A Study on Interpretable Machine Learning and Applications

Nguyen Thanh Phu

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

A Study on Interpretable Machine Learning and Applications

Nguyen Thanh Phu

Supervisor: Professor Huynh Van Nam

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology

Knowledge Science

March, 2021

Acknowledgment

Firstly, I would like to express my profound gratitude towards my supervisor Professor Huynh Van Nam. His professional advice and guidance have given me the confidence to go further and explore my potential whenever I have to face with difficulties in my academic and daily life. He has passed for me not just his experience and knowledge but also his passion in pursuing academic research career path.

I would like to thank my second supervisor Professor Takashi Hashimoto, my advisor for minor research project Professor Minh Le Nguyen and the committee members for spending their time to discuss and give me valuable comments and suggestions on my research. I would not have finished my work without their great support.

I want to thank my parents, my elder sister's family and my girlfriend for providing me with continuous encouragement and support throughout my Doctoral program. This accomplishment would not have been possible without their supports.

I would also like to give thanks to all of my friends, especially my labmates. Their enthusiastic support and friendship have helped me through hard times when studying overseas.

Finally, I wish to express my appreciation for the financial supports from the MEXT scholarship, JAIST Research Grant and The US Office of Naval Research Global under Grant No. N62909-19-1-2031 during my Doctoral program.

Nguyen Thanh Phu
November, 2020

Contents

Acknowledgment **i**

1 Introduction **1**
1.1 Interpretable Machine Learning and XAI 1
1.2 Research Objectives 3
1.3 Dissertation Structure 3

2 Literature Review **5**
2.1 A Taxonomy of Interpretable ML Approaches 5
2.1.1 Traditional Interpretable Models 6
2.1.2 Causal Interpretable Models 20
2.2 Evaluation for Interpretability 24
2.2.1 An Explanation Evaluation Taxonomy 24
2.2.2 The Predictive Accuracy - Descriptive Accuracy - Relevancy (PDR) Framework 26
2.2.3 A Unified Hierarchical Framework for Explanation Evaluation 28
2.3 Applications of Interpretable ML and XAI 30

3 Improving The Efficiency of Interpretable Unsupervised Learning 36
3.1 Background 36
3.2 The Proposed Clustering Framework - *RICS* 37
3.2.1 Notations 38
3.2.2 A k-Means Like Clustering Framework 38
3.2.3 A Context-Based Dissimilarity Measure for Categorical Data . 41
3.3 Experimental Evaluation 42
3.3.1 Testing Data Sets 43
3.3.2 Experimental Results 44
3.4 Summary 45

4 Transparent Supervised Learning Instead of Black-Box Models 46
4.1 Background 47
4.2 A Transparent Classification System for Knowledge Discovery 49
4.2.1 Notations 49
4.2.2 GSOM-based Interpretable Classification System (GSIC) . . . 49

4.3 Experimental Evaluation 53
4.3.1 Testing Data Sets 53
4.3.2 Experimental Setups and Final Results 54
4.3.3 Interpretability Analysis for ICU Sepsis Use Case 56
4.4 Summary 61

5 Enhancing Supervised Learning with Uncertainty Management 62
5.1 Introduction of Uncertainty in Machine Learning 63
5.2 Evidence Theory 64
5.3 IEBC (Inner Evidence-Based Classifier) 66
5.3.1 Notations 66
5.3.2 The Proposed Classification System 66
5.4 Experimental Evaluation 71
5.4.1 Testing Data Sets 71
5.4.2 Experimental Setups and Final Results 71
5.5 Summary 74

6 General Discussion 75

7 Conclusion 80

Publications 82

References 83

List of Figures

2.1 A taxonomy of interpretable ML approaches. 6
2.2 Black-box model vs. transparent model [21]. 7
2.3 An example of a decision tree with positive and negative class (binary) and three attributes. The red path represents for a decision rule which leads to the positive class if the testing instances satisfy the tree's conditions [14]. 8
2.4 The typical process for conducting a cohort study [28]. 10
2.5 An example of a retrospective cohort study conducted on neurosurgery field [29]. 10
2.6 The results of kNN classification on Gaussian-based data with two neighborhood sizes ((a) $k = 1$ and (b) $k = 20$). With the value of $k = 1$, kNN has a tendency to overfit the problem, while with a larger k, the algorithm will ignore small patterns of data concentration [31]. 12
2.7 The GSOM for the zoo dataset [32]. 13
2.8 An example of LIME for explaining a single prediction. Given a prediction provided by a trained model that a patient has the flu, LIME will emphasize the symptoms in the patient's history that led to the prediction. Specifically, sneeze and headache contribute to the "flu" prediction, while "no fatigue" is evidence against it [35]. 15
2.9 An illustration of LIME's intuition. The blue and pink background represent for the decision function f of a black-box model which is unknown to LIME. While the bold red cross sign is the decision made by f that need to be explained. LIME will generate the explanation by sampling its neighboring instances and weighting the results by their *"distance"* to the explained instance (represented by their size in the figure). The dashed line is the learned explanation that is locally (but not globally) faithful [35]. 16
2.10 Explanations for the risk of hypoxaemia in the next five minutes during surgery [37]. 17
2.11 The saliency maps for images in the top-1 predicted class that belong to ILSVRC-2013 dataset [38]. 19
2.12 The representation of a feed-forward neural network as an SCM [40]. 21

2.13 An example of generating counterfactual visual explanations for a query image I. It explains the reason that I was classified as class c (Crested Auklet) instead of c' (Red Faced Cormorant) by finding a region in a distractor image I' and a region in I (red boxes) so that if exchanging the highlighted region in both images then the resulting image $I^\star$ would be classified more confidently as c' [41]. 23
2.14 Taxonomy of evaluation approaches for interpretable models [15]. . . 24
2.15 Different stages of an interpretable ML process in the data-science life cycle. 27
2.16 Impact of interpretation methods on descriptive and predictive accuracy [4]. 28
2.17 A unified hierarchical framework for evaluating explanations in interpretable ML [42]. 29
2.18 Example of a decision list for estimating 1-year stroke risk following diagnosis of atrial fibrillation from patient medical history [43]. 30
2.19 The overall scheme of QSAR [48]. 32
2.20 An illustration of a positive and negative counterfactual explanation. For the positive counterfactual explanation (a), it show the amount of tolerances (highlighted in yellow). While the negative counterfactual explanation (b) provides reasons for the loan rejection with suggestions on how to improve the results with the increase (green, dashed) or decrease (red, striped) of each feature [51]. 35

4.1 Illustration for the general structure of *GSIC* 50
4.2 The 2D map generated by GSOM for the ICU sepsis data set. 59
4.3 K-means clustering on generated GSOM 60
4.4 Representative rules for each sub-cohort. 60

5.1 The overall process of the proposed classification system *IEBC* 67

List of Tables

3.1 Details of data sets for the experiment that are collected from UCI . 43
3.2 The purity results of the clustering experiment on 14 testing data sets 44
3.3 The NMI results of the clustering experiment on 14 testing data sets 45
3.4 The ARI results of the clustering experiment on 14 testing data sets . 45

4.1 Details of data sets collected from UCI and MIMIC III 54
4.2 AUC of classification results of 8 datasets from UCI and MIMIC III . 55
4.3 Summarized information of ICU sepsis data set [74]. 57
4.4 Statistics for sepsis criteria with patients and mortality rate [74]. . . . 57
4.5 Comparison of decision rules generated by GSIC and CART 58
4.6 Inference rules of each cluster for predicting mortality cases 59

5.1 Intersection of pieces of evidence from two evidence sources 70
5.2 Characteristics of 10 datasets collected from UCI and MIMIC III . . 73
5.3 AUC of classification results of 10 testing datasets - part 1 73
5.4 AUC of classification results of 10 testing datasets - part 2 73

Chapter 1

Introduction

Nowadays, applications that implement machine learning and data mining techniques are ubiquitous in all aspects of everyday life. In several tasks such as classification or recognition, those techniques have been proved to be able to perform equally and even surpass humans. However, models that can achieve prominent performance are normally complex, opaque and have low interpretability. It is a non-trivial task to explain the underlying behaviors of those models and the reason for their final outcomes. In many domains that require to make high-stakes decisions such as healthcare, medicine or finance, interpretability is considered as one of the most important factors for the adoption of those machine learning models.

1.1 Interpretable Machine Learning and XAI

One of the main reasons for the popularity of applications that implement machine learning techniques is a significant improvement in their performance. Recently, advancements in deep learning techniques have brought applications of machine learning back to life with robust performance and more real-life experience such as image recognition or automatic driving car. Specifically, with the abundance of data, deep learning techniques could achieve prominent accuracy due to their capability of capturing complicated relationships hidden in the data. However, such high accuracy comes with high complexity and opaqueness in the models [1].

It is challenging to explain for the results of those so-called *black-box* models - which is one of the essential requirements when employing them within a decision support system, especially in domains that need the interpretability for making high-stakes decisions [2]. In recent years, there is the raising of *eXplanatory Artificial Intelligence (XAI)* and *Interpretable Machine Learning (Interpretable ML)*

fields which aim to resolve the mentioned problem. Specifically, XAI promotes transparency in whole or parts of systems and the explainability for their decisions. In the work of [3], the explanations for decisions made by algorithms are claimed to be vital to guarantee fairness, detect potential algorithmic or data bias and to make sure that the algorithms perform as expected. In machine learning domain, the two terms explainability and interpretability have different meaning but are frequently used interchangeably. In the remaining parts of this dissertation, we also do not differentiate the meaning of those two terms when using them.

In the work of [4], interpretable ML is considered as the utilization of ML models in order to exploit relevant domain knowledge about hidden relationships in available data. Insights that are formed by that relevant knowledge can be used to guide communication, actions and further discovery. On the other hand, the interpretability of ML models was also mentioned as the first step to assure the explainability of the models which *"must be complete with the capacity to defend their actions, provide relevant responses to questions and be audited"* [3].

In real applications, there are two main situations where interpretability is required as an essential property of a system. Firstly, it is necessary for troubleshooting when a problem occurs. As observed in several ML applications, intentionally hidden information or unforeseen behaviors may be embedded into the models [3]. For example, a deep neural network can misclassify an image while applying a certain perturbation on the same image [5]. In the other work of [6], deep neural networks can be tricked to misclassify inputs with no resemblance to the true category. Neural networks that process natural languages can also occur similar problems [7]. With the increase of research on adversarial examples of ML models, there is more emphasis placed on the interpretability and explainability of ML models in order to promote the understanding of their decision-making process as well as problem detection for suspected cases.

Secondly, in the fields that require to make high-stakes decisions such as medicine, healthcare or finance, interpretability is considered as one of the main requirements for the adoption of ML systems for data analysis [8]. In the work of [9], it was mentioned that the clinical decision related to radiation treatment should not be based merely on the accuracy of a prediction system but also on an informed understanding of the relationship among patients' characteristics, radiation response and treatment plans. Moreover, it is also challenging for the application of neural networks in predicting medical outcomes when comparing with the use of interpretable methods such as logistic regression [10]. In a review in the applications of deep neural networks for health informatics [11], the lack of interpretability is pointed out as the main reason that limits the adaptation of deep neural networks into the healthcare section. Furthermore, in [12], the need for opening ML black boxes is posed as one of the biggest challenges for ML applications in the medical field.

Besides the two aforementioned situations, with the widespread of everyday life applications that implement ML techniques, the global community is becoming more aware of their impact and the related issues. Social awareness is reflected in the implementation of legal documents such as the European Union directive for General Data Protection Regulation (GDPR). GDPR defined the right of explanation as providing an individual with *"meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject"* [13]. Furthermore, several GDPR-like laws are being adopted in many countries such as the California Consumer Privacy Act (CCPA) or the Privacy Amendment (Notifiable Data Breaches) to Australia's Privacy Act. Although the right of explanation is not clearly declared in those documents, it is still a big step toward the full recognition and implementation of this right in the legit documents.

1.2 Research Objectives

The main objectives of this research are firstly improving the robustness of explainable ML models. Specifically, there is a general observation that explainable models tend to have low performance when dealing with complex problems. On the other hand, more complicated methods can achieve significantly higher accuracy but have lower interpretability. As an effort to mitigate this problem, we conduct an investigation on a common explainable model for the clustering task and propose a new clustering method with higher performance while still do not notably increase its complexity.

The second objective of our research is instead of improving an explainable model that may underfit the problem, we focus on a systematic combination of explainable supervised and unsupervised ML methods to form a new framework that can model complex problems while still preserve their interpretability.

Finally, a refinement is added to our previous proposed system with the introduction of uncertainty management. A demonstration is given on how uncertainty management can help to improve the effectiveness of the system as well as provide more explanations and new knowledge about the underlying data to users.

1.3 Dissertation Structure

The dissertation contains seven chapters. The summarized content of each chapter is described as below:

– In the first chapter, we briefly introduce the concepts of interpretable ML as

well as XAI and give the motivation of our research about why it is necessary for a system to be explainable. Then we lead directly into the research problem and our research objectives.

– In chapter 2, we conduct a review of the related literature of our research. Particularly, a survey on new trends of XAI and interpretable ML is conducted to clarify current approaches for achieving an explainable ML model. After that several different methods to evaluate the interpretability of ML models will be mentioned. Finally, in this chapter, we introduce applications of interpretable ML in a broad range of areas including healthcare, material discovery and banking sector.

– In chapter 3, we conduct an investigation on a common interpretable method for the clustering task. Specifically, a brief introduction on the clustering method will be provided and its working mechanism is analyzed for better understanding as well as improvement. After that, we describe in detail our newly proposed clustering method which is based on the common one with several major improvements. In other to prove the merit of our proposed method, an experiment will be conducted and demonstrated in this chapter.

– In chapter 4, a discussion on the use of transparent learning methods over black-box models is given. Moreover, we introduce a new transparent classification framework based on a systematic combination of supervised and unsupervised methods. An experimental evaluation is given with a wide range of real data in healthcare field as well as general data. Furthermore, an analysis on the interpretability of the proposed system is also provided.

– In chapter 5, a refinement is added to the proposed system mentioned in chapter 4 with uncertainty management. Firstly, we cover some background concepts of the evidence theory and fuzzy clustering in this chapter. Then details on our proposed classification system which utilized the above concepts will be described. Finally, an in-depth comparative experiment is conducted to evaluate our proposed methods.

– In chapter 6, we provide a general discussion on the remained limitations as well as problems in our work and in the research community related to the field of interpretable ML and XAI.

– Finally, in chapter 7, we summarize and draw conclusions on our research and mention about our future work.

Chapter 2

Literature Review

In this chapter, we would like to provide a big picture of recent work on *interpretable Machine Learning (ML)* and *eXplainable Artificial Intelligence (XAI)* fields including a taxonomy of current approaches on attempts to achieve explainable systems. Moreover, due to the lack of a consensus definition of interpretability, we further elaborate on several different methods for evaluating the quality of the explanations and the interpretability of those systems. Finally, we introduce a number of applications of interpretable ML in critical decision-making and explainability-required fields such as healthcare, credit scoring in the banking system and material discovery.

