TOWARDS SIMPLE, EASY TO UNDERSTAND, AN INTERACTIVE DECISION TREE ALGORITHM

Thanh-Nghi Do
College of Information Technology, Cantho University
1 Ly Tu Trong Street, Ninh Kieu District
Cantho City, Vietnam
dtnghi@cit.ctu.edu.vn

Abstract: *Data mining is intended to extract hidden useful knowledge from large datasets in a given application. This usefulness relates to the user goal, in other words only the user can determine whether the resulting knowledge answers his goal. Therefore, data mining tool should be highly interactive and participatory. This paper presents an interactive decision tree algorithm using visualization methods to gain insight into a model construction task. We show how the user can interactively use cooperative tools to support the construction of decision tree models. The idea here is to increase the human participation through interactive visualization techniques in a data mining environment. The effective cooperation can bring out some progress towards reaching advantages like, the user can be an expert of the data domain and can use this domain knowledge during the whole model construction, the confidence and comprehensibility of the obtained model are improved because the user was involved in its construction, we can use the human pattern recognition capabilities. The experimental results on Statlog and UCI datasets show that our cooperative tool is comparable to the automatic algorithm C4.5, but the user has a better understanding of the obtained model.*

Keywords: Visual Data Mining, Machine Learning, Classification, Information Visualization, Human Factors.

1. INTRODUCTION

In recent years, real-world databases increase rapidly (double every 9 months [5]). So the need to extract knowledge from very large databases is increasing. Knowledge Discovery in Databases (KDD [4]) can be defined as the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. Data mining is the particular pattern recognition task in the KDD process. It uses different algorithms for classification, regression, clustering and association. Classification is one of the major tasks of data mining. Over the years, many classification algorithms have been proposed such as decision tree classifiers which have become very popular [8]. They have shown practical relevance for classification, regression. Successful applications of decision tree algorithms have been reported for various fields, for example in text categorization, marketing and bioinformatics, etc [9]. Decision tree algorithms efficiently classify the data with a good accuracy. However, state-of-the-art algorithms do not incorporate the user in the tree construction process. Thus, the user would like sometimes to explain or even understand why a model constructed by a decision tree algorithm performs a better prediction than many other algorithms. Data mining is intended to extract hidden useful knowledge from large datasets in a given application. This usefulness relates to the user goal, in other words only the user can determine whether the resulting knowledge answers his goal. Therefore, understanding the result produced by a data mining algorithm is as important as the accuracy. There has been two cooperative decision tree algorithms based on visualization techniques, PBC [1], CIAD [12]. The idea here is to increase the human participation through interactive visualization techniques in a data mining environment. Our investigation also aims at using visualization methods to try to involve more intensively the user in the construction of the decision tree model and to try to explain the results. The effective cooperation can bring out some progress towards reaching advantages:

- the user can be an expert of the data domain and can use this domain knowledge during the whole model construction,

- the confidence and comprehensibility of the obtained model are improved because the user was involved in its construction,

- we can use the human pattern recognition capabilities.

A new interactive decision tree algorithm using visualization methods is to gain insight into a model construction task and interpreting results. We will illustrate how to combine some strength of different visualization methods to help the user in the construction of decision tree models and improve the comprehensibility of results too. The numerical test results on Statlog and UCI datasets [2], [11] show that our interactive decision tree approach is competitive to the automatic algorithm C4.5, but the user has a better understanding of the obtained model.

We briefly summarize the content of the paper now. In section 2, we introduce the automatic decision tree algorithm for classification problems. In section 3, we show how the user can interactively use cooperative tools to support the construction of decision tree models. We present numerical test results in section 4 before the conclusion in section 5.

2. C4.5: AUTOMATIC DECISION TREE ALGORITHM

Decision trees are powerful and popular tools for classification and prediction. The attractiveness of decision trees is due to the fact that, in contrast to neural networks, decision trees represent rules that facilitate human interpretation. Decision tree is a classifier in the form of a tree structure (c.f. figure 1), where each node is either: a leaf node holds the class prediction, a decision node specifies some test to be carried out on a single attribute-value with one branch and sub-tree for each possible outcome of the test. One inductive rule (IF-THEN) is created for each path from the root to a leaf, each dimension value along a path forms a conjunction and the leaf node holds the class prediction.

