Stacking Bagged and Dagged Models

Kai Ming Ting and Ian H. Witten
Department of Computer Science
University of Waikato
Hamilton, New Zealand
{kaiming,ihw}@cs.waikato.ac.nz

Abstract

In this paper, we investigate the method of *stacked generalization* in combining models derived from different subsets of a training dataset by a single learning algorithm, as well as different algorithms. The simplest way to combine predictions from competing models is majority vote, and the effect of the sampling regime used to generate training subsets has already been studied in this context—when bootstrap samples are used the method is called *bagging*, and for disjoint samples we call it *dagging*. This paper extends these studies to stacked generalization, where a learning algorithm is employed to combine the models. This yields new methods dubbed *bag-stacking* and *dag-stacking*.

We demonstrate that bag-stacking and dag-stacking can be effective for classification tasks even when the training samples cover just a small fraction of the full dataset. In contrast to earlier bagging results, we show that bagging and bag-stacking work for stable as well as unstable learning algorithms, as do dagging and dag-stacking. We find that bag-stacking (dag-stacking) almost always has higher predictive accuracy than bagging (dagging), and we also show that bag-stacking models derived using two different algorithms is more effective than bagging.

1 Introduction

Wolpert (1992) proposed stacked generalization as a general method of using a high-level model to combine lower-level models to achieve greater predictive accuracy. Although it has met with some success for regression tasks (Breiman, 1996a), its application to classification tasks has been limited. However, very recently we have successfully applied stacked generalization to classification tasks (Ting & Witten, 1997), a domain in which there has been a significant amount of research on model combination. However, previous research has largely been restricted to such simple methods as majority vote and weight averaging (e.g., Breiman, 1996b; 1996d; Hansen & Salamon, 1990; Perrone & Cooper, 1993; Oliver & Hand, 1995).

The term *bagging* refers to the use of majority vote to combine multiple models derived from a single learning algorithm using bootstrap samples (Breiman, 1996b). Another, which we call *dagging*, is similar but uses disjoint samples rather than bootstrapping. The present paper investigates the use of a learned model instead of majority vote to combine the individual models, thereby adopting the framework of stacked generalization. We call the resulting methods *bag-stacking*, in which bootstrap samples are used as training data for the individual models, and *dag-stacking*, which uses disjoint samples.

Breiman (1996b) concluded that bagging only works with unstable learning algorithms such as decision tree learners. In contrast, our results show that bagging and bag-stacking (and also dagging and dag-stacking) both work with stable learning algorithms too—they can work well when the individual models are derived from just a small fraction of the full dataset. The implication of these results is that models can be derived much more quickly (because less training data is used), and that the methods of bagging and dagging, as well as bag-stacking and dag-stacking, apply to more learning algorithms than was previously thought.

This paper concentrates on comparing the predictive accuracy of bagging with bag-stacking, and dagging with dag-stacking, when used to combine models derived by a single learning algorithm and by two different learning algorithms. Section 2 formally introduces the notions of bagging, dagging, bag-stacking and dag-stacking. Section 3 reports results obtained when stacking models derived by a single learning algorithm, and Section 4 examines the stacking of models derived by two different learning algorithms. Section 5 discusses some further issues, followed by related work and conclusions.

2 Bagging, Dagging, Bag-Stacking and Dag-Stacking

Given a training dataset $\mathcal{L} = \{(y_n, x_n), n = 1, \ldots, N'\}$, where y_n is the class value of the nth instance and x_n is a vector representing its attribute values, we consider subsets of samples produced by one of two sampling regimes:

bootstrap samples — randomly sample $\mathcal{L}$ with replacement into K subsets of size N, where $N \leq N'$;

disjoint samples — randomly sample $\mathcal{L}$ without replacement into K disjoint subsets of size N, where $KN \leq N'$.

Use some learning algorithm to derive K models $\mathcal{M}_k$ from the subsets. The learning algorithm is called a *level-0 generalizer*, and the resulting models are *level-0 models*. The *bagging* method (Breiman, 1996b) uses majority vote to combine the classification outputs of models derived from bootstrap samples. The method that we call *dagging* uses the same majority vote to combine the outputs of models derived from disjoint samples.

Now, instead of majority vote, let us consider the use of a higher-level learning algorithm to combine the level-0 models in the spirit of stacked generalization (Wolpert, 1992; Ting & Witten, 1997). However, unlike the previous implementations of stacked generalization, we do not employ cross-validation to generate the higher-level data. Instead, we use $\mathcal{L}$ as a *test* set for each of the K models—despite the fact that subsets of $\mathcal{L}$ were used to train those models. Suppose that there are I output classes, and let $p_{ki}(x)$ denote the probability that the kth model assigns to the ith class given the test instance x. The vector

$$P_{kn} = (p_{k1}(x_n), \ldots, p_{ki}(x_n), \ldots, p_{kI}(x_n))$$

gives the kth model's class probabilities for the nth instance, and at the end of the testing process, the data assembled from the output of the K models is

$$\tilde{\mathcal{L}} = \{(y_n, P_{1n}, \ldots, P_{kn}, \ldots, P_{Kn}), n = 1, \ldots, N'\}.$$

This is called *level-1 data*. Use a learning algorithm, which we call the *level-1 generalizer*, to derive a model $\tilde{\mathcal{M}}$ that predicts the class from this level-1 data. $\tilde{\mathcal{M}}$ is called the *level-1 model*.

