Model Combination in the multiple-data-batches scenario

Kai Ming Ting[1] and Boon Toh Low[2]

[1] Department of Computer Science,
University of Waikato, Hamilton, New Zealand.
E-mail: kaiming@cs.waikato.ac.nz
[2] Department of Systems Engineering and Engineering Management,
Chinese University of Hong Kong, Shatin, Hong Kong
E-mail: btlow@se.cuhk.edu.hk

Abstract. The approach of combining models learned from multiple batches of data provide an alternative to the common practice of learning one model from all the available data (i.e., the data combination approach). This paper empirically examines the base-line behaviour of the model combination approach in this multiple-data-batches scenario. We find that model combination can lead to better performance even if the disjoint batches of data are drawn randomly from a larger sample, and relate the relative performance of the two approaches to the learning curve of the classifier used.
The practical implication of our results is that one should consider using model combination rather than data combination, especially when multiple batches of data for the same task are readily available.
Another interesting result is that we empirically show that the near-asymptotic performance of a single model, in some classification task, can be significantly improved by combining multiple models (derived from the same algorithm) if the constituent models are substantially different and there is some regularity in the models to be exploited by the combination method used. Comparisons with known theoretical results are also provided.

Keywords: model combination, data combination, empirical evaluation, learning curve, near-asymptotic performance.

1 Introduction

When different batches of data for the same task are available, the usual approach is to combine all available data and produce one classifier. This is an intuitive approach that stems from the conventional wisdom: "more data the better". Here we investigate an approach which differs from the way data is utilised. It learns one classifier for each batch of data (using the same learning algorithm) and then combines the classifiers' predictions. We call the former "*data combination*" and the latter "*model combination*". Figure 1 shows these two types of combination at the data and model levels.

While there has been a considerable amount of research on methods to combine multiple models reported in the literature (e.g., Brodley, 1993; Breiman,

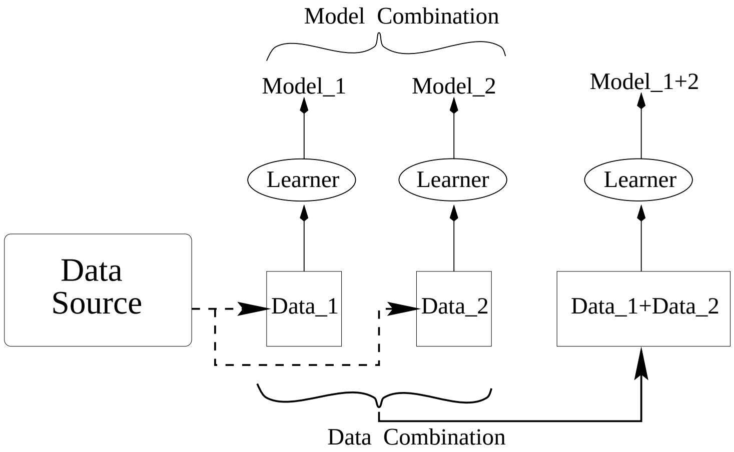

Fig. 1. Combination at different levels – data or model.

1996a,1996b; Freund & Schapire, 1996; Perrone & Cooper, 1993; Krogh & Vedelsby, 1995), investigation into combining models from a single learning algorithm induced using *completely disjoint sets of data* has been limited. Most work shows that combining multiple models induced from either one type or different types of learning algorithm using *a single dataset* performs better than a single model induced from the same dataset.

This paper concentrates on a situation where multiple batches of data are available for a single classification task. The scenario might be collections of data in consecutive years or in different events from the same source. We attempt to determine the conditions under which model combination performs better than data combination in this scenario. Thus, our focus here is not on different types of model combination method. Specifically, we address the question: is model combination a viable option as compared to data combination in classification tasks? If the answer is yes, when should one use it?

The intuition is that different batches of data provide some variation of data representation in the description space. Theories induced separately from these independent batches become "specialists" in different parts of the space. Model combination allows cooperation between these specialists. Data combination destroys the data variation, and forms a global representation.

Some research on multiple model combination focus on varying the *induction bias* of the learning algorithm(s) to generate models of uncorrelated errors. These include varying learning parameters of a single learning algorithm and using different types of learning algorithm. Others use sampling methods to create *multiple overlapping data subsets* from a given dataset. Here we show that *data variation in different data batches*, even if the disjoint batches are drawn randomly from a larger sample, can also produce substantially different models from a single learning algorithm. We review related work on using multiple models to enhance performance in the next section. The experimental design based on a hypothesis is described in Section 3. The results of the experiments are reported in Section 4 and followed by discussion and conclusions.

2 Related Work

We focus our review on how multiple models are generated, with just a minor note on how they are combined since that is not our emphasis.

Some work on multiple models employs sampling methods to generate the models, e.g., bagging (Breiman, 1996a) and boosting (Freund & Schapire, 1996; Quinlan, 1996; Breiman, 1996b). Each sample dataset has either the same data size as the available dataset or a high percentage of the total instances. A set of n classifiers are produced from n sets of samples and they are combined by voting or weighted voting. Ali and Pazzani (1996) use a k-fold partitioning to generate k models by training on all but the ith partition k times. Breiman (1996c) investigates boosting models derived using small bites of the entire dataset. In this formalism, the multiple models are usually produced from a single learning algorithm.