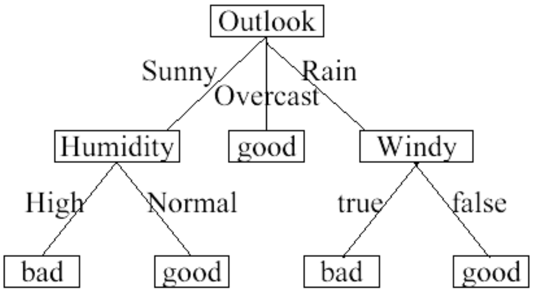


Figure 1. Decision tree for the Weather dataset

Most algorithms that have been developed for learning decision trees are variations on a core algorithm that employs a top-down, greedy search through the space of possible decision trees. Tree is constructed in a top-down recursive divide-and-conquer. At start, all the training data are at the root. Attributes are categorical (if continuous-valued, they are discretized). Data are partitioned recursively based on selected attributes. Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain). An induced tree may overfit the training data, therefore pruning sub-trees or branches, in a bottom-up manner, is necessary to improve the estimated accuracy on new data. C4.5 proposed by [13] is one of the popular decision tree algorithms for classification.

Let us consider a classification task with m datapoints in the n-dimensional input (n attributes) space R^n, denoted by the x_i (i=1, …, m), having corresponding labels c_i. The number of classes is denoted by k, the number of datapoints of class c_k is denoted by m_k. Test attributes in C4.5 are selected on the basis of statistical measure, the highest information gain (purity measure) is defined as:

The expected information $I(x_1, x_2, \ldots, x_m) = -\sum_{i=1}^{k} \frac{m_i}{m} \log_2 \frac{m_i}{m}$

The entropy of an attribute A, $E(A) = \sum_{j=1}^{\#values(A)} \frac{m_{1j} + m_{2j} + \ldots + m_{kj}}{m} I(m_{1j}, m_{2j}, \ldots, m_{kj})$

where m_{kj} is the number of datapoints having class c_k and attribute value v_j

The information gain of an attribute A, $Gain(A) = I(x_1, x_2, \ldots, x_m) - E(A)$

The C4.5 has shown practical relevance for classification. Successful applications of C4.5 have been reported for various fields, for example in text categorization, marketing and bioinformatics, etc. The C4.5 algorithm efficiently classifies the data without requiring much computation. It is able to generate understandable rules (IF-THEN) and also handle both continuous and categorical variables. However, state-of-the-art algorithms do not incorporate the user in the tree construction process. Thus, the user would like sometimes to explain or even understand why a model constructed by a decision tree algorithm performs a better prediction than many other algorithms.

3. INTERACTIVE DECISION TREE

Data-mining is intended to extract hidden useful knowledge from large datasets in a given application. This usefulness relates to the user goal, in other words only the user can determine whether the resulting knowledge answers his goal. Therefore, data mining tool should be highly interactive and participatory. The idea here is to increase the human participation through interactive visualization techniques in a data mining environment. The effective cooperation can bring out some progress towards reaching advantages [6], [10] such as:

- the user can be an expert of the data domain and can use this domain knowledge during the whole model construction,

- the confidence and comprehensibility of the obtained model are improved because the user was involved at least partially in its construction,

- we can use the human pattern recognition capabilities.

3.1. Data visualization

Over the last decade, a large number of visualization methods developed in different domains have been used in data exploration and knowledge discovery process. The visualization methods are used for data selection (pre-processing step) and viewing mining results (post-processing step). Some recent visual data mining methods try to involve more intensively the user in the data-mining step through visualization. We only present three visualization techniques: the 2D scatter-plot matrices [3], the parallel coordinates [7] and bar visualization [1] significantly used for data exploration. We believe that these methods are valuable.

3.1.1. 2D scatter-plot matrices

The data points are displayed in all possible pair wise combinations of dimensions in 2D scatter plot matrices. For n-dimensional data, this method visualizes n(n-1)/2 matrices. Figure 2 depicts the Segment dataset from the UCI repository which contains 2310 datapoints in 19-dimensional input space with 7 classes (corresponding to 7 colors).

