



The University of Sydney

Accurate and Efficient Selection of Voting Ensembles

Technical Report Number 564

February 2005

Daren Ler¹, David Abraham², Liz Crawford², Irena Koprinkska¹

¹ School of Information Technologies
University of Sydney

² Department of Computer Science
Carnegie Mellon University

ISBN 186487 711 1

**School of Information Technologies
University of Sydney NSW 2006**

Accurate and Efficient Selection of Voting Ensembles

Daren Ler¹, David Abraham², Elisabeth Crawford², and Irena Koprinska¹

¹School of Information Technologies, University of Sydney, NSW 2006, Australia
{ler, irena}@cs.usyd.edu.au

²Computer Science Department, Carnegie Mellon University, Pittsburgh PA, 15232, USA
{dabraham, ehc+}@cs.cmu.edu

Abstract

In this paper, we present a method for efficient selection of heterogeneous majority voting ensembles. Given a set \mathcal{A} of algorithms, the set of possible voters \mathcal{V} over \mathcal{A} is exponential in the size of \mathcal{A} . Thus, it is not computationally feasible to select the best ensemble by evaluating each possibility. Instead, we compute the classification accuracies of a small subset of $\mathcal{A} \cup \mathcal{V}$, and use these values to predict the accuracy of all the remaining elements in the union. We demonstrate that this procedure, called landmarking, estimates performance well, allowing for the selection of good voting ensembles at significantly reduced computational cost. We also conduct statistical tests to show a link between the correlation of performance patterns and diversity of a pair of algorithms.

1 Introduction

Given a new dataset requiring classification, it is hard to know which set of learning algorithms form the voting ensemble with the greatest generalisation ability (Wolpert, 1996a, 1996b; Michie et al., 1994). If unlimited time was available to make the decision, hold-out testing – e.g. cross-validation, could be used to evaluate the performance of a large set of voting ensembles (e.g. see (Schaffer, 1997)). However, the number of dataset representations and the number of voters to consider makes this option unviable, particularly for real-world applications.

There are many real-world learning problems where efficient methods for selecting voting ensembles are needed. Consider for instance the problem of deploying email classification agents in an organisation. It has been demonstrated (Crawford et al., 2002), that due to the different ways in which people classify their email, different algorithms work better for different users. As such, when setting up an email classifier for a user, we would like to be able to evaluate the performance of many different algorithms and ensembles so we can choose the best one for that individual.

Recently, a means of efficiently estimating the performance of a set of algorithms on a dataset, without having to evaluate all of the algorithms was proposed (Ler et al., 2004a, 2004b, 2005). Based on meta-learning (Giraud-

Carrier et al., 2004; Vilalta and Drissi, 2002) and landmarking (Ler et al., 2004a; Pfahringer et al., 2000), the method uses the performance values of a subset of the available algorithms to estimate the performance of the entire set. More specifically, given a set of k algorithms \mathcal{A} , and a corpus of n datasets \mathcal{S} , the method calculates the performance of each algorithm in \mathcal{A} on each dataset in \mathcal{S} . Given a new dataset, s_{new} , the performances of $a_i \in \mathcal{L}' \subset \mathcal{A}$ on s_{new} are calculated and used to estimate the performances of $a_j \in \mathcal{A} \setminus \mathcal{L}'$. The estimates are based on the performance pattern of the algorithms in \mathcal{A} observed over the datasets in \mathcal{S} . Experiments in (Ler et al., 2004a) demonstrate empirically that the performance of all the algorithms in \mathcal{A} can be estimated with moderate success, even when $|\mathcal{L}'|$ is small. Returning to our email example, we can evaluate the performance of all applicable classification algorithms on the datasets for a subset of the users in an organisation. We can then use a method such as the one described in (Ler et al., 2004a) to efficiently estimate the best algorithm to use for each of the other users' email.