Multiple models can also be produced by varying the learning parameters of a single learning algorithm. Work in generating multiple neural networks (Hansen & Salamon, 1990; Perrone & Cooper, 1993) by using different initial random weight configurations or/and orders of training data; multiple decision trees (Kwok & Carter, 1990; Buntine, 1991; Oliver & Hand, 1995) by selecting different tests at each node, generating option trees, or pruning a tree in different ways; and multiple rules (Kononenko & Kovačič, 1992) by stochastic search guided by heuristics. These works re-order the rank of the classes by (weighted) averaging the outputs of multiple neural networks, or class probabilities of multiple trees, or by using Naive Bayesian combination of different rules.

Chan and Stolfo (1995; 1996) investigate various model combination methods. They show that some combination method (for models learned from disjoint partitions of a dataset) can outperform one single model learned from the entire dataset in some domains. While there is some overlap with our work, their investigation is limited in two ways. First, only two datasets are used. Second, it is unclear when a combination method (for models learned from partitions of data) is better than a single model learned from the entire dataset.

Some methods provide guidance as to how to partition the description space. Some use the information gain criterion (Utgoff, 1989), user-provided information (Tcheng, Lambert & Rendell, 1989), or hand-crafted rules (Brodley, 1993) to guide the recursive partitioning process in a tree structure; and others (Ting, 1994; Wettschereck, 1994) employ a confidence measure provided from one particular learned model, during classification, to decide which one of the two different models shall be used for final prediction. The former methods apply different types of learning algorithm for each of the mutually exclusive partitions, and the latter methods derive different types of model independently using the entire dataset. Baxt (1992) describes a situation where the data is pre-sorted manually into two different groups according to some criterion (i.e., high and low risk groups in a medical diagnostic task), and then separate neural networks are trained using each data group. During classification, the network trained using the low risk group is used if its output is below certain threshold, otherwise the other network is used. This method may only be applicable when the information about the sorting criterion is available.

Provost and Hennessy (1996) describe a distributed approach to learning a single ruleset from several rulesets induced from disjoint partitions of a given dataset. They ensure that the ruleset is a superset of the rules induced from the entire dataset. This is achieved by maintaining the invariant-partitioning property during the rule learning process. This property guarantees that each rule which is satisfactory over the entire dataset will be satisfactory over at least one subset. This approach aims at speeding up the process of learning a set of rules that cover the entire dataset. It is not meant to generate multiple different models to enhance performance. Brazdil and Torgo (1990) also work on converting different models into one unique model.

Most of this work assumes that a single dataset is used for multiple model generation and combination. The exceptions are Chan and Stolfo's (1996), Provost & Hennessy's (1996), Breiman (1996c) and Baxt's (1992) investigations. Only the first two studies have the similar working assumption to ours, i.e., multiple batches of data for the same task are available without any prior information about them.

3 Experimental Design

The basis of the experimental design rests on the hypothesis that

> *the relative performance of model combination and data combination is related to the learning curve of a learning algorithm used.*

At the beginning of the learning curve, where the training data size is relatively small, data combination will usually give a big gain in performance; whereas at the near-asymptotic region of the curve, additional data only improves the performance marginally. The near-asymptotic region is grossly defined here to be the region where doubling the training data size gives little performance gain. The effect of doubling the training data sizes (X & Y) in two different regions of a learning curve is illustrated in Figure 2. The reverse is true for model combination. When little data is available, the estimated measure(s) required for successful model combination can be inaccurate; thus, the performance gain would be marginal. Large amounts of data enable more accurate estimation and therefore a better performance gain. Thus, the experiments are designed to unveil the learning curves of learning algorithms and model combination methods used for each dataset.

We choose to combine two classifiers because it is the simplest combination; its study provides an understanding of its base-line behaviour in model combination. In support of our choice, the empirical study conducted by Chan and Stolfo (1995; 1996) indicates that a combination of two classifiers performs better than those using more than two classifiers in the same working assumption. This is also true when this combination is being stacked up to form a tree, i.e., a binary tree is better than higher order trees (Chan & Stolfo; 1995).

We employ a recent model combination method (Ting, 1996) to conduct our investigation. This is based on the characterisation and estimation of predictive

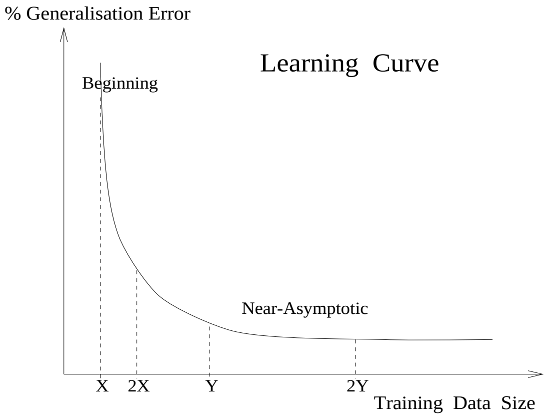

Fig. 2. Performance gain as a result of doubling the training data sizes (X & Y) in two different regions of a learning curve.

accuracy for each prediction of a classifier. Each classifier is trained independently and uses a cross-validation method to perform an estimation of predictive accuracy. During classification, the prediction of a classifier which has the best predictive accuracy is selected among the constituent classifiers as the final prediction. See the Appendix for a more detailed description of the method. An "oracle" combination method is also used for comparison. It always makes the correct prediction from its constituent classifiers, if one exists.

4 Experiments and Results

Two inductive learning algorithms, IB1* and NB* (Ting, 1994; 1997), are used in our experiments. IB1* is a variant of IB1 (Aha, Kibler & Albert, 1991) that incorporates the modified value-difference metric (Cost & Salzberg, 1993) and NB* is an implementation of the Naive Bayes (Cestnik, 1990) algorithm. Both algorithms include a method (Fayyad & Irani, 1993) for discretising continuous-valued attributes in the preprocessing. This preprocessing improved the performance of the two algorithms in most of the continuous-valued attribute domains studied by Ting (1994). We use the nearest neighbour for making prediction in IB1* and the default settings are the same as those used in IB1[3] in all experiments. No parameter settings are required for NB*.