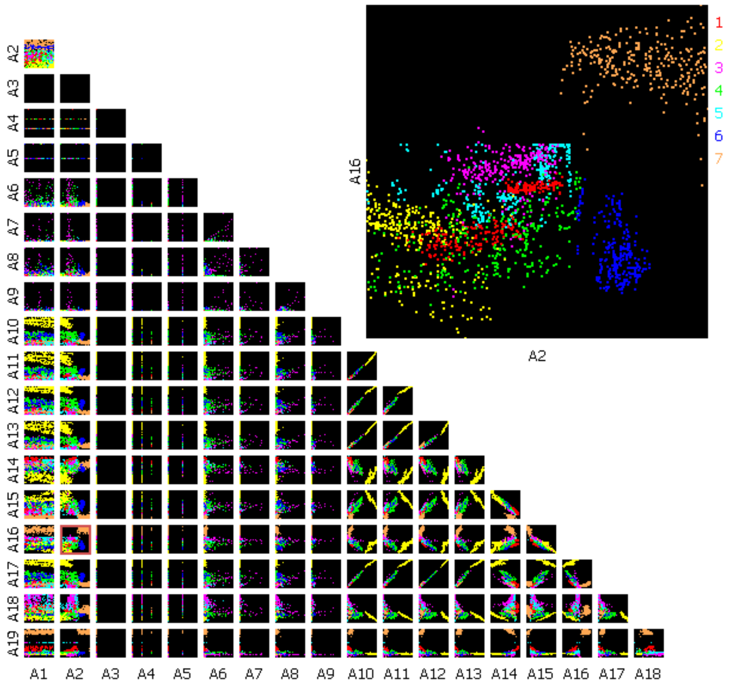


Figure 2. Visualization of the Segment dataset with 2D scatter-plot matrices

3.1.2. Parallel coordinates

The parallel axes represent the data dimensions. A data point corresponds to a poly-line intersecting the vertical axes at the position corresponding to the data value. Figure 3 depicts the Segment dataset with parallel coordinates.

3.1.3. Bar visualization

This method divides the display into n equal sized bars (regions) for n-dimensional space with each bar corresponding to a dimension. Within a bar, the sorted attribute values are mapped to pixels in a line-by-line according to their order. Figure 4 presents an example of bar visualization of the Segment dataset.

No single visualization tool is the best for high dimensional data exploration: some visualization methods are the best for showing partitions of data, some other visualization methods can handle very large dataset. In all cases, we would like to combine different visualization techniques to overcome the single one. The same information is displayed in different views with different visualization techniques providing useful information to the user.

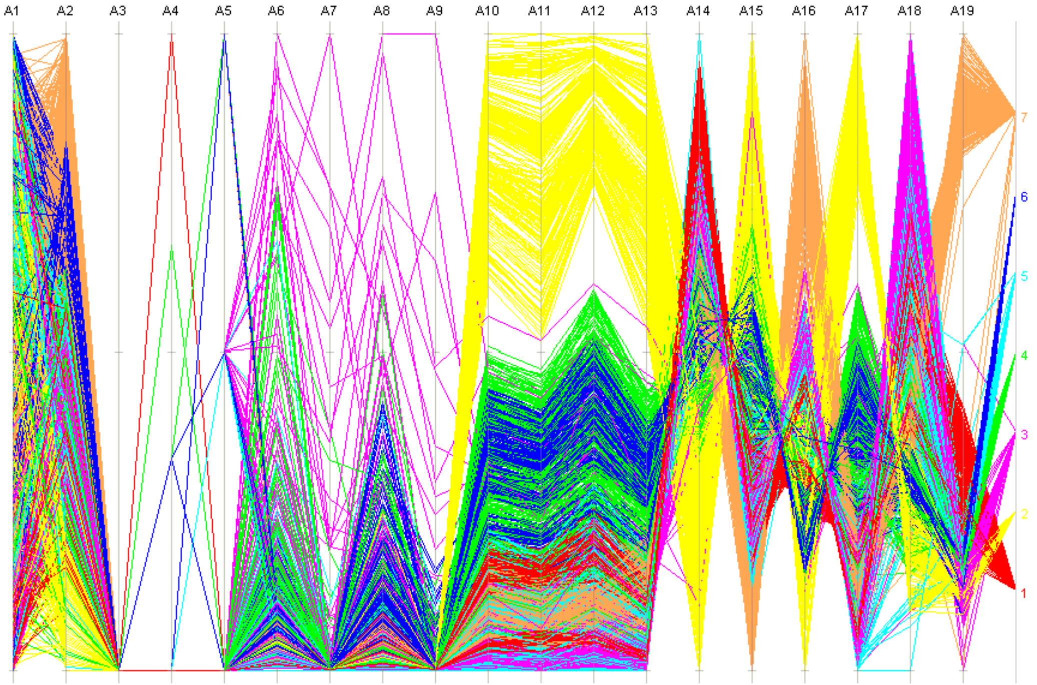


Figure 3. Visualization of the Segment dataset with parallel coordinates

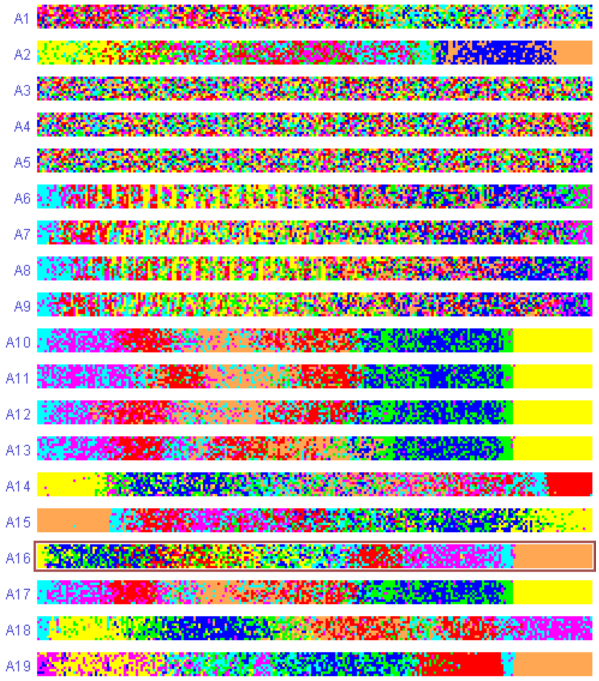


Figure 4. Visualization of the Segment dataset with bar visualization

Furthermore, interactive linking and brushing can be also applied to multiple views: the user can select points in one view and these points are automatically selected (highlighted) in the other available views. Thus, the linked multiple views provide more information than the single one.

