Theory Combination: an alternative to Data Combination

Kai Ming Ting KAIMING@CS.WAIKATO.AC.NZ
University of Waikato, Hamilton, New Zealand

Boon Toh Low BTLOW@SE.CUHK.HK
Chinese University of Hong Kong, Shatin, Hong Kong

Abstract

The approach of combining theories learned from multiple batches of data provide an alternative to the common practice of learning one theory from all the available data (i.e., the data combination approach). This paper empirically examines the base-line behaviour of the theory combination approach in classification tasks. We find that theory 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 theory 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 theory, in some classification task, can be significantly improved by combining multiple theories (of the same algorithm) if the constituent theories are substantially different and there is some regularity in the theories to be exploited by the combination method used. Comparisons with known theoretical results are also provided.

Keywords: theory 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 "*theory combination*". Figure 1 shows these two types of combination at the data and theory 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, 1996a,1996b; Freund & Schapire, 1996; Perrone & Cooper, 1993; Krogh & Vedelsby, 1995), investigation into combining theories from a single learning algorithm induced using *completely disjoint sets of data* has been limited. Most work shows that combining multiple models induced from

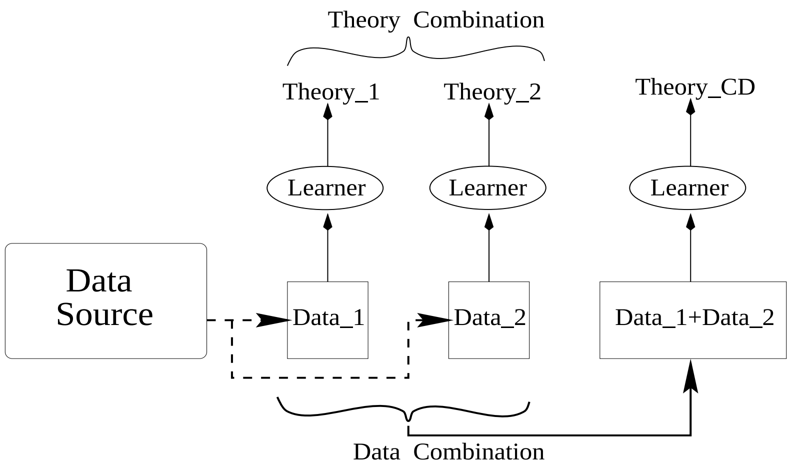

Figure 1: Combination at different levels – data or theory.

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 theory combination performs better than data combination. Thus, our focus here is not on different types of theory combination method. Specifically, we address the question: is theory combination a viable option as compared to data combination in classification tasks? If the answer is yes, when should one use it?

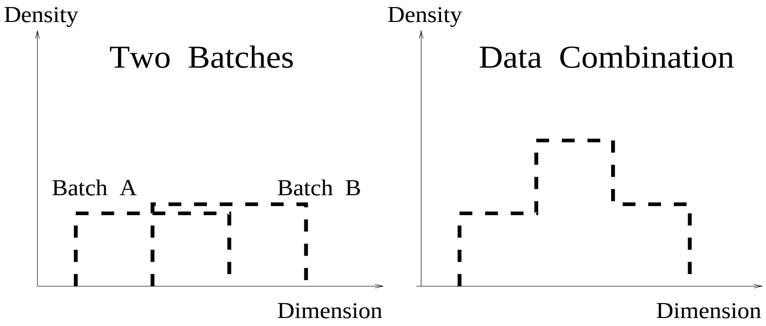

Figure 2: Different views of data in separate batches and data combination.

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. Theory combination allows cooperation between these specialists. Data combination destroys the data variation, and forms a global representation. Figure 2 depicts the intuition using the data density in an one-dimensional space domain. Seeing the data in the right diagram, a learning algorithm would usually produce a strong theory centred on the high data density region, and the low density regions may be neglected. Theories induced from separate batches of data are more localised because of their more limited view of the data. In combining linear regression estimators, some (Meir, 1994; Sollich & Krogh, 1996) have attributed the improved performance of theory combination, under the same multiple-data-batches scenario, to variance reduction for the resultant classifier (see Section 6).

Some research on multiple model combination focus on varying the *induction bias* of the learning algorithm(s) to generate theories 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 theories 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 three discussion sections of different issues and finally the 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 k-fold partitioning to generate k models by training on all but the ith partition k times. In another approach, Fayyad, Weir and Djorgovski (1993) use a covering algorithm to combine the rulesets induced from several sets of random subsamples. The multiple models are usually produced from a single learning algorithm, though there is no such restriction in this formalism.

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 tests with information gains close to the maximum, 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 theory combination methods. They show that some combination method (for theories learned from partitions of a dataset) can outperform one single theory 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 theories learned from partitions of data) is better than a single theory learned from the joint dataset.

Jacobs, Jordan, Nowlan and Hinton (1991) use a gating network and a stochastic one-out-of-n selector to decide which of the n "expert" networks should be used for an instance. During training, the gating network allocates a new instance to one or a few experts, and if the output is incorrect the weight changes are localised to these experts and the gating network. For each input, the gating network produces n outputs (i.e., p_j for $j = 1..n$) to the selector which acts like a multiple-input, single output stochastic switch, where p_j is the probability that the switch will select the output from expert j.

Some methods provide guidance as to how to partition the description space. Some use 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 theory, during classification, to decide which one of the two different theories 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 train different types of theory 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 instead. This method may only be applicable when the information about the sorting criterion is available.

Stacked generalisation (Wolpert, 1992) is a general method of combining multiple models learned from the entire dataset. The models can either be induced from the same or different learning algorithms (Merz, 1995; Ho, Hull & Srihari, 1994). The problem of combination is seen as another learning problem and it uses a learning algorithm at a higher level to achieve the aim.

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 that 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.

Most of this work assumes that a single dataset is used for multiple model generation and combination. The exceptions are Chan & Stolfo's (1996), Provost & Hennessy's (1996) and Baxt's (1992) investigations. Only the first two studies have similar working assumption to ours, i.e., multiple batches of data 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 theory combination and data combination is related to the learning behaviour (i.e., 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 usually will 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 3. The reverse is true for theory combination. When little data is available, the estimated measure(s) required for successful theory 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 theory combination methods used for each dataset.

