Expert Systems With Applications 148 (2020) 113262

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Applying Dempster–Shafer theory for developing a flexible, accurate and interpretable classifier

Sergio Peñafiel[a,*], Nelson Baloian[a], Horacio Sanson[b], José A. Pino[a]

[a] Department of Computer Science, Universidad de Chile, Santiago, Chile
[b] Allm Inc., Tokyo, Japan

ARTICLE INFO

Article history:
Received 15 July 2019
Revised 2 December 2019
Accepted 29 January 2020
Available online 30 January 2020

Keywords:
Supervised learning
Expert systems
Gradient descent
Dempster-Shafer theory
Interpretability

ABSTRACT

Two approaches have traditionally been identified for developing artificial intelligence systems supporting decision-making: Machine Learning, which applies general techniques based on statistical analysis and optimization methods to extract information from a large amount of data looking for possible relations among them, and Expert Systems, which codify experts knowledge in rules, which are then applied to a specific situation. One of the main advantages of the first approach is its greater accuracy and wider generality for the application of the methods developed which can be used in various scenarios. By contrast, expert systems are usually more restricted and often applicable only to the domain for which they were originally developed. However, the machine learning approach requires the availability of large chunks of data, and it is much more complicated to interpret the results of the statistical methods to obtain some explanation of why the system decides, classifies, or evaluates a situation in a certain way. This issue may become very important in areas such as medicine, where it is relevant to know why the system recommends a certain treatment or diagnoses a certain illness. Likewise, in the financial sector, it might be legally required to explain that a decision to reject the granting of a mortgage loan to a person is not due to discriminatory causes such as gender or race. In order to be able to have interpretability and extract knowledge of available data we developed a classification method based on Dempster-Shafer's Plausibility Theory. Mass assignment functions (MAF) must be established to apply this theory and they assign a weight or probability to all subsets of the possible outcomes, given the presence of a certain fact on a decision scenario. Thus MAF assignments encode expert knowledge. The method learns optimal values for the weights of each MAF using the Gradient Descent method. The presented method allows combination of MAF which have been generated by the method itself or defined by an expert with those that are derived from a set of available data. The developed method was first applied to controlled scenarios and traditional data sets to ensure that classifications and explanations are correct. Results show that the model can classify with an accuracy which is comparable to other statistical classification methods, being also able to extract the most important decision rules from the data.

1. Introduction

The development of methods for classifying based on machine learning techniques has been one of today's most rapidly growing technical fields in artificial intelligence in the last years (Jordan & Mitchell, 2015).

Two fundamental concepts associated with classifying instruments are performance (also called accuracy for simplicity) and interpretability. Model performance is associated with the idea of how good a classification method works. This concept has several dimensions, and accordingly, several indicators have been proposed. They are widely used today associated with this concept. Some of these indicators are accuracy, sensitivity or recall, specificity or precision, area under the ROC curve, and F1 score (Tharwat, 2018).

The concept of interpretability relates to the possibility a human user of that instrument has for tracking the decisions it made in order to classify a sample in one or another set.

Although both characteristics are desirable in a classification instrument, experience has shown that they collide, i.e., we often find limited or non-existent interpretability in most accurate methods and vice versa. As an example, Deep Learning methods based on multi-level Neural Networks are considered black-box algorithms. On the other hand, there are highly interpretable

[*] Corresponding author.
E-mail addresses: spenafie@dcc.uchile.cl (S. Peñafiel), nbaloian@dcc.uchile.cl (N. Baloian), horacio@allm.net (H. Sanson), jpino@dcc.uchile.cl (J.A. Pino).

https://doi.org/10.1016/j.eswa.2020.113262
0957-4174/© 2020 Elsevier Ltd. All rights reserved.

methods like Decision Trees, linear models, or rule-based models. However, these latter methods have restrictions that do not allow them to achieve high accuracy; for example, linear models cannot learn non-linear relationships between attributes. Rule-based models and Decision Trees are constrained to have a reduced number of rules or depth in the tree. Otherwise, the model increases its complexity and makes it less interpretable.

Sometimes the interpretability of results given by the method may be more important than high accuracy in the prediction. As an example, consider a physician who is trying to diagnose a patient's disease based on his/her symptoms. For the physician, it is more important to know which symptoms are related to a particular disease instead of having a machine predicting the disease without any explanation (Caruana et al., 2015) even in case the machine predictions are very accurate. Giving these explanations makes the users trust the system, and thus generating a helpful tool to support decision-making. Moreover, in many situations, interpretability goes beyond the trustworthiness of models because explaining decisions is a required aspect for its application. For instance, Chilean legislation on acceptance of customers' mortgage credit applications prohibits financial institutions from using certain variables to make their decision, such as ethnic group and skin color. A client whose loan application was declined may appeal to these institutions, and they must explain the cause of the credit refusal. In order to satisfy this requirement, these institutions are forced to use interpretable methods even if other models can predict better profitable potential clients; otherwise, they cannot give the previously mentioned explanation.

The exact definition of interpretability can vary depending on the context in which the model is applied. For instance, for fuzzy rule-based systems or traditional rule-based systems, interpretability is highly related to the number of rules used and the complexity of them (García, Fernández, Luengo, & Herrera, 2009). For black-box, machine learning techniques, and particularly neural network methods, interpretability relates to the ability of the model for explaining every classified instance. These two definitions for interpretability have drawbacks which make difficult its understandability.

For the case of rule-based systems, applying the rules must result in each record fitting into a class. This constraint may imply the development of complicated rules, and thus, a less interpretable model (Casillas, Cordón, Herrera, & Magdalena, 2003). Another drawback is the lack of a standard procedure to deal with the problem of missing values. If a rule relates to an attribute that is missing, there is no clear way to continue with the classification, e.g., taking the positive or negative branch or switch to another rule. If the rule concerns many attributes and just one is missing, the decision is still unclear. Missing values is a common problem when working with real data.

For the case of instance explanation of black-box methods, authors have various definitions. Some of them only distinguish between useful attributes and non-useful attributes (Štrumbelj & Kononenko, 2014), and other explanation methods give a kind of rules or human-readable statements for that instance. However, the drawback is the fact that explanations change for each instance, e.g., for a record, an attribute X could be necessary for classification, but, for another record, the same attribute X could be useless. This change does not allow giving a clear explanation of the whole model. Thus, the model is not interpretable since the user cannot understand the decision of a completely new record without tracking the whole classification process again.

A new meaning of interpretability we propose is having simple rules concerning all possible attributes of a sample and then discovering which rules are essential to the process of deciding to which class a sample belongs. In other words, each rule has a weight that indicates how important is that rule in order to decide whether or not a sample belongs to a particular class. This property defines an interpretability score for every rule instead of for a specific record or the whole model like previous approaches. Eventually, the model can drop the lightest rules to have fewer rules thus obtaining a simpler model if needed.

One advantage of this latter interpretability approach, which existing ones cannot handle, is that rules are independent of each other. This property means that every single rule is simple, understandable and applies evenly to all data records without needing to make any previous assumption. Another advantage is that the model allows the user to test custom rules. This ability is a desirable feature for developing expert knowledge because it helps to validate a hypothesis and measure its importance to explain the studied system.

In this work, we present the development of a classification method that can be applied to a wide range of decision-making scenarios. The proposed model achieves accuracy comparable to those based on Artificial Neural Networks. At the same time, the model is highly interpretable according to the definition we proposed above, thus inheriting all the mentioned advantages. In addition, this method allows classifying samples for which attributes values may be missing.

The key idea to achieve this goal is to develop a rule-based classifier based on the Dempster–Shafer Theory (Shafer et al., 1976). This theory is a generalization of Bayesian theory allowing to assign uncertainty directly. Mass assignment functions (MAF) are the elements that encode knowledge in this theory which, given the presence of a particular fact on a decision scenario (e.g., the level of humidity) assigns a weight or probability for all subsets of the possible outcomes (heavy rain, light rain, no rain). The theory also defines the Dempster Rule (Shafer, 2016), which indicates how to combine different MAFs (e.g., temperature, air pressure, and humidity) for obtaining a single result (probability of heavy, light or no rain). The use of uncertainty allows us to develop a more sophisticated rule-based model that can express complex scenarios without losing the simplicity of rules.

The model developed in this work includes a novel way to optimize mass assignment functions based on gradient descent techniques from the evidence provided by the training data. Moreover, we show how to extract explanations for classifications once the model is trained. This process is done by generating and optimizing meaningful rules automatically using the training data, which makes the model interpretable.

