Our fruitful relationship with Sugeno inference, from FUMOSO to pyFUME

Leone Bacciu[1][0009-0006-2175-8779], Paolo Cazzaniga[2,3][0000-0001-7780-0434], Chiara Gallese[4][0000-0001-8194-0261], Caro Fuchs[5][0000-0002-4815-0310], Uzay Kaymak[5][0000-0002-4500-9098], Daniele M. Papetti[6][0000-0002-3574-6027], and Marco S. Nobile[1,3][0000-0002-7692-7203]

[1] Ca' Foscari University of Venice, Italy marco.nobile@unive.it
[2] University of Bergamo, Italy
[3] Bicocca Bioinformatics, Biostatistics and Bioimaging Research Center (B4), Italy
[4] University of Turin, Italy
[5] Eindhoven University of Technology, The Netherlands
[6] University of Milano-Bicocca, Italy

Abstract. Fuzzy Inference Systems (FIS) effectively model and reason with complex and uncertain information in an interpretable, understandable, and transparent way. Takagi-Sugeno-Kang (TSK) is one of the most widespread types of FIS, appreciated for its ability to output crisp values by leveraging linear models as consequents. In this tribute to Prof. Sugeno, we discuss our previous works based on TSK inference. In particular, we focus on pyFUME, a Python library for automatically estimating first-order TSK FIS from data. We introduce a relevant advancement in pyFUME, i.e., the ability to handle categorical variables within the consequents, which significantly enhances the model's performance in regression and classification tasks. This improved version completes the foundation for pyFUME to handle mixed-type data. Our results on three diverse datasets show the capabilities of our method, which performs better than the previous implementation.

Keywords: Sugeno inference · Fuzzy modeling · Mixed datasets.

1 Introduction

Professor Michio Sugeno is the primary creator of the Takagi-Sugeno-Kang (TSK or Sugeno) method [24,25], and his work has revolutionized the use of fuzzy logic in decision-making processes. The Sugeno method differs from other fuzzy inference systems as it provides a well-defined framework by expressing rules in the form of mathematical functions. This characteristic makes it particularly suitable for various applications, including control systems, pattern recognition, and decision support. Sugeno's pioneering efforts have significantly advanced the practical implementation of fuzzy logic by offering a versatile and effective tool for handling uncertainty and non-linearity in real-world applications. Starting from Sugeno's work, we developed several tools and methods described hereafter.

Dynamic fuzzy modeling is a knowledge-driven approach based on fuzzy logic that allows for analyzing heterogeneous complex systems and carrying out dynamical simulations of the temporal evolution of the system without requiring any precise quantitative parameterization [15]. In a Dynamic Fuzzy Model (DFM), linguistic variables and terms are associated with system components, providing a qualitative description of potential states over time. Fuzzy rules, derived from these linguistic elements, govern the behavior of DFMs, distinguishing between *inner* and *outer* variables: the former can either appear as antecedent or as consequent in fuzzy rules, while the latter represent *input* and *output* variables. Specifically, input variables influence system dynamics, while output variables represent observable components. The synchronous application of fuzzy rules, employing the zero-order Sugeno method [24], drives the temporal evolution of non-input variables. Simulation and analysis can be conducted through FUMOSO (FUzzy MOdels SimulatOr) [15,23], an open-source software allowing user-specified settings for simulation time, variable states, and input functions.

An alternative knowledge-driven approach is Fuzzy-mechanistic modeling of compleX systems (FuzzX), which is a general-purpose methodology for hybrid modeling that couples the quantitative description and analysis of well-known and detailed processes, along with other phenomena whose functioning is not well characterized and can only be described using linguistic concepts [22]. Fuzzy rules in FuzzX control variables or parameters of the mechanistic module, facilitating the integration of precise and imprecise information to model emergent behaviors accurately. In particular, FuzzX can overcome different limitations of other hybrid modeling approaches, as it allows for dynamically integrating a detailed quantitative model with a qualitative fuzzy rule base system by synchronizing the qualitative and quantitative modules during the simulation thanks to the variables, and parameters, working as an interface between the two modules. A simulation step of a model defined with FuzzX is realized by calculating the dynamics of the mechanistic module and followed by a fuzzy inference.