Our studies employ two artificial domains (i.e., waveform and LED24) and four real-world datasets (i.e., euthyroid, nettalk(stress), splice junction and protein coding) obtained from the UCI repository of machine learning databases (Merz & Murphy, 1996). The two noisy artificial domains are introduced by Breiman, Friedman, Olshen and Stone (1984). Each instance of the waveform domain contains twenty-one relevant and nineteen irrelevant continuous-valued attributes. There are three uniformly distributed classes in this domain. Each class consists of a random convex combination of two of the three waveforms

[3] IB1 stores all training instances and uses maximum differences for attributes that have missing values, and computes Euclidean distance between any two instances.

with Gaussian noise added. The LED24 domain has seven boolean attributes indicating whether the light-emitting diodes are on or off, plus seventeen irrelevant binary attributes. Each attribute value is inverted with a probability of 0.1. The task is to classify the input as one of the ten digits.

The euthyroid dataset is one of the sets of Thyroid examples from the Garvan Institute of Medical Research in Sydney described in Quinlan, Compton, Horn and Lazarus (1987). It consists of 3163 case data and diagnoses for one of the many thyroid disorders: euthyroidism. Eighteen binary attributes and seven continuous-valued attributes are used in this dataset. The task is to predict whether a patient suffers euthyroid or not.

The goal of the NETtalk task (Sejnowski & Rosenberg, 1987) is to learn to pronounce English words by studying a dictionary of correct pronunciations. Each letter to be pronounced is presented to the classifier together with the three preceding and three succeeding letters in the word. The goal is to produce phoneme and stress that constitute the pronunciation of the letter. The nettalk(stress) dataset of 5438 instances is for the prediction of stress (five classes), produced from the NETtalk Corpus of the 1000 most common English words.

The splice junction dataset, courtesy of Towell, Shavlik and Noordewier (1990), contains 3177 instances of sixty sequential DNA nucleotide positions and each position can have one of the four base values. The task is to recognize, given a DNA sequence, two types of the splice junction or neither.

The protein coding dataset, introduced by Craven and Shavlik (1993), contains DNA nucleotide sequences and its classification task is to differentiate the coding sequences from the non-coding ones. Each sequence has fifteen nucleotides with four different values each. This dataset has 20,000 sequences.

In what follows, we first perform the experiments using the artificial domains and then the real-world datasets.

4.1 Artificial Domains

To simulate different batches of data, different seeds are used to generate the data in the waveform and LED24 domains. We examine data batches of equal size. The training data size is varied but the testing data size is fixed at 5000 instances. For each training data size, two models are induced from a learning algorithm (either IB1* or NB*) from two batches of data. Model combination uses these two models to produce a final prediction. Data combination concatenates the two batches of data to form a single training dataset and produces a model using the same learning algorithm. For each training data size, it is repeated 10 trials using different seeds in data generation, and the average error rate and its standard error are reported. Figures 3 and 4 shows the results of the experiments (i.e., the learning curves). The horizontal-axis shows the training data size of data combination; the single model induced from a batch of half of this training data size is designated as "1/2 Data Size"[4]. The results of model combination and

[4] There are two such models that perform very similarly. One of their learning curves is eliminated to provide a better readability of the plot.

the oracle (which makes incorrect prediction if and only if both models predict incorrectly) are also shown. Plots (a) and (b) in each figure show the results using IB1* and NB*, respectively.

We summarise the results as follows. When using IB1*, model combination performs significantly better than data combination in both the waveform and LED24 domains in almost all the experimental training data sizes. Two average error rates are regarded to be significantly different if they differ by more than or equal to two standard errors (with $\geq$ 95% confidence). Similar performance is observed for NB* in the waveform domain. Note that in these three cases, the performance difference is small between data combination and the single model induced from half of the training data size. The general trend is that the positive performance gain of model combination over data combination becomes bigger towards the near-asymptotic region of the learning curve. This is the region where the classifier's performance gain as a result of doubling the data size does not gain as much as that at the beginning of the curve. For NB* in the LED24 domain, data combination performs significantly better than model combination when using small training data sizes and then becomes marginally worse as the data size gets larger. In all cases, model combination and data combination are always better than model combination's constituent models. The oracle shows the optimal performance for model combination and it is the best among the four methods.

One interesting phenomenon is that, in the LED24 domain, IB1* seems to reach its (near-)asymptotic performance from data size 800 onwards. But, the approach to combine models learned from half of these data sizes can still significantly improve the algorithm's (near-)asymptotic performance! The performance of the oracle indicates that the two models are significantly different since the performance difference between the oracle and its constituent models is as large as 15%! We will come back to this point in Section 5.

4.2 Real-World Datasets

In this experiment, we employ four real-world datasets; and for each dataset, we simulate two different batches by random subsampling of the training data for data combination into two disjoint subsets of equal size. The size of training data size is varied from 10% to 90% of the entire dataset. Each data size is repeated over 20 trials, except in the protein coding dataset for its huge data size and only 10 trials are used. The testing dataset is from the remaining portion of the entire dataset not used for training.