Previous work has shown that ensembles can be more accurate than their component algorithms if these components disagree with each other – or rather, are diverse (Hanson and Salamon, 1990; Krogh and Vedelsby, 1995). As such, we would like to be able to efficiently estimate the performance of ensembles, something that is not looked at in (Ler et al., 2004a). More formally, we want to consider voting committees \mathcal{V} comprised of subsets of \mathcal{A} so that we can select the best performing learner from $\mathcal{A} \cup \mathcal{V}$. In this paper we extend the functionality of the landmarking method in (Ler et al., 2004a) to consider majority voting committees (Dietterich, 1997) over a set of algorithms \mathcal{A} . Some preliminary discussion of this extension appears in (Authors, 2005).

2 Landmarking

Meta-learning, where a set of easily computable dataset characteristics (meta-features) are mapped to performance predictions for algorithms, has been used as a method for algorithm selection – see (Vilalta and Drissi, 2002) for a survey. More recently, landmarks were proposed as meta-features (Fürnkranz and Petrak, 2001; Pfahringer et al., 2000). Let a *landmarker element* be an algorithm whose performance is utilised as a dataset characteristic; and a

landmarker be a function over the performance of a set of landmarker elements, whose output resembles the performance (e.g. accuracy) of one specific learning algorithm. Then the process of landmarking is simply the process of generating a set of algorithm estimators (landmarkers) for some given set of candidate algorithms (Ler et al., 2004a). In this paper we select landmarkers according to the following criteria proposed in (Ler et al., 2004a):

- **Correlativity:** Each landmarker should resemble the algorithm being landmarked. Changes in the performance of the candidate algorithm should be highly correlated with changes in the landmarker.
- **Efficiency:** Given a set of algorithms running the selected landmarker elements should take significantly less time than running the full set of algorithms.

The correlativity of two learning algorithms can be measure by the correlation coefficient r (Wickens, 1995), where:

$$r = \cos \angle(a, l) = \frac{a \cdot l}{\|a\| \cdot \|l\|} = \frac{\text{cov}(a, l)}{\sigma_a \sigma_l}$$

Intuitively, two algorithms whose performance patterns are similar over multiple datasets will be closer to each other in a performance space. In the context of landmarking, a is interpreted as a dependant algorithm and l as a set of independents (the landmarker elements). This is depicted in Figure 1.

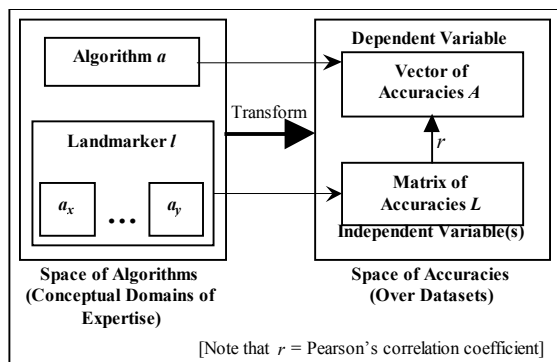


Figure 1. Operationalising $a.l$ – an example

When interpreting a as a dependent and l as a set of independents, we observe that selecting l is essentially the subset selection problem. Many heuristics for subset selection have been proposed (e.g. r^2 , adjusted r^2 , Mallows's C_p , AIC , BIC – see (Miller, 2002)). Any of these heuristics can be used to select correlated landmarkers. However, the efficiency criterion also needs to be satisfied. Similarly to (Ler et al., 2004a) we use a utility measurement for efficiency that calculates the reduction in computational cost afforded by running only a subset of the algorithms as compared to running the entire set.

3 Landmarking Majority Voting Ensembles

An ensemble can be more accurate than its component algorithms only if those component algorithms are diverse – i.e. they do not make the same mistakes (Hanson and Salamon, 1990). More precisely, if the error rates of the component algorithms are each less than 0.5, and if the errors are independent, then the probability that the majority vote is wrong is the area under the binomial distribution where more than half algorithms are wrong. This is further emphasised in (Krogh and Vedelsby, 1995), where it is shown that the generalisation error of an ensemble is equal to the average error of ensemble members, minus the average variance (or ambiguity) among them. A survey of ensemble techniques can be found in (Dietterich, 1997).