Finally, pyFUME [6] is a Python library for automatically estimating Fuzzy Inference Systems (FISs) from data. pyFUME simplifies FIS creation from data, offering flexibility for customization. Indeed, it supports various data processing steps, including loading data from *.csv*-files, splitting it into training and testing datasets, imputing missing values, and performing feature selection. Specifically, feature selection can be performed using several approaches, such as sequential forward selection or a more advanced approach based on a Fuzzy Self-Tuning PSO (FST-PSO) [14]. The latter approach has the additional advantage of optimizing the number of rules of the FIS. In addition, pyFUME also enables data clustering using various methods, including a swarm intelligence method based on FST-PSO, designed to reduce the risk of local minima entrapment [4]. The antecedent sets and consequent parameters of a Sugeno FIS can be estimated, and the model can be simplified using the Graph-Based Simplification approach [5]. Finally, pyFUME can compute several evaluation metrics (e.g., the Mean Absolute Percentage Error or MAPE), providing a comprehensive FIS development and evaluation toolkit.

In the context of Explainable Artificial Intelligence (XAI), the TSK method's ability to provide rule-based explanations for its decisions is essential for developing transparent and understandable AI systems. The European AI Act, which will soon be enforced in all Member States, specifies that users should be able to "correctly interpret the high-risk AI system's output." [8] This means that the system should offer sufficient information for users to understand why a particular output was produced.

The newly introduced article 86, called "Right to explanation of individual decision-making", specifically says in paragraph 1 that "*Any affected person subject to a decision which is taken by the deployer on the basis of the output from a high-risk AI system [...], and which produces legal effects or similarly significantly affects that person in a way that they consider to have an adverse impact on their health, safety or fundamental rights shall have the right to obtain from the deployer clear and meaningful explanations of the role of the AI system in the decision-making procedure and the main elements of the decision taken.*" In crucial applications, such as medical devices, interpretable models are the only way to ensure the respect of citizens' fundamental rights [8,13]. As part of the AI Act's mandate of debiasing tests (Art. 10, data governance), TSK represented the foundation for the development of FAnFAIR, a methodology for the (semi)automatic assessment of potential biases in datasets affecting the fairness of a derived AI system [9]. In this paper, we present an enhancement of pyFUME, namely the integration of categorical variables in consequents, which makes pyFUME more robust and versatile for addressing the inherent complexity of mixed-type data sets. To assess the effectiveness of this new version, we performed tests on three distinct datasets. The results show improved performance compared to the previous version of pyFUME, confirming its potential as an efficient and versatile tool for data-driven (interpretable) modeling.

2 Fuzzy modeling and the pyFUME project

Fuzzy Sets (FS) provide a mathematical framework to represent the vagueness and uncertainty typical of real world concepts and variables [28]. In FS theory, elements of the universe of discourse can simultaneously belong to more than one set with different membership degrees, determined by the Membership Functions (MF). A MF $\mu : U \rightarrow [0,1]$ maps each element of the universe of discourse U to its corresponding membership value; the membership degree ranges from 0 (the element does not belong to the set) to 1 (the element fully belongs to that set). Fuzzy Inference Systems (FIS) built on top of FS provide a formal framework to perform reasoning and take decisions by using a collection of if-then rules. Fuzzy rules can be split into two parts:

1. *antecedents*, i.e., the part of the rule that describes the conditions under which the rule applies;
2. *consequents*, i.e., the part of the rule that determines the system's output.

Since the antecedents are expressed through fuzzy logic, each rule is associated with a degree of satisfaction in $[0,1]$. Conversely, the consequents can assume various forms and their impact on the overall output depends on the satisfaction degree of the corresponding antecedent. FIS can be characterised according to their consequents: when the output of rules involve fuzzy sets, it is called Mamdani inference; when the output are functions, then it is called Takagi-Sugeno-Kang (TSK) inference.

In order to simplify the development of FIS, multiple libraries have been proposed for several languages, including MATLAB and R [26]. For Python, two possible options are scikit-fuzzy and Simpful. The latter is a library designed to provide a simple and lightweight API for the development of FIS based on either Mamdani or Sugeno inference [20]. Simpful supports many features, can parse complex fuzzy rules involving AND, OR, and NOT operators, and exploit arbitrarily shaped FS. On top of Simpful, we developed pyFUME, a Python library for the estimation of FIS based on (first-order) TSK inference [6]. The functioning of pyFUME can be summarized as follows:

1. *Data loading and splitting*, these steps include optional normalization options, data imputation, and feature selection.
2. *Clustering*, this step groups data in J clusters by means of an available clustering algorithm (fuzzy c-means [2] (possibly paired with swarm intelligence [4,21], Gustafson-Kessel clustering [10] or, for mixed datasets, fuzzy c-prototypes [16]).
3. *Antecedent estimation*, this step generates the fuzzy sets and antecedents of the rules based on the clustering. Every rule describes one cluster, and is formed as follows:

$$\begin{aligned}\mathbf{R}_j: \quad & \textbf{IF } (x_1 \text{ is } A_{j1}) \text{ and } \ldots \text{ and } (x_N \text{ is } A_{jN}) \\ & \textbf{THEN } (\text{OUTPUT is } y_j)\end{aligned}$$

   where, $j = 1, \ldots, J$ denotes the cluster and rule number, $\mathbf{x} = (x_1, \ldots, x_N)$ denotes the input vector, N is the number of input features (linguistic variables), A_{ji} is the fuzzy set for the i^{th} linguistic variable with respect to cluster j, and y_j is the consequent function for rule R_j. The firing strengths of the rules are also computed in this step for the training set. The firing strength for rule R_j is: $\beta_j = \min(\mu_{A_{j1}}(\mathbf{x}), \ldots, \mu_{A_{jN}}(\mathbf{x}))$, for $j = 1, \ldots, J$, where $\mu_{A_{ji}}$ is the membership function for the fuzzy set of feature i and cluster j. The estimation of the antecedent sets is performed by fitting the convex envelope of such membership values, as described in [7].
4. *Consequent estimation*, this step generates the consequents, which can be zero-order ($y_j = c_j$) or first-order ($y_j = \mathbf{a}_j^\intercal \mathbf{x} + b_j$) linear models [1]. The overall output y^* is computed as follows:

$$y^* = \frac{\sum_{j=1}^{J} \beta_j y_j}{\sum_{j=1}^{J} \beta_j}.$$

   The linear models are fit by considering the firing strengths as per [1].
5. *Simpful model generation*: the final step is the generation of the Simpful file that contains the model.

3 Handling categorical consequents in pyFUME

Regardless of the popularity of TSK, a few approaches using mixed datasets can be found in the literature. One example is Liu *et al.* [12], where the authors use a Firing-strength Transform Matrix to handle the categorical inputs. In [18], an ensemble-based multistage scheme is applied: first, a logistic regression model is used to transform the binary feature space into a numerical feature, then the logistic regression output, together with the continuous variables, is used to train a second stage of models consisting of an ensemble of two TIK models. In this work, we propose a different strategy based on one-hot encoding.

To retain Simpful's high level of flexibility and readability, we implemented a novel refined syntax for crafting customized replacement rules within Simpful's output functions. This necessity arises due to the distinctive characteristic of categorical variables, wherein the direct substitution of linguistic terms in the consequent with their corresponding numerical values, as commonly executed with continuous variables, is not feasible. Instead, a methodological construct becomes necessary to substitute categorical linguistic terms with numerical representations, mainly when the linguistic term falls within a specific category. These new characteristics required a dedicated syntax, that is based on a novel replacement pattern in the consequent that is structured as follows:

{ **IF** `variable` **IS** `category` **THEN** `value` }

This part of the rule is replaced at runtime with `value` if the `variable` corresponds to the specific `category`; otherwise, it is replaced by a 0.Thanks to this syntax, we could implement one hot encoding – often employed in machine learning to bridge the gap between qualitative and quantitative data – in the consequents. The approach consists of adding a dummy feature for each possible value of the categorical variable: the i^{th} new dummy feature will have a value 1 if the data point of that row belongs to the i^{th} possible category, and 0 otherwise. Thanks to this approach, pyFUME can process categorical variables when creating first-order Sugeno models. In particular, it is now possible to pass the `categorical_indices` argument to the pyFUME methods, which intuitively contains the list of indices/columns of categorical variables to treat them accordingly. The library then takes care of creating the parameters matrix, which contains, in the i^{th} row, the parameters of the i^{th} linear model (output function), and the Simpful model, which exploits the new syntax.
Here is an example of the procedure to generate the one-hot encoding matrix, the first matrix is the design matrix where both of the variables are categorical:

$$\left[\begin{array}{c|c} var_1 & var_2 \\ 0 & 0 \\ 2 & 0 \\ 1 & 1 \\ 2 & 0 \end{array}\right] \xRightarrow[\text{dummy variables}]{\text{Replace with}} \left[\begin{array}{ccc|cc} 1&0&0&1&0\\ 0&0&1&1&0\\ 0&1&0&0&1\\ 0&0&1&1&0 \end{array}\right] \xRightarrow[\text{columns}]{\text{Remove redundant}} \left[\begin{array}{cc|c} 1&0&1\\ 0&0&1\\ 0&1&0\\ 0&0&1 \end{array}\right]$$

Starting from the one-hot encoding matrix, the consequent takes the form:
{ `IF` var_1 `is 0 THEN` o_1 } + { `IF` var_1 `is 1 THEN` o_2 } + { `IF` var_2 `is 0 THEN` o_4 }, where o_j is the coefficient computed by pyFUME.