Figures 5 to 8 show the results for the four datasets. In the euthyroid, nettalk(stress) and splice junction datasets, the performance trends of model combination and data combination generally in accordance to the results observed the artificial domains. Model combination performs worse than data combination at the beginning of the learning curve in all three datasets. At the near-asymptotic region of the curves, model combination performs better using NB* and comparably using IB1* in the euthyroid dataset; model combination performs better using IB1* and comparably using NB* in the other two datasets.

For the protein coding dataset shown in Figure 8(a), the trends of the curves (for IB1*) seem to suggest that the near-asymptotic region of the learning curve does not appear in the figure, i.e., we do not see the complete learning curve (which contains the two regions shown in Figure 2). Nevertheless, model combination is significantly better than data combination when the training data sizes are between 10% to 50% of the whole available data. The trend is reversed when the training data sizes are 80% and 90%. When NB* is used, model combination performs worse than data combination at the beginning of the curve, and they perform comparably at the near-asymptotic region.

We have also investigated the effect of data batches of different sizes, and it has demonstrated similar results as the ones with equal size. An investigation into the effect of overlapping data batches on the performance of model combination is also conducted. Model combination shows progressive performance degradation as the percentage of overlap increases. These results are not shown due to lack of space. Please refer to Ting and Low (1996) for details.

5 Discussion

One phenomenon observed in the experiments is that the relative performance between model combination and data combination is related to the learning behaviour of the classifier. This is consistent with our hypothesis stated in Section 3. If the performance improvement is small when the data is combined, model combination usually performs comparably or better than data combination. This is evident when we observed the usual learning curves in all datasets. Even for IB1* in the protein coding dataset (shown in Figure 8(a)), when the complete learning curve is not observed, this phenomenon still occurs.

We provide some operational guidance on when to use model/data combination in two situations. In the first situation, where data is collected in batches, generate a model using the first batch of data. When the second batch is available, combine the two data batches to generate another model. In the second situation, where a large dataset is given, one can generate a model using this dataset and then another model using only half of the dataset. In each situation, estimate the predictive error rate for both models using some error estimation method (e.g., cross-validation). If these two models demonstrate a small difference in predictive error rate, then a model combination approach should be employed. Otherwise, use the model which derived using all the available data.

Our empirical result seems to be stronger, in terms of the expected performance of model combination, than a theoretical result (Kearns & Seung, 1995) based on the same working assumption. This theoretical work seeks "... the possibility of somehow combining the independent hypotheses in a way that considerably outperforms any single hypothesis" (Kearns & Seung, 1995). The 'single hypothesis' refers to any of the model combination's constituent models. Our result indicates that model combination can significantly outperform not only its constituent models but the model learned from aggregating the available data; in spite of the the fact that this result only shows the base-line behaviour of model combination, i.e., combining two models.

It might be assumed that the models learned in the near-asymptotic region would be very similar as shown by their indistinguishable performance. However, our experimental results suggest that the two models learned from separate data batches in this region are substantially different even though they demonstrate the same performance. This is evident in most of the experimental datasets, where the performance of the oracle is significantly better than those of its constituent models and data combination in the near-asymptotic region. This indicates that the assumption is incorrect, at least in the near-asymptotic region.

What surprises us is that *the near-asymptotic performance of a single model can be significantly improved by combining multiple models of the same learning algorithm.* This can only happen if the constituent models are substantially different and there is still some regularity in the models to be exploited by any combination methods. Figure 4(a) shows an example of performance improvement as a result of model combination's exploitation in the near-asymptotic region. This empirical evidence of further significant improvement on the near-asymptotic performance of a learning algorithm using multiple models is new to us, to the best of our knowledge. Previous work (e.g., Hansen & Salamon, 1990; Perrone & Cooper, 1993; Oliver & Hand, 1995; Chan & Stolfo, 1995; Breiman, 1996a,1996b; Freund & Schapire, 1996; Ali & Pazzani, 1996) only show the possibility of improving the performance using multiple models in some datasets, without considering the (near-)asymptotic performance; despite the theoretical result of boosting (Schapire, 1990) shows that this is possible.

Though our investigation is limited to one type of model combination method, we believe that the results are applicable to other reasonable combination methods (e.g., Ali & Pazzani, 1996) judging from the performance of the "oracle" combination method. In all datasets, the oracle performs significantly better than data combination, sometimes with huge margins. This shows there is plenty of room for any reasonable combination method to gain advantage. We also believe that the results would also hold when other types of learning algorithm are used (e.g., decision trees and neural networks). Indeed, in a subsequent work, Ting and Witten (1997) confirm our conjecture here by showing that stacked generalisation (i.e., a different model combination method (Wolpert, 1992)) also behaves similarly in the same scenario by using either NB, IB1 or C4.5 (a decision tree learning algorithm (Quinlan, 1993)).

6 Conclusions

A comparison of the base-line behaviour of model combination and data combination using two randomly drawn disjoint data batches reveals that model combination compares favourably when the performance gain due to data combination is small, or the performances of the models induced from the data batches are at the near-asymptotic region of the learning curve. The empirical evidence presented in this paper and subsequently by Ting & Witten (1997) show that this result is not sensitive to the particular learning algorithm or model combination method employed.

The practical implication of our results is that one should consider using model combination rather than the common method of data combination, especially when multiple batches of data for the same task are readily available.

Another interesting result is that we empirically show that it is possible to significantly improve the near-asymptotic performance of a single model by combining multiple models of the same algorithm if the models are substantially different, and there is some regularity in the models to be exploited by the combination method used.