To classify a new instance, the level-0 models $\mathcal{M}_k$ are used to produce a vector $(p_{11}, .., p_{1I}, \ldots, p_{k1}, .., p_{kI}, \ldots, p_{K1}, .., p_{KI})$ which is input to the level-1 model $\tilde{\mathcal{M}}$, and the output of $\tilde{\mathcal{M}}$ is the final classification result for that instance. To estimate each method's predictive accuracy we always use a completely separate test dataset $\mathcal{T}$.

Depending on the sampling strategy used to produce the data from which the level-0 models are derived, we call this implementation of stacked generalization *bag-stacking* or *dag-stacking* (**b**ootstrap/**d**isjoint samples **ag**gregation by **stacking**).

The following subsections describe pertinent details of the level-0 and level-1 generalizers used in this paper.

2.1 Level-0 Generalizers

Two learning algorithms are used at level 0: C4.5, the well-known decision tree learner (Quinlan, 1993), and NB, a re-implementation of a naive Bayesian classifier (Cestnik, 1990). Only unpruned trees are derived from C4.5 since, as Breiman (1996b) discovered, aggregation from bagged models seems to eliminate overfitting.[1] It is necessary for our implementation that the level-0 generalizers produce output class probabilities $p_i(x)$ for any instance x (where, in all cases, $\sum_i p_i(x) = 1$), and we now exhibit the formulas that are used to estimate this.

C4.5: Consider the leaf of the decision tree at which the instance x falls. Let m_i be the number of (training) instances with class i at this leaf, and suppose the majority class at the leaf is $\hat{I}$. Let $E = \sum_{i \neq \hat{I}} m_i$. Then

$$p_{\hat{I}}(x) = 1 - \frac{E+1}{\sum_i m_i + 2},$$
$$p_i(x) = (1 - p_{\hat{I}}(x)) \times \frac{m_i}{E}, \text{ for } i \neq \hat{I}.$$

NB: Let $p(i|x)$ be the posterior probability of class i, given instance x. Then

$$p_i(x) = \frac{p(i|x)}{\sum_i p(i|x)}.$$

[1]Our experiments confirm this.

In both cases the class that the level-0 model predicts for an instance x is that $\hat{I}$ for which $p_{\hat{I}}(x) > p_i(x)$ for all $i \neq \hat{I}$.

Breiman (1996b) claims that bagging can only improve the predictive accuracy of learning algorithms that are unstable, where an "unstable" learning algorithm is one for which small perturbations in the training set can produce large changes in the derived model. C4.5 and NB are unstable and stable respectively, which enables us to investigate bag-stacking and dag-stacking under both conditions.

In Section 3, only one learning algorithm—either C4.5 or NB—is used to derive all of the level-0 models. In Section 4, both are used together.

2.2 The Level-1 Generalizer

We previously discovered that stacked generalization works well when a multi-response linear regression algorithm, MLR, is used as the level-1 generalizer (Ting & Witten, 1997). Consequently we use the same algorithm for both bag-stacking and dag-stacking.

MLR is an adaptation of a least-squares linear regression algorithm that Breiman (1996a) used in regression settings. Any classification problem with real-valued attributes can be transformed into a multi-response regression problem. If the original classification problem has I classes, it is converted into I separate regression problems, where the problem for class ℓ has instances with responses equal to one when they have class ℓ and zero otherwise.

The input to MLR is level-1 data. The linear regression for class ℓ is simply

$$LR_\ell(x) = \sum_{k=1}^{K} \sum_{i=1}^{I} \alpha_{ki\ell} p_{ki}(x).$$

Choose the coefficients $\{\alpha_{ki\ell}\}$ to minimize

$$\sum_{(y_n, x_n) \in \mathcal{L}} \left[y_n - \sum_k \sum_i \alpha_{ki\ell} p_{ki}(x_n) \right]^2 .$$

The coefficients $\{\alpha_{ki\ell}\}$ are constrained to be non-negative. This is accomplished by using a constrained least-squares algorithm described by Lawson & Hanson (1995) to derive non-negative regression coefficients for each class.

We are now in a position to describe the working of MLR. To classify a new instance x, compute $LR_\ell(x)$ for all I classes and assign the instance to that class ℓ which has the greatest value: $LR_\ell(x) > LR_{\ell'}(x)$ for all $\ell' \neq \ell$.

