The Problem of Missing Values in Decision Tree Grafting

Geoffrey I. Webb

School of Computing and Mathematics
Deakin University
Geelong, Vic, 3217, Australia.

Abstract. Decision tree grafting adds nodes to inferred decision trees. Previous research has demonstrated that appropriate grafting techniques can improve predictive accuracy across a wide cross-selection of domains. However, previous decision tree grafting systems are demonstrated to have a serious deficiency for some data sets containing missing values. This problem arises due to the method for handling missing values employed by C4.5, in which the grafting systems have been embedded. This paper provides an explanation of and solution to the problem. Experimental evidence is presented of the efficacy of this solution.

Keywords: Grafting, Decision Tree Learning; Missing Values

1 Introduction

Grafting is a technique for postprocessing inferred decision trees that has been demonstrated to increase predictive accuracy across a wide cross-selection of learning tasks (Webb, 1997). Grafting traverses the decision tree identifying new branches that can be added profitably to the tree. The opportunity to identify useful additional branches arises due to the top-down divide-and-conquer strategies used in conventional decision tree induction algorithms. When evaluating potential new branches, these algorithms do not consider evidence from outside the region r of the instance space covered by the node currently being processed. By considering evidence about sub-regions of r provided by examples outside r, grafting has been demonstrated to identify additional branches that significantly increase predictive accuracy. Indeed, over a wide cross-selection of learning tasks, grafting has been demonstrated to increase predictive accuracy as frequently as pruning (Webb, 1997). Further, pruning and grafting are complementary. Pruning then grafting outperforms either pruning or grafting alone (Webb, 1997). (Grafting then pruning is inappropriate as pruning will remove all nodes added by grafting as these new nodes will have little support in terms of the local evidence that pruning can consider.)

This paper addresses a problem encountered by previous grafting systems (Webb, 1996, 1997) when they are applied to some data sets containing missing values. Put simply, these systems, which are implemented as extensions to C4.5 (Quinlan, 1993), can add branches to a node which cause items classified by that branch, that have missing values for the attribute tested at the branch, to be passed down the new branch, rather than defaulting to the branch associated with the original node as intended.

For data sets with many missing values, this can result in large increases in predictive error. A simple solution to this problem, investigated in this paper, is to prevent grafting on attributes for which there are missing values at the current node. This measure is shown to prevent the extreme increases in predictive error previously identified without general negative side-effects. It also offers the additional benefit of substantially reducing the size of the inferred trees.

2 Decision Tree Grafting

Machine learning can be viewed as a process of partitioning an instance space. The instance space is the multi-dimensional geometric space formed by projecting axes for each attribute in the learning task. Top-down decision tree induction algorithms, such as C4.5 (Quinlan, 1993) and CART (Breiman, Friedman, Olshen, & Stone, 1984), recursively partition the instance space until consideration of the training examples within a partition fails to support further partitioning. Each leaf of the inferred decision tree corresponds to a partition. Every item that falls within a partition is classified as belonging to the class with which the corresponding leaf is labeled (the class that dominates the partition within the training set).

During learning, when considering potential new partitions within an existing partition p, only training examples within p are considered. As a consequence, areas of the instance space that are not occupied by any training examples are allocated to partitions purely as a side-effect of forming partitions for areas that are occupied by training examples. This can be undesirable, as illustrated by an example drawn from Webb (1997), presented in Figs. 1 and 2. Figure 1 presents a simple two attribute learning task projected onto a two dimensional instance space. Objects belong to three classes, $*$, $\bullet$, and $\diamond$. The reader is invited to consider the most likely class for an unclassified object, labeled ?, with attribute values $A = 6, B = 1$. Informal surveys suggest that the most common intuition is that such an object has highest probability of belonging to class $\diamond$. In this context it is illuminating to consider the partitions that correspond to the decision tree formed by C4.5 for this learning task. This is presented in Fig. 2. It can be seen that C4.5 forms a partition that assigns class $*$ to the unclassified object. This occurs because, once it has sufficiently partitioned the instance space to adequately accommodate all objects in the training set, no consideration is given to areas of the instance space, such as that in question, that are not occupied by training examples.