Acknowledgment

Ross Quinlan and Robert Schapire help to clarify the issue on (near-)asymptotic performance. Z. Zheng, B. Pfahringer, M. Cameron-Jones, I. Witten and members of the machine learning group and the anonymous reviewers provide helpful comments during the early draft of this paper. Ronny Meir provides some pointer to the related work in NIPS. This paper is prepared with supports from a grant from the Chinese University of Hong Kong and Department of Computer Science, University of Waikato.

Appendix – The Method of Model Combination

We use the composite learner framework (Ting, 1996) as the method for model combination in our experiments. Ting uses the term *the characterisation of predictive accuracy* to mean the use of a measure in an induced model as an indicator for its predictive accuracy. The posterior probability and the measure of typicality (defined as inter-concept distance divided by intra-concept distance) are employed as the characterisation of predictive accuracy for NB* and IB1*, respectively.

During training, the algorithm (either NB* or IB1*) performs a k-fold cross-validation to estimate predictive accuracy from the characterisations for each batch of data independently. In a k-fold cross-validation, a dataset is partitioned into k equal-size subsets and perform training using all subsets except the ith subset k times. At each fold, the ith subset is used as the testing set. Thus, each instance will be tested only once; and a training set of size n will have the testing results of size n at the end of the cross-validation. We set k to three in our experiments.

For each predicted class, the individual test results from the k-fold cross-validation are then sorted according to the values of the characterisation (i.e., posterior probability, or typicality on the x-axis), shown in the left plot of Figure A. The aim is to produce a binned graph that relates the average value of the characterisation to its binned predictive accuracy for each class, shown in the right plot of Figure A. The transformation process, from the left plot to the right plot, goes as follows. First, use a bin/window which contains a fixed number of instances in the left plot. This bin is transformed to a point in the right plot by averaging the values of the characterisation for all instances in the bin on the x-axis and converting the number of correct/incorrect classifications into accuracy on the y-axis. This process is repeated in the style of a "moving window", i.e., the next bin is obtained by dropping the leftmost instance and adding an instance adjacent to the rightmost instance of the current bin. At the end of the training process, a model induced from all the n training instances and the binned graphs for all classes are stored for future classification. The above procedure is conducted for each model derived using one batch of data. There are two models in the model combination used in our experiments.

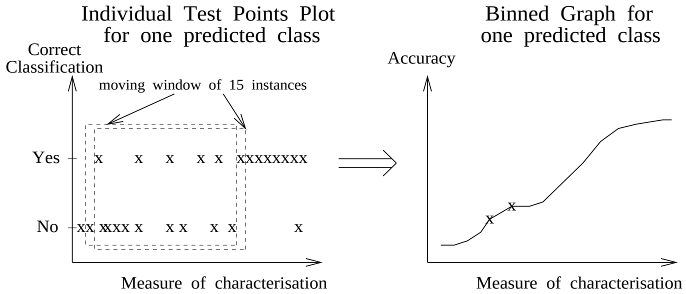

Fig. A. Transforming individual cross-validation test points to a binned graph for one predicted class.

To classify an instance X, it is inputed into both models and the model which has the higher estimated predictive accuracy for its output is chosen to make the final prediction. The predictive accuracy, $\mathrm{PA}(C, H|X)$, is obtained by referring to the predicted class' (C) binned graph with the corresponding value of the characterisation (H). Formally, the selection process can be defined as follows.

$$
\begin{aligned}
C_f &= C_{M1} \quad \text{if } \mathrm{PA}(C_{M1}, H_{M1}|X) > \mathrm{PA}(C_{M2}, H_{M2}|X),\\
&= C_{M2} \quad \text{if } \mathrm{PA}(C_{M1}, H_{M1}|X) < \mathrm{PA}(C_{M2}, H_{M2}|X),\\
&\quad\quad\quad\ \text{else random select.}
\end{aligned}
$$

where C_f : final prediction;
C_M & H_M : Model M's prediction & characterisation;
$\mathrm{PA}(C, H|X)$: predictive accuracy of C and H given instance X.

References

Aha, D.W., D. Kibler & M.K. Albert (1991), Instance-Based Learning Algorithms, *Machine Learning, 6*, pp. 37-66.

Ali, K.M. & M.J. Pazzani (1996), Error Reduction through Learning Multiple Descriptions, *Machine Learning*, Vol. 24, No. 3, pp. 173-206.

Baxt, W.G. (1992), Improving the Accuracy of an Artificial Neural Network using Multiple Differently Trained Networks, *Neural Computation*, Vol. 4, No. 5, pp. 772-780, The MIT Press.

Brazdil,P. & Torgo,L. (1990), Knowledge Acquisition via Knowledge Integration. In *Current Trends in Knowledge Acquisition*, Wielinga, B. et al.(eds.).

Breiman, L. (1996a), Bagging Predictors, *Machine Learning*, Vol. 24, No. 2, pp. 123-140.

Breiman, L. (1996b), Bias, Variance, and Arcing Classifiers, *Technical Report 460*, Department of Statistics, University of California, Berkeley, CA.

Breiman, L. (1996c), Pasting Bites Together for Prediction in Large Data Sets and On-Line, [ftp.stat.berkeley.edu/users/pub/breiman/pasting.ps].

Breiman, L., J.H. Friedman, R.A. Olshen & C.J. Stone (1984), *Classification And Regression Trees*, Belmont, CA: Wadsworth.

Brodley, C.E. (1993), Addressing the Selective Superiority Problem: Automatic Algorithm/Model Class Selection, in *Proceedings of the Tenth International Conference on Machine Learning*, pp. 17-24.

Buntine, W. (1991), Classifiers: A Theoretical and Empirical Study, in *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pp. 638-644, Morgan-Kaufmann.

