Learning Fuzzy Relations and Properties for Explainable Artificial Intelligence

Régis Pierrard, Jean-Philippe Poli, Céline Hudelot

Learning Fuzzy Relations and Properties for Explainable Artificial Intelligence

Régis Pierrard[1,2], Jean-Philippe Poli[1], and Céline Hudelot[2]

[1]CEA, LIST, 91191 Gif-sur-Yvette cedex, France.
[2]Université Paris-Saclay, CentraleSupélec, Mathématiques et Informatique pour la Complexité et les Systèmes, 91190, Gif-sur-Yvette, France.
{*regis.pierrard, jean-philippe.poli*}*@cea.fr, celine.hudelot@centralesupelec.fr*

Abstract

The goal of explainable artificial intelligence is to solve problems in a way that humans can understand how it does it. However, few approaches have been proposed so far and some of them lay more emphasis on interpretability than on explainability. In this paper, we propose an approach that is based on learning fuzzy relations and fuzzy properties. We extract frequent relations from a dataset to generate an explained decision. Our approach can deal with different problems, such as classification or annotation. A model was built to perform explained classification on a toy dataset that we generated. It managed to correctly classify examples while providing convincing explanations. A few areas for improvement have been spotted, such as the need to filter relations and properties before or while learning them in order to avoid useless computations.

1 Introduction

Providing an explanation to the outputs provided by AI-based systems, and in particular machine-learning-based ones, has become more and more important [1]. Indeed, the European Union, with the General Data Protection Regulation [2], introduces a *right to explanation* that is going to come into effect in May 2018. Although unclear, this right states that a data subject has the right to "meaningful information about the logic involved" [3]. Besides, many fields directly dealing with human beings, such as healthcare, cannot rely blindly on unexplained automatically-generated decisions. For instance, a computer-aided diagnosis cannot be fully trusted without an explanation that a doctor can understand and check. The potential explanation should be understandable not only by the expert of the targeted domain, but also by end-users.

Most machine learning algorithms are weakly able, or even unable, to return explained outputs. Many works are being done in the deep learning community to provide deep learning algorithms the ability to be interpretable. While interpretable means that it is possible to identify, or even quantify like in decision trees, the parameters and variables that play a role in the decision, explainable has a stronger meaning. An output is explainable if it is possible to make an explicit link between the values of the variables and the impact they have on the output. Although deep learning algorithms are really efficient, they are not suited to provide a personalized explanation for each output. Sparse linear models [15] are much more interpretable and can be also efficient, but they cannot provide a true explanation. A more explainable solution relies on rule sets. Rules are convenient for generating explanations. However, they are not able to deal with vagueness or uncertainty, which makes them unable to adapt to variations in data. Fuzzy rules are a suitable way to both manage vagueness and provide explained outputs. The fuzzy logic framework allows to manage the inaccuracy

of the input data and also the vagueness of the vocabulary used to explain outputs. A usual way to generate those rules is to find out frequent attributes in the dataset that is being studied and then to combine those attributes to build reliable rules. Thus, frequent attributes that can easily be interpreted, such as fuzzy relations and fuzzy properties, can be used to generate an explanation.

In this paper, we propose an explanation algorithm based on frequent fuzzy relations and properties. The purpose of our approach is to perform a task, such as classification or annotation, providing not only an output but also an explanation that can be useful for understanding where the output comes from or just for describing the input. Inputs are composed of fuzzy or crisp entities. First, frequent relations and properties are assessed and learnt on a training set. Then, they are used to perform a task that requires explanation.

The remainder of this paper is organized as follows. Section 2 sets up definitions and results that we need but are not part of our contribution. Section 3 reviews related works. Section 4 is devoted to describing our approach. In section 5, we present an example of explained classification of images from a toy dataset that we made. We conclude in section 6.

2 Background

2.1 Rule-based Systems

Rule-based systems are a particular case of expert systems [22]. Rules offer a structuration of the knowledge involved in such systems as a IF-THEN pair of condition and conclusion. The principle is then to fire rules regarding the presence of facts and to observe their conclusions.

These systems differ in terms of formalism used for knowledge representation (i.e. logic, fuzzy logic, etc.) and the algorithms which are used to infer new knowledge (i.e. RETE, Mamdani inference, etc.).

Activated rules involve pieces of evidence which can be reformulated to build an explanation. With the need of explanation, everyone can observe a renewed interest in rule-based systems [17].

In this article, we chose to use fuzzy logic because fuzzy rules look closer to natural language than other formalisms and can thus facilitate the generation of human-readable explanations.

2.2 Fuzzy Relations

A crisp relation represents either the presence or the absence of an interconnectedness between the elements of two or more sets (i.e association, interaction, etc.).

Zadeh generalized this concept to take into account various degrees of strength of relation [24] which can be represented by membership grades in a fuzzy relation.

