

Decision Trees in Numerical Attribute Spaces

Thierry Van de Merckt

IRIDIA cp194/6 - Université Libre de Bruxelles

Av. F. Roosevelt 50, 1050 Bruxelles

Belgium.

(THVDM@ULB.AC.BE)

Abstract

Research in Supervised Concept learning has not paid special attention to Numerical Attribute Spaces, although they present particular characteristics. One of them is that the combinatorial complexity of finding the correct disjunctions of the target concepts is much higher than with Nominal Attributes. In this paper, we argue that new kinds of heuristics using numerical similarities among training instances should be used. In the framework of TDIDT, we present a new approach that uses unsupervised heuristics issued from Cluster Analysis for selecting generalisation steps. Empirical tests show promising results, at least when the target concept is only partially represented in the training.

1 Introduction

Top Down Induction of Decision Trees (TDIDT) techniques have been widely studied over the last decade. The major part of research efforts has improved the ability of these algorithms to cope with *missing values* in attributes descriptions [Quinlan, 1989; Breiman *and al.*, 1984; Cestnik *and al.*, 1987] and to improve their resistance to *noise* [Breiman *and al.*, 1984; Quinlan, 1986; Clark and Niblett, 1987; Cestnik *and al.*, 1987]. Some authors have also tried to better understand the intrinsic limitations of TDIDT induced by their bias and to partially overcome the related problems [Matheus *and al.*, 1989; Rendell *and al.* 1989; Yang *and al.*, 1991] but no special attention has been devoted to the special case of using TDIDT in Numerical Attributes Spaces.

Producing a Decision Tree using Numerical, and especially Continuous attributes, rises some specific problems compared to using Nominal attributes. Some of these problems are explained in [Van de Merckt, 1992] and will not be reported here. To make things clear, before explaining the problems that have motivated this research, section 2.1 exposes the principles of TDIDT algorithms applied to Numerical Spaces. Next, the problems are exposed in section 2.2 and some solutions are proposed in section 2.3. Section 3 reports some encouraging empirical results of our approach. Section 4 concludes by a discussion.

2 TDIDT and Numerical Attributes Spaces

Supervised Concept Learning algorithms start with a set of objects, called the *training set*, belonging to a set of mutually

exclusive *classes*. Each object, whose class is known, is described by a vector of attribute values. The general induction task is to develop a *classification rule* that can determine the class of any object from its values of the attributes [Quinlan, 1896]. TDIDT techniques build a concept under the form of a Decision Tree where each leaf is labeled by a class and each node tests an attribute of an object whose outcome specifies the branch to be followed in the Decision Tree. An unseen instance is then classified by following a path from the root node to a leaf which assigns its labeled class to the instance.

All learning algorithms use the following schemata [Gams and Lavrac, 1987]:

1. Preprocessing of the Training Set: Numerical Attributes discretisation, Missing values treatment, etc.
2. Construction of a classification rule:
 - 2.1 Building a classification rule, called the *growing* phase;
 - 2.2 *Pruning* or Truncation of classification rules to avoid overfitting of the training data.

In this paper we will focus only on point 2.1. No pruning techniques will be incorporated in our tests.

2.1 The Algorithm

The general Algorithm of TDIDT for building a classification rule with numerical attributes is iterative: starting from a collection C of objects belonging to a node N , the algorithm generates a test T on an object based on a key value (called a cut-point) that produces two branches: the right branch is followed by instances whose attribute value is less than the key value and the left branch is followed by the others [Breiman *and al.*, 1984; Cestnik *and al.*, 1987]. Test T therefore produces a binary partition $\{C_1, C_2\}$ of C . This process is repeated on all C_i until some stopping condition is verified, i.e., that C_i contains objects belonging to one single class. This process generates binary trees. The pseudo-code of the algorithm is the following:

Develop-a-Tree-From [N] - node N contains C_i instances - :

0. IF Stopping condition is True for N THEN Stops

1. For Each candidate Attribute A_j

Find all candidate cut-points $\{a_{jl}\}$ and For all a_{jl}

1. Produce a double Predicate testing the value of attribute A_j against a cut-point a_{jl} : $P_{jl} = [\leq [A_j, a_{jl}] ; > [A_j, a_{jl}]]$