Decision tree grafting traverses an inferred decision tree looking for areas of the instance space such as this, that are not occupied by training examples (or are occupied only by training examples misclassified by the current partition). It explores the ancestors of a leaf at which such areas are found, seeking alternative cuts that could have been imposed at those ancestors that appear likely to

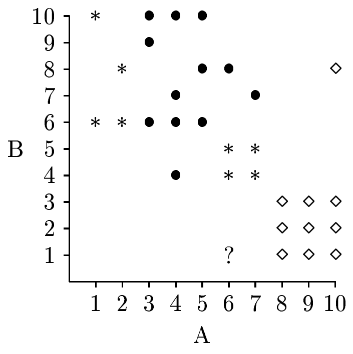
Fig. 1. Example instance space

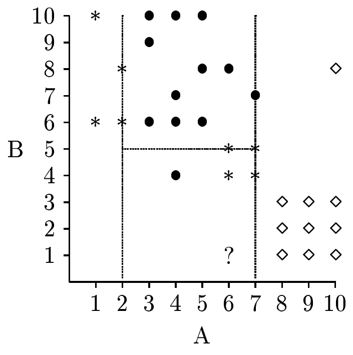
Fig. 2. Example instance space as partitioned by C4.5

provide better predictive accuracy in that region than the leaf in question. Such cuts are imposed on the tree at the leaf. Extensive comparison (Webb, 1996, 1997) has shown that, in general, grafting improves the predictive accuracy of both pruned and unpruned decision trees learned by C4.5.

It has been hypothesised (Webb, 1997) that grafting has a similar effect to learning ensembles of classifiers (Ali, Brunk, & Pazzani, 1994; Breiman, 1996; Dietterich & Bakiri, 1994; Freund & Schapire, 1995; Kwok & Carter, 1990; Oliver & Hand, 1995; Nock & Gascuel, 1995; Schapire, 1990; Wolpert, 1992), in that both consider more evidence about the best classifications for objects in regions of the instance space that are poorly represented in the training data. Both aproaches form more complex partitions of the instance space than conventional learning. However, while these partitions are implicit in an ensemble, formed by resolution of conflicting predictions from each classifier, they are explicit with grafting, which directly represents the partitions with a single decision tree. This can be expected to make the classifiers formed by grafting more readily comprehensible than those formed using ensembles of classifiers. On the other hand, however, current grafting techniques are not as effective at reducing prediction error as good ensemble induction techniques.

3 Grafting with Missing Values

This paper addresses a problem that was identified when previous grafting techniques were extended to accommodate discrete valued attributes. For the annealing data set, grafting unpruned trees increased the average predictive error from a cross-validation experiment from 5.4% to 84.6%. For pruned trees the average predictive error was increased from 8.0% to 44.1%. In this data set every object has values missing for at least some attributes and for most objects more attribute values are missing than are known.

To understand how such missing values can affect grafting to C4.5 trees, it is necessary to understand how C4.5 handles missing values. There are two distinct stages at which missing values must be accommodated, during induction and during application of the inferred tree for classification. As grafting is implemented as a postprocessor that is applied after the initial decision tree is inferred the manner in which missing values are handled by C4.5 during induction is not of significance here. How they are handled during classification can have profound impact, however, as this determines how the structures created by grafting are subsequently interpreted.

During classification, objects with missing values on an attribute are passed down all branches below a test on that attribute. All objects are assigned an initial weight of 1. When a missing value causes an object to be passed down multiple branches, it is passed down each branch with a diminished weight. The weight of an object with a missing value for attribute a when passed down a branch b below a test on a is multiplied by m/n when m is the number of training objects with known values for a that pass down b and n is the number of training objects with known values for a at the node from which b descends. All leaves that are reached then vote for the class to be assigned to the object, the vote of a leaf equaling the weight of the object when it reaches the leaf.

This process can cause a number of pitfalls for decision tree grafting. Grafting is predicated on the principle that unless there is evidence that an alternative class should be assigned to an object, that the class assigned by the original decision tree should be preferred. Thus, the default action at any grafted branch should be to pass the object toward the original leaf. However, consider the effect if a test on an attribute a is grafted onto a leaf l for class c at which all training objects belonging to c have missing values for a. If there are substantial numbers of such objects, it is likely that when classifying future objects, more such will be encountered. However, all such future objects will be misclassified, as the technique for classifying objects with missing values will strongly weight these objects towards the branches for which there are objects with known values, none of which will lead to the original leaf! This will happen even if there are large numbers of training objects for class c that reach l, so long as none have known values for a. This is clearly undesirable.

