

Graphical Tools for FID

C.Z. Janikow

*Department of Mathematics and Computer Science
University of Missouri – St. Louis*

janikow@umsl.edu

L. Finney

*Department of Mathematics and Computer Science
University of Missouri – St. Louis*

lmfinney@alum.wustl.edu

Abstract – FID3.2 is a public software tool for supervised classification, implementing the popular and efficient recursive partitioning technique of decision trees, while utilizing fuzzy representation and approximate reasoning for dealing with noise and language uncertainty. It is loosely based on the ID3 decision tree algorithm. It handles nominal, continuous, and linguistic attributes and classes. It can also operate with noisy and unknown features, both in training and in testing data. For continuous attributes that are not pre-partitioned, the system generates fuzzy partitioning using either a top-down or a bottom-up method. Finally, FID uses a number of inferences from two classes: set-based and exemplar-based. Recently, we have implemented a GUI interface to work with FID3.2. The interface provides graphical visualization of data, partitions, and the tree, and it also allows limited graphical manipulations. This paper describes the GUI interface.

I. INTRODUCTION

Today, we face huge quantities of information and data, so computer-based warehousing and processing becomes a necessity. In particular, computer programs extracting knowledge from data successfully alleviate the knowledge acquisition bottleneck [1]. Supervised classification programs extract some form of knowledge from training data with known classifications. The extracted knowledge is often in the form of an explicit data structure with some inference method. Among such systems, those building decision trees are very popular, due to their conformity, comprehensibility, accuracy, and performance [9][11].

Decision trees implement the recursive partitioning algorithm, which is a data-driven technique for building tree-based knowledge models [9]. The algorithm starts with all the training data in the root node, and it recursively selects a test, based on the available attributes, to split the node, until some stopping criteria has been satisfied, no more tests are available, or just to avoid overspecialization [11]. The test maximizes some measure, such as the information gain [9][11]. The generated tree can provide comprehensible information about the domain, or it can be used for classification when coupled with a simple inference procedure – match a new datum against the tree, select the leaf that matches it, and report the decision associated with that leaf.

Neural networks provide more robust classification behavior, but they lack similar levels of comprehensibility [2]. A fuzzy approach attempts to bridge the gap between incomprehensible quantitative processing and comprehensible qualitative processing. Fuzzy sets provide basis for fuzzy representation [3]. Fuzzy sets and fuzzy logic allow the

modeling of language-related uncertainties, while providing a symbolic framework for knowledge comprehensibility [4][5][6]. In fuzzy rule-based systems, the symbolic rules provide for ease of understanding and/or transfer of high-level knowledge, while the fuzzy sets, along with fuzzy logic and approximate reasoning methods, provide the ability to model fine knowledge details [6]. Thus fuzzy representation provides means for dealing with problems of uncertainty, noise, and inexact data [7]. It has been successfully applied to problems in many industrial areas [8]

Fuzzy Decision Tree (FID) combines fuzzy representation, and its approximate reasoning, with symbolic decision trees. As such, they provide for handling of language related uncertainty, noise, missing or faulty features, robust behavior, while also providing comprehensible knowledge interpretation. FID has three major components: one for partitioning continuous attributes, one for building an explicit tree, and one for knowledge inference from the tree. These have been described and reported in [12][13][14]. The most recent working version is FID3.2.

Recently, we have added a GUI-based interface to FID3.2, called FID3.2GUI. This paper illustrates the interface and its functionality. Both FID3.2 and FID3.2GUI are available online [15].

II. GUI FUNCTIONS

FID3.2GUI is a new GUI-based wrapper to the existing FID3.2 system. FID3.2GUI allows access to the same functionality in a much easier-to-use interface. It uses the same input files as FID3.2. However, it also allows the user to modify, delete, or add new fuzzy sets to domain attributes. Classifications, attributes, and events are displayed graphically to give the user a much better understanding of the data. Additionally, the generated fuzzy tree is displayed with user-controllable level of detail.