2.1 A Taxonomy of Interpretable ML Approaches

Currently, there are existing several different ways to classify interpretable ML techniques which are commonly based on their ability to provide comprehensive explanations and the types of those explanations. Specifically, interpretable ML techniques are generally divided into two main branches: *intrinsically interpretable methods* and *post-hoc models* [14, 4, 15, 16, 17, 2, 18]. Intrinsically interpretable (or transparent) methods are designed so that the self-explanatory capability is incorporated into the structure of those models [16] and ready to provide insights into the relationships they have learned from the data [4]. On the other hand, post-hoc models are defined as a second model to provide the explanations for existing (usually black-box) models [16].

However, recently in the work of [14], they noted a newly rising approach for building explainable models with *causal interpretability*, and named the aforementioned approaches as the traditional ones. According to [14], traditional approaches provide merely statistical interpretability while causal interpretability aims at an-

swering causal relationship-related questions such as *"What if"* (causal interventional interpretability) and *"Why"* (counterfactual interpretability). In the taxonomy that is presented in this section, we would like to consider the idea of organizing current approaches as traditional and causal ones as suggested in [14]. Moreover, due to the gap and several disparities in the classification of interpretable ML approaches, we would like to summarize and propose a more consistent and coherent taxonomy for interpretable ML techniques as described in Figure 2.1. Details about this taxonomy will be presented in the following subsections.

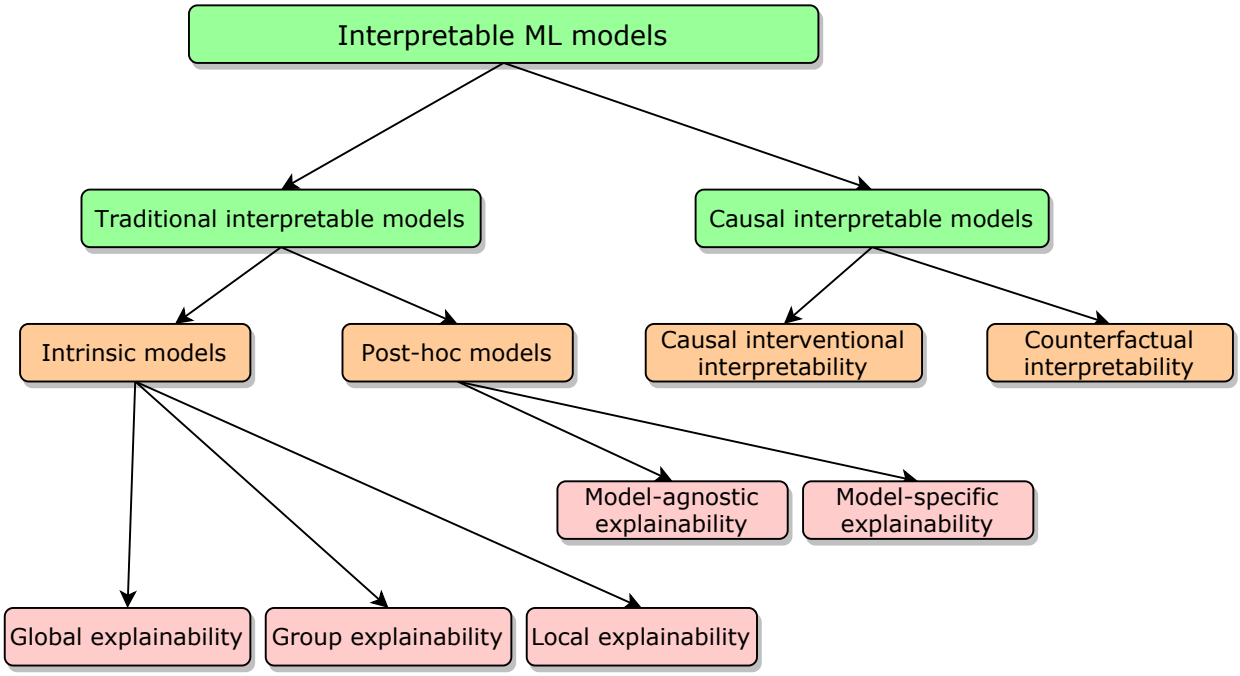


Figure 2.1: A taxonomy of interpretable ML approaches.

2.1.1 Traditional Interpretable Models

Traditional interpretable ML techniques can generally be grouped into two categories: intrinsically interpretable models and post-hoc models, depending on the time when the interpretability is obtained [14]. It is a common distinction that can be found in recent XAI related surveys [18, 17] that based on the models' aim of providing explanations. Specifically, *intrinsically interpretable models* are designed in order that insights into the decision-making process as well as knowledge on the underlying data can be incorporated directly into the models [4]. Several common examples of intrinsically interpretable models can be given such as decision trees [19], association rule learning [20] or linear regression.

In contrast, *post-hoc models* are proposed for generating explanations for a second

model (usually a black-box model) [16]. Recently, deep neural networks such as *CNN (Convolutional Neural Network)* or *RNN (Recurrent Neural Network)* become ubiquitous in a wide area of applications because of their prominent performances. However, those models are considered as black-boxes because it is hard to track their decision-making processes as well as understand what they have really learned from the training data as described in Figure 2.2. Because of those reasons, deep neural networks have become one of the main targets for post-hoc models which aim to extract knowledge from the trained networks as well as provide explanations for their decisions.

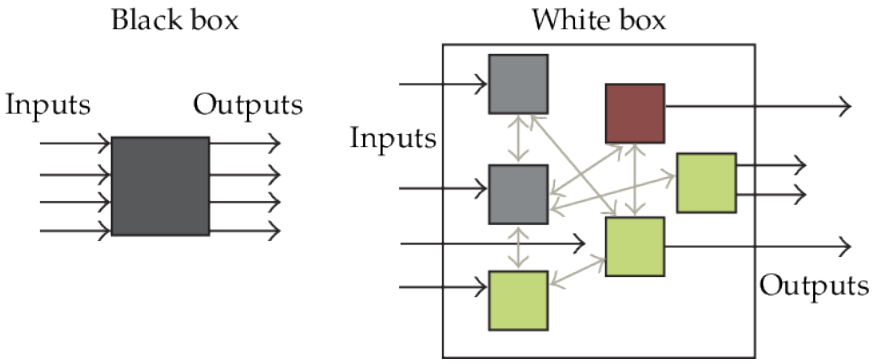


Figure 2.2: Black-box model vs. transparent model [21].

A. Intrinsically Interpretable Models

Intrinsically interpretable (or transparent) models are approaches that incorporate the interpretability directly into the model structures [16]. Based on the characteristics of their structure, the proposed models can provide users with insights into their decision-making processes as well as useful knowledge about the underlying data. Based on the types of their explainability given by the intrinsic models, they are further divided into three categories: *global explainability*, *group-based explainability* and *local explainability* [16, 22, 23]. Details of those categories will be described in the following parts.

Global Explainability

Globally interpretable models are transparent about their working mechanism and decision-making processes [16]. In other words, the models can provide explanations about their overall behavior [24]. There are several models that are deemed to be globally interpretable such as decision trees, rule-based models or linear regression. Moreover, globally interpretable models can also be constructed on the foundation of the aforementioned models.

Besides, according to [16], models can provide global interpretability by promot-

ing to incorporate interpretability constraints such as enforcing sparsity terms or imposing semantic monotonicity constraints in classification models [25]. Similarly, decision trees are pruned by replacing subtrees with leaves to encourage long and deep trees rather than wide and more balanced trees [26]. Trained ML models can be simplified by applying those constraints, moreover it could also improve the comprehensibility of the models. In this part, we briefly review two popular globally interpretable methods including decision trees [19], and linear regression.

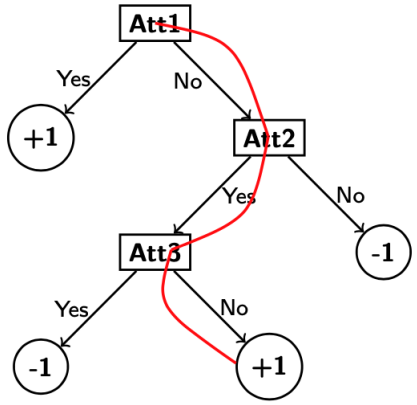


Figure 2.3: An example of a decision tree with positive and negative class (binary) and three attributes. The red path represents for a decision rule which leads to the positive class if the testing instances satisfy the tree's conditions [14].

Decision Trees is a popular interpretable classification method that was proposed by [19]. Among several variations of this method, CART (Classification and Regression Tree) algorithm is one of the most well-known binary decision tree learning algorithms proposed by [27]. With the input is a set of data instances, those instances will then be split based on the feature that has the largest information gain (IG). For splitting the nodes at the most informative features, the objective function will be maximized for the information gain at each split as defined by the following formula.

$$IG(D_p, att) = I(D_p) - \left[\frac{N_{left}}{N_p} I(D_{left}) + \frac{N_{right}}{N_p} I(D_{right})\right]$$

Where att is the feature to perform the split, D_p and D_{left}, D_{right} are the dataset of the parent, left and right child nodes respectively, I is the impurity measure, N_p is the total number of samples at the parent node, and N_{left}, N_{right} are the number of samples in the left and right child node. The information gain provides the distinctive amount between the impurity of the parent node and the sum of the child node impurities—the lower the impurity of the child nodes, the larger the information gain.

Specifically, in the CART algorithm, the impurity measure is implemented as the Gini index (GI). Basically, the Gini index aims to minimize the probability of misclassification as defined in the Eq. (2.1) where $p(i|t)$ is the proportion of the samples that belong to class c for a particular node t.

$$I_{GI}(t) = \sum_{i=1}^{c} p(i|t)(-p(i|t)) = 1 - \sum_{i=1}^{c} p(i|t)^2 \tag{2.1}$$

The splitting procedure will be conducted at each child node iteratively until the leaves are pure. This means that the samples at each node all belong to the same class. An example of a decision tree's result is illustrated in Figure 2.3.

Linear Regression is another method regarded as being interpretable. The linear regression captures linear relationships between a dependent variable (target) and independent variables (features). The weight of each feature represents the mean change in the prediction given a one-unit increase of the feature. Accordingly, it can be interpreted as the features with larger weights has more effect on the final result. Specifically, the target value can be expressed as a linear combination of the features with their first value randomly initialized weights:

$$y = w_0 + w_1 a_1 + w_2 a_2 + ... + w_k a_k$$

where y is the target (class); $a_1, a_2, ..., a_k$ are the attribute values; and $w_0, w_1, ..., w_k$ are the weights assigned to each attribute correspondingly.

The final values of the weights are optimized from the training data. In detail, the linear regression method optimizes a set of $k+1$ coefficients w_j by minimizing the sum of the squares of these differences over all the training instances. Given n training instances; denote the ith one with a superscript i. Then the sum of the squares of the differences can be defined as

$$\sum_{i=1}^{n} \left(y^i - \sum_{j=1}^{k} w_j a_j^i \right)^2$$

where the expression inside the parentheses is the difference between the ith instance's actual class and its predicted class. This sum of squares is what we have to minimize by choosing the coefficients appropriately.

Group-Based Explainability

The concept of group-based explainability was mentioned in the work of [23]. In this work, they discussed the application of interpretable ML in endorsing to improve

healthcare quality where cohort inspection is a common practice that can help experts in the field to understand the pathology of specific groups of patients. From this information, doctors can provide better effective treatment.

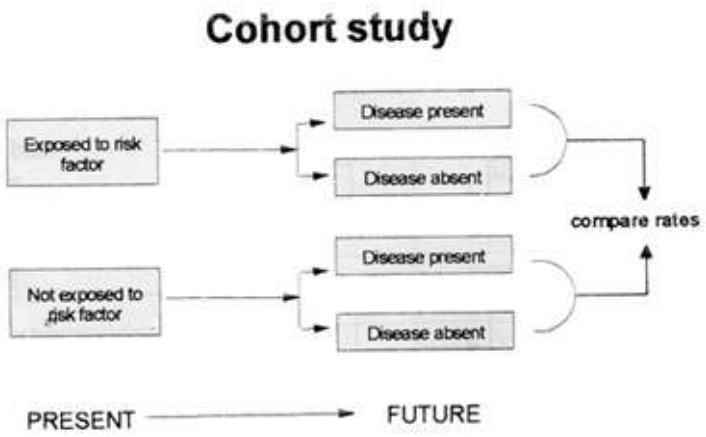


Figure 2.4: The typical process for conducting a cohort study [28].

Basically, in epidemiology, a cohort is a group of people who share a common experience, condition or characteristic. And cohort study aims to measure and compare the incidence of disease in two or more study cohorts [28]. In medical research, cohort studies are very strong and popular designs, however, they are also very time consuming and expensive. Currently, due to the availability of medical data (especially patients' medical records), a retrospective cohort study can be conducted in a less expensive and faster way with the help of machine learning and data mining techniques.

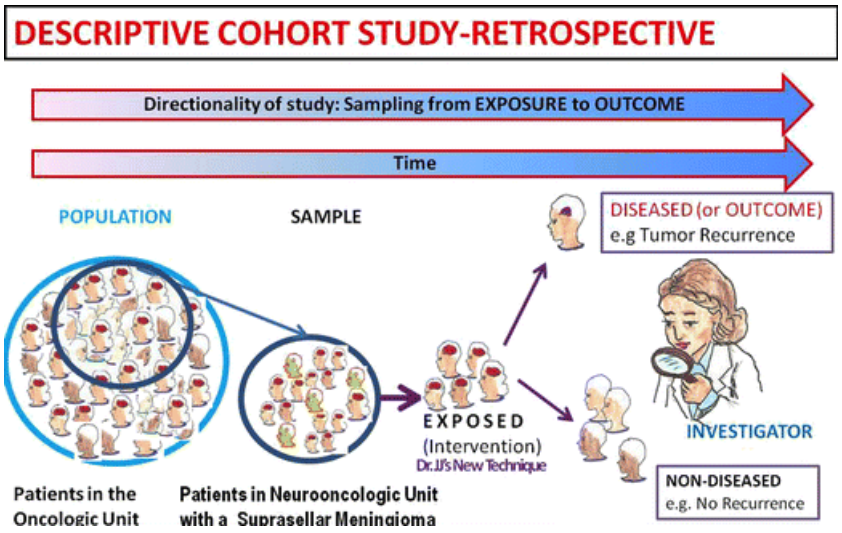


Figure 2.5: An example of a retrospective cohort study conducted on neurosurgery field [29].

Specifically, the investigation and knowledge about cohorts play an important role when building applications based on patients' medical information. The cohort-specific explanations then can be a target when developing interpretable models for healthcare applications. The importance of providing learned information about the subgroups of a population is also essential in other fields such as biology or environmental study. The illustration in Figure 2.5 is an example of retrospective cohort study on the neurosurgery field.