Two potential solutions to this problem that might be considered are:

1. modifying C4.5's method of handling missing values for grafted branches so that the greatest weight is directed toward the leaf for the original class; and
2. prohibiting grafting on an attribute a at leaf l for class c if the number of training objects with known values for a for class c at l is not greater than the number of all other classes.

However, neither of these measures will solve the problem as it is possible that there will be many attributes with missing values, and that even if each branch gives greatest weighting on the path toward the original leaf, the gradual diminution of weight over many such branches may eventually lead to reclassification of an object. A further dimension is added to the problem when it is considered that an object being classified might have missing values for attributes tested at ancestors of the leaf in question, so that diminution of the weight assigned to it at the current leaf may result in reclassification (due to the relative numbers of votes for each class from other branches of the tree) even if the change in weight at the leaf in question is relatively small.

A simple solution to this problem is to prohibit grafting at a leaf l of a cut on an attribute a if any training object that reaches l has a missing value for a. This:

1. prevents the direct problem of undue weight being assigned to grafted branches; and
2. reduces the likelihood of objects with missing values having their weights adversely diminished, by reducing the likelihood of such objects being encountered during classification.

Applying this constraint to the C4.5x grafting algorithm reduced predictive error on the annealing data set from 84.6% to 5.5% for unpruned trees and from 44.1% to 7.5% for pruned trees, suggesting that it has provided an effective solution to the identified problem. Section 5 evaluates the effect of this measure on a wide range of learning tasks from the UCI Repository of Machine Learning Databases (Merz & Murphy, 1998). Before presenting this evaluation, however, it is necessary to describe the implementation of grafting that the new measure is embedded within. This is done in the next section.

4 C4.5x3

The problem of missing values was first identified during the extension of the C4.5x (Webb, 1997) grafting system to perform grafting on discrete valued attributes. Previous versions of C4.5x (Webb, 1996, 1997) had performed grafting only on continuous attributes.

C4.5x acts as a postprocessor for C4.5 decision trees. It traverses the inferred decision tree. At each leaf l, it explores ancestors of l, looking for tests that could have been imposed at those ancestors that project across the partition of the instance space for l, and for which the available evidence suggests that the expected accuracy within that projection for a specific classification is higher than the expected accuracy within the existing leaf. All such tests are grafted onto the existing node in order from highest expected accuracy to lowest.

The expected accuracy for a test is evaluated using a Laplacian measure (Niblett & Bratko, 1986). The expected accuracy equals

$$\frac{m+1}{n+2} \tag{1}$$

where m is the number of training objects belonging to the specified class that pass a test and n is the total number of training objects that pass the test. This measure favours tests with high resubstitution accuracy over those with low resubstitution accuracy and also favours tests supported by many training objects over those for which few training objects provide support.

As well as requiring a higher accuracy estimate, a new test is only expected to produce higher accuracy than the existing leaf if a binomial sign test reveals that it is statistically unlikely that the class distribution for the objects used as evidence for imposing the test would have been obtained if the underlying class distribution reflected the estimated error rate for the original leaf. A significance level of 0.05 is used for the purposes of this test.

For continuous attributes, tests of the form $a \leq v$ or $a > v$ are considered, for all values of attribute a at the ancestor node that could reach the target leaf l, with the constraint that no such test may reclassify a training object that is correctly classified at l. For discrete attributes, tests of the form $a = v$ are considered. Such tests are not considered if an ancestor of l has a test on a as in this case only a single value of a may reach l and hence a test on a could not form a new partition of non-zero volume. (It is assumed that grafting will be applied in the context of trees learned without subseting. It would, however, be straight forward to extend the system to accommodate such tests in which case only values for the attribute that can reach l should be considered.) The alternative branches associated with grafting such single value tests are formed using C4.5's subseting facility.

The resulting algorithm is presented in Appendix A.

5 Evaluation

Cross-validation experiments were performed on 18 data sets representing a wide cross-selection of those in the UCI repository that contain missing values. These data sets are presented in Table 1. This table lists the numbers of cases, classes, continuous attributes, and discrete attributes, as well as the percentage of attribute values that are missing.

