

A new fingram-based software tool for visual representation and analysis of fuzzy association rules

David P. Pancho, Jose M. Alonso
European Centre for Soft Computing
33600 Mieres, Asturias, Spain

Email: {david.perez,jose.alonso}@softcomputing.es

Jesús Alcalá-Fdez
Department of Computer Science and Artificial Intelligence
University of Granada, CITIC-UGR
18071 Granada, Spain
Email: jalcala@decsai.ugr.es

Abstract—This paper presents an open source software tool for generation, visual representation and expert analysis of the so-called fuzzy inference-grams (fingrams in short). Fingrams are extended to fuzzy association rules in this work, expanding the scope of this methodology. The tool automatically generates fingrams and displays them to the user for expert analysis. It takes as input a simple configuration file that may be provided by whatever fuzzy modeling tool (it is actually provided by two widely known tools, GUAJE and KEEL). Fingrams graphically represent multi-dimensional fuzzy systems in 2D. The first version of this new software handles three different kinds of fuzzy systems (fuzzy rule-based systems for classification and regression, but also fuzzy association rules). Fingrams illustrate the interaction among fuzzy rules in terms of rule co-firing, i.e., paying attention to pairs of rules fired at the same time by a given input. Their visualization looks like a social network where nodes represent fuzzy rules while edges show the degree of interaction among pairs of rules. Thus, both representation and analysis are supported by social network analysis techniques. The utility of the new software tool is illustrated in a case study on visual representation and analysis of association rules.

Index Terms—Fuzzy System Software, Fuzzy Association Rules, Comprehensibility Analysis, Expert Analysis, Social Network Analysis, Open Source Software, Java.

I. INTRODUCTION

Fuzzy association rules have been successfully applied to a wide variety of problems [1] with the aim of identifying and representing dependencies among items. For instance, effective fuzzy associative classification [2], mining of medical databases [3], and so on. Unfortunately, systems made up of automatically extracted fuzzy association rules are rarely as interpretable as desired. Such systems are usually compound of a large number of complex rules that involve many different variables and are therefore quite hard to interpret.

Interpretability is an appreciated and distinguishing capability of fuzzy systems [4]. However, even though it becomes essential in the case of applications involving extensive interaction with human beings, it is important to highlight the fact that fuzzy systems are not interpretable *per se*. Of course, the high semantic expressivity (close to natural language) of linguistic variables [5] and rules [6], [7] favors interpretability. Anyway, only a careful design guarantees the fulfillment of interpretability requirements [8].

Fuzzy inference-grams (fingrams) have recently arisen as a powerful tool to support an interpretability-driven design of fuzzy systems [9]. Fingrams make it easy to analyze graphically the fuzzy inference layer of fuzzy systems from a comprehensibility point of view. With that aim, fuzzy rule bases are represented in an aesthetically pleasing way, thus displaying the interactions among hundreds of rules in the form of close-to-tree graphs easy to interpret. Such graphs are actually social networks which comprise nodes representing fuzzy rules and edges characterizing co-firing among rules. Moreover, they are not affected by the well-known curse of dimensionality characteristic of fuzzy systems. Therefore, experts can comfortably analyze fingrams, even in case they include a large number of rules, with the aim of understanding the structure and behavior of the represented fuzzy system.

The last release of GUAJE¹ provides a software module for fingram generation and analysis [10], [11]. It has been successfully applied to the analysis of real classification and regression problems [9].

The current work introduces a stand-alone software tool able to generate and visualize fingrams. The main novelty resides in the fact that it handles not only fuzzy systems designed by GUAJE but fuzzy systems designed by whatever fuzzy modeling tool. To do so, the new tool takes as input a standard configuration file easy to be generated. In addition to GUAJE, the last release of a widely-known tool like KEEL² [12], [13] is also able to automatically produce the required configuration file. We have developed and integrated a new software module into KEEL with that aim. As a result, the new tool manages not only fuzzy rules for classification and regression but also fuzzy association rules. Obviously, as detailed in this paper, fingrams have been extended to properly represent fuzzy association rules. Other kinds of fuzzy rules will be considered in the near future.

The rest of the contribution is organized as follows. Section II presents some preliminaries including basic aspects

¹GUAJE stands for Generating Understandable and Accurate fuzzy models in a Java Environment: [<http://www.softcomputing.es/guaje>]

²KEEL stands for Knowledge Extraction based on Evolutionary Learning: [<http://www.keel.es>]

related to fuzzy association rules and figrams, as well as a brief introduction to KEEL software. Section III introduces the new stand-alone figram generation and visualization software, as well as the structure of the defined configuration files. Section IV presents a case study where the main benefits of considering figrams in relation with fuzzy association rules are sketched. Finally, some conclusions and future works are pointed out in Section V.