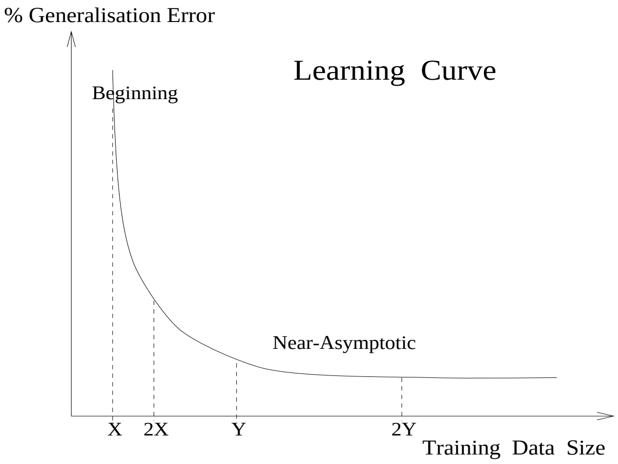

Figure 3: Performance gain as a result of doubling the training data sizes (X & Y) in two different regions of a learning curve.

We choose to combine two classifiers because it is the simplest combination; its study provides an understanding of its base-line behaviour in theory combination. In support of our choice, the empirical study conducted by Chan and Stolfo (1995; 1996) indicates that 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 theory combination method (Ting, 1996b) to conduct our investigation. This is based on the characterisation and estimation of predictive 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; 1996a), 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 as used in IB1[1] in all experiments. No parameter settings are required for NB*.

Our studies employ two artificial domains and four real-world datasets obtained from the UCI repository of machine learning databases (Merz & Murphy, 1996). The two noisy artificial domains are the waveform and LED24 domains introduced by Breiman, Friedman, Olshen and Stone (1984). Each instance of the waveform domains 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 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 four real-world datasets are the euthyroid, nettalk(stress), splice junction and protein coding. The selection criteria are that the datasets must have large number of instances and each class must be supported by large enough instances. A brief description of each of these datasets is as follows.

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. In this task, 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[2]. The task is to recognize, given a DNA sequence, two types of the splice junction or neither.

1. IB1 stores all training instances and uses maximum differences for attributes that have missing values, and computes Euclidean distance between any two instances.
2. The original dataset has 3190 sequences where a small number of them contains some combination values (i.e., values combined from four base values). These sequences are eliminated in our experiments.

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. The protein coding and splice junction datasets are the only two datasets used in Chan & Stolfo's (1995; 1996) investigation.

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 first 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 theories are induced from a learning algorithm (either IB1* or NB*) from two batches of data. Theory combination uses these two theories to produce a final prediction. Data combination concatenates the two batches of data and produces a theory 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 4 and 5 shows the results of the experiments (i.e., the learning curves). The horizontal-axis shows the training data size of data combination; the single theory induced from a batch of half of this training data size is designated as "1/2 Data Size"[3]. The results of theory combination and the oracle (which makes incorrect prediction if and only if both theories 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*, theory 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 theory induced from half of the training data size. The general trend is that the positive performance gain of theory combination over data combination becomes bigger towards the near-asymptotic region of the learning curve. This is where the classifier's performance gain as a result of doubling the data size does not gain as much as at the beginning of the curve. For NB* in the LED24 domain, data combination performs significantly better than theory combination when using small training data sizes and then becomes marginally worse as the data size gets larger. An explanation to this effect is provided in Section 5. Theory combination and data combination are always better than theory combination's constituent theories[4]. The oracle shows the optimal performance for theory combination and it is the best among the four methods.

3. Since the performances of both theories are very similar, one of the curves is eliminated to provide a better readability of the plot.
4. Note that in some cases, the error rate on curve "1/2 Data Size" at training data size D is different from the error rate on curve "Data Combination" at data size $D/2$. This is due to data fluctuation as a result of data generated with different seeds.

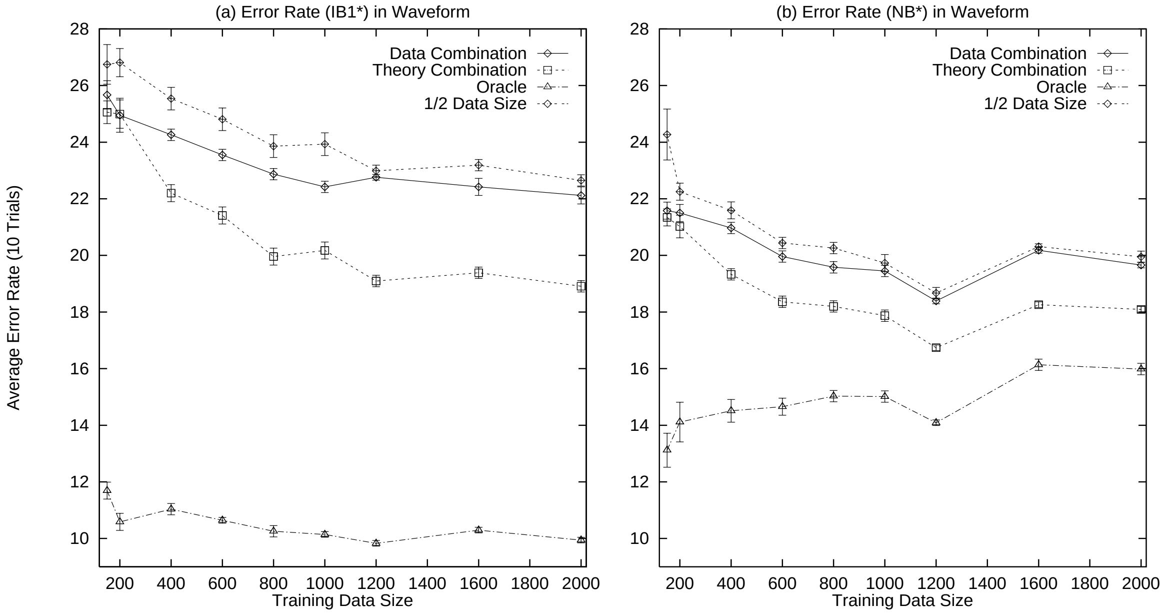

Figure 4: Learning Curves in the Waveform domain (1:1 ratio for the two batches).