More formally, a fuzzy relation is a fuzzy set defined on the Cartesian product of crisp sets $X_1 \times X_2, \ldots, X_n$ in which tuples $(x_1, \ldots, x_n)$ will have varying degrees of membership within the relation. Such a relation is denoted $R(x_1, \ldots, x_n)$ and is included in or equal to $X_1 \times X_2, \ldots, X_n$.

Several papers can be found in the literature about fuzzy relations on various domains. In the spatial domain, [6] describes the relative positions of fuzzy geometrical region. Online temporal relations have been introduced in [9, 25] between vague time periods or fuzzy events. Regarding spatio-temporal relations, Le Yaouanc *et al.* [10] proposed relations for assessing if and how an object is spatially evolving in a given time span. The use of such relations change fuzzy rules into so-called fuzzy relational rules.

2.3 Fuzzy Relational Rules

A fuzzy relational rule [13, 23] is a fuzzy rule that contain a fuzzy relation in its antecedents, such as:

IF X_1 is $A_1 \wedge \cdots \wedge X_n$ is $A_n \wedge \{ \bigwedge_{i,j,k} \big((X_i, X_j) \text{ is } R_k\big)\}$

THEN Y is B

where $X_1, \ldots, X_n$ are the antecedent variables defined on the universes $U_1, \ldots, U_n$, Y is the consequent variable defined on the universe V, $A_1, \ldots, A_n, B$ are fuzzy subsets and for each k R_k is a fuzzy relation defined on $U_i \times U_j$. $\wedge$ is a t-norm.

While the evaluation of the rule is complex in the general case, it is simple when inputs are just singletons. For such inputs $(x_1^*, \ldots, x_n^n)$, the strength of the rule is

$$F(y) = A_1(x_1^*) \wedge \cdots \wedge A_n(x_n^*) \wedge \{ \bigwedge_{i,j,k} R_k(x_i^*, x_j^*)\} \wedge B(y) \quad (1)$$

3 Related Works

In this section we briefly introduce the related work about rule learning, then more specifically about fuzzy relations and properties learning.

3.1 Learning Rules

The set of rules is obviously the key point of the performances of rule-based systems. However, interviews of experts and the formalization of their knowledge are often the difficult and tedious part of the work. This justifies the enthusiasm of researchers for rule learning, especially since the availability of large amounts of data.

Rule learning or rule induction is an area of machine learning which consists in extracting a set of formal rules from observations. These set of rules may represent a full model of the data or local patterns in the data [26]. Techniques vary regarding the formalism and the goal of the rules but always consider observations described by attributes and eventually a target variable. Nowadays, rule learning is also used in data mining.

The two most known approaches are the induction of association rules and decision trees. Association rules are typically used for product associations (i.e. market basket analysis). Several successful approaches rely on frequent itemsets mining [27] by browsing a lattice. Decision trees are often built with a greedy partitioning algorithm based on a splitting criteria, classically the entropy of subsets obtained by partitioning regarding a selected attribute.

More recently, Evans and Grefenstette [29] propose a differentiable inductive logic framework to learn explanatory rules from noisy data.

These various algorithms has been ported to fuzzy logic, like fuzzy a-priori algorithm [31], fuzzy decision trees [30]. In these algorithms, fuzzy logic brings its support to deal withthe vagueness of knowledge and the uncertainty of the decision. The difficulty with rule learning in fuzzy logic relies on the fact that both linguistic variables and rules have to been inducted from data.

In this paper, we pay a special attention to fuzzy relations and properties learning.

3.2 Learning Fuzzy Relations and Properties

Neural networks have been used to learn fuzzy rules. Ciaramella *et al.* [11] proposed a fuzzy relational neural network architecture to learn rules relying on fuzzy relations. These relations are built during the learning phase. As a consequence, a linguistic interpretation of these relations is difficult to obtain. That means this kind of model is not well-suited for generating explanations. González *et al.* [12] presented an algorithm for learning fuzzy relational rules [13]. It relies on a genetic algorithm to learn fuzzy relational rules that are suitable for interpretability. As there are many possible fuzzy relations, the search space is big. In order to curb the search space complexity, two filter phases are performed. The first one consists in keeping only the relations that are considered relevant by an expert relatively to the problem that is being addressed. The second phase consists in pruning the set of relations that can be learnt using a measure of information. This measure quantifies how much information a fuzzy relation provides regarding the goal.

4 Proposed Approach

In this section, we present our proposed approach, a new way of generating explained outputs. We first describe the global goal of our approach in Section 4.1. Then, in Section 4.2, we detail this approach that consists in extracting the frequent fuzzy relations and properties from inputs. In Section 4.3, we go into further detail on how frequent fuzzy relations and properties are extracted before describing how explained outputs are generated in Section 4.4.

