*Journal of Universal Computer Science, vol. 19, no. 4 (2013), 484-501*
*submitted: 3/8/12, accepted: 20/2/13, appeared: 28/2/13* 

# Improving Accuracy of Decision Trees Using Clustering Techniques

**Javier Torres-Niño**
(Computer Science Department, University Carlos III Madrid
Av. Universidad, 30, Leganes, 28911, Madrid, Spain
javier.torres@uc3m.es)

**Alejandro Rodríguez-González**
(Bioinformatics at Centre for Plant Biotechnology and Genomics UPM-INIA
Parque Científico y Tecnológico de la U.P.M. Campus de Montegancedo
Pozuelo de Alarcón, 28223, Madrid, Spain
alejandro.rodriguezg@upm.es)

**Ricardo Colomo-Palacios**
(Computer Science Department, University Carlos III Madrid
Av. Universidad, 30, Leganes, 28911, Madrid, Spain
ricardo.colomo@uc3m.es)

**Enrique Jiménez-Domingo**
(Computer Science Department, University Carlos III of Madrid
Av. Universidad, 30, Leganes, 28911, Madrid, Spain
enrique.jimenez@uc3m.es)

**Giner Alor-Hernandez**
(División de Estudios de Postgrado e Investigación, Instituto Tecnológico de Orizaba
Avenida Oriente 9 No. 852 Col. Emiliano Zapata, Orizaba, 94320,Veracruz, Mexico
galor@itorizaba.edu.mx)

**Abstract:** Data mining is an important part of information management technology. Simply put, it is a method to extract and analyze meaningful patterns and correlations in a large relational database. In Data mining, Decision trees are one of the most worldwide used tools for decision support. In the emerging area of Data mining applications, users of data mining tools are faced with the problem of data sets that are comprised of large numbers of features and instances. Such kinds of data sets are not easy to handle for mining because decision trees generally depends on several parameters like dataset used and configuration of the tree itself among others in order to build an accurate model classification. In this work a novel hybrid classifier system is presented for improving accuracy of decision trees using clustering techniques. This system is formed by a clustering algorithm, a decision tree and an optional module for identifying appropriate parameters for the clustering algorithm. These three modules working together are capable to increase the accuracy of the solutions. The validation of the results of this work has been performed using several well-known datasets and applying two decision trees algorithms. The accuracy percentages are compared in order to show our proposal improvement, obtaining good results. Finally two clustering algorithms have been used to compare the accuracy between different proposals.

**Keywords:** decision tree, clustering, accuracy improvement
**Categories:** H.3.3, I.6.1, I.5.2

## 1 Introduction

Decision trees are among the best algorithms for data classification, providing good accuracy for many problems in relatively short time. Decision trees can be defined as decision support tools which use tree-like models of decision and their possible consequences [Safavian and Landgrebe 1991]. Decision trees are usually used in decision analysis problems to help in the identification of the most suitable strategy for reaching a concrete goal [Quinlan 1990]. Decision tress have been used in different areas like health [Chang & Chen 2009]), quantum computation [Bacon and van Dam 2010] and language processing [Nadkarni et al. 2011] among others. However, big datasets with big groups of very similar instances and some small groups of stranger cases can be a problem for this kind of classifier, as they tend to overlook corner cases.

To solve this kind of problem, instance selection algorithms may be used, so all groups get a similar number of instances, improving the accuracy on corner cases. Although these instance selection algorithms provide additional information about the domain, that knowledge is never used in classification itself, which could decrease the error rate.

The accuracy of decision trees has always been a problem. Several approaches have been developed to improve the quality of results provided by decision trees from different perspectives. This is the case of [Zurada 2010] or [Mahmood et al. 2010] that describe two new decision tree algorithms, called C4.45 and C4.55, to increase area under the curve over C4.5 decision tree algorithm obtaining promising and interesting results.

The use of a clustering algorithm gives a great flexibility because any clustering algorithm can be used, although the quality of the results and time of execution may vary depending on which one is used. Clustering is widely considered as one of the most important tools for unsupervised learning problems and is defined ensuring that a cluster is a group of objects which are “similar” between them and are “dissimilar” to the objects that belongs to other clusters [Hartigan 1975]. Then, the aim of clustering algorithms is to classify the data into groups that share some features between them.

A clustering algorithm should satisfy some properties to ensure its performance: scalability, discovering clusters with an arbitrary shape, capacity of deal with noise and uncertainty, high dimensionality, dealing with different kind of attributes and some requirements for domain knowledge for the determination of input parameters [Gan et al. 2007].

This technique is general enough as to be used with any classifier, but different decision tree algorithms are used during the evaluation and results section of this article, as baseline results are readily available. The combination of clustering and decision trees offers several improvements in the classification process compared with other approaches. This new hybrid approach means a novel idea to the field of classification of datasets. In this sense, the use of clustering techniques at decision trees offers significant advantages in accuracy and required computational time and is a novel idea for classification of a group of data.

In this work, we propose a hybrid classifier system, combining a clustering algorithm with a decision tree, where the former is used both as an instance selection

algorithm and a front-end classifier, so only instances that are not classified by it are fed to the decision tree.

The paper consists of five sections and is structured as follows. Section 2 surveys the relevant literature about the described problem. Section 3 shows the architecture of the proposed system. Section 4 shows the evaluation of the system. Finally, the paper ends with a discussion of research findings, limitations and concluding remarks.