Given a set of heterogeneous algorithms A , instead of simply returning the $a_i \in A$ with the best estimated performance, we want to select the best voting ensemble. Formally we want to consider all possible majority voting ensembles over A , namely $V = \{v_1, \dots, v_m\}$ (where V is simply the powerset of A with the singletons and empty set removed). We denote by v_i components the subset of A that comprises v_i . Our aim is to use landmarking to allow us to return the best algorithm/voter from $A \cup V$ for a new dataset.

Intuitively, we assume that if two algorithms have uncorrelated or inversely correlated performance (i.e. their patterns of performance are dissimilar), the probability that they are diverse is higher (since there is more evidence that they work differently); and hence, would form a better ensemble. On the other hand, if two algorithms have correlated accuracies, then the likelihood is that they will not be diverse, and consequently, would form a poor ensemble. When performing landmarking, we assume that there exist clusters of algorithms that each correspond to a particular (similar) pattern of performance. Thus, by evaluating representatives from these clusters, we are able to estimate the performance of the remaining algorithms in those clusters. Analogously, when forming voters comprised of algorithm components from different clusters, the possibility for good ensembles is greater.

By considering the all the voters, one would typically expect to incur additional computational cost in the landmarking process, since given $|A|$, we now want to consider $2^{|A|} - |A| - 1$ more possible options. However, by exploiting the relationship between A and V , we are able to revise the landmarking method described in (Ler et al., 2004a) to estimate the performance of each algorithm and voter with little or no additional computational cost.

The most straightforward method of extending the landmarker generation algorithm to encompass majority voting ensembles is to simply add each possible voter to the list of algorithms to be estimated. Thus, given a new dataset s_{new} , we would evaluate $L' \subset A$ and use those performance values to evaluate $A \cup V$. One potential problem with this approach is that the ratio of independents (i.e. $|L'|$) to dependents (i.e. $|A \cup V| - |L'|$) may be too small. For example, given $|A| = 10$, $|A \cup V| = 2^{10} - 1 = 1023$; thus, no matter what subset of A we run, the ratio will be no better than

Input: k algorithms $A = \{a_1, \dots, a_k\}$, the voters over A , namely $V = \{v_1, \dots, v_t\}$ (where $t = 2^{|A|} - |A| - 1$) and $\mathbb{P} = \{P_1, \dots, P_k\} \cup \{P_{k+1}, \dots, P_{k+t}\}$, where each P_i is the vector of performance values of a_i on a corpus of datasets $S = \{s_1, \dots, s_n\}$ if $i \leq k$, and the vector of performance values of v_{i-k} on S if $i > k$.
Output: the chosen landmarking elements $L' \cup V_L$, (where $L' \subset A$, and V_L is the subset of V with each $v_i \in V_L$, $v_i.components \subset L'$), and $\mathcal{C} = \{C_j \mid \text{each } C_j \text{ holds the coefficients for the linear regression function whose dependent is the } i\text{-th algorithm/voter in } A \cup V, \text{ and independent(s) are } L' \cup V_L\}$.

- Let the powerset of A (excluding ϕ and A itself) be $\mathbb{L} = \{L_1, \dots, L_m\}$, where $m = 2^k - 2$, and
- V_{L_i} is the subset of V such that for each $v_j \in V_{L_i}$, $v_j.components \subset L_i$.
- $r^2(a_j, L_i)$ and $coefficients(a_j, L_i)$ be the r^2 value and coefficients for the least squares linear regression function with dependent a_j and independent(s) $\{a_x \mid a_x \in L_i\}$ respectively (note that each regression function is calculated using the relevant P – i.e. corresponding to a_j and L_i), and
- $efficiency\ gained$ is $((eff(A) - eff(L_i)) / eff(A))$, where $eff(L_i)$ is the computational cost of each L_i , and $eff(A)$ is the computational cost of A .