Table 1: Datasets used in experiments

Dataset	#Training	#Classes	#Attr.	#Test
DNA	2000	3	60	1186
Satellite	4435	6	36	2000
Letters	15000	26	16	5000
Shuttle	43500	7	9	14500

3 Stacking models derived by one algorithm

We now describe experiments to investigate bag-stacking and dag-stacking of models that are derived by a single learning algorithm. The four datasets used are the moderate to large ones used in the Statlog project (Michie *et al.*, 1994) and also by Breiman (1996b, 1996d); they are summarized in Table 1. Each includes separate training and test sets. Small datasets are not used because we investigate models derived from only a fraction of the original dataset.

The following subsections compare the predictive error rate of bagging to that of bag-stacking, and dagging to that of dag-stacking. We vary the data size N from which level-0 models are derived, and also the number K of level-0 models that are combined together.

3.1 Bag-Stacking

For each dataset, parts of the training data $\mathcal{L}$ are used to derive models and the entirely separate test set $\mathcal{T}$ is used to assess their error rate.

E_S	is the error rate of a single model derived from all of $\mathcal{L}$;
E_B	and E_{BS} are the error rates of bagging and bag-stacking respectively.

Tables 2 and 3 show these figures for the level-0 generalizers C4.5 and NB respectively. In Table 2, which uses C4.5, E_S is the testing error rate of a *pruned* tree; as noted earlier, unpruned trees are used elsewhere. The following four columns give E_B and E_{BS} for $K = 10$, 20 and 50, based on level-0 models each derived from just N instances of the training set $\mathcal{L}$, for small values of N and for $N = N'$. The size of the samples used is indicated in the first column. Results for $K = 50$ are not available for the Letters dataset because they take too long to compute (see Section 5).

Table 2: Error rates when bagging and bag-stacking C4.5 models

Dataset		$K = 10$		$K = 20$		$K = 50$	
N	E_S	E_B	E_{BS}	E_B	E_{BS}	E_B	E_{BS}
DNA							
100		9.6	**5.7**	6.5	**5.4**	6.4	6.0
200		9.6	**5.7**	7.8	**4.9**	6.9	**5.3**
400		6.5	6.3	**4.9**	**4.8**	$\hat{\mathbf{4.8}}$	$\underline{\check{\mathbf{4.4}}}$
800		6.0	6.0	**5.6**	**5.4**	**5.0**	**5.0**
2000	5.8	6.0	6.0	**4.9**	**4.9**	$\hat{\mathbf{4.8}}$	**5.0**
Satellite							
100		17.7	16.9	16.9	15.6	16.4	**14.2**
200		16.2	15.5	15.8	**14.1**	15.6	**13.8**
400		14.9	**13.7**	**14.2**	**13.5**	**13.5**	**12.5**
800		**13.2**	**12.9**	**12.6**	**12.0**	**12.9**	**12.1**
4435	14.8	**11.5**	**11.6**	**11.9**	**11.8**	$\hat{\mathbf{11.2}}$	$\underline{\check{\mathbf{11.0}}}$
Letters							
100		43.2	34.1	36.6	28.6		
200		35.2	28.7	29.8	24.5		
400		27.7	24.2	23.2	20.4	NA	NA
800		21.8	19.7	18.4	16.5		
1600		16.4	15.6	14.5	13.3		
15000	12.9	**8.6**	**8.4**	$\hat{\mathbf{7.5}}$	$\underline{\check{\mathbf{7.1}}}$		
Shuttle							
400		0.586	0.462	0.614	0.393	0.559	0.276
800		0.510	0.379	0.497	0.283	0.510	0.234
1600		0.372	0.179	0.366	0.172	0.310	0.097
3200		0.200	0.131	0.193	0.117	0.179	0.090
43500	0.48	**0.028**	$\underline{\check{\mathbf{0.007}}}$	$\underline{\hat{\mathbf{0.021}}}$	$\underline{\check{\mathbf{0.007}}}$	**0.028**	$\underline{\mathbf{0.014}}$

For each dataset, bold face is used to indicate error rates that are lower than the value of E_S. In addition, underlining is used to compare the results with the value of E_B that was obtained using the largest values for K and N, that is, with $K = 50$ level-0 models ($K = 20$ for the Letters dataset) each trained using the size of the full dataset ($N = N'$). The lowest error rates for E_B and E_{BS} in each dataset are marked with $\hat{}$ and $\check{}$ respectively.

The figures in bold show the remarkable result that both bagging and bag-stacking can achieve better predictive accuracy than a single model derived using the entire dataset—even when their level-0 models are derived from a small fraction of the dataset. This is apparent in the DNA and Satellite datasets when combining C4.5 models, and in all datasets when combining NB models. In almost all cases, bag-stacking yields superior performance to bagging, except in only two cases in combining C4.5 models (i.e., N=2000, K=50 in the DNA dataset; N=4435, K=10 in the Satellite dataset).

