http://dx.doi.org/10.1109/FUZZ-IEEE.2017.8015503

The FISDeT Software: Application to Beer Style Classification

Giovanna Castellano, Ciro Castiello, and Anna Maria Fanelli
Department of Informatics
University of Bari Aldo Moro
via E. Orabona 4, Bari, I-70125, Italy
Emails: giovanna.castellano@uniba.it, ciro.castiello@uniba.it, annamaria.fanelli@uniba.it

Abstract—**This paper presents FISDeT, a Python tool that enables the design of a Fuzzy Inference System (FIS) based on the standard language FCL. FISDeT includes a GUI that enables the user to easily define and update the rule base of a FIS. Given a FIS, the tool can perform the inference of fuzzy rules. To show the main features of FISDeT, in this paper we employ the tool to develop fuzzy rule-based systems that can solve the problem of beer style classification. The integrated testing facilities of FISDeT enable a comparison among the created classifiers.**

I. Introduction

Fuzzy Inference Systems (FISs) are versatile tools to manage knowledge about specific domains. Their strong points are the flexibility to handle imprecise information and the readability of their knowledge base which is expressed in form of rules. A FIS can be manually compiled, possibly resorting to the expertise of people who are well-acquainted with the domain of interest, or automatically derived, triggering some specific procedure oriented to derive knowledge directly from available data. In any case, information is stored in the classical IF-THEN format which is appropriate both to accumulate knowledge and to exploit it in order to make inferences on novel observations.

FISs have been successfully applied to solve control and decision making problems, especially those which can be hardly translated into a standard mathematical model, while being easily described by means of linguistic IF-THEN rules [1]–[3]. As a consequence, several software tools have been proposed in literature to generate, manipulate and execute a FIS, in order to support even the users with reduced expertise. There exist some powerful and well known commercially available tools, such as the Fuzzy Toolbox and the Adaptive Neuro-Fuzzy Inference System (ANFIS), both provided by Matlab (www.mathworks.com), or the fuzzy package provided by Wolfram Mathematica. Several other tools are currently available as open source software (mostly from SourceForge or Google-Code) and are based on Java or C++ languages. Despite the importance of free software in the scientific research community, some of the existing packages and libraries enable the design of a FIS only for special purposes. Such specific tools are simple and easy to use, but they usually suffer from limited functionalities. Among the general-purpose tools, some open source products are worthy to be mentioned since they offer good performance and human-friendly interfaces [4]. FisPRO (Fuzzy Inference System Professional) is a general-purpose tool developed in Java that provides an interactive environment for designing and optimizing fuzzy inference systems allowing the easy integration of knowledge (coming from experts or extracted from data) [5]; GUAJE (Generating Understandable and Accurate fuzzy models in a Java Environment) [6] is a user-friendly portable tool designed and developed to simplify the processes of knowledge extraction and representation for fuzzy systems, paying special attention to interpretability issues; Xfuzzy is an environment which allows the easy description, verification, and synthesis of fuzzy systems using the specification language XFL [7]. Several other non commercial, open source, software and fuzzy packages have been reviewed in [8]. Despite the large number of packages available for FIS creation and manipulation, only few of them adopt the Fuzzy Control Language (FCL), a structured language oriented to the definition of control systems based on fuzzy logic. Such a language has been standardized by the IEC 61131-7 specification [9].

In [10] we recently introduced a new software tool called FISDeT (Fuzzy Inference System Development Tool) aiming to facilitate the creation and the manipulation of fuzzy rule-based systems. FISDeT supports the FCL standard and it has been designed bearing in mind the main guidelines for developing software for fuzzy systems [8]: interoperability, novelty, usability, and relevance. In fact, the adoption of the FCL standard enables the production of fuzzy inference systems which are portable regardless of the working platforms or the imposed representational constraints (interoperability). An analysis of the existing panorama let us focus on some points of interest and originality, resorting to Python language as a tool of implementation, and equipping FISDeT with a suitable GUI (novelty). The introduction of a GUI visual support, together with the open source qualification of our tool, provide some benefits for those interested in analyzing, verifying or adapting FISDeT for their own purposes (usability). FISDeT is publicly available on the well-known Github platform, with a project of progressive integration of resources for user-supporting and software updating (relevance).