2. Related work

2.1. Interpretable methods

Some of the methods mentioned above are interpretable, i.e., that they can give an explanation about the decision they make when doing classifications. The most remarkable interpretable methods are described below.

2.1.1. Decision trees and random forest

A Decision Tree (Olshen & Stone, 1984) is a method for classification and regression based on simple relations among attributes. The method builds a binary tree, where each inner node represents a simple rule of an attribute, for example $X_1 > 4$. The branches are the results of the rules, a left branch means the above condition was false and a right branch means that it was true. Leaf nodes are the predicted classes. The method tries to find the rules that separate most the classes.

This method is clearly interpretable since we can descend in the decision tree with our data and know why the method gives a response.

A Decision Trees drawback is that they cannot handle complex data rules, and then the accuracy is lower compared to other methods. An extension of Decision Trees are Random Forests (Breiman, 2001) which is another method of classification that uses a collection of decision trees. Each tree in the forest is built using a variation of the data and restrictions are applied in order to ensure that trees with different characteristics will be obtained. The outcome of applying a Random Forest to a classification problem is the output node, which was chosen by the highest number of trees belonging to the forest.

2.1.2. Bayesian derived methods

Naive Bayes (Domingos & Pazzani, 1997) is a method that uses the Bayes' Theorem to find the class that gives the most posterior probability according to the data. This method makes the assumption that the probability of a class given an attribute is independent from the other attributes. Another similar method is Bayesian Networks (Friedman, Geiger, & Goldszmidt, 1997), which is a directed graph that models a certain problem. In this graph, each node is a variable or an attribute of the problem and each edge corresponds to the dependency or conditionality of these variables. In this method interpretability comes directly by the structure of the graph because it encodes all the knowledge about relationships among variables. However, the structure of the graph has to be set beforehand, therefore this method does not allow to find new knowledge, but to validate it.

As we can see from these methods, when interpretability is clearly present in these models the accuracy is lower, and when the model gets better accuracy it becomes more complex and interpretability is weaken.

2.2. Interpretability indicators in Fuzzy rule based systems

Several authors have proposed indicators to measure the interpretability of a system. Generally, these indicators are applied to fuzzy rule-based systems (FRBS). For example, García et al. (2009) propose a methodology to measure the performance and interpretability of rule-based genetics models. For performance measuring, they used indicators such as accuracy, true positive rate, and Cohen's kappa indicator, and for interpretability, they use two measurements the size of the rule set and the average of the number of antecedents. Let s_i to be a rule statement, s_i can be written as $c_1 \wedge c_2 \wedge \cdots \wedge c_k$, then the number of antecedents for this rule ant_i is k. For example, a rule with a statement "BMI $\geq$ 35" has only one antecedent. The average number of antecedents (ANT) is defined as:

$$ANT = \frac{1}{r}\sum_{i=1}^{r} ant_i \tag{1}$$

Where r is the number of rules. These two indicators are widely used in FRBS. As another example, (Ishibuchi & Nojima, 2007) use the number of rules as an interpretability metric and the error of classification $(1 - Accuracy)$ as an accuracy metric to test several configurations of a model and they find the Pareto-optimal configuration for these two variables.

However, the above metrics for interpretability heavily depend on the problem and the data they used. For example, datasets that have few attributes will tend to have much fewer rules than datasets with many attributes. To address this problem, a reformulation of these indicators was proposed by (Gorzałczany & Rudziński, 2017) the main change is that now the indicators are normalized and thus they can be comparable to performance indicators and comparable with other problems and datasets. In their work, the proposed indicator is the average of another three normalized indicators defined below:

$$\begin{aligned} Q_{RATR} &= \frac{1}{r}\sum_{r_i \in RS} \frac{ant_i - 1}{m - 1} \\ Q_{ATR} &= \frac{ANT - 1}{m - 1} \\ Q_{FS} &= \frac{n_{FS} - 1}{\sum_{i=1}^{m} a_i - 1} \end{aligned} \tag{2}$$

Where m is the number of attributes, ant_i is the number of antecedents of the ith rule, ANT is the average number of antecedents defined above, n_{FS} is the average number of active fuzzy sets, and a_i is the number of rules concerning the ith attribute.

Note that Q_{RATR}, Q_{ATR}, and Q_{FS} are in the range [0,1], being 0 the best value for high interpretability and 1 the worst. For Q_{RATR} and Q_{ATR}, a value of 0 means that the model uses simple rules that concern only to one attribute per rule, and a value of 1 means that the rules use all the attributes. For Q_{FS}, a value of 0 means that the model select only one rule to perform the classification, and a value of 1 means that they select all the rules.

The authors proposed to average these three indicators to measure the complexity of the model Q_{CPLX}, and then the interpretability Q_{INT} is the complement value of the complexity.

$$\begin{aligned} Q_{CPLX} &= \frac{Q_{RATR} + Q_{ATR} + Q_{FS}}{3} \\ Q_{INT} &= 1 - Q_{CPLX} \end{aligned} \tag{3}$$

2.3. Interpretability in other methods

Another approach that gained interest over recent years is to explain and interpret the results of a non-interpretable method for both specific instances and the whole model. Many strategies have been proposed to extract interpretability. Some of them are discussed and examples are presented below.

The first approach to understanding black-box methods is to analyze them formally by comparing or reducing them to interpretable methods. For example, Gal (2016) analyzed the formulas used to optimize the stochastic regularization techniques of an artificial neural network with dropout; the author proves that the formula obtained is equivalent to the optimization of a Bayesian network. Although this kind of techniques seems to be correct for finding interpretability, the intermediate steps to build the equivalence could be as complex as the original model. Then even if we have a new representation in an interpretable model, this does not necessarily mean that we can extract interpretable results from it.

Another approach is to use a strategy similar to sensitivity analysis to find the features that contribute most to classifications (Olden & Jackson, 2002; Štrumbelj & Kononenko, 2014). In the case of artificial neural networks (ANN), which are known to be one of the least interpretable methods, there are efforts to understand the relations between inputs and network computations using this technique. Olden and Jackson (2002) present four methods for understanding the mechanics of ANNs. The mentioned methods are based on testing with small random variations of the data and comparing the outputs differences. A drawback of this kind of approaches is that the output of the model has to be recomputed for every perturbation the method includes in the input; for large inputs or large ANNs, this method cannot perform.

Recent studies about interpretability instead of trying to understand the model itself, they only focus on the results of the model. For a model these techniques build an *explanation model* that uses the results of the original model trying to find interpretable relationships among attributes that are related to the results of the model. Some of these models can even use a different representation for the data than the original model embedding, and then they can obtain interpretability from this new representation, which is often simpler and more understandable (Bach et al., 2015; Datta, Sen, & Zick, 2016; Lipovetsky & Conklin, 2001; Ribeiro, Singh, & Guestrin, 2016a; Shrikumar, Greenside, & Kundaje, 2017).

For example, Ribeiro, Singh, and Guestrin (2016b) present Local Interpretable Model-Agnostic Explanations (LIME) as a technique to generate explanations for the results of any classifier. LIME model tries to find the local linear separation for an instance using a custom representation, but using the original model classifications for this instance. LIME has been tested in text based datasets, image datasets and tabular data showing that it can get interpretable results independent of the representation of the data. Also, LIME text results were validated with humans who had to decide which explanation of a method fits better in a real scenario showing that the interpretations the model produces are the best results.

Another example of explanation methods for artificial neural networks is DeepLift presented by Shrikumar et al. (2017). In their work they present a process that can be applied to a trained neural network when receiving an input. This process is similar to backpropagation, but with computing gradients for convenient functions that are directly related to the importance of features. The output of this process is weighted by each attribute indicating the contribution of these features to the classification.

Using these various models for extracting interpretable results, there are now current works that propose to combine them. For example, Lundberg and Lee (2017) present a method to unify the results of several explanation models to provide a measure of feature importance by adding the interpretability evidence that the different methods provide.

2.4. Dempster–Shafer theory

The Dempster–Shafer Theory (DST) (Shafer et al., 1976), also called the theory of belief functions, is a generalization of the Bayesian theory that is more expressive than classical Bayesian models since it allows to assign “masses” to multiple outcomes measuring the degree of uncertainty of the process.

