Interactive Construction of Classification Trees Using Treemaps

Manoel Gomes de Mendonça Neto
Universidade Salvador, Nuperc,
Salvador, Brasil, 40171-100
mgmn@unifacs.br

and

Daniela Soares Cruzes
Universidade Salvador, Nuperc,
Salvador, Brasil, 40171-100
daniela@unifacs.br

and

Christiane da Costa Santana
Universidade Salvador, Nuperc,
Salvador, Brasil, 40171-100
christiane.santana@unifacs.br

Abstract

Most of the approaches published in the literature proposes a completely automatic process to generate decision trees. These approaches miss valuable expert tacit knowledge input during the construction of the tree. This paper describes an approach for interactive construction of decision trees. The approach is user-centered. It combines the strengths of the user and the computer to build better decision trees. The user provides domain knowledge and evaluates intermediate results of the algorithm. The computer automatically creates patterns satisfying user constraints and generates appropriate visualizations of the produced tree. A tool was developed to support this approach. It combines treemap visualization, visual data mining mechanisms, and the J48 (Weka) algorithm to interactively build a decision tree.
Keywords: Visual Data Mining, Decision Tree, Classification.

1. INTRODUCTION

During the last decade, data repositories has grown faster than our ability to analyze them. The area of Knowledge Discovery from Databases (KDD) has appeared as an attempt to balance this equation. KDD aims to extract non trivial, previously unknown and potentially useful information from data repositories [9],[15]. KDD is a highly interactive process that goes from the definition of the analysis goals to the extraction and assimilation of knowledge from the data repository.

The main activity of this process is data mining. Data mining is characterized by the use of algorithms to extract useful knowledge from pre-processed data sets. Classification is one of the most common tasks in data mining [15], and the construction of decision trees is one of the most popular classification methods. Decision Trees are intuitive, easy to interpret; relatively fast to construct and, compared to other classification methods, have equal or better accuracy [8].

Many approaches for decision trees construction have been proposed in the literature. However, most of them, focus on the algorithm and follows a completely automatic process for constructing a decision tree. In these approaches, only a few parameters are configured by the analysis expert before the start of the tree construction process. As they do not involve the expert in construction process itself, precious human knowledge is wasted, instead of being inserted on final model [3],[4]. This knowledge includes tacit domain and context knowledge that resides in the expert head and was gained over the years of experience in data gathering and analysis. This knowledge is not coded formally in the data so it cannot be incorporated automatically by machine learning algorithms.

This work describes an interactive structure for the construction of decision trees, which combine mechanisms for user intervention with traditional algorithms for decision tree construction [8]. This structure allows the intervention of the expert at any time during the construction of the tree. This enables the expert to verify and, if desired, direct on the choice of classification attributes and splitting points of the tree. This interactivity allows the expert to bring his domain and context knowledge to the tree construction process. In this user-centered approach, the expert and the computer can both contribute their strengths to the tree building process: the user providing domain knowledge and evaluating intermediate results of the algorithm, the computer automatically extracting patterns, satisfying the user constraints, and generating appropriate visualizations of these patterns.

For this approach to work, one needs effective interaction mechanisms between the machine and the experts. For that, we use visual data mining techniques [16], that combine powerful visualization techniques with interactive query mechanisms. In particular, we use a hierarchical visualization technique called Treemap [6],[18],[19], combined with graphical widgets for fast tree exploration and querying. These mechanisms allows the experts to visualize and explore the intermediate trees built by the J38 classification algorithm [12]. This structure allows human interaction to be performed with the computer during the whole classification process.

2. DECISION TREE CLASSIFIERS

Decision tree classifiers [8] learn a discrete-valued classification function, which is represented by a decision tree. Figure 1 shows an example of a tree for deciding if one can play outside or not. The tree has leaf and intermediate nodes. Each intermediate node corresponds to an attribute test. Edges symbolize all possible outcomes of the test in the node. The leaf node contains the label of one of the existing classes (in the example, yes or no). A path from the root to a leaf node defines a classification rule in the decision tree. In Figure 1, the left most path contains the rule: IF outlook=sunny AND humidity<=75 THEN yes (one can play outside).

The construction of the decision tree consists of two phases: (1) the construction itself; and, (1) the pruning of the tree. In the first phase, the tree is constructed by recursively partitioning the training set until each partition consists mostly of records of the same class. This set is then labeled as a leaf node. For each intermediate node, an attribute is selected. This attribute must not have been used yet in the classification path. The chosen attribute is the one that promotes the maximum segregation among the records with respect to the classification criteria. If the selected attribute is categorical, a sub-tree is created for each of its possible values. If the attribute is numerical, the algorithm verifies the value (or values) that better splits the data with respect to class segregation. A split (normally binary) is then created, with tests of the type “less than (<)” and “equal or greater than (≥)” the chosen split value.