In this paper, we are going to show the main features of FISDeT while employing it to create and manipulate fuzzy rule based systems to be applied in a specific classification problem. Particularly, we explore the wide domain of beer styles which is often neglected by casual consumers, but it is taken in greater account among enthusiasts and connoisseurs. Actually beer is a term denoting a great variety of alcoholic beverages obtained through the fermentation of cereals (typically, malted barley) which in most cases is added with some other ingredients, like hops, used for flavouring and/or preserving the final product. According to a number of factors — ranging from the fermentation conditions, to the variety of the involved ingredients and the pre-processing of malt which can be even roasted — the beer specializes into several styles, i.e. labels denoting the overall character of the beverage. In many cases, the geographical origin contributes to characterize a style too: the beer, in fact, is widespread through the world and every human culture inevitably introduced some peculiarity in the brewing process. In this context, we carried on a twofold exploitation of FISDeT to create fuzzy classification systems for beer styles. On the one hand, we turned to an expert for compiling a set of rules in form of IF-THEN assertions that can be easily translated into a FIS file through the facilities offered by FISDeT. On the other hand, we built up a dataset of pre-classified beers, useful to trigger an automatic process of knowledge extraction in form of a FIS. The derived FIS file is then acquired through FISDeT in order to visualize its contents and eventually modify some settings. The integrated facilities allows also to test the FIS files, thus enabling a comparison between the created classification models.

II. Overview of FISDeT

In this section we provide a quick overview of FISDeT, specifying some characteristics and sketching its architecture. A thorough presentation of FISDeT is reported in [10].

A. Implementation details and standards

FISDeT has been implemented in Python, partly realizing from scratch the involved modules and partly resorting to *pyfuzzy* [11], an open source library for fuzzy logic computation. This implementation choice resides one of the characterizing feature of our tool: to the best of our knowledge, no open source tool based on the Python language has been developed so far. The main advantage deriving from the adoption of Python and *pyfuzzy* consists in platform independence: as known, Python has excellent portability and is compatible with any existing operating system.

Moreover, FISDeT has been implemented taking into account the issue of standardization related to the development of fuzzy systems through the adoption of specific tools. In many cases, in fact, the derived models are encoded into some particular formats, with very different specifications and syntax. Conversely, FISDeT adheres to a standard FIS design, established in terms of a Fuzzy Control Language (FCL). FCL is a standard for Fuzzy Control Programming published by the International Electrotechnical Commission (IEC) [9], an organization devoted to the preparation and publication of International Standards for all electrical, electronic and related technologies (collectively known as "electrotechnology"). IEC standards are fully consensus-based and rely also on a cooperation with the International Organization for Standardization (ISO) to ensure that international standards fit together seamlessly. The international Standard IEC-61131 is specifically conceived for programmable controllers and it is composed by several parts. Among them, Part 7 (IEC-61131-7) is devoted to fuzzy control programming and specifies the particular syntax to define the fuzzy control applications programmed in FCL. Particularly, they must be encapsulated in Function Blocks (FBs), i.e. constructs introduced in a previous part of the IEC-61131 standard. The specific FB to be used in FCL must include some standard sections encoded as VAR_INPUT and VAR_OUTPUT, corresponding to the definition of the input and output linguistic variables. These elements simply specify the variable names and types. The linguistic variables are then described through linguistic terms, that are fuzzy sets whose membership functions are intended as piece-wise linear functions specified as sequences of points. The description of the linguistic variables is included in the FUZZIFY and DEFUZZIFY sections, which are composed by various TERM statements. The FUZZIFY and DEFUZZIFY sections refer to input and output variables respectively; particularly, the DEFUZZIFY section enables also the definition of singleton membership functions and includes a METHOD statement to specify the defuzzification method chosen among a set of candidates. Finally, the RULEBLOCK section includes the rule base specified in terms of IF-THEN statements where the linguistic variables are related to the linguistic terms.

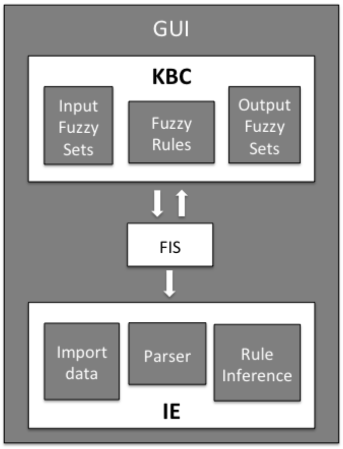


Fig. 1: Architecture of FISDeT.