Ten 10-fold cross validation experiments were performed for each data set. Each 10-fold cross validation experiment involved dividing a data set into 10 partitions of as near as possible to equal size. For each partition, each treatment

Table 1. Description of data sets

Name	Cases	Classes	Contin.	Discr.	% Unknown
annealing	898	6	6	32	65
audio	226	23	—	69	2
autos	205	7	14	10	1
breast-slov	286	2	—	9	< 1
breast-wisc	699	2	9	—	< 1
cleveland-hd	303	2	6	7	< 1
crx	690	2	6	9	1
dis	3772	2	7	22	6
echocardiogram	74	2	5	1	5
hepatitis	155	2	6	13	6
horse-colic	368	2	8	13	25
hungarian-hd	294	2	6	7	20
hypo	3772	4	7	22	6
labor-neg	57	2	8	8	36
mushroom	8124	2	—	22	1
primary tumor	339	22	—	17	4
sick	3772	2	7	22	6
soybean large	683	19	—	35	3

was applied to learn a decision tree from the 9 remaining partitions. Each of these trees was then evaluated by application to the selected partition.

For each of the ten such experiments run for each data set, the data was randomly partitioned into different partitions. Results are presented in the form of mean values across the 100 runs for a data set resulting from the ten cross-validation experiments each comprising ten runs.

Table 2 presents the mean error for each data set. For pruned trees, prohibiting grafting on attributes with unknown values increases error for 4 domains and decreases it for 7 domains. A binomial sign test reveals that the probability of obtaining such a result by chance is 0.274, which does not enable us to conclude that this measure provides a general tendency to decrease error for pruned trees. For unpruned trees, prohibiting grafting on attributes with unknown values increases error for 4 data sets and decreases error for 10. A binomial sign test reveals that the probability of obtaining such an outcome by chance is 0.090 which is also not significant at the 0.05 level. However, given that the measure solves the initial problem (large increases in error resulting from grafting on unknown values), and, while not providing a clear general advantage over a wide variety of domains, certainly does not provide a general disadvantage, prohibiting grafting on attributes with unknown values appears desirable. It should also be noted that with only 18 data sets available for comparison, the power of the test statistic is low, meaning that the failure to obtain a significant result provides little evidence that there is not indeed a general underlying advantage to

Table 2. Summary of mean percentage error rates

	Pruned Trees		**Unpruned Trees**	
	No Unk	**Unk**	**No Unk**	**Unk**
annealing	7.5	44.1	5.4	84.6
audio	22.9	22.9	23.3	23.2
autos	18.3	18.7	16.6	16.9
breast-slov	26.8	26.8	30.2	30.3
breast-wisc	5.3	5.3	5.1	5.2
cleveland-hd	22.4	22.4	22.5	22.5
crx	14.4	14.7	16.0	16.3
dis	1.1	1.1	1.2	1.2
echocardiogram	30.7	30.6	28.4	28.7
hepatitis	18.4	18.6	18.2	19.0
horse-colic	15.5	16.4	18.3	19.3
hungarian-hd	20.6	20.5	22.0	21.6
hypo	0.5	0.6	0.6	0.6
labor-neg	21.0	20.5	20.8	20.5
mushroom	0.0	0.0	0.0	0.0
primary tumor	59.0	58.6	58.3	58.4
sick	1.4	1.4	1.5	1.4
soybean large	8.4	12.0	8.8	13.3
Win-loss summary		7/4		10/4
Win-loss p		0.274		0.090

the new technique.

A further issue that is relevant in some contexts is the relative complexity of two inferred classifiers. This is particularly an issue, when the inferred classifier is to be subject to human scrutiny. Table 3 presents the numbers of nodes produced by each technique.

The unknown values constraint limits the number of grafts that can be performed. Thus, it can never increase the complexity of a tree, only decrease it. It can be seen that some decrease in average tree size has been achieved for every data set. For some data sets this decrease is arguably negligible (audio, autos, breast-slov, breast-wisc, cleveland-hd, dis, echocardiogram, mushroom, and sick). For each of the remaining eight data sets, however, the decrease in the number of nodes exceeds 10%. For annealing, the average size of the pruned trees is decreased by 73% and for horse-colic it is reduced by 83%. This result strengthens the case for preferring the prohibition of grafting on attributes with unknown values.