Table 3: Error rates when bagging and bag-stacking NB models

Dataset		$K = 10$		$K = 20$		$K = 50$	
N	E_S	E_B	E_{BS}	E_B	E_{BS}	E_B	E_{BS}
DNA							
100		10.5	7.1	9.0	5.8	9.9	4.8
200		8.7	6.0	8.4	5.3	9.5	4.9
400		5.2	4.7	5.3	$\underline{\mathbf{4.1}}$	5.2	$\underline{\mathbf{4.1}}$
800		4.8	$\underline{4.3}$	4.8	$\underline{\mathbf{3.8}}$	4.7	$\underline{\check{\mathbf{3.6}}}$
2000	4.2	4.5	$\underline{4.2}$	4.5	$\underline{\check{\mathbf{3.6}}}$	$\hat{4.4}$	$\underline{\mathbf{3.7}}$
Satellite							
100		$\underline{\mathbf{20.9}}$	$\underline{\mathbf{16.1}}$	$\underline{\mathbf{21.0}}$	$\underline{\mathbf{16.4}}$	$\underline{\hat{\mathbf{20.7}}}$	$\underline{\mathbf{15.2}}$
200		$\underline{\mathbf{21.1}}$	$\underline{\mathbf{17.6}}$	$\underline{\mathbf{21.5}}$	$\underline{\mathbf{16.4}}$	$\underline{\mathbf{20.8}}$	$\underline{\mathbf{15.2}}$
400		$\underline{\mathbf{22.7}}$	$\underline{\mathbf{17.1}}$	$\underline{\mathbf{22.5}}$	$\underline{\mathbf{16.7}}$	$\underline{\mathbf{21.0}}$	$\underline{\check{\mathbf{15.1}}}$
800		$\underline{\mathbf{22.9}}$	$\underline{\mathbf{20.5}}$	**23.1**	$\underline{\mathbf{20.0}}$	$\underline{\mathbf{23.0}}$	$\underline{\mathbf{19.3}}$
4435	23.3	$\underline{\mathbf{23.2}}$	$\underline{\mathbf{21.1}}$	23.3	$\underline{\mathbf{21.2}}$	**23.1**	$\underline{\mathbf{20.9}}$
Letters							
100		57.8	39.1	47.7	$\underline{\mathbf{30.4}}$		
200		42.6	$\underline{\mathbf{29.1}}$	**37.7**	$\underline{\check{\mathbf{24.4}}}$		
400		39.9	$\underline{\mathbf{28.4}}$	38.0	$\underline{\mathbf{25.6}}$	NA	NA
800		39.2	$\underline{\mathbf{29.4}}$	37.9	$\underline{\mathbf{27.1}}$		
1600		38.5	$\underline{\mathbf{30.8}}$	**37.5**	$\underline{\mathbf{28.7}}$		
15000	38.7	**37.7**	$\underline{\mathbf{32.6}}$	**37.5**	$\underline{\mathbf{32.1}}$		
Shuttle							
400		$\underline{\mathbf{8.179}}$	$\underline{\mathbf{7.614}}$	$\underline{\mathbf{8.152}}$	$\underline{\mathbf{7.593}}$	$\underline{\mathbf{8.124}}$	$\underline{\mathbf{6.628}}$
800		$\underline{\mathbf{8.048}}$	$\underline{\mathbf{6.310}}$	$\underline{\mathbf{8.159}}$	$\underline{\mathbf{6.366}}$	$\underline{\mathbf{8.076}}$	$\underline{\check{\mathbf{6.297}}}$
1600		$\underline{\mathbf{8.145}}$	$\underline{\mathbf{7.248}}$	$\underline{\mathbf{8.097}}$	$\underline{\mathbf{7.041}}$	$\underline{\hat{\mathbf{8.028}}}$	$\underline{\mathbf{6.366}}$
3200		$\underline{\mathbf{8.090}}$	$\underline{\mathbf{7.145}}$	$\underline{\mathbf{8.303}}$	$\underline{\mathbf{7.097}}$	$\underline{\mathbf{8.221}}$	$\underline{\mathbf{6.531}}$
43500	9.75	$\underline{\mathbf{9.297}}$	$\underline{\mathbf{6.759}}$	$\underline{\mathbf{9.407}}$	$\underline{\mathbf{6.724}}$	**9.455**	$\underline{\mathbf{6.724}}$

Now turn attention to the underlined figures. It is apparent when bagging and bag-stacking C4.5 models that the error rate tends to decrease as the training size N increases—as one might expect. However, the evidence when bagging and bag-stacking NB models is mixed. While the expected trend is followed in the DNA dataset, in the Satellite dataset the error rate seems to *increase* with N. In the Letters dataset, the error rate of bag-stacking demonstrates a U-shape trend while bagging follows a normal trend as N increases. Nonetheless, in almost all cases the predictive error rates of both bagging and bag-stacking decrease when K increases.