## 2 State of the art

Nowadays, clustering is a widespread technique that it is applied to different fields and ambits; marketing [Kazienko 2008], insurance [Pasierb et al. 2011], health [Holzinger et al. 2008], biology [Bhattacharya et al. 2012], advertisement recommendation [Rodríguez-González et al. 2012] and classification [Kajdanowicz et al. 2010] among others. Especially relevant are some works centered in the classification of environmental situations such as [Rännar and Andersson 2010] and [Stern 2010]

The first work tries to make a four-step strategy based on principal component analysis and hierarchical clustering, for selecting structurally dissimilar organic substances from a list of commercial, high volume production chemicals while the second paper use a multi-agent system for clustering environmental data, solving the problem of the increasing amount of information in this field.

There are some previous works using clustering algorithms in combination with other techniques to select instances, such as the one proposed by [Wang and Chiang 2009]. In this case, a preprocessor is used to select a subset of instances to use in a clustering-based classification model, in order to improve its results. Similarly, [Kajdanowicz et al. 2011], goes into detail into an instance selection algorithm based in entropy measures of the dataset in order to choose the optimal instances for good classification results.

Other works have proposed hybrid classifier systems combining different artificial intelligence techniques with decision trees. For instance, [Pei-Chann et al. 2010] presents a hybrid classification model by integrating a case-based reasoning technique, a fuzzy decision tree (FDT), and genetic algorithms (GAs) to construct a decision-making system for data classification in various database applications. The model is major based on the idea that the historic database can be transformed into a smaller case base together with a group of fuzzy decision rules. As a result, the model can be more accurately respond to the current data under classifying from the inductions by these smaller case-based fuzzy decision trees.

[Kuang 2011] proposes a new hybrid cluster validity method based on particle swarm optimization, for successfully solving one of the most popular clustering/classifying complex datasets problems. The proposed method for the solution of the clustering/classifying problem, designated as PSORS index method, combines a particle swarm optimization (PSO) algorithm, Rough Set (RS) theory and a modified form of the Huang index function. In contrast to the Huang index method which simply assigns a constant number of clusters to each attribute, this method could cluster the values of the individual attributes within the dataset and achieves both the optimal number of clusters and the optimal classification accuracy.

[Karaboga and Ozturk 2009] propose an Artificial Bee Colony (ABC) algorithm which is applied to classification benchmark problems (13 typical test databases). The performance of the ABC algorithm on clustering is compared with the results of the Particle Swarm Optimization (PSO). ABC and PSO algorithms drop in the same class of artificial intelligence optimization algorithms, population-based algorithms and they are proposed by inspiration of swarm intelligence.

[Kashef and Kamel 2009] present a cooperative bisecting k-means (CBKM) clustering algorithm. The CBKM concurrently combines the results of the BKM (Bisecting K-means) and KM (K-means) at each level of the binary hierarchical tree using cooperative and merging matrices. Undertaken experimental results show that the CBKM achieves better clustering quality than that of KM, BKM, and single linkage (SL) algorithms with comparable time performance over a number of artificial, text documents, and gene expression datasets.

[Zho and Chen 2002] present a novel machine learning approach named hybrid decision tree (HDT) that virtually embeds feed forward neural network in some leaves of a binary decision tree which is motivated by recognizing that dealing with unordered/ordered attributes is similar to performing qualitative/quantitative analysis. HDT employs unique techniques of tree growing, neural processing, incremental learning and constructive induction, which enables it to generate accurate and compact HDTs and deal gracefully with new appended data.

[Chin-Yuan et al. 2011] propose a hybrid model developed by integrating a case-based data clustering method and a fuzzy decision tree for medical data classification. Two datasets from UCI Machine Learning Repository were employed for benchmark test. Initially a case-based clustering method was applied to preprocess the dataset thus a more homogeneous data within each cluster will be attainted. A fuzzy decision tree was then applied to the data in each cluster and genetic algorithms (GAs) were further applied to construct a decision-making system based on the selected features and diseases identified. Finally, a set of fuzzy decision rules were generated for each cluster.

[Ruey-Shiang et al. 2011] presents a hybrid intelligence method which integrating genetic algorithm and decision learning techniques for knowledge mining of an in vitro fertilization (IVF) medical database. The proposed method can not only assist the IVF physician in predicting the IVF outcome, but also find useful knowledge that can help the IVF physician tailor the IVF treatment to the individual patient with the aim of improving the pregnancy success rate. [Shukla and Tiwari 2009] propose a novel methodology, genetically optimized cluster oriented soft decision trees (GCSDT), to glean vital information imbedded in the large databases. In contrast to the standard C-fuzzy decision trees, where granules are developed through fuzzy (soft) clustering, in the proposed architecture granules are developed by means of genetically optimized soft clustering. In the GCSDT architecture, GA ameliorates the difficulty of choosing an initialization for the fuzzy clustering algorithm and always avoids degenerate partitions. This provides an effective means for the optimization of clustering criterion, where an objective function can be illustrated in terms of cluster's center. Growth of the GCSDT is realized by expanding nodes of the tree, characterized by the highest inconsistency index of the information granules. In order to validate the proposed tree structure it has been deployed on synthetic and machine learning data sets.

A similar hybrid system has been explored by [Kajdanowicz and Kazienko 2009], where it is proven that a combination of clustering of decision trees, albeit in a different way to what is proposed here, can improve the quality of results in classification tasks.

