Proceedings of the Workshop on Knowledge Discovery in Health Care and Medicine of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD KDHCM), Athens, Greece, 5-9 September, 2011, pp. 28–39.

Analysis of Decision Tree Pruning Using Windowing in Medical Datasets with Different Class Distributions

Pedro Santoro Perez and José Augusto Baranauskas

Department of Computer Science and Mathematics, University of Sao Paulo, Ribeirao Preto, SP, Brazil
pedrosperez@usp.br, augusto@usp.br

Abstract. Windowing searches for a sample of the training set that would provide better and smaller models than those obtained from the entire set. Originally, the classifiers are pruned only at the end of each trial, and the technique is considered to be interesting in unbalanced datasets. We have performed experiments to verify if pruning the intermediary trees within a trial is beneficial; if windowing is better than the tree inducer alone; and the behavior of windowing in medical datasets with different class distributions. Our main conclusions are: windowing is not better than the tree inducer alone and not beneficial when comparing balanced and unbalanced domains. However when the emphasis is on human understanding of the induced knowledge, as well as on extensibility, windowing tends to build smaller trees, which are more likely to be interpreted by humans, without significant differences in AUC.

1 Introduction

Windowing is a technique whose idea is to find a subsample of a dataset that provides enough information for an inducer to train a classifier and have results similar to or better than those achieved by training a model from the entire dataset, reducing the complexity of the learning problem [6]. This way, windowing can be seen as a subsampling technique [22], but, unlike other subsampling techniques (*e.g.*, bootstrapping), windowing tend to provide more class balanced and informative samples. Windowing was first proposed by [21] in the context of decision trees as a way to deal with memory restrictions in the late 1970's to some relatively large datasets. [20] argues windowing is interesting for two reasons: i) in some cases, especially those free of noise, it may make the time taken to build the model shorter (*e.g.*, when the dataset is very large and a model that perfectly classifies all training instances is achieved in the first iterations). For most cases, however, windowing makes that time longer; ii) windowing may produce more accurate classifiers. Memory is still an issue, given the existence of huge databases on medical and biological domains.

The study in [5] states that windowing contributes to better threshold choices for continuous attributes in decision tree induction. In [16], windowing is seen

Proceedings of the Workshop on Knowledge Discovery in Health Care and Medicine of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD KDHCM), Athens, Greece, 5-9 September, 2011, pp. 28–39.

as a way to make unstable learning algorithms more stable. Some other studies have explored windowing in the context of rule induction [11], as they have noted the technique produces better results with that kind of inducer than with decision trees, since rules are learned independently of each other and are less susceptible to changes in the class distribution. In spite of the experience Quinlan had with windowing, those studies also state that the technique is not good to be used with decision trees, especially in noisy domains [12]. Possibly because of that, very little has been researched towards improving the combination between windowing and decision trees. Our group believes there is still some aspects of windowing that could be explored further, *e.g.*, applying the technique in domains with balanced and unbalanced class distribution, which we will discuss here.

Our group has been focusing on symbolic classifiers, *i.e.*, those which can be written as a set of rules. Instead of just concerning about accuracy or any other performance measure, this kind of classifier can also be used as a way to provide insights on the domain being considered when interpreted by humans, a very important characteristic in the medical domain [13]. In this context, one advantage of windowing over other techniques used to improve classifiers' performance is that its use with symbolic classifiers still provides symbolic classifiers. Our contribution here evaluates windowing in medical datasets with different class distribution, verifying if it performs better than the decision tree inducer alone and in unbalanced domains.

The remaining sections are organized as follows: Section 2 gives details about windowing and describes the objectives of this study; Section 3 describes some functions to measure class distribution balance; Section 4 introduces the datasets used here; Section 5 describes in detail the experiments performed and whose results are shown in Section 6. Conclusions are presented in Section 7.

2 Windowing

This section describes windowing as it is implemented in C4.5 Release 8. Algorithm 1 shows the pseudo-code of windowing, where N represents the number of instances in the training set and x_i and y_i $(i = 1, \ldots, N)$ represent a vector containing the attribute values and the class label for instance i, respectively. The $\| E \|$ operator returns 1, if E is true, or 0, otherwise.

Before the learning process begins, a subset of the training set is chosen, forming the initial window (Line 3), from which a model is induced (Line 6); the model is then used to predict the class of all training examples, which might produce some misclassifications (Lines 8-9); if the errors found are less than the errors of the best classifier so far (initially, $N+1$), the current classifier is kept as the best one (Lines 10-13); if there were errors outside the window, the window is updated and used to train another classifier; the resulting model is tested again and the process is repeated until no misclassifications occur outside the window.