Majority Voting all subsets landmarker generation algorithm:

- [1] For each $L_i \in \mathbb{L}$:
- [2] For each $x_j \in A \cup V$:
- [3] If $x_j \notin L_i \cup V_{L_i}$ then:
- [4] Find $coefficients(x_j, L_i \cup V_{L_i})$ and $r^2(x_j, L_i \cup V_{L_i})$ using $\{P_k \mid \text{for each } x_k \in x_j \cup L_i \cup V_{L_i}\}$.
- [5] Else // x_j is an independent, and its performance need not be estimated
- [6] $coefficients(x_j, L_i \cup V_{L_i})$ is a vector of 0s except for the coefficient for x_j , which is 1.
- [7] $r^2(x_j, L_i \cup V_{L_i}) = 1$.
- [8a] Find $mean_Heuristic(L_i) = [\sum_{\forall x_k \in A \cup V} r^2(x_k, L_i \cup V_{L_i})] / |A \cup V|$. // Heuristic 1: r^2 only criterion
- [8b] Find $mean_Heuristic(L_i) = ((eff(A) - eff(L_i)) / eff(A)) + [\sum_{\forall x_k \in A \cup V} r^2(x_k, L_i \cup V_{L_i})] / |A \cup V|$. // Heuristic 2: $efficiency\ gained + r^2$ criterion
- [9] $\mathcal{C} = \{C_j \mid \forall x_j \in A \cup V, C_j = coefficients(x_j, L' \cup V_L), L' = \arg \max_{\forall L_i \in \mathbb{L}} mean_Heuristic(L_i)\}$.

Figure 3. The majority voting all subsets landmarker generation algorithm

1:100! This may or may not be a problem depending on the number of distinctive clusters of performance patterns that arise from the algorithms/voters in $A \cup V$. If the number of clusters is small (i.e. $\leq |A|$), then it is likely that the generated landmarkers will work. However it would be better to evaluate some subset of V as well as A .

Assuming the additional computational cost required to generate majority voting classifications – exclusive of the computational cost of all the component algorithms for that voter – is negligible, then if L' is evaluated, the performance of $2^{|L'|} - |L'| - 1$ of the voters from V can correspondingly be computed at negligible cost. Thus, for each potential subset of A we consider (that is consider as independents to be evaluated), we may compute the additional independents that correspond to the voting ensembles over that subset of A . Going back to our example with $|A| = 10$; suppose we choose to evaluate L' , where $|L'| = |A|/2 = 5$, then the ratio of dependents is $2^5 - 1 : (2^{10} - 1) - (2^5 - 1)$, or 1:32; without the additional voter independents, this ratio is $5 : (2^{10} - 1) - 5$, or 1:203.6.

For any given subset L' of A , the set of possible voters over L' is denoted by $V_{L'} = \{v_i \mid v_i \in V \text{ and } v_i.components \subset L'\}$. Having selected L' , we might consider if all or only a subset of $V_{L'}$ should be used as independents. Essentially, this choice ends up being a trade-off between the computational complexity of the landmarker generation algorithm, and its expressiveness (or rather, the size of the search space it can cover when searching for landmarkers – i.e. searching for linear regression functions). However, we postpone answering this particular question within this paper and instead utilise the entire subset $V_{L'}$ when considering the independents L' .

The majority voting landmarker generator algorithm is described in Figure 3.

4 Experimental Setup

For our experiments we use 80 classification datasets (i.e. \mathcal{S}) randomly chosen from the UCI repository (Blake and Merz, 1998) and the following 8 classification learning algorithms (i.e. A^*) from WEKA (Witten and Frank, 2000): naïve Bayes, k -nearest neighbour (with $k = 5$), polynomial and RBF support vector machines, the decision tree J4.8 (a WEKA implementation of C4.5), decision forest, decision stump and Ripper). In each trial, the majority voting landmarker generation algorithm is evaluated on a subset of 5 (of the 8) algorithms (i.e. A) and the corresponding set of $2^5 - 5 - 1 = 26$ voters over A (i.e. V) using the leave-one-out cross-validation approach; thus, there are a total of $8^8 C_5$ trials.