Cestnik, B. (1990), Estimating Probabilities: A Crucial Task in Machine Learning, in *Proceedings of the European Conference on Artificial Intelligence*, pp. 147-149.

Chan, P.K. & S.J. Stolfo (1995), A Comparative Evaluation of Voting and Meta-learning on Partitioned Data, in *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 90-98, Morgan Kaufmann.

Chan, P.K. & S.J. Stolfo (1996), On the Accuracy of Meta-learning for Scalable Data Mining, in *Journal of Intelligent System*, to appear.

Cost, S & S. Salzberg (1993), A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features, *Machine Learning, 10*, pp. 57-78.

Craven, M.W. & J.W. Shavlik (1993), Learning to Represent Codons: A Challenge Problem for Constructive Induction, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pp. 1319-1324.

Fayyad, U.M. & K.B. Irani (1993), Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, in *Proceedings of 13th International Joint Conference on Artificial Intelligence*, pp. 1022-1027.

Freund, Y. & R.E. Schapire (1996), Experiments with a New Boosting Algorithm, in *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 148-156, Morgan Kaufmann.

Hansen, L.K. & P. Salamon (1990), Neural Network Ensembles, in *IEEE Transactions of Pattern Analysis and Machine Intelligence, 12*, pp. 993-1001.

Kearns, M. & H.S. Seung (1995), Learning from a Population of Hypotheses, *Machine Learning, 18*, pp. 255-276, Kluwer Academic Publishers.

Kononenko, I. & M. Kovačič (1992), Learning as Optimization: Stochastic Generation of Multiple Knowledge, in *Proceedings of the Ninth International Conference on Machine Learning*, pp. 257-262, Morgan Kaufmann.

Krogh, A. & J. Vedelsby (1995), Neural Network Ensembles, Cross Validation, and Active Learning, in *Advances in Neural Information Processing Systems 7*, G. Tesauro, D.S. Touretsky & T.K. Leen (Editors), pp. 231-238.

Kwok, S. & C. Carter (1990), Multiple Decision Trees, *Uncertainty in Artificial Intelligence 4*, R. Shachter, T. Levitt, L. Kanal and J. Lemmer (Editors), pp. 327-335, North-Holland.

Merz, C.J. & Murphy, P.M. (1996), *UCI Repository of machine learning databases* [http:// www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

Oliver, J.J. & D.J. Hand (1995), On Pruning and Averaging Decision Trees, in *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 430-437. Morgan Kaufmann.

Perrone, M.P. & L.N. Cooper (1993), When Networks Disagree: Ensemble Methods for Hybrid Neural Networks, in *Artificial Neural Networks for Speech and Vision*, R.J. Mammone (Editor), Chapman-Hall.

Provost, F.J. & D.N. Hennessy (1996), Scaling Up: Distributed Machine Learning with Cooperation, in Proceedings of the Thirteen National Conference on Artificial Intelligence, pp. 74-79, Menlo Park, CA: AAAI Press.

Quinlan, J.R. (1993), *C4.5: Program for machine learning*, Morgan Kaufmann.

Quinlan, J.R. (1996), Boosting, Bagging, and C4.5, in *Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 725-730, AAAI Press.

Quinlan, J.R., P.J. Compton, K.A. Horn & L. Lazarus (1987), Inductive Knowledge Acquisition: A Case Study, in *Applications of Expert Systems*, J.R. Quinlan (Editor). Turing Institute Press with Addison Wesley.

Schapire, R.E. (1990), The Strength of Weak Learnability, *Machine Learning, 5*, pp. 197-227, Kluwer Academic Publishers.

Sejnowski, T.J. & C.R. Rosenberg (1987), Parallel networks that learn to pronounce English text, *Complex Systems*, 1, pp. 145-168.

Tcheng, D., B. Lambert, C-Y. Lu & L. Rendell (1989), Building Robust Learning Systems by Combining Induction and Optimization, in *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pp. 806-812.

Ting, K.M. (1994), Discretization of Continuous-Valued Attributes and Instance-Based Learning, *TR 491*, Basser Department of Computer Science, University of Sydney.

Ting, K.M. (1996), The Characterisation of Predictive Accuracy and Decision Combination, in *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 498-506, Morgan Kaufmann.

Ting, K.M. (1997), Discretisation in Lazy Learning Algorithms, to appear in the special issue on Lazy Learning in *Artificial Intelligence Review Journal*.

Ting, K.M. & B.T. Low (1996), Theory Combination: an alternative to Data Combination, *Working Paper 96/19*, Department of Computer Science, University of Waikato. [http://www.cs.waikato.ac.nz/cs/Staff/kaiming.html].

Ting, K.M. & I. H. Witten (1997), Stacked Generalization: when does it work?, *Working Paper 97/1*, Dept of Computer Science, University of Waikato.

Towell, G., J. Shavlik & M. Noordewier (1990), Refinement of Approximate Domain Theories by Knowledge-Based Artificial Neural Networks, in *Proceedings of the Eighth National Conference on Artificial Intelligence.*

Utgoff, P.E. (1989), Perceptron Trees: A case study in hybrid concept representations, *Connection Science, 1*, pp. 337-391.

Wettschereck, D. (1994), A Hybrid Nearest-Neighbor and Nearest-Hyperrectangle Algorithm, in *Proceedings of the Seventh European Conference on Machine Learning, LNAI-784*, pp. 323-335, Springer Verlag.

Wolpert, D.H. (1992), Stacked Generalization, *Neural Networks*, Vol. 5, pp. 241-259, Pergamon Press.