III. GUI INTERFACE

A. Example used for Illustrations

To demonstrate the GUI, we will use simple data from the *FID3.2 User Manual* [15]. The data is defined with two partitioned attributes: a nominal attribute X with values A , B , and C and a continuous attribute $Size$ with three fuzzy sets termed *Small*, *Medium*, and *Large*, and two fuzzy classes: *Yes* and *No*. The attributes and classes are presented in Fig. 1, while the actual data points are listed in Table 1. The data also illustrates the expression language for FID3.2. Features can be listed using actual values from the universe of discourse (such

as 0.4), nominal values (such as *A*), fuzzy terms (such as *Large*), and be undefined (-1). Classes can be listed again using values from the universe of discourse (here only 0 and 1) or the fuzzy term (such as *No*).

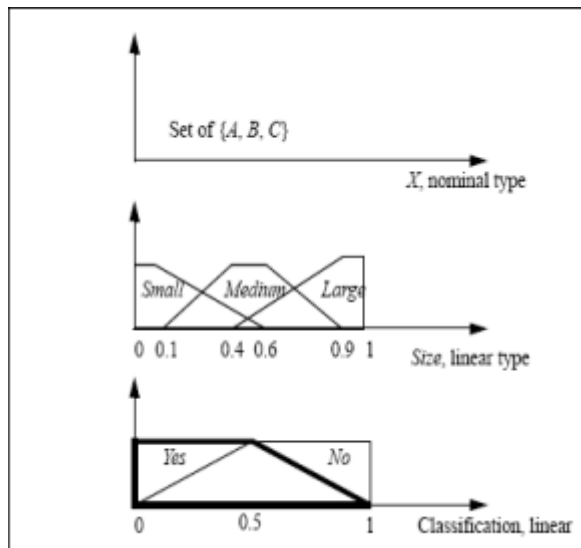


Fig. 1 Attributes and classes.

TABLE I
SampleData

Training event	Numerical domains		Classification	Weight
	<i>X</i>	<i>Size</i>		
1	<i>A</i>	0	0	1
2	-1	0.4	0	1
3	<i>A</i>	0.8	0	1
4	<i>B</i>	0.5	1	1
5	-1	0.5	<i>No</i>	1
6	<i>B</i>	<i>Large</i>	<i>No</i>	1

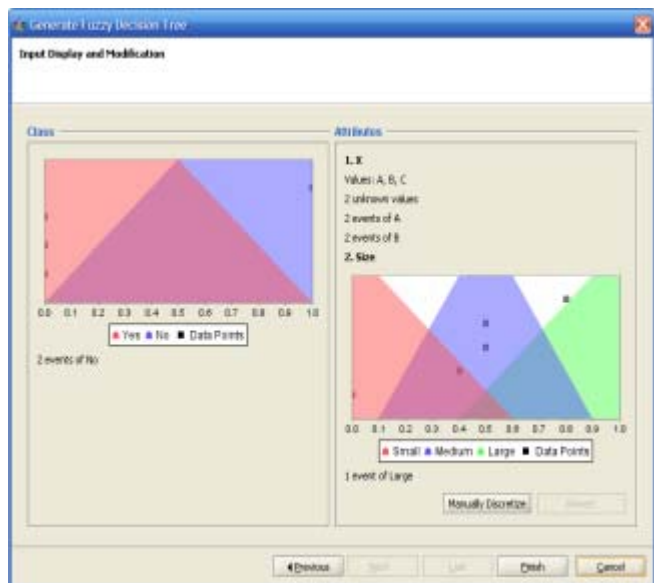


Fig. 2 Input: classes, attributes, and data superimposed.

B. Visualizing Events and Classes

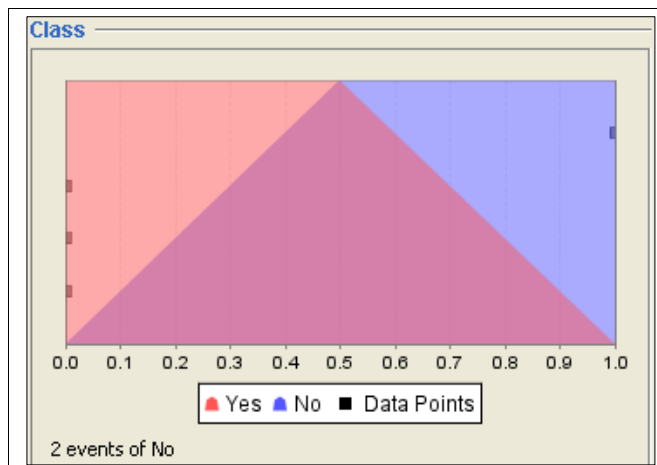


Fig. 3 Classes and data.