In each fold (k) within a trial, 79 datasets (\mathcal{S}_k) are used to generate two sets of landmarkers – the first on heuristic 1: r^2 and the second on heuristic 2: $r^2 + efficiency\ gained$. Each set of generated landmarkers indicates two subsets of $A \cup V$: (1) the algorithms that must be evaluated (independents: $L' \cup V_L$), and (2) those which will be estimated (dependents: $\{A \cup V\} \setminus \{L' \cup V_L\}$). On the dataset left out (s_{new}), the accuracies of the independents ($L' \cup V_L$) are evaluated via the landmarkers (i.e. regression functions) generated and used to estimate the independents ($\{A \cup V\} \setminus \{L' \cup V_L\}$). To evaluate the accuracy of each component algorithm ($a_i \in A$) and voter ($v_j \in V$) on each dataset, ten-fold cross-validation was employed.

Either variant of the algorithm described in Section 3 (i.e. the variant using line 8a or 8b) will return one set of landmarkers based all possible $L' \subset A$. We have modified the

algorithm so that only landmarker element subsets (i.e. $|\mathbf{L}'|$) of the same size will be considered. Accordingly, in each trial the results for each of the 4 set sizes of \mathbf{L}' ($1 \leq |\mathbf{L}'| \leq 4$, since $|\mathcal{A}| = 5$) are reported. Thus, on each fold of each trial, for each of the two heuristics, we generate 4 sets of landmarkers (one for each $|\mathbf{L}'|$) and estimate 4 sets of accuracies (for each $\mathbf{a}_i \in \mathcal{A}$ and $\mathbf{v}_j \in \mathcal{V}$) over the UCI datasets left-out. The following three measurements are then taken:

Efficiency gained (EG): The percentage of computation saved by employing a landmarker. This is the portion of the computational time saved by running only the landmarker elements (\mathbf{L}') of the landmarker in question as opposed to all algorithms (\mathcal{A}) – i.e. $(\text{eff}(\mathcal{A}) - \text{eff}(\mathbf{L}')) / \text{eff}(\mathcal{A})$.

Rank order correlation (r_s): Spearman’s rank order correlation coefficient r_s . This determines the correlation between: (1) the rank order of the accuracies estimated via the landmarkers, and (2) the rank order of the accuracies evaluated via ten-fold cross-validation.

Algorithm-pair ordering (AO): The percentage of algorithm/voter pairs from $\mathcal{A} \cup \mathcal{V}$ whose orderings are correct. An algorithm-pair has a correct ordering if their estimated (via the landmarkers) and evaluated (via ten-fold cross-validation) accuracies sit in the same order. Note that there are ${}^3\text{C}_2 = 465$ algorithms/voter pairings over a basis of 5 algorithms. However, when all 5 algorithms and the consequent 26 voters are evaluated (i.e. used as independents), no estimation is required, and we are simply performing ten-fold cross-validation. Accordingly, for some chosen \mathbf{L}' , $h = 2^{|\mathbf{L}'|} - 1$ algorithms/voters are evaluated, not estimated; thus (for $|\mathbf{L}'| > 1$) ${}^h\text{C}_2$ algorithm pairs will mimic the ten-fold cross-validation orderings. The *Assured AO* is the percentage of AO associated with pairings that are guaranteed to be correct – see Table 1.

$ \mathbf{L}' $ ($ \mathcal{V}_{\mathbf{L}'} $)	1 (0)	2 (1)	3 (4)	4 (11)	5 (26)
Assured AO (%)	0.0	0.7	4.5	22.6	100.0

Table 1. The AO percentage assured to be correct for each $|\mathbf{L}'|$

5 Results

The expected distribution of winning algorithms/voters based on each $\mathcal{A} \subset \mathcal{A}^*$ is given in Table 2.