3.2. Interactive construction of decision trees

The cooperative method tries to involve the user in the construction of decision tree model with multiple linked views and brushing. The starting point of the cooperation here is the multiple views used to visualize the same dataset. The user can choose appropriate visualization methods to gain insight of data. The interactive graphical methods provide some utilities for example brushing, zoom, linking, etc. that can help the user to select by him-self test attributes and split points or oblique cuts according to best pure partitions. The top level with full dataset corresponds to the root of the decision tree. Without requiring a heuristic or statistical measure (e.g., information gain) in automatic decision tree algorithms, the human eye is an excellent tool for spotting natural patterns. The user can chooses by him-self test attributes and an arbitrary number of split points (with bar visualization or parallel coordinates) or an oblique cut in 2 dimensions (with 2D scatter-plot matrices). After that, the pure partition can be assigned to a leaf node holding the class prediction of its only color. The visualization of the remaining partition has to be examined in a further step. On lower levels, partitions of the datapoints inherited from upper levels are visualized on the multiple views. And then, datapoints are partitioned recursively based on the human pattern recognition capabilities. The user can be an expert of the data domain and can use this domain knowledge during the model construction.

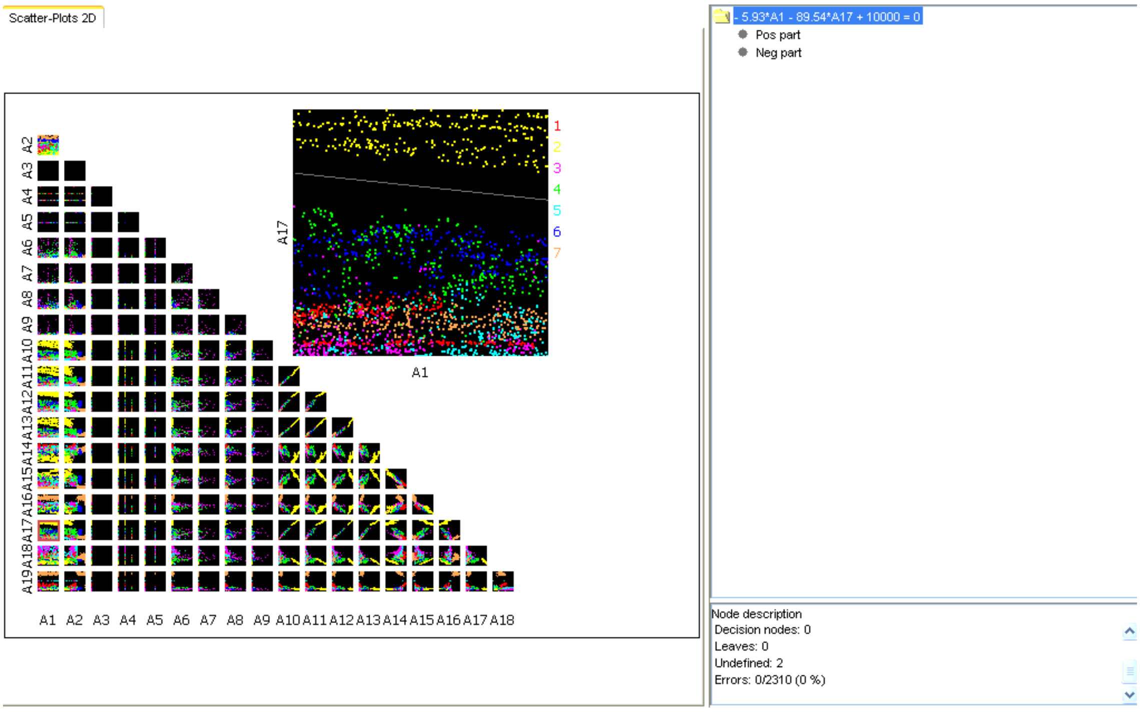
Furthermore, the user is also possible to do backtracking in the tree construction phase. No changes are required from the habitual case other than the direct modification of the tree node. The user can delete this node and then choose test attributes and split points (cuts) in another way. A tree view represents the obtained result in the graphical mode more intuitive than from the columns of numbers or the rules set at the output of the automatic algorithms. The user can easily extract the inductive rules and to prune the tree in the post-processing stage. The user has a better understanding of the obtained model because he was involved in the tree construction phase.

