FrIDA — A Free Intelligent Data Analysis Toolbox

Christian Borgelt and Gil González Rodríguez

***Abstract*—This paper describes a Java-based graphical user interface to a large number of data analysis programs the first author has written in C over the years. In addition, this toolbox is equipped with basic visualization capabilities, like scatter plots and bar charts, but also with specialized visualization modules for decision and regression trees as well as prototype-based classifiers. The architecture is like a toolbox: individual tools refer to the different data analysis methods. All parts of this toolbox (Java as well as C based) are free and open software under the Gnu Lesser (Library) Public License.**

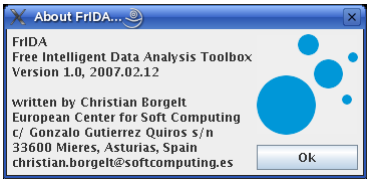

Fig. 1. FrIDA — Free Intelligent Data Analysis Toolbox.

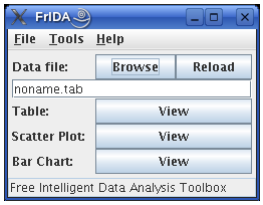

Fig. 2. The main window of FrIDA.

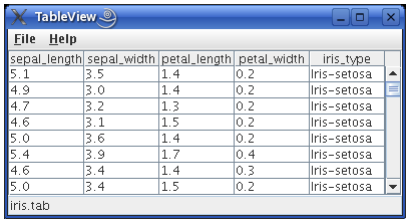

Fig. 3. A simple table view.

I. Introduction

Since 1996 the first author of this paper has been developing a large number of data mining and intelligent data analysis programs. However, most of these programs are command line programs in C and thus not particularly user friendly. Even though this did not hinder them to become popular, as expert users may even prefer command line programs due to the possibility of using them in scripts, a novice user is usually repelled by a command line interface. Starting with the decision and regression tree programs and continuing with association rule induction as well as fuzzy and probabilistic cluster induction, individual system-independent graphical user interfaces in Java were then provided.

In a recent effort, we have tried to combine these individual user interfaces into a single toolbox, in order to make them more easily accessible, and also in order to convey to a user, who may have been using one of the programs already, the large variety of methods that are available. As a consequence the FrIDA program was written, which combines all individual user interfaces that were developed so far in a uniform and consistent architecture. A main focus in the development was to provide quick and easy ways to view generated models and prediction results. This is reflected in a large number of "View" buttons spread over the dialog boxes, which allow a user to view a table or a generated model with a single click. In addition, the main window (see Figure 2) allows for simple and direct data visualization in scatter plots and bar charts.

Christian Borgelt and Gil González Rodríguez are both with the European Center for Soft Computing, Edificio Científico-Tecnológico, c/ Gonzalo Gutiérrez Quirós s/n, 33600 Mieres, Asturias, Spain (email: {christian.borgelt,gil.gonzalez}@softcomputing.es).

This paper is organized as follows: in Section II we present the basic visualization capabilities of FrIDA, which are easily accessible from the main window. In Section III we briefly discuss the data formats that can be handled by FrIDA and how FrIDA can be configured to use them. Sections IV and V are devoted to two special tools. In the former section, prototype-based clustering (fuzzy and probabilistic) and FrIDA's user interface to the underlying program is discussed, as well as it's visualization module for cluster models. Section V then turns to the highly popular method of decision trees (the FrIDa implementation is very popular as an independent program), and Section VI lists other available methods (without a detailed discussion). Section VII summarizes the presentation.

II. Data Visualization

The main window of FrIDA (see Figure 2) allows a user to easily load a data table (the format of which is fairly flexible, see Section III) and to visualize it. In addition to the mandatory simple table viewer (see Figure 3), 3-dimensional scatter plots and bar charts can be generated. Examples are shown in Figures 4 and 6, which depict a scatter plot and a bar chart of the well-known Iris data [11]. These visualization modules are highly configurable, for example, w.r.t. layout and color, as the dialog boxes shown in Figures 5 (attribute selector for the scatter plot) and 7 (layout dialog box for the bar chart) demonstrate. Of course, in both visualizations the displayed structure (cube enclosing the data or the bar chart) can easily and freely be rotated, moved, and zoomed with the mouse.

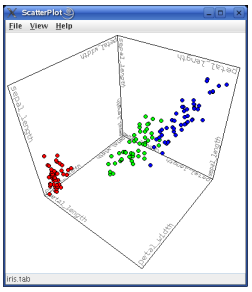

Fig. 4. A 3-dimensional scatter plot.