No. of components	1	2	3	4	5
E[Freq. of wins]	0.39	0.11	0.38	0.10	0.02
Dev[Freq. of wins]	0.07	0.02	0.05	0.03	0.01

Table 2. The frequency of winners in terms of learner components

Based on the ten-fold cross-validation accuracy of each algorithm in \mathcal{A}^* on each dataset in \mathcal{S} , Table 2 reports the average frequency and standard deviation of winners (for each component size) over the various permutations of \mathcal{A} . This table essentially tells us that roughly 39% of the time, the actual winner is an individual algorithm and for the re-

maining 61% of the time, the winner is a voter (e.g. 2% of the winners are voters over 5 component algorithms).

Tables 3 and 4 present the results from our experiments for the variants based on the r^2 and $r_2 + \text{efficiency gained}$ heuristics respectively. These tables each report the mean AO (algorithm-pair orderings) and r_s (rank correlation) achieved over: (i) the individual algorithms (\mathcal{A}) only, (ii) the voters (\mathcal{V}) only and (iii) combinations of both ($\mathcal{A} \cup \mathcal{V}$). Also, as the computational overhead of computing the accuracy on a voter is considered to be negligible (after having already accounted for the cost of its components), the EG (efficiency gained) reported is thus only over individual algorithms.

Note that these mean values are themselves computed over the mean (per fold) values attained over each of the 56 permutations of \mathcal{A} . Accordingly, the companion standard deviations for the results given in Tables 3 and 4 are reported in Tables 5 and 6.

$ \mathbf{L}' $ ($ \mathcal{V}_{\mathbf{L}'} $)	1(0)	2(1)	3(4)	4(11)	5(26)*
E[AO](\mathcal{A})	0.657	0.707	0.801	0.902	1.000
E[AO](\mathcal{V})	0.714	0.727	0.764	0.831	1.000
E[AO]($\mathcal{A} \cup \mathcal{V}$)	0.707	0.726	0.771	0.844	1.000
E[r_s](\mathcal{A})	0.418	0.517	0.691	0.849	1.000
E[r_s](\mathcal{V})	0.558	0.593	0.678	0.792	1.000
E[r_s]($\mathcal{A} \cup \mathcal{V}$)	0.539	0.589	0.687	0.807	1.000
E[EG]	0.794	0.497	0.337	0.182	0.000

* Corresponds to 10-fold cross-validation.

Table 3. The mean AO, r_s and EG results from the landmarkers generated using the r^2 only variant

$ \mathbf{L}' $ ($ \mathcal{V}_{\mathbf{L}'} $)	1(0)	2(1)	3(4)	4(11)	5(26)*
E[AO](\mathcal{A})	0.658	0.720	0.803	0.894	1.000
E[AO](\mathcal{V})	0.715	0.727	0.760	0.824	1.000
E[AO]($\mathcal{A} \cup \mathcal{V}$)	0.707	0.727	0.768	0.837	1.000
E[r_s](\mathcal{A})	0.419	0.532	0.686	0.831	1.000
E[r_s](\mathcal{V})	0.558	0.594	0.670	0.777	1.000
E[r_s]($\mathcal{A} \cup \mathcal{V}$)	0.540	0.591	0.682	0.793	1.000
E[EG]	0.800	0.672	0.522	0.243	0.000

* Corresponds to 10-fold cross-validation.

Table 4. The mean AO, r_s and EG results from the landmarkers generated using the $r^2 + \text{efficiency gained}$ variant

The results from our experiments show that even when only utilising a single landmarker element, the chosen set of landmarkers (i.e. column with $|\mathbf{L}'| = 1$) is still able to produce a reasonable result, and achieves significant improvement in terms of efficiency gained. As expected, when $|\mathbf{L}'|$ increases, so too do AO and r_s , while EG decreases (i.e. as $|\mathbf{L}'| \rightarrow |\mathcal{A}|$, the results approach those of ten-fold cross-validation). Correspondingly, the accuracy results (i.e. AO and r_s) on both heuristics are very similar; and not surprisingly, the EG results are better on the $r^2 + \text{efficiency gained}$

variant, as it tries harder to find more efficient landmarker elements.