[Ali et al. 2009] also proposes a hybrid method using clustering as a preprocessing stage and decision trees as the main classifier, applied in a dialogue system. A similar, but more generic system is also presented in [Kazienko 2000]. There is however a crucial difference with our work, they use a decision tree to classify instances in each cluster; we use a single decision tree for all instances, making our system lighter and faster.

Similar concepts as the ones presented here have also been applied to clustering algorithms, in order to use them as classifiers [Barak et al. 2011], where a threshold is also used to discriminate clusters to use in classification. However, the classification is only based on this, and do not uses a hybrid method, as in this work.

Despite clustering being a popular and commonly used technique that is applied to different fields, no works have been reported in which clustering is used to improve the accuracy of decision trees. For this reason, this paper makes an original an important contribution by creating a complete brand new system that allows improving the previous results obtained with the use of these techniques.

## 3 Architecture of the hybrid classifier system

Our developed system is divided in three main parts: 1) clustering module (including a clustering algorithm), 2) decision tree module; and 3) an optional module that identifies good parameters to use in the clustering algorithm. In figure 1, the clustering module is firstly used to build a clustering model by using the underlying algorithm. After this step, the decision tree module creates two models: 1) one using the whole dataset and, 2) a second one with only the unclassified instances from the clustering module. Next, the system is ready for classifying instances, which works in a similar way to a training process, and it is further described in section 3.2.

The parameter adjuster is separated from the system, and it can be used to automatically obtain good parameters for the clustering module, without the necessity of build the full model. It takes the dataset as an input and produces a set of parameters for the clustering module, which can be automatically used or presented to the user for manual configuration.

### 3.1 Clustering Module

The clustering module uses any clustering algorithm as an incomplete classifier, meaning that some of the instances will not be assigned to any class. The basic idea is to tag some clusters created by the underlying clustering algorithm with a class, so all the instances in that cluster are assigned to that class.

#### 3.1.1 Building the model

A clustering model is built on training data, removing the class attribute as that is the one we are trying to predict. Then, each cluster of the model is evaluated to check if it

contains a majority of instances of a single class. For doing this, we count the number of instances of each class that is assigned to each cluster.

For each class and cluster we compute the proportion of instances of that class against the total number of instances assigned to that cluster. If it exceeds a threshold, we consider that this cluster is classified. Any cluster where no class exceeds the threshold is marked as non-classified.

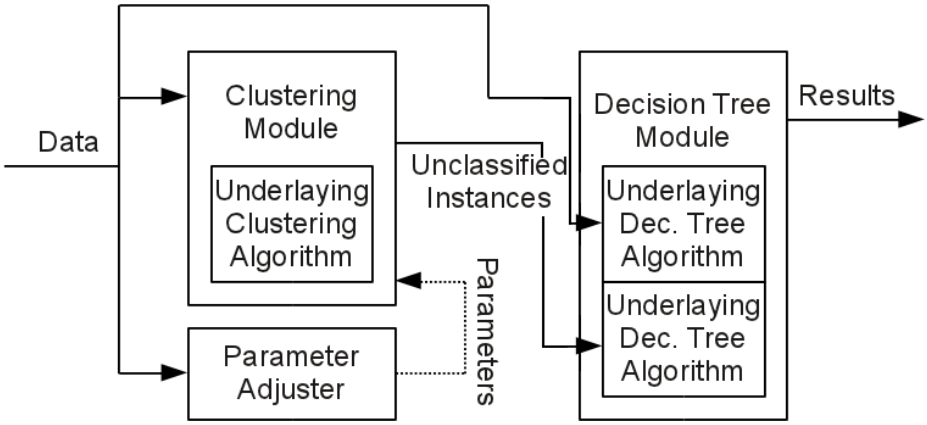


*Figure 1: Architecture of the hybrid classifier system*

### 3.1.2 Using the model

In order to classify an instance, the first step is to pass it to the underlying clustering algorithm, assigning a cluster to the instance using the preexisting clustering model. Then, the cluster is looked up in the table created during the construction of the model, to check if that cluster is marked of classifying for any class. The instance is classified with the same class of the cluster, or, if the cluster is marked as non-classifying, the instance is left without any class.

Easy is the name or alias that receives the instances that were classified by the clustering module. On the other hand, hard is the name or alias that receive the ones that were left without classify. The clustering module usually classifies common instances that are usually classified correctly by any classifier.

## 3.2 Decision Tree Module

This module automates the construction of decision tree models, and it is a wrapper around any decision tree algorithm. It builds two different models and uses each one of them to get a total of 4 different classifiers.

The model is composed of two different decision tree models using the same algorithm. One of them is trained on the entirety of the training set, while the other is trained only on the training instances unclassified by the clustering module.

Each one of the decision trees is then evaluated using both the testing set, and the instances of the testing set that were not classified by the clustering module. This process creates a total of 4 classifiers:

(1) The decision tree alone. This is the baseline for comparisons.
(2) The decision tree trained on "hard" instances and used to classify all instances.
(3) The model trained on all instances but only used to classify "hard" instances.
(4) The model trained on "hard" instances and used to classify "hard" instances.

The first method is just a decision tree, while the second one is equivalent to using an instance selection method to reduce the number of instances used to train the decision tree model. However, it is not the same as the normal way to do instance selection using clustering, where some instances of each cluster are chosen, while in the method proposed here, the selection decision is made on a cluster level. Methods 3 and 4 are the proposed new ways to mix clustering and decision trees to create better classifiers.