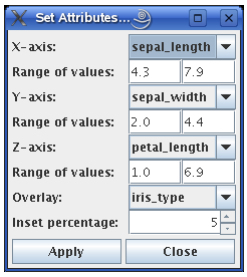

Fig. 5. Attribute selector.

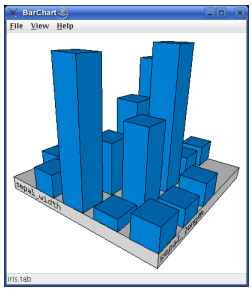

Fig. 6. A 3-dimensional bar chart.

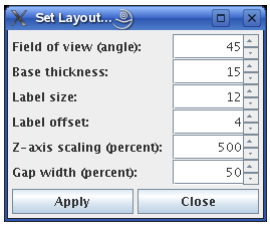

Fig. 7. Layout dialog box.

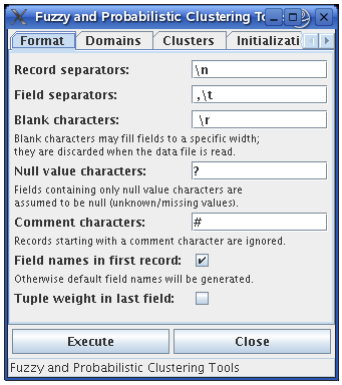

Fig. 8. The data format configuration tab.

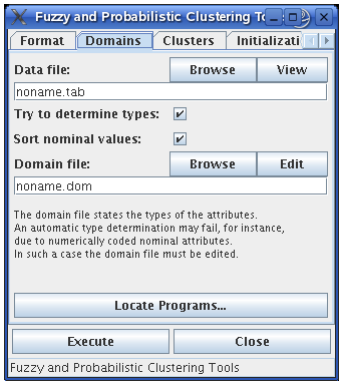

Fig. 9. The domain determination tab.

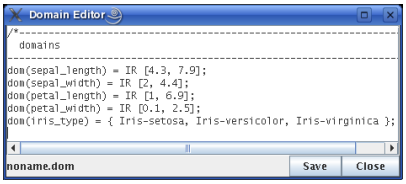

Fig. 10. A simple domain description editor.

The main window also provides access to all intelligent data analysis tools, which are available from a simple menu. They are organized as individual tool boxes, which combine interfaces to all programs that are needed for a specific method. For example, the decision and regression tree tools combine interfaces to a program for the data type determination, decision or regression tree induction, pruning and execution. The clustering tools combine interfaces for data type determination as well as cluster induction and execution. These examples already indicate that some interfaces (here the automatic data type determination) are available multiple times, so that they are close at hand wherever one may need them.

III. Data Format

The data format that can be read by FrIDA is defined by five sets of characters. It is assumed that the input file is divided into records, and each record into fields, with special separator characters (record separators and field separators) providing for this division. In addition, blank characters may have been used to fill fields to specified with, for example, in order to align them in an editor. These blank characters will be removed when the file is read. The fourth set of characters identifies null values (fields containing only blanks and null value characters are assumed to be null), the fifth comment records (a record starting with a common character is ignored).

In addition, it is possible to generate default field/column names, in case the data file does not contain them, but consists purely of data. Finally, each record may contain in its last field an occurrence counter, which can be used to handle multiple occurrences of the same tuple without having to provide duplicates. (For the viewers, however, such a weight column is always treated as a normal data column. It is only the model generation programs, like, for example, a decision tree inducer, that interprete these values as weights and adapt the model induction process accordingly.)

The data format can be configured independently for each toolbox, using the data format tab, which is always the first in all individual toolboxes (see Figure 8). The main window is also independent and maintains its own settings of the data format parameters. It can be set through a data format dialog available in the “File” menu of the main window. However, the possibility of copying these settings between the main window and the toolboxes will also be available soon.

Once the data format is fixed, the domains of the individual fields (columns) have to be determined, which is made simple by an automatic type determination module, as shown in Figure 9. Since such an automatic type determination can fail (for example, if nominal attributes are coded by integer numbers), the domain description may also be altered with a simple editor, see Figure 10. (However, this way of changing the types of attributes will be improved in the near future.)

Since the underlying programs are written in C, this tab also comprises the possibility to locate them on the file system, in case they have not been placed in the same directory as the graphical user interface and are not reachable through the “PATH” environment variable. In the standard distribution of FrIDA such locating is not necessary.

