Effecting Transfer via Learning Curve Analysis

Christophe Giraud-Carrier and Dan Ventura
Brigham Young University, Department of Computer Science
Provo, UT 84602, USA

Abstract

We consider the idea of using learning curves to characterize learning tasks and ask the question of whether such a characterization is useful for inductive transfer. We compute a database of learning curves for 49 datasets and 9 learning algorithms and demonstrate its utility in transferring information about expected performance.

1 Introduction

Popular in the early days of Machine Learning research, learning curves have mostly been ignored in recent years. Single performance measures, such as n-fold cross-validation accuracy and or area under the ROC curve, have become a kind of de facto standard. Unlike these singular measures, learning curves depict the *evolution* of some learning performance measure as a function of the size of the training set.

Recent results using learning curves highlight the fact that training set size must be taken into account when comparing algorithms' performance, as conclusions regarding relative predictive performance for small training set sizes may be reversed as training set size increases [5]. Learning curves have also recently been used as the basis for a method to predict the relative performance of a pair of learning algorithms [4]. First, a library of learning curves is computed with progressive sampling [3, 6]. New datasets are then matched against the database using information from partially computed learning curves with the goal of predicting which of two algorithms will produce the better accuracy on the dataset in question. We extend and generalize this idea to demonstrate knowledge transfer about expected performance.

2 Experimental Method

Assuming that the metric of interest is accuracy, we define a learning curve $C_{a,d}$ as a function parameterized by the number of training instances, such that $C_{a,d} : \mathbb{N} \rightarrow (0, 1)$ and $C_{a,d}(n) = p$, where p is the percentage of n training instances from d that a correctly classifies. We can estimate $C_{a,d}$ by training a on a random sample of d for various values of n, testing the resulting model on a holdout set to obtain a point estimate and interpolating between our point estimates. The standard arguments for curve estimation apply, of course.

Given a meta-database of datasets together with their associated learning curves

and the partial learning curve of a new query dataset, we want to know whether it is possible to predict accurately:

- The best performance one might expect on the query dataset.
- How many examples would be needed to achieve a pre-defined level of performance on the query dataset.

In other words, we wish to know whether our experience at learning tasks in the past can be used to predict how well we might do at tasks in the future. Note that predicting the above is somewhat akin to deciding the PAC-learnability of the query dataset (in the sense of finding ϵ and N) [2, 7].

Part of the answer to the above questions hinges on whether or not datasets have characteristic learning curves. Although one may argue that a learning curve is likely to be affected by both the dataset and the learning algorithm, our hope is that sufficient information may be gleaned from aggregate learning curves (over datasets) to make prediction possible.

We report on a series of experiments using learning algorithms from the Weka toolkit [8] and datasets from the UCI machine learning repository [1]. For all experiments we use the algorithms' default settings. The following describes the experimental setup:

1. For each algorithm and each dataset, we compute a learning curve, yielding a set of learning curves, $\{C_{a,d}\}$. To do this, we estimate the curves at points defined as 5%, 10%, ..., 95% of the data available by making a random draw from d, training a and computing an accuracy on the data not used for training. At each point, we repeat the experiment 10 times and average to avoid potential problems with sample selection and presentation order.
2. For each dataset d, we compute an aggregate learning curve, C_d, by averaging d's learning curves over all learning algorithms.
3. For each query dataset e,
   (a) We compute C_e up to some pre-defined point (e.g., 20%) by averaging the corresponding partial learning curves $C_{a,e}$
   (b) We find the aggregate curve C_d closest to C_e using the R^2 goodness of fit measure
   (c) We predict e's performance from the latter part of C_d and compare it with the actual performance by completing C_e.

3 Experimental Results

Our experiments are based on a total of 49 different data sets (Abalone, Adult-all, Anneal, Audiology, Balance-scale, Breast cancer, Breast Wisconsin, Bupa, Car, Cmc, Connect-4, Credit-australian, Credit-german, Dermatology, Diabetes, Ecoli, Glass, Haberman, Heart-c, Heart-h, Heart-statlog, Hepatitis, Hypothyroid, Ionosphere, Iris, Isolet5, Kr-vs-kp, Labor, Letter, Lymph, Mushroom, Nursery, Page-blocks, Post-operative, Primary-tumor, Segment, Sick, Sonar, Soybean, Spambase, Tae, Tic-tac-toe, Vehicle, Vote, Vowel, Waveform, Wine, Yeast, Zoo) and 9 different algorithms (J48, Naive Bayes, Conjunctive Rules, NNge, HyperPipes, IB3, Multi-layer Perceptron, an RBF network, and SMO).

Figure 1 shows, for each dataset e (leave-one-out procedure), how the root mean squared (RMS) error between e and its closest match evolves as the length of the partial learning curve for e increases.

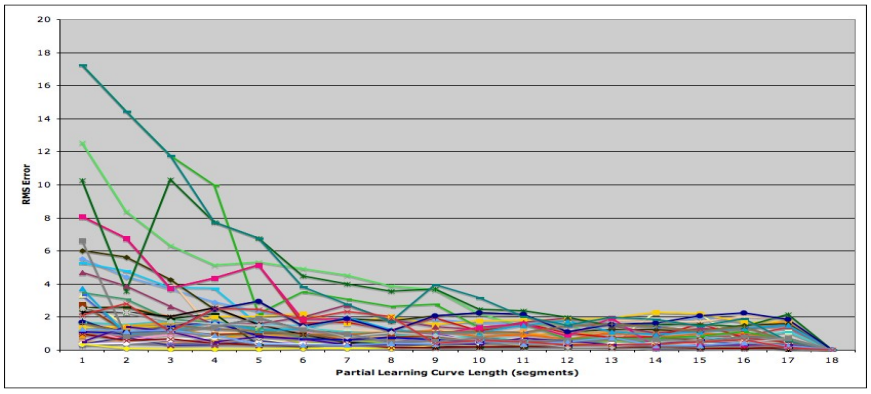

Figure 1: Error vs Partial Learning Curve Length

The graph clearly shows that the error rapidly converges towards 0. In fact, for 45 datasets, considering five curve segments is sufficient to achieve very low error. To gain further insight into the behavior of each dataset, Figure 2 provides a kind of spectrum of homogeneity.