2. Generate a partition $\{C_{i1}, C_{i2}\}$ of C_i using P_{jl}

3. Calc Partition' score based on a given Selection Measure $SM[C_{i1}, C_{i2}]$

2. Select the double Predicate P_{jl} with best score

3. Create children nodes $[N_1, N_2]$ of N using P_{jl}

Cut-Point Selection - The strategy used for selecting candidate cut-points uses the numerical information contained in the instances. The idea is that it is useless to select a cut-point that only separates instances belonging to the same class. The selected cut-points are therefore restricted to the set of *points that, at least, separate two instances of different classes*, which are called *informative cut-points*. This strategy is implemented by sorting all values of training instances belonging to the current node. A cut-point is created as the average value between two consecutive figures of the sorted list if it separates at least two instances of different classes. While non-informative cut-points would never be chosen by an entropy-like measure, this process spares time during the growing phase since these useless cut-points are not evaluated.

Predicate Construction - For each selected cut-point, two predicates are created, one for the left branch and one for the right one. To avoid errors of omission, one of the two predicates contains the equal sign which is assigned to the one that covers the largest part of training instances belonging to the current node. Therefore an unknown instance whose value is equal the cut-point would follow the most probable path.

Selection Measure - All Selection Measures that have been developed so far use an entropy- or probability-based measure on the Class information (for a survey see [Mingers, 1989]) and hence, does not use all information available in numerical figures contained in the training. Next sections explain the problem that may occur when using these criteria in Numerical spaces and present new Selection Measures.

2.2 Finding Sound Clusters

All Selective Inductive systems are laying on an implicit hypothesis (which is called the *Similarity hypothesis*): it assumes that Instance Space proximity entails similar Class membership value [Rendell and al., 1989]. A concept description therefore may be seen as a class membership function defined over the Instance Space [Rendell and Cho, 1990] whose axes are the attributes describing the objects. When the axes are restricted to the subset of attributes issued from the Decision Tree, we speak about the Description Space. Each leaf, which represents a disjunct of a classification rule, is a hyper-rectangle whose edges are orthogonal to the axes. In this sense, TDIDT can be seen as a *supervised clustering algorithm* where leaves represent *clusters*, i.e., class labeled regions of the description space.

A first major difference between Nominal versus Numerical Attribute Spaces regarding TDIDT, is the way the similarity hypothesis is used. Indeed, in Nominal Spaces, an attribute with m different values generate m paths from the current node, producing m clusters of instances that share a common value for the attribute [Quinlan, 86]. The strategy for producing a partition from a node and hence, for generating the clusters, focusses on the similarity of the training instances (common values for the tested attribute). Once this has been done, the Selection Measure heuristically selects the attribute that entails the strongest correlation between similarity and class information. By doing this, the algorithm identifies the attributes that best fit into the similarity hypothesis. Things are different

when working with Numerical Attributes. Indeed, the binary partition of instances produced by a cut-point, at a given node, does not imply a high level of proximity among instances following the same path for the tested attribute. Here the strategy for creating a partition from a node does not ensure or focus on similarity. It is by its iterative Divide-and-Conquer strategy that TDIDT will finally produce clusters with a satisfactory level of proximity/similarity, as far as class prediction is concerned. It is remarkable to observe that the Similarity Hypothesis is used in its inverse order in this case than when working with Nominal attributes. Indeed, with Nominal values, TDIDT uses attribute similarity for generating the clusters (partition based on common values) and class similarity is evaluated as a side effect to identifies interesting correlations. While in Numerical spaces, only class similarity guides the generation of clusters and proximity of instances within each cluster is a side effect of the Divide-and-Conquer strategy. This observation rises the question of the adequacy of a classical entropy-based TDIDT for finding correct disjunctive Numerical descriptions, as Fig.1 illustrates.

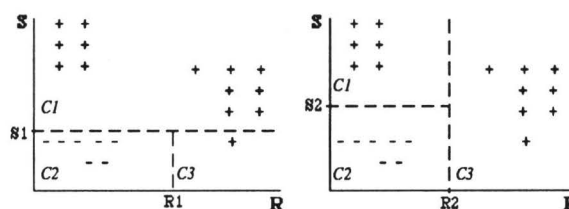


Fig. 1a: Entropy-based cut

Fig. 1b: Contrast-based cut

The target concept of Fig.1 is composed of one cluster of negative examples and two clusters (a disjunction) of positive examples (Fig.1b). The left case (Fig.1a) would be chosen by classical entropy-based TDIDT techniques and would result in merging the single positive instance of cluster C3 and the negative cluster C2 if pruning should be used, although it is clearly part of the upper positive cluster. This would lead to produce an incorrect concept description with less predictive power, even if the target concept is perfectly represented in the training.