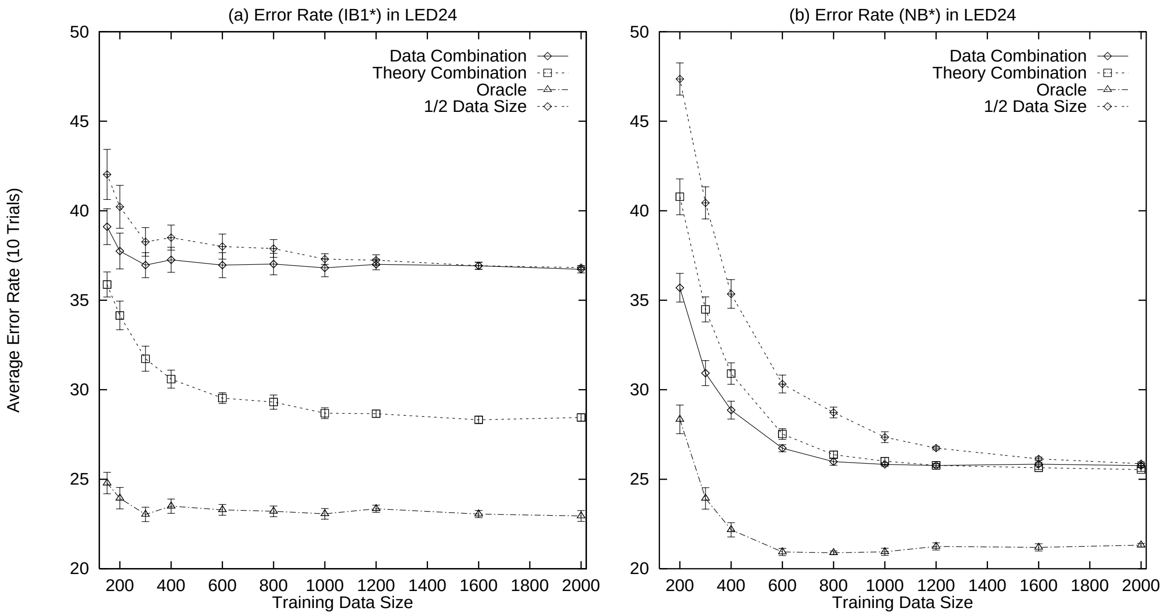

Figure 5: Learning Curves in the LED-24 domain (1:1 ratio for the two batches).

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

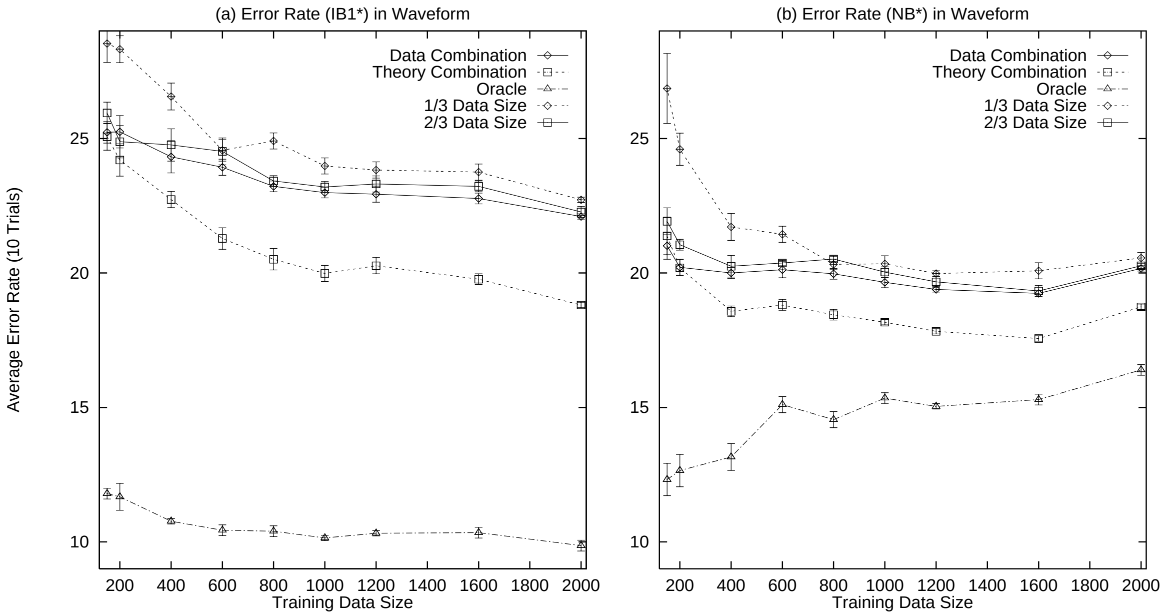

Figure 6: Learning Curves in the Waveform domain (1:2 ratio for the two batches).