Code listing 1 illustrates how to use a clustering algorithm to classify instances, and mark them as "hard" or "easy". The difference between the methods is how to use these instances in the decision tree:

- **Training:** Methods 1 and 3 train on all instances (Tr), Methods 2 and 4 in "hard" instances (unclassified by the clustering algorithm).
- **Classification:** Methods 1 and 2 use only the decision tree for classification. Methods 3 and 4 first try to classify using the clustering algorithm, and apply the decision tree only to "hard" (unclassified) instances.

We have not figured out any general way to precisely select which of the three new models is more appropriate for each dataset without evaluating them, so all of them are evaluated and the one with the lowest error rate is chosen as the final classifier. There seems to be some correlation between the best method and the correct/error ratio on the clustering output, explained on detail on the evaluation section.

```
Split DS into training set (Tr) and test set (Ts) (for each
cross-validation fold)
CA = CreateCluster(Tr, Parameters)
foreach Cluster in CA
  Cluster.Class = -1
  foreach Class in Dataset:
    n = Count(Instances of Class in Cluster)
    if (n/len(Cluster.Instances) > Threhold)
      Cluster.Class = Class
  if Cluster.Class = None:
    Cluster.HardInstances += Cluster.Instances
```

*Code Listing 1. Cluster Classifier*

### 3.1 Parameter Adjuster Module

This module offers a quick way to test which sets of parameters are more adequate for a given dataset, without having to run the whole classifier for each set of parameters. This is achieved by using a heuristic function on the results (number of correctly,

incorrectly and unknown instances) of the training set, to evaluate a priori which sets of parameters create better results.

This module employs the list of parameter values to test, that depends on the clustering algorithm. All of them take a classification threshold as defined in 3.1, but a list of algorithm-dependent parameters is also employed (e.g.: the number of clusters for K-Means or the minimum and maximum clusters for X-Means).

With this data, the module builds a clustering model for each set of algorithm-dependent values to test. For each of these models, the module calculates the classifying clusters, as defined in 3.1, for each threshold to test. Then, the training set is evaluated in all of the resulting models. This process is detailed in Code Listing 2.

```
Best = 0
foreach Param in Parameters:
  foreach Thr in Thresholds:
    CA = CreateCluster(Tr, Param)
    foreach Cluster in CA
      Cluster.Class = -1
      foreach Class in Dataset:
        n = Count(Instances of Class in Cluster)
        if (n/len(Cluster.Instances) > Thr)
          Cluster.Class = Class
    if Cluster.Class != None:
      foreach Instance in Cluster:
        if Instance.Class = Cluster.Class: c++
        else:  e++
    h = (c-2e)/(Tr.NumInstances * (Param.Nc)^0.1)
    if h > Best.h:
      Best.h = h
      Best.Parameters = Param
      Best.Threshold = Thr

Use Best for the real model
```

*Code Listing 2. Parameter adjuster*

Finally, a heuristic function is applied to all those results, and the set of parameters with the highest heuristic value is selected as a good candidate for use in the complete classifier. This process provides the parameters for the clustering algorithm as well as the threshold that has been identified as the best candidate. The heuristic value is computed as shown in Equation 1, where c is the number of correctly classified instances, e the number of incorrectly classified instances, $N_i$ the total number of instances, and $N_c$ the number of clusters.

$$h = \frac{c - 2e}{N_i \sqrt[10]{N_c}}$$

*Equation 1: Heuristic function*

The objective of the heuristic function is not only to maximize the number of correctly classified instances while minimizing the classification error (the dividend), but also to maintain a feasible resource usage (the root of $N_c$). The number 10 is used to control resource usage: while 10 is the recommended value, it can be increased when resources are plentiful and decreased when resources are sparse. This function can easily be modified depending on the target metric to optimize.

## 4 Evaluation

In this section, we evaluate the performance of our system, comparing against the bare decision trees, in order to determine if the added complexity improves the results.

### 4.1 System evaluation

The system was evaluated using K-means as the clustering algorithm and C4.5 as the decision tree, using different datasets. Different datasets have been used for different parts of the evaluation, but they are all summarized in Table 1.

| Data set | Instances | Attributes | Classes |
|---|---|---|---|
| **breast-w** [Zwitter and Soklic, 1998] | 699 | 10 | 2 |
| **credit-a** [Credit Approval Data Set, 2011] | 690 | 16 | 2 |
| **hepatitis** [Gong and Cestnik 1988] | 155 | 20 | 2 |
| **labor** [Matwin, 1988] | 57 | 17 | 2 |
| **ionosphere** [Sigillito 1989] | 351 | 35 | 2 |
| **car** [Bohanec 1997] | 1728 | 6 | 4 |
| **adult** [Kohavi and Becker 1996] | 48842 | 14 | 2 |
| **magic04** [Bock 2007] | 19020 | 11 | 3 |

*Table 1: Datasets used for the system evaluation*

The first five datasets are used during parameter adjustment and heuristic function development, and are chosen for being small and thus, fast to work with. The 4 last datasets are larger and used to evaluate the system as a whole, using the knowledge previously obtained with the other datasets.