II. PRELIMINARIES

In this section, we first introduce the basic definitions of fuzzy association rules along with some related quality measures. Then, we describe the basic aspects of figrams. Finally, we present a brief description of the KEEL software tool remarking its main features.

A. Fuzzy Association Rules

Association rules are used to identify and represent dependencies between items in a dataset [14]. These are representations of the type $X \rightarrow Y$, in which X and Y are item sets and $X \cap Y = \emptyset$. Therefore, if the items in X exist in a pattern then it is highly probable that the items in Y are also in the pattern. In addition, X and Y should not have items in common [15], [16]. A high number of previous studies on mining association rules focused on datasets with discrete or binary values; however, in real-world applications, data usually consists of quantitative values. Designing Data Mining algorithms, able to deal with various types of data, presents a challenge to workers in this research field.

Fuzzy sets have frequently been used to describe associations between data in Data Mining because of its similarity to human reasoning [17]. The use of fuzzy set theory extends the types of relationships between data, improves the interpretation of rules in linguistic terms, and avoids unnatural boundaries in the partitioning of the attribute domains. Different studies have presented methods for mining fuzzy association rules from datasets with quantitative values [1], [18], [19], [20], [21].

Let us consider a simple dataset with two attributes (A_1 and A_2) and three linguistic terms (LOW, MIDDLE, HIGH). A simple but illustrative example of fuzzy association rule is A_1 is LOW \rightarrow A_2 is MIDDLE.

Support and *Confidence* are the most common measures used to assess association rules. For a fuzzy association rule $X \rightarrow Y$, they are defined as:

$$Support(X \rightarrow Y) = \frac{\sum_{x_p \in D} \mu_{XY}(x_p)}{|N|} \quad (1)$$

$$Confidence(X \rightarrow Y) = \frac{\sum_{x_p \in D} \mu_{XY}(x_p)}{\sum_{x_p \in D} \mu_X(x_p)} \quad (2)$$

where $\mu_X(x_p)$ is the matching degree of the pattern x_p with the antecedent of the rule; $\mu_{XY}(x_p)$ is the matching degree of the pattern x_p with the antecedent and consequent of the rule; and $|N|$ is the number of patterns in the dataset D .

The classic techniques for mining association rules attempt to discover rules whose support and confidence are greater than the user-defined thresholds *minimum support* and *minimum confidence*. However, several authors have noted some drawbacks of this framework that lead it to find many more rules than it should [22], [23]. In recent years, several researchers have proposed other measures for the selection and ranking of patterns according to their potential interest to the user [24]. For instance, the *Lift* [25] measure represents the ratio between the actual confidence of the rule and its expected confidence. This measure obtains values in $[0, \infty)$, detecting negative dependence (in case *Lift* is smaller than 1) or independence between items. It is defined as:

$$Lift(X \rightarrow Y) = \frac{Confidence(X \rightarrow Y)}{\sum_{x_p \in D} \mu_Y(x_p) / |N|} \quad (3)$$

B. Figrams

The notion of figram was introduced in [26]. The graphical representation of figrams can be seen as a social network where fuzzy rules are represented by nodes, while the relations among rules are represented by weighted edges whose value is computed using a specific metric, usually a rule co-firing metric. In addition, we have recently proposed a novel methodology for visual representation and exploratory analysis of the fuzzy inference process in fuzzy rule-based systems (FRBSs) based on figrams [9]. It includes the following steps:

- 1) **Network generation:** Given a dataset, a rule base, a fuzzy reasoning mechanism and a rule co-firing metric, then the complete set of relations among rules is constructed, obtaining an initial social network. It is formalized in a square matrix and represented by a graph.
- 2) **Network scaling:** Since the previous network is usually very dense and difficult to analyze, a scaling process is required. As result, we obtain a simplified social network highlighting the most relevant relations among rules while preserving the backbone of the initial network.
- 3) **Network drawing:** A layout algorithm automatically places the nodes and edges of the scaled network guided by aesthetical criteria.
- 4) **Network visualization:** The user can analyze the final graph (that is what we call figram), studying the general structure of the resultant network or zooming out a particular subset of nodes/rules. The analysis of figrams is supported by social network analysis techniques (centrality-based analysis techniques, community mining techniques, and so on). It offers many valuable possibilities, such as, understanding the structure and behavior of the related fuzzy rule base, uncovering the most significant rules according to specific criteria, etc.