This example illustrates the second problem, related to the first one, of using TDIDT in Numerical Spaces: finding correct disjunctions is far more difficult in Numerical Spaces because, by their continuous nature, they offer a very large combinatorial complexity for generating partitions and hence, for producing more or less generalized disjuncts. It also illustrates that *finding the correct disjuncts within a given class is an unsupervised task*, even if belonging to Supervised Concepts Learning! The lack of instance space similarity measure in the process of generating partitions may lead, as Fig.1 shows, to take wrong decisions. These considerations led us look at a way to reintroduce instance similarity measures for selecting partitions as well as to look into Unsupervised Clustering Algorithms to find heuristics that could be useful regarding the discovery of sound disjuncts in a Numerical Instance Space.

2.3 A Contrast Measure for Cut-Points Selection

The shape and therefore the generalisation performed over a cluster in the instance space depends on the selection of cut-points at the different depths of the Tree. The problem was therefore to implement some new heuristic through the Selection

Measure. By looking at Fig.1, it can be seen that, when projecting the instances on each candidate attribute, attribute R (R2 cut-point on Fig.1b) produces two very distinct clusters while attribute S (with S1 on Fig.1a) does not: in this case R2 would be chosen. The idea, illustrated by this example, is to search for clusters that are as much contrasted as possible from the instance space proximity point of view, *even if they contain elements belonging to distinct classes*.

A well known heuristic used by Unsupervised Clustering algorithms is the *Error Sum of Squares* [Kaufman and Rousseeuw 90]. It aims to find clusters that maximise the Euclidean distance among them while minimising the distance of the elements within each of them. We have derived an attribute Selection Measure based on this idea. The principle is *to rely on the Similarity hypothesis and to select, in an unsupervised way* (without looking at the class labels), *cut-points which produce the more contrasted partitions*. Edwards & Cavalli-Sforza designed in 1965 an unsupervised clustering algorithm based on a polythetic divisive method using the following criterion:

$$\text{Edwards-Cavalli Criteria} = \frac{|A| \cdot |B|}{|A \cup B|} \cdot \|\bar{x}(A) - \bar{x}(B)\|^2$$

where A & B represent the resulting sets and where $\bar{x}(A)$ is the vector of average values of the elements belonging to the subset A. It has been proved that maximising this criterion is equivalent to the Error Sum of Squares criterion [Kaufman and Rousseeuw, 1990].

Unsupervised Monothetic Contrast Criterion - This measure is a *monothetic* version of Edwards-Cavalli's criterion: it evaluates the partition issued from a candidate cut-point *without looking at the class label of instances*. It favours cut-points that produce the most contrasted partitions. It is given by:

$$\text{Contrast}(C_i, C_j, A_k) = \frac{|C_i| \cdot |C_j|}{|C_i \cup C_j|} \cdot (m_{k_i} - m_{k_j})^2$$

where C_i & C_j are the resulting binary partition, m_{k_i} is the mean value of C_i for the Attribute A_k that produced the current partition and $|C_i|$ is the number of elements of C_i .

Despite the fact that this Selection Measure is unsupervised, one has to remember that a partition submitted for evaluation is produced in a supervised way, since only *informative cut-point* are selected. This criterion, being a pure implementation of the ideas described above, is too drastic for three reasons: (i) it will always "prefer" a more contrasted partition against a zero-entropy partition, even if the differences in the contrast measures are not significant, (ii) there is no way to choose between two cut-points that share very close contrast scores and (iii) it has no ability to evaluate the correlation between attribute similarity and class information. These reasons lead, as the experiments will show, to produce overly complex Trees (although they are very accurate in some cases). To avoid these problems, we have designed a second Selection Measure that is a mix between an unsupervised & a supervised decision.