B. FISDeT architecture

As shown in Fig. 1, the architecture of FISDeT can be broadly described in terms of a pair of modules communicating with a data structure (the FIS file adhering to the FCL standard encoding). The Knowledge Base Creator (KBC) module is organized to enable the user-friendly realization of a FIS. The user can describe the input/output variables: for each of them the linguistic terms can be specified as fuzzy sets whose membership functions may be triangular, trapezoidal or piece-wise linear (singletons are also allowed for the description of output variables). The fuzzy sets are suitably plotted for visualization and the user can proceed with the insertion of other variables and/or the specification of additional linguistic terms for the same variable (variables and linguistic terms can be modified or removed by the user at any time). The description of output variables includes also the specification of a defuzzification method. Finally, the user can easily assemble all the inserted information to build up the rule base, which is structured in form of IF-THEN statements in a very intuitive way: it just suffices to correlate the input/output linguistic variables to compose each single rule.

The KBC module is realized in Python and is endowed with GUIs specifically designed to drive the user through the various steps of the knowledge base compilation. Additionally, it is important to highlight that all the above operations can be replied also on the basis of existing FIS files: they can be imported in FISDeT (provided that they are encoded in the FCL standard) and the user is allowed to modify them adding, removing or updating their contents.

The Inference Engine (IE) module takes charge of the inference process producing the output of the fuzzy predictive models. Particularly, once the FIS file has been created or imported into FISDeT, the user is allowed to provide numerical input values. The IE module acquires the input and reports the output results evaluated on the basis of the involved fuzzy rule base. In this context, it is also possible to perform the predictive inference pertaining to an entire set of data: the dataset can be imported in FISDeT and analyzed in a batch fashion, so that a more comprehensive outcome is produced (this is of great interest whenever real-world problems are to be tackled). The output inferred from a single input configuration is displayed through the GUI, while the results coming from multiple data evaluations are stored in a csv file.

The procedures composing the IE module have been written in Python. They include also the mechanisms for parsing the FIS file encoded in the FCL standard.

III. Beer Style Classification

According to a recent report drawn up by the World Health Organization [12], beer is the second most consumed alcoholic beverage in the world, being also the first most consumed alcoholic beverage both in the American and European Regions. Those data should came as no surprise if we consider that beer is one of the most ancient beverages in the human history, whose origins may be traced back to the Sumerian and Babylonian civilizations: cuneiform tablets from 4000 years ago demonstrate how a culture of beer was already established when the writing practice was at its early stage.

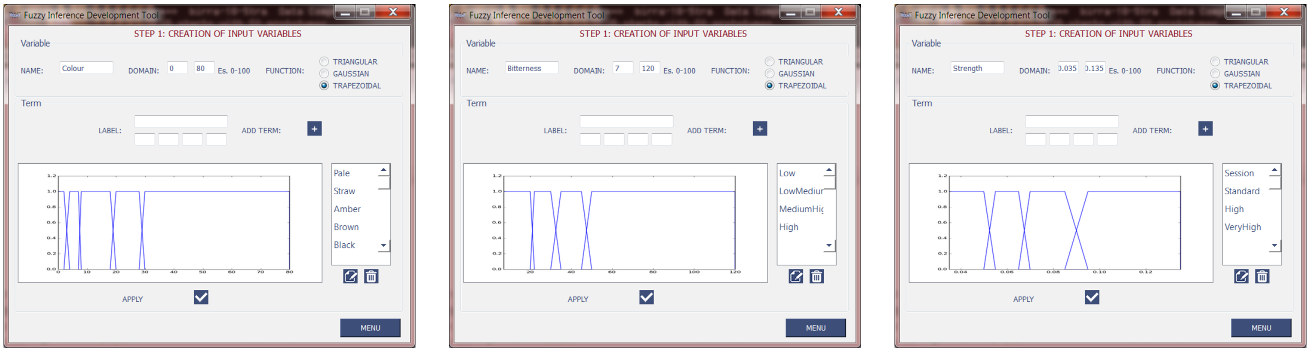
Broadly speaking, we use to think of beer in some of the classical forms that are commercially popular. However, there exists a wide variety of beers exhibiting different traits and local connotations. Beer is a fermented beverage produced from a source of starch which is often represented by (but not limited to) malt. The usual beer ingredients are water, malt, hops, yeast, and adjuncts, so that the composition of the final product is mostly represented by water, ethanol (alcohol), carbohydrate, carbon dioxide and protein. However, this is a simplification of a reality that is much more complex: hundreds of other minor components play a big role in the flavor and character of the beer [13]. To a certain extent, the concept of *beer style* is useful to capture such a diversification, which relies also on a number of other factors involving local customs, brewing traditions, and amount of ingredients.