4.1 Goal

Our goal is to make our approach able to solve a problem while providing an explanation to the solution of this problem. How relevant this explanation is depends on the frequent fuzzy relations and properties that have been extracted. That means that the original set of fuzzy relations and properties from which the frequent ones are extracted has to be constructed wisely. A poor choice of relations could lead to irrelevant explanations. That means that

the relevancy of outputs depends on how expressive the model is.

Several works have been done about the *expressivity*, or *expressive power*, of a language. Baader [19] gave a formal definition of the expressive power of knowledge representation languages. While this definition enables to compare the expressive power of two different knowledge representation languages[1], it does not define formally the expressive power of one knowledge representation language. Raghu *et al.* [21] proposed an approach to measure the expressive power of neural networks. They define the expressivity as the influence of the architecture of a neural network over the resulting functions it computes. That is why, in the following, we define the expressivity in a way that is more suited to the formalism we work in.

The inputs we deal with are composed of entities. They are objects or part of objects. They can be either crisp or fuzzy. For example, those can be fuzzy objects in an image or words in a text. We also handle fuzzy relations and properties. We will use them to characterise entities and the relations between them. Let R be the set of all the relations we work with. Let P be the set of all the properties as well. The expressivity $\mathcal{E}$ of the kind of model we propose can then be defined as the set of all possible combinations of relations from R and properties from P. All these combinations of relations and properties are potential explanations for a model.

Let us assume in the following that relations and properties are chosen by an expert. That dismisses the possibility that explanations are irrelevant because some relations and properties are not available and cannot be learnt. Explanations are a combination of relations and properties applied to entities. They are included in $\mathcal{E}$. Theoretically, the more expressive a model is, the more relevant explanations could be. However, it also depends on how well relevant relations and properties are learnt. A very expressive model may not lead to relevant explanations if the learning process is not efficient. Also, the expressivity of a model built with our approach is limited by the number of relations and properties that can be represented by a fuzzy relation or a fuzzy set. Besides, the sizes of R, P and E have an impact on how long the learning phase is.

[1]This definition states that two knowledge representation languages have the same expressive power if and only if one language can be expressed by the other and vice versa.

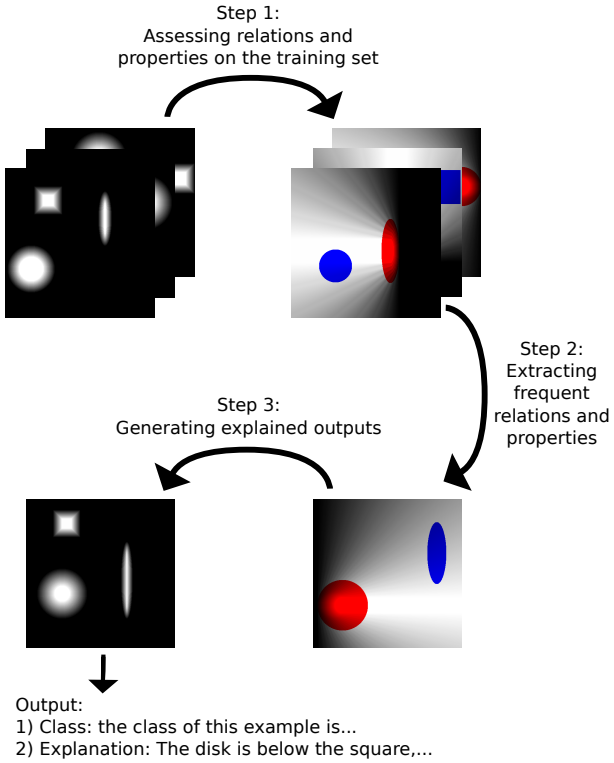


Figure 1: Schema representing the workflow. There are three different steps. Here, the output the classification of the input image and its explanation.

4.2 Overview

In this section, we detail the steps that make up our approach. As stated in the previous section, the goal of a model built with our approach is to return an explained output to a specific problem. In order to achieve this task, relevant fuzzy relations and properties are inferred from a training set by extracting the most frequent ones. Then, they are used to perform a given task. As all relevant relations and properties have a linguistic interpretation, the ones that are used to perform the task are also used to generate an explanation.

Let us assume that an expert sets the relations and properties that may be of interest. Given the entities in the data that are used, the expressivity $\mathcal{E}$ of the model we want to build is known. The workflow, represented on Fig. 1, is the following:

1. Assessing every relations and properties from $\mathcal{E}$ on the training set. That means that n-ary relations are computed for each possible n-tuple of entities and properties are computed for each entity. Those are the features that are computed for each example. Depending on the size of $\mathcal{E}$, this step may take a long time;

2. Extracting frequent relations and properties from $\mathcal{E}$ based on the results of the previous step. The algorithm that we use for performing this task is described in the next section;

3. Generating explained outputs using frequent relations and properties computed in the previous step.

As stated in [12], another filter might be needed additionally to the filtering performed by the expert. Indeed, $\mathcal{E}$ might be too big which can make step 1 long. Moreover, fuzzy entities may not exist in the original raw dataset, so an additional step might be needed at the beginning of the workflow to compute them. In step 2, how frequent we would like relations and properties to be is not set automatically. This parameter, called minimum support threshold and defined in the next section, has an impact on the final result. For instance, if this parameter is set too high, then a risk exists that no relation or property is extracted.