A first software module for figram generation and analysis was recently developed into the free software GUAJE [10], [11]. Moreover, GUAJE provides a tutorial guide [27] for creation and analysis of figrams. It was especially thought

for allowing end-users not used to working with GUAJE to handle fignrams easily.

C. KEEL

KEEL is a Java software tool for assessing evolutionary algorithms for Data Mining problems: classification, regression, pattern mining, clustering, and so on [12], [13].



Fig. 1. The main window of KEEL.

The current version of this tool (its main window is depicted in Fig. 1) consists of four blocks:

- 1) **Data Management:** This block provides a set of tools which can be used to export KEEL format datasets to other formats, to import other format datasets to KEEL format, to visualize and edit existing KEEL format datasets, to make partitions of datasets included in KEEL, etc.
- 2) **Design of Experiments:** The Experimental block is used to design the desired experiments using a graphical interface. The objective is to use available datasets and algorithms to generate a directory structure with all the necessary files needed to run the designed experiments in the local computer selected by the user. This module provides many options in different areas: type of validation, type of learning (unsupervised learning, classification, etc.) and so on.
- 3) **Educational Experiments:** This block has the main objective of designing the desired experiments using a graphical interface and an on-line execution of those experiments from an educative perspective. It is also possible to stop and resume experiments as you need. Furthermore, you can see the results of those experiments into the environment.
- 4) **Modules:** This block includes new modules extending the functionalities of the KEEL software tool: imbalanced learning, non-parametric statistical analysis, and so on.

These blocks make KEEL software tool useful for different types of user. Next, a brief description of the main features of KEEL is shown:

- It contains a Knowledge Extraction Algorithms Library, compound of supervised and unsupervised algorithms, remarking the incorporation of multiple evolutionary learning algorithms. Moreover, it includes a representative set of methods for different areas of Data Mining such as association rules, decision trees, fuzzy rule based systems, and so on.
- It provides pre-processing algorithms: transformation, discretization, instance and feature selection, etc.
- It has a statistical analysis library to analyze algorithms.
- It provides a user-friendly interface, oriented to the analysis of algorithms.
- It is aimed at designing experiments with multiple datasets and algorithms connected among them to obtain the desired results. Experiments are generated from the interface for an off-line run in the local computer selected by the user.
- It provides an educational support which allows the creation of experiments in on-line mode in order to learn the performance of several algorithms.

For further details about the KEEL software tool, please refer to [12], [13].

III. A NEW SOFTWARE TOOL FOR AUTOMATIC GENERATION AND VISUALIZATION OF FIGNRAMS

This section introduces the new software tool for fignram generation and analysis. It has been designed as a stand-alone tool, which can be invoked from any fuzzy design software. With this solution, we encourage developers of fuzzy system software to enhance their own tools by adding modules able to take profit of the fignrams produced by this new tool.

The software developed³ requires a *.fs* input file with all demanded information about the designed fuzzy system, for instance the accumulated firing degrees for all the examples covered by each rule. Notice that, GUAJE and KEEL, in their last releases, present modules for producing the information needed (we have developed new software modules for that aim as part of this work).

More concisely, Table 2 details the structure of such configuration file. The first line should contain the type of rules (Classification, Regression or Association). Second line changes depending on the selected type of rules. It includes the classes (with the number of examples per class in brackets) for classification problems; the output linguistic labels (with number of examples corresponding to each label in brackets) in case of regression problems⁴; and the possible output variables along with the attached linguistic terms for association rules.

The next three lines show three thresholds. Blank threshold represents the minimum level an example should fire a rule to be taken into account. Goodness thresholds (low and high) correspond to the minimum and maximum levels of rule firing

³Fignrams Generator software is freely available in [<https://sourceforge.net/projects/fignrams/>]