After selecting the input data files, FID3.2GUI allows the user to visualize the domain with the actual data superimposed. This is illustrated in Fig. 2, 3, and 4. Fig. 2 illustrates classes and attributes, while Fig. 3 and 4. separate them.

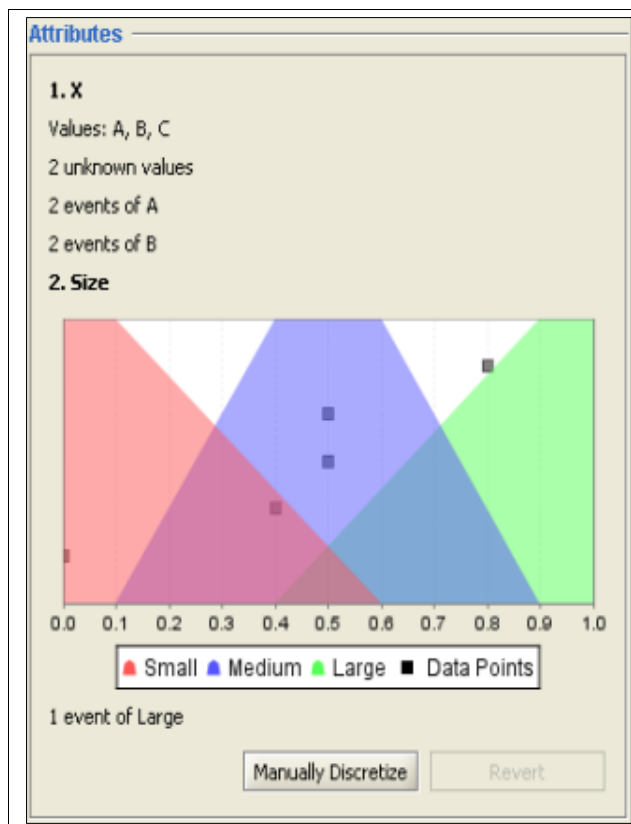


Fig. 4 Details of the attributes and events.

The class definition is illustrated with the corresponding placement of the events superimposed on the class definitions. All input class values are shown – numeric data as data points,

fuzzy terms as fuzzy trapezoids. Multiple numeric data are indicated along the vertical axis, while multiple events for a fuzzy-term class are denoted below the graph (see Fig. 2 left, and Fig. 3).

The attributes are illustrated in Fig. 2 right and also Fig. 4. X is a nominal attribute, so the possible nominal values are listed (A , B , and C). All events are listed below: there are six events in the input, 2 each for A and B , and 2 listed as unknown (specified in the input file with -1).

$Size$ is a linear attribute with a range of 0.0 to 1.0, discretized into three fuzzy sets as in Fig. 1. Among the six events, five have numeric values and one is *Large* (Table 1). The numeric values are superimposed on the graph, while the events described by the fuzzy terms are listed below (see Fig. 4). Again, the vertical axis is used only to indicate multiple events with the same numeric attribute value.

C. Modifying Attribute Definitions

Regardless whether the attribute definitions were given by the user or system generated, by clicking on the *Manually Discretize* button, Fig. 4, the user can modify the definitions of the continuous attributes. This is particularly useful if the system itself is let to discretize some domains – the user can then visualize and adjust the partitions.

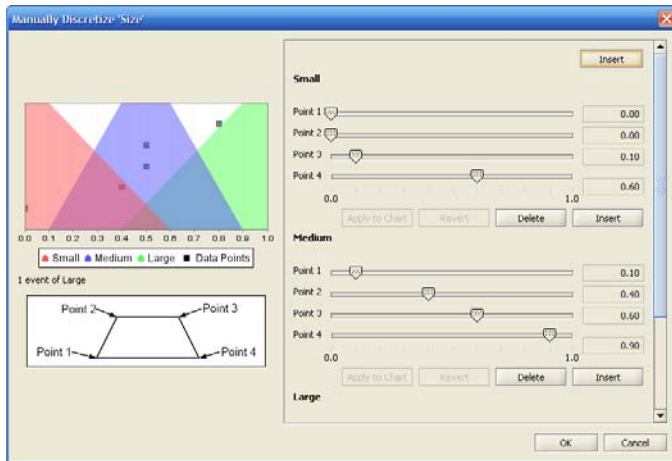


Fig. 5 Manual discretization.