4 Results

We tested the improved version of pyFUME described in the previous Section (called *Mixed*, in what follows, which makes use of Simpful version 2.12.0 and pyFUME version 0.3.4) against the previous version (called *Continuous*, in what follows, which makes use of Simpful version 2.11.1 and pyFUME 0.2.25) and a simple baseline alternative approach (called *Baseline*). The comparison is carried out by measuring the performance of the models obtained using three different publicly available datasets: a 10-fold cross-validation with a Wilcoxon test to assess statistically significant differences computed on either the MAPE or the F1-score. These datasets pertain to data collected for different contexts, i.e., healthcare, transportation, and insurance companies.

The first dataset is the *Medical insurance charge prediction*[7] (1338 instances), which contains a mixed set of attributes about patients (2 continuous and 4 categorical); the regression task regards the prediction of the amount of the medical insurance bill after a visit to the hospital. The second well-known dataset is the *Auto-MPG*[8] (392 instances), in which the explanatory variables are the characteristics of car models and their engines (4 continuous and 3 categorical); the regression task regards the miles-per-gallon of the cars. The third dataset is the *Pediatric appendicitis*[9] (780 instances), which contains a list of pediatric patients admitted to the hospital with abdominal pain. It is a classification task with the objective of predicting whether the patient has appendicitis or not, using fourteen features encompassing laboratory tests, physical examinations, and clinical scores (6 continuous and 8 categorical).

Table 1. Results summary

	# clusters	Mixed	Continuous	Baseline	Mixed	Continuous	Baseline
			MAPE			R2	
Insurance	4	0.321	0.705	0.426	0.797	0.564	0.739
Auto	2	0.086	0.125	0.100	0.867	0.735	0.847
			F1			Accuracy	
Appendicitis	2	0.856	0.769	0.850	0.821	0.727	0.820

For the first two datasets, we compute the MAPE and R2 scores to measure the performance of three tested approaches in tackling the regression task. For the third dataset, the performance related to the binary classification task is assessed by computing the F1-score and accuracy. It is worth noting that pyFUME exploits a linear model for the classification task, applying a threshold of 0.5 to the output. The baseline models used in the comparison are a Linear Regression model for the regression tasks and a Logistic Regression model for binary classification, from Scikit-learn [17] (version 1.3.2) with the default parameters. Both models make use of categorical (nominal) features via one-hot encoding. The number of

[7] https://www.kaggle.com/datasets/niranjanank/medical-insurance-charge-prediction
[8] https://www.kaggle.com/datasets/uciml/autompg-dataset
[9] https://www.kaggle.com/datasets/joebeachcapital/regensburg-pediatric-appendicitis

clusters leveraged by pyFUME was chosen based on Xie-Beni validity criterion [27]. Although normalization often improves model performance [3], we experimentally verified that, with the datasets considered in this paper, the difference was not substantial (data not shown).

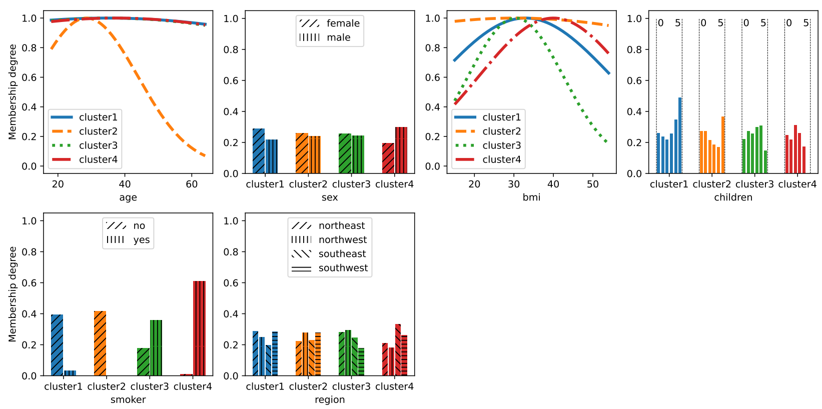

Fig. 1. Plot of the antecedents for the Insurance dataset.