Mixed Supervised/Unsupervised Monothetic Criterion - This criterion uses a contrast-based approach and weights contrast figures by the entropy measure [Quinlan, 1896] of the resulting sets. While using a proximity measure for selecting partitions, it is able to identify its correlation with the class information. It favours partitions with high contrast and low entropy. When entropy is zero, it is replaced by 10E-6, which will results in

selecting, upon all zero-entropy partitions, the one with the highest contrast value. It is defined by:

$$CE(C_i, C_j, A_k) = \frac{\text{Contrast}(C_i, C_j, A_k)}{E(C_i, C_j)}$$

where $E[C_i, C_j]$ is the entropy of the partition $[C_i, C_j]$, given by:

$$E(C_i, C_j) = -\frac{1}{|C_i \cup C_j|} \sum_{g=i,j} \sum_{f=1}^p |C_{gf}| \cdot \log_2 \frac{|C_{gf}|}{|C_g|}$$

where $|C_{gf}|$ is the number of instances belonging to class f in the set C_g , $g=i,j$ and where $|C_g|$ is the number of instances belonging to the set C_g . It can be written as:

$$CE(C_i, C_j, A_k) = -\frac{|C_i| \cdot |C_j| \cdot (m_{k_i} - m_{k_j})^2}{\sum_{g=i,j} \sum_{f=1}^p |C_{gf}| \cdot \log_2 \frac{|C_{gf}|}{|C_g|}}$$

In next sections, *Contrast* & *CE* measures will be called *Ct-2* & *Ct-6* respectively.

3 Empirical Tests

Ct-2 & Ct-6 have been evaluated by comparing their performance with the Entropy Gain ratio of Quinlan [86] and with a Random Selection which provides an unbiased Selection Measure.

3.1 Experiments Description

For each Training set, a new Tree, using one the the four Selection Measures, has been developed and tested. Training sets were generated by a random selection of a given percentage of each class in the data set (called *Training%*), leaving some unseen instances for testing. To observe the learning rates of the four criteria, we started with a very low *Training%*, gradually increasing it by *adding* a number of randomly chosen *new* instances. This procedure "simulates" a system that incrementally receives more and more information from the universe. This protocol has been repeated 20 times. Figures in charts are the average performance over the 20 runs.

Performance indicators - Performance is evaluated by:

Accuracy: classification accuracy was calculated over the test set (excluding training data);

Complexity of the concepts: the number of disjuncts, that is, the number of leaves in the Tree;

Learning rate: conveys the shape of the learning curve, i.e., the speed at which classification accuracy increases when new instances are brought to the system.

Stabilisation rate: this is a new performance indicator that aims to capture the speed at which the dispersion of n tests around the average learning curve becomes narrow and stabilises. The Stabilisation rate can be efficiently represented by plotting the Variation Ratio which is the ratio between the Standard Deviation of Accuracy and its mean value. It gives the relative importance of the Standard Deviation independently of Accuracy level. As charts will show, when the *training%* increases, Accuracy becomes more stable over the 20 runs (Min & Max are closer). It means that, given an amount of randomly selected training instances, whatever the chosen instances in the total set are, the system always produces nearly the same Accuracy. My intuition is that a concept description may be stable at a level of training because of two factors: (i) there is always some redundant information in a data set. For each domain, there should exist