The very idea of beer style cannot be generally defined and there is no acknowledged way to classify the large number of beer styles available. Some classifications rely solely on flavor characteristics, some others take into account historical and/or geographical elements, with reference to the country or even to the town where a style originated. In our attempt to face this peculiar classification task using FISDeT, we considered a twofold approach. On the one hand, we resorted to the expertise of a beer connoisseur, i.e. a publican we interviewed (who is unaware of fuzzy systems). On the other hand, we purposely built up a dataset of beers that are pre-classified in different styles on the basis of some features. Successively we used this dataset to automatically extract knowledge about the beer style domain in form of fuzzy rules (a specific method for FIS generation from data was applied). In any case, we aimed at deriving interpretable fuzzy rules to be expressed and manipulated with FISDeT, as well as to be evaluated by means of the inference engine enclosed in our tool. In this sense, we organized our investigation in such a way that the features and the styles characterizing the beers are the same both in the rules mentioned by the expert and in the description of the samples inside the collected dataset, as described in the following sections.

A. Using FISDeT to create an expert-dictated FIS

To face the beer style classification problem we firstly used FISDeT to manually create a suitable FIS. We resorted to an expert of this domain who was able to formulate an ensemble of rules to identify different varieties of beers. This expert — a publican accustomed to deal with beers in his own job — is not acquainted at all with fuzzy sets and rule-based systems; however, the possibility to express his knowledge in form of natural language sentences helped to fill the gap. Moreover, the GUI of FISDeT allowed to intuitively visualize the linguistic terms as fuzzy sets and to collaborate for a better tuning of the membership functions.

We settled on the involvement of eight different beer styles: *Blanche*, *Lager*, *Pilsner*, *India Pale Ale (IPA)*, *Stout*, *Barlewine*, *Porter* and *Belgian Strong Ale*. Such a selection appears to be wide enough to include beers with different traits. Even if the considered styles lend themselves to be geographically connoted, we neglected this kind of characterization. In fact, the local aspect of beer styles is nowadays considered less relevant, and a style made famous in some specific country may see its most authentic realization in some other world regions [13]. Moreover, according to the expert, the beer peculiarities can be directly appreciated even by casual drinkers by resorting to sensory features. This is



Fuzzy Inference Development Tool
STEP 1: CREATION OF INPUT VARIABLES
Variable
Term
LABEL:
ADD TERM:
APPLY
MENU

(a)

NAME: Colour DOMAIN: 0 80 Es. 0-100 FUNCTION: ○ TRIANGULAR ○ GAUSSIAN ● TRAPEZOIDAL

APPLY ☑

Terms: Pale, Straw, Amber, Brown, Black

(b)

NAME: Bitterness DOMAIN: 7 120 Es. 0-100 FUNCTION: ○ TRIANGULAR ○ GAUSSIAN ● TRAPEZOIDAL

APPLY ☑

Terms: Low, LowMediur, MediumHic, High

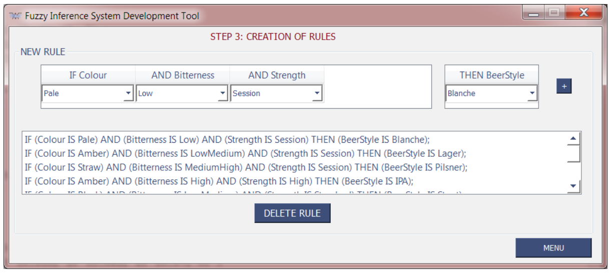
(c)

NAME: Strength DOMAIN: 0.035 0.135 Es. 0-100 FUNCTION: ○ TRIANGULAR ○ GAUSSIAN ● TRAPEZOIDAL

APPLY ☑

Terms: Session, Standard, High, VeryHigh

Fig. 2: Definition of fuzzy sets for the three linguistic variables (a) beer color (b) beer bitterness and (c) beer strength.



Fuzzy Inference System Development Tool
STEP 3: CREATION OF RULES
NEW RULE
+
DELETE RULE
MENU

IF Colour: Pale | AND Bitterness: Low | AND Strength: Session | THEN BeerStyle: Blanche

```
IF (Colour IS Pale) AND (Bitterness IS Low) AND (Strength IS Session) THEN (BeerStyle IS Blanche);
IF (Colour IS Amber) AND (Bitterness IS LowMedium) AND (Strength IS Session) THEN (BeerStyle IS Lager);
IF (Colour IS Straw) AND (Bitterness IS MediumHigh) AND (Strength IS Session) THEN (BeerStyle IS Pilsner);
IF (Colour IS Amber) AND (Bitterness IS High) AND (Strength IS High) THEN (BeerStyle IS IPA);
```

Fig. 3: Compilation of linguistic fuzzy rules.