All fuzzy set plots in this paper were generated using Simpful's functionalities [6,16]. Figures 1, 3, 5 represent the antecedents and show the fuzzy sets for each linguistic variable. The categorical variables are represented as bar plots, where each colored group refers to a specific cluster, and each bar hatch represents a different category, while the height of the bar represents the corresponding MF [16]. The bar plots in Figures 2, 4, 6 denote the variables' influence on the prediction. They are based on the parameters of the linear regression models of the consequents, normalized as $b'_i = b_i \frac{\max \mathbf{X}_i - \min \mathbf{X}_i}{\max \mathbf{y} - \min \mathbf{y}}$, where b_i is the parameter in the model, $\mathbf{X}_i$ is the i^{th} column in the training dataset, and $\mathbf{y}$ is the target variable vector.

Insurance. Table 1 reports the performance of the three tested approaches in terms of MAPE and R2 score, showing that the *Mixed* approach outperformed the other models (both p-values < 0.01). Regarding the four clusters reported in Figure 1, we observe that some have very specific characteristics that identify well-defined sub-populations. For instance, Cluster 4 identifies the sub-population of obese and severely obese people, most of whom are smokers. Considering the consequents (Figure 2), we note that 'BMI' (body mass index) is the most important variable having the highest impact on the amount of the bill and that Age is also positively correlated. These results reflect our expectations for this group, for excessive weight is likely to yield medical issues, thus higher medical bills. This is not necessarily the case for other groups, e.g., Cluster 1 and 3, where 'smoking' is the most influential variable. Finally, Cluster 2 is composed of non-smokers around 30: here 'age' is the most relevant variable.

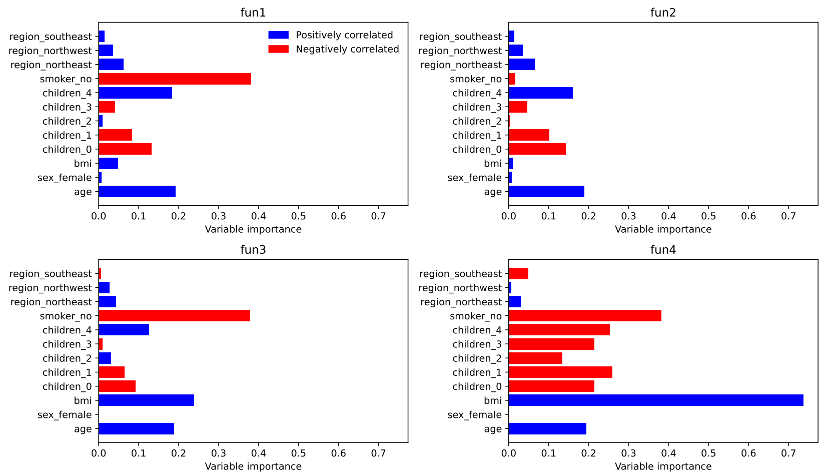

Fig. 2. Plot of the consequents for the Insurance dataset.

Auto. Also for this dataset, the *Mixed* approach outperforms the others (both p-values < 0.01) as reported in Table 1. The two clusters highlighted in Figure 3 are remarkably distinct; Cluster 1 contains more powerful (higher cylinder count, higher horsepower) but heavier cars, which have a slightly lower acceleration, while Cluster 2 contains less powerful and lighter cars. Another important distinction is the country of origin: the powerful cars are mainly from the US, while the others are mainly European and Asian cars. Focusing on the consequents in Figure 4, we observe that the most important variables are the year-related categories; indeed, newer vehicles are *positively correlated* with MPG, while older vehicles are *negatively correlated* with MPG. The intercept of the linear regression for Cluster 1—which is the mean output response of the rule when all the features are equal to 0—is 28 MPG, relatively low, but expected in the case of very powerful cars characterized by inefficient engines. For Cluster 2 the bias is 60 MPG, which indicates a very efficient vehicle.

Pediatric appendicitis. For this dataset, the performance of the *Mixed* approach is better than the *Continuous* (p-value < 0.01) and similar to the *Baseline* (p-value > 0.7) (see Table 1). Figure 5 shows that Cluster 1 includes people with an overall poor health condition: high temperature, nausea, loss of appetite, and possible ongoing infection (denoted by high WBC count and CRP level). The consequents (Figure 6) are very similar; the difference regards the intercept of linear regression. For Cluster 1, the intercept is 0.59, suggesting that the patients who strongly activate this rule will tend to have appendicitis; for Cluster 2, it is 0.25, suggesting the opposite.

