Decision Tree Instability and Active Learning

Kenneth Dwyer and Robert Holte

Department of Computing Science, University of Alberta,
Edmonton AB, Canada
{dwyer,holte}@cs.ualberta.ca

Abstract. Decision tree learning algorithms produce accurate models that can be interpreted by domain experts. However, these algorithms are known to be unstable – they can produce drastically different hypotheses from training sets that differ just slightly. This instability undermines the objective of extracting knowledge from the trees. In this paper, we study the instability of the C4.5 decision tree learner in the context of active learning. We introduce a new measure of decision tree stability, and define three aspects of active learning stability. Several existing active learning methods that use C4.5 as a component are compared empirically; it is determined that query-by-bagging yields trees that are more stable and accurate than those produced by competing methods. Also, an alternative splitting criterion, DKM, is found to improve the stability and accuracy of C4.5 in the active learning setting.

Keywords: Decision tree learning, evaluation of learning methods, active learning, ensemble methods.

1 Introduction

Decision tree learners constitute one of the most well studied classes of machine learning algorithms. The relative ease with which a decision tree classifier can be interpreted is one of its most attractive qualities. However, decision tree learners are known to be highly unstable procedures – they can produce dramatically different classifiers from training sets that differ just slightly [1,2]. This instability undermines the objective of extracting knowledge from decision trees. Turney [2] describes a situation in which decision trees were used by engineers to help them understand the sources of low yield in a manufacturing process: “The engineers frequently have good reasons for believing that the causes of low yield are relatively constant over time. Therefore the engineers are disturbed when different batches of data from the same process result in radically different decision trees. The engineers lose confidence in the decision trees, even when we can demonstrate that the trees have high predictive accuracy.”

We have studied the instability of the C4.5 decision tree learner [3] in the context of both passive and active learning [4]. In this paper, we present the results of the active learning study. Instability is a concern in active learning because the decision tree may change substantially whenever new examples are labelled and added to the training set.

This paper asks an important new question: How stable are the decision trees produced by some well-known active learning methods? Experiments are conducted that compare the performance of several active learning methods, which use C4.5 as a component learner, on a collection of benchmark datasets. It is determined that the query-by-bagging method [5] is more stable *and* more accurate than its competitors. This is an interesting result because no previous active learning study has trained a single decision tree on examples selected by a committee of trees. Query-by-bagging tends to yield larger trees than do the other methods; yet, we provide evidence that these increases in size do not usually entail a loss of interpretability. Our second contribution is a set of new definitions of "stability," which fill an important gap between the extremities represented by existing measures. Finally, the DKM splitting criterion [6,7] is shown to improve the stability and accuracy of C4.5 in the active learning setting.