The construction of the tree is guided by the objective decreasing the difficulty of classification. Thus, the choice of the attribute that will constitute the decision node, as well as the selection of the split point for numerical attributes, is carried through the maximization of the discrimination between the classified class [7],[8].

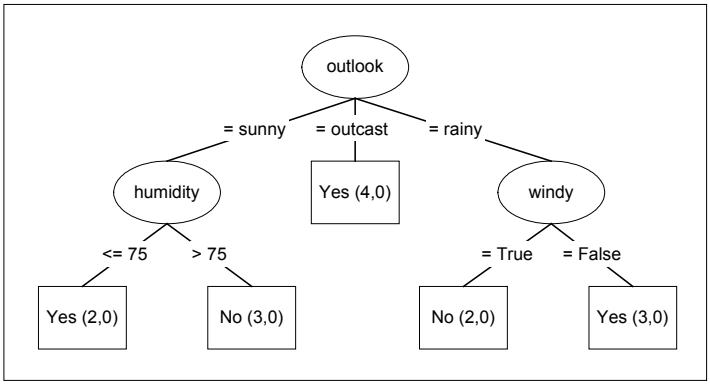

Fig. 1. A Decision Tree

Many times the resulting trees have very specialized rules. These trees adjust excessively to the peculiarities of the training set, and tend to produce deductive models of the training set instead of inductive models of the real world. On those trees, leafs nodes are supported by a small number of examples that represents isolated facts and that do not reflect reality. The pruning phase consists of simplifying the constructed tree to remove its excessively specialized parts. This generates less complex and more significant structures in terms of induction of the real world.

3. INTERACTIVELY BUILDING DECISION TREE CLASSIFIERS

Many approaches for decision tree construction have been proposed in the literature. Most of them, however, focus on completely automated algorithms for tree construction, allowing only a few parameters to be configured before the start of the algorithm. Typically, the user involvement in this process is limited to the choice of the data and the parameterization of algorithm that will be used. Once the process of construction starts, the algorithm does not allow user intervention or the visualization of intermediate results, only of the final model is shown to the user [3].
Ankerst proposed an interactive model for decision tree construction [4]. In this model the user cooperates with the computer in the construction of the tree. The user can choose the next node to be expanded, select the attribute that will partition the data as well as its split points, or to leave those decision to the system

3.1 A Model for Interactive Decision Tree Construction

Figure 2 shows the schema that we adopt for an interactive session of a decision tree construction. When the session is started, the tool associates a notepad with it. The user can use this notepad to record comments, decisions, insights, or any other information he considers worth of registering. The notepad can be called at any time during the classification session. The start of the session is also when the user prepare data for the classification.

When the session ends, the user can, in accordance with his necessity, store or discard the tree produced up to that moment together with all notes made during the session.

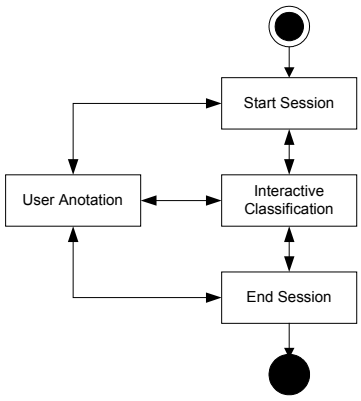

Fig. 2. A Classification Session

The central phase of a classification session is the interactive construction of the classification tree. For this phase, we adopted an adapted version of the interaction model proposed by Ankerst [4]. The adapted model is shown in Figure 3. This figure shows the activities performed by the user and by the computer. The activities performed by the user are shown in white rectangles and the activities performed by the computer are shown in gray rectangles.

The model is centered in the user. After the beginning of a work session, the user starts the interactive construction of the tree by visualizing and exploring the tree constructed up to that the moment. After exploring the tree, the user can choose one of the following operations:

- Manually remove a node of the tree. This corresponds to a manual pruning, where the current node is transformed into a leaf node and all its children are removed from the tree;
- Ask the system to suggest a list of possible classification attributes and splits points for a specific node. The attributes are ordered by its mathematical information gain for class segregation;
- Choose an attribute and its split point, and ask for the system to execute the expansion of a node based on this attribute;
- Ask for the system to explore data. In this case the user may be in doubt about which attribute choose, or the options must be researched for a more accurate choose;
- Ask for the system to automatically expand a node of the tree. In this case the attribute with bigger mathematical gain is automatically chosen by the system;
- Ask for the system to automatically expand a level of the tree. In this case all nodes in the current level are automatically expanded by the system;
- Ask to the system to prune the tree constructed up to that moment. In this case, the user has to input the parameters for pruning, and the system will automatically remove from the tree all nodes that it judges too specialized for effective classification.