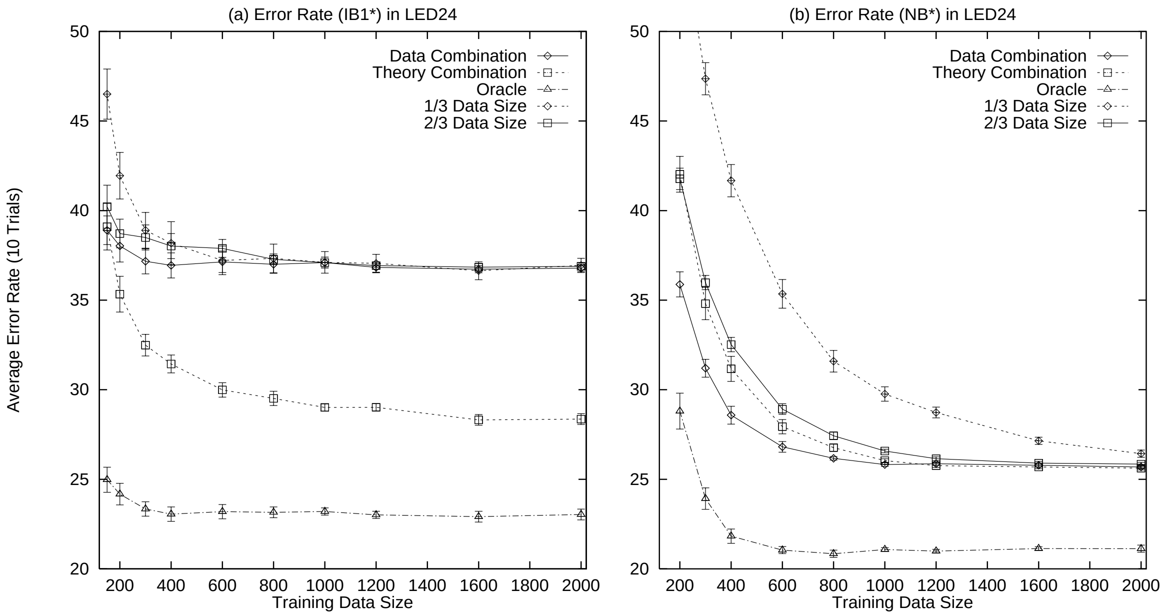

Figure 7: Learning Curves in the LED-24 domain (1:2 ratio for the two batches).

back to this point in Section 5.

To investigate the effect of data batches of different sizes, we experiment with two batches which have one third and two third of training data size for data combination. This experiment demonstrates similar results as the previous one and the results are shown in Figures 6 and 7.

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 8 to 11 show the results for the four datasets. In the euthyroid, nettalk(stress) and splice junction datasets, the performance trends of theory combination and data combination generally in accordance to what we have observed previously. Theory 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, theory combination performs better using NB* and comparably using IB1* in the euthyroid dataset; theory combination performs better using IB1* and comparably using NB* in the nettalk(stress) and splice junction datasets.

For the protein coding dataset shown in Figure 11(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 3). This suggests that improvements are possible for all methods when more data is available. Nevertheless, theory 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 reverse when the training data sizes are 80% and 90%. When NB* is used, theory combination performs worse than data combination at the beginning of the curve, and they perform comparably at the near-asymptotic region.

In the euthyroid dataset, note that the unstable performance at the near-asymptotic region of the curve in Figure 8(a) could be due to skew class distribution (the default accuracy is 90.7%) and the small number of testing data. Also, NB* used in the data combination shows non-monotonic learning curve in Figure 8(b). This could be due to random data fluctuation since the performance difference between the theories using different training sizes are not significant.

Notice that the performance difference between the oracle and data combination is smaller when NB* is used with comparison to that when IB1* is used in almost all the datasets, including the artificial ones. This signifys that theory combination has less room for improvement using NB*. This phenomenon could be due to the nature of the algorithm, i.e., IB1* is more sensitive to local variability than NB* because the latter summarises the training data into a few probability parameters whereas the former employs all the instances. To use Breiman's (1996b) terminology, NB* is a more *stable* classifier than IB1*. Our finding here agrees with that found by Breiman (1996b). It is more likely to improve the performance of an unstable classifier by combining multiple theories induced from it.

4.3 The Effect of Overlapping Data Batches

In this section, we examine the effect of overlapping data batches to theory combination. For each real-world dataset, we use one third of it for testing purposes, and then randomly sample the remaining data into two batches with varying degrees of overlap, range from

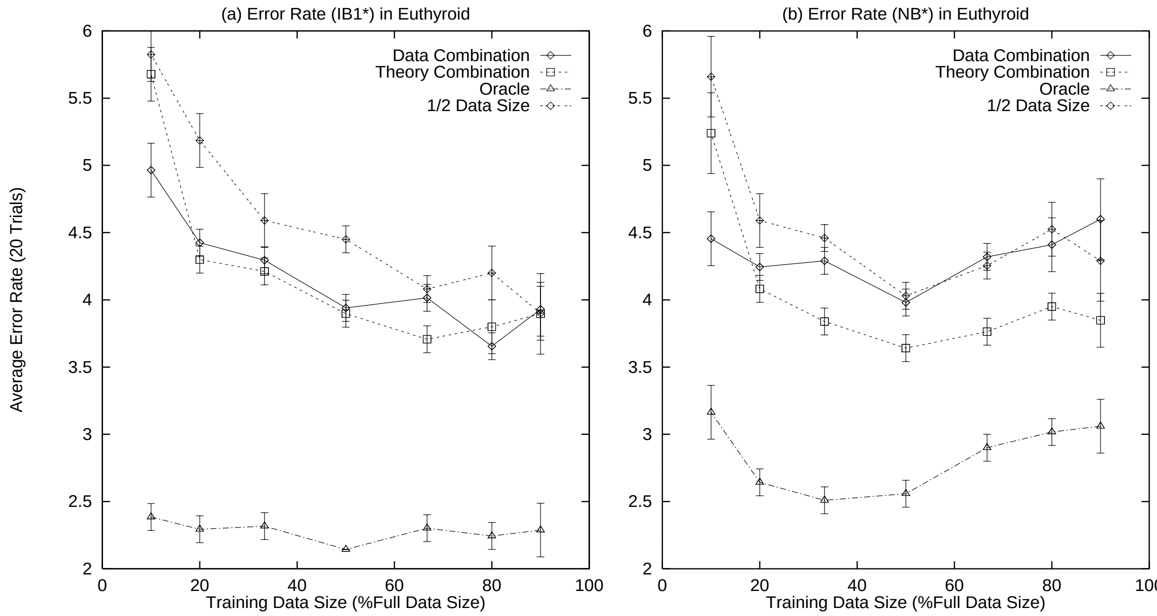

Figure 8: Learning Curves in the Euthyroid dataset.