a benefit for our investigation, since we are interested in assessing the beers as objects of classification to be evaluated in terms of perceptible (possibly measurable, even in an imprecise way) dimensions. On this basis, the features considered for beer style classification were: *color*, *bitterness* and *strength*. On the one hand, the appreciation of these features is straightforward for anyone how is going to enjoy a beer; on the other hand, these traits are included in the guidelines compiled by distinguished institutions, which are currently adopted as reference for brewers and beer competitions [14]. Interestingly enough, those guidelines try to conjugate analytical computable data with perceived characters, resorting to attributes expressing relative terms of intensity in natural language (“Low”, “Medium”, “High”, etc.). This approach, which represents a common practice whenever fuzzy rules must be compiled, is analogous to the one we followed during the interview with the expert.

Once assumed the three considered features as linguistic variables, the expert defined five linguistic terms (“Pale”, “Straw”, “Amber”, “Brown”, and “Black”) associated to the beer color, four linguistic terms (“Low”, “Low-Medium”, “Medium-High”, and “High”) associated to the beer bitterness, and four linguistic terms (“Session”, “Standard”, “High”, and “Very High”) associated to the beer strength. As shown in Fig. 2, the corresponding fuzzy sets were specified and represented in FISDeT using trapezoidal membership functions. Successively, the publican’s expertise has been exploited to derive a rule base which has been easily collected through the GUI depicted in Fig. 3. The following lines show an excerpt of the created FIS encoded in the FCL standard.

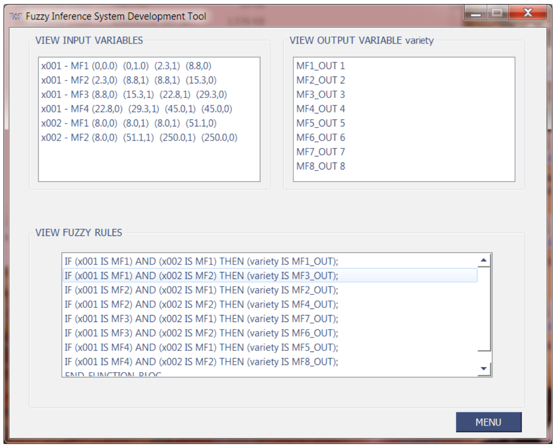
```
FUNCTION_BLOCK expert-FIS
    VAR_INPUT
        Colour : REAL;
        Bitterness : REAL;
        Strength : REAL;
    END_VAR

    VAR_OUTPUT
        BeerStyle : REAL;
    END_VAR

    FUZZYFY Colour
        TERM Pale := (0, 0) (0, 1) (2, 1) (4, 0) ;
        TERM Straw := (2, 0) (4, 1) (7, 1) (8, 0) ;
        TERM Amber := (7, 0) (8, 1) (18, 1) (20, 0) ;
        TERM Brown := (18, 0) (20, 1) (28, 1) (30, 0) ;
        TERM Black := (28, 0) (30, 1) (80, 1) (80, 0) ;
    END_FUZZYFY

    FUZZYFY Bitterness
        TERM Low := (7, 0) (7, 1) (20, 1) (22, 0) ;
        TERM LowMedium := (20, 0) (22, 1) (30, 1) (35, 0) ;
        TERM MediumHigh := (30, 0) (35, 1) (45, 1) (50, 0) ;
        TERM High := (45, 0) (50, 1) (120, 1) (120, 0) ;
    END_FUZZYFY

    FUZZYFY Strength
        TERM Session := (0.035, 0) (0.035, 1) (0.05, 1) (0.055, 0) ;
        TERM Standard := (0.05, 0) (0.055, 1) (0.065, 1) (0.07, 0) ;
        TERM High := (0.065, 0) (0.07, 1) (0.085, 1) (0.095, 0) ;
        TERM VeryHigh := (0.085, 0) (0.095, 1) (0.135, 1) (0.135, 0) ;
    END_FUZZYFY

    DEFUZZYFY output
        TERM Blanche := 1;
        TERM Lager := 2;
        TERM Pilsner := 3;
        TERM IPA := 4;
        TERM Stout := 5;
        TERM Barlewine := 6;
        TERM Porter := 7;
        TERM BelgianStrongAle := 8;
        ACCU:MAX;
    END_DEFUZZYFY

    RULEBLOCK first
        RULE 0:IF (Colour IS Pale) AND (Bitterness IS Low) AND
(Strength IS Session) THEN (BeerStyle IS Blanche);
        RULE 1:IF (Colour IS Amber) AND (Bitterness IS LowMedium)
AND (Strength IS Session) THEN (BeerStyle IS Lager);
```

```
        RULE 2:IF (Colour IS Straw) AND (Bitterness IS MediumHigh)
  AND (Strength IS Session) THEN (BeerStyle IS Pilsner);
        RULE 3:IF (Colour IS Amber) AND (Bitterness IS High) AND
  (Strength IS High) THEN (BeerStyle IS IPA);
        RULE 4:IF (Colour IS Black) AND (Bitterness IS LowMedium)
  AND (Strength IS Standard) THEN (BeerStyle IS Stout);
        RULE 5:IF (Colour IS Amber) AND (Bitterness IS High) AND
  (Strength IS VeryHigh) THEN (BeerStyle IS Barlewine);
        RULE 6:IF (Colour IS Brown) AND (Bitterness IS LowMedium)
  AND (Strength IS Session) THEN (BeerStyle IS Porter);
        RULE 7:IF (Colour IS Brown) AND (Bitterness IS MediumHigh)
  AND (Strength IS VeryHigh) THEN (BeerStyle IS BelgianStrongAle);
     END_RULEBLOCK
  END_FUNCTION_BLOCK
```