### 4.2 Parameter Investigation

In this subsection we try to investigate the relationship between the parameters of the clustering algorithm and the results of the classifier. Although the results are different

with different datasets and algorithms, the conclusions that can be extracted are the same.

We start by plotting the increase in accuracy of each method compared to the decision tree alone, changing the number of clusters, in Figure 2.

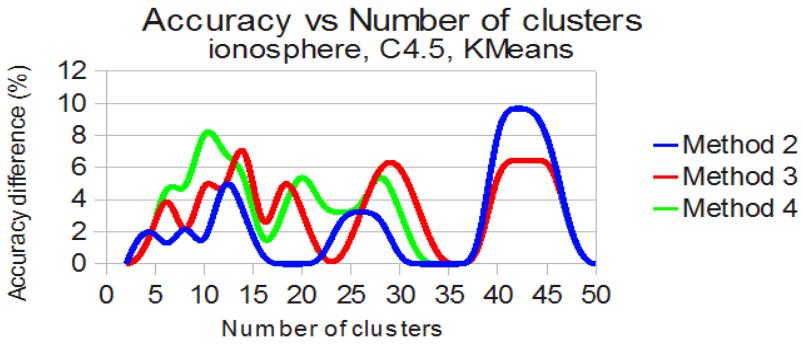


*Figure 2: Results vs Number of Clusters*

By comparing all methods we observer very similar behavior, which motivated us to try a heuristic function in the first place, as it will produce values valid for all three methods.

The general behavior is a good start (peaking at about 15 clusters), a decrease in accuracy as clusters are added and occasional peaks (around 30 and 45). As this peaks can prove difficult to predict and be unreliable as they seem to proceed from local maximums in the clustering algorithm, we added a factor to the heuristic function to reward a low number of clusters.

This also improves performance, as models with less clusters are faster to generate than more complex ones. Furthermore, it reduces the risk of overfitting that comes with a more complex model. However, too few clusters and the clustering model does not add nothing to the overall model, making it an unnecessary step.

The ideal number of clusters is usually in the low end of the spectrum, between 3 and 10 times the number of classes in the dataset, allowing for enough clusters for some of them to specialize in certain classes; while maintaining the complexity low.

Repeating the experiment but changing the threshold instead of the number of clusters yields the results are shown in Figure 3.

Here, methods 2 and 3 are unsurprisingly very similar in behavior, but method 1 behaves differently by having very bad results for low thresholds. In all cases, accuracy in-creases with the threshold (as this causes fewer errors in the clustering classifier), up to a point where the accuracy starts to drop, signaling that the clustering module is not classifying many instances, leaving all the work to the decision tree. Threshold of 100% is mostly equal to a decision tree alone, as the clustering module will only classify instances that fall in a cluster entirely of one class, which is very rare.

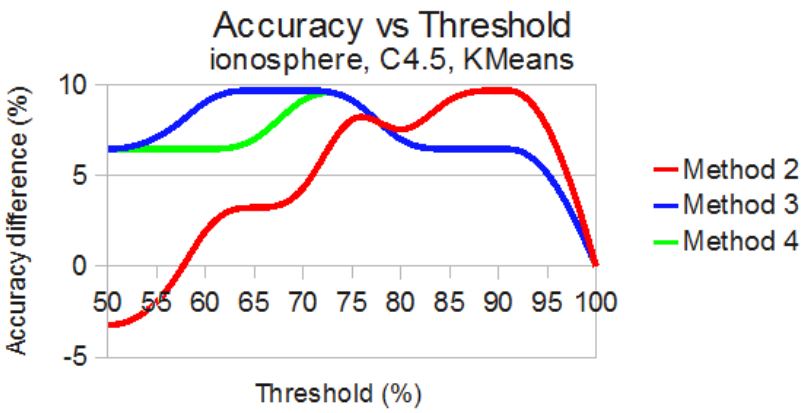


*Figure 3: Results vs Threshold*

The optimal value of the threshold is the bigger value that can be used without starting to drop accuracy, with is usually a little more than the decision tree accuracy when used alone.

Good values are usually between 2 and 10% more than the accuracy of a decision tree over the dataset. Too high, and the clustering model is unused; too low and the clustering model will lower the quality of the results.

Even though it seems that setting the threshold over the accuracy will always produce better results than the decision tree alone, as the clustering will only use clusters better than the tree, it is important to note that averages are not an accurate measure of the behaviour over all dataset. Some classes will always produce better results than the average, and if the threshold is between the global average and the class average, it will usually make the results of that class worse, as the threshold is below its original accuracy.

This also highlights one benefit of this system in certain cases: it helps balance the confusion matrix, making it more homogeneous, which can be useful in some use cases.

### 4.3 Heuristic function

To evaluate the heuristic function, we compare the predicted parameters with the actual best parameters, as detailed on Tables 2 and 3.