4.3 Extracting Frequent Fuzzy Relations and Properties

In order to find relevant fuzzy relations and properties, our approach relies on mining frequent relations and properties from the training set. To do that, we use a frequent itemset mining algorithm. Given that the examples from one class of data should share some relations and properties, there should be a correlation between those data.

At the end of step 1 in the workflow that we presented in the previous section, we get a fuzzy formal context [14]. A relational database with fuzzy values can be represented as a fuzzy formal context by a triplet $\langle \mathcal{O}, \mathcal{A}, \mathcal{R} \rangle$. $\mathcal{O}$ is a finite set of objects, $\mathcal{A}$ is a finite set of attributes and $\mathcal{R}$ is a binary fuzzy relation defined as $\mathcal{R} : \mathcal{O} \times \mathcal{A} \rightarrow [0, 1]$. Here, $\mathcal{O}$ is the set of examples in the dataset and $\mathcal{A}$ is the set of relations and properties that have been computed for each example and each entity in step 1. For any $o \in \mathcal{O}$ and any $a \in \mathcal{A}$, $\mathcal{R}(o, a)$ is the value of the relation or property a assessed on example o.

Here, a frequent set of attributes means that the support[2] of this set is larger than a minimum support threshold that has to be specified by a human user.

A closure operator is defined on this fuzzy formal context. It takes a set of attributes as argument. What this operator returns can be interpreted as the set of attributes that are shared by all the objects that include all the attributes from the argument of the operator. A set of attributes is said to be closed if and only if it is equal to its closure.

The frequent itemset mining algorithm [20] we use goes through two phases. First, it looks for every frequent closed set of attributes. Then, frequent set of attributes can be derived from all closed set of attributes.

The efficiency of this algorithm relies on how correlated data are. If they are highly correlated, then the number of frequent closed set of attributes is usually much smaller than the number of frequent set of attributes, which makes the search faster. Data from the same dataset or the same class usually share some relations and properties. The features computed in step 1 of the workflow should then be highly correlated. That is why such a frequent itemset mining algorithm is well suited to our approach.

4.4 Explanation Generation and Evaluation

Once frequent relations and properties have been extracted, explanations have to be generated. Depending on the value we set for the minimum support threshold in the previous step, some extracted relations might not be so relevant and our model could overfit the data. Ideally, a few relations and properties would be extracted when the value of the minimum support threshold is equal to 1. However, that is not realistic to expect this to happen. That is why we have to set this value carefully.

Furthermore, assessing how good an explanation is is very important. One solution is asking one or several experts to do it. We could then have an overall assessment of the model. Nevertheless, this solution may be very time-

[2]This the number of occurrences of this set of attributes out of all the objects in the fuzzy formal context.

consuming depending on the number of examples to assess.

5 Example and Experiments

This section is devoted to detail the workflow of our approach on an example of explained classification. Section 5.1 describes the problem of explained classification. After presenting the dataset in section 5.2 and the relations and properties that were used for solving this problem in section 5.3, we depict the expressivity of the model that is build in this example in section 5.4. Then, the workflow is detailed step by step in section 5.5. Finally, experiments and their analysis are presented in section 5.6.

5.1 Explained Classification

Classification is a well-known problem in the field of AI, and especially in machine learning. It consists in recognizing to which class a new input belongs. Supervised learning algorithms are used to solve it. They train a model on a training set which is composed of data that include their class, or label, and some other features that are the actual input to a classifier.

Explained classification is a specific type of classification. When a classifier is fed with an unknown input, it should return the class of this input but also an explanation stating why this input belongs to this particular class. There are several ways to solve such a problem. One could think about building a training set of examples containing their class but also an explanation for why they belong to their class. While this solution may be very efficient, building such a training set would be extremely time-consuming and would require an expert to explain all these examples. Another solution, the one we propose with our approach, is to rely on symbolic learning and use the learnt symbols for classifying unknown examples and providing an explanation. The symbols to learn are the relations and properties applied to the entities present in the inputs of the training set. As we wrote in the previous section, the fuzzy relations and properties we use are selected by an expert that can be a human being or an ontology for example. This task is much less time-consuming than providing an explanation for every example of the training set.