Let X be the set of all states of a system called frame of discernment. A mass assignment function $\mathbf{m}$ is a function that satisfies:

$$\mathbf{m}: 2^X \to [0,1], \quad \mathbf{m}(\phi) = 0, \quad \sum_{A \subseteq X} \mathbf{m}(A) = 1 \tag{4}$$

Where A is a subset of X and ϕ is the empty set. The term $\mathbf{m}(A)$ can be interpreted as the probability of getting exactly the outcomes of the set A, and not a subset of A.

The belief metric is presented as the total evidence to support an outcome, and it is given by the following expression:

$$Bel_m(A) = \sum_{B \subseteq A} \mathbf{m}(B) \tag{5}$$

The plausibility metric is defined as the total amount of evidence that can support an outcome. This formulation is the following:

$$Pl_m(A) = \sum_{B \cap A \neq \phi} \mathbf{m}(B) \tag{6}$$

Multiple evidence sources expressed by their mass assignment functions of the same frame of discernment can be combined using the Dempster Rule (DR) (Shafer, 2016). Given two mass assignment functions $\mathbf{m_1}$ and $\mathbf{m_2}$, a new mass assignment function $\mathbf{m_c}$ can be constructed by the combination of the other two using the following formula:

$$\begin{aligned}\mathbf{m_c}(A) &= \mathbf{m_1}(A) \oplus \mathbf{m_2}(A) \\ &= \frac{1}{1-K} \sum_{B \cap C = A \neq \phi} \mathbf{m_1}(B)\mathbf{m_2}(C)\end{aligned} \tag{7}$$

Where K is a constant representing the degree of conflict between $\mathbf{m_1}$ and $\mathbf{m_2}$ and it is given by the following expression:

$$K = \sum_{B \cap C = \phi} \mathbf{m_1}(B)\mathbf{m_2}(C). \tag{8}$$

2.5. DST Applications in supervised learning

Several models using Dempster–Shafer Theory to solve supervised learning tasks have been proposed and they can be divided into two groups: Dempster-Shaffer supporting post-processing of other classifiers, which are often called data fusion methods in the literature; and methods which use DST as part of the classification process. Both approaches are described in the Table 1.

3. Proposed model

This section describes the implementation details of the classifier. The classifier is based on the previous work about stroke risk detection (Peñafiel et al., 2018), where a framework for using DST as a classifier was presented, introducing the concept of rule and interpretable decisions. In this work we generalize the model to handle multi-class classification problems, and also we present a new way to obtain the mass values based on gradient descent without losing interpretability.

3.1. Dempster shafer implementation

In this work, we use the elements of the Dempster-Shafer Theory to implement a classification model; the main features of the model are listed below:

Let $K = \{a_1, \ldots, a_k\}$ to be set of all possible classes of a classification task. Mass assignment functions (MAF) are represented as vectors with $k+1$ values, where k is the number of classes. The vector contains one value for each singleton class and another one for the complete set. Mass of null set is omitted because it is always 0, and masses for the other combinations are also omitted.

This decision means that our model supports only general uncertainty, instead of specific uncertainty as the theory indicates. The main reason for this choice is that power set grows exponentially with the number of classes, which implies that all computations would involve exponential-length vectors, and then their complexity would have been exponential too. We propose a tradeoff solution, which includes the complete set mass value where uncertainty can be measured, but keeping the length of the vector linear with the number of different classes of the problem.

$$\mathbf{m}: (m_1, m_2, \ldots, m_k, m_U) \tag{9}$$

The terms $m_1, \ldots m_U$ represent the scalar values mass function $\mathbf{m}$ assign to each singleton and to the complete set respectively.

To illustrate this solution, consider a disease detection problem where a model has to determine if a certain disease is present (P) or absent (A) in a patient. In this problem there are two classes P and A, and then a MAF $\mathbf{m}$ over this frame of discernment, assigns values to the P singleton, A singleton and the complete set, for example if $\mathbf{m}$ assigns these values

$$\begin{aligned}&\mathbf{m}(\phi) = 0, && \mathbf{m}(\{P\}) = 0.5 \\ &\mathbf{m}(\{A\}) = 0.2, && \mathbf{m}(\{P, A\}) = 0.3\end{aligned} \tag{10}$$

The corresponding vector to represent $\mathbf{m}$ is the following one:

$$\mathbf{m}: (0.5, 0.2, 0.3) \tag{11}$$

Methods to compute belief and plausibility for each outcome given a mass assignment function are implemented in the model using the formulas presented in 5 and 6. Note that for a single outcome the belief is just the mass of the singleton, therefore

Table 1
Summary of DST applications.

Work	Description	Limitations
Mulyani, Rahman, Riza et al. (2016)	The model combines Dempster-Shafer Theory and Naive Bayes classifier. They applied an expert system based on medical surveys to estimate the values of the mass assignment functions for several diseases from their symptoms and then the model predicts an outcome. If there is more than one outcome predicted by DST then the Naive Bayes classifier is applied.	In many cases, the model only uses the result of Naive Bayes without considering the knowledge of DST. In this work, we propose a new classifier without any dependency on other classifiers.
Denoeux (1995)	The author proposed a variation of KNN classifier which instead of using a neighbor's votes, neighbors support evidence for the corresponding class. Nearest neighbors support more certainty than the farthest ones. The decision is then made by the application of the Dempster Rule instead of using vote counting.	The method does not provide explanations in addition to the predicted classes. Our proposed model will cover interpretable classification.
Fixsen and Mahler (1997)	A modified version of the Dempster-Shafer theory that includes prior knowledge about the possible outcomes is proposed. This prior knowledge is built using the training data. Then the process of classification is explained using a custom distance function for the evidence and computations using DST and Dempster Rule.	The prior discovered knowledge can be considered interpretable. However, the custom distance function adds a complex operation that impedes further analysis on interpretability.
Denoeux (2000)	A model based on a modification of an RBF artificial neural network architecture which behaves similar to a Multilayer Perceptron. The model uses the weights of neuron links as evidence input for the Dempster-Shafer theory as well as the Dempster Rule for pooling.	Similar as before, the method does not perform interpretable classification since it is based on ANNs, which are black-box models.
Q. Chen, Whitbrook, Aickelin, and Roadknight (2014)	A classifier that examines the most important features. A mass is assigned according to the training values and finally the predicted class is obtained by the application of the combination rule. This classifier was tested in three traditional datasets and compared to other classifiers. The obtained accuracy is in most cases comparable to the other methods.	Although the method uses procedures to obtain relevant features, there is no direct discussion about interpretability.
Peñafiel et al. (2018)	A system based on Dempster-Shafer Theory to predict the risk of a patient having a heart attack or stroke based on past medical checkup data. The model is based on rules, which are statements that can be tested to be true according to data. Using these rules the system builds the mass assignment functions their values are established by defined formulas computed with the training data. The model gets a 61% accuracy when predicting stroke occurrences for patients within next year. An important output of this model is the identification of the rules that most contribute to a positive classification of a patient that has a stroke.	The drawback of this approach is that the model reaches low levels of accuracy. In the proposed model, this will be improved using gradient descent optimization techniques.

no computation is needed to obtain this value. The plausibility is computed just as the sum of the mass value of the singleton for the class (m_i) and the mass for the complete set (m_U), since our method does not consider subsets that are combinations of outcomes except for the complete set.

$$\begin{aligned} \forall a_i \in K \quad Bel_m(\{a_i\}) &= m_i \\ \forall a_i \in K \quad Pl_m(\{a_i\}) &= m_i + m_U \end{aligned} \tag{12}$$

In our example, the belief and plausibility for each class is $Bel_m(\{P\}) = 0.5,\ Bel_m(\{A\}) = 0.2,\ Pl_m(\{P\}) = 0.5 + 0.3 = 0.8$ and $Pl_m(\{A\}) = 0.2 + 0.3 = 0.5$.

A method for computing the Dempster rule between 2 mass assignment functions m_A and m_B is provided by the model using the Dempster Rule formula 7.