Table 3 shows the results using the formula specified in section 3.3, while Table 2 shows the result using the same heuristic but without the Nc root factor. We can see how the division by the number of clusters makes the heuristic choose smaller values for the number of clusters, slightly improving accuracy (as first observed in 4.2) and reducing the computing time.

| Dataset | Predicted | | | Real | | |
|---|---|---|---|---|---|---|
| | Nc | Thr | Acc | Nc | Thr | Acc |
| **breast-w** | 30 | 75% | 96,43% | 22-28 | 85% | 98,57% |
| **credit-a** | 35 | 75% | 89,13% | 14 | 90% | 89,13% |
| **Hepatitis** | 10 | 75% | 87,10% | 9-15 | 80% | 87,10% |
| **Ionosphere** | 35 | 75% | 83,21% | 12 | 85% | 95,77% |
| **Labor** | 25 | 75% | 91,66% | | | 100% |

*Table 2: Heuristic predictions, no Nc factor*

| Dataset | Predicted | | | Real | | |
|---|---|---|---|---|---|---|
| | Nc | Thr | Acc | Nc | Thr | Acc |
| **breast-w** | 4 | 75% | 97,86% | 22-28 | 85% | 98,57% |
| **credit-a** | 35 | 75% | 89,13% | 14 | 90% | 89,13% |
| **Hepatitis** | 6 | 75% | 87,10% | 9-15 | 80% | 87,10% |
| **Ionosphere** | 30 | 90% | 91,55% | 12 | 85% | 95,77% |
| **Labor** | 35 | 75% | 95,66% | | | 100% |

*Table 3: Heuristic predictions, final equation*

In general, our heuristic does provide values that produce results nearing the best ones that can be obtained with this system. This allows quick experimentation to determine if the system is appropriate to use with a given dataset. Nevertheless, the heuristic function can never be completely trusted, as is shown in the labor dataset, where most combinations of parameters result in 100% accuracy, yet the heuristic function chooses a too big number of clusters, resulting in less than optimal results.

### 4.4 Overall system accuracy

The data from the datasets is partitioned before being used, using a 10% of the data for adjusting the parameter, and the rest for a 10-fold cross-validation, which implies that each fold uses 81% of the total data of the data set for training and 9% for testing.

To test the system performance against that of the original decision tree, we use the last four data sets: ionosphere, car, adult and magic04. For each dataset, four

variants, according to whether two options are used or not: balancing the classes and adding noise (10% of the class labels are randomly changed).

| Dataset | Original | Balanced | Balanced & Noise | Noise |
|---|---|---|---|---|
| **ionosphere** | 3 (7.3%) | 1 (9.6%) | 1 (1.8%) | 2 (8.7%) |
| **car** | 2 (1.5%) | 3 (3.8%) | 3 (1.8%) | 4 (8.3%) |
| **adult** | 1 (3.4%) | 4 (3.0%) | | |
| **magic04** | | 3 (1.9%) | 4 (3.8%) | 2 (2.8%) |

*Table 4: Method that produced the best resultsand relative improvement (blank if differences are not significant)*

| Dataset | Decision tree | | Best method | | |
|---|---|---|---|---|---|
| | Avg. | Std. Dev. | Parameters | Avg. | Std. Dev. |
| **ionosphere** | 0.125238 | 0.065084 | M3, 10c, 90% | 0.099523 | 0.067425 |
| **car** | 0.221138 | 0.0970920 | M2, 12c, 80% | 0.217680 | 0.090618 |
| **adult** | 0.150041 | 0.014947 | M4, 14c, 90% | 0.153026 | 0.013264 |
| **magic04** | 0.118820 | 0.011498 | M2, 20c, 95% | 0.119083 | 0.011658 |

*Table 5: Results for the original datasets*

| Method | Avg. | Std. Dev. |
|---|---|---|
| **1** | 0.125223 | 0.065084 |
| **2** | 0.110952 | 0.054379 |
| **3** | 0.099523 | 0.067425 |
| **4** | 0.236507 | 0.123974 |

*Table 6: Results for the ionosphere dataset 10 clusters, 90% threshold*

Results are show in Tables 4 to 6. Table 4 shows the method that obtained the best results for each dataset, or a blank cell if there are no significant differences at

95% confidence. It also includes the relative difference between the best new method and the decision tree.

Table 5 shows the details of the data summarized in the previous table for the original datasets, showing the average error and standard deviation of the 10 folds of the cross-validation, as well as the parameters used to obtain the results (determined by the parameter adjuster module), that is: method, number of clusters and threshold. Finally Table 6 shows the details for the ionosphere dataset.

In most cases, the new system significantly improves the results over the decision tree alone (method 1) in up to 3% in absolute error, or almost 9% relative to the accuracy. However, in most cases, the increase is smaller, of around 0.5% in accuracy, or a relative difference around 3%. All methods are the best for at least one dataset, but the results for each of them depend on the dataset. There is no clear overall best and we have not found a way to predict which is the best before executing them.

The performance of the proposed method is very irregular as shown in the large variances between methods in the ionosphere dataset (Table 6). Thus, it is important for good performance to run all methods to see which one is more apt for each situation, as some configurations make the results much worse than a plain decision tree. Thankfully, the heuristic function reduces the search space to configure the algorithm, but some experimentation is still required.

For some datasets, especially in noisy environments, the proposed methods offer a good improvement over the decision tree. Of all noisy cases, in all except one, the proposed method offered better or equivalent performance to that of the decision tree alone. This makes this technique especially interesting for such cases, using methods 2 and 4, where some noise will be removed from the decision tree training set, improving the model generated by the tree.

### 4.5 Performance analysis

The performance of this system greatly depends on the chosen algorithms. As the system trains one clustering model and two decision trees, building the complete system model takes almost as much time as the training of two decision trees and a clustering model. Some time is saved as the second decision tree does not train on all training instances.

When evaluating instances, the system is also slower, but not in such a noticeable way as in the previous step. In the worst case (an instance is passed through both classifiers) the evaluation time is the sum of the time spent by the clustering and the decision tree.