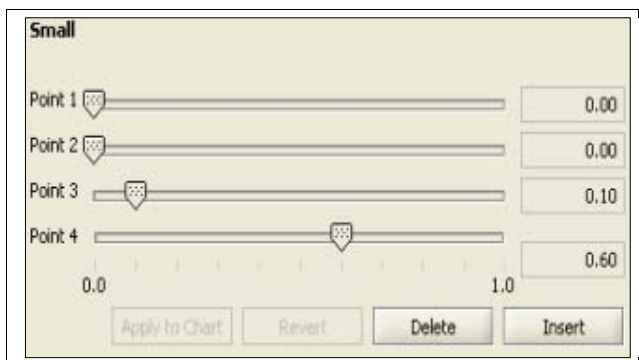


Fig 6. Initial fuzzy set.

Fig. 5 illustrates the interface to modification of the attribute $Size$. The interface shows the definition of fuzzy terms, with the data superimposed, and allows modifications. Each fuzzy term can be modified separately. The user can add or delete fuzzy terms, and can change the definition of a fuzzy term using the sliders – all only as long as the basic fuzzy set principles are not invalidated. This is illustrated in Fig. 6. Moving the slider of the third corner to the right widens the fuzzy set associated with the term *Small* (the system will only allow certain modification). The resulting set is presented in Fig. 7 and illustrated in Fig. 8. Now the user can commit the change (*Apply*) or undo (*Revert*).

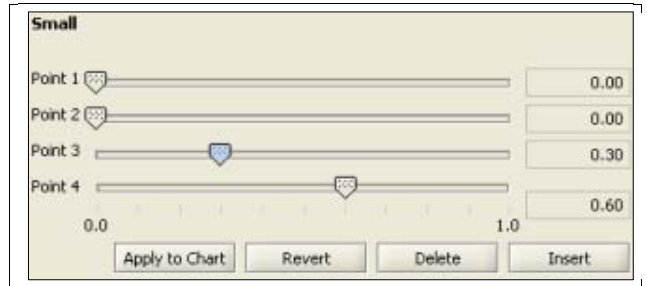


Fig 7. Fuzzy set after change.

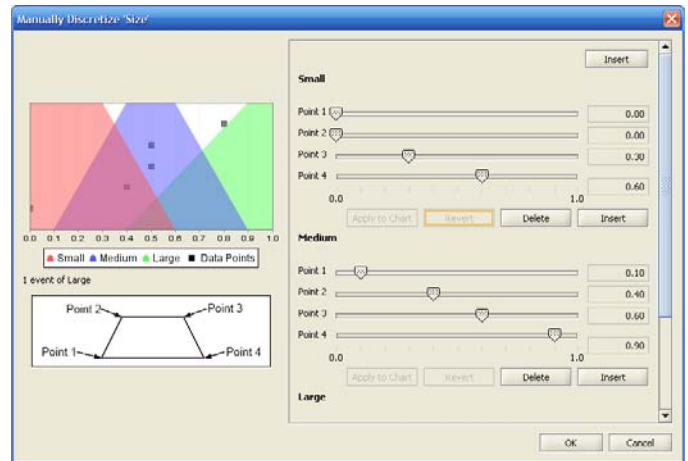


Fig 8. Attribute change applied.

D. Fuzzy Decision Tree

Once the attributes are defined, FID43.2GUI makes use of FID3.2 as the computing engine to create the fuzzy decision tree in a single window. When the tree is generated, FID3.2GUI displays both the input (attribute domains, data, and classifications) and the output tree. This is illustrated in Fig. 9 – the tree is on the right, and again in Fig. 10 – only Root details are shown.

Details of individual leaves can also be accessed by clicking on them. This is illustrated in Fig. 11 – after clicking on the leaf $X=A$.