The initial window is not actually sampled randomly. First the training set is shuffled; then the algorithm tries to build a window as uniform as possible,

Proceedings of the Workshop on Knowledge Discovery in Health Care and Medicine of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD KDHCM), Athens, Greece, 5-9 September, 2011, pp. 28–39.

Algorithm 1 Windowing

Require: Instances: a set of N labelled instances $\{(x_i, y_i), i = 1, 2, \ldots, N\}$
W: the initial window size, default value $W \leftarrow \max\{0.2N, 2\sqrt{N}\}$
I: the tentative window increment, default value $I \leftarrow \max\{0.2W, 1\}$
Ensure: bestClassifier: the best classifier found

```
1: procedure window(Instances,W,I)
2: N ← size(Instances)
3: window ← sample(W,Instances)
4: bestClassifierErrors ← N + 1
5: repeat
6:    classifier ← induceClassifier(window)
7:    [classifier ← prune(classifier)]
8:    windowErrors ← Σ_{x_i ∈ window} ‖ classifier(x_i) ≠ y_i ‖   // Errors within the window
9:    testErrors ← Σ_{x_i ∉ window} ‖ classifier(x_i) ≠ y_i ‖   // Errors outside the window
10:   if (windowErrors + testErrors < bestClassifierErrors) then
11:      bestClassifier ← classifier
12:      bestClassifierErrors ← windowErrors + testErrors
13:   end if
14:   increment ← max(min(testErrors, I), testErrors /2)
15:   increment ← min(increment, N - size(Instances − window))
16:   window ← window + getMisclassifiedInstances(increment, Instances − window)
17: until (testErrors = 0)
18: [bestClassifier ← prune(bestClassifier)]
19: return bestClassifier
```

i.e., the class distribution gets to be as balanced as possible. Considering c as the actual number of all class values of a dataset and $\mathcal{E} = W/c$ as the expected number of instances for each class value in the initial window W, for any given class value, if the number of instances labelled with it is at least equal to $\mathcal{E}$, then it will be represented by $\mathcal{E}$ instances in the initial window; otherwise, all instances representing that class value will be added to the window. This often leads to better results, especially in cases of unbalanced classes [20]; it also contributes to better threshold choices for continuous attributes [5]. In spite of being as uniform as possible, the class distribution and the examples chosen to be in the initial window are subject to a random sampling, which means that one may have different initial windows, leading to different decision tree classifiers.

As it can be seen in Algorithm 1, at least half of the misclassified examples outside the window is added to it at each iteration (Lines 14-16), provided that there are enough examples. This is done to make the model converge faster. The method `getMisclassifiedInstances` (Line 16) takes `increment` misclassified examples from outside the window and adds them to the window.

The process can be repeated more than once. Each repetition is called a *trial* and starts with a different initial window, which often generates a different final

Proceedings of the Workshop on Knowledge Discovery in Health Care and Medicine of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD KDHCM), Athens, Greece, 5-9 September, 2011, pp. 28–39.

classifier. By default, C4.5 uses 10 trials. For the output, it offers 2 possibilities: i) the best tree classifier from all trials is returned to the user; or ii) the system takes the best classifier of each trial and combines them into only one rule-based classifier. In this paper, we are going to focus on the former.

An important point concerns the fact that, in the original implementation, only the best tree of each trial is pruned, and that is done only when the trial is over (Line 18). Line 7 is not present in Quinlan's implementation, so that all intermediary trees within a trial are kept unpruned and, to decide which tree is the best so far within the current trial, the algorithm uses the sum of the errors found in the whole training set. At the end of each trial, the tree returned is pruned and its estimated errors are compared to the estimated errors of the best tree found in all trials processed so far. The process of pruning trees is known to be important because it helps the tree improve its generalization error.

Considering these ideas, we have performed experiments including Line 7 in order to verify if pruning intermediary trees within a trial is worth the extra effort. For completeness, we have also evaluated if the original windowing (*i.e.*, without Line 7) brings significant benefits over the tree inducer alone. Since [20] states that windowing is not guaranteed to return better results, but it should be tried especially in unbalanced domains, the experiments have been performed in datasets with different class distributions. The question that arises from it is how to numerically measure class distribution balance.