⁴The software permits the use of multi-input-single-output (MISO) FRBSs for regression problems.

```

1 Classification/Regression/Association
2 C1(100),C2(50), .../Low(20), Medium(10), .../Vble_1(LABEL_0,LABEL_1), Vble_2(LABEL_0), ...
3
4 Blank threshold: 0.1
5 Goodness Threshold (high): 0.5
6 Goodness Threshold (low): 0.1
7
8 Rules: 11
9
10 Rule1: If Vble_1 is Low THEN Class is C1
11 Correct Conclusion => (22.93300) (0.63703) => 98(0.333), 99(0.264), 100(0.123), ...
12 Incorrect Conclusion => (4.89900) (0.22268) => 17(0.109), 21(0.235), 47(0.212), ...
13 Positive Examples => (20.47700) (0.68257) => 98(0.333), 100(0.123), 102(0.421), ...
14 Negative Examples => (4.89900) (0.22268) => 17(0.109), 21(0.235), 47(0.212), ...
15
16 Rule2: If Vble_2 is Medium THEN Vble_Output is Low
17 Correct Conclusion => (3.00300) (0.25025) => (1.0), 14(0.352), 17(0.829), ...
18 Incorrect Conclusion => (0.00000) (0.00000) => There are no items
19 Positive Examples => (0.00000) (0.00000) => There are no items
20 Negative Examples => (0.00000) (0.00000) => There are no items
21
22 Rule3: IF Vble_3 is LABEL_1 THEN Vble_4 is LABEL_0
23 Rule Support => (737.32151) (0.67830) => 0(0.561), 1(0.407), 2(0.872), ...
24 Antecedent Support => (747.00990) (0.68722) => 0(0.561), 1(0.407), 2(0.872), ...
25 Consequent Support => (998.73089) (0.91879) => 0(0.561), 1(0.407), 2(0.872), ...

```

Fig. 2. Structure of .fs files

that are considered when computing the *goodness measure* [9] in classification and regression problems. They are not taken into account in case of fuzzy association rules.

Next line yields the total number of rules. Then, information related to each rule is detailed. For illustrative purposes, we have included three examples of rules (each one corresponding to one of the three types of rules that are considered). Thus, Rule1, Rule2, and Rule3 exemplify rules of classification, regression and association respectively. In all cases the first line gives the number of the rule along with its linguistic description.

In case of classification rules, each rule is accompanied by four lines. The first two lines give the enumeration of correct and incorrect covered examples, i.e. examples covered with the same/different output class. The next line lists the positive examples, i.e. correct examples that fires the rule above the high goodness threshold. The last line enumerates the negative examples, i.e. incorrect examples plus the correct examples that are fired below the low goodness threshold. In all the four cases, the values in parenthesis at the beginning of lines are the accumulated and average firing degrees of the enumerated examples, and the value after every example gives the level up to they fires the corresponding rule. For regression problems, each rule comes with almost the same information. Formally, the four lines are the same. The only difference turns out in the interpretation of the meaning of correct/incorrect and positive/negative examples. The interested reader can get additional information in [9].

Finally, in case of fuzzy association rules, each rule description is accompanied by three lines. They are related to the support of the entire rule, but also its antecedent or consequent parts alone. For each of them, we enumerate the list of supported examples along with the total accumulated support (first parenthesis) and the average support (second). Notice that support is computed by Eq. 1.

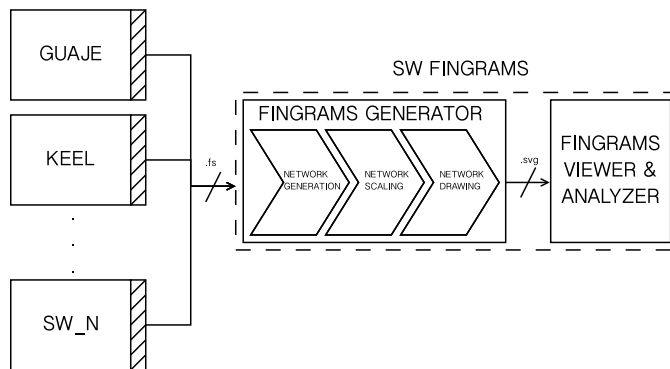


Fig. 3. Software scheme.

We developed the software using Java. We chose this programming language because it is characterized by high penetration in academic and professional tools, modular design and maintenance, and interoperability through operating systems. The creation, visualization and analysis of fingrams are done in four steps as sketched in Fig. 3 (following the four steps introduced in II-B). Some implementation details are given below:

- 1) **Network generation:** We have developed ad-hoc software for this task. It is in charge of creating the co-firing matrix that characterizes all the relations among rules.
- 2) **Network scaling:** We have included an implementation of Fast-Pathfinder [28], a variant of Pathfinder that reduces the complexity of the original algorithm, written in C. It is called through the java native interface.
- 3) **Network drawing:** We call to the open source visualization software Graphviz⁵ to generate the layout of the final fingram. Notice that, Neato option is chosen

⁵Graphviz is freely available: [<http://www.graphviz.org>]

to generate a layout based on the Kamada-Kawai [29] spring embedded algorithm. The result is stored in a .svg file.

- 4) **Network visualization:** We developed a SVG viewer (using Batik library for Java⁶) aimed at making user-friendly the visualization of the generated figrams. It lets user move, zoom in/out and rotate the displayed figram. Even more, an auxiliary tab panel provides rankings of rules according to popular social network measures like *Centrality of Degree*, *Page Rank*, etc.