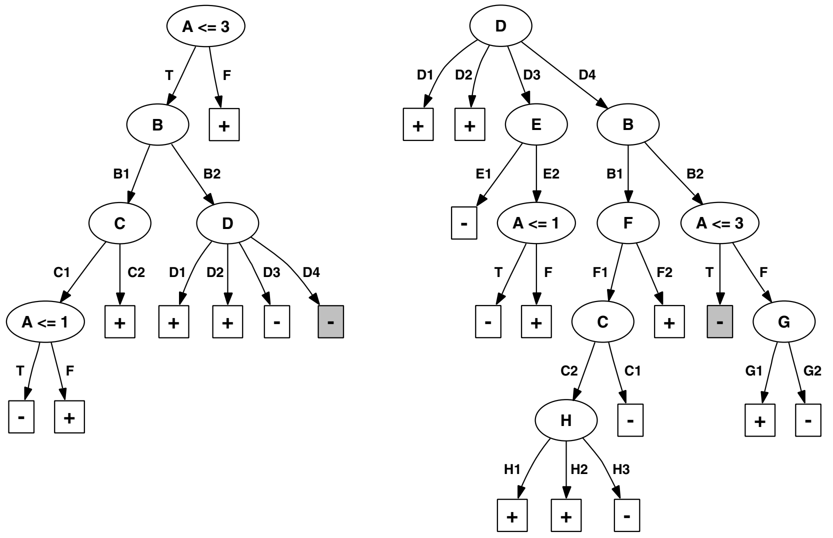
2 Decision Tree Instability

A learning algorithm is said to be *unstable* if it is sensitive to small changes in the training data. When presented with training sets that differ by some small amount, an unstable learner may produce substantially different classifiers. On the other hand, stable learning algorithms, such as nearest neighbour, are less sensitive in this regard [8]. As a concrete example of instability, consider the decision trees shown in Fig. 1. These two trees were grown and pruned by C4.5 using data from the lymphography dataset, which was obtained from the UCI repository [9]. For clarity, each attribute is denoted by a single letter,[1] and the class labels are 'malign lymph' (+) and 'metastases' (−). The data was converted to a two-class problem by deleting the examples belonging to the 'normal find' and 'fibrosis' classes, which account for 6 of the 148 instances. Figure 1(a) displays a tree, T_{106}, that was induced from a random sample consisting of 106 (roughly 75%) of the available examples. A single instance, randomly chosen from the unused examples, was appended to this training set, from which C4.5 produced the tree T_{107} that is shown in Fig. 1(b). The two trees differ considerably in size, as T_{107} contains nearly double the number of decision nodes appearing in T_{106}. Moreover, there is just one path from the root to a leaf node along which both trees perform the same set of tests when classifying an example; this is the path that consists of the tests $\{A\leq3=T, B=B2, D=D4\}$ and predicts the negative class. In all other cases, the trees apply different reasoning when making predictions. The fact that these changes were caused by the addition of one training example illustrates the instability of the C4.5 algorithm.

3 Quantifying Stability

In this paper, two types of stability are examined: *semantic* and *structural* stability. A learner that is semantically stable will, when presented with similar

[1] The values of each discrete attribute are enumerated, whereas the possible outcomes for a test on a continuous attribute are T (true) and F (false).



(a) Tree grown from 106 examples (b) Tree grown from 107 examples

Fig. 1. Decision trees grown from two subsets of the lymphography dataset that differ in a single training example. The shaded leaf in each tree highlights the only case in which both trees perform the same set of tests when predicting the class label.

data samples, tend to produce hypotheses that make similar predictions. For a learner to be structurally stable, a stronger condition must be satisfied, namely, the hypotheses that it creates from closely related data sets must be syntactically similar. Thus, structural stability is a sufficient condition for semantic stability, but the converse is not true. It is also possible to formulate a measure of stability that is not purely semantic or purely structural, but which considers some characteristics of a classifier's structure.

To measure semantic stability, we adopt a learner-independent measure called *agreement* [2]. Given two training sets, the learner induces a pair of hypotheses; the agreement is defined as the probability that a randomly chosen unlabelled example is assigned to the same class by both models. In practice, agreement is estimated by having the models classify a randomly selected set of examples.

There is no consensus on how to quantify the structural stability of decision trees. Two existing measures, called *Discrepant* [10] and *Common* [11], report minimal stability when two trees differ at the root node. However, it is possible for trees to differ at the root, and yet be quite similar or even identical elsewhere; neither of these metrics are sensitive to such an occurrence. We propose a novel measure, called *region stability*, that we argue is more appropriate for comparing the structure of decision trees.

Each leaf in a decision tree is a decision region whose boundaries are defined by the unordered set of nodes and branches that make up the path from the root to the leaf. The region stability measure compares the decision regions in one tree with those of another. Specifically, it estimates the probability that two trees classify a randomly selected example in "equivalent" decision regions. Two decision regions are considered to be equivalent if they perform the same set of tests and predict the same class label.

Region stability is estimated by having the two trees classify a randomly chosen set of unlabelled examples. As an illustration, suppose that the region stability score for the trees in Fig. 1 is computed using 100 examples, 5 of which are classified in the shaded leaf in each tree. Since this is the only decision region that the trees have in common (all other pairs of regions differ in at least one test), the region stability score is 0.05. The effect of using unlabelled examples in the calculation is that more weight is assigned to a region that classifies a larger portion of these examples in the event that their distribution is non-uniform.