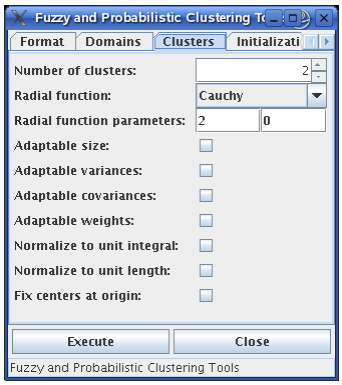

Fig. 11. Fuzzy/probabilistic cluster parameters.

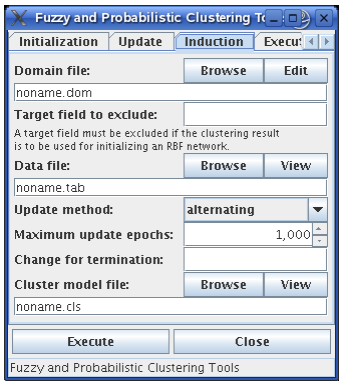

Fig. 12. Fuzzy/probabilistic cluster induction.

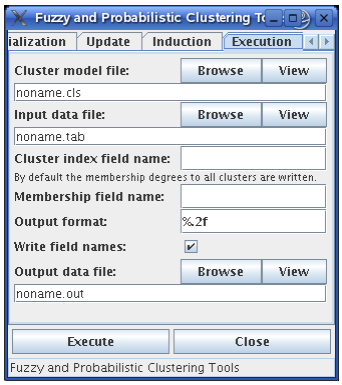

Fig. 13. Fuzzy/probabilistic cluster execution.

IV. Fuzzy and Probabilistic Clustering

FrIDA comprises a very flexible program for prototype-based clustering, which combines classical (crisp) k-means clustering [14], standard fuzzy c-means clustering [1], [3], [15], more sophisticated fuzzy clustering [13], [12], expectation maximization for a mixture of Gaussians [10], [4], hard and soft learning vector quantization [17], [18], [22] as well as several extended and generalized methods [6], [7], [8].

An impression of the power of this program is given by the tabs of the corresponding toolbox that are shown in Figures 15 to 13, which show the basic parameters that can be used to configure the cluster prototypes as well as the cluster induction and execution. Note, for example, that the cluster prototypes can have adaptable sizes, variances, covariances, and weights (or prior probabilities). The cluster centers may be normalized to unit length, or they may be fixed at the origin, allowing only the size and shape to vary. These latter options can be advantageous for document clustering [8].

The cluster induction tab (Figure 12) makes it possible to keep, but not use, a target attribute, so that the clustering result can be used to initialize a radial basis function neural network. The cluster execution tab allows a user to execute a cluster model like a classifier, assigning membership degrees for each cluster to each data tuple, or assigning each tuple to the best cluster with an indication of the highest membership degree.

Clustering results may be visualized, wherever a cluster model is used as an input or output, by simply pressing the accompanying "View" button. An example screen shot (of a Gustafson-Kessel clustering [13] result for the Iris data [11] is shown in Figure 14.[1] The saturation of the color indicates the degree of membership to the cluster having the highest degree of membership and colors code the different clusters. In order to make the individual clusters more easily visible, their centers can be marked, together with the 1-, 2-, or 3-σ ellipses of the cluster-specific covariances matrices. (Figure 14 shows only the centers—as small squares— and the 1-σ ellipses.)

[1]This screen shot is of a visualization program written in C, which is currently ported to Java, to make it platform independent. However, a specialized version for Windows exists, so that even now it can be executed on the most widely used systems.

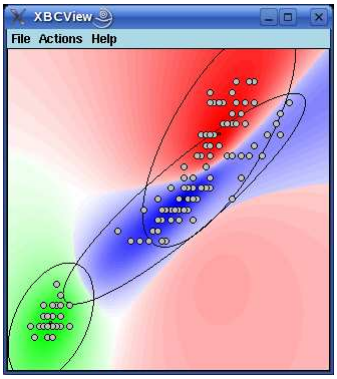

Fig. 14. Fuzzy and probabilistic cluster visualization.

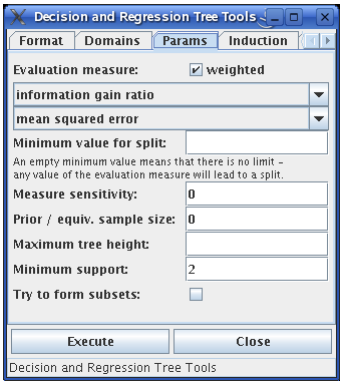

Fig. 15. Decision/Regression Tree parameters.