5.2 Dataset

We made our own dataset for illustrating this example. This is a dataset of images that contains each three fuzzy shapes: a square, a disk and an ellipse. These shapes are the entities that will be handled in this example. As we do not know initially which entity has a specific shape, those are handled as objects. Images from this dataset are divided into four classes. The difference between the classes is the spatial distribution of the fuzzy shapes. The shapes in each image of the same class are similarly spatially distributed. Examples from each class are shown in Fig. 2. The dimensions and the fuzziness of each shape in each image vary independently of the class. Each class contains 166 images and so the whole dataset is composed of 664 examples.

According to the way this dataset has been built, the

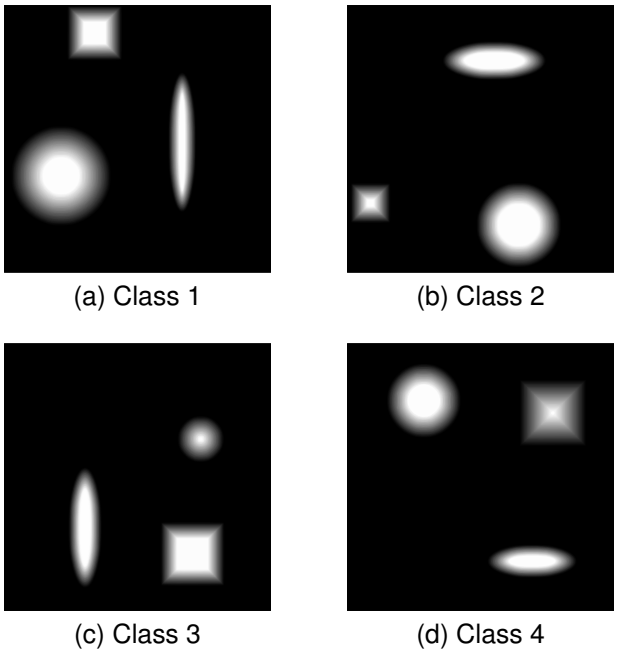


Figure 2: Examples from each class of the dataset used in the example of explained classification in section 5.

expected explanation for justifying why a new image belongs to a particular class should mention the relative position of fuzzy objects to each other and/or the absolute position of each shape in images. The relative distance between each shape also vary independently of the class, so it should not be a part of the explanation.

5.3 Interpretable Fuzzy Relations and Properties

5.3.1 Fuzzy Relations

The spatial domain has benefited from numerous studies over the last decade. Schockaert *et al.* [4] introduced a fuzzy region connection calculus framework inspired by the region connection calculus [5]. This framework includes fuzzy topological relations such as the degree of equality or the degree of overlapping between two fuzzy regions. Isabelle Bloch [6] proposed many fuzzy spatial relations and properties. She proposed set operations such as the degree of intersection between two fuzzy sets and geometric properties like the volume and the perimeter of a fuzzy set. She also proposed a relation that assesses the directional relative position between objects using fuzzy mathematical morphology. Vanegas Orozco [7] worked on geometrical relations such as the parallelism and the alignment between fuzzy regions. Clément *et al.* [16] proposed new fuzzy spatial relations representing objects imbricated in each other. Colliot *et al.* [8] studied the symmetry of fuzzy objects and proposed a measure that defines the degree of symmetry of an object with respect to a given plane.

Nevertheless, to keep this example comprehensible, we limited the relations that are used in this example to directional relations. We use particularly four of them: *to the left of*, *above*, *to the right of* and *below*. These fuzzy directional relations come from fuzzy mathematical morphology [6].

5.3.2 Fuzzy Properties

The properties are shape-related. There is one property for assessing how close or far to a disk a shape is, another one for assessing how close or far to a square a shape is and a third one for assessing how close or far to an ellipse a shape is. We call them *is disk*, *is square* and *is ellipse*.

Chanussot *et al.* [18] presented a way to extend to fuzzy shapes the shape signature based on the distance of boundary points to the shape centroid. We use this signature to build our three properties.

Let S be the signature of an entity. The property *is disk* is defined as:

$$isDisk(S) = \begin{cases} 1 - \Delta & \text{if } \Delta \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

with $\Delta = \frac{\max(S) - \min(S)}{\text{mean}(S)}$.

The properties *is square* and *is ellipse* are defined differently from *is disk*. They both return the absolute value of the correlation coefficient between S and the signature of a reference shape. These reference shapes are a perfect square and a perfect ellipse for *is square* and *is ellipse* respectively.

5.4 Expressivity

In this example, we have 4 different spatial relations and 3 different properties. The spatial relations are binary relations. They link two entities with each other whereas properties are just applied to one entity. As there are 3 different entities in the inputs from our dataset, the total number of applied relations and properties is equal to 33. Fig. 3d shows an example of one of these 33 applied relations and properties. It handles one spatial relation, *to the right of*, and two entities, *ellipse* and *disk*.

The expressivity of our model is limited because the number of possible combinations and the computation time quickly increase with the number of relations and properties.