Local Explainability

Locally interpretable models are designed to be more justifiable and can explain the reasons for specific decisions made by them [16]. While globally interpretable models provide a degree of transparency about their inner structure which can ensure that they work as expected, locally interpretable models provide users with understandable rationale for their specific decisions.

k-Nearest Neighbors is a popular non-parametric classifier proposed by [30]. k-Nearest Neighbors (k-NN) is developed on the idea that the information about the label of a target pattern x' can be inferred from its k nearest neighbors. Specifically, k-NN assigns the class label of the target instance as the label of the majority of its k-nearest patterns in the training data space. In order to find the nearest neighbors, a similarity measure is needed to be defined in the data space. Usually, in a q dimensional space $\mathbb{R}^q$, it is reasonable to employ the Minkowski metric (p-norm) [31].

$$||x' - x_j||^p = \left(\sum_{i=1}^{q} |(x_i)' - (x_i)_j|^p \right)^{1/p}$$

In case $p = 2$, the similarity measure corresponds to the Euclidean distance. In other data spaces, adequate distance functions have to be chosen, e.g., the Hamming distance in $\mathbb{B}^q$. In the case of binary classification where the label set is denoted as $\mathbf{y} = \{1, -1\}$, then the kNN function to decide the class of a new instance x' is defined as

$$f_{kNN}(x') = \begin{cases} 1 & \text{if } \sum_{i \in \mathcal{N}_k(x')} y_i \geq 0 \\ -1 & \text{if } \sum_{i \in \mathcal{N}_k(x')} y_i < 0 \end{cases}$$

where $\mathcal{N}_k(x')$ is the set of indices of the k nearest instances.

The essential part of k-NN is the choice of k value which defines the locality of k-NN. Choosing the value of k too small or too large can negatively affect the performance depend on the imbalance and the agglomerations of the underlying data. Practically, the optimized value of k can be derived using model selection

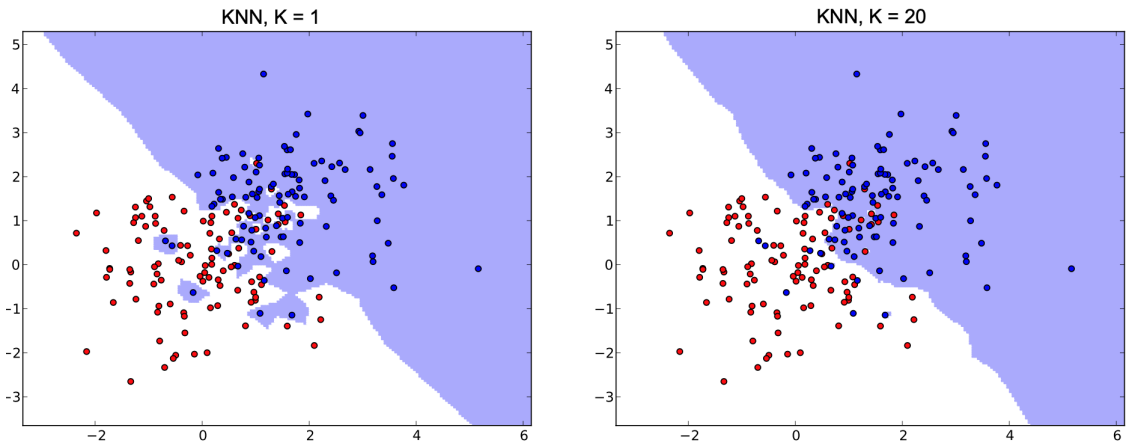


Figure 2.6: The results of kNN classification on Gaussian-based data with two neighborhood sizes ((a) $k = 1$ and (b) $k = 20$). With the value of $k = 1$, kNN has a tendency to overfit the problem, while with a larger k, the algorithm will ignore small patterns of data concentration [31].

techniques such as cross-validation. Figure 2.6 shows the influence on the clustering of data with different k values.

Growing Self-Organizing Maps (GSOM) [32] is a data representation method that is usually used in clustering task. GSOM is an improved version of Self-Organizing Maps (SOM) [33]. GSOM inherits the capability of preserving the topology structure of the underlying data as well as is able to learn a new representation dynamically. It is normally used as a tool for mapping high-dimensional data into a low-dimensional feature map.

Unlike SOM, GSOM does not predefine the number of nodes for the generated map. GSOM learning process includes three phases: initialization, growing and smoothing phases. In the initialization phase, GSOM is started with four nodes and can grow the number of nodes during the learning process. All of the four nodes are boundary nodes (nodes have at least one of its immediate neighboring positions free of a node), therefore new nodes can be grown from all of the four boundary nodes. In the initialization phase, each node is set with a random weight vector.

In the growing phase, the new input data will be assigned to their nearest nodes by comparing the distance (similarity) between new input data with all weight vectors using Euclidean distance. The process can be described by the following formulation.

$$|v - w_{q'}| \leq |v - w_q|, \forall q \in \mathbb{N} \tag{2.2}$$

$$\text{where} \begin{cases} q' & : \text{assigned node (winner)} \\ q & : \text{all nodes in the network} \\ v, w & : \text{input and weight vectors respectively} \end{cases}$$

After assigning new data to their nearest nodes (winner nodes), weight vector of winner nodes and their neighbors will be adjusted in the way that closer neighbors will be adapted more than further ones. The weight adaptation can be described by Eq. 2.3. After that, it increases the error value of the winner node (the difference between input vectors and weight vectors). If the total error of a node is greater than the growth thread-hold (a hyper-parameter set by users), a new node will be grown from that node (if it's boundary node) or distribute the weight to neighbors if it's not a boundary node. The new grown node's weight vector will be initialized so that it will match with weights of neighboring nodes.

$$w_j(k+1) = \begin{cases} w_j(k), & j \notin N_{k+1} \\ w_j(k) + LR(k) * (x_k - w_j(k)), & j \in N_{k+1} \end{cases} \tag{2.3}$$

$$\text{with} \begin{cases} LR(k) & : \text{learning rate} \\ w_j(k), w_j(k+1) & : \text{weight vector of node } j \text{ before and after being adjusted} \\ N_{k+1} & : \text{neighborhood of the winning node at } (k+1)_{th} \text{ iteration} \end{cases}$$

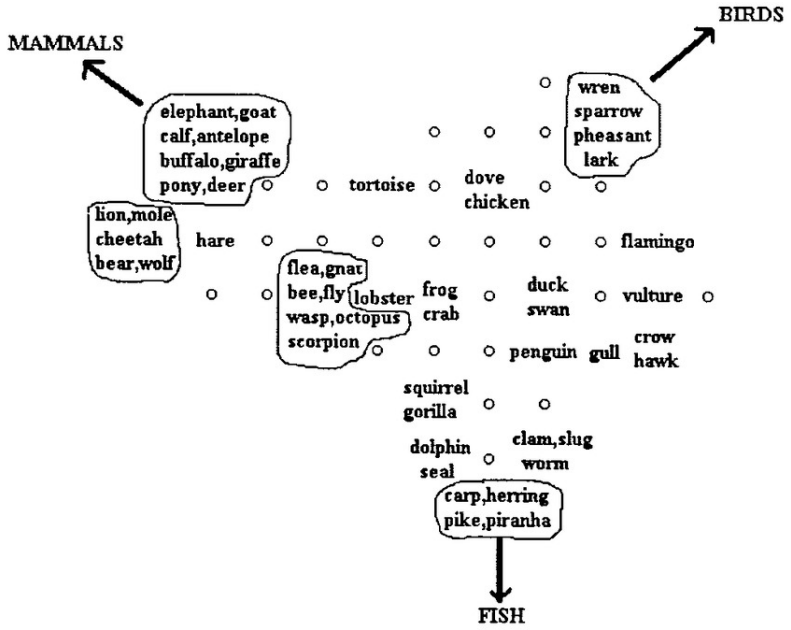


Figure 2.7: The GSOM for the zoo dataset [32].

The smoothing phase takes place after the new node growing phase. Inputs to the network are the same as growing phase with a smaller learning rate. As a result

of that, the weight adaptation is done with a lower rate of adaptation in order to smooth out existing quantization error. One of the demonstrated results of GSOM with zoo dataset from UCI repository is depicted in Fig. 2.7. It's is interesting in what we can be able to observe from the generated map, animals that have similar biological characteristics are forming groups (mapped closely to each other). It shows that GSOM has the capability of reserving salient structures of the original data while mapping them into a lower dimensional space.

B. Post-hoc Models

When ML models do not satisfy the aforementioned descriptions to be considered as transparent models, a separate method must be devised and applied to the model to explain their decisions [17]. Specifically, the purpose of post-hoc explainability techniques (or post-modeling explainability) is to be able to extract understandable information about how an already trained model generates its predictions for any given input.

There is an observation about the trade-off between model accuracy and explanation fidelity for transparent and post-hoc models [2]. Specifically, intrinsically interpretable models can provide undistorted explanations about their decision-making process but may sacrifice prediction performance to some extent. While post-hoc models are limited in their approximate nature while keeping the underlying model accuracy intact [16]. Moreover, there is criticism about the use of post-hoc models to explain for the original black-box ones such as the doubts about the fidelity of the explanations provided by post-hoc models or if the post-hoc models can mimic the behaviors of the original ones so why do we need those black-boxes at the beginning [2].

There are generally two types of post-hoc approaches: *model-agnostic* and *model-specific* methods [17, 22, 14]. Model-agnostic methods are built to give explanations for any kind of models while model-specific methods are designed for a specific kind of black-box model such as deep neural network or random forest. Details about these two types of post-hoc approaches will be described in the following parts.

Model-Agnostic Explainability

Model-agnostic techniques for post-hoc explainability are designed to be plugged to any model with the intent of extracting information from its prediction procedure [17]. In several cases, simplification techniques are utilized to generate proxies that mimic behaviors of black-box models with the purpose of tracking their decision-making procedure and reducing the models' complexity. On the other hand, model-agnostic methods may focus on extracting knowledge directly from the models or visualizing the processes to ease the interpretation of their behavior [17].

Basically, model-agnostic explanation methods are not dependent on the original black-box models. For this reason, model-agnostic methods can have the advantage of reusing capability with different black-box models. However, in some cases, model-specific methods can provide more useful information due to their designation for a specific system [34]. In the next part, we would like to review two popular model-agnostic methods named *Local Interpretable Model-Agnostic Explanations (LIME)* [35] and *Shapley Additive Explanations (SHAP)* [36].

Local Interpretable Model-Agnostic Explanations - (LIME) [35] is a popular framework that generates local explanations for black-box models. LIME approximates the prediction of any black-box via local surrogate interpretable models. It provides explanations for an instance by perturbing it around its neighborhood. The perturbed samples are then fed to a complex model for labeling and weighted based on their proximity to the original data. Finally, LIME learns an interpretable model on the weighted perturbed data and their associated labels to create the explanations.

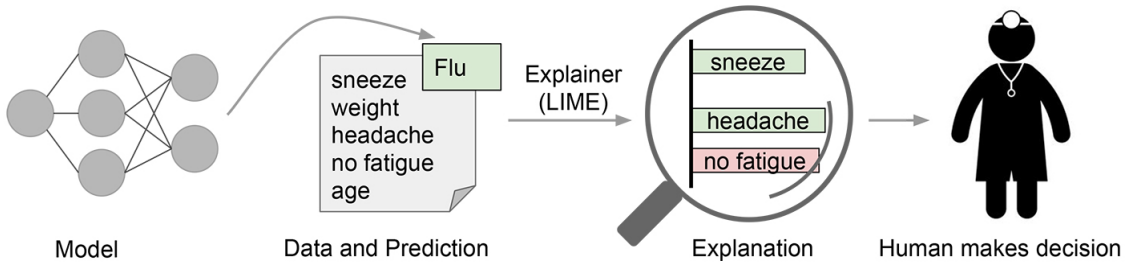


Figure 2.8: An example of LIME for explaining a single prediction. Given a prediction provided by a trained model that a patient has the flu, LIME will emphasize the symptoms in the patient's history that led to the prediction. Specifically, sneeze and headache contribute to the "flu" prediction, while "no fatigue" is evidence against it [35].

Specifically, in order to formulate the problem, they assume that interpretable explanations need to use a representation that is understandable by human. Those representations are denoted as $g \in G$, where G is a class of potentially interpretable models. The domain of g is $\{0, 1\}$ that acts over the absence or presence of interpretable components. Also, $\Omega(g)$ is defined as the complexity of the explanation g. The unfaithfulness an explanation g when approximating a model f is measured by a function $\mathcal{L}(f, g, \pi_x)$. Then, LIME ensures the interpretability and the local fidelity of its explanations by minimizing the function $\mathcal{L}$ and the complexity of the explanations $\Omega(g)$ as the following.

$$\xi(x) = \underset{g \in G}{argmin}\ \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

where π_x is a proximity measure to find neighbors of a instance x.

In order to be model-agnostic, no assumptions are made about the working mechanism of the model f. Therefore, the optimized value of $\mathcal{L}$ is approximated by drawing samples around an instance x and weighted by the distances to its neighbors π_x. The primary intuition behind LIME is presented in Figure 2.9, sample instances are drawn from both in the vicinity of x (which have a high weight due to πx) and far away from x (low weight from πx). Even though the original model may be too complex to explain globally, LIME presents an explanation that is locally faithful.

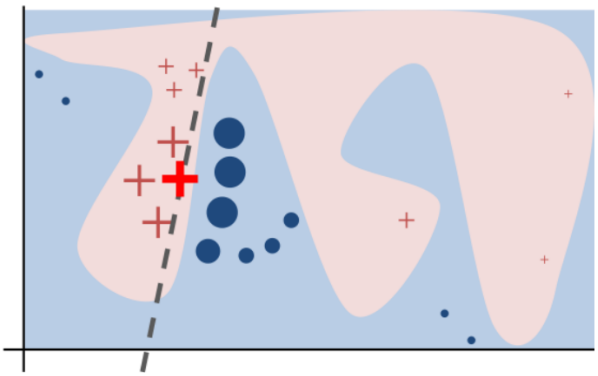


Figure 2.9: An illustration of LIME's intuition. The blue and pink background represent for the decision function f of a black-box model which is unknown to LIME. While the bold red cross sign is the decision made by f that need to be explained. LIME will generate the explanation by sampling its neighboring instances and weighting the results by their *"distance"* to the explained instance (represented by their size in the figure). The dashed line is the learned explanation that is locally (but not globally) faithful [35].

Kernel Shapley Additive Explanations - (Kernel SHAP) [36] is a unified version of the linear LIME method and Shapley values that assigns each feature an importance value for a particular prediction. Shapley values is a common method to explain model predictions from the cooperative game theory. The explanations given using Shapley values satisfy desirable properties including

Local accuracy: the explanation model $g(x')$ matches the original model $f(x)$ when $x = h_x(x')$, where $\phi_0 = f(h_x(0))$ represents the model output with all simplified inputs toggled off (i.e. missing).