3 Impurity Functions as Class Balance Measures

Since we have used datasets with very different numbers of classes (from 2 to 43), the class distribution itself cannot be considered a comparable measure. First assume there are c classes numbered $1, 2, \ldots, c$ and denote the proportions of the classes by $\mathbf{p} = (p_1, p_2, \ldots, p_c)$. We are looking for some function $\phi(\mathbf{p})$ obeying the following properties: (i) it is a symmetric function of $\mathbf{p}$; (ii) its maximum is achieved when all $p_j = 1/c$; (iii) its minimum is achieved when $p_i = 1$ and every other $p_j = 0$ (for all $j \neq i$) and (iv) its range is $[0, 1]$, *i.e.*, $\phi(\mathbf{p}) : \mathbb{R}^c \rightarrow [0, 1]$, where a value close to one is obtained when class proportions are more balanced (close to a uniform distribution) and a value close to zero is obtained when class proportions are more unbalanced. Functions obeying the first three properties are known as impurity functions [4, p. 32]. The range of some known impurity functions is usually $\mathbb{R}^+$, but limited by some constant (once c is fixed). Therefore, some appropriate normalization is needed in order to ensure the fourth property.

In this study we suggest four different measures, shown in (1) product-of-frequency-based, (2) entropy-based, (3) frequency-based and (4) gini-based, where n_i is the number of instances belonging to class i and, therefore, $p_i = n_i/N$. For numerical precision, the second form of (1) is preferable. Figure 1 shows how they behave in 2-class datasets. It is straightforward to show that ϕ_b and ϕ_g are equivalent for 2-class domains, which does not hold for $c > 2$.

Proceedings of the Workshop on Knowledge Discovery in Health Care and Medicine of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD KDHCM), Athens, Greece, 5-9 September, 2011, pp. 28–39.

$$\phi_b(\mathbf{p}) \triangleq c^c \prod_{i=1}^{c} p_i = e^{cln(c)+\sum_{i=1}^{c} ln(p_i)} \quad . \tag{1}$$

$$\phi_e(\mathbf{p}) \triangleq \frac{-\sum_{i=1}^{c} p_i log_2(p_i)}{log_2(c)} \quad . \tag{2}$$

$$\phi_f(\mathbf{p}) \triangleq 1 - \frac{\sum_{i=1}^{c} \mid p_i - \frac{1}{c} \mid}{2\left(1-\frac{1}{c}\right)} \quad . \tag{3}$$

$$\phi_g(\mathbf{p}) \triangleq \frac{c}{c-1} \sum_{i=1}^{c} p_i(1-p_i) \quad . \tag{4}$$

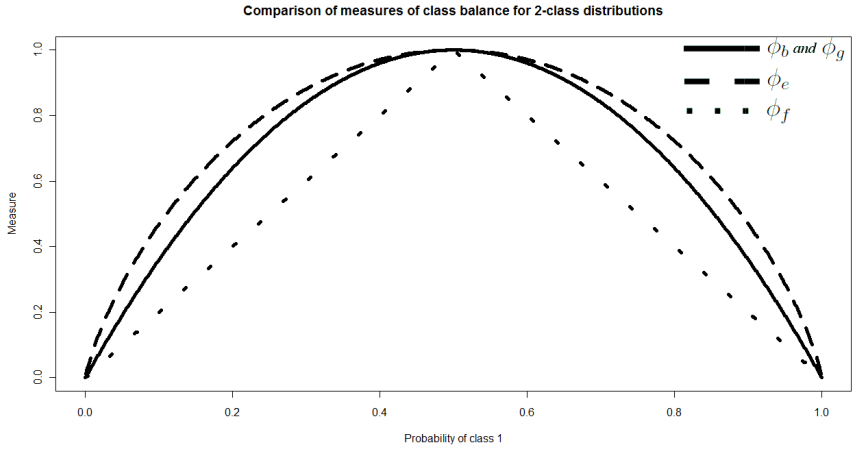

Fig. 1. Behavior of measures ϕ_b, ϕ_e, ϕ_f and ϕ_g in 2-class distribution problems.

4 Datasets

The experiments reported here used 32 datasets, all of which representing real medical data. The medical domain often imposes difficult obstacles to learning algorithms: high dimensionality, huge or very small amounts of instances, several possible class values, unbalanced classes, etc. The biomedical domain is of particular interest since it allows to evaluate windowing under real and difficult situations often faced by human experts. Table 1 shows a summary of the datasets, none of which having missing values for the class attribute.