When comparing two decision regions that test a particular continuous attribute, the thresholds (or cut-points) are checked for equality. However, C4.5 only places a threshold at a value that exists in the training data, and so it can be impossible for identical thresholds to appear in two trees that are induced from slightly different samples. In some learning tasks, small discrepancies of this sort may be considered superficial. For this reason, the region stability measure accepts a parameter $\epsilon \in [0, 100]\%$ that specifies a permitted margin of error between thresholds defined on a continuous attribute a. Let $\min(a)$ and $\max(a)$ be the minimum and maximum values for a in the entire dataset. Two thresholds defined on a are considered equal if they are within $\epsilon \cdot [\max(a) - \min(a)]/100$ units of one another. Note that the path from the root to a leaf may contain multiple tests on the same continuous attribute that specify distinct thresholds. The leftmost path in Fig. 1(a), for example, contains the tests A$\leq$3 and A$\leq$1. Since the first of these tests is redundant given the second, only the test A$\leq$1 is considered when comparing this decision region to another.

4 Instability in Active Learning

In active learning, the learner has the ability to choose points from the instance space on which to train a classifier. Although the ability of active learning methods to make more efficient use of unlabelled data has been well documented [5,12,13,14], little attention has been given to the stability of these techniques. This study focuses on pool-based active learning, or *selective sampling*, in which the learner draws a batch of m examples from a pool of unlabelled examples U on each iteration. The selective sampling methods tested in this study all make use of a base learning algorithm, which in this case is C4.5.

The stability of a sampling method is measured with respect to the decision trees that are induced by C4.5 from the training examples chosen by that method. We propose three different aspects of stability in active learning, which are named *PrevStab*, *FinalStab*, and *RunStab*. Each of these is quantified by applying a distance measure Φ (e.g. region stability or agreement) to specific pairs of trees. Let T_i denote the tree induced from the labelled data at iteration i of selective sampling. PrevStab quantifies the similarity of trees that are induced on consecutive iterations. For $i > 1$, trees from iterations i and $i - 1$ are compared; that is, the score $\Phi(T_i, T_{i-1})$ is computed. Calculating this score for all consecutive pairs of trees induced during active learning reveals the amount of change that occurs as a result of adding each new batch of examples to the training set. FinalStab compares the tree induced at each iteration to the tree that is induced on the final iteration of selective sampling. At iteration i of n, for $i < n$, FinalStab computes $\Phi(T_i, T_n)$; thus, it evaluates the manner in which the sequence of trees progresses toward the final tree that is produced. Finally, RunStab quantifies stability across different selective sampling "runs," in which distinct initial training sets are used. This score may be interpreted as the degree to which an active learning method yields similar trees from different initial training data. Given r runs of selective sampling, the RunStab score at iteration i is obtained by computing $\Phi(T_i^j, T_i^k)$ for each pair of runs $\{j, k\} \leq r, j \neq k$, and then taking the average of these values. Here, T_i^p is the set of labelled examples at iteration i when using the p^{th} initial training set.

When assessing the stability of a selective sampling procedure, some properties of the stability scores are desirable. For example, the FinalStab scores should increase as more examples are labelled and added to the training set, reaching reasonably high levels in the later stages of active learning. An increasing sequence of FinalStab scores implies that the learner produces decision trees that have progressively more structure in common with the final tree. PrevStab scores are expected to be low during the initial iterations, as the selective sampling method explores the instance space. Later, when the sampling method presumably begins refining the hypothesis, the PrevStab scores should increase and maintain a fairly high level. Last of all, high RunStab scores are desirable, especially in the late stages of selective sampling. If this is not the case, then the sampling method is sensitive to the particular examples that form the initial training set, and the trees it produces during different runs are dissimilar.

5 Experiments

5.1 Experimental Procedure

Experiments were carried out using four selective sampling methods: uncertainty sampling [13], query-by-bagging and query-by-boosting [5], and bootstrap-LV [14]. These are all uncertainty-based approaches, which heuristically select examples based on how confidently their true labels can be predicted. The latter three methods each form a committee of decision trees (a committee size of 10 was used here), and request the labels of the examples for which the committee "vote" is most evenly split. It is worth noting, however, that active learners exist which optimize other criteria, such as the expected future error [15]. Random sampling was also included in these experiments as a basis for comparison.

The sampling methods used C4.5 Release 8 [3] as a base learner. Experiments were duplicated using C4.5's default gain ratio splitting criterion (hereafter called entropy) and the DKM criterion [6,7] to grow trees. Each of these splitting criteria are defined by an impurity function $f(a, b)$, which is $a \log_2(a) + b \log_2(b)$

for entropy and $\sqrt{2ab}$ for DKM. Here, a and b represent the probabilities of each class within a given subset of examples formed by the split. Additionally, C4.5 Release 8 applies a penalty term to splits on continuous attributes that is specifically designed for entropy; our modification for DKM is described in [4].