5.5 Workflow

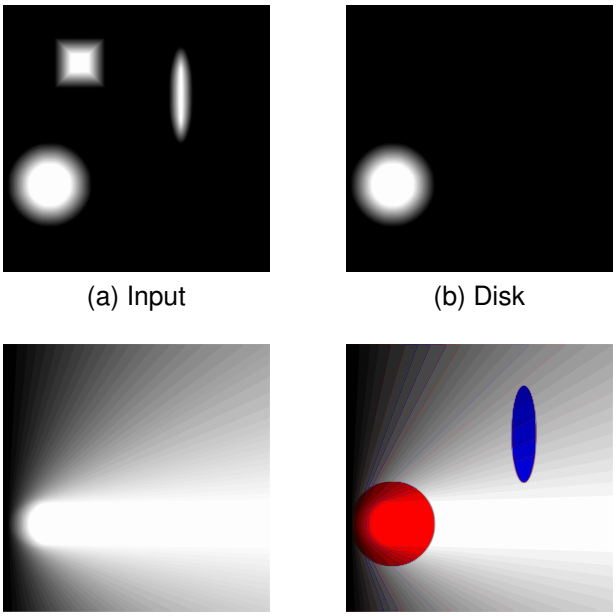
In this section, the example allows us to go into further detail regarding the workflow that is presented in section 4.2. For the sake of clarity, we refer to each entity in the inputs by their shape. However, the model does not know what the shape of an entity is until shape properties are assessed.

5.5.1 Step 1

During this step, all the relations and properties selected by the expert are assessed on the entities of each exam-

ple in the training set. Fig. 3 shows an example of how we assess the relation ellipse *to the right of disk*. In this relation, the disk is the reference. That is why we first have to extract the disk from the input in Fig. 3a. We get the image shown in Fig. 3b. Then, we compute the fuzzy landscape corresponding to *to the right of disk*. It is represented in Fig. 3c. We are now able to assess the relation *ellipse to the right of disk* that is displayed in Fig. 3d. In order to do so, we compute the degree of intersection [6] between the fuzzy set corresponding to *to the right of disk* and the fuzzy set corresponding to the ellipse. To assess *square to the right of disk*, we would perform the same operation using the fuzzy set corresponding to the square instead of the one corresponding to the ellipse. We repeat this process for each relation and each entity to assess all the relations our model can express.

Relations such as *to the right of disk*, shown in Fig.3c, take time to compute. Thus, it is important to avoid computing any useless relation in this step.



(a) Input (b) Disk

(c) *To the right of* disk (d) Ellipse *to the right of* disk

Figure 3: Example of how an input is used to compute a specific relation. Here, the goal is to compute the relation *ellipse to the right of disk*.

5.5.2 Step 2

Now, we would like to extract the most relevant relations and properties among the ones that have been assessed in step 1. In order to perform this task, we split the training set into 4 subsets. Each subset corresponds to one class. The idea is that one class of images probably has a lot of correlated data, so we can take advantage of this using the frequent itemset mining algorithm that we presented in section 4.3. We apply this algorithm for each subset. Thus, we obtain a set of relevant relations and properties for each class.

The results of this step depends on the the value of the minimum support threshold. If it is too low, we may get many irrelevant relations, and if it is too high, we may get no relations or very few.

5.5.3 Step 3

For each class, there is a subset of relevant relations and properties. We can built fuzzy relational rules based on this subset. We already know that the consequent of those rules is the class. The antecedent should be different from one class to another. Still, two classes might share common relevant relations and properties. While that may not be an issue, we can dismiss it by removing from all the subsets the intersection of two, three or four of these subsets. Then, as all our relations and properties are linguistically interpretable, we can identify the class of an unknown input and provide an explanation for this decision. As there are several sets of frequent relations and properties, there are several possible output. The class is decided by the maximum degree of membership to a class that has been computed using all the rules that were generated.

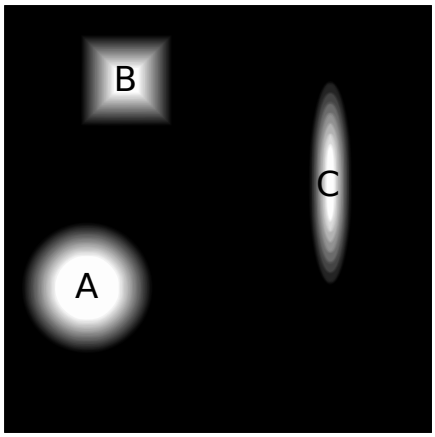
We wrote in section 4.4 that evaluating the relevancy of an explanation is tricky. Although it is still an ongoing issue, a few measures like the number of relations and properties in the antecedent or the value of their support can help to decide whether or not an explanation meets the requirements or not.