Furthermore, the deviation of the accuracy results (for AO and r_s) across the various permutations is not very high, showing that the landmarking method can work across different permutations of candidate algorithms. However, the efficiency gained result (EG) does have a fairly high deviation; this is because the variability of efficiency across algorithms tends to be much more predominant than the variability across the correlation of accuracy patterns. Thus, when certain subsets of the more efficient algorithms are left out in some of the trials (i.e. permutations of A^*) their absence in terms of efficiency gained (i.e. EG) is more noticeable. Accordingly, the expected variance of the (normalised) computational cost of each algorithm in A^* (across \mathcal{S}), is 0.1164, whereas, the variance across the correlation of each pair of accuracy patterns is 0.0148.

$ L^* (V_{L^*})$	1(0)	2(1)	3(4)	4(11)
Dev [AO](A)	0.033	0.028	0.020	0.015
Dev [AO](V)	0.016	0.014	0.013	0.015
Dev [AO]($A \cup V$)	0.016	0.016	0.014	0.014
Dev [r_s](A)	0.087	0.066	0.041	0.032
Dev [r_s](V)	0.036	0.030	0.028	0.029
Dev [r_s]($A \cup V$)	0.037	0.035	0.030	0.028
Dev [EG]	0.069	0.153	0.115	0.020

Table 5. The standard deviation for the AO , r_s and EG results from the landmarkers generated using the r^2 only variant

$ L^* (V_{L^*})$	1(0)	2(1)	3(4)	4(11)
Dev [AO](A)	0.032	0.025	0.021	0.020
Dev [AO](V)	0.016	0.012	0.015	0.018
Dev [AO]($A \cup V$)	0.016	0.013	0.016	0.017
Dev [r_s](A)	0.086	0.057	0.042	0.042
Dev [r_s](V)	0.036	0.026	0.032	0.034
Dev [r_s]($A \cup V$)	0.037	0.027	0.033	0.033
Dev [EG]	0.069	0.107	0.129	0.100

Table 6. The standard deviation for the AO , r_s and EG results from the landmarkers generated using the $r^2 + \text{efficiency gained}$ variant

An interesting observation is that when only one algorithm is evaluated, voters are estimated more accurately (5-6% better) than individual algorithms (with either variant). This may be because a single pattern of performance is insufficient to represent the number of variant clusters of performance patterns among the algorithms/voters. However, the rate of improvement on AO and r_s is less for V and $A \cup V$ as compared to A ; and consequently, for $|L^*| > 1$, the AO values for A tend to be higher.

Finally, a test was also conducted to validate the claim made in Section 3, that the pattern of performance of a pair of algorithms is inversely correlated to the diversity of the same pair of algorithms. For each pair from algorithms from A^* , the following were computed:

- Expected diversity: This is the average disagreement measure (Skalak, 1996; Ho, 1998) over each dataset in \mathcal{S} , for the pair of algorithms in question.
- Accuracy correlation: This is the correlation (i.e. Pearson's r) between the accuracies observed over the datasets in \mathcal{S} , for the pair of algorithms in question.

With $|A^*| = 8$, the above were computed for ${}^8C_2 = 28$ pairs of algorithms. The linear correlation (again Pearson's r) was then computed between the set of expected diversity values and accuracy correlation values. The resultant correlation coefficient, $r = -0.9506$ (with an F-value of 243.73, and p-value of 0), strongly suggesting that the claim is true; i.e. that the higher the positive correlation between the patterns of accuracy of two algorithms, the lower the diversity between those two algorithms.

6 Conclusion

In this paper we propose a new approach for estimating the performance of voting ensembles using landmarking. It uses the performance values of a subset of single algorithms and ensembles of these algorithms (called landmarking elements) to estimate the performance of the remaining set of single algorithms and ensembles. The experimental results show that even when a single landmarking element is used the percentage of correct algorithm-pair orderings is about 70%, a very promising result given a corresponding efficiency gain of 80%. As the number of algorithms that are evaluated increases, the accuracy of the landmarkers also increases. We also conducted statistical tests to show a link between the correlation of performance and diversity of a pair of algorithms. The results show that the higher the positive correlation between the patterns of accuracy of two algorithms, the lower the diversity between them.