Sixteen datasets were obtained from the UCI repository that each contained at least 500 training examples; this ensured that the unlabelled pool was reasonably large. Since DKM only handles two-class problems, multi-class datasets were converted to two-class ones by designating a single class as the target concept, and aggregating the remaining classes into the class "other." The target class used for each dataset is shown in Table 1. Furthermore, the region stability measure requires that each example be classified by exactly one leaf. Thus, attributes with an unknown value rate greater than 10 percent were removed, and any remaining instances that still contained missing values were deleted. The modified datasets are available online at `http://www.cs.ualberta.ca/~dwyer/ecml2007/`.

For each dataset, one third of the data from each class was randomly set aside for evaluation; these examples were used to measure the stability and error rate of the induced classifiers. Of the remaining instances, 15 percent were randomly chosen to form the initial training set, while the others constituted the pool of unlabelled examples. The batch size m was set to be 2 percent of the number of examples in a given dataset, to a minimum of 10 and a maximum of 50, and active learning ceased once two-thirds of the pool examples had been labelled. For a given dataset, 25 runs were performed, using different initial training sets. The same evaluation set was used during each run, in order to remove a source of variation when comparing stability and error rate across runs. Finally, the region stability scores were computed using ϵ values of 0, 5, and 10 percent.

5.2 Evaluation

In order to determine whether one selective sampling method was superior to another on a given dataset, a summary statistic was devised to convert each sequence of scores into a single value. Our summary statistic is a weighted average that assigns greater weight to later iterations. It was argued in Sect. 4 that structural stability is most desirable in the later stages of active learning; ideally, the learner will have a reasonable grasp of the target concept at this point, and new examples will serve mainly to refine the model, rather than to alter it significantly. A high level of stability during early iterations of active learning is of little value if stability deteriorates in later rounds.

After completing 25 runs of n iterations on a given dataset, the mean score for a statistic was calculated on each iteration. A weighted average was then computed as $\frac{1}{n}\sum_{i=1}^{n} w_i \cdot s_i$, where s_i is the mean score on iteration i; the weights w_i increased linearly as a function of i, according to the equation $w_i = \frac{2i}{n(n+1)}$. Semantic and structural stability were measured by calculating FinalStab, PrevStab, and RunStab using Turney's agreement and the region stability measure. The learning curve for error rate was summarized using the same weighted averaging scheme as for stability, under the assumption that a lower error rate is also most desirable in the later stages of active learning.

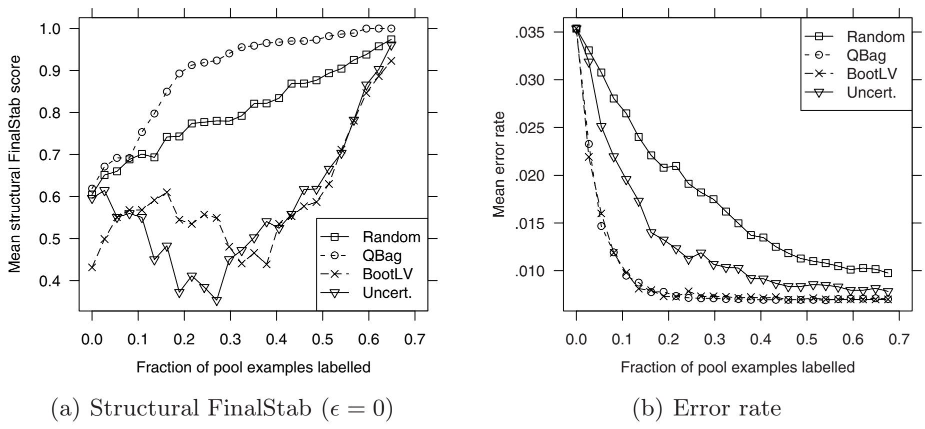


Fig. 2. Plots of two statistics for the kr-vs-kp dataset when using DKM. QBag is the most stable and the most accurate sampling method in this case. In all experiments, learning ceased when 2/3 of the instances in the pool had been labelled.