5.6 Experiments and Analysis

For these experiments, 65% of the examples from each class are part of the training set. Other examples compose the test set. Several values of the minimum support threshold have been tested. The results we present in this section have been obtained when it is equal to 0.75. The dataset being simple, the expressivity of our model is sufficient to class every example of the test perfectly. However, our main point of interest is the relevancy of the explanation that are generated.

First, Fig. 4 shows an example of classification. We can see that the most obvious relations are used for explaining the output. Moreover, one can notice that he relations *disk is below square* and *square is above disk* express the same thing. That seems obvious that if one of these two relations is used, the other one will be too. However, that is not the case for *ellipse is on the right of square* for example. There is no relation *square is on the left of ellipse* in the explanation. That is due to the way these relations are computed. Indeed, when we use fuzzy mathematical morphology, The fuzzy landscape we get depends on the shape of the reference object. So, for instance, *on the right of square* is not exactly the same as *on the right of ellipse*. That is why there are slight differences that can impact the relations and properties involved in the antecedent of the rule.

Fig. 5 shows that rules extracted for class 1 are much better when classifying examples from class 1 than when classifying examples from other classes. That was expected. Besides, we can notice that an antecedent whose size is equal to 9 leads on average to the same result as an antecedent whose size is equal to 6. This behavior is interesting because there is usually a trade-off between performance and ability to explain. Long antecedents lead to an explanation that is too long and short antecedents do not bring enough information. On this dataset, there is a range of size for which the user can get a longer explanation without harming the performance of the classifier. Furthermore, the lower the value of the minimum support threshold, the longer the antecedents. So being able to set this threshold to the right value is important in dealing with the trade-off between performance and explainability.



IF object A *is disk* And object B *is square* And square *is above* disk And disk *is below* square And ellipse *is on the right of* square And ellipse *is on the right of* disk THEN class is class 1

Figure 4: Example of a rule generated for class 1. That enables us to generate an explanation to the classification.

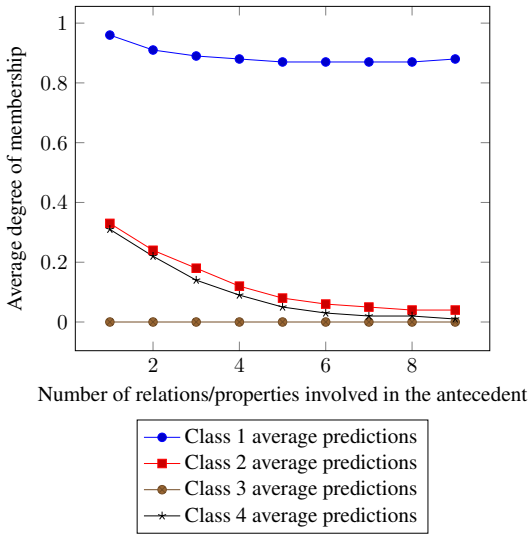


Figure 5: Evolution of the average degree of membership to class 1 with the number of relations/properties involved in the antecedent of the rule. The rules that are used here are the ones that have been extracted for class 1.

6 Conclusion

In this paper, we proposed an approach for building a model which contributes to an explainable AI based on learning relevant relations and properties in a dataset. To do so, we assess on a training set the relations and properties that have been selected by an expert. Then, the most interesting ones are extracted by using a frequent itemset mining algorithm. Several kinds of model can be built with this approach, such as a classifier as we showed in a detailed example.

This example has been applied on a toy dataset that we made. We focused on assessing the ability of our model to explain outputs. Results were encouraging as we showed that the model could generate plausible explanation. We also noticed that the trade-off between efficiency and explainability was not so sharp.

Several points still need to be studied. Building a very expressive model is very time-consuming due to the computation of every relation and property. We need to find a way to avoid useless computations by filtering some relations before or while assessing them. Moreover, a well-known issue in explainable AI is the assessment of explanations. There is no convenient way to do it at the moment. Also, being able to find the right value for the minimum support threshold is important for tending toward more efficiency or more explainability. Those are the areas that need to be investigated.

References

[1] F. Doshi-Velez and B. Kim (2017). A Roadmap for a Rigorous Science of Interpretability.

[2] Parliament and Council of the European Union (2016). General Data Protection Regulation.

[3] B. Goodman and S. Flaxman (2016). EU regulations on algorithmic decision-making and a "right to explanation". AI Magazine. 38. . 10.1609/aimag.v38i3.2741.

[4] S. Schockaert, M. De Cock and E. E. Kerre (2009). Spatial reasoning in a fuzzy region connection calculus. Artificial Intelligence, Volume 173, Issue 2, Pages 258-298.

[5] D. Randell, Z. Cui and A. Cohn (1992). A Spatial Logic based on Regions and Connection. Principles of Knowledge Representation and Reasoning: Proceedings of the 1st International Conference, Pages 165-176.