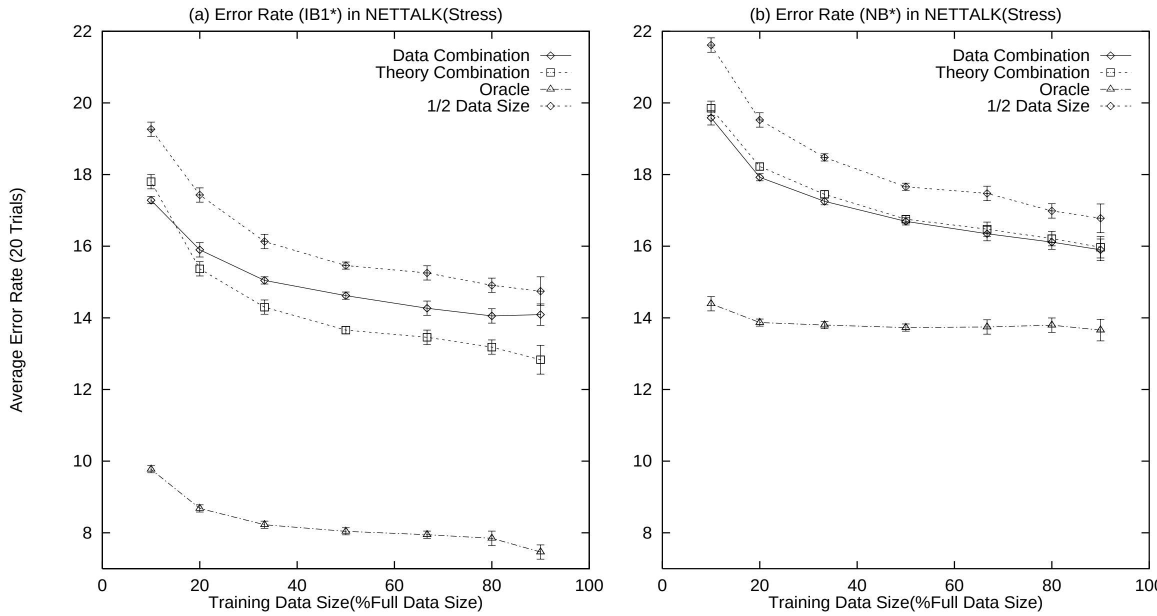

Figure 9: Learning Curves in the Nettalk(Stress) dataset.

0% to 50%. The size of each training batch is kept constant, i.e., one third of the entire dataset, for all trials. Figure 12 shows the performance of theory combination and oracle in four datasets. In almost all cases, theory combination and oracle show progressive performance degradation as the percentage of overlap increases.

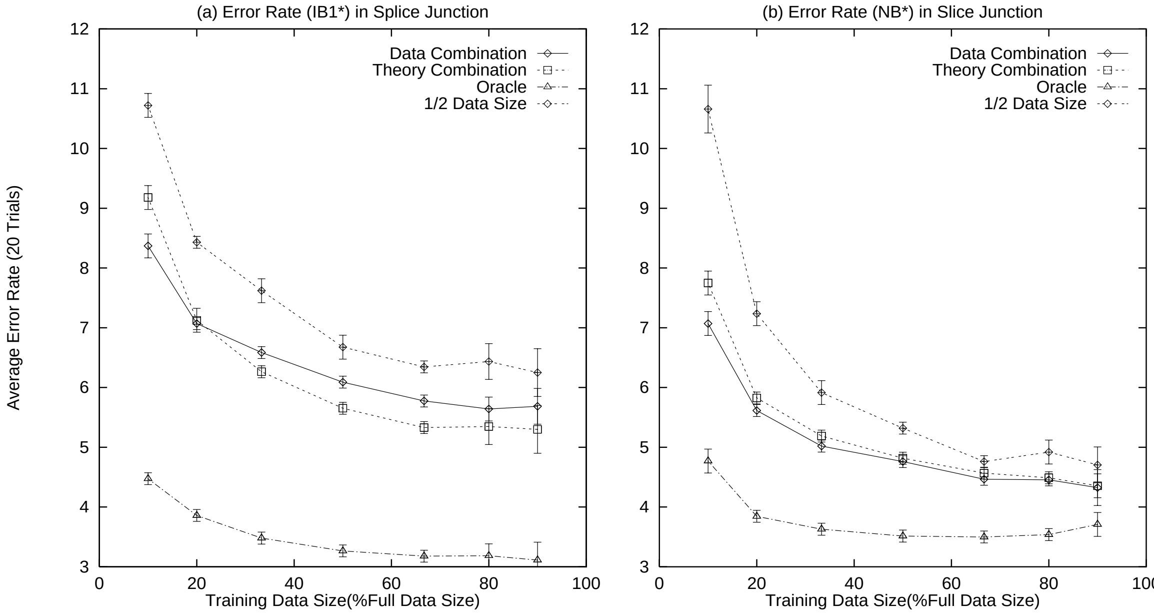

Figure 10: Learning Curves in the splice junction dataset.