Breast Cancer, *Lung Cancer*, *CNS* (Central Nervous System Tumour Outcome), *Colon*, *Lymphoma*, *GCM* (Global Cancer Map), *Ovarian 61902*, *ECML*

Proceedings of the Workshop on Knowledge Discovery in Health Care and Medicine of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD KDHCM), Athens, Greece, 5-9 September, 2011, pp. 28–39.

Table 1. Summary of the datasets used in the experiments. ATTR, $a_{\#}$ and a_a stand for the total number of attributes and for the number of numerical and nominal attributes, respectively; MISS represents the percentage of attributes with missing values, not considering the class attribute; the last four columns bring measures of class distribution balance, as defined by (1), (2), (3) and (4). Datasets are in ascending order of ϕ_b.

#	Dataset	N	c	ATTR($a_{\#}$,a_a)	MISS	ϕ_b	ϕ_e	ϕ_f	ϕ_g
1	Allhyper	3772	5	29 (7, 22)	5.54%	0.00	0.09	0.03	0.07
2	C. Arrhythmia	452	16	279 (206, 73)	0.32%	0.00	0.65	0.40	0.73
3	Thyroid 0387	9172	32	29 (7, 22)	5.50%	0.00	0.36	0.24	0.46
4	Allhypo	3772	4	29 (7, 22)	5.54%	0.00	0.23	0.10	0.19
5	ECML	90	43	27679 (27679, 0)	0.00%	0.00	0.96	0.76	0.99
6	Lymphoma	96	9	4026 (4026, 0)	5.09%	0.01	0.77	0.58	0.82
7	Lymphography	148	4	18 (3, 15)	0.00%	0.02	0.61	0.39	0.71
8	ANN Thyroid	7200	3	21 (6, 15)	0.00%	0.03	0.28	0.11	0.21
9	P. Patient	90	3	8 (0, 8)	0.42%	0.11	0.62	0.43	0.63
10	Hypothyroid	3163	2	25 (7, 18)	6.74%	0.18	0.28	0.10	0.18
11	HD Switz.	123	5	13 (6, 7)	17.07%	0.20	0.85	0.63	0.89
12	Sick	3772	2	29 (7, 22)	5.54%	0.23	0.33	0.12	0.23
13	GCM	190	14	16063 (16063, 0)	0.00%	0.39	0.97	0.82	0.99
14	Dermatology	366	6	34 (1, 33)	0.06%	0.49	0.94	0.80	0.96
15	Hepatitis	155	2	19 (6, 13)	5.67%	0.66	0.73	0.41	0.66
16	H. Survival	306	2	3 (2, 1)	0.00%	0.78	0.83	0.53	0.78
17	Splice Junction	3190	3	60 (0, 60)	0.00%	0.81	0.93	0.72	0.92
18	Breast Cancer	286	2	9 (0, 9)	0.35%	0.84	0.88	0.59	0.84
19	WBC	699	2	9 (9, 0)	0.25%	0.90	0.93	0.69	0.90
20	C. Method	1473	3	9 (2, 7)	0.00%	0.90	0.97	0.84	0.97
21	Leukemia nom.	72	2	7129 (0, 7129)	0.00%	0.91	0.93	0.69	0.91
22	Leukemia	72	2	7129 (7129, 0)	0.00%	0.91	0.93	0.69	0.91
23	Pima Diabetes	768	2	8 (8, 0)	0.00%	0.91	0.93	0.70	0.91
24	CNS	60	2	7129 (7129, 0)	0.00%	0.91	0.93	0.70	0.91
25	Colon	62	2	2000 (2000, 0)	0.00%	0.92	0.94	0.71	0.92
26	Ovarian 61902	253	2	15154 (15154, 0)	0.00%	0.92	0.94	0.72	0.92
27	HD Hungarian	294	5	13 (6, 7)	20.46%	0.92	0.94	0.72	0.92
28	WDBC	569	2	30 (30, 0)	0.00%	0.94	0.95	0.75	0.94
29	Lung Cancer	32	3	56 (0, 56)	0.28%	0.96	0.99	0.89	0.99
30	Liver Disorders	345	2	6 (6, 0)	0.00%	0.97	0.98	0.84	0.97
31	Heart Statlog	270	2	13 (13, 0)	0.00%	0.99	0.99	0.89	0.99
32	HD Cleveland	303	5	13 (6, 7)	0.18%	0.99	0.99	0.91	0.99