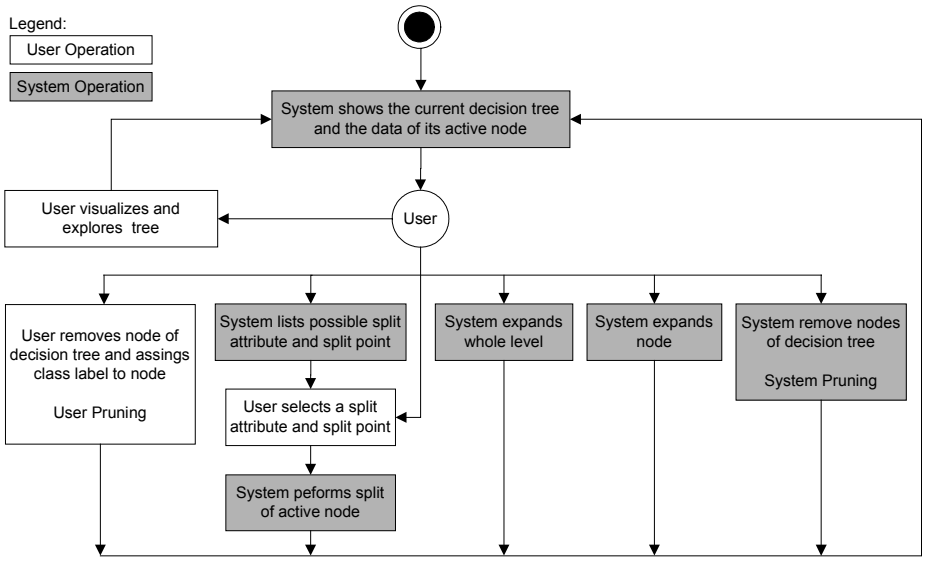

Fig. 3. Model for Interactive Decision Tree Construction

The model described in Figure 3 has some important differences from the model originally proposed by Ankerst. Our model allows the expansion of a complete level of the tree. It also allows to prune the tree at any time during its construction. Most important of all, the visualization and interactive exploration mechanisms are quite different from those proposed by Ankerst [5],[13], that is a pixel-based approach where each pair attribute-value is represented by one colored pixel in the visual screen. We propose the use treemaps [19] associated with query devices and details on demand controls that are typical from visual data mining tools [14],[16]. These resources are instrumental to the success of the model shown in Figure 3, they are intuitive and allow the user to explore the characteristics of these trees and the data associated with their nodes.

4. MINER - CLASSIFICATION TOOL

4.1 Preparation

The first step on the tool is the preparation for classification, we assume here that data preparation is done before, which may involve cleaning and data transformations. The tool supports the selection of attributes in case of data sets with large numbers of variables - performing some preliminary selection operations to bring the number of variables to a manageable range. These resources were inherited from Weka tool and help the user to identify the most relevant variables and determine the complexity and/or the general nature of models that can be taken into account in the next stage.

Data can be imported from a file in various formats: ARFF, CSV and can also be read from a URL or from an SQL database (using JDBC). The figure 4 shows the selection of some attributes to be used in the next stage.