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i$$

Missingness: Missingness constrains features where $x'_i = 0$ to have no attributed impact.

$$x'_i = 0 \implies \phi_i = 0$$

Consistency: if a model changes so that some simplified input's contribution increases or stays the same regardless of the other inputs, that input's attribution should not decrease.

$$f'_x(z') - f'_x(z' \backslash i) \geq f_x(z') - f_x(z' \backslash i) \text{ for all inputs } z' \in \{0,1\}^M, \text{ then } \phi_i(f', x) \geq \phi_i(f, x)$$

As pointed out by [36] about the relationship between additive feature attribution methods (including LIME), they pointed out that LIME can have a unique solution for the weight of each feature as the Shapley values. However, for LIME due to the heuristically choosing of the loss function $\mathcal{L}$, weighting kernel π_x and regularization term Ω, in some cases, LIME cannot guarantee the local accuracy and consistency which results in intuitive behavior [36]. To mitigate the limitation of LIME, kernel SHAP was proposed with a specific way to defined the formulations of the aforementioned loss function, weighting kernel and regularization term as the following so that LIME can return a unique explanation $g(x')$ for a given instance x that satisfy the desirable properties. An example of SHAP's result is illustrated in Figure 2.10.

$$\Omega(g) = 0$$

$$\pi_{x'}(z') = \frac{(M-1)}{(M \text{ choose } |z'|)|z'|(M-|z'|)}$$

$$\mathcal{L}(f, g, \pi_{x'}) = \sum_{z' \in Z} [f(f_x(z')) - g(z')]^2 \pi_{x'}(z')$$

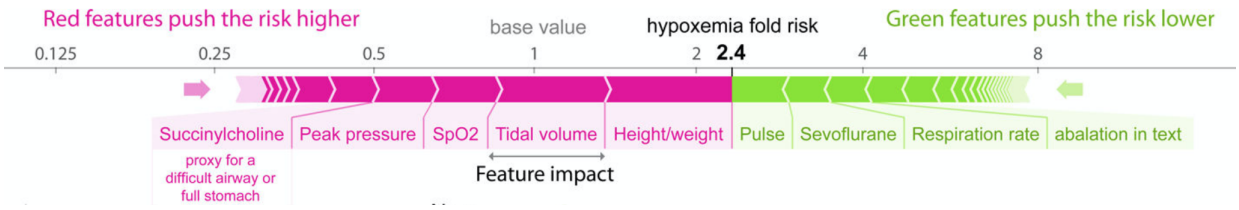


Figure 2.10: Explanations for the risk of hypoxaemia in the next five minutes during surgery [37].

Model-Specific Explainability

According to [17], model-specific explainability is tailored or designed to explain certain ML models. Many model-specific methods are designed for Deep Neural Networks (DNN), which is a class of models that are widely used because of its prominent performance in spite of being very opaque in terms of interpretability,

a typical example of black-box models [34] . Besides, model-specific explanations are also provided for other common black-box models such as random forests or other ensemble learning methods. In the next part, we would like to introduce a notable model-specific explanation method for convolutional neural networks which is saliency maps [38].

Saliency Maps for Convolutional Networks is firstly proposed by [38] for visualizing regional parts of pictures that most affect the classification process of a trained convolutional network. Specifically, in order to query about spatial support of a particular class c in a given image I_0, the pixels of I_0 will be ranked based on their influence on the score model $S_c(I_0)$ as follows.

$$S_c(I) = w_c^T I + b_c$$

Practically, the class score model $S_c(I)$ that is used in deep convolutional networks is much more complicated and highly non-linear. However, given a specific input image I_0, the value $S_c(I)$ can be approximated with a linear function in the neighborhood of I_0 with the computation of the first-order Taylor expansion:

$$S_c(I) \approx w^T I + b$$

Then w can be computed as the derivative of S_c at the point of I_0.

$$w = \frac{\partial S_c}{\partial I}\bigg|_{I_0}$$

After computing the derivative w using back-propagation, the saliency map $M \in \mathbb{R}^{m \times n}$ of an image I_0 that has m rows and n columns can be obtained by rearranging the elements of the vector w. In the case of grey-scale images, the number of elements in w is equal to the number of pixels in I_0, and the saliency map for I_0 can be built as $M_{ij} = |w_{h(i,j)}|$ with $h(i,j)$ is the index of the element in w which is corresponding to the pixel at i^{th} row and j^{th} column. For multi-channel images, assuming that the color channel c of the pixel (i,j) of image I_0 corresponds to the element indexed $h(i,j,c)$ of w. Then the saliency map for I_0 across all color channel can be derived as the maximum value of those channel $M_{ij} = max_c|w_{h(i,j,c)}|$.

In Figure 2.11, the saliency maps are illustrated for the highest-scoring class on randomly selected ILSVRC-2013 test set images [38]. According to the results, prominent objects in images are highlighted with a brighter color which shown their effects on final classification results. Moreover, in this work, they proposed a method for detecting objects in images with the background of saliency maps with a threshold scheme for spotting whole objects.

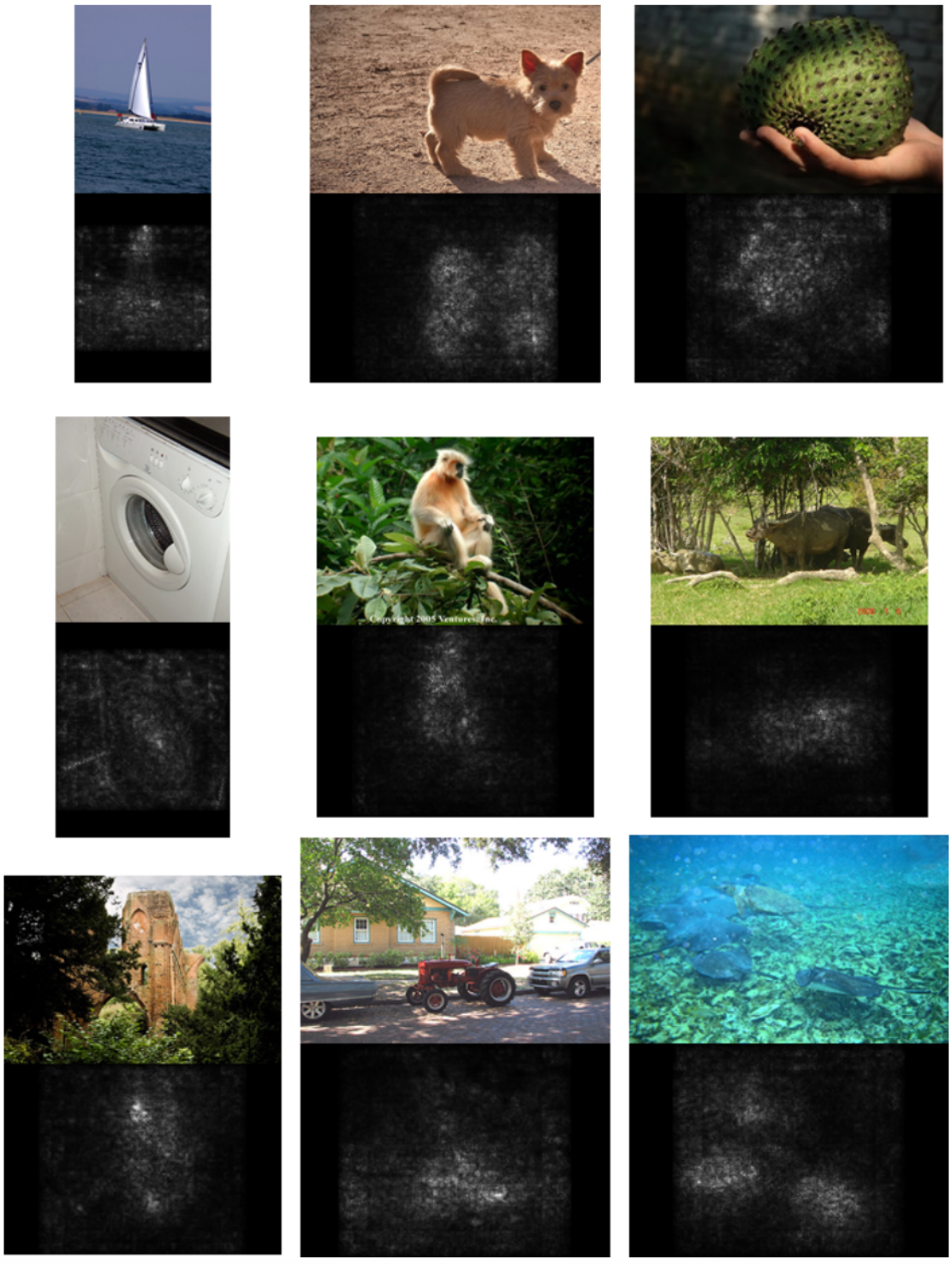


Figure 2.11: The saliency maps for images in the top-1 predicted class that belong to ILSVRC-2013 dataset [38].

2.1.2 Causal Interpretable Models

In this subsection, we discuss the general notions and mechanisms of causal interpretability frameworks. According to [14], current ML models only optimized to discover correlations - not causal information which can be a problem when it's is required for decision-making in real-world situations. An example can be listed as the policy-making that is related to smoking and cancer. This problem raises the need for causal ML models which can capture real causal information. Specifically, a causal interpretable model can help us to gain insights into the real causes of decisions made by ML algorithms, improve their performance, and prevent them from failing in unexpected circumstances.

As mentioned in the work of [39], there are three different levels of interpretability and argues that generating counterfactual explanations is the way to achieve the highest level of interpretability. Below are those levels of interpretability and their definitions:

- Statistical (associational) interpretability: Aims to uncover statistical associations by asking questions such as "How would seeing x change my belief in y?'
- Causal interventional interpretability: Is designed to answer "What if" questions.
- Counterfactual interpretability: Is the highest level of interpretability, which aims to answer "Why" questions.

Before leading into the details of approaches to providing causal interventional and counterfactual interpretability, we would like to introduce several background concepts of causal inference as the following [14].

- *Structural Causal Models (SCM)* is defined as a 4-tuple variable $M(X, U, f, P_u)$ where X is a finite set of endogenous variables which are usually observable, U is a finite set of hidden variables, f is a set of function $\{f_1, f_2, ..., f_n\}$ where each function represents a causal mechanism such that $\forall x_i \in X$, $x_i = f_i(Pa(x_i), u_i)$ and$Pa(x_i)$ is a subset of $(X \setminus \{x_i\}) \cup U$ and P_u is a probability distribution over U.
- *Causal Bayesian Network (CBN)* is a representation of a SCM $M(X, U, f, P_u)$. CBN is a directed graph model $G(V, E)$ where V is the set of observable variables X and E denotes the causal mechanism.

- *Average Causal Effect (ACE)* of a binary random variable x on another random variable y (outcome) is defined as :

$$ACE = \mathbb{E}[y|do(x = 1)] - \mathbb{E}[y|do(x = 0)]$$

where $do(.)$ is the operator which denotes the corresponding interventional distribution defined by the SCM or CBN.

A. Causal Interventional Interpretability

Causal interventional interpretability is designed to explain the role and importance of each component of a ML model on its decisions with concepts from the causality [14]. As traditional interpretability cannot provide explanations for better understanding of ML models, several causality approaches have been employed to solve the problem. According to [14], currently there are two common approaches in building model-based explanations with causality. The first approach is considering the complex models in the form of Structural Causal Models (SCM) and defined a way to estimate the Average Causal Effect (ACE) of each component inside the original models. The second approach is the assembly of a good performance model, domain knowledge on the causal graph and appropriate visualization tools. In this part, we would like to introduce a notable method to generate model-based explanations for complex models proposed by [40] that belongs to the first approach. More details about other methods can be found in the survey of causal interpretable models [14].

Causal Attributions for Neural Networks was proposed by [40] in order to identify the causal influence of an input to a neural network's output. Specifically, they tried to address the question: "What is the causal effect of a particular input neuron on a particular output neuron of the network?". In order to provide the answer for that question, a neural network (generally a feed-forward neural network) is considered as an SCM $M([l_1, l_2, ..., l_n], U, [f_1, f_2, ..., f_n], P_u)$ where l_i ($i \in \{1, 2, ..., n\}$) is the layer i of the network and f_i is the causal function at layer i.

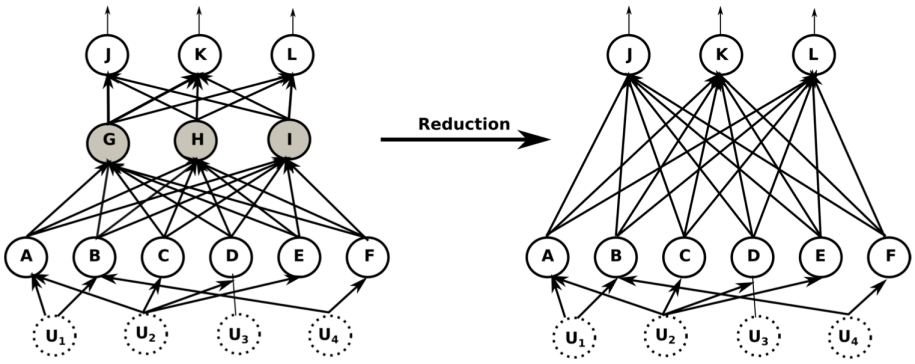


Figure 2.12: The representation of a feed-forward neural network as an SCM [40].

Practically, only neurons in the input and output layers can be observable, therefore the causal structure of the original neural network can be reduced as $M([l_1, l_n], U, f', P_u)$. Consequently, the causal attribution of an input neuron x_i for an output neuron y can be defined as the value of the average causal effect ACE of a continuous variable x_i on the target variable y as the following.

$$ACE^{y}_{do(x_i=\alpha)} = \mathbb{E}[y|do(x_i = \alpha)] - baseline_{x_i}$$

where $\mathbb{E}[y|do(x_i = \alpha)]$ is the interventional expectation of y given the intervention $do(x_i = \alpha)$ and $baseline_{x_i}$ is the average ACE of x_i on y which are defined as below. Details on the computation of the interventional expectation and the baseline can be found in [40].

$$\mathbb{E}[y|do(x_i = \alpha)] = \int_y yp(y|do(x_i = \alpha))\, dy$$
$$baseline_{x_i} = \mathbb{E}_{x_i}[\mathbb{E}_y[y|do(x_i = \alpha)]]$$

B. Counterfactual interpretability

Counterfactual interpretability is considered as the example-based interpretation that aims to provide data instances that are able to explain for the model or the underlying data distribution [14]. Specifically, counterfactual explanations is designed to answer the causal question "Why" as mentioned in the previous parts by providing counterfactual examples that help users understand a model's decisions. Furthermore, counterfactual examples can be obtained by performing minimal changes in the original instance's features and have a predefined output. Practically, the problem is formulated as a new type of conditional probability $P(y_x|x', y')$ that indicates how likely the outcome of an observed instance y' would change to y_x if x' is set to x [14].

Currently, there are several different approaches to generate counterfactual examples for explaining decisions of ML models. However, main approaches vary from reducing the distance between models' predictions and counterfactual outcomes to leverage adversarial or prototype examples. Applications of counterfactual explanations are also diverse including image, video classification, natural language processing or credit classification in banking. In this part, we would like to introduce an approach that is used for generating counterfactual visual explanations for the image classification task that was proposed in the work of [41].

Counterfactual Visual Explanations for convolutional neural networks (CNN) was proposed by [41] with the aim of generating explanations for the decision of deep computer vision systems by identifying what and how regions of an input image

would need to change in order for the system to produce a specific output. Specifically, given a specific image I which is predicted as class c by the system, a faithful counterfactual explanation to identify the aforementioned reason could be generated by detecting the smallest replaceable spatial regions in I and I' (an image classified as class c') which can make the system predict I as class c'.