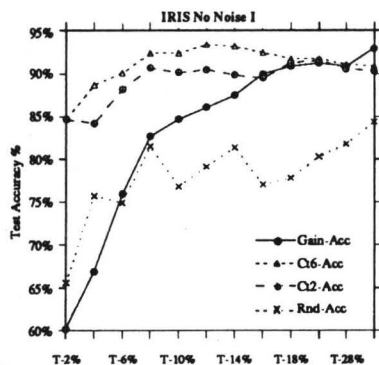


Fig.2: Iris (no noise) Accuracy

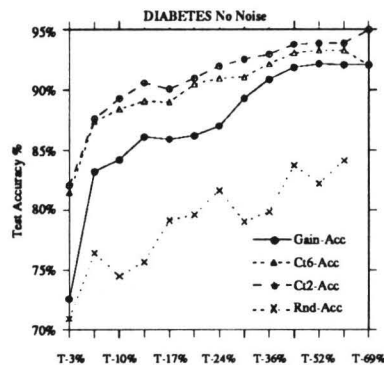


Fig.3: Diabetes (no noise) Accuracy

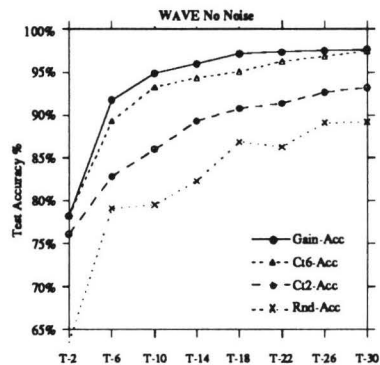


Fig.4: Wave Form (no noise) Accuracy

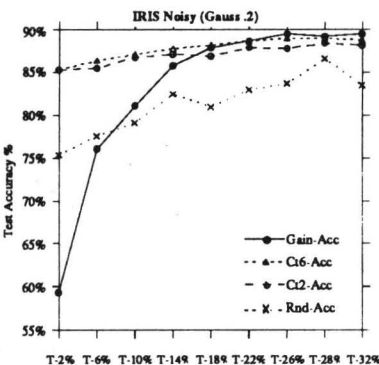


Fig.5: Iris (with noise) Accuracy

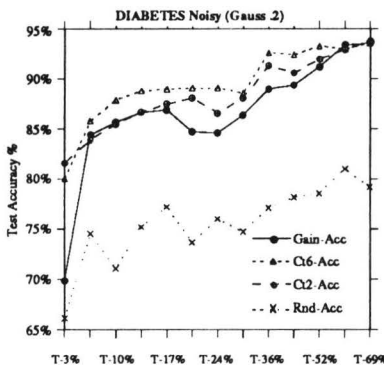


Fig.6: Diabetes (with noise) Accuracy

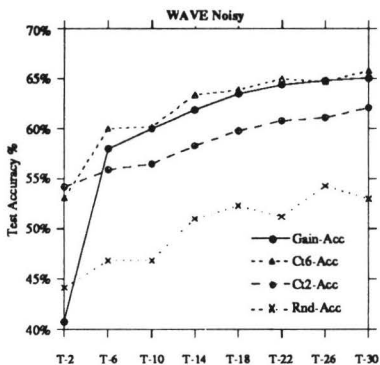


Fig.7: Wave Form (with noise) Accuracy

a training level for which the information contained by the randomly selected instances always gives a partial but sufficient view of the target concept; (ii) the capacity of the learning algorithm to anticipate (to "induce" we could say!) the "shape" of the target concept with a few numbers of instances. The first factor depends on the domain but the second one depends on the power of the learning algorithm or on the adequacy of its bias for the domain. David Aha [1991] has stressed on the fact that the learning rate is a very useful indicator for finite-size training sets and it seems to me that it is also important that the performance is not too sensitive to the choice of the training instances. Indeed, relying on a system that has a high performance dispersion is like betting on our lucky star each time we use it.

Data Sets - Data used in these experiments are:

Iris: this is a well known database in pattern recognition literature used by Fisher in 1936. It contains 3 classes of 50 instances each, where a class refers to a type of Iris plant. Attributes are numerical: Sepal Length & Width and Petal Length & Width in cm. It does not contain noise or residual variation (attributes are accurate for the learning task).

Diabetes: there are 5 numerical Attributes (Glufast, Glutest, Sspg, Instest, Relwt) with a high correlation between Glutest & Glufast and between Instest & Sspg. The number of instances per Class is: C1=26, C2=35, C3=84, Total=145.

Wave Forms: while the two first sets are real world data, this one is artificial. It has been proposed in [Breiman and al., 1984]. It is composed of 3 classes, each of them being a linear combination of three distinct wave forms. Each instance is composed by a vector of 21 continuous values. Random noise can be added to the set. This is a difficult

problem when noise is present. For these experiments, we generated a total set of 600 elements (200 per class).

Each data set has been used in a noise free version and a noisy one. For Diabetes and Iris, noise has been added to attribute values of all instances using a Normal distribution with 0 mean and a standard deviation equals to 1/5 of the standard deviation of the attribute on the whole data set.

3.2 Results

All charts use • for Gain, ⊕ for Ct-2, Δ for Ct-6 and x for Random cut-point selection. Training levels has been noted by T-y% (y% per class) for Iris and Diabetes and T-y (y instances per class) for Wave. Due to place considerations, charts for noisy data sets will not be reproduced in this paper, except for Accuracy. However, the main effects of noise on performance indicators will be commented.

Accuracy & Learning Rate - Figures 2 to 7 significantly illustrates the difference of Accuracy and Learning Rate of Ct-6 and Ct-2 compared to the Gain criterion.