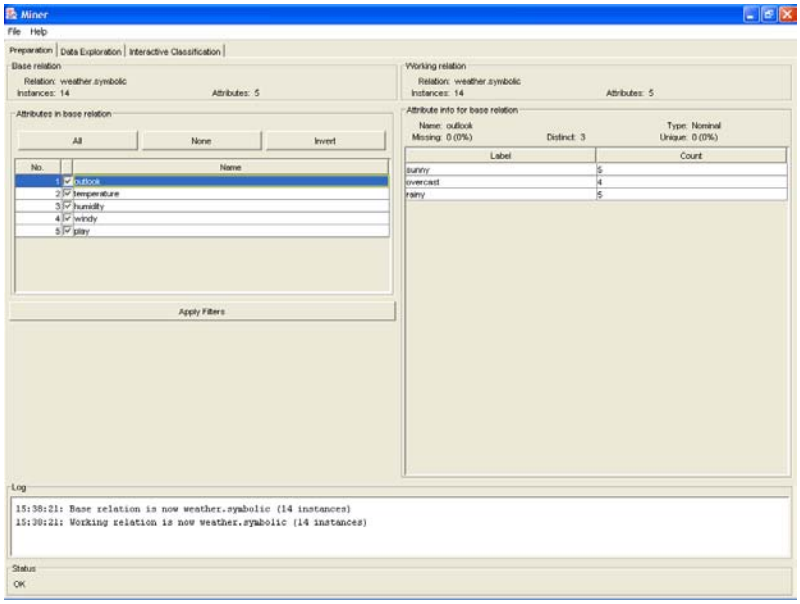

Fig 4. Preparation

4.2 Data Exploration

Treemaps is an information visualization technique proposed by Shneiderman for visualizing hierarchical structures [18],[19]. They use 100% of the available space for information visualization, mapping hierarchies in rectangular regions. The traditional representation of trees use lines to establish the connection between parents and children nodes of a hierarchy. This type of representation has two significant disadvantages: (1) a great portion of the available visual space is spent in the organization of the nodes; and, (2) non-trivial hierarchical structures generate trees of difficult visualization.

Because all the space available for drawing is used, Treemaps allow the efficient visualization of large hierarchies that can be in the order of thousands of items [18]. It is also very efficient in coding node attributes using the rectangles size and color. The visualization and exploration tools were inherited from TreeMiner, a tool that was developed at Salvador University [1],[2], it integrates treemaps with interactive queries and details on demand devices, offering an efficient mechanism for exploration of hierarchical data. The figure 5 shows the visualisation of an hierarchy of attributes .

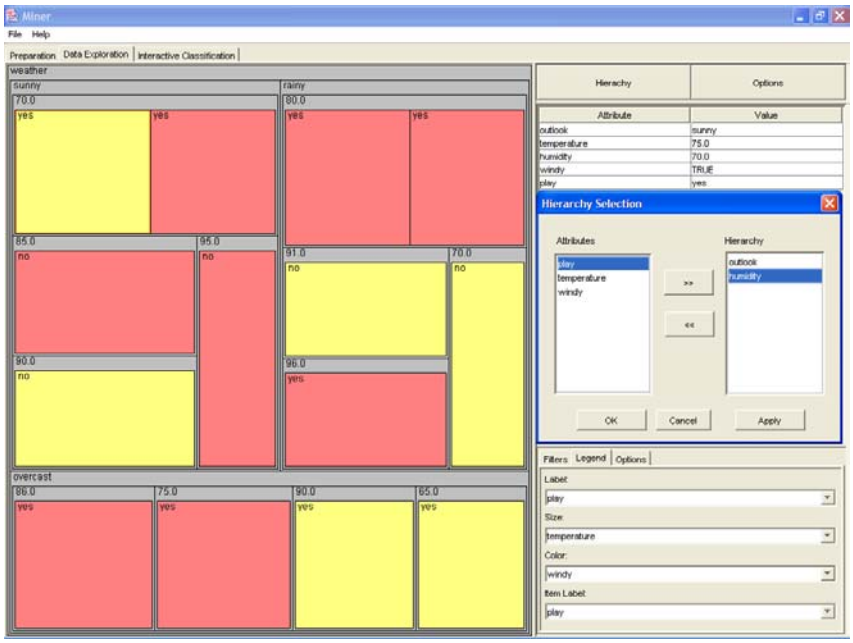

Fig 5. Data Exploration

4.3 Interactive Classification

The main part of the tool we developed is the interactive classification. The tool implements the model showed in the figure 6 (a,b,c and d) for interactive decision tree construction adapting the TreeMiner visualization structures, dynamic query and details on demand devices.

As illustrated in the Figure 3 model, the visualization of intermediate trees is brought up to date at each step of the tree construction process, allowing the expert to explore it, and re-direct the tree building process if necessary.

In the case of decision trees, the visual attribute "size" is especially indicated for representing the number of records that supports a node. The visual attribute "color" is especially indicated to represent node's purity degree – by which all records of a node belongs to the same class. Each class (e.g., yes and no) can be associated with a distinct color (e.g., red and yellow). The purity degree of a leaf node can be visualized by the cleanness of the rectangle color representing it. This creates an intuitive representation for the decision tree nodes support and purity at any point of its construction..

Figure 6 (d) shows the decision tree of Figure 1 represented as a treemap. This representation not only shows the values of the classifying attributes, but it also shows the purity and support of its leaf nodes. As hinted before, the purity of the node is represented through the colors, and the number of records that supports the nodes is represented through the size of the rectangles. Thus, the treemap offers a view of the decision tree and the class distribution of the mined base.

Beyond the visualization, the developed tool supplies interaction mechanisms based in visual data mining techniques to assist the user to explore the produced tree. Upon selecting a node in the tree, the information about that node is shown in right superior corner of the screen, see Figure 6. This shows the following to the user: the dominant class of the node, the amount of registers that supports the node, and the total number of impurities in the node (i.e., the number of records that do not belong to the node dominant class). Moreover, if necessary, the user can also request the list of all the registers that support the selected node.