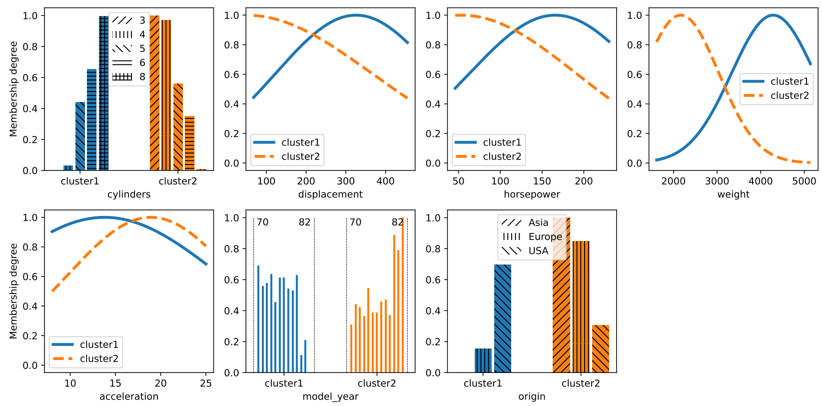

Fig. 3. Plot of the antecedents for the Auto dataset.

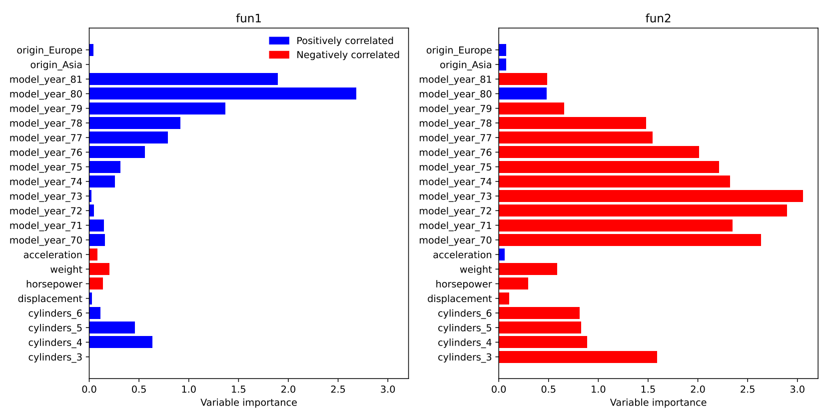

Fig. 4. Plot of the consequents for the Auto dataset.

5 Conclusion

In this tribute to Prof. Michio Sugeno and his work on fuzzy inference systems, we summarized the authors' past and ongoing works that were built on top of Sugeno's work, especially with the pyFUME library, which automatically estimates TSK FIS from data. The use of TSK FIS, instead of more frequently used machine learning approaches (such as neural networks), allows for a higher degree of interpretability of the model, a key characteristic for applications that operate in critical domains. Indeed, as demanded by the AI Act, it is fundamental to use systems able to offer sufficient information on the output provided in high-risk domains. In this context, we extended pyFUME to effectively leverage categorical variables not only during the clustering step and construction of FS but also in the consequents, paving the way for the applicability of pyFUME to

