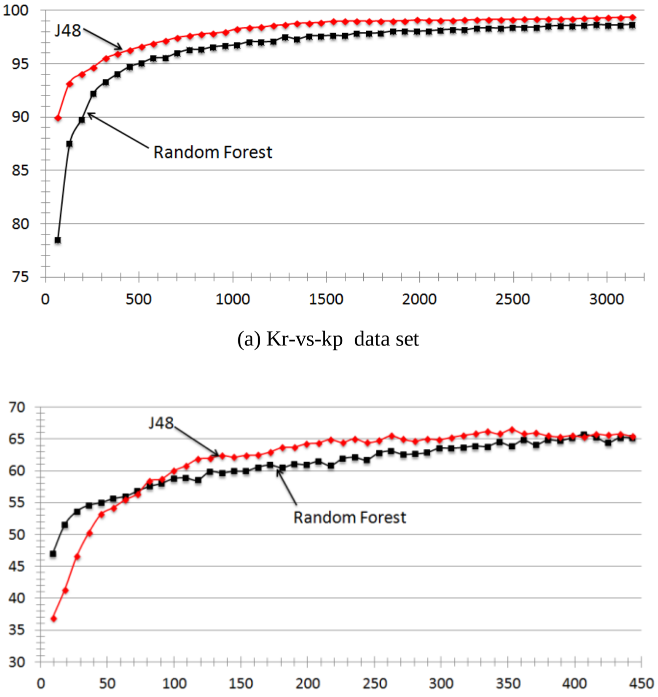


Figure 2. Comparison of J48 and RF cross-validation learning curves for Adult data set.

Next we take a look at the two smaller data sets: Kr-vs-kp and Arrhythmia. We focus on the learning curves for J48 and Random Forest. The results are displayed in Figure 3. We see that for the Kr-vs-kp data set J48 appears to provide a smoother learning curve, which is consistent with the results in Table III that shows that J48 has lower variance. The results for the Arrhythmia data set are not so clear: the results in table III suggest that Random Forest produces better learning curves but based on Figure 3b this is unclear. However, the difference could be explained by the fact that J48 performs extremely poorly for very low training set sizes (worse than guessing the majority class) and the poor performance permits increased variance in results. In the future such small data sets perhaps should be omitted, or the training set sizes should not be permitted to become so small—the smallest evaluated training set size in Figure 3b corresponds to a training set with just nine examples.

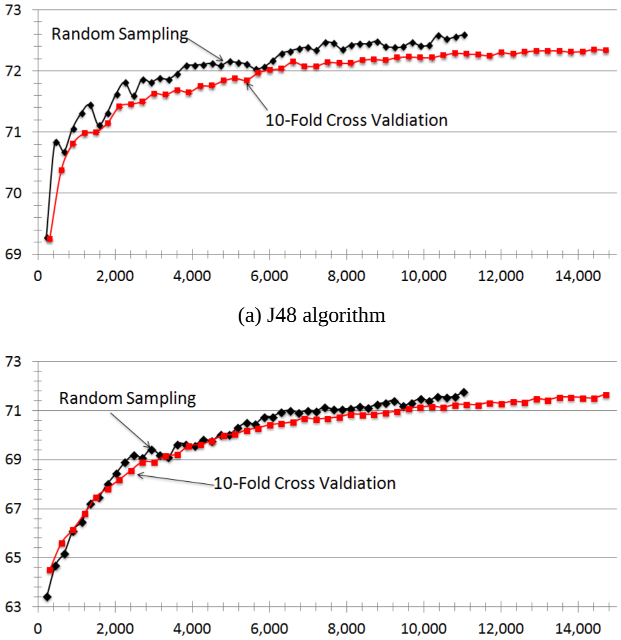


(a) Kr-vs-kp data set

(b) Arrhythmia data set

Figure 3. Comparison of J48 and RF cross validation learning curves.