In bagging and bag-stacking C4.5 models, the lowest error rate is achieved by bag-stacking models derived using a small fraction of the full DNA dataset. In the other three datasets, the lowest error rate is achieved by bag-stacking models derived using the size of the full dataset. In bagging and bag-stacking NB models, the lowest error rate is achieved by bag-stacking models derived using a small fraction of the full dataset in

Table 4: Bagging, dagging, bag-stacking and dag-stacking C4.5 models

Dataset	E_S	E_B	E_{BS}	E_D	E_{DS}
DNA					
N=100, K=20		6.5	5.4	8.3	5.4
N=200, K=10		9.6	5.7	7.8	5.6
N=400, K=5		9.1	8.6	7.6	6.0
all data	5.8				
Satellite					
N=100, K=44		16.2	14.5	16.5	14.7
N=200, K=20		15.8	14.1	15.2	13.4
N=400, K=10		14.9	13.7	14.0	14.4
N=800, K=5		14.4	14.0	14.9	14.2
all data	14.8				
Letters					
N=400, K=20		23.2	20.4	23.9	21.0
N=800, K=10		21.8	19.7	21.0	18.6
N=1600, K=5		20.1	18.2	20.6	18.4
all data	12.9				
Shuttle					
N=400, K=50		0.559	0.276	0.510	0.297
N=800, K=50		0.510	0.234	0.517	0.207
N=1600, K=20		0.366	0.172	0.345	0.172
N=3200, K=10		0.200	0.131	0.221	0.172
all data	0.048				

Table 5: Bagging, dagging, bag-stacking and dag-stacking NB models

Dataset	E_S	E_B	E_{BS}	E_D	E_{DS}
DNA					
N=100, K=20		9.0	5.8	10.5	5.0
N=200, K=10		8.7	6.0	5.8	4.8
N=400, K=5		5.4	4.8	4.8	4.7
all data	4.2				
Satellite					
N=100, K=44		20.8	14.9	20.7	15.0
N=200, K=20		21.5	16.4	21.5	17.4
N=400, K=10		22.7	17.1	22.2	20.0
N=800, K=5		22.7	20.4	23.9	20.2
all data	23.3				
Letters					
N=400, K=20		38.0	25.6	38.0	24.5
N=800, K=10		37.7	32.6	37.7	28.6
N=1600, K=5		38.0	31.4	38.4	31.2
all data	37.8				
Shuttle					
N=400, K=50		8.124	6.628	8.062	6.324
N=800, K=50		8.076	6.297	7.966	7.021
N=1600, K=20		8.097	7.041	7.910	6.607
N=3200, K=10		8.090	7.145	8.028	6.821
all data	9.75				

all four datasets.

3.2 Dag-Stacking

This section compares two methods of combining dagged models. As in the above investigations, all error rates are calculated using the test set $\mathcal{T}$, which is entirely separate from the training data $\mathcal{L}$.

> E_D and E_{DS} are the error rates of dagging and dag-stacking respectively.

The figures E_S, E_B and E_{BS} from the previous subsection are also included, for ease of comparison. Results are tabulated in Table 4 for C4.5 models and Table 5 for NB models. The first column indicates the values of K and N used—because subsets are disjoint for dagged models it is necessary that $KN \leq N'$ in all cases. In order to reduce the number of tests that had to be done, E_S, E_B and E_{BS} figures from the previous work were re-used by choosing $K = 5$, 10, 20, 50 subsets of $N = 100$, 200, 400, 800, 1600, and 3200 instances wherever possible. For the DNA dataset, the 2000 training instances were split into 20, 10, and 5 equal subsets. For the Satellite dataset, 4400 of the 4435 training instances were split into 44 subsets, and 4000 of them were split into 20, 10, and 5 subsets. For Letters, only about half of the training data was used, and split into 20, 10, and 5 subsets. For the Shuttle data, 20,000 and 40,000 of the 43,500 training instances were split into 50 subsets, and 32,000 of them were split into 20 and 10 subsets.

Examination of the values of E_D and E_{DS} in the final column of both tables reveals that dag-stacking, like bag-stacking, almost always yields a lower predictive error rate than dagging. The only exception is combining C4.5 models with $N = 400$ and $K = 10$ in the Satellite dataset. However, comparing bag-stacking with dag-stacking, E_{BS} *vs* E_{DS}, gives no clear indication of either method being superior to the other. The same is true when comparing bagging with dagging, E_B *vs* E_D.