V. Decision and Regression Trees

No data analysis program is complete without a possibility to induce and execute decision trees. FrIDA contains a very flexible version of a decision tree induction program. Its most distinguishing feature is the large number of attribute evaluation/selection measures, which allow to configure so that it behaves similar to ID3 (information gain), C4.5 (information gain ratio), CART (Gini index), or CHAID (χ^2 measure). Several other common features are also supported, see Figure 15.

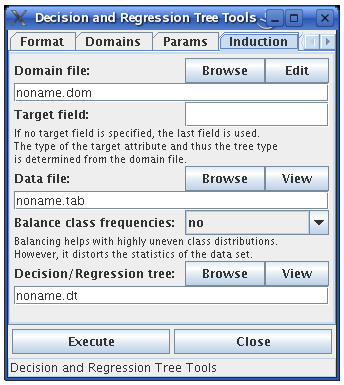

Fig. 16. Decision/Regression Tree induction.

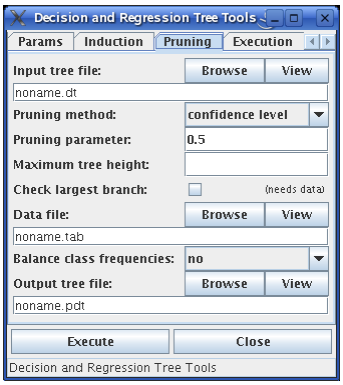

Fig. 17. Decision/Regression Tree pruning.

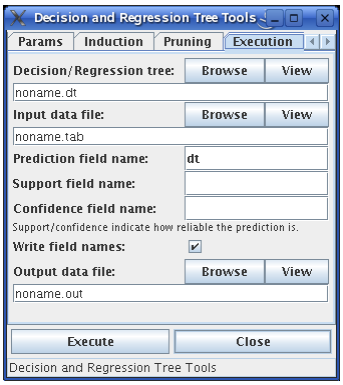

Fig. 18. Decision/Regression Tree execution.

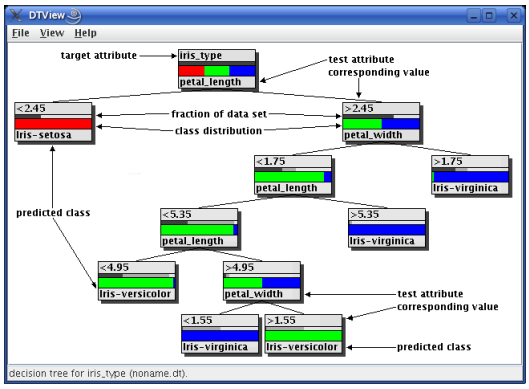

Fig. 19. Decision Tree visualization.

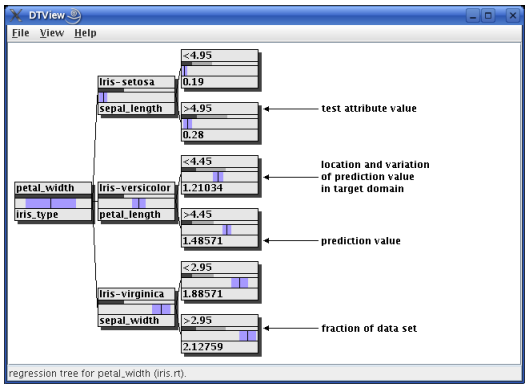

Fig. 20. Regression Tree visualization.

Figures 16 to 18 show the tabs, with which a decision or regression tree can be induced, pruned, or executed on new data. Pruning may use a second data table (different from the one used for induction, reduced-error pruning) or may be achieved with pessimistic or confidence level pruning. Execution allows for computing a tuple-specific confidence of the classification result, based on the decision or regression tree leaf with which the classification was achieved.

However, one of the strongest features of the decision and regression tree tools is the visualization module. Its layout is highly configurable, as can already be seen from Figures 19 and 20, which show a decision tree and a regression tree for the Iris data, together with some explanations of the different features of the visualization. The tree display shows the class distribution (for decision trees) or the value distribution (for regression trees) for the individual nodes, as well as the data distribution on the nodes, both relative to the total data set and relative to the parent node.