To exemplify how the weighted averaging scheme characterizes the scores that are produced by the selective sampling procedures, consider Fig. 2(a). Here, the structural FinalStab scores for 4 of the 5 sampling methods on the kr-vs-kp dataset are plotted as a function of the training set size. Query-by-bagging (QBag) was clearly the most stable method on this dataset, with a weighted score of .953, while random sampling (Random) placed second, at .858. Bootstrap-LV (BootLV) and uncertainty sampling (Uncert), were more closely matched. Although Uncert was marginally more stable than BootLV during the final iterations, the former's instability between $x = .13$ and $x = .27$ dropped its weighted score (.638) below that of BootLV (.644).

In order to assess the statistical significance of the experimental results, the following methodology was applied [16]. The null hypothesis is that all the sampling methods are equivalent. For each dataset, ranks were assigned to the methods based on their weighted scores. Next, the Friedman test[2] was applied to each method's average rank, and the critical value was computed using the F_F statistic. If the null hypothesis could be rejected based on this value at a chosen significance level α, the Nemenyi test was used to determine whether significant differences existed between any given pair of sampling methods.

To illustrate this process, consider the weighted error rates displayed in Table 1. For each dataset, ranks (shown in parentheses) are assigned in order of increasing error rate, with averages used in the event of a tie. With an average rank of 4.375 over all datasets, Random is the worst-ranking method, while QBag ranks the best, on average (1.625). At $\alpha = .05$, the null hypothesis is rejected based on the Friedman test. The critical difference is 1.527, and so differences between average ranks of at least this amount are statistically significant. Thus, QBag, QBoost, and BootLV are each significantly more accurate than Random;

[2] The tests of statistical significance that were applied are described in [16].

Table 1. Weighted average error rates when using the DKM splitting criterion, (ranks in parentheses). For each dataset, the lowest observed error rate is bolded.

Dataset (target class)	**Random (R)**	**QBag (G)**	**QBoost (T)**	**BootLV (L)**	**Uncert (U)**
anneal (not-3)	.144 (4)	**.121** (1)	.135 (3)	.125 (2)	.150 (5)
australian (+)	**.129** (1.5)	**.129** (1.5)	.131 (5)	.130 (3.5)	.130 (3.5)
car (acceptable)	.090 (5)	**.077** (1)	.082 (4)	.078 (2)	.081 (3)
german (bad)	.293 (5)	**.274** (1)	.285 (2)	.290 (4)	.289 (3)
hypothyroid (+)	.006 (5)	**.002** (2)	**.002** (2)	**.002** (2)	.004 (4)
kr-vs-kp (no-win)	.014 (5)	**.007** (1.5)	.008 (3)	**.007** (1.5)	.010 (4)
letter (k)	.015 (5)	**.011** (2)	**.011** (2)	**.011** (2)	.013 (4)
nursery (priority)	.056 (5)	**.038** (1.5)	.039 (3)	**.038** (1.5)	.044 (4)
pendigits (9)	.016 (5)	**.010** (1.5)	**.010** (1.5)	.012 (4)	.011 (3)
pima-indians (+)	.286 (5)	.283 (2)	**.280** (1)	.284 (3)	.285 (4)
segment (cement)	.020 (5)	**.011** (1)	.012 (2.5)	.012 (2.5)	.019 (4)
tic-tac-toe (−)	.217 (5)	**.197** (1)	.201 (2)	.207 (3)	.211 (4)
vehicle (opel)	**.227** (1)	.231 (5)	.229 (3.5)	.228 (2)	.229 (3.5)
vowel (hud)	.056 (5)	**.033** (1)	.036 (2)	.037 (3)	.049 (4)
wdbc (malignant)	.073 (4)	.068 (2)	**.067** (1)	.069 (3)	.076 (5)
yeast (nuclear)	.256 (4.5)	**.250** (1)	.253 (2.5)	.256 (4.5)	.253 (2.5)
Avg. rank	(4.375)	(1.625) **R,U**	(2.500) **R**	(2.719) **R**	(3.781)

also, QBag is superior to Uncert. A letter beside the average rank of a sampling method S indicates that S is significantly better than the method corresponding to that letter. For example, the R and U in the QBag column of Table 1 imply that QBag is significantly more accurate than Random and Uncert, respectively.