Note that, due to the representation of mass assignment functions, the sum of the Dempster Rule formula has at most $k+1$ elements. For each singleton this expression is the sum of each singleton of one MAF with the uncertainty of the other MAF, and the product between singleton masses. The result mass for the uncertainty of the combined MAF is simply the product of the uncertainty of each MAF. The formula is presented below:

$$\begin{aligned} \mathbf{m_A} &= (m_{A1}, m_{A2}, \ldots, m_{Ak}, m_{AU}) \\ \mathbf{m_B} &= (m_{B1}, m_{B2}, \ldots, m_{Bk}, m_{BU}) \\ \mathbf{m_A} \oplus \mathbf{m_B} &= K'(m_{A1}m_{B1} + m_{A1}m_{BU} + m_{AU}m_{B1}, \\ &\qquad \vdots \\ &\qquad m_{Ak}m_{Bk} + m_{Ak}m_{BU} + m_{AU}m_{Bk}, \\ &\qquad m_{AU}m_{BU}) \end{aligned} \tag{13}$$

Where K' is the normalization term in Dempster Rule. Note that K' can be computed afterwards as the reciprocal of the sum of the vector elements.

For example if we have $\mathbf{m_A}$: (0.5, 0.2, 0.3) and $\mathbf{m_B}$: (0, 0.3, 0.7) then the combination of these MAFs is the following:

$$\begin{aligned} \mathbf{m_A} \oplus \mathbf{m_B} = K(&0.5 * 0 + 0.5 * 0.7 + 0.3 * 0, \\ &0.2 * 0.3 + 0.2 * 0.7 + 0.3 * 0.3, \\ &0.3 * 0.7) \\ = K(&0.35,\ 0.29,\ 0.21) \\ = (&0.412, 0.341, 0.247) \end{aligned} \tag{14}$$

3.2. Rules and generators

Dempster-Shafer Theory is often applied in the design of expert systems because it is a good method to combine knowledge originated from various sources which may not be related to each other and even contradictory (for examples, from various stakeholders participating in a complex decision making process (Baloian, Frez, Pino, & Zurita, 2018)). These systems usually use rules or hypotheses to express conditions for the evidence of the problem. For example, in disease detection problems a rule could be "if a patient has high blood pressure then she/he is more likely to have a stroke"; in this case the condition is to have high blood pressure and the evidence is to be more likely to have a stroke.

Concerning the classification process for the model, in a general case a classifier tries to find the category to which an observation belongs. The observation encodes information in the form of a vector X, often called the feature vector. For example, in disease detection problems observations could contain age, blood pressure, past diseases, etc. Then, a rule can be mathematically defined as a pair $(\mathbf{m}, s)$ that relates a mass assignment function $\mathbf{m}$, with a predicate or boolean function (i.e. a function that evaluates to true or false) s with domain in the feature space.

Using this new representation, rules defined by experts are still possible to be defined and used in this context. However, the aim of this model is to perform automatic classification based on evidence presented by suitable data. Then rules should be generated by machine. In order to do it, an algorithm to generate rules from the available data is required; this method, called rule generator, creates rules using only training data and defined parameters.

The current model provides rule generators for single-attribute rules based on statistical ranges, i.e., for each continuous attribute in the feature vector, mean μ and standard deviation σ are computed for the attribute using the training data. Using these values the domain of the variable is partitioned in several ranges, e.g., if we set 3 ranges, these will be: values lower than $\mu - \sigma$, values between $\mu - \sigma$ and $\mu + \sigma$, and values greater than $\mu + \sigma$. For the case of categorical attributes it is possible to create a rule for each outcome of the attribute, i.e., if possible values are $k_1, k_2, \dots k_n$ then there will be a rule stating that the value is equal to k_1, another for values equal to k_2 and so on.

These rule generators define the conditions for the rules. However we need to set the initial values for MAFs as well. A possible strategy to set these values is to use prior knowledge about the problem. Again this knowledge can be obtained from experts or automatically, e.g., by using statistics measures of the training set (Peñafiel et al., 2018). Since these values will be optimized afterwards, another valid strategy is to consider that at the beginning we do not know anything about the problem, and then MAFs should have a high uncertainty. In fact we can set the value 1 for the complete set and 0 for the rest and then this assignment expresses full uncertainty. However, using this full uncertainty MAF as initial state is useless because the model cannot distinguish the classes (all of them have the same values) and this is required to perform the first classification. Alternatively, a better solution is to assign a high value for the complete set, e.g., 0.9 and the remaining 0.1 is distributed among the other singletons randomly. In our disease detection problem a MAF with values (0.04, 0.06, 0.9) could be a possible initial MAF for a rule.

3.3. Classification

The model operates using a rule set *RS* defining the knowledge, regardless of the source (experts or automatically), and the input feature vector x describing the features of the sample we want to classify. The model performs the following tasks to obtain the predicted class.

1. Apply the predicate of each rule in the rule set using the feature vector x as input. Filter out the rules that do not satisfy the predicate.
2. Combine the mass assignment function of the selected rules using Dempster Rule, obtaining a combined mass assignment function.
3. Compute the belief for each class from the combined mass assignment function. The predicted class will be the class with the maximum belief.

Mathematically if *RS* represents the rule set and x the input, the mass set for x, M_x, and the predicted outcome $\bar{y}$ can be defined as follows:

$$\begin{aligned} M_x &= \{\mathbf{m} \mid (\mathbf{m}, p) \in \mathrm{RS} \wedge p(x)\} \\ \mathbf{m_f} &= \bigoplus_{m \in M_x} \mathbf{m} \\ \bar{y} &= \underset{class}{argmax}\, \mathrm{Bel}(\mathbf{m_f}) \end{aligned} \tag{15}$$

3.4. Optimization via gradient descent

In order to get the best results for the model, it is necessary to fit the mass values of each rule according to the training data.

To do this task, a loss function and a method of optimization should be used to update mass values. Loss functions measure the error a model obtains in classifications comparing the predicted classes with the actual outcomes. During the optimization problem the aim of the model is to minimize the error computed by the loss function, and an algorithm of optimization is used to accomplish it. The existing literature reports on various optimization methods that can be applied depending on the nature and structure of the problem. In our case, we opted for the gradient descent as optimization method because it has been widely used successfully in other machine learning methods like artificial neural networks.

The model implements two loss functions Mean Squared Error (MSE) and Cross Entropy (CE) (De Boer, Kroese, Mannor, & Rubinstein, 2005). Both functions are normally used as loss functions in other classifiers. Also, both functions are differentiables, so they can be used as target functions in gradient based optimization.

The gradient descent algorithm is an iterative algorithm that optimizes variable values in order to minimize or maximize the value of a target function.

Mathematically, given the target function J which depends on the variable x_t and our goal is to minimize the value of J the updated value for x_t called x_{t+1} is given by the following formula:

$$x_{t+1} = x_t - \alpha \frac{\partial J}{\partial x} \tag{16}$$

Where the value α is called the learning rate. This algorithm gives us a sequence of values for $x_0, \dots, x_k$ that minimize J. Initial value for x (i.e x_0) is usually selected randomly.

In our case, the target function is any of the mentioned loss functions and the variables to be optimized are the mass values for each rule. The next formula shows the optimization for the mass $m^{(i)}$

$$m_{t+1}^{(i)} = m_t^{(i)} - \alpha \frac{\partial \mathrm{Loss}}{\partial m^{(i)}} \tag{17}$$

Besides normal gradient descent, the model supports any variation of gradient descent methods such as Stochastic Gradient Descent (SGD) and Adam (Kingma & Ba, 2014).

Recalling the restrictions of mass assignment functions from the Definition 4, the method of gradient descent presented above cannot be used because after updating the values, these may not satisfy the restrictions. For example, after updating, some mass values could become negative or the sum of the masses could exceed 1.

To solve this problem, it is important to notice that our optimization task is not unrestricted. For optimization with restriction an equivalent method to gradient descent is presented by Correa and Lemaréchal (1993), called projected gradient descent. It basically consists of the same idea of gradient descent, but projecting the values on the domain where variables are defined after updating. The same work proves that this method converges to the optimum for the restricted problem. Therefore, the correct formula for updating the mass values is the following:

$$m_{t+1}^{(i)} = \pi_C\left(m_t^{(i)} - \alpha \frac{\partial \mathrm{Loss}}{\partial m^{(i)}}\right) \tag{18}$$

Where π is the orthogonal projection function and C is the set of masses that satisfy Dempster-Shafer constraints.

The model processes the complete training set several times (epochs) in order to adjust correctly the mass values (Fig. 1). The next step is to define a condition to stop iteration, so we can state that the model has converged.

The condition used in the model consists of evaluating the difference between the loss in two adjacent epochs; if this difference in absolute value is smaller than a threshold ϵ then it is said that the model has converged and it stops iterating. Mathematically, it is defined by:

$$|Loss_t - Loss_{t-1}| \leq \epsilon \tag{19}$$

3.5. Interpretability

Let us recall that one of the main purposes of the classifier is to be interpretable. In this section we will explain why interpretability is a first class citizen of this model.