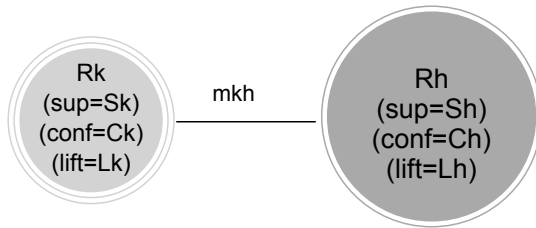


Fig. 4. Figram's interpretation.

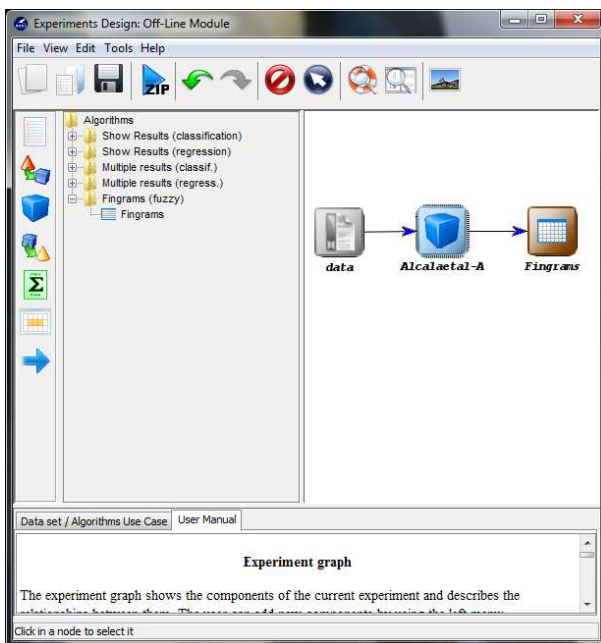


Fig. 5. KEEL's module for figram generation.

Figrams have already been successfully applied to real classification and regression problems [9]. With the new software tool presented in this paper we extend figram definition to fuzzy association rules.

It is worthy to note that figrams are customized for the particularities of the selected type of rules. In case of fuzzy association rules, figrams have been enhanced with specific information of interest such as the *Support*, *Confidence* and *Lift* of the represented rules. They are computed by equations (1) to (3). As can be seen in Fig. 4, those quality measures

(*sup=Support*, *conf=Confidence* and *Lift*) are included in the plotted nodes. Moreover, the node size is proportional to the value of *Support* while its color is in a grey scale (being close to white for low values of *Lift* and black in the opposite case). In addition, the number of border lines in a node is related to the number of premises in the rule represented by the node. The edge weight (*mkh*) indicates the rule co-firing between rules. The more inputs fire two rules, the higher the edge weight between them ($mkh \in [0, 1]$). The thickness of edges is also proportional to the edge weights. For a more detailed analysis and comprehension of figrams, the interested reader is referred to Section IV of [9] where figram analysis is deeply explained.

We have enhanced GUAJE and KEEL with their own modules able to generate .fs files from the designed fuzzy systems. Classification and regression systems constructed with GUAJE, and association fuzzy rules built with KEEL can be exported with the aim of generating and visualizing figrams with the software presented in this paper (see Fig. 5).

IV. A CASE STUDY ON ANALYZING FUZZY ASSOCIATION RULES WITH FIGRAMS

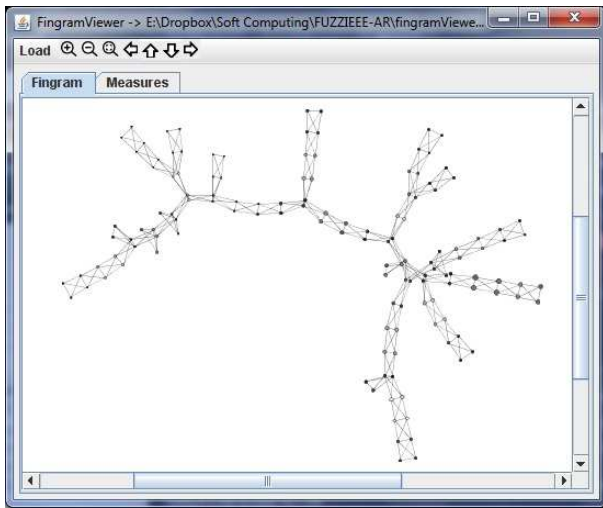
This section presents a case study that shows the main characteristics and benefits of using figrams for fuzzy association rules. We use the database *Stulong*⁷ that deals with a study (carried out from 1976 to 1999) of the risk factors of atherosclerosis in a population of middle-aged men.