Space-wise, once one of the methods is chosen, only a decision tree and a clustering model need to be kept in memory, still bigger than a single decision tree.

In conclusion, the system performance is not as good as decision trees, but as clustering and trees are two of the fastest techniques, it still can rival against more complicated techniques as neural networks. Also, for the cost of training three primitive models, you get four composite models, amortizing a bit the additional computational cost.

## 5 Conclusions and future work

In this paper we have presented a hybrid classification system that improves classification accuracy of any given decision tree algorithm by combining it with a clustering algorithm. The results exceeded our expectation, since clustering algorithms operate blindly (i.e. not taking the class into account) over the data, but yet manage to improve the accuracy of the system greatly, when compared to the basic system.

Interestingly, this method is very useful in noisy environments, one of the weaknesses of decision trees, which often get confused by noise. Filtering the training instances with the clustering algorithm seems to eliminate some noise from the decision tree training set, simplifying and improving the model it generates.

Nevertheless, more experimentation is needed, but this work shows promises given the positive results founded. As future work, we will be centered in a more exhaustive evaluation trying to check all the possible configuration values for the algorithms in order to try to improve the accuracy results. It is also interesting to try to combine different algorithms, such as the clustering algorithm in Sert et al. (2012), which takes into account classes when learning the model. Cluster validity indexes could also be used in order to select better clusters when building the clustering model.

## References

[Ali et al. 2009] Ali, S. A., Sulaiman, N., Mustapha, A., & Mustapha, N. (2009). K-Meaiis Clustering to Improve the Accuracy of Decision Tree Response Classification. Information technology journal, 8(8), 1256-1262.

[Bacon and van Dam 2010] Bacon, D. and van Dam, W. “Recent progress in quantum algorithms”; Commun. ACM53, 2 (2010), 84-93.

[Barak et al. 2011] Barak A., Gelbard R., “Classification by clustering decision tree-like classifier based on adjusted clusters”; Expert Systems with Applications, 38, 7, 2011, 8220-8228.

[Bhattacharya et al. 2012] Bhattacharya, A., Chowdhury, N. and De Rajat, K. “Comparative Analysis of Clustering and Biclustering Algorithms for Grouping of Genes: Co-Function and Co-Regulation”; Current Bioinformatics, 7, 1 (2012), 63-76.

[Bock 2007] Bock, R. K. “MAGIC Gamma Telescope Data Set”; Major Atmospheric Gamma Imaging Cherenkov Telescope project (MAGIC). (2007)

[Bohanec 1997] Bohanec, M. “Car evaluation data set”. (1997)

[Chang and Chen 2009] Chang, C. and Chen, C. “Applying decision trees and neural network to increase quality of dermatologic diagnosis”; Expert Systems with Applications. 36, 2, (2009), 4035-4041.

[Chin-Yuan et al. 2011] Chin-Yuan, F., Pei-Chann, C., Jyun-Jie, L. and Hsieh, J.C. “A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification”; Applied Soft Computing, 11, (2011), 632–644

[Credit Approval Data Set 2011] "Credit Approval Data Set"; (2011). Last accessed: January, 14. Available online at: http://archive.ics.uci.edu/ml/datasets/Credit+Approval

[Gan et al. 2007] Gan, G., Ma, C. and Wu, J. "Data Clustering: Theory, Algorithms, and Applications"; ASASIAM Series on Statistics and Applied Probability, 20, 466, (2007).

[Gong and Cestnik 1988] Gong, G. and Cestnik, B. "Hepatitis Data Set"; Jozef Stefan Institute/Yugoslavia, (1988),

[Hartigan 1975] Hartigan, J.A. "Clustering Algorithms"; 99th. John Wiley & Sons, Inc. 1975

[Holzinger et al. 2008] Holzinger, A., Geierhofer, R., Mödritscher, F. and Tatzl, R. "Semantic Information in Medical Information Systems: Utilization of Text Mining Techniques to Analyze Medical Diagnoses"; Journal of Universal Computer Science, 14, 22, (2008), 3781-3795.

[Kajdanowicz and Kazienko 2009] Kajdanowicz, T., and Kazienko, P. "Hybrid Repayment Prediction for Debt Portfolio"; In Computational Collective Intelligence, Semantic Web, Social Networks and Multiagent Systems, (2009), 850–857.

[Kajdanowicz et al. 2010] Kajdanowicz, T., Kazienko, P. and Doskocz, P. "Label-Dependent Feature Extraction in Social Networks for Node Classification"; Lecture Notes in Computer Science, Social Informatics, 6430, (2010), 89-102.

[Kajdanowicz et al. 2011] Kajdanowicz, T., Plamowski, S., and Kazienko, P. "Training set selection using entropy based distance"; In IEEE Jordan Conference On Applied Electrical Engineering and Computing Technologies (AEECT), (2011), 1 –5.

[Karaboga and Ozturk 2009] Karaboga, D. and Ozturk, C. "A novel clustering approach: Artificial Bee Colony (ABC) algorithm"; Applied Soft Computing, 11, (2009), 652–657.

[Kashef and Kamel 2009] Kashef, R. and Kamel, M.S. "Enhanced bisecting k-means clustering using intermediate cooperation"; Pattern Recognition, 42, (2009), 2557-2569.