Rules, as we defined in the implementation, are the combination of a mass assignment function and a predicate or statement, after the training and optimization phase, mass values for each rule have changed, and they have converged to the optimal values for classification. From an interpretability point of view, that means that the model "learned" how much a rule statement is contributing to the classification and which outcome the statement predicts; mass values can be analyzed to distinguish informative rules from redundant rules.

For example, in the disease detection problem, if after training, a rule with predicate "if blood pressure is high", ends up with the following values using our representation $m = (0.7, 0.08, 0.22)$, that implies the mass is 0.7 for the P singleton, it is 0.08 for the A singleton and it is 0.22 for the complete set. Then, this rule contributes to predict that the disease is present, since mass of P singleton is much greater than the corresponding one for the A singleton. Also the mass of the complete set shows the uncertainty of the rule so the value 0.22 tells that there is not much uncertainty in this rule. We can then state that high blood pressure is related to the presence of the disease.

Therefore interpretability and knowledge discovery are directly extracted from the analysis of the mass values of a rule after training and its statement.

3.6. Model complexity

In addition to the algorithm of the proposed model, we will provide a basic analysis about its theoretical complexity.

Let X to be the set feature vectors with size n and m attributes for each vector, let k to be the number of classes and let RS to be the set of rules with length r.

In order to predict all feature vectors of X then n single predictions must be performed. For a single prediction, first we need to check which statements of all rules in RS should be applied, this can be done in $O(r)$. Then, Dempster Rule is applied to the selected rules, which have a complexity proportional to the length of mass vector, i.e. it is $O(k)$, therefore applying r times the Dempster Rule is $O(rk)$. Finally, as this process is repeated n times the final complexity for the prediction is $O(nrk)$.

The training process has an additional variable that is the number of epochs, let E be this number. In one epoch, we perform n single training processes, each of these single processes have a single prediction phase which is $O(rk)$ as described above. Then, the gradients are computed and the values are updated, this is proportional to the number of mass values to be optimized which is $O(rk)$. Finally, the masses are projected to satisfy the constraints which again is $O(rk)$. This training is repeated E times which implies the complexity for the whole training process is $O(Enrk)$.

If the model uses only single-attribute generated rules, the number of rules r is proportional to the number of attributes m, thus, prediction and training are $O(nmk)$ and $O(Enmk)$ respectively. This implies that the execution time of the model does not explode with the growth in the number of any of the parameters it uses.

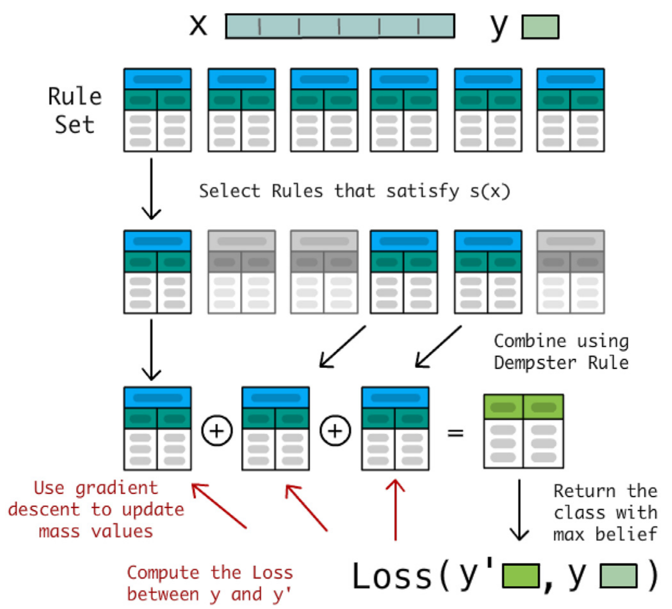
x
y
Rule Set
Select Rules that satisfy s(x)
Combine using Dempster Rule
⊕
⊕
=
Use gradient descent to update mass values
Return the class with max belief
Compute the Loss between y and y'
Loss(y' 🟩, y 🟩)

Fig. 1. Classification Process in DS Model. Starting from a feature vector x and a rule set, the model selects the rules that satisfy the predicate, the MAF of these rules are combined using Dempster Rule, and then the predicted class is the one with maximum belief. Finally the loss is computed using the real value y and the mass values are updated using gradient descent.

4. Results and discussion

We presented a new interpretable model for solving classification problem, so two aspects should be tested: the correct classification power and the interpretability. In this work three kinds of experiments will be performed increasing their complexity.

4.1. 2-D Distributions

The first experiment consists of testing the model using a 2-attribute fictional datasets as input. These datasets are generated by known functions or distributions, representing the simplest controlled scenarios for any classifier. The purpose of them is to present and visualize the first classifications and explanations of the model.

The first dataset (A1) contains 500 points which are random uniformly distributed in the rectangle $[-1, 1] \times [-1, 1]$ the class of each point is determined by the sign of the y component. A point that has $y < 0$ belongs to the "blue" class and the rest belong to the "red" class.

The second dataset (A2) contains 500 points which are generated by sampling two Gaussian distributions with mean in a random point within the rectangle $[-1, 1] \times [-1, 1]$ and an standard deviation of 0.25. The class of each point corresponds to which distribution generates it.

The model was tested on these two datasets using the following configuration: The model uses the single-attribute rules generator with 3 breaks, a learning rate of 0.002, the threshold of convergence is 0.0001, the loss function is MSE and the optimizer is Adam. For validation, the dataset is split in a training set (70%) and a testing set (30%) for each dataset the model was tested three times using different splittings and the average of the accuracy is presented.

Fig. 2 shows the results of the model. The model achieves an accuracy of 0.982 for dataset A1, and an accuracy of 0.987 for dataset A2.

The final optimized rules the model obtains for the case of datasets A1 and A2 are presented in Tables 2 and 3 respectively.

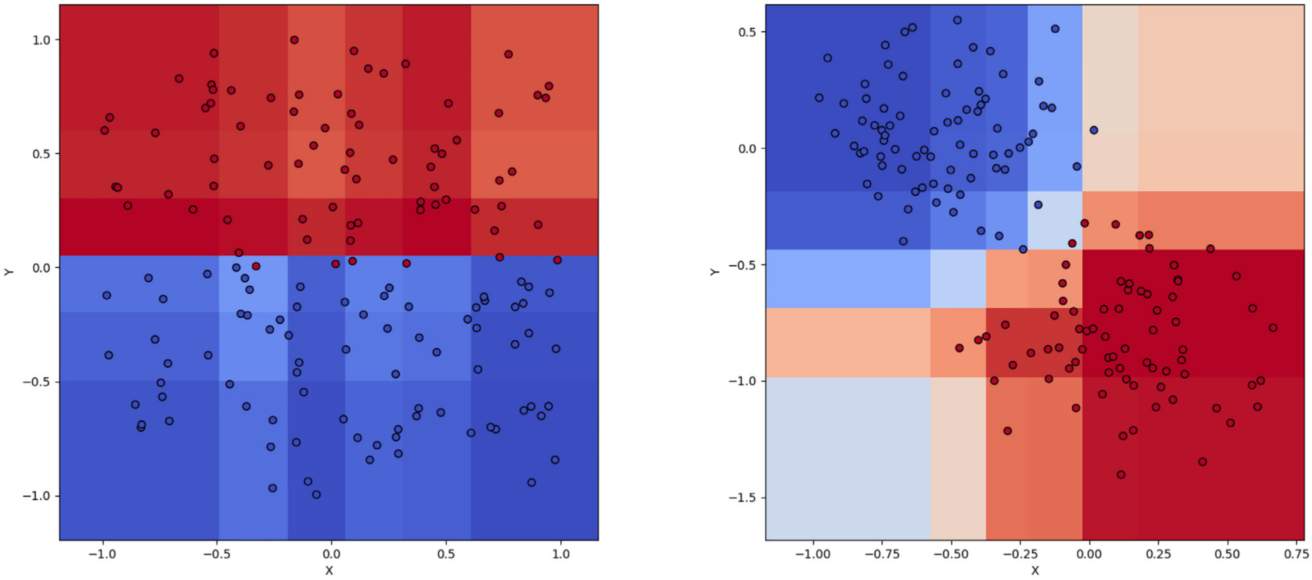

Fig. 2. Model results in controlled scenarios A1 (left) and A2 (right). Each dot is a dataset record, its color indicates the class, the background color is the predicted class of the model in that region, darker colors means more certainty in prediction.