The tool also has, see bottom right corner of Figure 6, filters that are typical of visual data mining. The "depth filter" hides deeper nodes of the tree, allowing to the user to quickly examine simplified trees, and take pruning decisions. The class filter redraw the tree using only the selected classes. The "impurity filter" allow the interactive verification of which nodes have impurity values over and under the set values. And, the "number of records filter" allow the verification of which nodes have support over and under the set values. It is important to observe that the time between user interaction with these filters and the tool updating of the visual screen is practically zero. With these interaction resources the user can quickly decide which node to remove or to expand, or if the obtained result is already satisfactory.

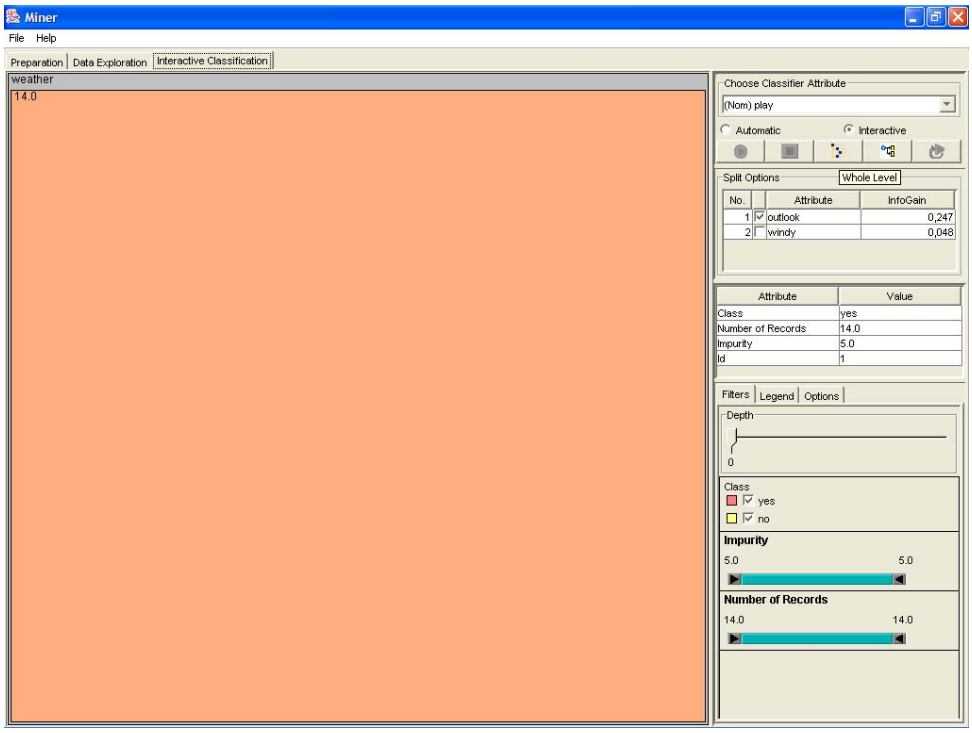

Fig 6 (a). Beginning of the Interactive Classification

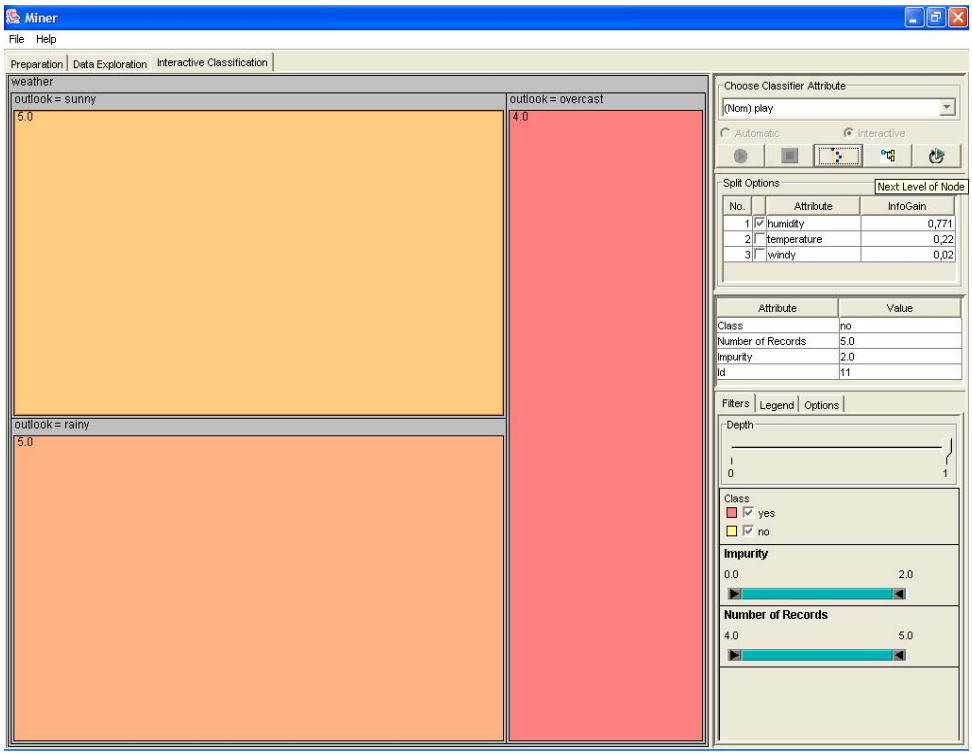

Fig 6 (b). Interactive Classification – Expand a whole level

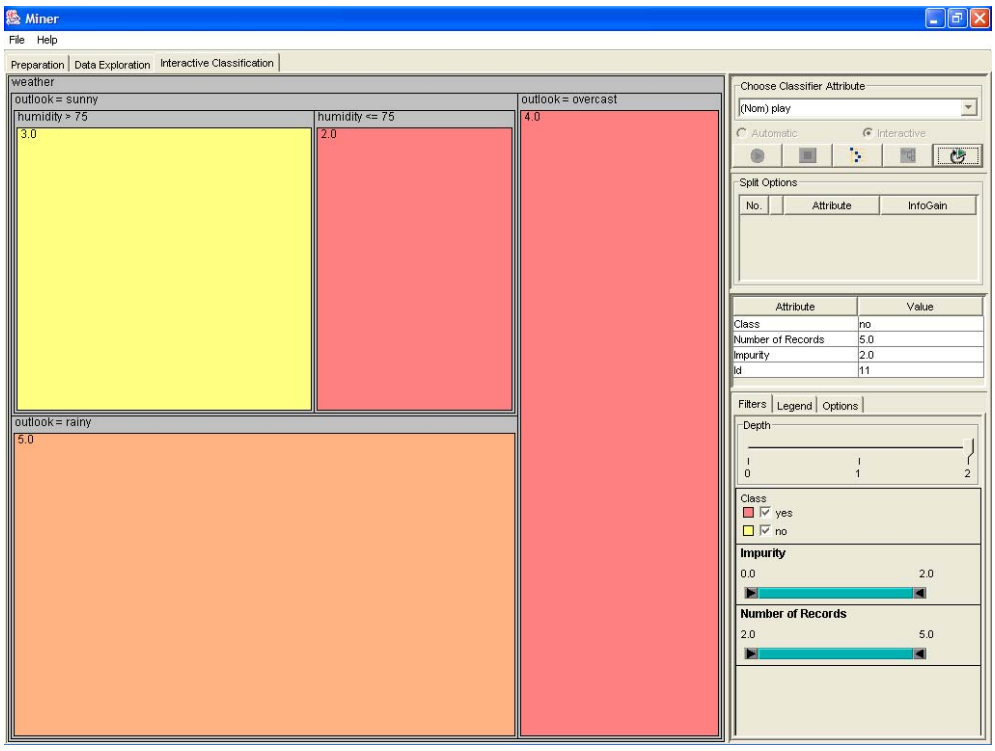

Fig 6 (c). Interactive Classification – Expand a selected node of the tree

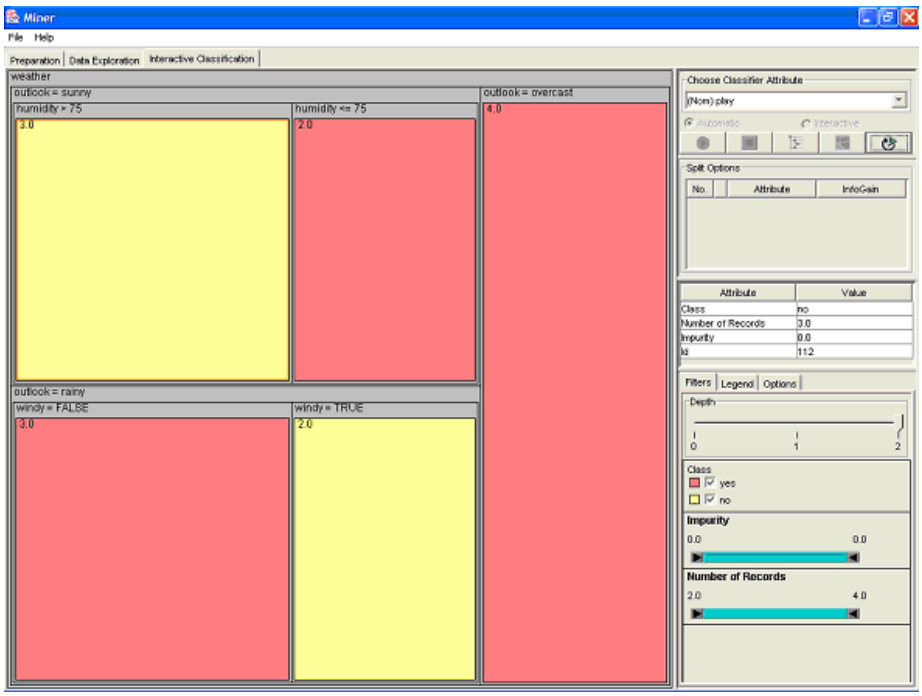

Fig 6 (d). The decision tree

5. SOME IMPLEMENTATION DETAILS

The tool that implements the interaction model described in Figures 2 and 3 was developed from two existing tools: (1) Weka, an environment for knowledge discovery, that makes possible, among others things, the construction of classification trees; and (2) TreeMiner, a treemap based hierarchical visual data mining tool. The two tools are implemented in Java and are public domain, this facilitated their integration, adaptation, and expansion.