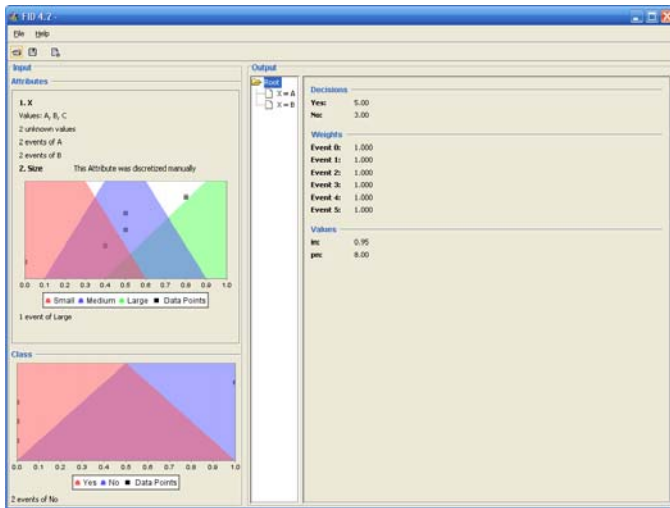


Fig 9. Output tree.

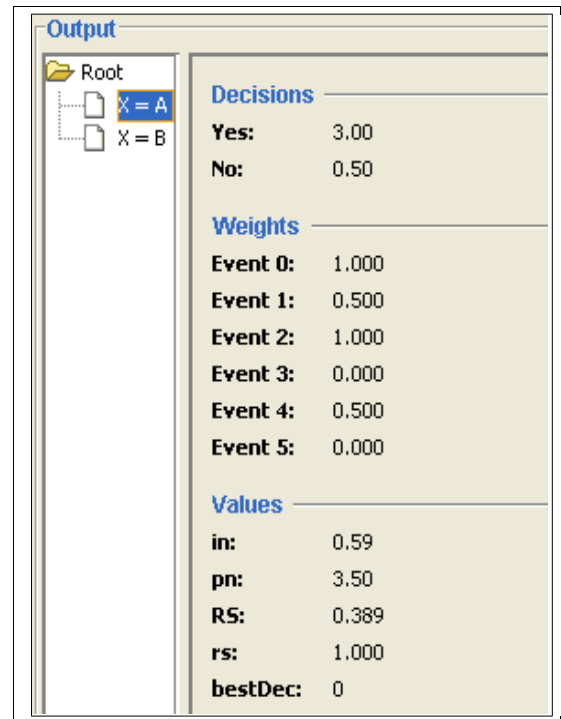


Fig 11. Leaf details.

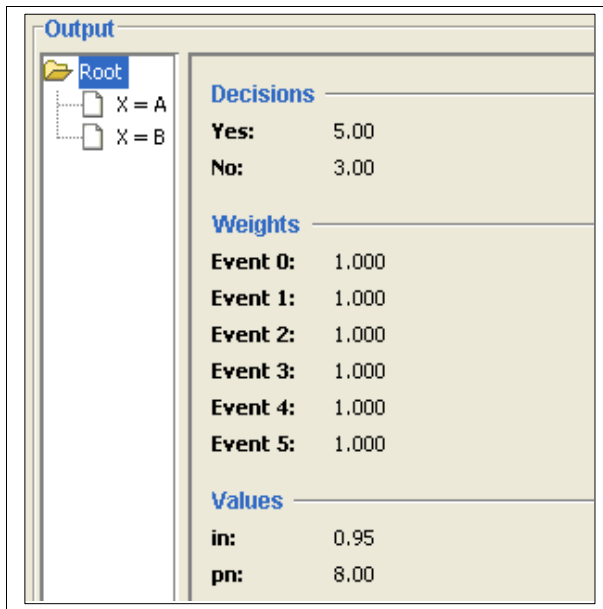


Fig 10. Root details of the fuzzy decision tree.

E. Input Parameters

FID32 and earlier versions allowed run-time parameters to be defined in two ways: through a parameters file and through command-line arguments (which override the default values in the parameter file). Both options still apply in FID4.2GUI – but the command-line parameters are substituted with a dialog window – illustrated in Fig. 12. The values from the window will override the defaults. Note that parameters with discrete values use select drop-down menus.