Thus far we have only examined learning curves generated via 10-fold cross validation. The learning curves generated for Random Sampling are generally not as well behaved as those generated using cross-validation (CV). Due to space concerns we do not show the learning curves for every data set, but instead focus on the Blackjack data set. The learning curves for this data set, shown in Figure 4, indicate that for both J48 and Random Forest, the learning curves generated using cross validation are better behaved.



(a) J48 algorithm

(b) Random Forest algorithm

Figure 4. Cross validation versus random sampling for Blackjack data set

The results in Figure 4 support our general claim that cross validation produces better-behaved curves, although in this case for a given training set size they produce slightly lower accuracies. However, the variance results, displayed in Table IV, are not quite so clear. The variance results show that cross validation is consistently better than random sampling when the models are induced using J48, but that when the models are induced using Random Forest, the two sampling schemes yield similar, although inconsistent, results.

TABLE IV
VARIANCES FOR CROSS VALIDATION AND RANDOM SAMPLING METHODOLOGIES FOR J48 AND RANDOM FOREST ALGORITHMS

Dataset	J48 CV	J48 RS	RF CV	RF RS
Coding	9.78	9.47	17.08	12.97
Adult	0.51	0.53	0.32	15.81
Blackjack	0.36	0.38	2.81	0.33
Boa1	0.20	0.29	0.31	3.80
Arrhythmia	41.46	51.37	15.87	0.19
Kr-vs-kp	3.54	5.36	12.08	15.49
Network1	2.37	2.37	2.19	1.78
Move	28.65	31.74	24.73	27.05
German	4.38	4.51	1.97	3.05

4. Conclusion

In this paper we examined how various factors impact how "well behaved" a learning curve is, based on monotonicity and low variance in classification performance. We focused on the how different classification algorithms and experiment methodologies impact the learning curves and then drew some conclusions based on our empirical results.

Of the learners that we evaluated, Naïve Bayes seems to produce the best-behaved learning curves. However, we do not recommend Naïve Bayes for two reasons: 1) based on Table II its accuracy is not competitive with J48 and Random Forest and 2) examination of the learning curves in Figure 1c indicates that Naïve Bayes' learning curves reach a plateau much earlier than the other methods, suggesting that perhaps the low variance is a consequence of achieving a consistent (but poor) level of performance. While we cannot prove this latter point, it makes sense that once additional data does not improve results, the exact subset of examples used for training may not matter. Given the issues with Naïve Bayes, our recommendation is to use J48 and Random Forest. The comparison of variance results for these two methods is inconsistent: in some cases J48 performs best and in others Random Forest performs best. Therefore based on the results in this paper over a limited number of data sets, we cannot conclude which method generates the best-behaved learning curves. In terms of methodology, the learning curves indicate that cross validation yields better-behaved learning curves than random sampling, as supported by the results in Figure 4. However, the variance results in Table IV are not nearly as conclusive. Thus, this also bears further investigation.

There are various areas for future research that we intend to pursue. First, we intend to analyze more data sets so that we can form stronger conclusions based on a larger sample size. We also plan to analyze a few additional learning algorithms. Better metrics can also help by measuring the "well-behavedness" of learning curves and we have some ideas on how to construct such metrics. Finally, we will vary the number of runs to see how this impacts the learning curves. Once some of these extensions have been implemented, we feel it is likely that stronger conclusions will be possible.

5. References

[1] K. Bache, and M. Lichman, M. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.

[2] F. Provost, D. Jensen, and T. Oates, :Efficient progressive sampling," in *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 23–32.

[3] G. M. Weiss and Y. Tian, "Maximizing classifier utility when there are data acquisition and modeling costs," *Data Mining and Knowledge Discovery*, 17(2): 253-282, 2008.

[4] Weka Learning Curves, http://weka.wikispaces.com/Learning+curves.

[5] I. H. Witten, E. Frank, M. A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques,"3rd Edition, Morgan Kaufmann, 2011.