As future work, we would like to study the link between the correlation of algorithm/voter accuracies and their diversity in greater detail, and use those results to form better ensembles even more efficiently. A landmarking extension that considers stacking systems over the given set of algorithms is also underway.

Acknowledgments

The authors from the University of Sydney would like to acknowledge the support of the Smart Internet Technology CRC in this research.

References

- [Authors, 2005] Estimating the Performance of Heterogeneous Majority Voting Ensembles via Landmarking. *Submitted to the 6th International Workshop on Multiple Classifier Systems*, 2005.
- [Blake and Merz, 1998] Blake, C., and Merz, C. UCI repository of machine learning databases. *University of California, Irvine, Department of Information and Computer Sciences*, 1998.

- [Crawford et al., 2002] Crawford, E., Koprinska, I., and Patrick, J. A multi-learner approach to e-mail classification. In *Proceedings of the 7th Australasian Document Computing Symposium*, 2002.
- [Dietterich, 1997] Dietterich, T. Machine-learning research: four current directions. *AI Magazine*, 18(4): 97-136, 1997.
- [Fürnkranz and Petrak, 2001] Fürnkranz, J., and Petrak, J. An evaluation of landmarking variants. In *Proceedings of the 2001 European Conference on Machine Learning, Workshop on Integrating Aspects of Data mining, decision support and Meta-learning*, pages 57-68, 2001.
- [Giraud-Carrier et al., 2004] Giraud-Carrier, C., Vilalta, R., and Brazdil, P. Introduction to the special issue on meta-learning. *Machine Learning*, 54(3):187-193, 2004.
- [Hanson and Salamon, 1990] Hanson, L., and Salamon, P. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:993-1001, 1990.
- [Krogh and Vedelsby, 1995] Krogh, A., and Vedelsby, J. Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems*, 7: 231-238, 1995.
- [Ler et al., 2005] Ler, D., Koprinska, I., and Chawla, S. A Hill-climbing Landmarker Generation Algorithm Based on Efficiency and Correlativity Criteria. In *Proceedings of the 18th International FLAIRS Conference, Machine Learning Track* (in press), 2005.
- [Ler et al., 2004a] Ler, D., Koprinska, I., and Chawla, S. A Landmarker Selection Algorithm Based on Correlation and Efficiency Criteria. In *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, pages 296-306, 2004.
- [Ler et al., 2004b] Ler, D., Koprinska, I., and Chawla, S. Comparisons between Heuristics Based on Correlativity and Efficiency for Landmarker Generation. In *Proceedings of the 4th International Conference on Hybrid Intelligent Systems* (in press), 2004.
- [Michie et al., 1994] Michie, D., Spiegelhalter, D., and Taylor, C. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.
- [Miller, 2002] Miller, A. *Subset Selection in Regression, 2nd Edition*. Chapman and Hall/CRC, 2002.
- [Pfahringer et al., 2000] Pfahringer, B., Bensusan, H., and Giraud-Carrier, C. Meta-learning by landmarking various learning algorithms. In *Proceedings of the 17th International Conference on Machine Learning*, 743-750, 2000.
- [Schaffer, 1997] Schaffer, C. Technical note: selecting a classification method by cross-validation. *Machine Learning*, 13(1): 135-143, 1993.
- [Vilalta and Drissi, 2002] Vilalta, R., and Drissi, Y. A perspective view and survey of meta-learning. *Journal of Artificial Intelligence Review*, 18(2): 77-95, 2002.
- [Wickens, 1995] Wickens, T. *The Geometry of Multivariate Statistics*. LEA Publishers, 1995.
- [Witten and Frank, 2000] Witten, I., and Frank, E. *Data Mining: Practical Machine Learning Tools with Java Implementations*. Morgan Kaufmann, 2000.
- [Wolpert, 1996a] Wolpert, D. The existence of a priori distinctions between learning algorithms. *Neural Computation*, 8(7): 1391-1420, 1996.
- [Wolpert, 1996b] Wolpert, D. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7): 1341-1390, 1996.