Summary

Summarizing the conclusions from these experiments, we find that

- stacking using MLR almost always yields lower predictive error rate than majority vote when combining either bagged or dagged models;

- bag-stacking and dag-stacking have comparable predictive accuracy, as have bagging and dagging;
- when using bagging or bag-stacking, it is sometimes better to use only a small fraction of the entire dataset to derive the models;
- bagging, dagging, bag-stacking and dag-stacking all work well with both unstable (i.e., C4.5) and stable (i.e., NB) learning algorithms.

4 Bag-stacking models derived by two different algorithms

In this section, we investigate bag-stacking models derived using both C4.5 and NB, and compare the predictive error rate to that of bagging. Table 6 shows the result of bagging and bag-stacking when the same number of models was generated by C4.5 and NB from the same sample size. The final two columns give the corresponding figures when only one learning algorithm is used.

These results reveals an important feature of bagging: unless the predictive error rate of the base models is fairly close, bagging does not improve accuracy. This is apparent in the last three datasets. Here the difference in error rate E_B between the two homogeneous models is large, and in each case the performance of bagging models derived from heterogeneous level-0 generalizers falls far short of that for the better of the two homogeneous cases. On the other hand, in the DNA dataset the heterogeneous model outperforms the better of the two homogeneous models because the difference in error rate is much smaller. This accords with the results of Ting & Witten (1997) when combining three different types of learning algorithms using majority vote. Although bag-stacking suffers from the same problem, it does so to a far smaller extent—as the last three datasets show.

To further investigate this phenomenon, we tried combining unequal numbers of models derived by C4.5 and NB from different values of N, so long as the two homogeneous cases yield comparable performance. Two datasets, in which combining heterogeneous models performs worse than combining homogeneous models, are used for this investigation: Satellite and Letters.[2]

[2]The Shuttle dataset is not used because the difference between the two homogeneous models are so great that no comparable performance can be obtained (see Tables 2 and 3).

Table 6: Bagging and bag-stacking level-0 models from different algorithms

	C4.5 & NB		C4.5		NB	
Dataset	E_B	E_{BS}	E_B	E_{BS}	E_B	E_{BS}
DNA						
N=100, K=20×2	6.0	4.8	6.3	5.7	9.9	5.2
N=200, K=10×2	4.6	3.3	7.8	4.9	8.4	5.3
N=400, K=5×2	4.9	4.1	6.5	6.3	5.2	4.7
Satellite						
N=200, K=20×2	18.4	14.7	15.7	14.3	21.7	17.2
N=400, K=10×2	20.1	14.5	14.2	13.5	22.5	16.7
N=800, K=5×2	18.4	13.2	13.2	12.9	22.9	20.5
Letters						
N=800, K=10×2	26.1	18.2	18.4	16.5	37.9	27.1
N=1600, K=5×2	26.0	18.2	16.4	15.6	38.5	30.8
Shuttle						
N=1600, K=10×2	3.310	0.172	0.366	0.172	8.097	7.041
N=3200, K=10×2	3.407	0.124	0.193	0.117	8.303	7.097

Table 7: Bagging and bag-stacking different numbers of level-0 models from two learning algorithms

	C4.5		NB		C4.5 & NB	
Dataset	E_B	E_{BS}	E_B	E_{BS}	E_B	E_{BS}
Satellite	$K = 10$		$K = 50$		$K = 10 + 50$	
N=100	17.7	16.9	20.7	15.2	20.2	14.0
N=200	16.2	15.5			20.0	14.0
N=400	14.9	13.7			19.9	13.2
N=800	13.2	12.9			19.7	12.6
Letters	$K = 10$		$K = 20$		$K = 10 + 20$	
N=200	35.2	28.7	37.7	24.4	31.5	22.5
N=400	27.7	24.2			27.7	19.8
N=800	21.8	19.7			24.6	17.5
N=1600	16.4	15.6			20.9	14.8

We refer to Tables 2 and 3 to choose the values of K and N, such that the two homogeneous cases have comparable performance. In both datasets, we choose the (near-)best performing bagging and bag-stacking NB models, i.e., $K = 50$ and $N = 100$ for the Satellite dataset, and $K = 20$ and $N = 200$ for the Letters dataset. Then, choose the comparable performing C4.5-derived models: $K = 10$, and $N = 100$ to 800 for the Satellite dataset, and $N = 200$ to 1600 for the Letters dataset. We re-state the results with these settings in the second and third columns of Table 7. Because bag-stacking is more tolerant of differences in level-0 performance, we expect bag-stacking heterogeneous models to yield better predictive accuracy in more cases than bagging in this experiment.

The results of the heterogeneous model are shown in the last column of Table 7. With bagging (E_B), the heterogeneous model has lower error rate than that for both homogeneous models in just one case ($N = 200$ in the Letters dataset) where the difference in error rates is small. With bag-stacking (E_{BS}), the heterogeneous model yields lower predictive error rates than both of the homogeneous ones in all cases. These results confirm our expectation.