Let us consider the decision tree construction on the Segment dataset based on the 2D scatter-plot matrices. With the human pattern recognition capabilities, the user can find out a project in 2 dimensions that has best pure partitions. The projection of attributes A1 and A17 is chosen to isolate class 2. Figure 5 depicts the oblique cut in 2 dimensions (attributes A1 and A17) for separating class 2 from the other ones. The user can assign the pure partition (yellow) to a leaf node holding the class 2 (c.f. figure 6). The visualization of the remaining partition has to be examined in a next step (c.f. figure 7).

According to best pure partitions, the user continues to find out a project in 2 dimensions to separate a class from the other ones. Figure 8 depicts the next oblique cut in 2 dimensions (attributes A1 and A19) for separating the class 7 from the other ones. The user can assign the pure partition to a leaf node (class 7).

The visualization of the remaining partition has to be examined in a next step. The same way, the user continues to construct the model. The interaction is finished when all classes have been assigned to leaves. We have obtained the decision tree depicted in figure 9.

A tree view in the graphical mode of the obtained result helps the user to easily interact to the decision tree. By clicking on a node, he can see its split cut and information like the accuracy, the number of leaves. The user is also possible to do backtracking if he wants anyway.



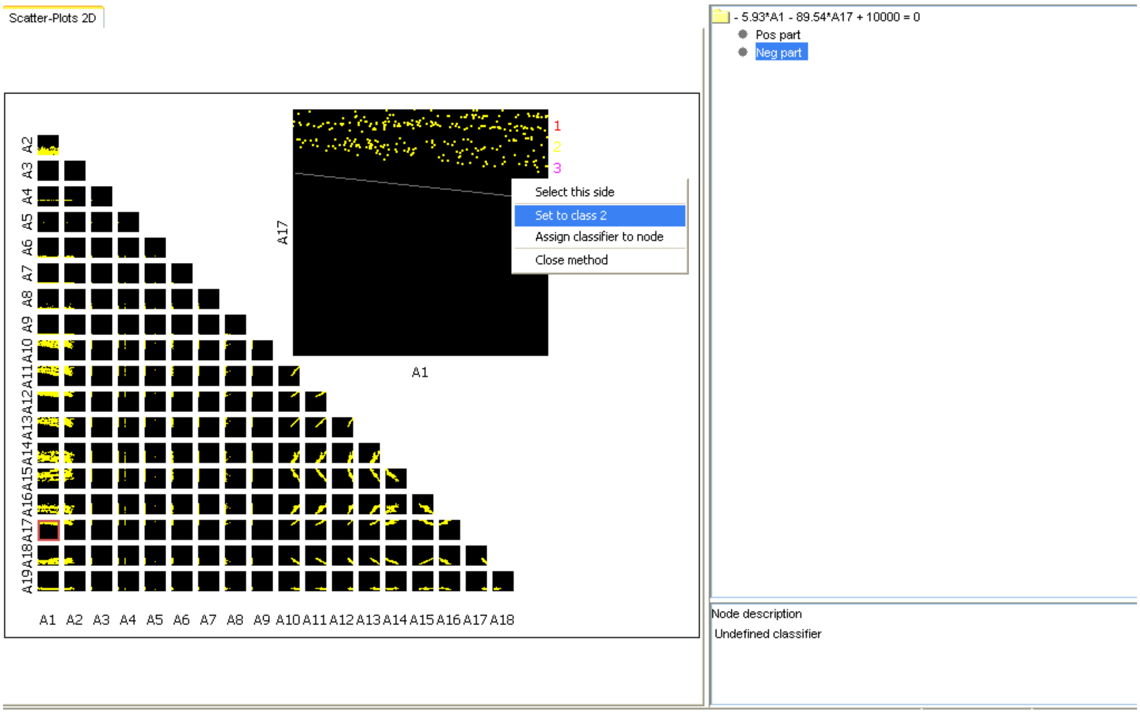
Scatter-Plots 2D

- 5.93*A1 - 89.54*A17 + 10000 = 0
- ● Pos part
- ● Neg part

Node description
Decision nodes: 0
Leaves: 0
Undefined: 2
Errors: 0/2310 (0 %)

A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
A18
A19
A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18
A17
A1
1
2
3
4
5
6
7

Figure 5. Oblique cut in attributes A1 and A17 for separating class 2 from the other ones