On *Iris*, 2% training level represents 1 instance per class and Ct-2&6 already achieve 84.7% accuracy against 60% for the Gain criterion for both free noise and noisy tests. The learning rate of Ct-2&6 is very good: they start at a high accuracy level from the early stages of training. At training levels larger than 22% the learning curves of Gain and Ct-2&6 are flat (average accuracy is 94% at T-70% for free noise sets and 92% for noisy ones) and no significant differences may be observed among them (this explains why the charts stop at T-32%). Bad results achieved by the Gain at low training levels is due to the fact that, at these levels, nearly all attributes are able to reach a low entropy value and Gain criterion has no heuristic to choose one

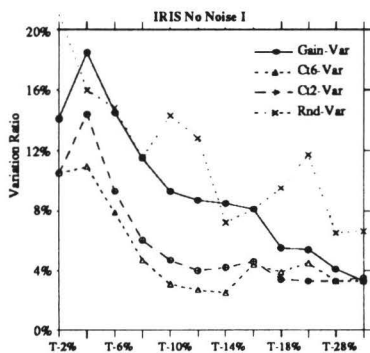


Fig.8: Iris (no noise) Variation Ratio

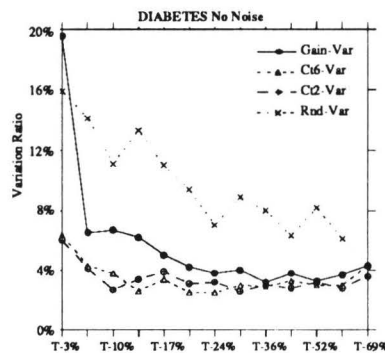


Fig.9: Diabetes (no noise) Variation Ratio

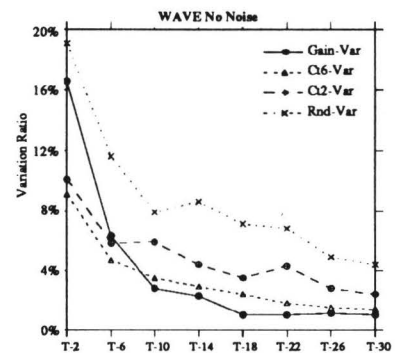


Fig.10: Wave (no noise) Variation Ratio

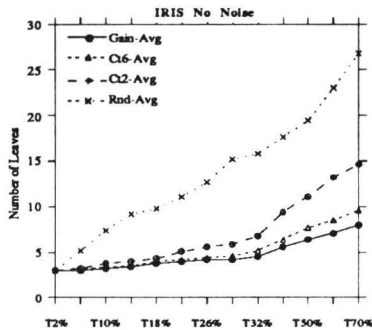


Fig.11: Iris (no noise) Complexity

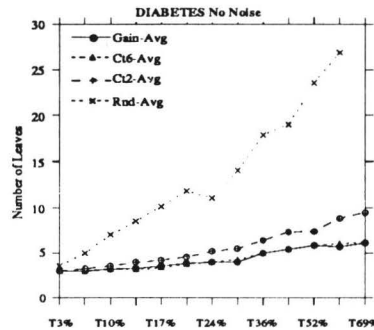


Fig.12: Diabetes (no noise) Complexity

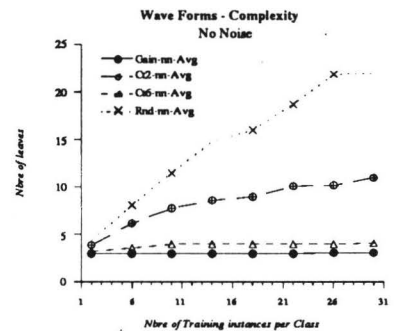


Fig.13: Wave (no noise) Complexity

among them. Inversely, Ct-2&6 select the attribute for which the distance between instances belonging to different partitions is maximum, which seems a good heuristic to apply in this case. Noise does not change much the observed results except for Random selection which improves its learning curve!

On *Diabetes*, same kinds of results as for *Iris* can be observed. However, in this case, the best performance of contrast-based criteria is significant even at large training%. Ct-2 achieve the best results on free noise sets, which is an interesting result regarding its unsupervised strategy and, as Fig.12 shows, its higher complexity! On noisy sets, Ct-6 overperforms Ct-2 because it produces less complex trees which is useful when noise is present.