6 Conclusions

Grafting has been demonstrated to increase predictive accuracy across a wide variety of learning tasks. However, the interaction of C4.5's mechanisms for missing

Table 3. Summary of mean number of nodes per tree

	Pruned Trees		**Unpruned Trees**	
	No Unk	**Unk**	**No Unk**	**Unk**
annealing	407.8	1527.2	684.5	2746.9
audio	91.8	94.3	138.8	145.0
autos	921.0	941.4	1206.8	1239.8
breast-slov	29.4	29.6	324.8	326.0
breast-wisc	123.1	123.8	245.5	249.3
cleveland-hd	184.3	184.6	352.3	352.8
crx	238.8	425.2	1171.3	1481.8
dis	180.3	183.1	730.9	744.6
echocardiogram	8.2	8.8	11.1	12.3
hepatitis	32.3	50.3	62.8	99.7
horse-colic	15.7	92.6	191.7	1680.8
hungarian-hd	23.9	27.4	158.6	190.8
hypo	305.6	341.7	436.4	494.2
labor-neg	10.6	19.0	26.7	53.3
mushroom	994.8	1014.3	994.8	1014.3
primary tumor	129.2	158.2	200.9	242.3
sick	595.8	626.6	846.7	898.1
soybean large	711.9	1243.5	1384.8	3448.1
Win-loss summary		18/0		18/0
Win-loss p		0.000		0.000

values with grafted branches led to extreme increases in error for a small number of domains with large numbers of missing values. This research has shown that the simple expedient of prohibiting grafting on attributes for which unknown values are present at a node overcomes this problem. This prohibition greatly reduces error on the domain for which the problem was initially diagnosed, annealing. It also reduces error in six other domains when grafting on pruned trees and ten other domains when grafting on unpruned trees. Small increases in error are evident only for four domains for each type of tree. A further benefit is a general decrease in the size of the inferred trees, in some cases by more than 80%. Given these benefits, a general prohibition against grafting on attributes with unknown values at a node appears warranted.

A Algorithm

Let $cases(n)$ denote the set of all training examples that can reach node n, unless there are no such examples in which case $cases(n)$ shall denote the set of all training examples that can reach the parent of n.

Let $value(a, x)$ denote the value of attribute a for training example x.

Let $pos(X, c)$ denote the number of objects of class c in the set of training examples X.

Let $class(x)$ denote the class of object x.

Let $Laplace(X, c) = \frac{pos(X,c)+1}{|X|+2}$ where X is a set of training examples, $|X|$ is the number of training examples and c is a class.

Let $upperlim(n, a)$ denote the minimum value of a cut c on continuous attribute a for an ancestor node x of n with respect to which n lies below the $a \leq c$ branch of x. If there is no such cut, $upperlim(n, a) = \infty$. This determines an upper bound on the values for a that may reach n.

Let $lowerlim(n, a)$ denote the maximum value of a cut c on continuous attribute a for an ancestor node x of n with respect to which n lies below the $a > c$ branch of x. If there is no such cut, $lowerlim(n, a) = -\infty$. This determines a lower bound on the values for a that may reach n.

Let $prob(x, n, p)$ be the probability of obtaining x or more positive objects in a random selection of n objects if the probability of selecting a positive object is p.