(European Conference on Machine Learning), *Leukemia*, *Leukemia nom.*, *WBC* (Wisconsin Breast Cancer), *WDBC* (Wisconsin Diagnostic Breast Cancer), *Lymphography* and *H. Survival* (*H.* stands for *Haberman's*) are all related to cancer and their attributes consist of clinical, laboratory and gene expression data. *Leukemia* and *Leukemia nom.* represent the same data, but the second one had

its attributes discretized [19]. *C. Arrhythmia* (*C.* stands for *Cardiac*), *Heart Statlog*, *HD Cleveland*, *HD Hungarian* and *HD Switz.* (*Switz.* stands for *Switzerland*) are related to heart diseases and their attributes represent clinical and laboratory data. *Allhyper*, *Allhypo*, *ANN Thyroid*, *Hypothyroid*, *Sick* and *Thyroid 0387* are a series of datasets related to thyroid conditions. *Hepatitis* and *Liver Disorders* are related to liver diseases, whereas *C. Method* (*C.* stands for *Contraceptive*), *Dermatology*, *Pima Diabetes* (Pima Indians Diabetes) and *P. Patient* (*P.* stands for *Postoperative*) are other datasets related to human conditions. *Splice Junction* is related to the task of predicting boundaries between exons and introns. Datasets were obtained from the UCI Repository [9], except *CNS*, *Colon*, *Lymphoma*, *GCM* and *ECML* were obtained from [2]; *Ovarian 61902* was obtained from [3]; *Leukemia* and *Leukemia nom.* were obtained from [1].

As it can be seen in Table 1, the four measures given by (1), (2), (3) and (4) may produce different dataset orders. When comparing them among datasets with the same number of classes, we would expect the same order, but that is not guaranteed to happen when comparing datasets with a different number of classes. Figure 2 shows the behavior of the four measures over the datasets.

5 Experimental Methodology

The experiments reported here were performed in Weka [14]. Instead of C4.5, we have used J48, and, since Weka does not provide any implementation of windowing, a Weka's Java class, named `weka.classifiers.meta.Windowing`, was implemented as a meta-inducer, following the rules and standards defined by Weka. Our implementation of windowing was based on C4.5, including all its features, except for the possibility to build a single rule-based classifier out of the best trials' classifiers, which remained as a feature yet to be implemented.

The implementation allows the choice of any inducer available in Weka to be used as the base inducer (Line 6 of Algorithm 1). For our purposes, the base inducer was J48. For our analysis, we have considered three inducers: 1) pruned J48; 2) windowing using J48 as the base classifier, where the intermediary trees were kept unpruned and only the trial best trees were pruned (like in C4.5, *i.e.*, Algorithm 1 without Line 7 but including Line 18), referred to as WTP further on; and 3) windowing using J48 as the base classifier, but pruning every tree, even the ones generated within the trials (Algorithm 1 without Line 18 but including Line 7), denoted as WAP from now on. Besides the pruning option, every other parameter was kept with its default value.

The two measures chosen to analyze the results are the weighted average area under the ROC curve and the final tree size (from now on, we'll refer to them as AUC and $SIZE$, respectively). We here consider smaller trees to be better than larger ones, considering all other measures the same, because smaller trees tend to be simpler and more accepted by domain experts. The experiments were performed with ten repetitions of 10-fold cross validation. The average of all repetitions for a certain inducer on a certain dataset was taken as the value of performance (AUC and $SIZE$) for that pair.

Proceedings of the Workshop on Knowledge Discovery in Health Care and Medicine of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD KDHCM), Athens, Greece, 5-9 September, 2011, pp. 28–39.

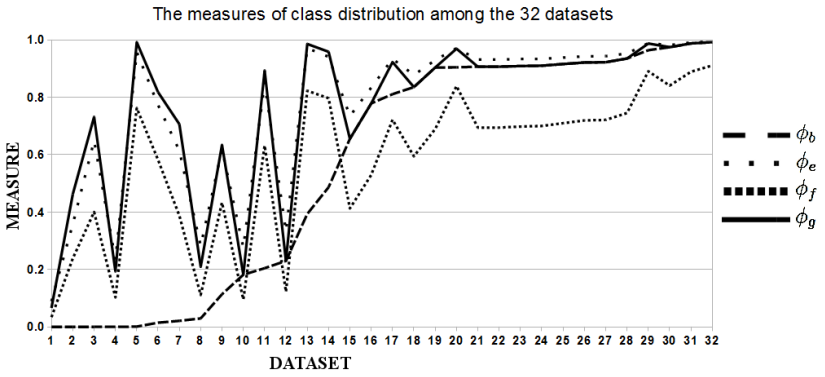

Fig. 2. Measures ϕ_b, ϕ_e, ϕ_f and ϕ_g over our 32 datasets (in ascending order of ϕ_b).