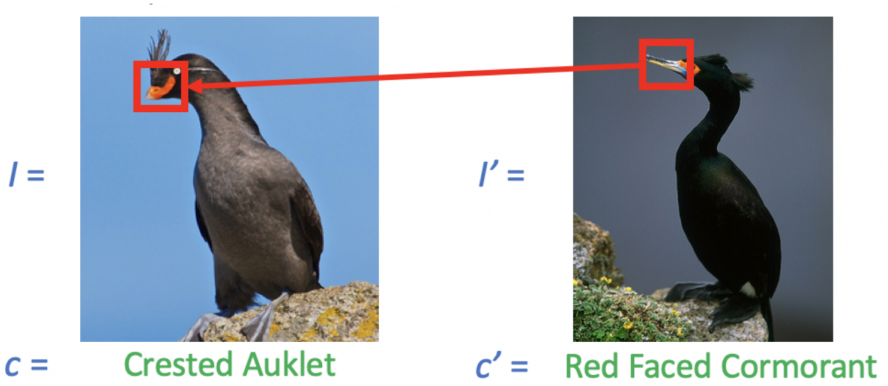


Figure 2.13: An example of generating counterfactual visual explanations for a query image I. It explains the reason that I was classified as class c (Crested Auklet) instead of c' (Red Faced Cormorant) by finding a region in a distractor image I' and a region in I (red boxes) so that if exchanging the highlighted region in both images then the resulting image $I^\star$ would be classified more confidently as c' [41].

By decomposing a CNN as two main parts: spatial feature extractor $f(I)$ and decision network $g(f(I))$, given two images I, I', a transformation T can be conducted as $I^\star = T(I, I')$ such that $I^\star$ is classified as class c' by the train model $g(f(.))$. The transformation is conducted by finding the minimum number of region replacements from I' to I to generate $I^\star$ that satisfies the condition (minimum-edit counterfactual problem). The transformation can be formulated as the following where P is the permutation matrix and $\mathbf{a}$ is the binary vector indicating the replacements (0 - no replacement, 1 - replacement).

$$f(I^\star) = (\mathbb{1} - \mathbf{a}) \circ f(I) + \mathbf{a} \circ Pf(I')$$

with $\mathbb{1}$ is a vector of all ones and $\circ$ represents the Hadamard product.

Then the minimum-edit counterfactual problem can formulated as

$$\underset{P,\, \mathbf{a}}{minimize} ||\mathbf{a}||_1$$

s.t. $c' = argmax\ g((\mathbb{1} - \mathbf{a}) \circ f(I) + \mathbf{a} \circ Pf(I'))$ with $a_i \in \{0, 1\}\ \forall i$ and $P \in \mathcal{P}$ where $\mathcal{P}$ is the set of all permutation matrices.

Given the resulting $\mathbf{a}$ and P , the set of pairs of spatial cells involved in the edits can be extracted as $S = \{(i, j, i', j') | a_{i \times j} = 1 \wedge P_{i \times j, i' \times j'} = 1\}$. The above optimization problem can be solved using greedy sequential exhaustive search or continuous relaxation as described in [41].

2.2 Evaluation for Interpretability

Evaluation of interpretability is a challenging task due to the lack of a consensus definition of interpretability and understanding of humans from the concept. Evaluation of causal interpretability is even more challenging due to the lack of ground-truth data for causal explanations and verification of causal relationships [14]. Currently, to the best of our knowledge, there is no unified guideline for evaluating the interpretability of both traditional and causal models. In this subsection, we would like to introduce three common interpretability evaluation methodologies including an explanation evaluation taxonomy [15], the PDR framework [4] and a unified hierarchical framework for explanation evaluation [42].

2.2.1 An Explanation Evaluation Taxonomy

In the work of [15], they proposed a taxonomy of approaches for evaluating the interpretability of ML models. The proposed taxonomy includes three main categories: application-grounded, human-grounded, and functionally-grounded evaluation methods. It emphasized the connections and significance of the interaction between specific tasks and human. Where the tasks could have a different range of whether they are real tasks, simple or proxy tasks. Also, the interaction with human when evaluating the interpretability of ML models is an essential factor as well. Depend on specific situations, different types of approaches could be utilized for the evaluation. Details of the taxonomy are depicted in Figure 2.14.

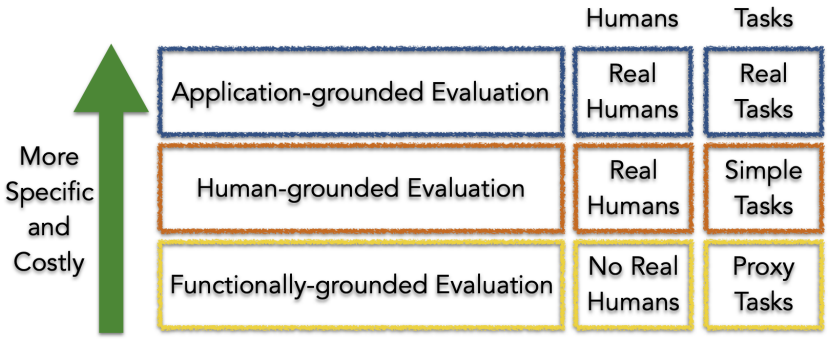


Figure 2.14: Taxonomy of evaluation approaches for interpretable models [15].

Application-Grounded Evaluation: Real Humans, Real Tasks

Application-grounded evaluation relates to the conduction of human experiments within a real-life application. If a system is designed for a specific task such as assisting doctors in diagnosing a specific disease, then it is suitable to perform the evaluation of this system with respect to that real task. This type of evaluation is intuitive and aligns with assessment methods in human-computer interaction and visualization fields where efforts are made to ensure that a system follows its intended task.

Specifically, the quality of an explanation will be evaluated in the context of its end-task such as whether the generated explanation can provide better identification of errors, new knowledge, or improve fairness. Two example cases of evaluation can be listed as:

- Domain expert experiment with the exact application task.
- Domain expert experiment with a simpler or partial task to shorten experiment time and increase the pool of potentially-willing subjects.

In both types of evaluation experiments, it is important to evaluate how well the produced explanations can assist humans in completing their tasks. In summary, this evaluation method directly tests the objective that the system is constructed for and results of the experiments with respect to those objectives will provide strong evidence for how successful the system is.

Human-Grounded Evaluation: Real Humans, Simplified Tasks

Human-grounded evaluation relates to the conduction of experiments with humans and simplified tasks while still preserves the nature of target applications. It shows the advantage when experiments with full-scale real tasks are challenging. Moreover, with simplified tasks, it is not necessary to employ highly-trained domain experts for conducting the tasks which in turn reducing the cost. Human-grounded evaluation is most appropriate when testing general notions of the quality of explanations.

However, it is also challenging in the case of tasks with specific end-goal such as identifying errors in a safety-oriented task or picking out relevant patterns in a science-oriented task. In that case, human-grounded evaluation can be conduct ideally while depends only on the quality of explanations. Several possible experimental evaluations can be listed as:

- Binary forced choice: humans are provided with pairs of explanations and will be asked to choose the one that they find of higher quality (basic face-validity test made quantitatively).

- Forward simulation/prediction: humans are provided with an explanation and an input and will be asked to correctly simulate the model's output (without regarding of the true output).
- Counterfactual simulation: humans are provided with an explanation, an input, and an output, and are asked what must be changed to change the method's prediction to the desired output (and related variants).

Functionally-Grounded Evaluation: No Humans, Proxy Tasks

Functionally-grounded evaluation does not require human-related experiments. It utilizes proxies to evaluate the quality of explanations. This type of evaluation is appropriate when there is a need to reduce experiment time and costs. Also, it can be used as the first step of assessment for immature systems or unethical when conducting on humans. The major point in this type of experiment is the selection of proxies. Generally, several considered interpretable models such as decision trees or rule lists can be used to evaluate the explanations. Several examples of functionally-grounded experiments can be listed as

- Demonstrate that there is the improvement on performance of a model regarding the proof that it is interpretable.
- Prove that a method performs better with respect to certain models such as being more sparse—compared to other baselines.

2.2.2 The Predictive Accuracy - Descriptive Accuracy - Relevancy (PDR) Framework

Before describing the details of the PDR framework proposed by[4], a general model of an interpretable ML process will be introduced as the background for later discussion. Specifically, according to [4], interpretable ML processes can be generalized as a broader data-science life cycle as in Figure 2.15. The process starts with defining the domain problem with the help of data. Then predictive models can be constructed by practitioners and fitted on the collective data. Finally, the aforementioned questions can be answered by analyzing the trained models (post-hoc analysis). The general process is iterative until satisfying answers are obtained. According to this procedure, two types of interpretability are also spotted out: *model-based interpretability* (in the modeling stage) and *post-hoc interpretability* (at the post-hoc analysis stage).

As a guideline to select and evaluate interpretation methods for a particular problem and audience, the authors proposed PDR framework which consists of three

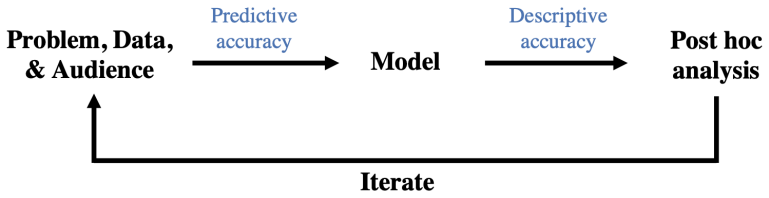


Figure 2.15: Different stages of an interpretable ML process in the data-science life cycle.

desiderata that should be used to select interpretation methods for a particular problem: *predictive accuracy*, *descriptive accuracy*, and *relevancy*. Specifically, It is important to guarantee that the explanations provided by an interpretation method should be faithful to the underlying process. For ensuring the trustworthiness of the generated explanations, it is necessary to maximize two types of accuracy: predictive accuracy (when approximating the underlying data relationships with a model) and descriptive accuracy (when approximating what the model has learned using an interpretation method). Moreover, the provided explanations also have to be relevant to a specific audience (usually users of the systems).

Predictive accuracy. At the stage of constructing the model, if the model poorly approximates the underlying relationships in the data, then information extracted from the model is unlikely to be accurate. There are various well-studied methods to evaluate the fitness of a model, especially in the supervised learning field. In the context of interpretability, the model's fitness is described as predictive accuracy.

Specifically, when dealing with interpretability problems, it is required for a model to possess a predictive accuracy above average. Also, the distribution of predictions is important and needs to be at an appropriate level for every available class. Moreover, the predictive accuracy should be stable with respect to reasonable data and model perturbations.

Descriptive accuracy is defined as the level that an interpretation method (post-hoc model) can unbiasedly capture the relationships in data that learn by ML models. Specifically, in the stage of post-hoc analysis, errors typically occur when interpretation methods are utilized to extract knowledge from a trained model. It is generally a difficult task to extract non-linear relationships learn by black-box models such as deep neural networks. In those cases, usually post-hoc models can only provide an imperfect representation of the relationships learned by a model.

Relevancy is considered as another aspect of interpretability. Specifically, an explanation is regarded as relevant when it can provide insights into a specific domain problem for a particular audience. In other words, an ML model need not have only high predictive accuracy but also the information extracted from it has to be relevant

as well. Relevancy can be seen as a key part when considering the trade-off between predictive and descriptive accuracy.

The influences of different types of interpretability (model-based and post-hoc) ML models to the mentioned desiderata are illustrated in Figure 2.16. In particular, post-hoc and model-based methods both aim to increase descriptive accuracy while model-based methods can also affect predictive accuracy. Both of the models can also affect the relevancy that depends on the type of output whether it is helpful for a particular problem and audience or not. Model-based interpretability with high descriptive accuracy can be achieved using a simpler model to fit the data, however, it can also negatively affect predictive accuracy. Post-hoc interpretability is the use of a second model to extract information from an already trained one which has no impact on predictive accuracy.

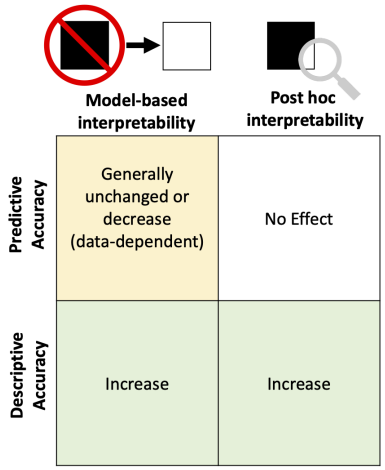


Figure 2.16: Impact of interpretation methods on descriptive and predictive accuracy [4].

2.2.3 A Unified Hierarchical Framework for Explanation Evaluation

In the work of [42], they proposed a unified framework to evaluate the explanations that are generated by interpretable ML models. Specifically, the evaluation is based on three general properties of an explanation: *generalizability*, *fidelity* and *persuasibility*. Before introducing their proposed framework, we briefly remind those properties' definitions. Firstly, *generalizability* is the property to reflect the generalization power of an explanation. In other words, generalization determines the quality and applicability of an explanation on the range of knowledge and guidance that it can provide to users. Secondly, *fidelity* of an explanation is defined as its degree of faithfulness regarding a target system. Specifically, a faithful explanation

can precisely capture the decision-making process of the target system and provide the correct evidence for a particular prediction. Finally, *persuasibility* reflects the degree of how humans comprehend and respond to the generated explanations. This property assesses the subjective aspect of explanations and usually involves with humans' evaluation. Depend on a specific group of users and tasks, the persuasibility of an explanation varies.

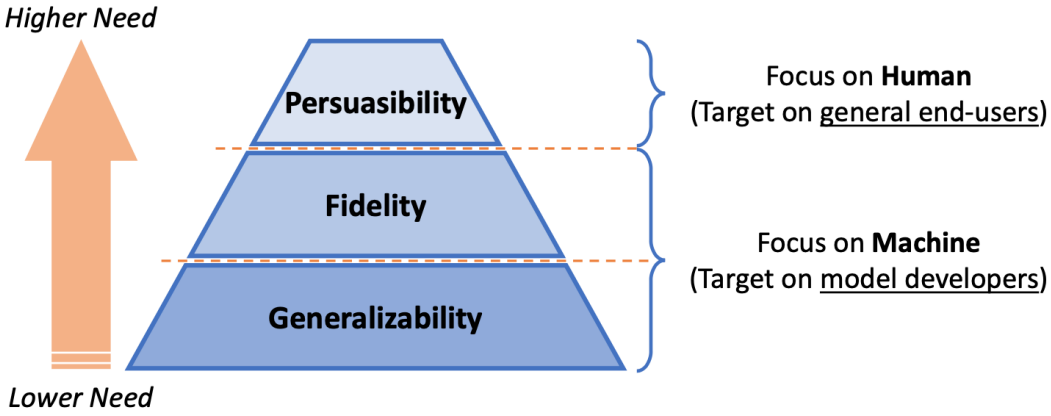


Figure 2.17: A unified hierarchical framework for evaluating explanations in interpretable ML [42].

Given the definitions of general properties of explanations that are generated by interpretable ML models, a unified framework for evaluating explanations can be defined as a hierarchical structure of three properties generalizability, fidelity and persuasibility from low level to highest level correspondingly as in Figure 2.17. Particularly, generalizability serves as the foundation in evaluation with basic requirements. In ML applications, proper generalizability of the generated explanations allows users to make accurate decisions. It guarantees that the explanations have both a good degree of generalization as well as reveal true knowledge for particular tasks. For the upper level, the reliability of generated explanations needs to be verified by evaluating its degree of fidelity to the original models. By assessing the fidelity, it helps improve the trustworthiness of the generated explanations for better decision-making. The two levels of generalizability and fidelity are considered from the machine perspective while the top tier persuasibility is more perceived from the human perspective.