Table 2
Resultant rules after model training for dataset A1.

Rule	Mass Blue	Mass Red	Uncertainty
$X \leq -0.32$	0.086	0.148	0.766
$-0.32 < X \leq 0.04$	0.136	0.077	0.787
$0.04 < X \leq 0.41$	0.000	0.000	1.000
$X > 0.41$	0.000	0.018	0.982
$Y \leq -0.34$	**0.662**	0.030	0.308
$-0.34 < Y \leq 0.04$	**0.529**	0.059	0.412
$0.04 < Y \leq 0.42$	0.000	**0.721**	0.279
$Y > 0.42$	0.000	**0.723**	0.277

Table 3
Resultant rules after model training for dataset A2.

Rule	Mass blue	Mass red	Uncertainty
$X \leq -0.43$	**0.674**	0.000	0.326
$-0.49 < X \leq -0.22$	**0.504**	0.000	0.496
$-0.22 < X \leq 0.04$	0.027	0.000	0.973
$X > 0.04$	0.000	**0.637**	0.363
$Y \leq -0.79$	0.000	**0.934**	0.066
$-0.79 < Y \leq -0.44$	0.000	**0.704**	0.296
$-0.44 < Y \leq -0.08$	0.180	0.141	0.680
$Y > -0.08$	**0.264**	0.072	0.665

Table 4
Results of controlled scenarios.

Experiment	Informative	Random	Redundant	Accuracy
B1	2	2	0	1.000
B2	2	0	2	1.000
B3	2	2	2	1.000

Results of Fig. 2 show that the proposed model can operate correctly as a classifier because the accuracy obtained was greater than 98% which means only few cases were classified wrong.

Evaluating the interpretability of the proposed model by the analysis of Tables 2 and 3, we found that the model was able to discover the most contributory rules correctly. For the case of dataset A1, all rules regarding X variable have a high value for uncertainty showing that they are not important for classification. On the other hand, rules regarding Y variable have much lower uncertainty, the rules which Y takes negative values assign high mass to blue outcome, while the ones that Y take positive values assign to red outcome which is correct according to the dataset A1 construction. For the case of dataset A2, the construction is more complex but the model can also distinguish the rules that contribute to the classification of each class correctly.

4.2. Multiple kinds of attributes

The second experiment will be similar to the previous one but we will use more attributes and they will have certain information. For example, if we build a dataset with four attributes, where only two of them are informative (i.e these two have a correlation with the instance class) and the other two attributes are random noise, the model should learn that the informative attributes are the most important to make classifications. Following this idea, another experiment is to have redundant attributes, i.e. attributes that are correlated with other attributes of the dataset, for example they could be linear combinations of them.

We present three experiments varying these kinds of attributes. Dataset B1 has 2 informative attributes and 2 random attributes; dataset B2 has 2 informative and 2 redundant attributes and dataset B3 has 2 of all kinds.

Table 4 presents the results for the model using the same configuration as in the previous case for the scenarios B1, B2, and B3. From the table is possible to note that in all cases the model achieves perfect accuracy.

The results of Table 4 show that the model can perform correct classifications when informative attributes are present in datasets. Also we showed that this feature is not affected by the presence of noisy or redundant attributes.

4.3. Traditional datasets

Besides the controlled scenarios mentioned above, the model has been tested in several traditional datasets. These datasets are the ones that appear in machine learning books and courses as example datasets to start using classification models, and they define common experiments to compare them.

Table 5
Results in traditional datasets.

Dataset	N. Attr.	Size	Classes	Rules	Acc. DSGD	Acc. RF	Acc. NB	Acc. KNN
Iris	4	150	3	28	0.959	0.943	0.971	0.971
Breast Cancer	9	700	2	108	0.962	0.947	0.820	0.608
Wine quality	13	6497	2	50	0.959	0.995	0.974	0.938
Heart Disease	9	462	2	34	0.727	0.667	0.727	0.673
Digits	64	1796	10	168	0.878	0.950	0.799	0.974
Gas Sensor	128	13,910	6	384	0.897	0.991	0.556	0.976

Specifically the datasets our model has tested are: Fischer Iris dataset (Fisher & Marshall, 1936), Wisconsin breast cancer, wine quality, heart disease and handwritten digits from UCI repository (Asuncion & Newman, 2007).

For each of the experiments performed in this section, our results will be compared with the results of the following classification algorithms: Random Forest (RF) with 100 trees, Naive Bayes (NB) and K-Nearest Neighbors (KNN) with $k = 5$.

The details of traditional datasets such as the number of attributes (N. Attr.), the number of records (Size), and the number of classes are presented in Table 5. This table also presents the number of rules our model uses and the accuracy (Acc.) on the test set for each model.

The application of the model to traditional datasets helps validate the previous results about accuracy. Results from Fig. 5 show that in most cases accuracy is over 85%. Comparing the results of the model with the other classification methods, our model reaches results similar to them. For the case of Breast Cancer and Heart Disease datasets, the model outperform the other models. In the other datasets tested the model has slightly lower accuracy. The difference of accuracy between our model to the best model is always lower than 10%. Another remarkable result is that the model performs well in multi-class classification tasks, from our experiments Iris and Digits are both multi-class datasets with 3 and 10 classes respectively, the metrics show that the model can also handle these kinds of data.

Gas drift array drifts dataset (Vergara et al., 2012) corresponds to a large dataset to test models, and it has 128 features and 13,910 records without missing values. The data comes from 16 chemical sensors exposed to 6 gases at different concentration levels. The goal is to predict the gas analyte according to the values of the sensors. The proposed model achieves an accuracy of 89.7%, which is a valid classification result. This result shows that our model can operate with large datasets and having good results.

Digits dataset presents an interesting case for the analysis of interpretability. This dataset contains handwritten digits in a 8x8 pixel box, the larger value means a mark is in the pixel, this information is flatten into a 64-length vector and passed as input. Although for the model this is like any other dataset, we can extract the importance of each rule and since rules are related to single attributes, we can know exactly which of these 64 pixels are contributory for the model to predict a digit and this is presented in Fig. 3.

The results of Fig. 3 shows that for the case of 0 digit darkest blue pixel, i.e. the ones that contribute most to the prediction of this class, are distributed in the extreme of the image and they tend to form a circle; for the case of 1 digit the darkest blue pixels tends to form a straight vertical line in the middle of the image. In both cases these results make sense with digit common drawings.

Breast Cancer dataset also presents an attractive case to look in detail. This dataset contains information about the cell nuclei of healthy (benign) and cancerous (malignant) cells retrieved from breast masses. This is the dataset that our model performs the best among the traditional datasets tested, achieving an accuracy of 96.2%. After training, the top 7 most important rules for the malignant class (i.e., the one that has higher mass in malignant singleton) are presented in Table 6.

In order to show the importance of having interpretability results, we can compare the rules obtained with the medical knowledge. From Table 6, one of the essential rules is the value of epithelial size, which appears in two of the most important rules. For this variable, higher values are associated with malignant cells. Checking this result with literature, Doyle et al. (2010) present a study about the differentiation of normal and malignant breast cells. They show that the malignant cells exhibit larger cell and epithelial nucleus sizes as compared to the healthy cells. This statement matches exactly the rules the model obtain from the interpretability, showing that it can verify this kind of knowledge.

As our proposed model is also rule-based, we can adapt the FRBS indicators for interpretability presented in Section 2.2 to our case. Q_{RANT} and Q_{ANT} do not need adaptation since we can count the number of antecedents directly from the statements. The indicator Q_{FS} needs to be re-interpreted to comply with our model. In FRBS n_{FS} refers to the numbers of active fuzzy sets using linguistic terms. We can note that fuzzy sets are similar to our definition of rule. Thus the most straightforward way to define n_{FS} in our case is to count the average number of rules an instance uses when the model performs a classification.

In order to test different configurations of our model, we can only use the top n rules that contribute most to each class (as explained in Section 3.5) for the classification, and all the other rules can be dropped. Applying this procedure, we can obtain a simpler model but still accurate. Table 7 shows the results for accuracy and interpretability measures for different configurations of the model. Fig. 4 shows the values of Q_{CPLX} and Error from the Table 7. Blue dots show configurations of the model. The red dot represents the more interpretable and accurate configuration, which corresponds to take the top 6 rules of each class. The dashed line represents all configurations that are equally accurate and interpretable than red dot (assuming that they have the same importance). This line is presented to compare with the other dots clearly.