[Kazienko 2008] Kaienko, P. "Web-Based Recommender Systems and User Needs --the Comprehensive View"; In proceedings of the 2008 Conference on New Trends in Multimedia and Network Information Systems, (2008), 245-258.

[Kazienko 2000] Kuncheva, L. I. (2000). Clustering-and-selection model for classifier combination. In Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on (Vol. 1, pp. 185-188). IEEE.

[Kedes and Birnstiel 1971] Kedes, L.H. and Birnstiel, M.L. "Reiteration and Clustering of DNA Sequences Complementary to Histone Messenger RNA"; Nature new biology, 230, (1971), 165-169.

[Kohavi and Becker 1996] Kovahi, R. and Becker, B. "Adult Data Set", Data Mining and Visualization, Silicon Graphics. (1996).

[Kuang 2011] Kuang, Y.H. "A hybrid particle swarm optimization approach for clustering and classification of datasets"; Knowledge-Based Systems, 24, (2011), 420–426.

[Mahmood et al. 2010] Mahmood, A., Rao, K.M. and Reddi, K. "A Novel Algorithm for Scaling Up the Accuracy of Decision Trees"; International Journal of Computer Science and Engineering, 2, 2, (2010), 126-131.

[Matwin 1988] Matwin, S. “Final settlements in labor negotiations in Canadian Industry Data Set”; University of Ottawa, (1988).

[Nadkarni et al. 2011] Nadkarni, P., Ohno-Machado, L. and Chapman, W. “Natural Language processing: an introduction”; Journal of the American Medical Informatics Association, 18, (2011), 544-551.

[Pasierb et al. 2010] Pasierb, K., Kajdanowicz, T. and Kazienko, P. “Privacy-Preserving Data Mining, Sharing and Publishing”; Journal of Medical Informatics & Technologies, 18, (2011).

[Pei-Chann. C. et al. 2011] Pei-Chann, C., Chin-Yuan, F. and Wei-Yuan, D. “A CBR-based fuzzy decision tree approach for database classification”; Expert Systems with Applications, 37, (2011), 214–225.

[Pelleg and Moore 2000] Pelleg, D. and Moore, A. “X-means: Extending K-means with Efficient Estimation of the Number of Clusters”; In ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning, (2000).

[Quinlan 1990] Quinlan, J.R. “Decision Trees and Decision-making”; IEEE Transactions on Systems, Men and Cybernetics, 20, 2, (1990).

[Rännar and Andersson 2010] Rännar, S. and Andersson, P.L. “A Novel Approach Using Hierarchical Clustering to Select Industrial Chemicals for Environmental Impact Assessment”; Journal of Chemical Information Models, 50, 1, (2010), 30-36.

[Rodríguez-González et al. 2012] Rodríguez González, A., Torres-Niño, J., Jiménez-Domingo, E., Gómez-Berbís, J.M., Alor-Hernandez, G. "AKNOBAS: A knowledge-based segmentation recommender system based on intelligent data mining techniques"; Computer Science and Information Systems, 9(2), (2012), 713-740

[Ruey- Shiang et al. 2011] Ruey-Shiang, G. Tsung-Chieh, J. and Shao-Ping, W. “Integrating genetic algorithm and decision tree learning for assistance in predicting in vitro fertilization outcomes”; Expert Systems with Applications, 38, (2011), 4437–4449.

[Safavian and Landgrebe 1991] Safavian, S.R. and Landgrebe, D. “A survey of decision tree classifier methodology”; IEEE Transactions on Systems, Men and Cybernetics, 21, 3, (1991).

[Sert et al. 2012] Sert, O.C., Dursun, K., Özyer, T., Jida, J., and Alhajj, R. “The Unification and Assessment of Multi-Objective Clustering Results of Categorical Datasets with H-Confidence Metric”; Journal of Universal Computer Science 18, (2012), 507–531.

[Shukla and Tiwari 2009] Shukla, S.K. and Tiwari, M.K. “Soft decision trees: A genetically optimized cluster oriented approach”; Expert Systems with Applications, 36, (2009), 551–563.

[Sigilito 1989] Sigillito, V. “Ionosphere Data Set”; Applied Physics Laboratory, Johns Hopkins University, (1989).

[Stern 2010] Stern, C. “Clustering of environmental data using a Multi-agent System”; 13th AGILE International Conference on Geographic Information Science 2010, Guimaraes, Portugal, (2010).

[Wang and Chiang 2009] Wang, J.-S., and Chiang, J.-C. “An Efficient Data Preprocessing Procedure for Support Vector Clustering”; Journal of Universal Computer Science 15, (2009), 705–721.

[Xing and Bao-Gang 2008] Xing, H.J. and Bao-Gang, H. “An adaptive fuzzy c-means clustering-based mixtures of experts model for unlabeled data classification”; Neurocomputing, 71, 4-6, (2008), 1008-1021.

[Xho and Chen 2002] Zho, Z.H. and Chen, Z.Q. “Hybrid Decision Tree, Knowledge-Based Systems”, 15, (2002), 515-528.

[Zurada 2010] Zurada, J. “Could Decision Trees Improve the Classification Accuracy and Interpretability of Loan Granting Decisions?”; In International Conference of System Sciences (HICSS), Hawaii, 5-8 Jan, (2010).

[Zwitter and Soklic 1998] Zwitter, M. and Soklic, M. “Breast Cancer Data Set”, Institute of Oncology, University Medical Center. (1998).