To post-process leaf l dominated by class c

```
1. Initialize to $\{\}$ a set of tuples $t$ representing potential tests.
2. For each continuous attribute $a$
   (a) Find values of
         $n$: $n$ is an ancestor of $l$
         $v$: $\exists x$: $x \in cases(n)$ & $v = value(a, x)$ & $v \leq min(value(a, y)$: $y \in cases(l)$ &
            $class(y) = c)$ & $v > lowerlim(l, a)$
         $k$: $k$ is a class
       that maximize $\mathcal{L}' = Laplace(\{x$: $x \in cases(n)$ & $value(a, x) \leq v$ & $value(a, x) >$
       $lowerlim(l, a)\}, k)$.
   (b) Add to $t$ the tuple $\langle n, a, v, k, \mathcal{L}', \leq \rangle$
   (c) Find values of
         $n$: $n$ is an ancestor of $l$
         $v$: $\exists x$: $x \in cases(n)$ & $v = value(a, x)$ & $v > max(value(a, y)$: $y \in cases(l)$ &
            $class(y) = c)$ & $v \leq upperlim(l, a)$
         $k$: $k$ is a class
       that maximize $\mathcal{L}' = Laplace(\{x$: $x \in cases(n)$ & $value(a, x) > v$ & $value(a, x) \leq$
       $upperlim(l, a)\}, k)$.
   (d) Add to $t$ the tuple $\langle n, a, v, k, \mathcal{L}', > \rangle$
3. For each discrete attribute $a$ for which there is no test at an ancestor of $l$
   (a) Find values of
         $n$: $n$ is an ancestor of $l$
         $v$: $v$ is a value for $a$
         $k$: $k$ is a class
       that maximize $\mathcal{L}' = Laplace(\{x$: $x \in cases(n)$ & $value(a, x) = v\}, k)$.
   (b) Add to $t$ the tuple $\langle n, a, v, k, \mathcal{L}', = \rangle$
```

```
4. Remove from $t$ all tuples $\langle n, a, v, k, \mathcal{L}, x \rangle$ such that $\mathcal{L} \leq Laplace(cases(l), c)$ or
   $prob(x, n, Laplace(cases(l), c)) \leq 0.05$.
5. Remove from $t$ all tuples $\langle n, a, v, c, \mathcal{L}, x \rangle$ such that there is no tuple $\langle n', a', v', k', \mathcal{L}', x' \rangle$
   such that $k' \neq c$ & $\mathcal{L}' < \mathcal{L}$.
6. For each $\langle n, a, v, k, \mathcal{L}, x \rangle$ in $t$ ordered on $\mathcal{L}$ from highest to lowest value
     If $x$ is $\leq$ then
       (a) replace $l$ with a node $n$ with the test $a \leq v$.
       (b) set the $\leq$ branch for $n$ to lead to a leaf for class $k$.
       (c) set the $>$ branch for $n$ to lead to $l$.
     else ($x$ must be $>$)
       (a) replace $l$ with a node $n$ with the test $a \leq v$.
       (b) set the $>$ branch for $n$ to lead to a leaf for class $k$.
       (c) set the $\leq$ branch for $n$ to lead to $l$.
```

References

Ali, K., Brunk, C., & Pazzani, M. (1994). On learning multiple descriptions of a concept. In *Proceedings of Tools with Artificial Intelligence*, pp. 476–483 New Orleans, LA.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees.* Wadsworth International, Belmont, Ca.

Dietterich, T. G., & Bakiri, G. (1994). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, *2*, 263–286.

Freund, Y., & Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Machine Learning*, pp. 23–37. Springer-Verlag.

Kwok, S. W., & Carter, C. (1990). Multiple decision trees. In Shachter, R. D., Levitt, T. S., Kanal, L. N., & Lemmer, J. F. (Eds.), *Uncertainty in Artificial Intelligence 4*, pp. 327–335. North Holland, Amsterdam.

Merz, C. J., & Murphy, P. M. (1998). UCI repository of machine learning databases. [Machine-readable data repository]. University of California, Department of Information and Computer Science, Irvine, CA.

Niblett, T., & Bratko, I. (1986). Learning decision rules in noisy domains. In Bramer, M. A. (Ed.), *Research and Development in Expert Systems III*, pp. 25–34. Cambridge University Press, Cambridge.

Nock, R., & Gascuel, O. (1995). On learning decision committees. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 413–420 Taho City, Ca. Morgan Kaufmann.

Oliver, J. J., & Hand, D. J. (1995). On pruning and averaging decision trees. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 430–437 Taho City, Ca. Morgan Kaufmann.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*, 197–227.

Webb, G. I. (1996). Further experimental evidence against the utility of Occam's razor. *Journal of Artificial Intelligence Research*, *4*, 397–417.

Webb, G. I. (1997). Decision tree grafting. In *IJCAI-97: Fifteenth International Joint Conference on Artificial Intelligence*, pp. 846–851 Nagoya, Japan. Morgan Kaufmann.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*, 241–259.

This article was processed using the LaTeX macro package with LLNCS style