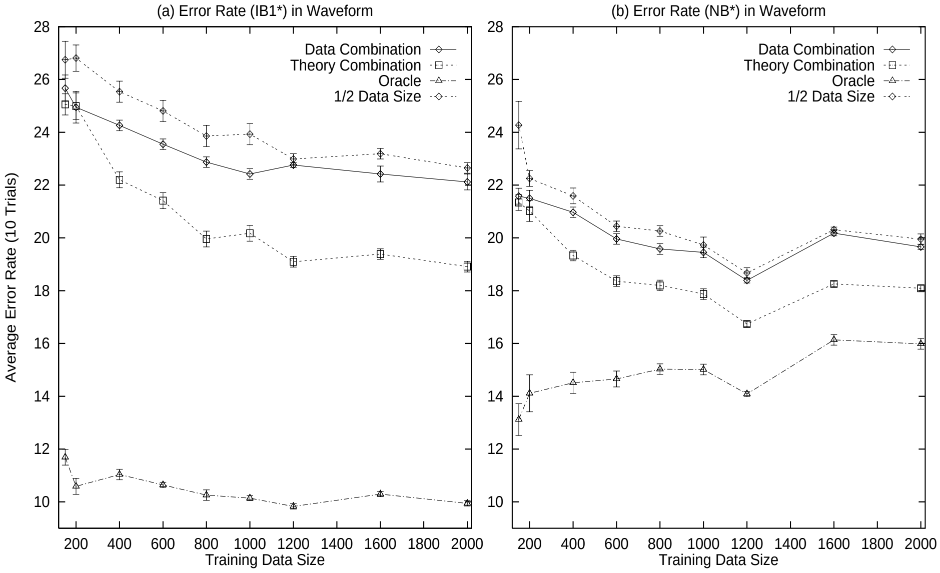

Fig. 3. Learning Curves in the Waveform domain.

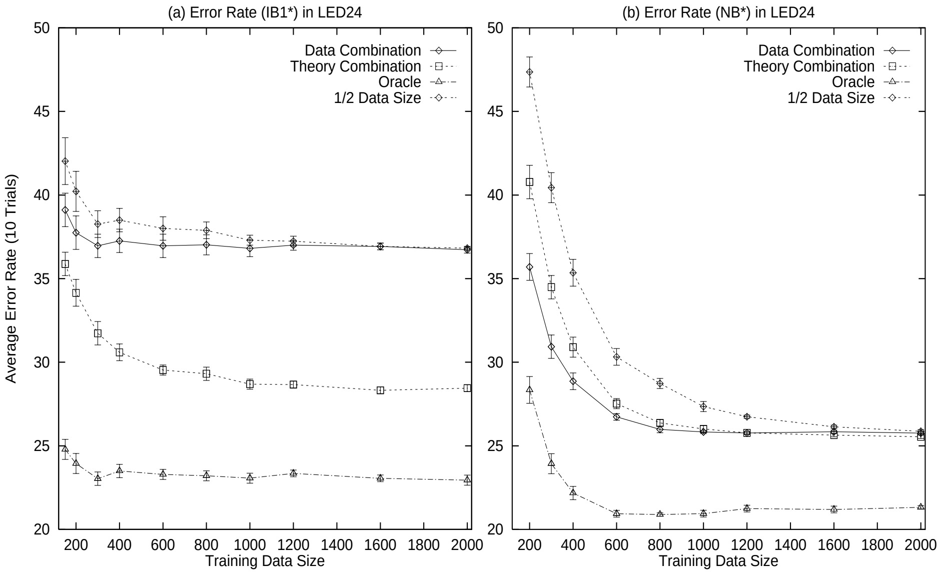

Fig. 4. Learning Curves in the LED-24 domain.

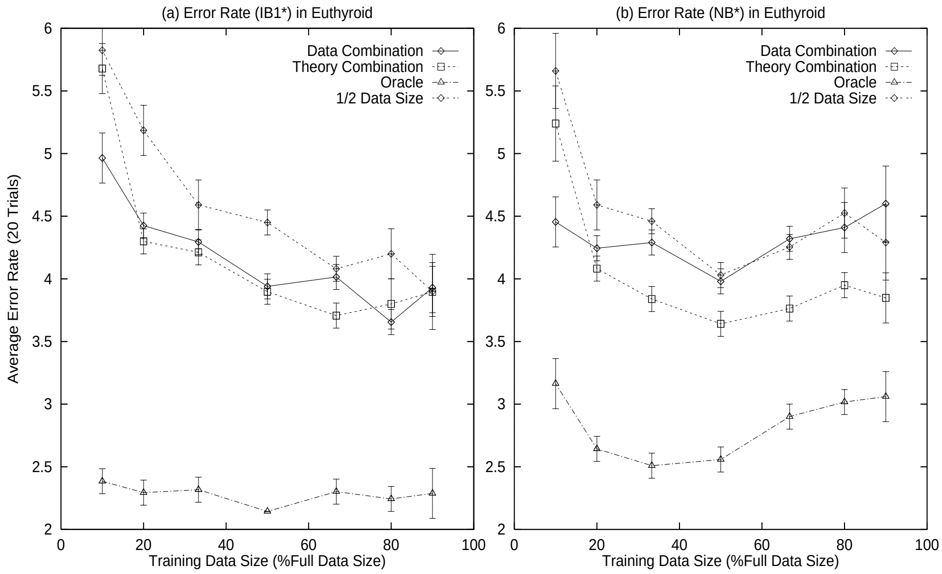

Fig. 5. Learning Curves in the Euthyroid dataset.