The best configuration for the model was obtained when using the top 6 rules for each class. From the chart, we can observe that the model can drop many rules and still getting good accuracy, note that the best configuration has only the 27% of the rules of the original model. Comparing our results to the results presented by Gorzałczany and Rudziński (2017), we can see that our results for the best configuration are comparable. Our method has a slightly higher error of 0.0524 vs. 0.0365, but we have a slightly lower complexity of 0.0522 vs. 0.0568, thus better interpretability.

4.4. Stroke risk assessment

The model has also been tested in a real case scenario. Peñafiel et al. (2018) tested a “weaker” version of the proposed model that does not include optimization of parameters using gradient descent. The model (DS-Stat) only uses statistical indicators to estimate the values of the mass assignment functions.

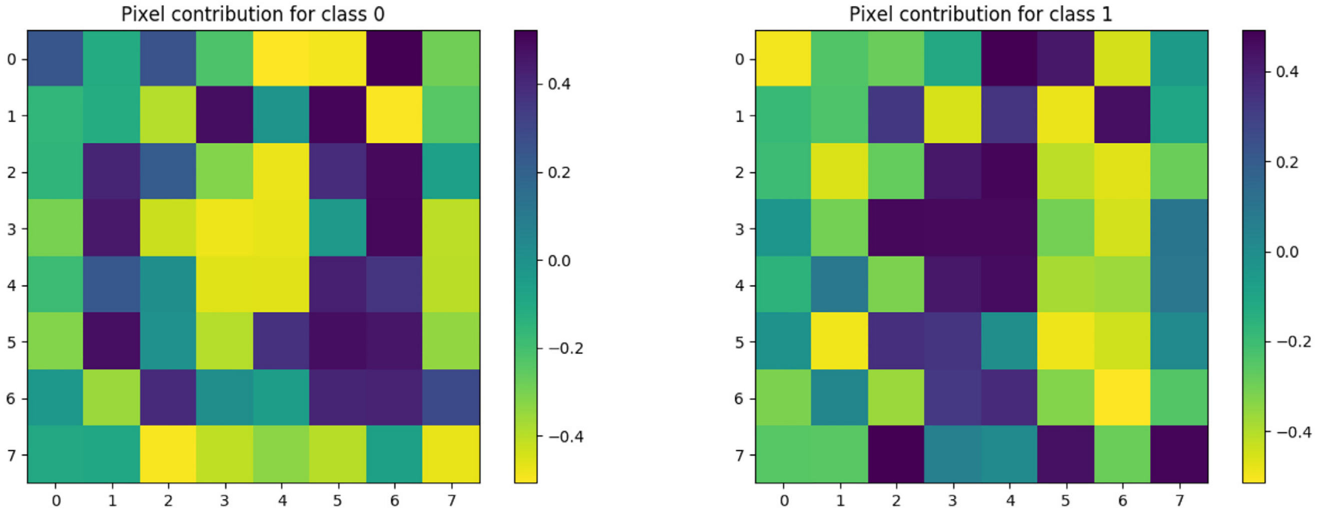

Fig. 3. Pixel importance for the classification of the classes 0 (left) and 1 (right) in the Digits dataset according to the interpretation of the rules.

Table 6
Most important rules for malignant class for breast cancer problem.

Rule	Mass benign	Mass malignant	Uncertainty
Clump thickness > 6.34	0.000	0.702	0.298
Bare nucleoli > 5.97	0.038	0.673	0.289
3.2 < epithelial size < 4.71	0.000	0.687	0.313
Size uniformity > 5.16	0.000	0.658	0.342
Marginal adhesion > 4.74	0.000	0.608	0.392
1.65 < mitoses < 2.84	0.002	0.597	0.401
Epithelial size > 4.71	0.017	0.580	0.403

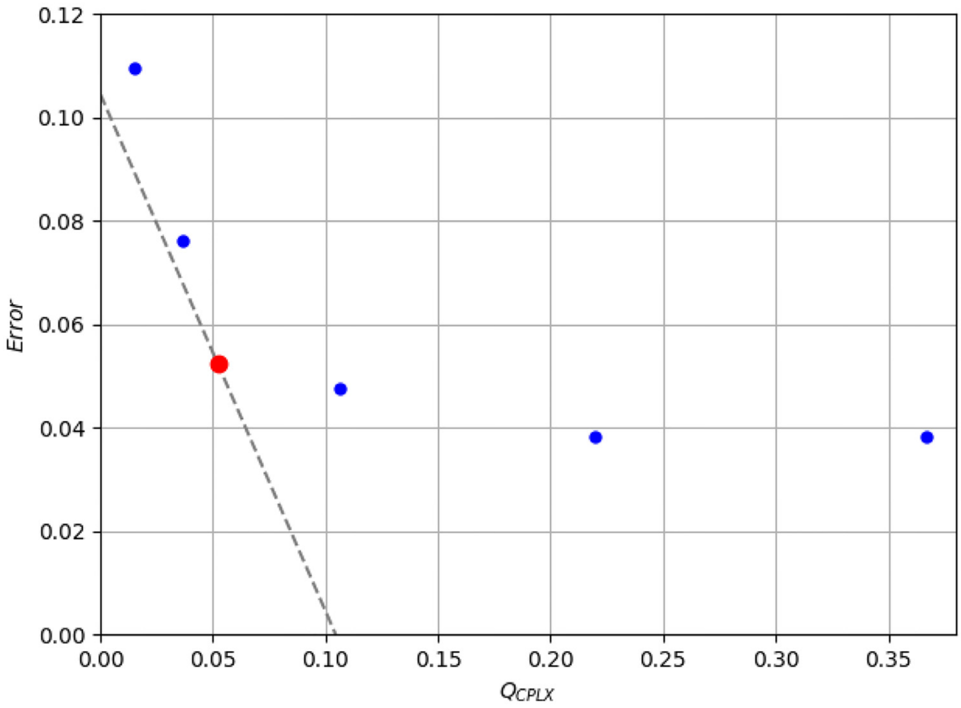

Fig. 4. Complexity and classification error measures for breast cancer problem.

Table 7
Interpretability and accuracy measures for Breast Cancer problem.

Top *n*	N. Rules	Q_{CPLX}	Accuracy	Error
all	45	0.3672	96.2%	0.0382
18	36	0.2195	96.2%	0.0382
9	18	0.1062	95.2%	0.0476
6	12	0.0522	94.8%	0.0524
3	6	0.0363	92.4%	0.0762
2	4	0.0155	89.1%	0.1095

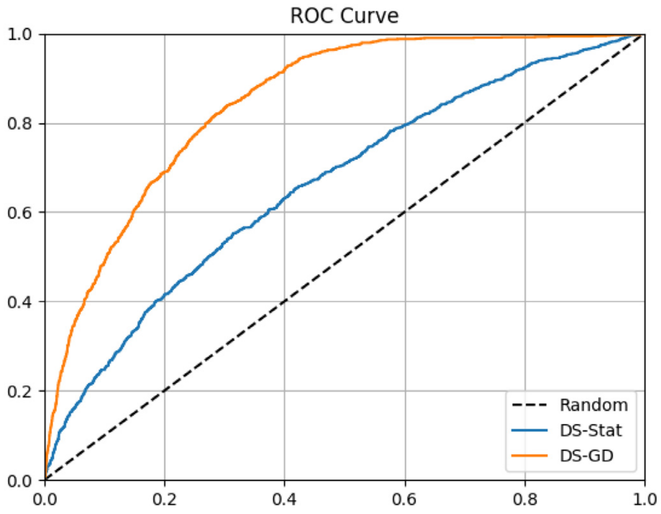

Fig. 5. Results for the stroke risk assessment problem.

The data comes from electronic health records (EHR) of the Hospital of Tsuyoyama, Japan. The data contains information about the patient demographics, disease history, and exam results. Unlike the previous dataset, this is a private dataset provided for limited research purposes which contains real information about patients between 2013 and 2016. The clinical events are stored, indicating the date when they were collected, patient information, and results. Diseases are encoded using the ICD10 codification standard (World Health Organization, 2001). The original data is a dump from the database of the hospital, there are more than 20 entities, and some tables have more than 100 attributes. The quality of the data is poor because it lacks normalization, and many of the attributes are completely missing, and others are redundant.