5 Discussion

When the full dataset is used to generate each level-0 model, our results for bagging are in agreement with those of Breiman (1996b)—bagging increases the predictive accuracy of unstable learning algorithms but not stable ones. However, when just a small proportion of the data is used to generate level-0 models, we find that bagging can improve the predictive accuracy of stable learning algorithms too. Although this is in accordance with the results of two studies of base-line behavior of a different kind of dagging (Ting & Low, 1997; Ting & Witten, 1997), which uses different model combination methods from majority vote, it contradicts the conventional wisdom that "the more data the better." It certainly has significant implications for learning time: since each level-0 model uses much less training data, it can be obtained much faster.

Like bagging and dagging, bag-stacking and dag-stacking are ideally suited to parallel processing because each level-0 model can be constructed independently. Moreover, because it uses the MLR method, level-1 learning can also benefit from parallelism. For a I-class problem, the regression for each class can be carried out independently on I CPUs.

The execution time of MLR depends on the number of classes involved as well as on the training set size. When K models are combined in a I-class problem, the number of attributes for level-1 data is KI and the regression algorithm must be executed I times. Figure 1 shows the execution time of MLR for the four datasets on a Sun SPARCserver 1000 machine. In Letters, which has 26 classes, execution time increases dramatically with the number of models. The increase is little more than linear for K up to 50 in the other three datasets, where the maximum number of classes is seven.

Note that the training data size for level-1 learning need not be N'. Since every learning algorithm has its own learning curve which levels off at the tail of the curve as the training data size increases, in most large databases, MLR only requires part of the full dataset to yield a model that performs well.

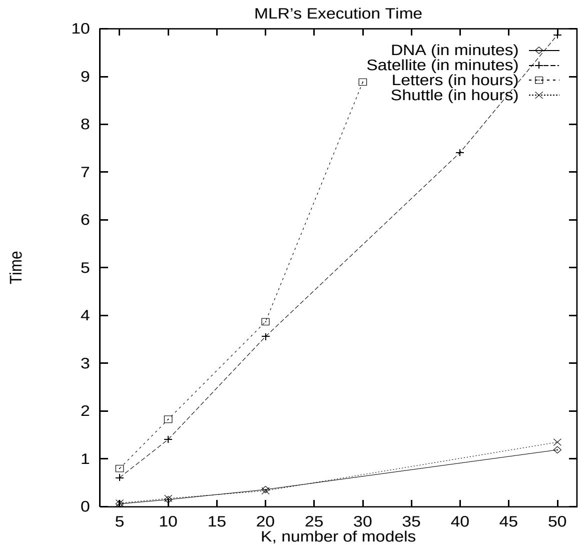

Figure 1: Computation time for MLR

It might be thought obvious that bag-stacking (or dag-stacking) will outperform bagging (or dagging) because the additional level-1 learning inevitably provides more information than a mere majority vote. However, not any learning algorithm is suitable for the level-1 generalizer. Ting & Witten (1997) show that of four learning algorithms tested, only MLR performs satisfactorily (the other three were C4.5, NB and IB1).

Note that in cases where the size of the full dataset is required to train level-0 models, the improvement of bag-stacking over bagging is usually small. One reason is that the level-1 data is produced from the majority of the data used to train level-0 models. Thus the probabilities (i.e., P_{kn}) employed in the level-1 data are over-estimated. One way to rectify this problem is to use the out-of-bag estimation (Breiman, 1996e) to get a better estimate of the probabilities.

6 Related Work

The research reported in this paper was inspired by Breiman (1996b, 1996d), as well as by our own work (Ting & Witten, 1997). Breiman (1996b) introduces the idea of bagging, and Breiman (1996d) also shows that combining models (using majority vote) derived using a small fraction of the entire dataset gives accuracy better than that of a single model derived using the entire dataset. Our contribution here is to show that stacking these procedures generally works even better than combining them using majority vote.

Wolpert introduced stacked generalization as long ago as 1992, and Ting & Witten (1997) shows convincingly how to make it work in classification tasks. The key is the use of output class probabilities of level-0 models as level-1 data, and the use of MLR as the level-1 generalizer. The present paper incorporates this framework, except that cross-validation is not used (details of the differences between implementations are summarized in Appendix). Other work on stacked generalization in classification tasks either has a more limited focus or evaluates the results on just a few datasets (LeBlanc & Tibshirani, 1993; Chan & Stolfo, 1995; Kim & Bartlett, 1995; Merz, 1995; Fan *et al.*, 1996).

Several researchers have investigated various methods of combining models produced by a single learning algorithm from the entire dataset. Different models have been generated by varying the learning parameters (Hansen & Salamon, 1990; Perrone & Cooper, 1993; Kwok & Carter, 1990; Oliver & Hand, 1995; Kononenko & Kovačič, 1992) and by using different sampling methods (Freund & Schapire, 1996; Ali & Pazzani, 1996). Techniques used to combine the individual models include (weighted) majority vote, weighted averaging, Bayesian/likelihood combination and distribution summation. None of this work uses a learning algorithm to perform level-1 learning.