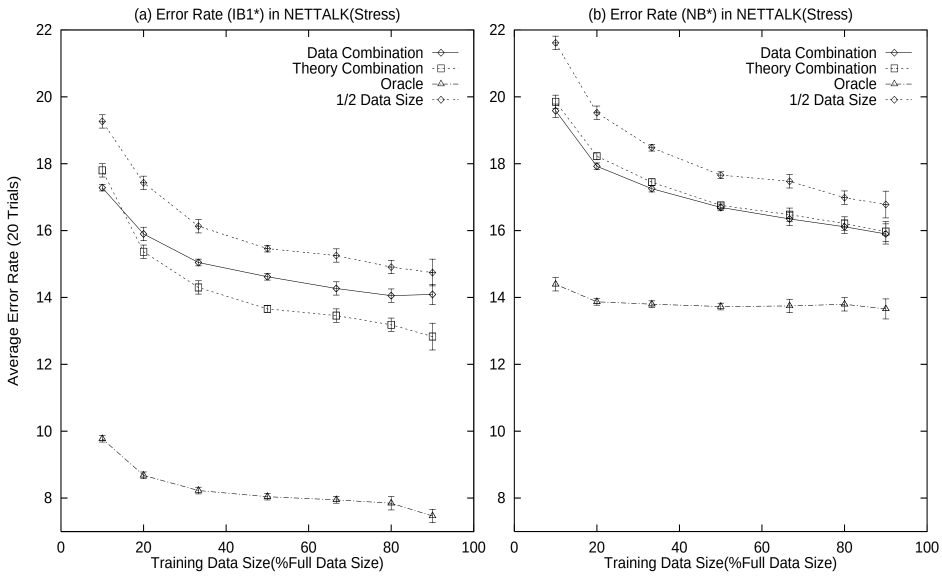

Fig. 6. Learning Curves in the Nettalk(Stress) dataset.

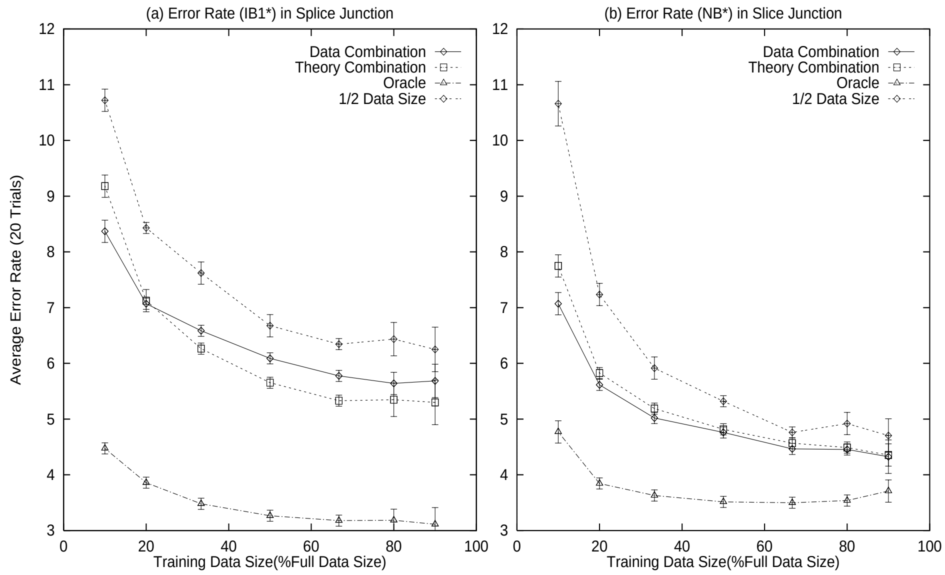

Fig. 7. Learning Curves in the splice junction dataset.

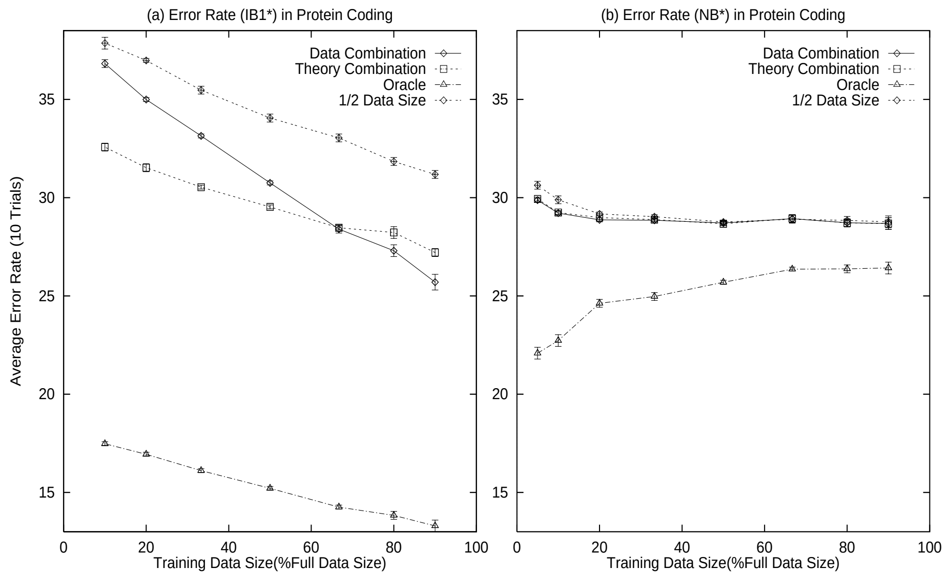

Fig. 8. Learning Curves in the protein coding dataset.