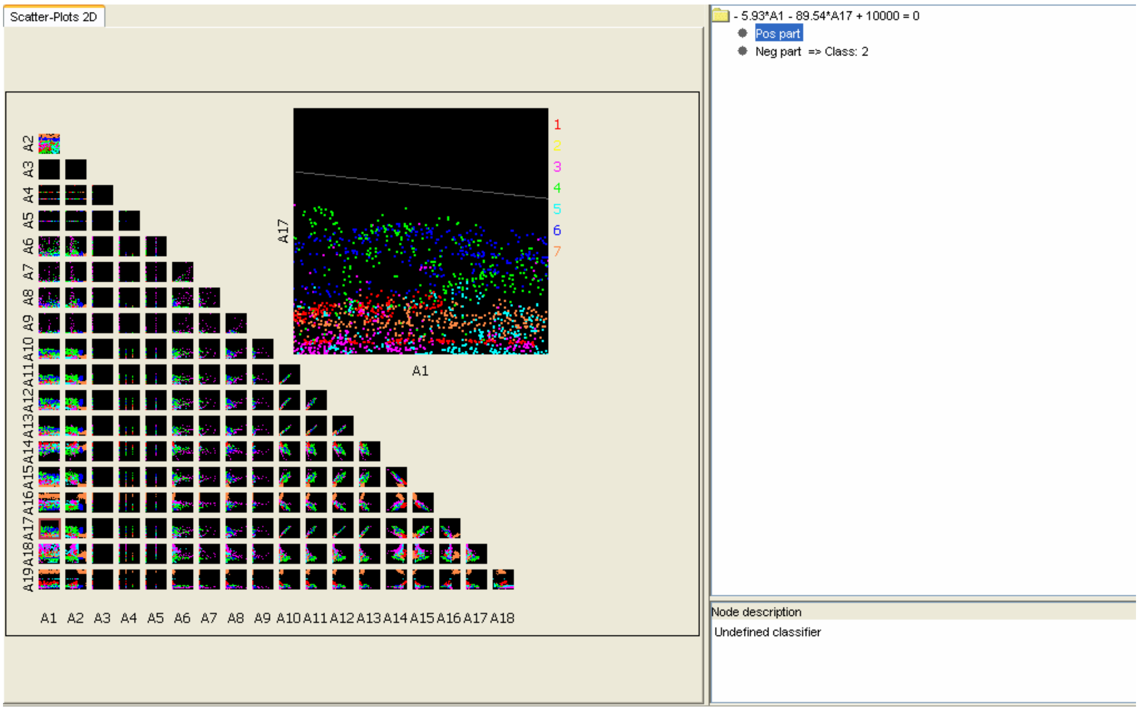


Scatter-Plots 2D

- 5.93*A1 - 89.54*A17 + 10000 = 0
- ● Pos part
- ● Neg part

Node description
Undefined classifier

Figure 6. Assigning the pure partition to the class 2



Scatter-Plots 2D

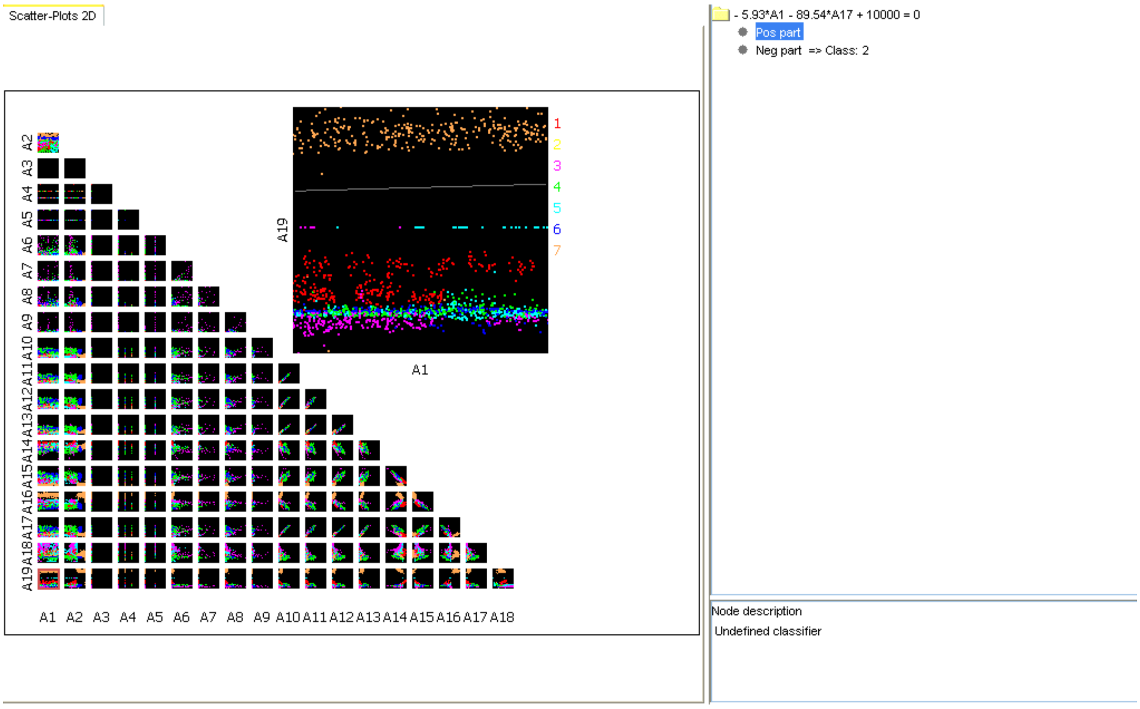
- 5.93*A1 - 89.54*A17 + 10000 = 0
- Pos part
- Neg part => Class: 2