A pre-processing of this data was performed to have a better representation of this information. The patients who have disease history and exam results that have more than ten missing values (exams or diseases) are excluded. There is a total of 27,876 patients after applying these filters, for each patient there are 37 different features extracted.

The problem to be addressed is the detection of stroke within the next year based on the filtered EHR data of the patient for the previous year.

The original model showed acceptable results regarding the accuracy with an area under the ROC curve of 61.2%. For the case of interpretability, a similar procedure that the one described in Section 3.5 was applied to obtain the most contributory rules for the stroke classification, all of these rules were verified with medical studies.

The proposed model has also been tested in this scenario using the same data. This experiment helps to verify the behavior of the model in a real case scenario, and it also helps to measure and compare the improvement in classification by adding the gradient descent optimization. Fig. 5 shows a ROC curve for the result of both methods. From this result, we can see a significant increase in the area under the curve. The new model achieves 81.6% of the area under the ROC curve, which is an improvement of a 33.3% with respect to the previous model.

5. Conclusion

We proposed above a new classification model using the Dempster–Shafer Theory of Belief Functions. The model adapts this theory to apply it to expert systems, but using a novel approach to optimize values inspired by many machine learning techniques.

The proposed model proved to be a valid classifier; it was able to predict outcomes by adjusting rules from data in both controlled and traditional datasets. The results obtained in controlled scenarios were as expected, and in most cases the model performed perfect classifications.

We showed that in all the tested cases, the model cannot only accurately predict outcomes but also explain them. This feature is generally absent in most traditional classifiers.

One of the drawbacks of the current model is that data must be discretized which may result in less accuracy. This can be seen in the experiments using traditional datasets, whose obtained scores were acceptable, but slightly lower than current state-of-the-art classifiers.

As future work we aim to improve the proposed model in several ways. The most obvious improvement is to use all the subset to express uncertainty instead of just the complete set. Another important one is to get rid of attributes discretization by having a score indicating the degree of belonging to a certain rule; this score can be computed using the attribute value without any discretization. Another interesting possible improvement is using plausibility as the estimator for the probability of the classes instead of the belief; unlike belief, plausibility considers in its formula the uncertainty of components which means that these values will be optimized more precisely. Finally, another improvement, which is especially important when working with large datasets, is to drop rules that are not contributory to the prediction of a class while the model is being trained. These rules can be detected because they should have high uncertainty, and their gradients should be close to 0 in the initial iterations. This feature will help the model to be simpler, speeds up the training process, and prevent overfitting.

Functions that are declared to be part of the model can be of any type. In this work we only tested generated rules using statistical analysis. However, experts could state the rules according to their knowledge, which can be even used along with those generated statistically and the rest of the process is still the same, even the optimization of mass values. Urban crime prediction could be a possible scenario to test the incorporation of expert knowledge. This scenario is particularly interesting because, in a previous work we applied Dempster-Shafer and expert rules (Baloian et al., 2017) but without any optimization of the mass values nor making any interpretation of its results.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

Sergio Peñafiel: Investigation, Methodology, Formal analysis, Software, Validation, Visualization, Writing - original draft. **Nelson Baloian:** Investigation, Project administration, Supervision, Writing

- original draft. **Horacio Sanson:** Conceptualization, Data curation, Funding acquisition, Resources. **José A. Pino:** Supervision, Writing - review & editing.

Acknowledgment

Sergio Peñafiel was partially financed by Conicyt (Chile) scholarship No 22180506.

References

Asuncion, A., & Newman, D. J. (2007). *Uci machine learning repository*: 12. Irvine, ca: University of california. http://www.ics.uci.edu/mlearn/mlrepository.html.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS One, 10*(7), e0130140.

Baloian, N., Bassaletti, C. E., Fernández, M., Figueroa, O., Fuentes, P., Manasevich, R., … Vergara, M. (2017). Crime prediction using patterns and context. In *2017 IEEE 21st international conference on computer supported cooperative work in design (CSCWD)* (pp. 2–9). IEEE.

Baloian, N., Frez, J., Pino, J. A., & Zurita, G. (2018). Supporting collaborative preparation of emergency plans. In *Multidisciplinary digital publishing institute proceedings: 2* (p. 1254).

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1721–1730). ACM.

Casillas, J., Cordón, O., Herrera, F., & Magdalena, L. (2003). Interpretability improvements to find the balance interpretability-accuracy in fuzzy modeling: An overview. In Interpretability issues in fuzzy modeling (pp. 3–22). Springer.

Chen, Q., Whitbrook, A., Aickelin, U., & Roadknight, C. (2014). Data classification using the Dempster–Shafer method. *Journal of Experimental & Theoretical Artificial Intelligence, 26*(4), 493–517.

Correa, R., & Lemaréchal, C. (1993). Convergence of some algorithms for convex minimization. Mathematical Programming, *62*(1–3), 261–275.

Datta, A., Sen, S., & Zick, Y. (2016). Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *Security and privacy (sp), 2016 IEEE symposium on* (pp. 598–617). IEEE.

De Boer, P.-T., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of Operations Research, 134*(1), 19–67.

Denoeux, T. (1995). A k-nearest neighbor classification rule based on Dempster–Shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics, 25*(5), 804–813.

Denoeux, T. (2000). A neural network classifier based on Dempster–Shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 30*(2), 131–150.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, *29*(2–3), 103–130.

Doyle, T. E., Goodrich, J. B., Ambrose, B. J., Patel, H., Kwon, S., & Pearson, L. H. (2010). Ultrasonic differentiation of normal versus malignant breast epithelial cells in monolayer cultures. *The Journal of the Acoustical Society of America, 128*(5), EL229–EL235.

Fisher, R., & Marshall, M. (1936). Iris data set. *RA Fisher, UC Irvine Machine Learning Repository, 440*.

Fixsen, D., & Mahler, R. P. (1997). The modified dempster-shafer approach to classification. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 27*(1), 96–104.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. Machine learning, *29*(2–3), 131–163.

Gal, Y. (2016). Uncertainty in deep learning. *University of Cambridge*.

García, S., Fernández, A., Luengo, J., & Herrera, F. (2009). A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability. Soft Computing, *13*(10), 959.

Gorzałczany, M. B., & Rudziński, F. (2017). Interpretable and accurate medical data classification–a multi-objective genetic-fuzzy optimization approach. *Expert Systems with Applications, 71*, 26–39.

Ishibuchi, H., & Nojima, Y. (2007). Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. *International Journal of Approximate Reasoning, 44*(1), 4–31.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science, 349*(6245), 255–260.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv:1412.6980.

Lipovetsky, S., & Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry, 17*(4), 319–330.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (pp. 4765–4774).

Mulyani, Y., Rahman, E. F., Riza, L. S., et al. (2016). A new approach on prediction of fever disease by using a combination of dempster shafer and naïve bayes. In *Science in information technology (ICSITECH), 2016 2nd international conference on* (pp. 367–371). IEEE.

Olden, J. D., & Jackson, D. A. (2002). Illuminating the ǣblack boxg: A randomization approach for understanding variable contributions in artificial neural networks. Ecological Modelling, *154*(1–2), 135–150.

Olshen, R., & Stone, C. (1984). Classification and regression trees. *Belmont, CA: The Wadsworth and Brook*.

Peñafiel, S., Baloian, N., Pino, J. A., Quinteros, J., Riquelme, Á., Sanson, H., & Teoh, D. (2018). Associating risks of getting strokes with data from health checkup records using dempster-shafer theory. In *Advanced communication technology (ICACT), 2018 20th international conference on* (pp. 239–246). IEEE.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016a). Model-agnostic interpretability of machine learning. arXiv:1606.05386.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016b). Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144). ACM.

Shafer, G. (2016). Dempster'S rule of combination. International Journal of Approximate Reasoning, *79*, 26–40.

Shafer, G., et al. (1976). *A mathematical theory of evidence*: 1. Princeton university press Princeton.

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. arXiv:1704.02685.

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems, 41*(3), 647–665.

Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*.

Vergara, A., Vembu, S., Ayhan, T., Ryan, M. A., Homer, M. L., & Huerta, R. (2012). Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical, 166*, 320–329.

World Health Organization (2001). *International classification of functioning, disability and health: ICF.*. World Health Organization.