B. *Using FISDeT to handle an automatically-created FIS*

As a second case study, we used FISDeT to handle a FIS that was automatically created from data. In order to collect data for the beer style classification problem, we referred to the website www.brewtoad.com which is advertised as "a place to create, share, and discover homebrew recipes". This site hosts thousands of beer recipes posted and shared by users: anyone interested in performing a homebrew process may refer to (or take the cue from) some of the reported instructions. The site provides also a number of additional information and, most interestingly, the beers corresponding to every listed recipe are illustrated in terms of the classical features adopted for beer description, being also classified in several beer styles. Particularly, color, bitterness and strength are considered for classification and are measured on specific scales: their values are expressed in terms of Standard Reference Method (SRM), International Bittering Units (IBU), and Alcohol by Volume (ABV), respectively[1]. As a final remark, it should be highlighted how the final classification of each beer into a specific style is not a deterministic process: for some recipes an additional hint specifies that they do not totally conform to the style they are associated with. This is in line with the fuzzy nuance of the beer style concept, and it represents also a useful source of noise to perturb our data collection.

By randomly referring to data in the brewtoad.com database, we built up a dataset composed of 400 samples (corresponding to distinct beer recipes), each of them being described in terms of crisp values of color, bitterness, and strength. The samples are associated with a class label ranging in the same selection of beer styles considered during the expert interview. Eight styles were considered, with 50 samples belonging to each style (class). The assembled dataset was used to automatically generate a FIS for beer style classification. To this aim, some specific algorithm for automatic rule extraction can be used, with a preference for methods enabling the construction of interpretable fuzzy systems [15]. Particularly, we employed DC*, an algorithm purposely designed to extract interpretable fuzzy rules from data on the basis of a double clustering process. DC* analyzes the multidimensional relationships among data and it is able to derive the minimal granulation level for each input feature while defining the linguistic terms [16]. DC* requires a single parameter to be set, that is the number of prototypes to be identified during the first clustering phase. This parameter represents also an upper bound for the number of generated rules. We performed three sessions of 10-fold cross validation in order to find a suitable value for the DC* parameter, specifying in turn 8, 16, and 32 prototypes (8 rules is the minimum reasonable amount to classify beers into eight styles). We found that 32 was the most appropriate number of prototypes, giving a classification accuracy of 86.25%. Using this number of rules we launched the DC* algorithm in order to automatically derive a FIS on the basis of the entire dataset. The resulting FIS is depicted in Fig. 4 which shows another visualization facility provided by FISDeT. As can be seen, DC* produced 8 rules involving only two input variables out of the three initially considered, namely color and bitterness. The color and the bitterness variables were associated with four and two linguistic terms, respectively. Each linguistic term is represented in form of a trapezoidal fuzzy set.


Fuzzy Inference System Development Tool
VIEW INPUT VARIABLES
VIEW OUTPUT VARIABLE variety
VIEW FUZZY RULES
MENU

```
x001 - MF1 (0,0.0)  (0,1.0)  (2.3,1)  (8.8,0)
x001 - MF2 (2.3,0)  (8.8,1)  (8.8,1)  (15.3,0)
x001 - MF3 (8.8,0)  (15.3,1)  (22.8,1)  (29.3,0)
x001 - MF4 (22.8,0)  (29.3,1)  (45.0,1)  (45.0,0)
x002 - MF1 (8.0,0)  (8.0,1)  (8.0,1)  (51.1,0)
x002 - MF2 (8.0,0)  (51.1,1)  (250.0,1)  (250.0,0)
```

```
MF1_OUT 1
MF2_OUT 2
MF3_OUT 3
MF4_OUT 4
MF5_OUT 5
MF6_OUT 6
MF7_OUT 7
MF8_OUT 8
```