At the top tier of persuasibility, it requires a higher level of interpretation that is needed to shorten the gap between users and models-generated explanations. For a specific task, it depends on the corresponding applications and user groups to the designation of explanations and the level of interpretability. Generally, model developers should pay more attention at the machine-related level (generalizability and fidelity) while end-users will concentrate more to the persuasibility at the highest level.

2.3 Applications of Interpretable ML and XAI

Applications of interpretable ML and XAI span across fields, especially in fields which require to make high-stakes decisions such as healthcare, medicine (personalized medicine, drug discovery), banking (credit evaluation), material science (material discovery). In this section, we introduce several applications of interpretable ML in aforementioned fields with the hope of providing a wider view on the need of interpretability in real applications from different perspectives.

A healthcare application - Interpretable classifiers using rules and Bayesian analysis for stroke prediction for patients with atrial fibrillation [43]

According to [44], until 2010, the estimated number of individuals with atrial fibrillation (AF) globally is around thirty-three million. AF is a disorder of heart rhythm that was originally caused by the interplay between genetic predisposition, ectopic electrical activity and abnormal atrial tissue substrate [45]. It has been found that AF has a strong association with ischemic stroke and is the cause of thromboembolism [46]. There are many indices and systems that have been proposed to assess the risk of stroke for patients having AF.

Among them, CHADS2 is a common index that was proposed by [47]. The score of a patient by CHADS2 will be computed by assigning points to the existence of specific factors. Particularly, CHADS2 considers 5 factors: the presence of congestive heart failure, hypertension, age 75 years or older, diabetes mellitus and history of stroke, transient ischemic attack or thromboembolism. CHADS2 has a high level of interpretability which provides doctors with easy to understand scoring assessment to the stroke risk of patients. However, the CHADS2 index was developed with a database of 1733 medicare beneficiaries and a limited number of factors.

if hemiplegia **and** age $>$ 60 **then** *stroke risk* 58.9% (53.8%–63.8%)
else if cerebrovascular disorder **then** *stroke risk* 47.8% (44.8%–50.7%)
else if transient ischaemic attack **then** *stroke risk* 23.8% (19.5%–28.4%)
else if occlusion and stenosis of carotid artery without infarction **then** *stroke risk* 15.8% (12.2%–19.6%)
else if altered state of consciousness **and** age $>$ 60 **then** *stroke risk* 16.0% (12.2%–20.2%)
else if age $\leq$ 70 **then** *stroke risk* 4.6% (3.9%–5.4%)
else *stroke risk* 8.7% (7.9%–9.6%)

Figure 2.18: Example of a decision list for estimating 1-year stroke risk following diagnosis of atrial fibrillation from patient medical history [43].

Currently, with the availability of medical data, especially patient medical records, more effective prediction systems can be developed which the capability of consider-

ing additional factors. In the work of [43], they proposed a new classification model that is interpretable as CHADS2 and constructed based on a much bigger number of data (12586 patients and 4148 factors). The proposed model named Bayesian Rule Lists (BRL) is presented in the form of decision lists that discretize a high-dimensional multivariate feature space into a series of simple readily interpretable decision statements. An example of the results of BRL for classifying stroke risk of patients with AF is shown in Figure 2.18. The decision lists generated by BRL contain a series of $if...then...$ statements where the if statements define a partition of a set of features and the *then* statements is the predicted outcome.

BRL is originally built on the definitions of Bayesian association rules and Bayesian decision lists. Particularly, Bayesian association rules have the form of $a \rightarrow b$ which is an implication with an antecedent a and a consequent b. In this case, b is the predicted label y which follows a multinomial distribution.

$$a \rightarrow y \sim \text{Multinomial}(\boldsymbol{\theta})$$

Consider $(\boldsymbol{x}, \boldsymbol{y})$ are observations classified by the above rule, $N_{.,l}$ is the number of observations with label l (with $l = \{1, ..., L\}$) and $N = \{N_{.,1}, ..., N_{.,L}\}$. Then a so-called posterior consequent distribution can be obtained as

$$\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\alpha} \sim \text{Dirichlet}(\boldsymbol{\alpha} + N)$$

Given the definition of Bayesian association rules, Bayesian decision lists can be defined as $D = (d, \boldsymbol{\alpha}, \boldsymbol{N})$ where d is an ordered antecedent list $d = (a_1, ..., a_m)$. Consider $N_{j,l}$ be the number of observations that satisfy a_j but not any of $a_1, ..., a_{j-1}$ and have label l, denote $\boldsymbol{N}_j = (N_{j,1}, ..., N_{j,L})$ and $\boldsymbol{N} = (\boldsymbol{N}_0, ..., \boldsymbol{N}_m)$. A Bayesian decision list then has the form as

if a_1 **then** $y \sim \text{Multinomial}(\boldsymbol{\theta})_1$, $\boldsymbol{\theta}_1 \sim \text{Dirichlet}(\boldsymbol{\alpha} + \boldsymbol{N}_1)$
else if a_2 **then** $y \sim \text{Multinomial}(\boldsymbol{\theta})_2$, $\boldsymbol{\theta}_2 \sim \text{Dirichlet}(\boldsymbol{\alpha} + \boldsymbol{N}_2)$
$\vdots$
else if a_m **then** $y \sim \text{Multinomial}(\boldsymbol{\theta})_m$, $\boldsymbol{\theta}_m \sim \text{Dirichlet}(\boldsymbol{\alpha} + \boldsymbol{N}_m)$
else $y \sim \text{Multinomial}(\boldsymbol{\theta})_0$, $\boldsymbol{\theta}_0 \sim \text{Dirichlet}(\boldsymbol{\alpha} + \boldsymbol{N}_0)$

The antecedents in BRL are a subset of a preselected collection of antecedents which in turn generated by using a frequent itemset mining technique with the constraints that each antecedent applies to a large amount of data and does not have too many conditions. More details about the generative process of Bayesian decision lists and the prediction for new observations can be found in [43].

Interpretable ML models for material discovery

Chemicals are indispensable part of human necessity including applications spanning a wide range of areas from the laboratory, industrial processes to household usage. In every aspect of those applications, it is existing the need of designing materials that possess specific properties. Moreover, it is important to guarantee that the new materials are not toxic to humans or harmful to the surrounding environment. QSAR (Quantitative Structure-Activity Relationship) model is one of the popular tools that are utilized to monitor activity, property and toxicity of chemicals. The use of QSAR is not limited to material science but also drugs, pharmaceuticals and other fields as well. The overall process of QSAR can be described in Figure 2.19.

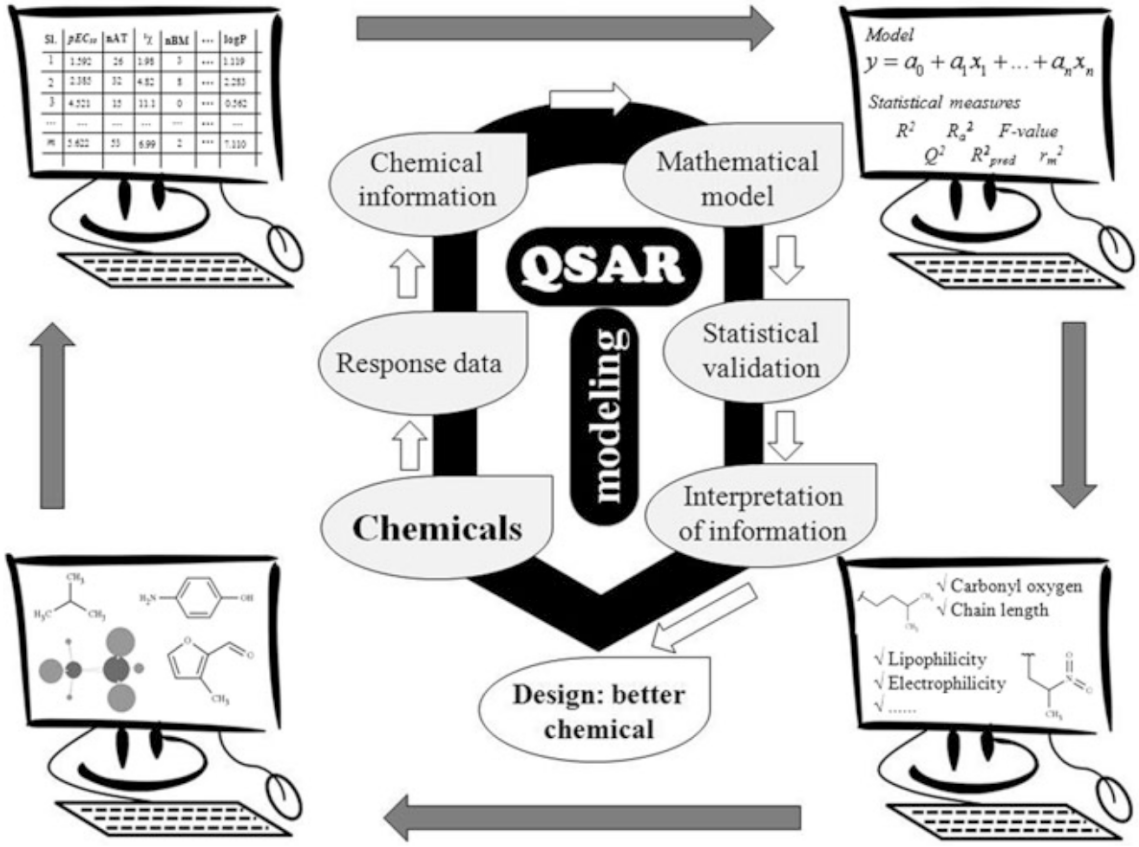


Figure 2.19: The overall scheme of QSAR [48].

QSAR is defined as the modeling on a set of structural chemicals that refers to the development of a mathematical correlation between a chemical response and quantitative chemical attributes defining the features of the analyzed molecules [48]. In other words, QSAR modeling originates from the concept of the correlation between response (material's properties, activities) and the chemical nature of molecules. Depend on the response (or endpoint) being property, activity or toxicity, there are different naming which are QSPR, QSAR, QSTR correspondingly. However, in this

part, we considering QSAR as the representation for those kinds of modeling. The basic formalization of QSAR technique can be described as the following.

$$\text{Biological activity} = f(\text{Chemical attributes})$$

The chemical attributes so-called predictors are usually the information derived directly from the chemical structure and physicochemical information. The predictors are represented in the form of numerical quantities. Then QSAR can be described as predictive mathematical models that are developed to explore the knowledge of chemistry and biology in a rational way to meet the desired need of the chemicals. QSAR models could be regression or classification models. An example of QSAR regression model can be seen in the following equation where Y is the response, $\{X_1, ..., X_n\}$ are descriptors and $\{a_1, ..., a_n\}$ are the contribution (weight) of each descriptor to the response.

$$Y = a_0 + a_1X_1 + a_2X_2 + a_3X_3 + ... + a_nX_n$$

In reality, the modeling of materials could be very complex with non-linear relations and thousands of predictors that make the process become a black-box. Several different methods could be used to model those relations and predictors including statistical methods including machine learning techniques or specific-designed QSAR models. Currently, deep neural networks show prominent results in QSAR studies [49], however, authors also mentioned that arcane descriptors and black-box models can affect the interpretability of the generated results. In the work of [50], authors encouraged the development and employment of interpretable descriptors in order to improve the interpretability of the model while still preserve accurate predictive capability. Moreover, there is another trend of implementing interpretable models that can provide explanations for the structure-property/activity/toxicity relationships as well as be able to achieve a reasonable accuracy.

Interpretable Credit Application Predictions With Counterfactual Explanations [51]

Lending is a massive business and credit risk assessment is one of the most important tasks in the business to decide whether to provide or reject loans to borrowers. Specifically, when an individual or business applies for a loan, the lender has to evaluate whether the business can reliably repay the loan principal and interest. For a business, there are two measures which commonly used to assess the credit risk: profitability and leverage [52]. For individual applications, several factors could be considered such as their income, real estate, etc. The more data collected about the borrowers, the better lenders can assess their creditworthiness. The choice of applications with low credit risk can get complicated when more factors are incorporated

which makes the assessment task become challenging.

Currently, with the abundance of relevant data as well as the development of machine learning and data mining techniques, credit risk assessment tasks can be automated and significantly improved. Tradition indices or linear techniques that are usually utilized to assess loan applications such as the FICO index or logit model are interpretable, however, their performance is incomparable to other prominent ML techniques such as deep neural networks, random forest or SVM. Those ML techniques are powerful due to their capability of extracting non-linear relationships from the data, however, they are almost black-boxes and it is a nontrivial task to explain for their decisions.

In the work of [51], they proposed the use of counterfactual explanations for a black-box model used in the credit risk assessment application. The counterfactual explanations can be provided for not just rejected cases but also positive (accepted) cases. Moreover, the counterfactual explanations are kept interpretable with a weighting scheme for attributes of a specific case. Specifically, they used the counterfactual generating process proposed by [53] with several adjustments to provide positive counterfactuals and shorter explanations. Basically, the counterfactual explanations are generated by calculating the smallest possible change that can be made to input such that the outcome change to the target class. Specifically, the following loss function $\mathcal{L}$ will be minimized to obtain counterfactual explanations.

$$\mathcal{L}(x, x', y', \lambda) = \lambda(\hat{f}(x') - y')^2 + d(x, x')$$

$$\underset{x'}{argmin}\underset{\lambda}{max}\mathcal{L}(x, x', y', \lambda)$$

where x is the input instance, x' is the closet instance to x that would change the outcome of the model f from y to y', $\hat{f}$ is the trained model, λ is the balance weight that ensure the desired output with smallest change to the input instance. Also $d(x, x')$ is the distance function between two instances which is the Manhattan distance weighted feature-wise with the inverse median absolute deviation (MAD) that is defined as the following.