On *Wave*, things are changing: Ct-2 is not so good compared to the Gain. This can be explained by the fact that the *Wave* concept is highly conjunctive: each class has a particular shape in the attribute space which is related to the *distribution* of the attribute values. In this case, the monothetic strategy of TDIDT is a big drawback for the unsupervised selection strategy of Ct-2 since, by ignoring the class information, correlation between conjunctive terms and class membership are not perceived during the divide-and-conquer search strategy. Therefore, to "discover" the conjunctive rules, it has to produce deeper trees that are more sensible to overfitting and to higher dispersions of training instances among nodes. Ct-6 behaves like the Gain criterion on this set.

From results for the three data sets, one may see that Ct-6 adapts its behavior to the best measure among Gain and Ct-2: on *Iris* and *Diabetes*, it behaves like Ct-2 and on *Wave*, like the Gain. This can be explained by the fact that when the contrast measure is not discriminatory enough, the entropy measure takes the advantage over the contrast and inversely. On free

noise cases, each criterion is the best for one of the three data sets (this is pure coincidence!) but in all cases, when noise is added, the mix similarity/class-diversity Ct-6 criterion is above the others (even if not always statistically significant).

Stabilisation Rate - Figures 8 and 9 show that, for *Iris* & *Diabetes*, in addition of being more accurate, Ct-2&6 have less accuracy deviation at low training level. They learn more quickly to stabilise their concept descriptions. In fact, on *Iris*, Ct-2&6 learn more to stabilise their concept descriptions than to improve accuracy when training% increases. It should be noted that Random selection has generally a much higher and erratic stabilisation curve. On *Wave*, the Stabilisation Rate of Ct-2 is not as good as those issued from supervised criteria. Here again, Ct-6 behaves like the Gain criterion.

When noise had been added to the data sets, no big differences have been observed on the curves. On *Iris*, nothing changed while on *Diabetes* the Gain criterion (as well as Random selection) had a better Stabilisation curve than without noise, Ct-2&6 being not affected. On *Wave*, asymptotical stabilisation rate values increased (about 5% for all criteria against 1% for free noise sets) and Ct-6 had the best decreasing rate (5% reached at T-10 against 8% for Gain and Ct-2 at the same training level).

The stability of the descriptions has been further investigated by looking at the structure of the Trees at different training levels for *Diabetes* and *Iris*. This analysis has shown that the descriptions found by Ct-2&6 are stable over time: the addition of new instances to the training set just resulted in a specialisation of previously founded descriptions (same attributes at near-top node levels), adversely to the Gain criteria whose descriptions were deeply changing whilst new instances were brought to the system. The reason for instability of Trees

issued from the Gain is that the Selection Measure only relies on counting the number of instances belonging to distinct classes. When there are only a few numbers of instances in the Training set, the addition of even a small amount of instances may drastically change the measure and hence, the decision. This is especially true when several attributes are nearly equivalent regarding entropy. Inversely, the contrast measure relies on averages of attribute values of the instances. Therefore the addition of new instances has a smoother effect.

Complexity - The complexity of Ct-6 is the same as the Gain criteria, while Ct-2, as expected, produces more complex Trees as the number of instances increases. Anyway Ct-2 has a good bias compared to Random selection whose complexity has a much higher increase rate. By analysing the Trees resulting from Ct-2&6, it has been seen that they both take the same decisions at near-top node levels. As the deep of the Tree increases, Ct-6 "moves" towards an entropy measure behavior when entropy, being closer to zero, puts a strong pressure to take low entropy level cut-points candidates. When noise had been added to the data sets, the same kinds of results have been observed: Ct-2 produces overly complex Trees while Ct-6 and Gain are very close of each other.

4 Discussion & Conclusion

In this paper we have tried to demonstrate two main points: (i) that the lack of using similarity measures in the selection of partitions during the growing phase of numerical DT may lead to produce underperforming descriptions and (ii) that using unsupervised heuristics may be useful for finding disjunctive descriptions, especially in Numerical Spaces where the combinatorial complexity of finding disjunctive terms is very high.