5.3 Experimental Results

Due to space limitations, only the average ranks are presented for each statistic; these are displayed in Table 2. Although the average ranks are sufficient for demonstrating our findings, detailed results may be viewed at `http://www.cs.ualberta.ca/~dwyer/ecml2007/`. In Table 2, the level of significance that was tested for each statistic is shown in the "α" column.

Error Rate. The committee-based methods achieved significantly lower error rates than did Random, independent of the splitting criterion employed. Uncert was also significantly less accurate than QBag.

Overall, the results support the findings of previous active learning studies that involved these sampling methods, in that Random was certainly the inferior approach, and committee-based methods usually produced lower error rates than Uncert [5,14]. However, there is a subtle, yet important factor distinguishing our experiments from existing research involving active learning with decision trees. The committee-based results reported in our experiments represent the performance of a single C4.5 decision tree that is trained on examples selected by a committee of trees. By contrast, in the original experiments involving QBag,

Table 2. Average ranks for the sampling methods on each statistic. The top half of the table diplays the results for the entropy criterion; bottom half: DKM criterion.

	Statistic	ϵ	α	**Random (R)**	**QBag (G)**	**QBoost (T)**	**BootLV (L)**	**Uncert (U)**
Entropy criterion	Error Rate	–	.05	4.406	2.000 **R**,**U**	2.188 **R**	2.719 **R**	3.688
	Tree Size	–	.01	1.062 **G**,**T**,**U**	4.062	3.938	2.500	3.250
	Seman. FinalStab	–	.05	4.281	1.969 **R**	3.000	3.094	2.656 **R**
	Seman. PrevStab	–	.05	4.000	2.062 **R**	3.531	3.094	2.312 **R**
	Seman. RunStab	–	.01	4.625	1.344 **R**,**U**	2.625 **R**	2.656 **R**	3.750
	Struct. FinalStab	0		3.562	2.875	2.875	3.000	2.688
	Struct. FinalStab	5	.10	3.375	3.031	2.875	2.812	2.906
	Struct. FinalStab	10		3.344	3.000	2.938	2.750	2.969
	Struct. PrevStab	0		3.312	2.719	3.406	3.062	2.500
	Struct. PrevStab	5	.10	3.188	2.812	3.469	2.969	2.562
	Struct. PrevStab	10		3.062	2.812	3.594	2.969	2.562
	Struct. RunStab	0		4.469	1.750 **R**,**U**	2.000 **R**,**U**	2.656	4.125
	Struct. RunStab	5	.01	4.188	1.656 **R**,**U**	2.188 **R**,**U**	2.594	4.375
	Struct. RunStab	10		4.156	1.656 **R**,**U**	2.250 **R**,**U**	2.562	4.375
DKM criterion	Error Rate	–	.05	4.375	1.625 **R**,**U**	2.500 **R**	2.719 **R**	3.781
	Tree Size	–	.01	1.125 **G**,**T**,**U**	4.125	3.938	2.625	3.125
	Seman. FinalStab	–	.05	4.000	2.188 **R**	3.250	2.750	2.812
	Seman. PrevStab	–	.05	3.594	2.312 **T**	3.844	2.812	2.438
	Seman. RunStab	–	.01	4.531	1.562 **R**,**U**	2.750 **R**	2.469 **R**	3.688
	Struct. FinalStab	0		3.281	2.562	3.188	3.094	2.875
	Struct. FinalStab	5	.10	3.156	2.594	3.125	3.031	3.094
	Struct. FinalStab	10		3.281	2.562	3.125	3.125	2.906
	Struct. PrevStab	0		2.875	2.812	3.844	2.906	2.562
	Struct. PrevStab	5	.10	2.844	2.875	3.688	2.938	2.656
	Struct. PrevStab	10		2.812	2.938	3.688	2.844	2.719
	Struct. RunStab	0		4.031	1.875 **R**,**U**	2.125 **R**,**U**	2.625	4.344
	Struct. RunStab	5	.01	3.719	2.031 **U**	2.188 **U**	2.750	4.312
	Struct. RunStab	10		3.875	1.938 **R**,**U**	2.344 **U**	2.562	4.281

for instance, a committee of C4.5 trees selected unlabelled examples that were subsequently used to train a bagged committee of trees [5]. Note that although a single decision tree trained from labelled data L is likely to be less accurate than a bagged committee trained from L, the committee is no longer intelligible [1].