Ting & Low (1997) study the base-line behavior of dagging empirically. Theoretical work on dagging includes Kearns & Seung (1995) and Meir (1994).

7 Conclusions and future work

This paper shows how stacked generalization can be successfully applied to combine bagged or dagged models derived from a single or multiple learning algorithms. Stacking using MLR almost always yields a lower predictive error rate than majority vote when combining either bagged or dagged models. Both bag-stacking and dag-stacking work for stable as well as unstable learning algorithms, even using subsets which cover only a small fraction of the full dataset to derive the level-0 models.

When combining models derived from different learning algorithms, it is necessary that the models perform comparably in order to guarantee increased predictive accuracy through bagging or bag-stacking. However, bag-stacking is more tolerant than bagging of differences in level-0 performance.

Dagging (dag-stacking) has been shown to be comparable to bagging (bag-stacking). This finding opens up an application of dagging and dag-stacking to on-line learnings, where data constantly arrives in batches.

One can also apply the same stacked generalization framework to arcing (Breiman, 1996c), yielding "arc-stacking" or the stacking of arced models. We plan to investigate this method in the near future.

Acknowledgments

The authors would like to thank the New Zealand Marsden Fund for financial support for this research.

Appendix—Implementations of stacked generalization

This Appendix recounts the key differences between stacked generalization as used in this paper and that described by Ting & Witten (1997). We denote the latter by SG for ease of reference. The differences are in the training process, and mainly affect the computational requirements.

Suppose that there are K level-0 models.

- SG relies on cross-validation to obtain level-1 data. Suppose J-fold cross-validation is employed. Let the learning time for each level-0 model be C and the testing time for each instance be t. Then the computational time required for the preparation of level-1 data is
$$K(JC + N't),$$
where $N' =$ is the size of the give dataset $\mathcal{L}$.
- Bag-stacking and dag-stacking need just one round of learning for each level-0 model. The computational time required to prepare the level-1 data is
$$K(C + N't).$$

We conclude that SG needs a factor of J more computation time than bag-stacking and dag-stacking, since $N't \ll C$ for a learning algorithm such as C4.5 and $N't \simeq C$ for NB. Moreover, C in bag-stacking and dag-stacking is often much less than for SG, since less training data is necessary.

Finally, SG requires a final training which uses the entire dataset $\mathcal{L}$ for each level-0 model to complete the whole training process. No such re-training is required for bag-stacking and dag-stacking.

The reader is referred to Ting & Witten (1997) for differences between the initial proposal of stacked generalization by Wolpert (1992) and the recent successful implementation for classification tasks.

References

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

Breiman, L. (1996a), Stacked Regressions, *Machine Learning*, Vol. 24, pp. 49-64.

Breiman, L. (1996b), Bagging Predictors, *Machine Learning*, Vol. 24, pp.123-140.

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

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

Breiman, L. (1996e), Out-of-bag Estimation, (ftp.stat.berkeley.edu/users/pub/breiman/OOBestimation.ps).

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.

Fan, D.W., P.K. Chan, S.J. Stolfo (1996), A Comparative Evaluation of Combiner and Stacked Generalization, in *Proceedings of AAAI-96 workshop on Integrating Multiple Learned Models*, pp. 40-46.

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.

Kim, K. & E.B. Bartlett (1995), Error Estimation by Series Association for Neural Network Systems, *Neural Computation 7*, pp. 799-808, MIT Press.

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.

Kwok, S. & C. Carter (1990), Multiple Decision Trees, *Uncertainty in Artificial Intelligence 4*, R. Shachter et al (Eds), pp. 327-335, North-Holland.

Lawson C.L. & R.J. Hanson (1995), *Solving Least Squares Problems*, SIAM Publications.

LeBlanc, M & R. Tibshirani (1993), Combining Estimates in Regression and Classification, TR 9318, Department of Statistics, University of Toronto.

Meir, R. (1994), Bias, Variance and the Combination of Estimators: The Case of Linear Least Squares, TR 922, Dept 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.

Michie, D., D.J. Spiegelhalter and C.C. Taylor (1994), *Machine Learning, Neural and Statistical Classification*, Ellis Horwood Limited.

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.

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

Ting, K.M. & B.T. Low (1997), Model Combination in the multiple-data-batches scenario, to appear in *Proceedings of the Ninth European Conference on Machine Learning.*

Ting, K.M. & I.H. Witten (1997), Stacked Generalization: when does it work?, to appear in *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence.* A long version is available as *Working Paper 97/3*, Department of Computer Science, University of Waikato. (http://www.cs.waikato.ac.nz/cs/Staff/kaiming.html).

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