VI. Other Data Analysis Methods

Apart from the two modules described above, of which the clustering module fits best into this conference due its strengths w.r.t. fuzzy clustering and fuzzy learning vector quantization, while the decision and regression tree module is a mandatory ingredient of every intelligent data analysis or data mining platform, FrIDA comprises modules for

- naive and full Bayes classifiers
- radial basis function neural networks
- multilayer perceptrons
- multivariate polynomial regression
- association rule induction

Not all of these modules are currently finished, but since the class structure used in FrIDA for the toolbox dialogs is fairly uniform, it is no problem to finish most, if not all of the toolboxes until the conference.

VII. Summary

In this paper we presented FrIDA, a Free Intelligent Data Analysis Toolbox. The program is still under development, but is already very powerful and supports all standard data mining and data analysis techniques. Due to the toolbox concept underlying it, it is very easily extendable, as new tools can be integrated basically by extending the tools menu of the main window.

Software

The toolbox described in this paper as well as all accompanying C programs will soon be made available at http://www.borgelt.net/frida.html.

REFERENCES

[1] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, NY, USA 1981

[2] J.C. Bezdek and N. Pal. *Fuzzy Models for Pattern Recognition*. IEEE Press, New York, NY, USA 1992

[3] J.C. Bezdek, J. Keller, R. Krishnapuram, and N. Pal. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Kluwer, Dordrecht, Netherlands 1999

[4] J. Bilmes. A Gentle Tutorial on the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. *Tech. Report ICSI-TR-97-021*. University of Berkeley, CA, USA 1997

[5] C. Borgelt and R. Kruse. Attributauswahlmaße für die Induktion von Entscheidungsbäumen: Ein Überblick. In: [19], 77–98

[6] C. Borgelt and R. Kruse. Speeding Up Fuzzy Clustering with Neural Network Techniques. *Proc. 12th IEEE Int. Conference on Fuzzy Systems (FUZZ-IEEE'03, St. Louis, MO, USA)*, on CDROM. IEEE Press, Piscataway, NJ, USA 2003

[7] C. Borgelt and R. Kruse. Shape and Size Regularization in Expectation Maximization and Fuzzy Clustering. *Proc. 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD 2004, Pisa, Italy)*, 52–62. Springer-Verlag, Berlin, Germany 2004

[8] C. Borgelt. *Prototype-based Classification and Clustering*. Habilitation thesis, University of Magdeburg, Germany 2005

[9] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, USA 1984

[10] A.P. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society (Series B)* 39:1–38. Blackwell, Oxford, United Kingdom 1977

[11] R.A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics* 7(2):179–188. Cambridge University Press, Cambridge, United Kingdom 1936

[12] I. Gath and A.B. Geva. Unsupervised Optimal Fuzzy Clustering. *IEEE on Trans. Pattern Analysis and Machine Intelligence (PAMI)* 11:773–781. IEEE Press, Piscataway, NJ, USA 1989. Reprinted in [2], 211–218

[13] E.E. Gustafson and W.C. Kessel. Fuzzy Clustering with a Fuzzy Covariance Matrix. *Proc. of the IEEE Conf. on Decision and Control (CDC 1979, San Diego, CA)*, 761–766. IEEE Press, Piscataway, NJ, USA 1979. Reprinted in [2], 117–122

[14] J.A. Hartigan and M.A. Wong. A k-means Clustering Algorithm. *Applied Statistics* 28:100–108. Blackwell, Oxford, United Kingdom 1979

[15] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis*. J. Wiley & Sons, Chichester, England 1999

[16] L. Kaufman and P. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. J. Wiley & Sons, New York, NY, USA 1990

[17] T. Kohonen. *Learning Vector Quantization for Pattern Recognition*. Technical Report TKK-F-A601. Helsinki University of Technology, Finland 1986

[18] T. Kohonen. Improved Versions of Learning Vector Quantization. *Proc. Int. Joint Conference on Neural Networks* 1:545–550. IEE Computer Society Press, San Diego, CA, USA 1990

[19] G. Nakhaeizadeh, ed. *Data Mining: Theoretische Aspekte und Anwendungen*. Physica-Verlag, Heidelberg, Germany 1998

[20] J.R. Quinlan. Induction of Decision Trees. *Machine Learning* 1:81–106. Kluwer, Dordrecht, Netherlands 1986

[21] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, USA 1993

[22] S. Seo and K. Obermayer. Soft Learning Vector Quantization. *Neural Computation* 15(7):1589–1604. MIT Press, Cambridge, MA, USA 2003