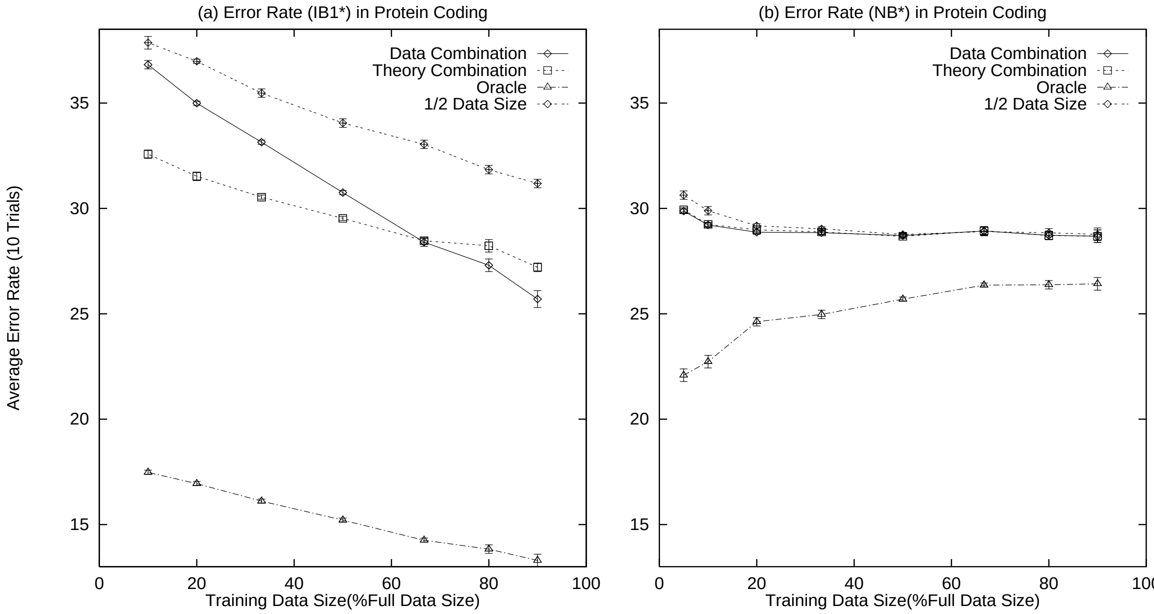

Figure 11: Learning Curves in the protein coding dataset.

.

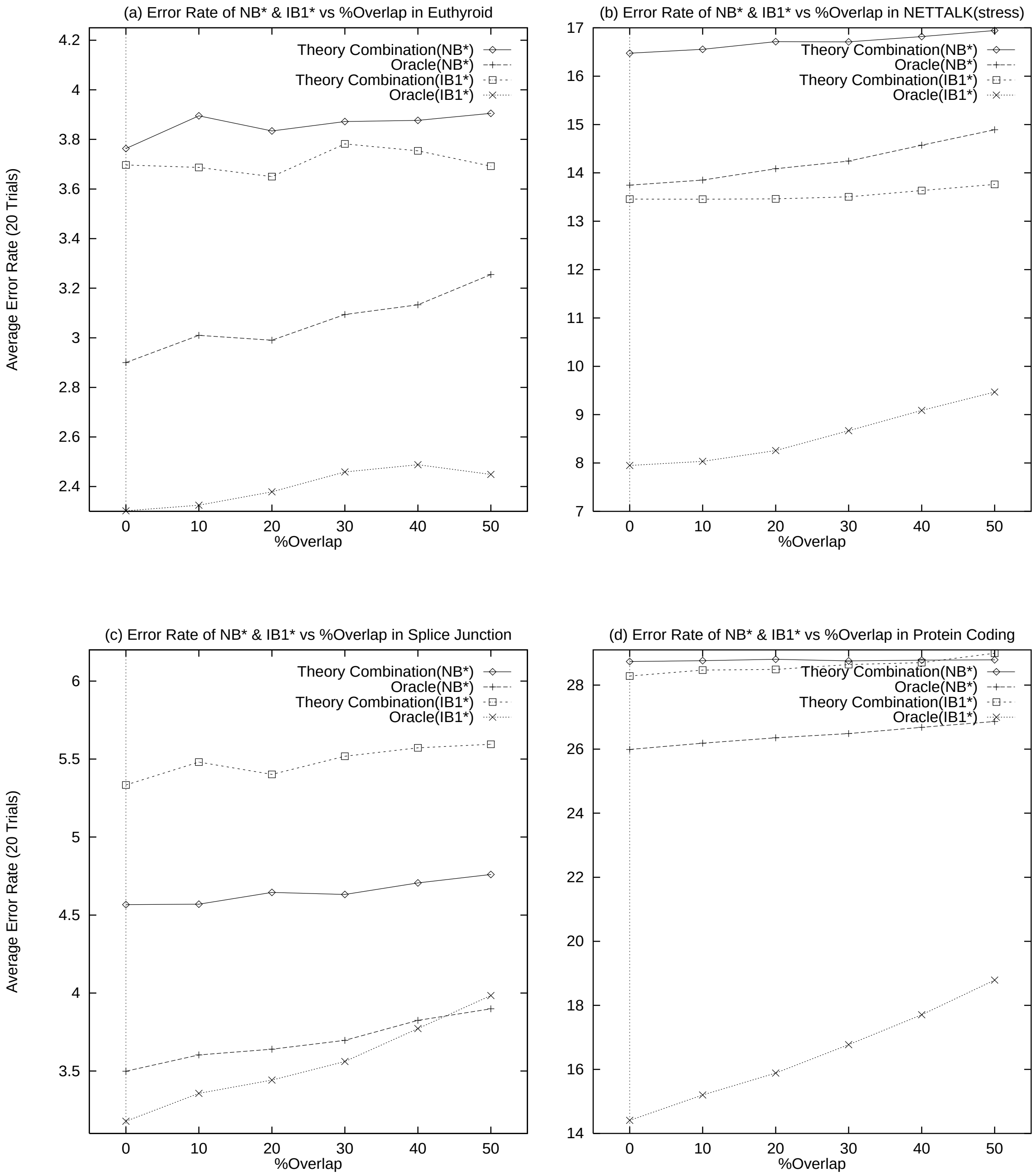

Figure 12: The effect of overlapping training data batches on theory combination in four real-world datasets. The results are averaged over 20 trials except in the protein coding dataset, where only 10 trials are conducted.

5. Criteria for Performance Improvement

The criteria for performance improvement using theory combination (with respect to data combination) are:

- relatively high percentage of only either one of the constituent theories making the correct classifications; or equivalent to high uncorrelated errors,
- each of these theories must demonstrate a high degree of prediction regularity, i.e., the uncorrelated correct classifications are not a result of random guesses. This randomness might be due to noise in the data or/and the stochastic nature of the classifier,
- accurate characterisation and estimation of predictive accuracy when the combination framework used here is employed. In general, this depends on how well the theory combination method employed takes advantage of the first two criteria.