Other methods have been proposed for training one type of classifier on examples selected by another type. Lewis and Catlett [13] employed a probabilistic classifier to select training examples for C4.5, and Domingos [17] used a committee to generate labelled data, from which a single classifier was trained. However, we are not aware of any previous study in which a committee of decision trees was used to train a single tree within the active learning framework.

Tree Size. In terms of the number of leaf nodes, the selective sampling methods consistently yielded larger trees than did Random (see "Tree Size" in Table 2). The trees grown by QBag tended to be the largest, containing 38 percent more leaves, on average, than those of Random. Does this imply that trees grown using QBag, for example, are more difficult to interpret than trees produced by Random? While no agreed-upon criterion exists for distinguishing between a tree that is interpretable and a tree that is not, one simple criterion is that there might exist a threshold t, such that any tree containing more than t leaves is uninterpretable. The datasets on which the weighted average leaf count for Random is at most t and the count for QBag is greater than t would then represent cases where QBag sacrifices intelligibility. Testing all integer values of t ranging from 1 to 25, we find that this occurs on at most 5 datasets ($t = 13$) when using DKM, and at most 3 datasets ($t = 11, 12$) with entropy. Thus, QBag's gains in accuracy are not typically made at the expense of intelligibility.

Stability. With regard to semantic stability, QBag achieved the best average rank on FinalStab, PrevStab, and RunStab, for both splitting criteria. Random was the least stable method in all but one case (PrevStab with DKM). Although Uncert also performed well on FinalStab and PrevStab, it was significantly less stable than QBag on the RunStab measure, as was Random. No strong conclusions could be drawn regarding the semantic stability of QBoost or BootLV.

As for structural stability, the RunStab results were highly significant. QBag and QBoost were the two best-ranked methods for all three values of ϵ that were used, while Random and Uncert were the worst. By definition, QBag and QBoost always choose the m highest scoring examples from the pool – the ones for which the committee vote is most divided. Therefore, a given example will be added to the training set if it receives a sufficiently high score at some iteration. BootLV, on the other hand, does not necessarily choose the highest scoring examples; it samples m times from a probability distribution in which the weight of an example is proportional to its score. Random is completely stochastic, which accounts for its instability across runs. As for Uncert, its low RunStab scores are a consequence of assigning scores to unlabelled examples based on the hypothesis of a single decision tree. Since this tree is generally unstable, the score assigned to a given example is likely to change considerably when the tree is induced from different training data. This problem is mitigated by the committee-based methods, as a committee of trees tends to be more stable than a single tree [1].

The results for structural FinalStab and PrevStab did not reveal any statistically significant differences between the sampling methods, even at $\alpha = .10$.

Table 3. Pairwise comparisons involving the QBag sampling method. The significance level α is indicated in parentheses where applicable.

(a) Structural FinalStab win-loss counts for QBag vs. Random

ϵ	**Entropy**	**DKM**
0	10-6 (.05)	11-4 (.05)
5	9-7 (.10)	10-6
10	9-6 (.10)	10-6

(b) Structural stability win-loss counts for DKM vs. entropy when using QBag

ϵ	**FinalStab**	**PrevStab**	**RunStab**
0	10-6 (.10)	11-5 (.05)	9-7
5	11-5 (.05)	12-4 (.05)	11-5 (.05)
10	12-4 (.05)	12-4 (.05)	10-6 (.10)

The ranges of the average ranks were smaller for these statistics; yet, Random had the worst average rank on FinalStab, for example, for all values of ϵ that were tested. A direct comparison between Random and QBag – the best method on most of the statistics – does in fact reveal significant differences. Table 3(a) compares QBag and Random on the structural FinalStab measure. Here, a win is recorded for the sampling method that achieves the higher weighted score on a given dataset. QBag obtains the most wins under all 6 conditions, and the Wilcoxon signed-ranks test, which is recommended when comparing two methods [16], finds that the QBag scores are significantly better in 4 of these cases. With regard to the PrevStab scores, there were no significant differences detected between QBag and Random by this test.