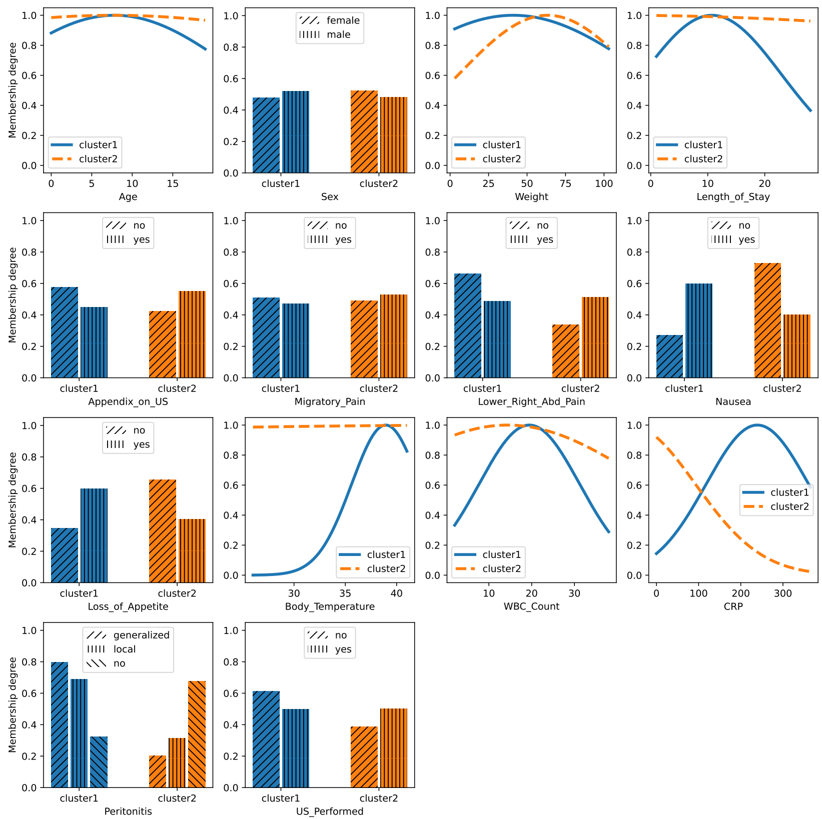

Fig. 5. Plot of the antecedents for the Appendicitis dataset.

any kind of datasets.The novel version of pyFUME was tested on three different mixed datasets and compared against the continuous version of pyFUME and baseline models. The results show that the mixed version of pyFUME outperforms the continuous version on all three datasets, while it outperforms the baseline model on both regression tasks and competes on the classification task. In addition, we showed how, thanks to the visualization of the fuzzy sets and the graphical representation of the linear regression coefficients, it is possible to grasp information regarding the model decision process, a key step to instill trust in models leveraged in high-risk scenarios.

In the future, we aim to apply pyFUME to large and high-risk datasets in collaboration with domain experts in order to further improve the interpretability for specific applications. Moreover, we will develop additional features to reduce the model's complexity, including the removal of features or fuzzy sets that do not bring useful information (using approaches similar to [11] and [19]). So doing, pyFUME would enable non-AI expert practitioners to estimate FIS automatically from data in Python, providing a friendly and extensible library that can be

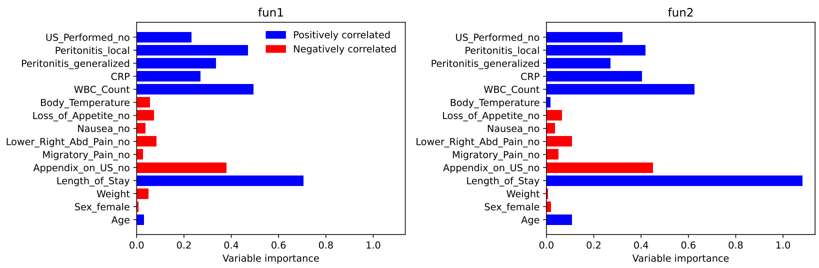

Fig. 6. Plot of the consequents for the Appendicitis dataset.

used out-of-the-box. In all new versions of pyFUME, categorical consequents are available and automatically enabled for all the categorical features specified.

References

1. Babuška, R.: Fuzzy modeling for control, vol. 12. Springer Science & Business Media (2012)
2. Bezdek, J.C.: Models for pattern recognition. In: Pattern Recognition with Fuzzy Objective Function Algorithms, pp. 1–13. Springer (1981)
3. Fuchs, C., Spolaor, S., Kaymak, U., Nobile, M.S.: The impact of variable selection and transformation on the interpretability and accuracy of fuzzy models. In: 2022 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). pp. 1–8 (2022)
4. Fuchs, C., Spolaor, S., Nobile, M.S., Kaymak, U.: A swarm intelligence approach to avoid local optima in fuzzy c-means clustering. In: 2019 IEEE Int Conf on fuzzy systems (FUZZ-IEEE). pp. 1–6. IEEE (2019)
5. Fuchs, C., Spolaor, S., Nobile, M.S., Kaymak, U.: A graph theory approach to fuzzy rule base simplification. In: Int Conf on Information Processing and Management of Uncertainty in Knowledge-Based Systems. pp. 387–401. Springer (2020)
6. Fuchs, C., Spolaor, S., Nobile, M.S., Kaymak, U.: pyFUME: a Python package for fuzzy model estimation. In: 2020 IEEE Conference on Fuzzy Systems (FUZZ-IEEE). pp. 1–8. IEEE (2020)
7. Fuchs, C., Wilbik, A., Kaymak, U.: Towards more specific estimation of membership functions for data-driven fuzzy inference systems. In: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). pp. 1–8. IEEE (2018)
8. Gallese, C.: The AI Act proposal: a new right to technical interpretability? In: Proceeding of the ISLC Conference 2023. Milano University Press (2024, in press)
9. Gallese, C., Scantamburlo, T., Manzoni, L., Nobile, M.S.: Investigating semi-automatic assessment of data sets fairness by means of fuzzy logic. In: 2023 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). pp. 1–10. IEEE (2023)
10. Gustafson, D.E., Kessel, W.C.: Fuzzy clustering with a fuzzy covariance matrix. In: Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on. pp. 761–766. IEEE (1979)
11. Kaymak, U., Babuška, R.: Compatible cluster merging for fuzzy modelling. In: 1995 IEEE Conference on Fuzzy Systems. vol. 2, pp. 897–904. IEEE (1995)