KEEL software is used to create fuzzy association rules with the method presented in [30]. –Of course, other methods may be used. However, we would like to remark that we are not focused on determining the best fuzzy association rules for the database under study. The emphasis is on illustrating the benefits of considering figrams as a powerful support tool for visualizing and analyzing the generated rules.– We generate the rules using the default parameters, except for the number of generations (5000), number of linguistic terms (5), minimum support (35%) and minimum confidence (98%). With this configuration, we obtain 134 rules.

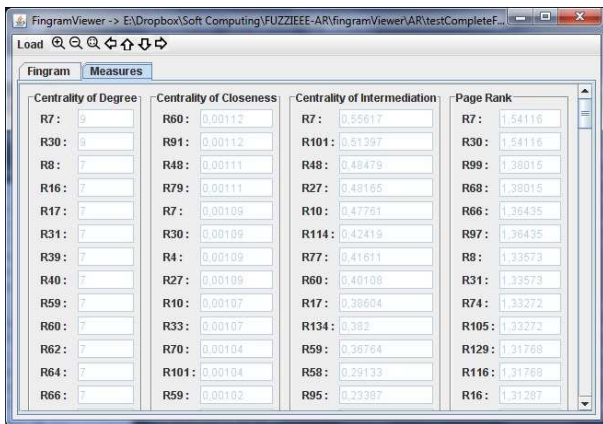
Once obtained this set of rules, we used the new module of KEEL that permits the generation of the .fs file (including the relevant information of the rules required for creating figrams) that is taken as input of the new software tool presented in this contribution. Then, we used such tool to generate the associated figram (Fig. 6). This figram permits the user to observe the overall structure of the given rule set. He/she can zoom out the zones of interest. It is also possible to filter the figram according to the quality measures. For instance, we selected the twenty rules with the highest *Lift* values (all above 1.12) for illustrative purpose. Regarding only those rules the original figram in Fig. 6 is translated into a simplified figram (Fig. 7). Notice that we have zoomed out some of the nodes in the figram to observe them in detail.

⁷The data resource is on [http://euromise.vse.cz/challenge2004]. At present, the data analysis is supported by the grant of the Ministry of Education CR Nr LN 00B 107.

⁶Batik is freely available: [http://xmlgraphics.apache.org/batik/]



(a) Fingram view.



(b) Measures view.

Fig. 6. Screenshots of the FingramViewer tool: The entire fingram is displayed on the left while several rule rankings are provided on the right.

This representation allows the user to see rules more related, meaning that they cover similar examples. Moreover, the bigger the nodes the higher the support of the rules, and the darker the node the higher the *Lift* of the rules. As an example of the subset of rules of Fig. 7, Rule 50 has higher rule *Support* than Rule 112, while Rule 114 has higher *Lift* than Rule 66.

Finally, it is interesting to highlight that *Support* (size of nodes) and *Lift* (grey tone of nodes) are somehow contradictory for the selected subset of rules under study. General rules with high *Support* usually imply low *Lift*, i.e. big light nodes as Rules 48, 50 or 66; whereas specific rules with low *Support* produces high *Lift*, i.e. small dark nodes as Rules 114, 119 or 121. Optimum rules should have high *Support* and *Lift* (they should be easy to identify by looking for big dark nodes in the fingram). On the other hand, rules good candidates to be further studied in detail, for improving them, are those with low *Support* and *Lift* (small light nodes).