Node description

Undefined classifier

A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
A18
A19
A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18
A17
A1
1
2
3
4
5
6
7

Figure 7. Visualization of the remaining partition



Scatter-Plots 2D

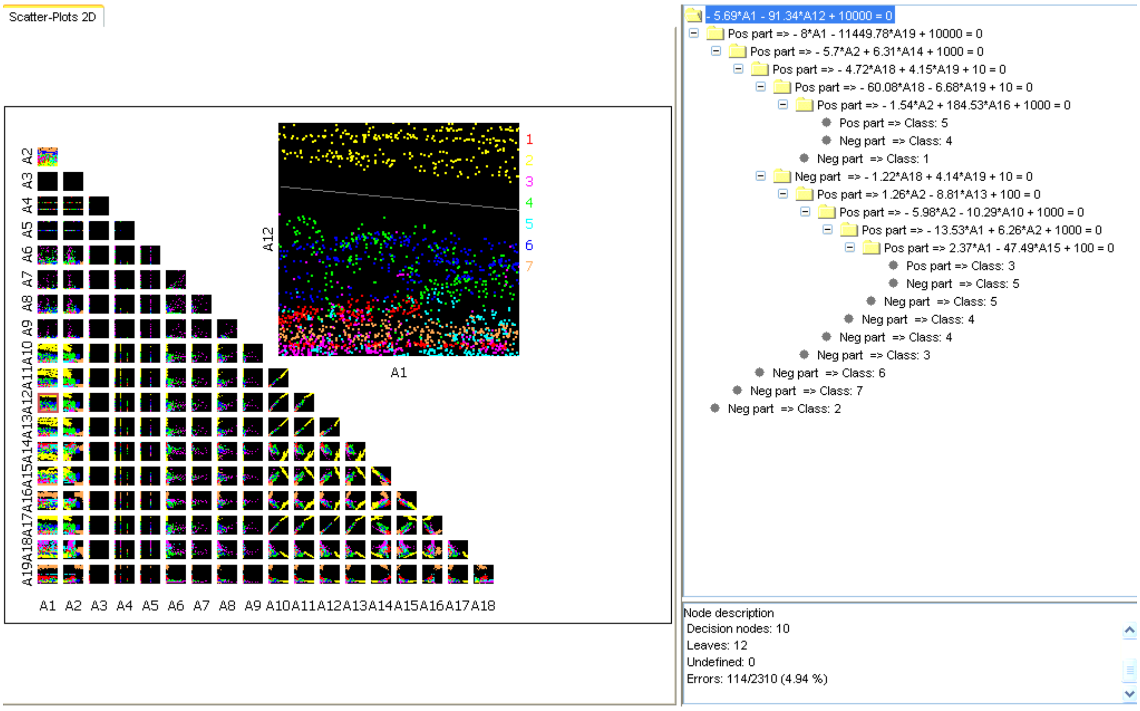
- 5.93*A1 - 89.54*A17 + 10000 = 0
- Pos part
- Neg part => Class: 2

Node description

Undefined classifier

A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
A18
A19
A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18
A19
A1
1
2
3
4
5
6
7

Figure 8. Oblique cut in attributes A1 and A19 for separating class 7 from the other ones


Scatter-Plots 2D
A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19
A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18
A12
A1
1
2
3
4
5
6
7

```
- 5.69*A1 - 91.34*A12 + 10000 = 0
  Pos part => - 8*A1 - 11449.78*A19 + 10000 = 0
    Pos part => - 5.7*A2 + 6.31*A14 + 1000 = 0
      Pos part => - 4.72*A18 + 4.15*A19 + 10 = 0
        Pos part => - 60.08*A18 - 6.68*A19 + 10 = 0
          Pos part => - 1.54*A2 + 184.53*A16 + 1000 = 0
            Pos part => Class: 5
            Neg part  => Class: 4
          Neg part  => Class: 1
        Neg part  => - 1.22*A18 + 4.14*A19 + 10 = 0
          Pos part => 1.26*A2 - 8.81*A13 + 100 = 0
            Pos part => - 5.98*A2 - 10.29*A10 + 1000 = 0
              Pos part => - 13.53*A1 + 6.26*A2 + 1000 = 0
                Pos part => 2.37*A1 - 47.49*A15 + 100 = 0
                  Pos part => Class: 3
                  Neg part  => Class: 5
                Neg part  => Class: 5
              Neg part  => Class: 4
            Neg part  => Class: 4
          Neg part  => Class: 3
      Neg part  => Class: 6
    Neg part  => Class: 7
  Neg part  => Class: 2
```

```
Node description
Decision nodes: 10
Leaves: 12
Undefined: 0
Errors: 114/2310 (4.94 %)
```