The oracle shows the optimal performance for any methods of theory combination. However, it is unlikely that any of these methods can match its performance. There are two main reasons. First, this is because most theory combination methods require to perform an estimation of some measure or the like (e.g., using a learning algorithm in stacked generalisation) that is required for theory combination. This estimation is bound to have errors, no matter how accurate it is. Second, prediction irregularity, as a result of noise in the data or/and stochastic nature of the learned theory, that cannot be utilized by any combination methods[5]. An example is shown in Figure 5(a) for the LED24 domain. Ting and Cameron-Jones' (1994) empirical result shows that by storing only one instance per class for IB1*, which is the perfect bias in this domain, achieves the best result (i.e., 25% error rate). A similar result is obtained by computing the Bayes rules (Breiman et al, 1984). Thus, the performance of the oracle below the 25% error rate mark is due to random guesses for any classifiers and no other combination method can do better. For the IB1* settings we are using here, the contribution of random guesses could be much higher. NB* also has the right bias in this domain because it is equivalent to an one instance per class instance-based learner (Ting and Cameron-Jones, 1994). It approaches the best result in the near-asymptotic region for all methods shown in Figure 5(b). This explains why theory combination can not outperform data combination in this region.

Note that the behaviour of the oracle is different from the usual learning curve in the waveform and protein coding datasets when NB* is employed. This indicates that some learning algorithm can specialise in different regions of the description space when the sizes of the data batches are relatively small. This can occur because the chances that the data in different batches fall in the same region of the description space are much less when the data size is small. This appears to suggest that it might be better to use less data, for some learning algorithm, when theory combination is intended. However, the combination method employed does not seem to be able to take advantage of this situation. This is

5. The factor due to prediction irregularity is usually excluded when analysing the effect of correlated errors in empirical study (e.g., in Ali & Pazzani (1996)), because it is difficult to be isolated in real-world datasets.

due to the fact that the estimation of the combination measure is hard and more error prone when the data size is small. This last note also applies when one or more classes are supported by only few instances in a dataset.

Though our investigation is limited to one type of combination method, we believe that the results in this paper 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, Chan & Stolfo (1995) have shown the potential of decision tree learning in the same working assumption[6].

6. Links to Theoretical Work

One phenomenon observed in the experiments is that the relative performance between theory 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, theory 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 (in Figure 11(a)), when the complete learning curve is not observed, this phenomenon still occurs.

This empirical result seems to be stronger, in terms of the expected performance of theory 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 theory combination's constituent theories. Our result indicates that theory combination can significantly outperform not only its constituent theories but the theory learned from aggregating the available data; in spite of the the fact that this result only shows the base-line behaviour of theory combination, i.e., combining two theories.

Some may assume that the theories learned in the near-asymptotic region would be very similar as shown by their indistinguishable performance. However, our experiment results suggest that the two theories 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 theories 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 theory can be significantly improved by combining multiple theories of the same learning algorithm.* This can only happen if the constituent theories are substantially different and there is still some

6. A subtle difference on the methods of theory combination employed might be worth noting here. Chan and Stolfo (1995) use stacked generalisation that essentially require all training data subsets at the low level to be seen by the high level generaliser during training. Whereas the training subsets are completely isolated for the method we used here.

regularity in the theories to be exploited by any combination methods. Figure 5(a) shows improvements (Figure 11(b) shows no improvements) as a result of theory 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 theories 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 (Schapire, 1996).

In regression settings, Meir (1994) mathematically analyses the effect of (linearly) combining several least squares linear estimators on the expected performance, under the same working assumption. While his general result agrees with ours, i.e., theory combination can significantly improve the performance of data combination in some situations, the details of the two results are at odds. Meir's result states that theory combination outperforms data combination for small training data size, and theory combination can be worse for intermediate sample sizes. This result is based on the analysis of bias/variance decomposition which are quite different in classification tasks (Kohavi & Wolpert, 1996). It also assumes that the data batches are independent for tractability; but our empirical result has no such assumption. Relaxing the non-overlapping and independence data subsets assumptions, Sollich and Krogh (1996) analytically show that theory combination can substantially improve the performance of data combination by optimizing the weights of the linear combination for intermediate sample size. This latter result is based a slightly different working assumption from ours.

7. Other Issues

A note of caution regarding the source of the data batches is in order here. Batches of data coming from seemingly similar but different sources, thus possibly different tasks, are not covered in our investigation in this paper. We do not think a theory combination method is a good choice in such a situation. The multitask learning method (Caruana, 1996) could be a better choice. The additional data can be used as an extra task to better support learning for the main task. A more subtle situation is that the data is pre-sorted into different groups according to some criterion when the data is collected (e.g., Baxt, 1992). This is an interesting scenario which we intend to explore in the near future.

When more than two batches are available, one possible method is to stack up the combinations in a binary tree structure as have been reported by Chan and Stolfo (1996). Nevertheless, more evidence is required to show that it is generally applicable. On the other aspect, there is no restriction that one must use a single learning algorithm for all batches of data. One may apply a model selection technique (Schaffer, 1993) to choose one among several learning algorithms for each batch of data, and then perform the combination. However, this incurs multiple folds of computational requirement.

The advantages of theory combination over data combination are that it allows faster learning time in some cases (see the analysis in the next paragraph) and uses less memory because the former employs less data for each learned theory, and separate copies of a learning algorithm can be run in parallel on multiple processors. These advantages make the theory combination approach more feasible for large datasets. Catlett (1991) studied a variety of sampling techniques to extract a subset from a large dataset for decision tree learning, but concluded that they are not a solution to the problem of scaling up to very large datasets. The results of our experiments with real-world datasets convey a message that sampling together with parallel processing can be a potential solution to this problem.

The additional computation required for theory combination differs from one method to the other and the type of learning algorithm used. For the combination method we used, the main additional computational load is the estimation of predictive accuracy during training. This requires a cross-validation method to be performed on the training data (see the Appendix). Assuming the time complexity of a learner is linear (such as NB*), i.e., $T(n, m) = n + m$, where n is the training data size and m is the testing data size. Because a three-fold cross-validation method is used, the theory combination method demands $T_s(n, m) = 4n + m$ for inducing a single theory. Data combination requires $T_d(2n, m) = 2n + m$. For example in the LED24 domain, NB* requires $T_s(n, m) = 6$ seconds and $T_d(2n, m) = 2.5$ seconds on a Sun SPARCserver 1000 machine, where $n = 1000$ and $m = 5000$. Whereas for an mn^2 bounded learner such as IB1*, it requires $T_s(n, m) = 6.5$ minutes and $T_d(2n, m) = 12$ minutes for the same example. Thus, parallel processing can help to speedup the computation in some cases, even with the relatively computational expensive theory combination method we used here.