In order to analyze if the differences observed were significant, we have used the Friedman test [10], a non-parametric technique based on ranks and widely used by the machine learning community. The most important advantage of choosing a non-parametric approach is not making the assumption of any prior probability distribution for the variables being considered (ANOVA, for example, assumes a normal distribution). For the case of rejecting the null hypothesis (which states that differences in the data are obtained by chance), a *post-hoc* test is necessary to check in which classifier pairs the differences actually are [7]. Again we chose a non-parametric approach and the p-values found for the possible pairs were adjusted according to [15], which describes a multiple test procedure that controls the family wise error rate. The tests were implemented and performed using the R software for statistical computing (`http://www.r-project.org/`). We have performed two kinds of tests: 1) versus control, using J48 as the control and verifying if the windowing inducers performed differently than it; and 2) all versus all, making all possible comparisons among the three inducers. All results reported in Section 6 consider a significance level of 5%.

To verify if windowing could bring benefits when applied to unbalanced datasets, these would have to be ordered by how balanced they are. At this point, we had to choose, among the four measures, which to use in our experiments. This task is equivalent to decide which splitting criteria should be used under decision tree induction, and therefore it has no single answer. We chose ϕ_b to order the datasets in our experiments, based in our perception and experience of working with classification tasks; the other three seem to be too optimistic. In some cases, such as when it disagrees more with the other three, giving a much lower value than them, the datasets were quite unbalanced and ϕ_e, ϕ_f and ϕ_g tended to give high scores. We found ϕ_b more realistic, but for some tasks the other three measures could be good as well. We have then ordered the datasets according to ϕ_b and chosen as a threshold the value of 0.50, which resulted in a group of 14 unbalanced datasets (below 0.50) and another of 18 balanced datasets. We applied the same statistical tests to the two groups.

Proceedings of the Workshop on Knowledge Discovery in Health Care and Medicine of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD KDHCM), Athens, Greece, 5-9 September, 2011, pp. 28–39.

6 Results and Discussion

The results of the experiments are shown in Table 2. As it can be seen, J48 alone had a better average rank than the other two inducers and WTP had a slightly better average rank than WAP. After performing the tests with all datasets, no significant differences in the *AUC* values were found in either case (against-a-control and all-versus-all).

Considering the *SIZE* values, the test found significant differences. In the all-versus-all case, the significant differences found were: WTP is better than WAP, with a p-value of less than 2.00×10^{-6}; and J48 is better than WAP, with a p-value of 2.43×10^{-5}. In the against-a-control case, the only significant difference found was between J48 and WAP, with a p-value of 2.43×10^{-5}. In spite of not being significant, WTP has a better average rank than J48. Trees provided by WTP are sometimes larger than those provided by J48, but they are notably smaller in some cases, for instance, H. Survival or Ovarian 61902.

When comparing the *AUC* values of the two groups of datasets (the balanced first 14 and the unbalanced last 18), the results showed that: for the balanced group, there were no significant differences in either case (against-a-control and all-versus-all), with p-values greater than 0.20, but J48 had a better average rank than the other two, and WAP was slightly better than WTP; for the unbalanced group, there were no significant differences either, with p-values greater than 0.70, but WTP was better than J48 (a little improvement, compared to the other group).

Considering the *SIZE* values: for the balanced group, WTP has done better (but not significantly) than J48 and both were significantly better than WAP; for the unbalanced group, J48 and WTP were similar and both were still significantly better than WAP. WAP had trees significantly larger than the other two inducers in all cases. This might have happened because, during training, the inducer found some pruned trees that seemed to be better (but larger), but the *AUC* values for WAP were not worth the extra effort of pruning all trees.

7 Conclusion

In this paper, a study on windowing has been conducted. The technique is a general purpose sampling method designed to enable induction of decision trees to address the problem of analyzing large datasets, already available, for example, in the form of medical and biological data. Also, the automated discovery of symbolic models is expected to be a common and important feature in medical and biological domains. In comparison, the datasets used here are of modest size. Nevertheless, they have been sufficient to illustrate the potential utility of windowing in some domains.

Based on the experiments performed, it is possible to state that pruning the intermediary trees didn't represent a benefit over pruning only the trials best trees, in spite of being more expensive. WAP was actually a little worse than WTP when considering *AUC* and significantly worse when considering *SIZE*.