The Weka Tool (Waikato Environment will be Knowledge Analysis) was developed at the University of Waikato in New Zealand [10],[12]. It implements algorithms for several data mining techniques, including the J48, an improved version C4.5 for building decision trees [17]. The Weka is available in following web address: http://www.cs.waikato.ac.nz/ml/weka/.

TreeMiner was developed at Salvador University [1],[2], it integrates treemaps with interactive queries and details on demand devices, offering an efficient mechanism for exploration of hierarchical data. The tool we developed for interactive decision tree construction adapts the TreeMiner visualization structures, dynamic query and details on demand devices, as described in Section 4 and portrayed in Figure 4.

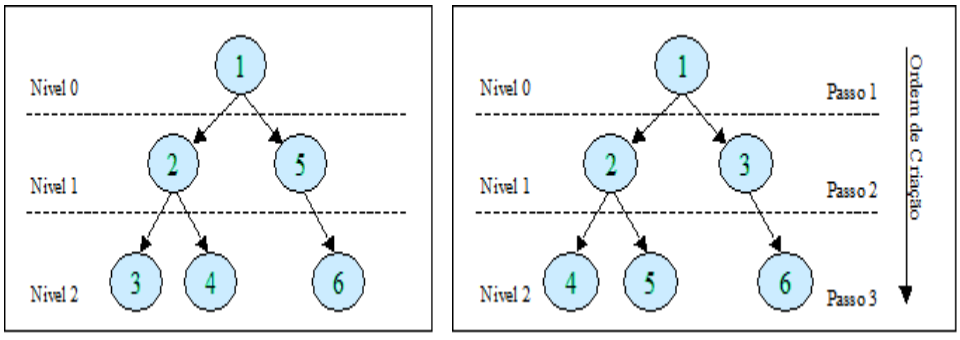

Fig. 7. Old and New Sequence of Tree Nodes Construction

From Weka tool, the tool we developed uses the structure for creating classification sessions, its resources for data preparation, its resource for algorithm parameterization and the base of the J48 algorithm for constructing classification trees. The algorithm, however, had to be adapted in order to make it possible the interaction with the user during the construction of the tree. It was adapted to allow: (1) the construction of the tree level by level, (2) the

construction of the tree node by node, and (3) listing the attributes and its splitting points for a given node. For the first functionality, the algorithm was modified to function level by level (breadth first) instead of recursively (depth first). This way, the user can interact better with the algorithm during the tree construction. Figure 7 illustrates the old and the new sequence of node construction carried by the algorithm.

Besides the cited adaptations in the Weka and the TreeMiner tools, the following functionalities were or are being added to our decision tree construction tool: (1) a functionality for user annotation is available during the whole tree construction process; (2) filters are being built to assist the selection of data for the training sets; (3) a functionality for mapping numerical attributes into categorical ones is being integrated into the tool; and , (4) the parameterization of the J48 algorithm is being modified to accept the assignment of weights for the attributes.

6. CONCLUSION

The current classification approaches allow a limited participation of the expert during the decision tree construction process. These approaches do not take advantage of the experts´ knowledge with respect to the data and domain being mined. With a cooperative classification approach, the expert and the computer can contribute with their best. The computer providing the capacity to recognize mathematical patterns. The experts providing their ability to interpret and promote a deeper understanding of these patterns. This combination should generate more trustworthy models.