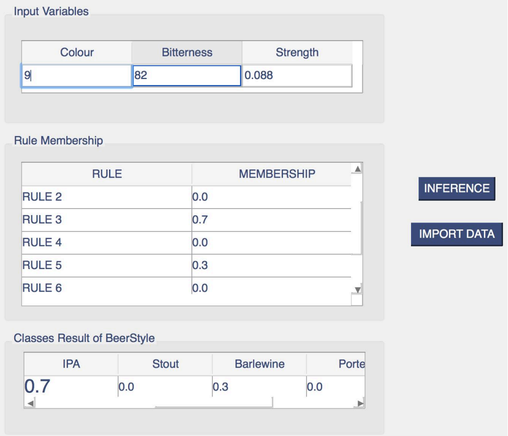
```
IF (x001 IS MF1) AND (x002 IS MF1) THEN (variety IS MF1_OUT);
IF (x001 IS MF1) AND (x002 IS MF2) THEN (variety IS MF3_OUT);
IF (x001 IS MF2) AND (x002 IS MF1) THEN (variety IS MF2_OUT);
IF (x001 IS MF2) AND (x002 IS MF2) THEN (variety IS MF4_OUT);
IF (x001 IS MF3) AND (x002 IS MF1) THEN (variety IS MF7_OUT);
IF (x001 IS MF3) AND (x002 IS MF2) THEN (variety IS MF6_OUT);
IF (x001 IS MF4) AND (x002 IS MF1) THEN (variety IS MF5_OUT);
IF (x001 IS MF4) AND (x002 IS MF2) THEN (variety IS MF8_OUT);
END_FUNCTION_BLOC
```

Fig. 4: The fuzzy rule base automatically generated from data.

C. *Using FISDeT to evaluate a FIS*

Once a FIS is available, it can be used to infer output values from input data through the Inference Engine embedded in FISDeT. Fig. 5 shows how the GUI enables the user to insert some input values which determine the firing of the rules and the fuzzy activation of output classes. We used this facility of FISDeT to test the pair of FISs at hand in a batch fashion against a whole set of data. To this end, we imported our beer dataset in FISDeT and we firstly launched the inference process using the expert-dictated FIS. Then, we imported in FISDeT the automatically generated FIS and run the inference process. Table I compares the classification accuracy obtained by the FIS dictated by the human expert and the FIS automatically extracted from data. Since the automatically-generated FIS has been tested on the same data employed for its creation, the most suitable comparison should be drawn between the performance of the expert-dictated FIS (81.25%)

[1] Actually, brewtoad.com involves a couple of additional features for beer style classification, namely Original Gravity (OG) and Final Gravity (FG). We did not considered such features, which represent very technical measurements of beer density, in order to adhere to the classification scheme proposed by the expert. Such a decision is also supported by the evidence that the ABV value for each beer is a piece of information deriving from OG and FG.



Input Variables

Colour	Bitterness	Strength
9	82	0.088

Rule Membership

RULE	MEMBERSHIP
RULE 2	0.0
RULE 3	0.7
RULE 4	0.0
RULE 5	0.3
RULE 6	0.0

Classes Result of BeerStyle

IPA	Stout	Barlewine	Porte
0.7	0.0	0.3	0.0

Fig. 5: The specified input values determine the inference of the IPA beer style.

TABLE I: Comparison between the created FISs

FIS	Performance	#rules	#input variables
Expert-dictated	81.25%	8	3
Automatically-generated	87.50%	9	2

and the average accuracy of the FISs produced by DC* during the 10-fold cross validation session performed on the basis of 32 prototypes (86.25%). To complete the comparison we observe that in [17] a number of classical classifiers (including C4.5, Bayes classifier, SVM, and k-Nearest Neighbor) have been applied to tackle the beer classification problem. They are generally able to outperform DC*: this comes with no surprise since those methods focus on accuracy only, without much regard to interpretability issues. On a different plane, we highlight how the FIS originated from data is composed by 8 rules involving only 2 features, being able to provide performance results that are in line with those produced by the expert. All such evaluations can be easily performed through the employment of FISDeT and its GUI facilities.