Comparison of Splitting Criteria. The weighted scores obtained when using the entropy splitting criterion were compared to those obtained with DKM, and the Wilcoxon signed-ranks tests was used to assess statistical significance. With respect to error rates, DKM was significantly more accurate than entropy when QBag or BootLV were used ($\alpha = .10$), and DKM never recorded less than 9 wins with any of the other methods. DKM frequently grew smaller trees, but the difference was significant only with BootLV ($\alpha = .10$). Remarkably, DKM yielded higher scores on the majority of datasets for every sampling method and every measure of structural stability, at all values of ϵ. We highlight the comparison between DKM and entropy when using QBag, as it has been shown in previous sections to be the superior sampling method. As the data in Table 3(b) reveals, the FinalStab and PrevStab scores for QBag improved significantly when DKM was used to grow trees. Similarly, QBag's RunStab scores improved for all values of ϵ when using DKM instead of entropy. Here, permitting a small margin of error between continuous thresholds revealed that DKM produced decision regions that were more similar than those formed with entropy. Finally, the semantic stability of C4.5 was less influenced by the choice of splitting criterion, as significant differences were detected only for PrevStab when using BootLV ($\alpha = .05$) or Random ($\alpha = .10$). In both instances, DKM was superior.

6 Conclusions

We have presented a methodology for evaluating the stability of decision tree learners in the context of active learning, which includes a novel measure of decision tree stability. The main conclusions drawn from our experiments are, first of all, that query-by-bagging (QBag) is the method of choice for training a single, interpretable decision tree, when using the C4.5 algorithm. QBag was found to be superior based on many of the stability measures, and was never significantly less stable than any other sampling method. Moreover, QBag produced the most accurate decision trees, and so the increased stability did not correspond with higher error rates. Although QBag yielded trees that were larger, on average, than those of the competing methods, we provided evidence that this would not usually be detrimental to intelligibility. The second important finding is that the

DKM splitting criterion improves the stability and accuracy of C4.5 in the active learning setting. In particular, since QBag performed better with DKM, this combination is recommended for training a single tree. It is important to emphasize that these conclusions are based on average performance across datasets, as no sampling method was superior on all the datasets that were tested.

Acknowledgments. This research was funded by NSERC, the Informatics Circle of Research Excellence (iCORE), and the Alberta Ingenuity Fund.

References

1. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
2. Turney, P.D.: Bias and the quantification of stability. Machine Learning 20(1-2), 23–33 (1995)
3. Quinlan, J.R.: Improved use of continuous attributes in C4.5. JAIR 4, 77–90 (1996)
4. Dwyer, K.D.: Decision tree instability and active learning. Master's thesis, University of Alberta (2007)
5. Abe, N., Mamitsuka, H.: Query learning strategies using boosting and bagging. In: Proc. ICML '98, pp. 1–9 (1998)
6. Dietterich, T.G., Kearns, M., Mansour, Y.: Applying the weak learning framework to understand and improve C4.5. In: Proc. ICML '96, pp. 96–104 (1996)
7. Drummond, C., Holte, R.C.: Exploiting the cost (in)sensitivity of decision tree splitting criteria. In: Proc. ICML '00, pp. 239–246 (2000)
8. Alpaydin, E.: Introduction to Machine Learning. MIT Press, Cambridge (2004)
9. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI ML Repository (1998)
10. Shannon, W.D., Banks, D.L.: Combining classification trees using MLE. Statistics in Medicine 18(6), 727–740 (1999)
11. Pérez, J.M., Muguerza, J., Arbelaitz, O., Gurrutxaga, I., Martín, J.I.: Consolidated trees: Classifiers with stable explanation. In: Singh, S., Singh, M., Apte, C., Perner, P. (eds.) ICAPR 2005. LNCS, vol. 3686, pp. 99–107. Springer, Heidelberg (2005)
12. Cohn, D.A., Atlas, L.E., Ladner, R.E.: Improving generalization with active learning. Machine Learning 15(2), 201–221 (1992)
13. Lewis, D.D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In: Proc. ICML '94, pp. 148–156 (1994)
14. Saar-Tsechansky, M., Provost, F.: Active sampling for class probability estimation and ranking. Machine Learning 54(2), 153–178 (2004)
15. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: Proc. ICML '01, pp. 441–448 (2001)
16. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. JMLR 7, 1–30 (2006)
17. Domingos, P.M.: Knowledge acquisition from examples via multiple models. In: Proc. ICML '97, pp. 98–106 (1997)