[6] I. Bloch (2005). Fuzzy spatial relationships for image processing and interpretation: a review. Image and Vision Computing, Volume 23, Issue 2, Pages 89-110.

[7] M. C. Vanegas Orozco (2011). Spatial relations and spatial reasoning for the interpretation of Earth observation images using a structural model.. Signal and Image Processing. Télécom ParisTech.

[8] O. Colliot, A. V. Tuzikov, R. M. Cesar and I. Bloch (2004). Approximate reflectional symmetries of fuzzy objects with an application in model-based object recognition. Fuzzy Sets and Systems, Elsevier, 147 (1), pp.141-163.

[9] J-P. Poli, L. Boudet and D. Mercier, "Online temporal reasoning for event and data streams processing," 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Vancouver, BC, 2016, pp. 2257-2264.

[10] J-M. Le Yaouanc, J-P. Poli. A Fuzzy Spatio-Temporal-based Approach for Activity Recognition. Springer Heidelberg. International Workshop on Semantic and Conceptual Issues in GIS (SeCoGIS 2012), Oct 2012, Florence, Italy. 7518, pp.314–324, 2012, LNCS.

[11] A. Ciaramella, R. Tagliaferri, W. Pedrycz and A. Di Nola. Fuzzy relational neural network, International Journal of Approximate Reasoning, Volume 41, Issue 2, 2006, Pages 146-163.

[12] A. González, R. Pérez, Y. Caises and E. Leyva (2012). An Efficient Inductive Genetic Learning Algorithm for Fuzzy Relational Rules. International Journal of Computational Intelligence Systems, Volume 5 - 2, Pages 212-230.

[13] R.R. Yager (1991). The Representation of Fuzzy Relational Production Rules. Applied Intelligence, Volume 1, Issue 1, Pages 35-42.

[14] R. Belohlavek (2002). Fuzzy Relational Systems: Foundations and Principles. New York: Kluwer/Plenum.

[15] J. Zeng, B. Ustun and C. Rudin (2017). Interpretable Classification Models for Recidivism Prediction. Journal of Royal Statistics - Series A.

[16] M. Clément, A. Poulenard, C. Kurtz and L. Wendling (2017). Directional Enlacement Histograms for the Description of Complex Spatial Configurations between Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 39, Issue 12.

[17] C. Grosan and A. Abraham (2011). Rule-Based Expert Systems. In Intelligent Systems: A Modern Approach, pp 149–185.

[18] J. Chanussot, I. Nyström and N. Sladoje (2005). Shape signatures of fuzzy star-shaped sets based on distance from the centroid. Pattern Recognition Letters, Volume 26, Issue 6, Pages 735-746.

[19] F. Baader (1996). A Formal Definition for the Expressive Power of Terminological Knowledge Representation Languages. Journal of Logic and Computation, Volume 6, Issue 1, Pages 33-54.

[20] R. Pierrard, J-P. Poli and C. Hudelot (2018). A Fuzzy Close Algorithm for Mining Fuzzy Association Rules. <hal-01698352>

[21] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli and J.S. Dickstein (2017). On the Expressive Power of Deep Neural Networks. https://arxiv.org/pdf/1606.05336.pdf

[22] Bruce G.Buchanan and Richard O.Duda (1983). Principles of Rule-Based Expert Systems. Advances in Computers, Volume 22, Pages 163-216

[23] R.R. Yager and D.P. Filev (1996). Relational partitioning of fuzzy rules. Fuzzy Sets and Systems, Volume 80, Issue 1, Pages 57-69.

[24] L.A. Zadeh (1965). Fuzzy Sets. Information and Control, Volume 8, Issue 3, Pages 338-353.

[25] S. Schockaert and M. De Cock (2008). Temporal reasoning about fuzzy intervals. Artificial Intelligence, Volume 172, Issues 8–9, Pages 1158-1193.

[26] J. Fürnkranz, D. Gamberger, N. Lavrač (2012). Rule learning in a nutshell. In: Foundations of Rule Learning, pages 19-55.

[27] R. Agrawal and R. Srikant (1994). Fast algorithms for mining association rules. Proceedings of the 20th International Conference on Very Large Data Bases, pages 487-499.

[28] J.R. Quinlan (1986). Induction of Decision Trees. Machine Learning, Volume 1, Issue 1, pages 81-106.

[29] R. Evans and E. Grefenstette (2017). Learning Explanatory Rules from Noisy Data. http://arxiv.org/abs/1711.04574

[30] R. L. P. Chang and T. Pavlidis (1977). Fuzzy decision tree algorithms. IEEE Transactions on Systems, Man and Cybernetics, Volume 7, Issue 1, pages 28-35.

[31] A. Mangalampalli and V. Pudi (2009). Fuzzy Association Rule Mining Algorithm for Fast and Efficient Performance on Very Large Datasets. IEEE International Conference on Fuzzy Systems, pages 1163-1168.