8. Conclusions

This paper shows that theory combination in the multiple-data-batches scenario is generally applicable. A comparison of the base-line behaviour of theory combination and data combination using two randomly drawn disjoint data batches reveals that theory combination compares favourably when the performance gain due to data combination is small or the performances of the theories induced from the data batches are at the near-asymptotic region of the learning curve.

The practical implication of our results is that one should consider using theory 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 theory by combining multiple theories of the same algorithm if the theories are substantially different and there is some regularity in the theories to be exploited by the combination method used.

9. Acknowledgment

Ross Quinlan and Robert Schapire help to clarify the issue on (near-)asymptotic performance. Zijian Zheng, Bernhard Pfahringer, Mike Cameron-Jones, Ian Witten and members of the machine learning group 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 support from the Direct Grant from the Chinese University of Hong Kong and Department of Computer Science, University of Waikato.

Appendix - The Method of Theory Combination

We use the composite learner framework (Ting, 1996b) as the method for theory combination in our experiments. Ting uses the term *the characterisation of predictive accuracy* to mean the use of a measure in an induced theory 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 three-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.

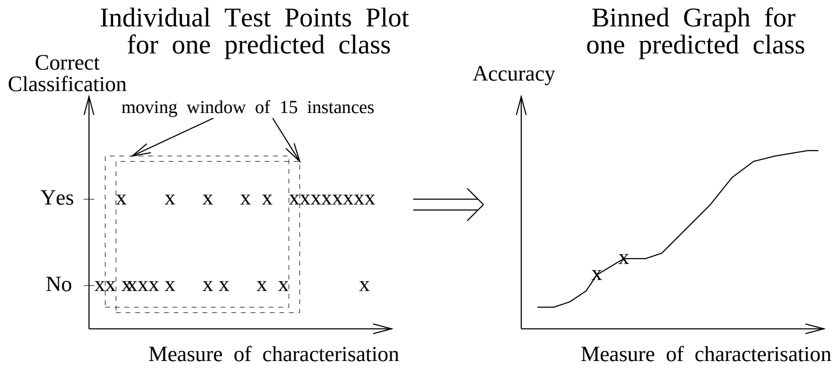

Fig. A. Transforming individual cross-validation test points to a binned graph for one predicted class.

For each predicted class, the individual test results from the three-fold cross-validation are then sorted according to the values of the characterisation (i.e., posterior probability, or typicality), 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. Each bin (the size is pre-determined) in the left plot is transformed to a point in the right plot by averaging the values of the characterisation for all test points in the bin. 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 theory induced from all the n training instances and the binned graphs for all classes are stored for future classification.

For each classification of an instance X, the theory which has the higher estimated predictive accuracy 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.

$$
\begin{aligned}
C_f &= C_{T1} \ \text{ if } \mathrm{PA}(C_{T1}, H_{T1}|X) > \mathrm{PA}(C_{T2}, H_{T2}|X), \\
&= C_{T2} \ \text{ if } \mathrm{PA}(C_{T1}, H_{T1}|X) < \mathrm{PA}(C_{T2}, H_{T2}|X), \\
&\phantom{= C_{T2}} \ \text{ else random select.}
\end{aligned}
$$

where C_f : final prediction;
C_T & H_T : Theory T'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.

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., 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.

Catlett, J. (1991), Megainduction: machine learning on very large databases. Doctoral dissertation, Basser Department of Computer Science, University of Sydney, Australia.

Caruana, R. (1996), Algorithms and Applications for Multitask Learning, in *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 87-95, 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.

Fayyad, U.M., N. Weir & S. Djorgovski (1993), SKICAT: A Machine Learning System for Automated Cataloging of Large Scale Sky Surveys, in *Proceedings of the Tenth International Conference on Machine Learning*, pp. 112-119, Morgan Kaufmann.

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.

Ho, T.K., J.J. Hull & S.N. Srihari (1994), Decision Combination in Multiple Classifier Systems, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 1, pp. 66-75.

Jacobs, R.A., M.I. Jordan, S.J. Nowlan & G.E. Hinton (1991), Adaptive Mixtures of Local Experts, in *Neural Computation 3*, pp. 79-87.

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

Kohavi, R. & D.H. Wolpert (1996), Bias Plus Variance Decomposition for Zero-One Loss Functions, in *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 275-283, Morgan Kaufmann.

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, MIT Press.

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.

Meir, R. (1994), Bias, Variance and the Combination of Estimators: The Case of Linear Least Squares, Technical Report No. 922, Department of Electrical Engineering, Technion, Haifa, Israel.

Merz, C.J. (1995), Dynamic Learning Bias Selection, in *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, FL: Unpublished, pp. 386-395.

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. (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.

Schaffer, C. (1993), Selecting a Classification Method by Cross-validation. *Preliminary Papers of the Fourth International Workshop on Artificial Intelligence and Statistics*, pp. 15-25.

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

Schapire, R.E. (1996), private communication.

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

Sollich, P. & A. Krogh (1996), Learning with ensembles: How overfitting can be useful, in *Advances in Neural Information Processing Systems 8*, D.S. Touretzky, M.C. Mozer & M.E. Hasselmo (Editors) , MIT Press.

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

Ting, K.M. (1994), *Discretization of Continuous-Valued Attributes and Instance-Based Learning*, Technical Report No.491, Basser Department of Computer Science, University of Sydney.

Ting, K.M. & R.M. Cameron-Jones (1994), Exploring a Framework for Instance Based Learning and Naive Bayesian Classifiers, in *Proceedings of the Seventh Australian Joint Conference on Artificial Intelligence*, pp. 100-107, World Scientific.

Ting, K.M. (1996a), Discretisation in Lazy Learning Algorithms, to appear in the special issue of Lazy Learning in *Artificial Intelligence Review Journal.*

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

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.