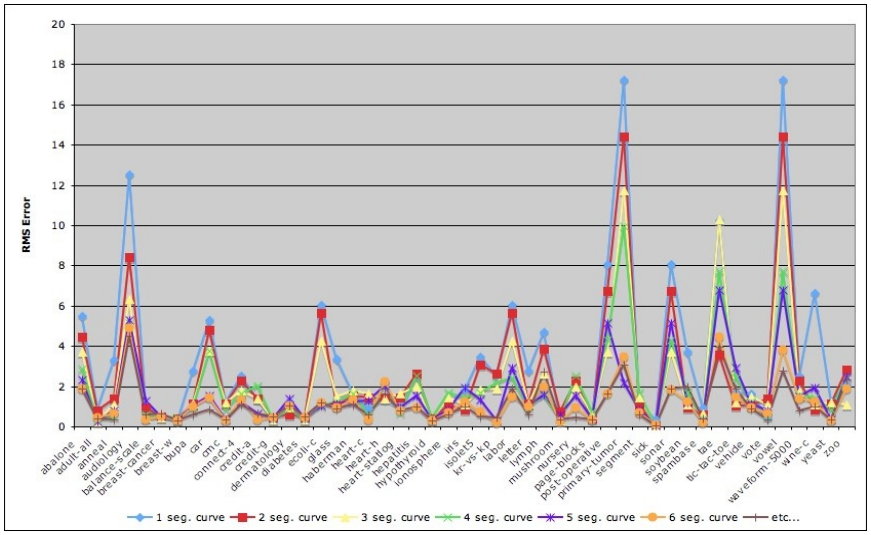

Figure 2: Error per Dataset vs Partial Learning Curve Length

Peaks in the spectrum correspond to datasets whose learning curves are significantly different from the others. The three high peaks in Figure 2 correspond to the three datasets with slow convergence in Figure 1.

Finally, Figure 3 shows how the number of datasets whose predictor is within ϵ (as measured by RMS) evolves with the length of the partial learning curve. Note, for example, that for 26 datasets, a partial learning curve a single segment in length produces a RMS error prediction of less than 5%; also note that with partial curves five segments long, 19 datasets have a RMS error prediction of less than 1%.

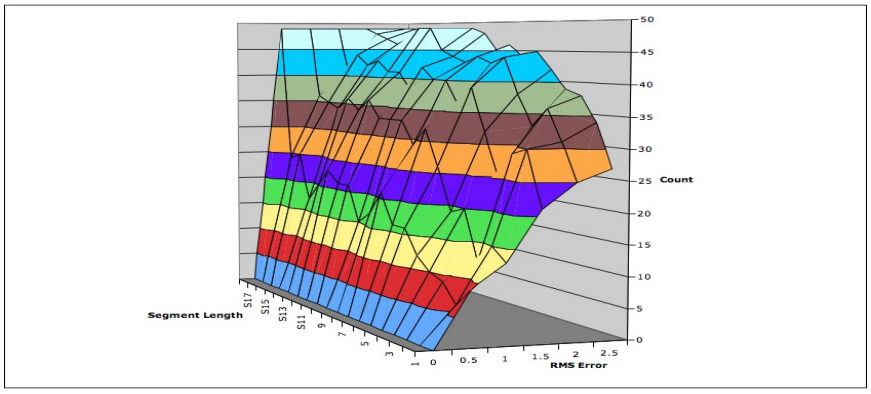

Figure 3: Transfer Utility as a Function of Partial Learning Curve Length

4 Conclusion

We have shown how learning curves may be used to effect a simple form of transfer. Our preliminary results are encouraging and demonstrate that some information about performance on new datasets may be inferred from information about performance in the past.

There are a number of important issues that are the subject of ongoing research, including the best way to sample when building the learning curves, whether learning curves should be normalized in some way, what impact the size of the feature space may have, and whether some notion of curve variance might improve results.

References

[1] C.I. Blake and C.J. Merz. UCI repository of machine learning databases. University of California, Irvine, Department of Information and Computer Science, 1998.

[2] M.J. Kearns and U.V. Vazirani. *An Introduction to Computational learning Theory*. The MIT Press, 1994.

[3] R. Leite and P. Brazdil. Improving progressive sampling via meta-learning on learning curves. In *Proceedings of the Fifteenth European Conference on Machine Learning*, pages 250–261, 2004.

[4] R. Leite and P. Brazdil. Predicting relative performance of classifiers from samples. In *Proceedings of the Twenty-second International Conference on Machine Learning*, 2005.

[5] C. Perlich, F. Provost, and J.S. Simonoff. Tree induction vs. logistic regression: A learning-curve analysis. *Journal of Machine Learning Research*, 4(2):211–255, 2003.

[6] F. Provost, D. Jensen, and T. Oates. Efficient progressive sampling. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, pages 23–32, 1999.

[7] L.G. Valiant. A theory of the learnable. *Communications of the ACM*, 17(11):1134–1142, 1984.

[8] I.H. Witten and F. Eibe. *Data Mining: Practical Machine Learning Tools with Java Implementations*. Morgan Kaufmann, San Francisco, CA, 2000.