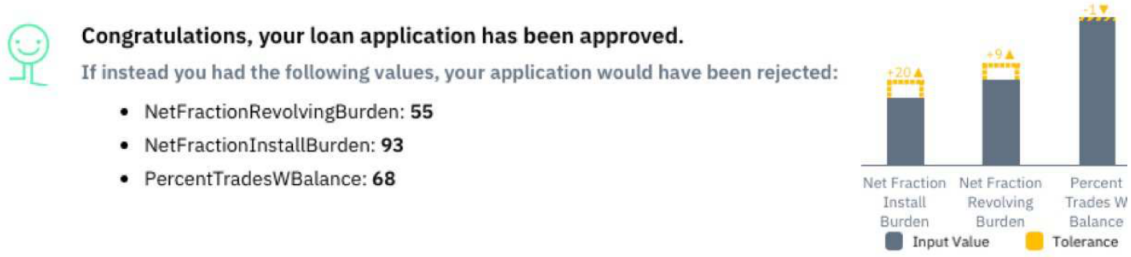
$$d(x, x') = \sum_{j=1}^{p} \frac{|x_j - x'_j|}{MAD_j}$$

The positive counterfactuals can be generated by setting the target y' to be the decision boundary such as $P(y = 1) = 0.5$ in the case of binary classification. Moreover, shorter counterfactual explanations can be generated by adding a weight

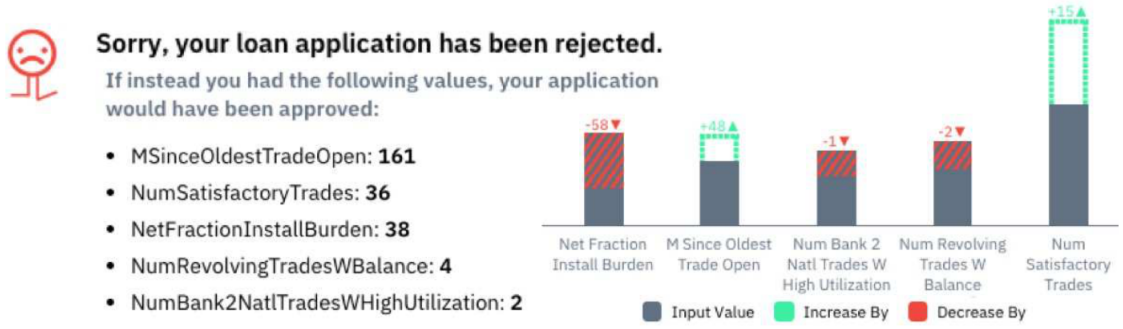
vector θ to the distance function as the following where the weight vector can be generated by the global feature importance or nearest neighbor approaches.

$$d_2(x, x') = \sum_{j=1}^{p} \frac{|x_j - x'_j|}{MAD_j} \theta_j$$

More details about this method can be found in [51]. An example of the generated counterfactual explanations is illustrated in Figure 2.20.



(a) Positive counterfactual explanation



(b) Counterfactual explanation

Figure 2.20: An illustration of a positive and negative counterfactual explanation. For the positive counterfactual explanation (a), it show the amount of tolerances (highlighted in yellow). While the negative counterfactual explanation (b) provides reasons for the loan rejection with suggestions on how to improve the results with the increase (green, dashed) or decrease (red, striped) of each feature [51].

Chapter 3

Improving The Efficiency of Interpretable Unsupervised Learning

Interpretable ML techniques are essential for applications in high-stakes decision-making fields such as healthcare, medicine and finance. However, there is an observation that ML techniques with a high degree of explainability are usually suffered from the performance problem when compared with other black-box models such as deep neural networks or random forests which possess prominent performance [4]. In an effort to mitigate the gap between interpretable ML models with black-box models in terms of performance while still preserve a reasonable level of interpretability, we investigate into popular clustering techniques and leverage our intuition about the information of hidden relationships in the underlying data to improve their performance. The content of this chapter is a part of our work that has been published in [54].

3.1 Background

In a wide range of fields, clustering is usually used as a popular method, especially in machine learning and data mining. Generally, clustering groups data into clusters where objects belong to the same cluster are similar to each other and different to objects in other clusters [55]. There are two main types of clustering techniques: hierarchical and partitional clustering [56]. In real-world applications, partitional methods are common due to their effectiveness in solving clustering problems with scalability. The k-means [57] is the most well-known and widely-used partitional-based method due to its intuitive clustering process as well as low computational

complexity. However, an inherent limitation of this approach is its data type constraint, as the k-means can only work with the numerical data type.

Many efforts have been made in order to mitigate the aforementioned limitation of k-means to allow it to cluster categorical data. Specifically, several k-means like methods for categorical data have been proposed such as k-modes [58], k-representatives [59], k-centers [60] and k-means like clustering algorithm [61]. Those clustering techniques implement a similar clustering fashion to the k-means method, however, they vary in the definitions of cluster mean and dissimilarity measure for categorical data.

Regardless of the important role of dissimilarity measures in clustering techniques, metrics to quantify the resemblance between categorical values are still not well-understood as no coherent metric is available for categorical data so far. One common method is encoding categorical data as numerical values so-called dummy coding (or indicator coding) [62, 63]. Particularly, in this method, binary values are used to indicate whether a categorical value is present or absent in the original data. However, by encoding each category as an independent variable, various significant features and characteristics of categorical data such as the distribution of categories or their mutual relationships may not be taken into account.

Moreover, the necessity of considering that information in order to quantify the resemblance between categorical values has been emphasized in literature [61, 64]. Particularly, for the problem of clustering with categorical data, most previous works have unfortunately neglected the semantic information potentially inferred from relationships among categories. In this chapter, we introduce a new clustering framework for categorical data that is capable of integrating not only the distributions of categories but also their mutual relationship information into the pattern proximity evaluation process of the clustering task. The effectiveness of the proposed clustering technique is then proven by a comparative study conducted on existing clustering methods for categorical data collected from UCI Machine Learning Repository [65].

The remaining part of this chapter is organized as follows. In the second section, we describe in detail our proposed clustering framework for categorical data. In the third section, information about the experimental evaluation that is conducted to prove the merits of our proposed method will be provided. Finally, we summarize the work in the last section.

3.2 The Proposed Clustering Framework - *RICS*

In this section, we introduce a new clustering framework, namely $RICS$, that can be able to integrate the mutual relationship information between categorical attributes

into the clustering process. The details of our proposed clustering technique are divided into two parts. The first part introduces core concepts and structure of a k-means like clustering algorithm. In the second part, we propose a new dissimilarity measure for categorical data. Before going into the details of the clustering algorithm, we would like to provide several notations that will be used in the rest of this chapter.

3.2.1 Notations

Given a categorical data set X that contains n instances and is described by d attributes. The notations used in the rest of this chapter are presented in the following.

- A feature (attribute) of X is denoted by A_j, $j \in \{1, ..., d\}$. For each A_j, its domain is denoted by $dom(A_j)$. Moreover, each value of A_j is denoted as a_l (or simply a) with $l \in \{1, ..., |dom(A_j)|\}$.
- An instance of X is presented as a vector $x = [x_1, ..., x_d]$ where the value of x at an attribute A_j is denoted as x_j, $j \in \{1, ..., d\}$.
- The frequency of $a_l \in dom(A_j)$ is denoted as $P(a_l)$ and calculated by

$$P(a_l) = \frac{count(A_j = a_l|X)}{|X|} \tag{3.1}$$

  similarly, for $a_l \in dom(A_j)$ and $a_{l'} \in dom(A_{j'})$ we have

$$P(a_l, a_{l'}) = \frac{count((A_j = a_l) \text{ and } (A_{j'} = a_{l'})|X)}{|X|} \tag{3.2}$$

3.2.2 A k-Means Like Clustering Framework

The clustering method introduced in this chapter basically implements the general structure of the k-means like clustering scheme [61]. Particularly, it still preserves the general process of the k-means method but comprises a modified concept of cluster centers based on the work of Chen et al. [60] and a weighting method for each categorical attribute.

Representation of Cluster Centers

Let $C = \{C_1, ..., C_k\}$ be the set of k clusters of X, for any two different clusters C_i and $C_{i'}$ we have

$$C_i \cap C_{i'} = \emptyset \text{ if } i \neq i' \text{and } X = \bigcup_{i=1}^{k} C_i \tag{3.3}$$

Furthermore, for each cluster C_i, the center of C_i is defined as

$$V_i = [v_{i1}, ..., v_{ij}, ..., v_{id}] \tag{3.4}$$

where v_{ij} is a probability distribution on the domain of an attribute A_j that is estimated by a kernel density estimation function K.

$$v_{ij} = [p(a_1), ..., p(a_{|dom(A_j)|})] \tag{3.5}$$

where

$$p(a_l) = \sum_{a \in dom(A_j)} f_i(a) K(a|\lambda_j) \tag{3.6}$$

with $f_i(a)$ is the frequency probability of an attribute value a in the cluster V_i.

$$f_i(a) = \frac{count(A_j = a|V_i)}{|V_i|} \tag{3.7}$$

Moreover, consider σ_{ij} as the set that contains all available values of attribute A_j that exist in cluster V_i

$$\sigma_{ij} = \{a, a \in dom(A_j)|V_i\} \tag{3.8}$$

then the kernel function $K(a|\lambda_j)$ to estimate the probability of those attribute values in cluster V_i is defined as

$$K(a|\lambda_j) = \begin{cases} 1 - \frac{|\sigma_{ij}|-1}{|\sigma_{ij}|}\lambda_j & \text{if } a = a_l \\ \frac{1}{|\sigma_{ij}|}\lambda_j & \text{if } a \neq a_l \end{cases} \tag{3.9}$$

where λ_j is the smoothing parameter for C_j and has the value range of $[0, 1]$. In order to select the best parameter λ_j, the least squares cross validation (LSCV) method [60] is utilized. In the case $a \notin \sigma_{ij}$, $K(a|\lambda_j)$ value is set to 0.

Finally, from (3.4)-(3.6), we have the general formulation to compare the dissimilarity between a data instance $x \in X$ and a cluster center V_i described as below.

$$D(x, V_i) = \sum_{j=1}^{d} d(x_j, v_{ij}) = \sum_{j=1}^{d} \sum_{a \in dom(A_j)} p(a) \times dis(x_j, a) \tag{3.10}$$

where $dis(x_j, a)$ is the measure to quantify the dissimilarity between two values of an attribute A_j. Detailed information about this measure will be described in subsection 3.2.3.

Weighting Scheme for Categorical Attributes

We applied a weighting scheme for categorical attributes where a larger weight is set to attributes that have a smaller sum of within-cluster distances and vice versa. More details of the weighting method can be found in [66].

Specifically, a vector of weights $W = [w_1, ..., w_d]$ will be assigned to each attribute where $w_j \leq 1$ and $\sum_{j=1}^{d} w_j = 1$.

The weighted dissimilarity measure between a data instance x and a cluster center V_i the can be defined as

$$D_w(x, V_i) = \sum_{j=1}^{d} w_j \times d(x_j, v_{ij}) \tag{3.11}$$

Based on these definitions, the clustering algorithm now aims to minimize the following objective function:

$$J(U, V, W) = \sum_{i=1}^{k} \sum_{g=1}^{n} \sum_{j=1}^{d} u_{i,g} \times w_j \times d(x_j, v_{ij}) \tag{3.12}$$

subject to

$$\begin{cases} \sum_{i=1}^{k} u_{i,g} = 1 & 1 \leq g \leq n \\ u_{i,g} \in \{0, 1\} & 1 \leq g \leq n,\ 1 \leq i \leq k \\ \sum_{j=1}^{d} w_j = 1 & 0 \leq w_j \leq 1 \end{cases} \tag{3.13}$$

where $U = [u_{i,g}]_{n \times k}$ is the partition matrix. The algorithm for the k-means like clustering framework is described in Algorithm 1.

Algorithm 1. ***k*-means like-clustering framework [61]**

Input: Data set $X = \{x_1, ..., x_n\}$
Output: Optimized clusters $C = \{C_1, ..., C_k\}$
1: Initialize centers for k clusters $V = [V_1, ..., V_k]$.
2: Initialize weights $W = [w_1, ..., w_d]$ and set $\lambda = 0$ for each attribute.
3: ***do***
4: Keep V and W fixed, generate U to minimize the distances between objects and cluster centers using Eq. (3.11).
5: Keep U fixed, update V using Eq. (3.5) and Eq. (3.6).
6: Generate W using formulas from [66].
6: ***while*** partitions still change.

3.2.3 A Context-Based Dissimilarity Measure for Categorical Data

In distance-based clustering methods, dissimilarity measures play a key role in their performance. In our work, for measuring the dissimilarity between categorical values, an extended version of the similarity measure proposed in [54] will be introduced with the capability of integrating not only the distribution of categories but also their mutual relationship information. To that end, the amount of information to describe the appearances of pairs of attribute values will be considered rather than merely information about single values. In order to reduce the computational cost, only pairs of attributes that are highly correlated with each other are selected.

Correlation Analysis for Categorical Attributes

In order to extract pairs of highly correlated attributes, we adopted the interdependence redundancy measure proposed by Au et al. [67] to quantify the dependency degree between each pair of attributes. Specifically, the interdependence redundancy value between two attributes A_j and $A_{j'}$ is quantified as in the following formula.

$$R(A_j, A_{j'}) = \frac{I(A_j, A_{j'})}{H(A_j, A_{j'})} \tag{3.14}$$

where $I(A_j, A_{j'})$ denotes the mutual information [68] between attribute A_j and $A_{j'}$ and $H(A_j, A_{j'})$ is their joint entropy value. We have the formulas for those measures as the followings.

$$I(A_j, A_{j'}) = \sum_{p=1}^{|dom(A_j)|} \sum_{q=1}^{|dom(A_{j'})|} P(a_{jp}, a_{j'q}) * \log \frac{P(a_{jp}, a_{j'q})}{P(a_{jp}) * P(a_{j'q})} \tag{3.15}$$

$$H(A_j, A_{j'}) = - \sum_{p=1}^{|dom(A_j)|} \sum_{q=1}^{|dom(A_{j'})|} P(a_{jp}, a_{j'q}) * \log P(a_{jp}, a_{j'q}) \tag{3.16}$$

According to Au et al. [67], the interdependency redundancy measure has the value range of $[0, 1]$. A large value of R implies a high degree of dependency between attributes.

For each attribute A_j, in order to select its highly correlated attributes, a relation set is defined and denoted as S_j. Specifically, S_j contains attributes whose the associated interdependency redundancy values with A_j are larger than a specific threshold γ.

$$S_j = \{A_{j'} | R(A_j, A_{j'}) > \gamma, 1 \leq j, j' \leq d\} \tag{3.17}$$

New Dissimilarity Measure for Categorical Data

For integrating the relationship information that is contained in the set S_j, the conditional probability of correlated attributes values is utilized to include the mutual relationships between categorical attributes. In particular, to quantify the similarity between categorical values of attribute A_j, the following measure is implemented.

$$sim(x_j, x'_j) = \sum_{A_{j'} \in S_j} \sum_{a \in dom(A_{j'})} \frac{1}{|S_j|} \times \frac{1}{|dom(A_{j'})|} \times \frac{2 \times \log P(\{x_j, x'_j\}|a)}{\log P(x_j|a) + \log P(x'_j|a)} \quad (3.18)$$

It could be easily seen that the similarity measure in Eq. (3.18) have the value range of $[0, 1]$. Specifically, when x_j and x'_j are identical, their similarity degree is equal to 1. Then, the dissimilarity measure between two values of an attribute that is used in Eq. (3.10) could be defined as below.

$$dis(x_j, x'_j) = 1 - sim(x_j, x'_j) \quad (3.19)$$

The extended dissimilarity measure defined in Eq. (3.19) satisfies the following conditions:

1. $dis(x_j, x'_j) \geq 0$ for each x_j, x'_j with $j \in \{1, ..., d\}$
2. $dis(x_j, x_j) = 0$ with $\forall j \in \{1, ..., d\}$
3. $dis(x_j, x'_j) = dis(x'_j, x_j)$ for each x_j, x'_j with $j \in \{1, ..., d\}$

For reducing the computational time of the proposed algorithm, the generation of the relation set for each feature will be carried out in advance. Furthermore, the degree of resemblance between attribute values will also be assessed beforehand and their values will be kept in a multi-dimensional matrix for later referring. The details of the $RICS$ algorithm are described in Algorithm 2.