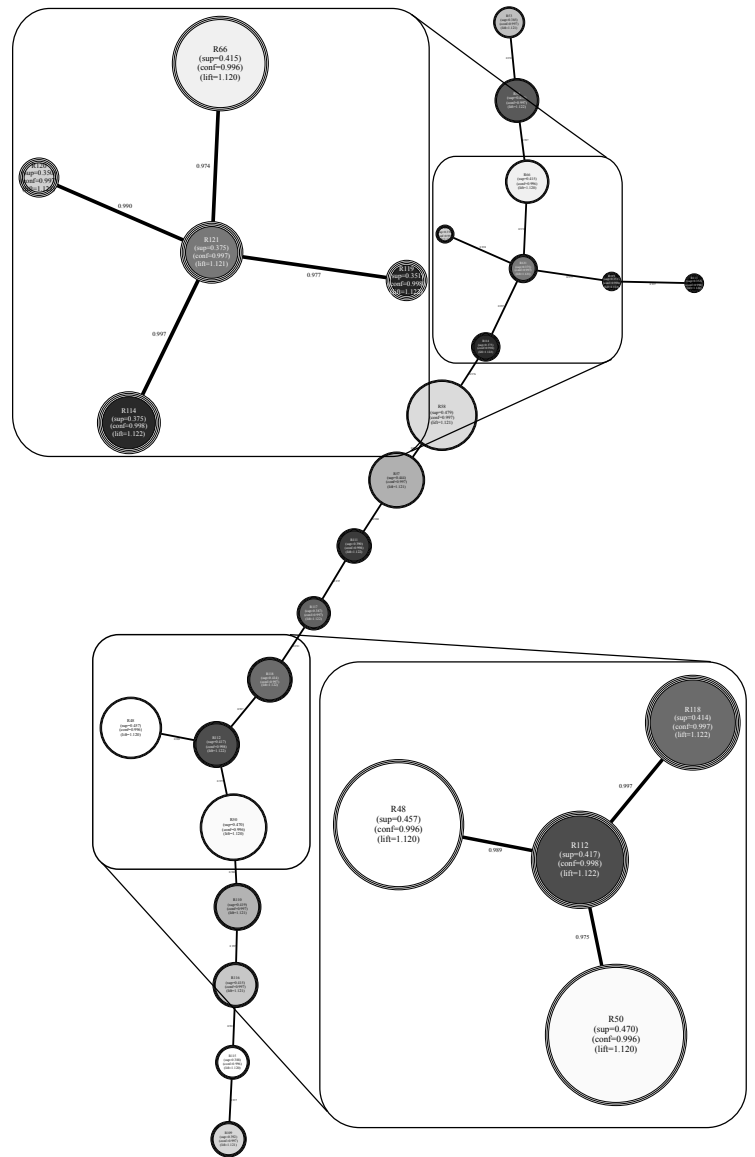


Fig. 7. Thresholded fingram with *Lift* > 1.12.

V. CONCLUSIONS AND FUTURE WORKS

A stand-alone open source software tool has been developed for creating fingrams. This software produces fingrams, thus allowing the analysis of fuzzy systems designed with GUAJE (classification or regression rules), KEEL (fuzzy association rules), or any other design software tool.

We adapt fingrams to deal with fuzzy association rules, including information of particular interest for this kind of rules. We demonstrate that fingrams can be extended and used in other kinds of fuzzy systems without much effort. It is part of our future work to enhance fingrams to work with more kind of fuzzy rules, being useful in many other fields.

A case study related to fuzzy association rules shows that fingrams are a valuable representation for this kind of rules. We sketch some of the potentials of fingram analysis in the design and optimization of fuzzy association rules.

We plan to develop software modules in other tools for fingrams generation, broaden the methodology. This would propitiate a high feedback and improvements from several profiles of users.

ACKNOWLEDGMENT

This work has been funded by the Spanish Ministry of Economy and Competitiveness under Grants TIN2011-29824-C02-01 and TIN2011-29824-C02-02 (ABSYNTHÉ project); the Spanish Ministry of Education and Science under Grant TIN2011-28488; the Andalusian Government under Grant P10-TIC-6858; and the European Centre for Soft Computing.