Proceedings of the Workshop on Knowledge Discovery in Health Care and Medicine of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD KDHCM), Athens, Greece, 5-9 September, 2011, pp. 28–39.

Table 2. *AUC* and *SIZE* values observed in the experiments. The datasets are in ascending order of ϕ_b. The last lines show the average rank in each case.

		AUC			SIZE		
#	**Dataset**	J48	**WTP**	**WAP**	J48	**WTP**	**WAP**
1	Allhyper	0.91	0.95	0.93	27.88	31.20	33.02
2	C. Arrhythmia	0.73	0.73	0.73	80.62	81.80	89.92
3	Thyroid 0387	0.98	0.98	0.98	375.42	337.30	357.04
4	Allhypo	1.00	1.00	1.00	27.88	26.64	27.26
5	ECML	0.54	0.55	0.55	57.22	58.30	57.88
6	Lymphoma	0.87	0.82	0.83	14.94	16.14	16.00
7	Lymphography	0.77	0.79	0.79	28.00	25.81	31.18
8	ANN Thyroid	0.99	0.99	0.99	33.60	33.36	37.96
9	P. Patient	0.49	0.49	0.43	1.36	2.18	10.12
10	Hypothyroid	0.95	0.95	0.95	12.42	14.04	15.08
11	HD Switz.	0.58	0.57	0.57	36.86	39.13	47.13
12	Sick	0.95	0.96	0.95	49.38	46.99	53.26
13	GCM	0.79	0.78	0.78	37.02	37.52	38.14
14	Dermatology	0.98	0.98	0.98	35.96	26.88	30.36
15	Hepatitis	0.67	0.72	0.75	17.66	15.44	20.60
16	H. Survival	0.57	0.62	0.57	21.84	5.26	33.36
17	Splice Junction	0.97	0.96	0.96	216.94	198.83	233.76
18	Breast Cancer	0.61	0.61	0.56	12.78	13.23	41.16
19	WBC	0.96	0.94	0.94	23.46	19.10	28.46
20	C. Method	0.66	0.67	0.66	247.96	225.52	298.44
21	Leukemia nom.	0.90	0.86	0.88	10.39	9.31	10.33
22	Leukemia	0.81	0.79	0.77	4.30	4.60	4.76
23	Pima Diabetes	0.75	0.70	0.73	43.40	47.94	54.68
24	CNS	0.55	0.51	0.53	7.92	8.00	8.34
25	Colon	0.81	0.77	0.75	6.98	7.00	7.02
26	Ovarian 61902	0.96	0.99	0.98	10.04	5.04	5.52
27	HD Hungarian	0.78	0.77	0.79	10.53	13.00	20.43
28	WDBC	0.93	0.93	0.94	22.46	21.84	24.86
29	Lung Cancer	0.60	0.61	0.61	13.04	12.96	16.22
30	Liver Disorders	0.65	0.63	0.63	50.02	53.00	61.06
31	Heart Statlog	0.79	0.77	0.76	34.64	33.94	44.62
32	HD Cleveland	0.77	0.75	0.78	42.52	38.91	57.43
Average rank (all)		1.84	2.05	2.11	1.69	1.53	2.78
Average rank (first 14)		2.00	1.89	2.11	1.64	1.71	2.64
Average rank (last 18)		1.72	2.17	2.11	1.72	1.39	2.89

We could conclude that pruning the intermediary trees is not enough to bring benefits by itself. Under these considerations, other changes to the algorithm would be necessary, *e.g.*, considering the estimated error as the error of a classifier within a trial, just like what happens with the trials best trees (future work).

As one adds examples to the window, the number of unseen instances decreases. This situation increases the chances to get a classifier with fewer errors, *i.e.*, trees at the beginning of a trial have less chances to be considered better than trees at the end. At the limit, all examples are inside the window, which guarantees that there will be no errors outside it (we will have only the resubstitution error, which is optimistic), even when the resulting tree is not the one with the best generalization power. This might be another motivation to suggest the estimated generalization error and not the errors found outside the window or even a combination of the two could also be considered.

Considering only accuracy and size, our results don't show a significant advantage of using windowing, not even for unbalanced cases, although it was a little better than for the balanced ones. One apparent advantage that could be pointed out is that WTP produced better average results for $SIZE$ than J48, although this difference was not significant. On the other hand, J48 was even better for AUC values than the two versions of windowing, although that difference was not significant either. Our results conflict with Quinlan's experience with windowing, which shows that windowing rarely provides worse trees, but agree with other authors, who say windowing is not suitable for decision trees in noisy domains. As future work, we will try and find a way to minimize the effect of noise on the algorithm.