IV. Conclusion

In this paper we presented FISDeT, a Python software enabling the design of fuzzy inference systems complying with the standard FCL language. Besides standardization, the main advantages provided by the tool are:

- Platform independence. FISDeT can be run on any hardware and operating system configuration supporting Python.
- Extensibility. FISDeT is an open source tool, which is of special interest for the research community.
- User-friendliness and simplicity. FISDeT offers a very simple and straightforward way to develop fuzzy rule-based systems. Its GUI allows non-expert users to easily define and apply fuzzy inference systems.

To show the main features of FISDeT, in this work we proposed its application to the development of FISs for beer style classification. We used FISDeT both to manually define a FIS dictated by an expert and to handle a FIS that was automatically generated from data. To this aim, we assembled a specific dataset concerning the classification of beer styles exploiting some pieces of information available on the Web. Experimentation with the tool on such a real-world application demonstrates that FISDeT offers several facilities to easily implement a FIS from scratch and to update an existing FIS. The FISDeT software package is still being updated and improved. As a hint for future work, it could be interesting to conceive some mechanism for integrating different knowledge bases, possibly deriving from heterogeneous sources (expert-dictated and automatically-generated FISs). The current release of FISDeT is available at https://github.com/Fisdet/FISDeT.

Acknowledgment

The authors would like to thank Vincenzo Pasquadibisceglie and Gianluca Zaza for their support in developing the very first release of FISDeT.

References

[1] L. A. Zadeh, "The role of fuzzy logic in the management of uncertainty in expert systems," *Fuzzy Sets Syst.*, vol. 11, no. 1-3, pp. 197–198, 1983.

[2] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*. New York, NY, USA: Wiley Interscience, 1994.

[3] V. Cherkassky, *Fuzzy Inference Systems: A Critical Review*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 177–197.

[4] D. P. Pancho, J. M. Alonso, and L. Magdalena, "Quest for interpretability-accuracy trade-off supported by fingrams into the fuzzy modeling tool guaje," *Int. J. Comput. Int. Sys.*, vol. 6, no. sup1, pp. 46–60, 2013.

[5] S. Guillaume and B. Charnomordic, "Learning interpretable fuzzy inference systems with fispro," *Inf. Sci.*, vol. 181, no. 20, pp. 4409–4427, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2011.03.025

[6] J. M. Alonso and L. Magdalena, *Generating Understandable and Accurate Fuzzy Rule-Based Systems in a Java Environment*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 212–219.

[7] I. Baturone, F. J. Moreno-Velo, S. Sanchez-Solano, A. Barriga, P. Brox, A. Gersnoviez, and M. Brox, "Using xfuzzy environment for the whole design of fuzzy systems," in *Proc. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007)*. IEEE, 2007, pp. 1–6.

[8] J. Alcalá-Fdez and J. M. Alonso, "A survey of fuzzy systems software: Taxonomy, current research trends, and prospects," *IEEE T. Fuzzy Syst.*, vol. 24, no. 1, pp. 40–56, February 2016.

[9] *International Electrotechnical Commission technical committee industrial process measurement and control. IEC 61131 - Programmable Controllers - Part 7: Fuzzy control programming.*, IEC, 2000.

[10] G. Castellano, C. Castiello, V. Pasquadibisceglie, and G. Zaza, "Fisdet: Fuzzy inference system development tool," *Int. J. Comput. Int. Sys.*, vol. 10, no. 1, pp. 13–22, 2017.

[11] "Pyfuzzy api documentation," April 2013. [Online]. Available: http://pyfuzzy.sourceforge.net/epydoc_pdf/api.pdf

[12] *Global status report on alcohol and health 2014*, World Health Organization, 2014. [Online]. Available: http://www.who.int/substance_abuse/publications/global_alcohol_report/en/

[13] R. Barth, *The chemistry of beer: the science in the suds*. Hoboken, NJ, USA: John Wiley & Sons, 2013.

[14] *2015 Beer Style Guidelines*, Brewers Association, 2015. [Online]. Available: https://www.brewersassociation.org/wp-content/uploads/2015/03/2015-brewers-association-beer-style-guidelines.pdf

[15] J. M. Alonso, C. Castiello, and C. Mencar, *Interpretability of Fuzzy Systems: Current Research Trends and Prospects*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 219–237.

[16] M. Lucarelli, C. Castiello, A. M. Fanelli, and C. Mencar, "Interpretable knowledge discovery from data with DC*," in *Proc. Conf. Int. Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15)*. Atlantis Press, 2015, pp. 815–822.

[17] C. Castiello, A. M. Fanelli, M. Lucarelli, and C. Mencar, "Interpretable fuzzy partitioning of classified data with variable granularity," *Fuzzy Set Syst.*, Submitted for publication.