12. Liu, M., Dong, M., Wu, C.: A new ANFIS for parameter prediction with numeric and categorical inputs. IEEE Transactions on Automation Science and Engineering **7**(3), 645–653 (2010)
13. Nobile, C.: Legal aspects of AI models in medicine. the role of interpretable models. In: Big Data Analysis and Artificial Intelligence for Medical Science. Wiley (2024)
14. Nobile, M.S., Cazzaniga, P., Besozzi, D., Colombo, R., Mauri, G., Pasi, G.: Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization. Swarm and evolutionary computation **39**, 70–85 (2018)
15. Nobile, M.S., Votta, G., Palorini, R., Spolaor, S., De Vitto, H., Cazzaniga, P., Ricciardiello, F., Mauri, G., Alberghina, L., Chiaradonna, F., et al.: Fuzzy modeling and global optimization to predict novel therapeutic targets in cancer cells. Bioinformatics **36**(7), 2181–2188 (2020)
16. Papetti, D.M., Fuchs, C., Coelho, V., Kaymak, U., Nobile, M.S.: Estimation of Fuzzy Models from Mixed Data Sets with pyFUME. In: 2023 IEEE Conf on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). pp. 1–8 (2023)
17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., e.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
18. Salgado, C.M., Fernandes, M.P., Horta, A., Xavier, M., Sousa, J.M.C., Vieira, S.M.: Multistage modeling for the classification of numerical and categorical datasets. In: 2017 IEEE Int Conf on Fuzzy Systems (FUZZ-IEEE). pp. 1–6 (2017)
19. Setnes, M., Babuška, R., Kaymak, U., van Nauta Lemke, H.R.: Similarity measures in fuzzy rule base simplification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **28**(3), 376–386 (1998)
20. Spolaor, S., Fuchs, C., Cazzaniga, P., Kaymak, U., Besozzi, D., Nobile, M.S.: Simpful: a user-friendly python library for fuzzy logic. International Journal of Computational Intelligence Systems **13**(1), 1687–1698 (2020)
21. Spolaor, S., Fuchs, C., Kaymak, U., Nobile, M.S.: A novel multi-objective approach to fuzzy clustering. In: 2019 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 850–857. IEEE (2019)
22. Spolaor, S., Nobile, M.S., Mauri, G., Cazzaniga, P., Besozzi, D.: Coupling mechanistic approaches and fuzzy logic to model and simulate complex systems. IEEE Transactions on Fuzzy Systems **28**(8), 1748–1759 (2019)
23. Spolaor, S., Scheve, M., Firat, M., Cazzaniga, P., Besozzi, D., Nobile, M.S.: Screening for combination cancer therapies with dynamic fuzzy modeling and multi-objective optimization. Frontiers in genetics **12**, 617935 (2021)
24. Sugeno, M.: Industrial Applications of Fuzzy Control. Elsevier Science Inc., New York, NY (1985)
25. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics (1), 116–132 (1985)
26. Wagner, C., Miller, S., Garibaldi, J.M.: A fuzzy toolbox for the R programming language. In: 2011 IEEE Conference on Fuzzy Systems (FUZZ-IEEE 2011). pp. 1185–1192. IEEE (2011)
27. Xie, X., Beni, G.: A validity measure for fuzzy clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(8), 841–847 (1991)
28. Zadeh, L.: Fuzzy sets. Information and Control **8**(3), 338–353 (1965)