Figure 9. Decision tree for the Segment dataset

4. NUMERICAL TEST RESULTS

The software program is written in Java on PC (Linux). It consists of three visualization methods like 2D scatter-plot matrices, parallel coordinates and bar visualization for involving the user in the construction of decision tree model. To validate the performances of the interactive decision tree tool (IDT), we have used it to classify Statlog and UCI datasets compared with the automatic algorithm C4.5. Thus, we have obtained the results concerning accuracy and tree size shown in table 1. The best result is in bold face for each dataset.

	Classes	Points	Attributes	Eval. protocol	IDT acc, size	C4.5 acc, size
Segment	7	2310	19	10-fold	95.10% **22**	**96.60%** 77
Shuttle	7	58000	9	train–test	99.82% **17**	**99.90%** 57
Pima	2	768	8	10-fold	**77.20%** **7**	76.53% 20

Table 1. Results on Statlog and UCI datasets

As we can see in table 1, our interactive decision tree tool has a good classification accuracy compared to the automatic algorithm C4.5, the accuracy is not significantly different.

However, the tree size of IDT is very compact so that the extracted rules (created for each path from the root to a leaf) facilitate human interpretation. The major advantage of the cooperative approach can help the user gains insight into data and model. This can help him to avoid the risk of wrong decisions because he gets more comprehensibility and confidence in the model constructed by him-self.

5. CONCLUSION AND FUTURE WORK

We have presented an interactive decision tree algorithm using visualization methods to gain insight into a model construction task. The idea here is to increase the human participation through interactive visualization techniques in a data mining environment. The user can choose appropriate visualization methods to gain insight of data. The interactive graphical methods provide some utilities like brushing, zoom, linking, etc. that can help the user to select by him-self test attributes and split points or oblique cuts according to best pure partitions in decision tree construction phase. Without requiring a heuristic or statistical measure (e.g., information gain) in automatic decision tree algorithms, the human eye is an excellent tool for spotting natural patterns. The effective cooperation can bring out some progress towards reaching advantages like, the user can be an expert of the data domain and can use this domain knowledge during the whole model construction, the confidence and comprehensibility of the obtained model are improved because the user was involved in its construction, we can use the human pattern recognition capabilities. The experimental results on Statlog and UCI datasets show that our cooperative tool is comparable to the automatic algorithm C4.5, but the user has a better understanding of the obtained model.

A forthcoming improvement will be to combine our method with other machine learning algorithms to construct another approach that can deal with a complex classification task.

REFERENCES

[1] M. Ankerst, M. Ester, and H-P. Kriegel.: Towards an Effective Cooperation of the Computer and the User for Classification. Proceeding of KDD'00, *6th ACM SIGKDD*, Boston, USA, 2000, pp.179-188.

[2] C. Blake and C. Merz.: UCI Machine Learning Repository. 1998.

[3] W. Cleveland.: *Visualizing Data*. AT&T Bell Laboratories, Hobart Press, 1993.

[4] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth.: From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3), pp.37-54, 1996.

[5] U. Fayyad, G. Piatetsky-Shapiro, and R. Uthurusamy.: Summary from the KDD-03 Panel – Data Mining: The Next 10 Years. in *SIGKDD Explorations*, 5(2), pp.191-196, 2004.

[6] U. Fayyad, G. Grinstein, and A. Wierse.: *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann Publishers, 2001.

[7] A. Inselberg.: The Plan with Parallel Coordinates. Special Issue on *Computational Geometry of The Visual Computer*, 1(2) , pp.69-97, 1985.

[8] Kdnuggets.: What data mining techniques you use regularly?. KDnuggets Polls, Nov, 2003.

[9] Kdnuggets.: Which data mining techniques you used in a successfully deployed application?. KDnuggets Polls, Sep 13-27, 2004.

[10] D. Keim.: Databases and Visualization. Tutorial Notes, *ACM-SIGMOD'96*, 1996.

[11] D. Michie, D.J. Spiegelhalter, and C.C. Taylor.: *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.

[12] F. Poulet.: Full-View: A Visual Data Mining Environment. *International Journal of Image and Graphics*, 2(1), pp.127-143, 2002.

[13] J-R. Quinlan.: *C4.5: Programs for Machine Learning*. Morgan-Kaufman Publishers, 1993.