This work defined a model and implemented a tool for the interactive construction of classification trees. The defined model is adapted from the cooperative classification model originally proposed by Ankerst. The implemented tool enacts this model, using treemap visualizations and visual data mining query devices to create efficient mechanisms for supporting user-computer interactions during the construction of decision trees.

The tool was developed by adapting a visual data mining tool, TreeMiner, and an environment for knowledge discovery, Weka. The resulting tool supports all phases of classification process, from the data selection to the interactive exploration of the obtained results.

As future work, we intend to include new mechanisms for visualizing and interacting with the decision trees. In particular, we intend to add a module that uses traditional tree drawing methods to visualize and interact with produced trees. This will allow the comparison of the treemap approach with more traditional approaches for tree visualizations and exploration. We also intend to incorporate a module for visually comparing generated trees by overlapping them on the computer screen at each iteration of the tree construction process.

References

[1] Almeida, M.: A Tool for Visual Data Mining Using Treemaps and its Applications (in Portuguese). Master of Computer Science Thesis, Salvador University (UNIFACS), Salvador, Ba, Brazil, (2003).

[2] Almeida, M.; Mendonça Neto, M. Using Treemaps to Internalize Knowledge (in Portuguese). In the Proceedings of the First Brazilian Workshop on Information Systems and Knowledge Management, SBC, Fortaleza, CE, Brazil, 2003. v. 1, p. 1-11.

[3] Ankerst, M., Elsen, C., Ester, M., Kriegel, H.-P.: Visual Classification: An Interactive Approach to Decision Tree Construction. In ACM SIGKDD 5th Int. Conf. On Knowledge Discovery and Data Mining, San Diego, CA, USA, pp. 392-396 (1999).

[4] Ankerst, M., Ester, M., Kriegel, H.-P.: Towards an Effective Cooperation of the User and the Computer for Classification. In ACM SIGKDD 6th Int. Conf. On Knowledge Discovery and Data Mining (KDD 2000), Boston, MA, pp. 179-188 (2000).

[5] Ankerst, M.: Visual Data Mining with Pixel-Oriented Visualization Techniques. Proc. Workshop Visual Data Mining (2001).

[6] Babaria, K.: Introduction to Treemap. University of Maryland. [on line] http://www.cs.umd.edu/hcil/treemap3/TreemapIntroduction.pdf (2001).

[7] Barlow, T., Neville, P.: Case Study: Visualization for Decision Tree Analysis in Data Mining. Proc. of IEEE Symposium on Information Visualization - INFOVIS´01 (2001).

[8] Breiman, L., Friedman, J., Olshen, R., and Stone, C.: Classification and Regression Trees. Belmont, CA: Wadsworth (1984).

[9] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery in Databases. In AI Maganize, pp. 37-54 (1996).

[10] Frank, E. and e. al: Weka [http://www.cs.waikato.ac.nz/ml/weka/], The University of Waikato.

[11] Garner, S.R.: ARFF-the WEKA dataset format. World Wide Web hypertext document at http://www.cs.waikato.ac.nz/~ml/weka/arff.html.

[12] Garner, S.R.: WEKA: The Waikato Environment for Knowledge Analysis. In Proc. of the New Zealand Computer Science Research Students Conference, pages 57—64 (1995).

[13] Keim, D. A.: Designing Pixel-Oriented Visualization Techniques: Theory and Application. In IEEE Trans. on Visualization and Computer Graphics, vol 6: IEEE Computer Society, pp. 59-78 (2000).

[14] Keim, D. A.: Information Visualization and Visual Data Mining. In IEEE Trans. on Visualization and Computer Graphics, vol 7, nº 1, pp. 100-107 (2002).

[15] Mendonça Neto, M.; Sunderhaft, N. A State of the Art Report: Mining Software Engineering Data. State of the Art Technical Report DACS-SOAR-99-3. U.S. Department of Defense (DoD) Data & Analysis Center for Software, Rome, NY, 1999. Also available in: http://www.dacs.dtic.mil/techs/datamining/datamining.pdf

[16] Oliveira, M., Levkowits, H.: From Visualization to Visual Data Mining: A Survey. In IEEE Transactions on Visualization and Computer Graphics, vol 9, nº 3, pp. 378-394 (2003).

[17] Quilan, J. R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA (1993).

[18] Shneiderman, B., Johnson, B.: Treemaps: A Space-Filling Approach to the Visualization of Hierachical Information Structures. Proc. of IEEE Information Visualization, pp. 275-282 (1991).

[19] Shneiderman, B. Tree visualization with tree-maps: 2-d space-filling approach. ACM Transactions on Graphics, vol. 11 , nº. 1, pp. 92-99 (1992).