The good performances of Ct-2 on Iris and Diabetes seem to confirm the utility of introducing similarity measures in the selection of cut-points for numerical attributes as well as the utility of using unsupervised selection strategies to produce disjunctive descriptions. However, our tests also reveal two main limitations. Firstly, when the training set is sufficiently large (i.e., when the instance space contains so many instances that the concept boundaries are clearly defined), the use of similarity measures is less important: the differences among criteria are smaller at high training levels than at low ones, as shown on Fig. 2, 3, 5 and 6. This is probably due to the fact that at low training% there are fewer constraints on the way discriminative descriptions may be generated, making the system dependent of using other heuristics like similarity. At high training% however, the number of instances allows the system to take pure discriminative decisions with a high confidence level. Secondly, as the Wave set has demonstrated, selecting cut-points based on pure attribute similarity, without looking at class information, may lead to build overly complex trees with less predictive power. As already mentioned in section 3.2, the reason for this is probably that to efficiently discover highly conjunctive descriptions, the algorithm needs a way to identify the correlation between conjunctive terms and class membership, which is not the case for the pure unsupervised Ct-2. From these considerations, it seems that there are two main characteristics concerning the target concept that may positively influence the utility of using unsupervised

similarity measures: (i) when the target concept is partially represented by the training and (ii) when the target concept is highly disjunctive and weakly conjunctive.

Results on the Wave set with Ct-6 seem to show that this last limitation may be eliminated by combining similarity and class diversity measures. Indeed, Ct-6 seems to combine the advantages of a contrast measure on scarce training sets (for accuracy as well as stabilisation rate) and of Entropy on conjunctive concepts. Moreover, when noise is added, its performance is better than Entropy.

Future works should evaluate the impact of using Ct-2&6 on pruning. Anyway, from some preliminary tests, Ct-6 seems to take a better advantage of pruning compared to Entropy. This could be explained by the positive effect of the contrast measure on near-top level nodes. At these levels, partial concept descriptions seem to be more robust against noise than when generated using Entropy.

References

- [Aha and al., 1991] Aha W. David, Kibler D. and Albert K. M. *Instance-Based Learning Algorithms*. Machine Learning vol.6, n° 1, January 1991.
- [Breiman and al., 1984] Breiman L., Friedman J.H., Olshen R.A. and Stone C.J. *Classification And Regression Trees*. Wadsworth & Brooks, 1984.
- [Cestnik and al., 1987] Cestnik Bojan, Kononenko Igor and Bratko Ivan. *Assistant 86: A Knowledge-Elicitation Tool for Sophisticated Users*. Proceedings of the Second European Workshop on Machine Learning, 1987.
- [Clark and Niblett, 1987] Clark P. and Niblett T. *Induction in Noisy Domains*. Proceedings of the Second European Workshop on Machine Learning, 1987.
- [Gams and Lavrac, 1987] Gams M., and Lavrac N. *Review of Five Empirical Learning Systems Within a Proposed Schemata*. Proceedings of the Second European Workshop on Machine Learning, 1987.
- [Kaufman and Rouseeuw, 1990] Kaufman Leonard and Rouseeuw Peter. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience Publication, 1990.
- [Matheus and al., 1989] Matheus Christopher J. and Rendell Larry A. *Constructive Induction on Decision Trees*. Proceedings of IJCAI, 1989.
- [Mingers, 1989] Mingers J. *An Empirical Comparison of Selection Measures for Decision-Tree Induction*. Machine Learning Vol. 3 n°4, March 1989.
- [Quinlan, 1986] Quinlan J. Ross. *Induction of Decision Trees*. Machine Learning Vol 1 n°1, 1986.
- [Quinlan, 1989] Quinlan J. Ross. *Unknown Attribute values in Induction*. Proceedings of the Sixth International Workshop on Machine Learning, Morgan Kaufmann, 1989.
- [Rendell and al., 1989] Rendell L., Cho H. and Seshu R. *Improving the Design of Similarity-Based Rule-Learning Systems*. International Journal of Expert Systems Vol.2 n°1, 1989.
- [Rendell, 1988] Rendell L. *Learning Hard Concepts*. Proceedings of the Third European Workshop on Machine Learning, 1988.
- [Rendell and Cho, 1990] Rendell L. and Cho H. *Empirical Learning as a Function of Concept Character*. Machine Learning Vol. 5 n° 3, August 1990.
- [Van de Merckt, 1992] Van de Merckt T. *NFDT: A System that Learns Flexible Concepts based on Decision Trees for Numerical Attributes*. Proceedings of the Ninth International Conference on Machine Learning, Morgan Kaufmann, 1992.
- [Yang and al., 1991] Yang Der-Shung, Blix Gunnar and Rendell Larry J. *The Replication Problem: A constructive Induction Approach*. Proceedings of the Sixth European Workshop on Machine Learning, 1991.