REFERENCES

- [1] M. Delgado, N. Marín, D. Sánchez, and M. Vila, "Fuzzy association rules: General model and applications," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 2, pp. 214–225, 2003.
- [2] K. Kianmehr, M. Kaya, A. M. ElSheikh, J. Jida, and R. Alhaji, "Fuzzy association rule mining framework and its application to effective fuzzy associative classification," *WIREs - Data Mining and Knowledge Discovery*, vol. 1, no. 6, pp. 477–495, 2011.
- [3] M. Delgado, D. Sánchez, M. J. Martín-Bautista, and M.-A. Vila, "Mining association rules with improved semantics in medical databases," *Artificial Intelligence in Medicine*, vol. 21, pp. 241–245, 2001.
- [4] J. Alonso and L. Magdalena, "Editorial: Special issue on interpretable fuzzy systems," *Information Sciences*, vol. 181, no. 20, pp. 4331–4339, 2011.
- [5] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Parts I, II, and III. Information Sciences*, vol. 8, 8, 9, pp. 199–249, 301–357, 43–80, 1975.
- [6] —, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, pp. 28–44, 1973.
- [7] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic systems," *IEEE Transactions on Computers*, vol. 26(12), pp. 1182–1191, 1977.
- [8] E. Trillas and S. Guadarrama, "Fuzzy representations need a careful design," *International Journal of General Systems*, vol. 39, no. 3, pp. 329–346, 2010.
- [9] D. P. Pancho, J. M. Alonso, O. Córdón, A. Quirin, and L. Magdalena, "FINGRAMS: visual representations of fuzzy rule-based inference for expert analysis of comprehensibility," *IEEE Trans. on Fuzzy Systems*, doi: 10.1109/TFUZZ.2013.2245130, 2013.
- [10] J. M. Alonso and L. Magdalena, "Generating understandable and accurate fuzzy rule-based systems in a java environment," in *Lecture Notes in Artificial Intelligence - 9th International Workshop on Fuzzy Logic and Applications*, vol. LNAI6857. Springer-Verlag, 2011, pp. 212–219.
- [11] J. M. Alonso, D. P. Pancho, and L. Magdalena, "Enhancing the fuzzy modeling tool GUAJE with a new module for fingrams-based analysis of fuzzy rule bases," in *IEEE International Conference on Fuzzy Systems*, 2012, pp. 1082–1089.
- [12] J. Alcalá-Fdez et al., "KEEL: a software tool to assess evolutionary algorithms for data mining problems," *Soft Computing*, vol. 13(3), pp. 307–318, 2009.
- [13] —, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Mult.-Valued Logic & Soft Computing*, vol. 17, pp. 255–287, 2011.
- [14] C. Zhang and S. Zhang, "Association rule mining: Models and algorithms," in *Lecture Notes in Computer Science, LNAI 2307*, 2002.
- [15] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *SIGMOD*, Washington D.C., 1993, pp. 207–216.
- [16] R. Agrawal and R. Srikant, "Fast algorithms forming association rules," in *International Conference Large Data Bases*, Santiago de Chile, Chile, 1994, pp. 487–499.
- [17] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*. Berlin: Springer-Verlag, 2004.
- [18] T. Sudkamp, "Examples, counterexamples, and measuring fuzzy associations," *Fuzzy Sets and Systems*, vol. 149, no. 1, pp. 57–71, 2005.
- [19] M. Kaya and R. Alhaji, "Utilizing genetic algorithms to optimize membership functions for fuzzy weighted association rules mining," *Applied Intelligence*, vol. 24, no. 1, pp. 7–15, 2006.
- [20] J. Alcalá-Fdez, R. Alcalá, and F. Herrera, "A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 857–872, 2011.
- [21] M. Wu, T. Hong, and C. Lee, "A continuous ant colony system framework for fuzzy data mining," *Soft Computing*, vol. 16, no. 12, pp. 2071–2082, 2012.
- [22] F. Berzal, I. Blanco, D. Sanchez, and M. Vila, "Measuring the accuracy and interest of association rules: A new framework," *Intelligent Data Analysis*, vol. 6, no. 3, pp. 221–235, 2002.
- [23] S. Brin, R. Motwani, J. Ullman, and S. Tsur, "Dynamic itemset counting and implication rules for market basket data," *ACM SIGMOD Record*, vol. 26, no. 2, pp. 255–264, 1997.
- [24] L. Geng and H. Hamilton, "Interestingness measures for data mining: A survey," *ACM Computing Surveys*, vol. 38, no. 3, pp. 1–32, 2006.
- [25] S. Ramaswamy, S. Mahajan, and A. Silberschatz, "On the discovery of interesting patterns in association rules," in *24rd International Conference on Very Large Data Bases*, San Francisco, CA, USA, 1998, pp. 368–379.
- [26] J. M. Alonso, D. P. Pancho, O. Córdón, A. Quirin, and L. Magdalena, "Social network analysis of co-fired fuzzy rules," in *Soft Computing: State of the Art Theory and Novel Applications*, R. R. Yager, A. M. Abbasov, M. Reformat, and S. N. Shahbazova, Eds. Springer, 2013, pp. 113–128.
- [27] D. P. Pancho, J. M. Alonso, and L. Magdalena, "Quest for interpretability-accuracy trade-off supported by fingrams into the fuzzy modeling tool guaje," *International Journal of Computational Intelligence Systems*. In press, 2013.
- [28] A. Quirin, O. Córdón, J. Santamaría, B. Vargas-Quesada, and F. Moya-Anegón, "A new variant of the pathfinder algorithm to generate large visual science maps in cubic time," *Information Processing and Management*, vol. 44, no. 4, pp. 1611–1623, 2008.
- [29] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information Processing Letters*, vol. 31, pp. 7–15, 1989.
- [30] J. Alcalá-Fdez, R. Alcalá, M. Gacto, and F. Herrera, "Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms," *Fuzzy Sets and Systems*, vol. 160, no. 7, pp. 905–921, 2009.