[20] states that windowing rarely produces worse trees than the tree inducer alone. On the other hand, the literature has not shown such good results. For example, [8] states that windowing has its performance deteriorated in noisy domains, because it ends up adding all noisy instances to the window, since these are very likely to be misclassified even by a good model [18]. Our results agree with the latter, but we think there is still some improvement to make on windowing, *e.g.*, finding a way to deal with noise.

Very great attention has been given to performance measures such as AUC, leaving the interpretation of the resulting model at a lower level, even when analyzing symbolic classifiers. There are many different methods for constructing decision trees, although none of them is universally the best, since different application domains lead to different problems, requiring different solutions. It is then expected that a combination of methods, like the one proposed in this work, may yield better classification models in some cases. In this study, we have shown windowing tend to build smaller classifiers, which are more likely to be interpreted by humans, without significant differences in terms of AUC. On another study, we have applied windowing to the classification task of predicting peptide activity. In that case, windowing has performed better than J48 alone in terms of accuracy and useful knowledge. The specialists involved in the study have selected some laboratory experiments based on the tree that windowing provided and they have reported interesting biological results [17].

Acknowledgments: This work was funded by FAPESP as well as a joint grant between CNPq/FAPEAM through the INCT ADAPTA Project.

References

1. Cancer program data sets. http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi (2010), Broad Institute
2. Dataset repository in arff (weka). http://www.upo.es/eps/bigs/datasets.html (2010), BioInformatics Group Seville
3. Datasets. http://cilab.ujn.edu.cn/datasets.htm (2010), Cilab
4. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth & Books, Pacific Grove, CA (1984)
5. Catlett, J.: Megainduction. Ph.D. thesis, Basser Dept. of Computer Science (1991)
6. Chen, Q.: Inductive Learning on Partitioned Data. Master's thesis, The University of Vermont (2004)
7. Demšar, J.: Statistical comparison of classifiers over multiple data sets. Journal of Machine Learning Research 7(1), 1–30 (2006)
8. Domingos, P.: Efficient specific-to-general rule induction. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96). pp. 319–322 (1996)
9. Frank, A., Asuncion, A.: UCI machine learning repository. http://archive.ics.uci.edu/ml (2010)
10. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics 11(1), 86–92 (1940)
11. Fürnkranz, J.: More efficient windowing. In: Proceedings of the Fourteenth National Conference on Artificial Intelligence. pp. 509–514. Citeseer (1997)
12. Fürnkranz, J.: Noise-tolerant windowing. In: Proceedings of the Fifteenth international joint conference on Artificial Intelligence. pp. 852–857. Citeseer (1997)
13. Gamberger, D., Lavrač, N., Zelezný, F., Tolar, J.: Induction of comprehensible models for gene expression datasets by subgroup discovery methodology. Journal of Biomedical Informatics 37(4), 269–284 (2004)
14. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explorations 11(1), 10–18 (2009)
15. Hommel, G.: A stagewise rejective multiple test procedure based on a modified Bonferroni test. Biometrika 75(2), 383–386 (1988)
16. Last, M., Maimon, O., Minkov, E.: Improving stability of decision trees. Int'l journal of pattern recognition and artificial intelligence 16(2), 145–160 (2002)
17. Lira, F., Perez, P.S., Baranauskas, J.A., Nozawa, S.R.: Synthetic peptides and prediction of antimicrobial activity using decision trees. Aquaculture Research (2011), submitted
18. Nettleton, D., Orriols-Puig, A., Fornells, A.: A study of the effect of different types of noise on the precision of supervised learning techniques. Artificial Intelligence Review 33(4), 275–306 (2010)
19. Netto, O.P., Nozawa, S.R., Mitrowsky, R.A.R., Macedo, A.A., Baranauskas, J.A.: Applying decision trees to gene expression data from DNA microarrays: A leukemia case study. In: XXX Congress of the Brazilian Computer Society, X Workshop on Medical Informatics. p. 10. Belo Horizonte, MG (2010)
20. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993), san Francisco, CA
21. Quinlan, J.: Discovering rules by induction from large collections of examples. In: Michie, D. (ed.) Expert Systems in the Micro-Electronic Age. Edinburgh University Press (1979)
22. Reinartz, T.: A unifying view on instance selection. Data Mining and Knowledge Discovery 6(2), 191–210 (2002)