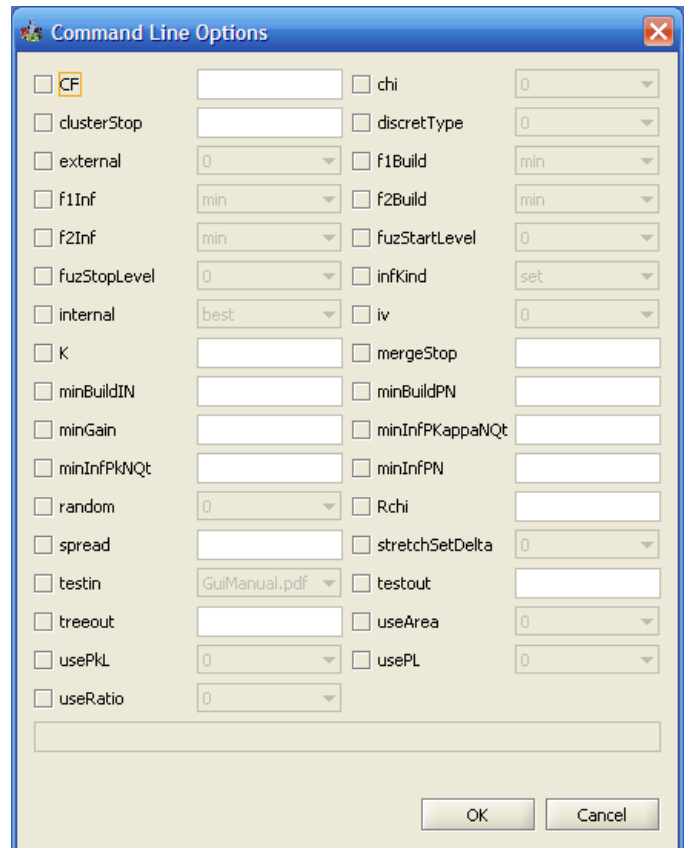


Fig 12 Run parameters.

IV. SUMMARY

This paper describes a new graphical interface to the popular FID3.2 Fuzzy Decision Tree. This interface provides better visualization of the data, attributes, and classes, and how they relate. It allows for manual manipulations of the attributes, with data superimposed, which is particularly useful when to inspect and adjust the system-generated partitions. This interface also simplifies the run parameters, and allows visualization of the details of the generated fuzzy tree.

We hope that this visual tool will further increase the popularity of the tool. Both FID3.2 and FID3.2GUI are available on line [15].

REFERENCES

- [1] R.S. Michalski, "Understanding the nature of learning", In *Machine Learning: An Artificial Intelligence Approach*, R. Michalski, J. Carbonell & T. Mitchell (eds.), Vol. II, pp. 3-26. Morgan Kaufmann, 1986
- [2] S. Sestino & T. Dillon, "Using single-layered neural networks for the extraction of conjunctive rules and hierarchical classifications", *Journal of Applied Intelligence* 1, pp. 157- 173, 1991.
- [3] L.A. Zadeh, "Fuzzy sets", *Information and Control* 8 (1965), pp. 338-353.
- [4] L.A. Zadeh, "Fuzzy logic and approximate reasoning", *Synthese* 30 (1975), pp. 407-428.
- [5] L.A. Zadeh, "A theory of approximate reasoning", In Hayes, Michie & Mikulich (eds) *Machine Intelligence* 9 (1979), pp. 149-194.
- [6] L.A. Zadeh, "The role of fuzzy logic in the management of uncertainty in expert systems", *Fuzzy Sets and Systems*, 11, 1983, pp. 199-227.
- [7] D. McNeill & P. Freiberger, "Fuzzy logic", Simon & Schuster, 1993.
- [8] A. Kandel & G. Langholz (eds.), "Fuzzy control systems", CRC, 1993.
- [9] J.R. Quinlan, "Induction on decision trees", *Machine Learning*, Vol. 1, 1986, pp. 81-106.
- [10] J.R. Quinlan, "Unknown attribute-values in induction", In *Proceedings of the Sixth International Workshop on Machine Learning*, 1989, pp. 164-168.
- [11] J.R. Quinlan, C4.5: "Programs for machine learning", Morgan Kaufmann, San Mateo, CA. 1993.
- [12] C.Z. Janikow, "Fuzzy decision trees: issues and methods", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 28, Issue 1, pp. 1-14. 1998.
- [13] C.Z. Janikow, M. Fajfer, "Fuzzy partitioning with FID3.1", *Proceedings of the 18th International Conference of the North American Fuzzy Information Society*, NY 1999, pp. 467-471.
- [14] M. Fajfer, C.Z. Janikow, "Bottom-up partitioning in fuzzy decision trees", *Proceedings of the 19th International Conference of the North American Fuzzy Information Society*, Atlanta 2000, pp. 326-330.
- [15] C.Z. Janikow. <http://www.cs.ums1.edu/~janikow/fid>.