3.3 Experimental Evaluation

In order to prove the merits of our proposed clustering method, a comparative experiment will be conducted on popular baseline clustering methods that can process categorical data. Particularly, the newly proposed clustering framework $RICS$ will be contrasted with the k-modes method [58], k-representatives [59] and k-means like clustering framework [61]. Each method will be executed 300 times per data set. The values of the hyper-parameters of our method will be set as follows. The threshold γ will be set to the value of 0.1 as it has been observed to produce generally good results. For the value of parameter k, it will be assigned to the same number of classes in each data set. The clustering results will be evaluated by three common measures which are Purity metric, Normalized mutual information (NMI) score and Adjusted Rand index (ARI). Details of those metrics can be found in [54]. The final results for three measures will be calculated by averaging the results of 300 running times.

Algorithm 2. *RICS* clustering framework

Input: Data set $X = \{x_1, ..., x_n\}$
Output: Optimized clusters $C = \{C_1, ..., C_k\}$
1: Generate relation set S_j for all attributes A_j using Eq. (3.14)-(3.17).
2: Precompute dissimilarity value $dis(a_l, a_{l'})$ for all $a_l, a_{l'} \in dom(A_j)$ with $j \in \{1, ..., d\}$ using Eq. (3.18), (3.19).
3: Initialize centers for k clusters $V = [V_1, ..., V_k]$.
4: Initialize weights $W = [w_1, ..., w_d]$ and set $\lambda = 0$ for each attribute.
5: ***do***
6: Keep V and W fixed, generate U to minimize the distances between objects and cluster centers using Eq. (3.11).
7: Keep U fixed, update V using Eq. (3.5) and Eq. (3.6).
8: Generate W using formulas from [66].
9: ***while*** partitions still change.

3.3.1 Testing Data Sets

In this experiment, we collect 14 data sets from the UCI Machine Learning Repository [65]. All of the selected data sets contain a variety of data types including categorical, integer and real values. In order to handle numerical values, a discretization tool from Weka [69] will be utilized to generate discrete categories that contain a certain range of equal intervals. Then those intervals will be treated normally as categorical values. Furthermore, the average dependency degree of each data set is estimated by averaging the interdependency redundancy values of all distinct pairs of attributes based on Eq. (3.14). Details of collected data sets are described in Table 3.1.

Table 3.1: Details of data sets for the experiment that are collected from UCI

Dataset	Inst.	Attr.	Classes	Data types	Avg. dependency degree
soybean	307	35	19	Categorical	0.153
hayes-roth	160	5	3	Categorical	0.113
wine	178	13	3	Integer, Real	0.089
voting-records	435	16	2	Categorical	0.085
dermatology	366	33	6	Categorical, Integer	0.052
breast-cancer	286	9	2	Categorical	0.027
post-operative	90	8	3	Categorical, Integer	0.014
chess	3196	36	2	Categorical, Integer	0.010
tictactoe	958	9	2	Categorical	0.006
splice	3190	61	3	Categorical	0.003
car	1728	6	4	Categorical	0
lenses	24	4	3	Categorical	0
nursery	12960	8	5	Categorical	0
balance-scale	625	4	3	Categorical	0

3.3.2 Experimental Results

As can be observed from the experimental results displayed in Tables 3.2, 3.3 and 3.4, there is no method that can outperform others for all of the testing data sets. However, it is noticeable that our proposed clustering framework $RICS$ has achieved comparative results while performed well in most of the data sets and for all three evaluation metrics. Specifically, it worked effectively for highly correlated data sets such as dermatology, hayes-roth, soybean or wine . Furthermore, the overall results for all three metrics have shown that our proposed framework has the best average results.

Particularly, by inspecting carefully into results on the purity metric in Table 3.2, the k-modes algorithm seems to outperform k-representatives and k-means like clustering method, and has a comparative performance with $RICS$. However, when combining the information that is extracted from the results on NMI and ARI metrics, k-modes actually has poor performances regarding those two more significant standards, while $RICS$ is still can achieve the best results over the total of 14 testing data sets.

Table 3.2: The purity results of the clustering experiment on 14 testing data sets

Data sets	RICS	k-means like framework	k-representatives	k-modes
soybean	**0.7176**	0.7142	0.7152	0.6099
hayes-roth	0.3954	0.3953	0.3998	**0.4079**
wine	**0.9397**	0.9214	0.9380	0.7707
voting-records	**0.8770**	0.8760	0.8764	0.8581
dermatology	0.8560	0.8506	**0.8593**	0.7116
breast-cancer	**0.7028**	**0.7028**	**0.7028**	**0.7028**
post-operative	**0.7111**	**0.7111**	**0.7111**	**0.7111**
chess	0.5223	0.5225	0.5222	**0.5761**
tictactoe	0.6534	0.6534	0.6534	**0.6558**
splice	**0.7586**	0.7572	0.6159	0.5188
car	**0.7150**	0.7059	0.7046	0.7004
lenses	0.6999	0.6981	**0.7018**	0.6446
nursery	0.4449	0.4502	0.4324	**0.4704**
balance-scale	0.5779	**0.5787**	0.5761	0.5496
Average	**0.6837**	0.6812	0.6721	0.6348

Table 3.3: The NMI results of the clustering experiment on 14 testing data sets

Data sets	RICS	k-means like framework	k-representatives	k-modes
soybean	0.7517	0.7473	**0.7545**	0.6069
hayes-roth	0.0041	0.0038	0.0011	**0.0050**
wine	0.7893	0.7580	**0.7941**	0.4252
voting-records	**0.5055**	0.5002	0.4990	0.4359
dermatology	**0.8551**	0.8512	**0.8551**	0.5735
breast-cancer	**0.0041**	0.0040	0.0018	0.0038
post-operative	0.0146	0.0140	0.0198	**0.0243**
chess	0.0006	0.0007	0.0002	**0.0187**
tictactoe	0.0346	**0.0393**	0.0087	0.0206
splice	**0.4620**	0.4592	0.2820	0.0473
car	**0.1435**	0.1234	0.1213	0.0475
lenses	**0.3444**	0.3442	0.3432	0.1880
nursery	0.0947	**0.1038**	0.0855	0.0601
balance-scale	0.0485	**0.0491**	0.0474	0.0313
Average	**0.2895**	0.2856	0.2724	0.1777

Table 3.4: The ARI results of the clustering experiment on 14 testing data sets

Data sets	RICS	k-means like framework	k-representatives	k-modes
soybean	0.4642	0.4655	**0.4754**	0.3748
hayes-roth	**-0.0102**	-0.0105	-0.0138	-0.0111
wine	**0.8200**	0.7721	0.8145	0.4287
voting-records	0.5642	0.5644	**0.5658**	0.5119
dermatology	**0.7494**	0.7421	0.7389	0.5503
breast-cancer	0.0018	0.0015	-0.0030	**0.0020**
post-operative	**-0.0105**	-0.0113	-0.0110	-0.0178
chess	-0.0001	0.0001	-0.0003	**0.0238**
tictactoe	0.0325	**0.0380**	0.0218	0.0247
splice	**0.3927**	0.3900	0.2021	0.0289
car	0.0555	**0.0598**	0.0537	0.0239
lenses	**0.2108**	0.2075	0.1835	0.0596
nursery	0.0578	**0.0637**	0.0559	0.0506
balance-scale	0.0507	**0.0522**	0.0505	0.0323
Average	**0.2413**	0.2382	0.2239	0.1487

3.4 Summary

In this chapter, we have introduced a new clustering method for categorical data. The proposed method can integrate both the information about the distributions of categories as well as their mutual relationships into the quantification of dissimilarity between data objects. The merits of the proposed method have been proved by an extended experiment. Particularly, $RICS$ has a competitive performance or even outperforms other baseline methods. The proposed clustering framework still preserves the general structure of k-means method which makes it interpretable. Moreover, the information about the relationships between features can also be visualized to provide users with more intuitive insights about the underlying data.

Chapter 4

Transparent Supervised Learning Instead of Black-Box Models

Machine learning and data mining techniques increasingly play a significant role in society under various forms of hardware and software applications. Those applications can exist in phones, smartwatches, cars or even at your home and have profound impacts on our private life as well as our work. One of the main reasons for their development is the improvement in the performance with complex models built on the base of a huge amount of collected data. Those models are designed to learn non-linear relationships between the underlying data and provide the predictions based on those learn information. However, little knowledge about what they have learned can be revealed for their users - which are us - due to the complexity of their working mechanism. This situation leads to several limitations including the hardness when debugging a certain problem produced by the systems or gaining the trust of users by proving their fidelity and fairness. Especially, in many domains that require transparency and interpretability such as medicine or healthcare, the characteristics of those models - usually named as *black-box* - become an important limitation when considering their adoption in those fields.

In this chapter, we introduce a two-stage binary classification system that can be applied for healthcare as well as general data. The proposed system still preserves a proper level of interpretability and can also achieve comparative results with popular classification techniques. The motivation behind the proposed system is the lack of effective classification methods for handling data generated by various distributions (such as healthcare or finance data) that can harmonize both performance and interpretability perspectives. In this work, we handle the problem by utilizing the divide and conquer strategy with a new disentangled representation of the underlying data. The content of this chapter is a part of our work that has been published in [70].

4.1 Background

The field of knowledge discovery in data (KDD) has advanced rapidly in recent times and had a significant impact on almost every aspect of our life. Its ubiquity can be seen from a wide range of applications in private portable devices to household electronic equipment and machines used in workplaces. One of the main reasons for its popularity is the advancement of machine learning (ML) and data mining techniques based on big data resources. For several ML tasks such as recognition or prediction, implementing complex ML techniques (so-called black-box) can help to achieve prominent performance. However, it also comes with the price of high complexity and opaqueness in those models [1].

Specifically, with the usage of those black-box models, their interpretability has also deteriorated. Especially, the problem becomes more serious with their applications in decision support systems that are usually implemented in crucial fields such as medicine, healthcare or finance - which require a proper level of transparency for making high-stakes decisions [2]. In order to resolve those limitations, efforts have been put into research in the field of general explanatory artificial intelligence (XAI) or specific interpretable ML. Particularly, in XAI-related research, transparency is enforced in whole or parts of systems with the requirement of explainability in their decisions. With the implementation of transparency and explainability, it helps to ensure the fidelity and fairness of the systems [3].

For the definition of interpretable ML, as defined in [4], it is the utilization of ML techniques for extracting relevant domain knowledge about relationships hidden inside the data. Knowledge can be considered as being relevant when it can provide insights to guide further communication, actions and discovery. Moreover, interpretability is regarded as the first step to guarantee the explainability of the models with the capability to defend their decisions, provide relevant responses to questions and be audited [3]. For many research in the field of XAI and interpretable ML, the terms of explainability and interpretability have slightly different meanings, however, can be used interchangeably.

In real-world applications, there are mainly two situations where the interpretability of systems is essential. The first situation is the need to debug the problems that are caused by ML models. Particularly, several examples of unexpected behaviors or intentionally hidden information embedded into ML models have been pointed out in the work of [3]. Some of them can be listed as the case of recurrent neural networks that can misclassify the same images which have been slight adjusted with perturbation methods [5]. Another example was described in [6] where deep neural networks can simply be tricked into the misclassification of inputs which are not similar to their true classes. Along with more findings on adversarial examples of ML models are revealed, it emphasizes the need to improve the

interpretability of those models so that users can understand and give their trust in ML model-based decision-making processes as well as detecting problems.

The second situation that interpretability is considered to be essential which is the applications of ML models in crucial fields such as healthcare or finance where they need to make high-stake decisions. In those fields, the interpretability of ML systems is regarded as the fundamental condition for implementing them in actual applications [8]. Furthermore, literature in the fields also emphasizes the indispensable role of interpretable ML systems. Particularly, in the work of [9], they argued that the clinical decisions for radiation treatment must not be based only on the accuracy of the prediction system but also on an informed understanding of the relationship among patients' characteristics, radiation response and treatment plans. Also in [10], the challenge was observed for the implementation of artificial neural networks for predicting medical outcomes comparing with the utilization of logistic regression for the same problem. In the review of applications of deep neural networks in health informatics [11], it was pointed out that the shortage of interpretability is the main reason that restricts the adaptation of neural networks into the healthcare sector. Also providing explanations for ML black-box models is considered as a significant challenge for the medical field [12].

Alongside the aforementioned technical and ethical requirements for the interpretability of ML models, with the increasing influence of ML applications in daily life, the community is realizing the potential problems and pushing efforts into the enforcement of interpretability and explainability of those applications. Those efforts are expressed in several legal documents such as the European Union directive for General Data Protection Regulation (GDPR). Particularly, GDPR defines the right of explanation as providing an individual with "meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject" [13]. Furthermore, other GDPR-like laws are also being adopted in various regions that can be listed as the California Consumer Privacy Act (CCPA) or the Privacy Amendment (Notifiable Data Breaches) to Australia's Privacy Act.

Due to the increasing demand of ML models that are both interpretable and can achieve high performance, especially in the field of healthcare applications, in this chapter we introduce a classification system named *GSIC* (GSOM-based Interpretable Classifying System). The proposed system is developed based on a systematic combination of unsupervised and supervised ML techniques that reflects our intuition about the distribution of the underlying data. Particularly, new data representations generated by GSOM (The Growing Self-Organizing Map)[32] play a key role to help overcome the curse of dimensionality problem as well as improve the efficiency and interpretability. GSOM is a popular dimensional reduction and visualization method that has the ability to dynamically learn new data representation and reveal salient relationships from underlying data. To prove the merits

of our proposed system, an experiment on the classification task will be conducted along with the demonstration on a use case on specific data set of sepsis patients in the Intensive Care Unit (ICU).

The remaining part of this chapter is described as follows. In Section 4.2, we represent the details of our proposed classification system *GSIC*. In Section 4.3, information about the classification experiment will be provided to evaluate the merits of our proposed system including a detailed analysis of sepsis data use case. Finally, in Section 4.4, we summarize our work and have a discussion about the limitations and future works.

4.2 A Transparent Classification System for Knowledge Discovery

4.2.1 Notations

We would like to provide notations that will be used to formulate our proposed system in the next subsection. A data set D comprises n instances and is described by l attributes. Consequently, $D = \{x_1, x_2, ..., x_n\}$. In order to refer to a specific instance and attribute, x and att notations can be utilized respectively. The GSOM mapping of D can be denoted as M that consists g codebook vectors (also called neurons or nodes) $M = \{m_1, m_2, ..., m_g\}$. The set of k clusters on M is represented as $C = \{C_1, C_2, ..., C_k\}$. Furthermore, $DT = \{DT_1, DT_2, ..., DT_k\}$ is the set containing decision trees generated for each cluster in C.

4.2.2 GSOM-based Interpretable Classification System (GSIC)

In this work, we introduce a new binary classification system based on the combination of GSOM representation and traditional machine learning methods to form a robust and intelligible learning model. The proposed system GSIC is inspired by the cognitive process of human thinking for the classification task by forming concepts (prototypes) and build discrete rules under each concept to be able to identify specific objects [71]. The merits of our proposed system two folds. Firstly, semantic groups of original data can be formed and analyzed in order to provide insights into the underlying data. Secondly, the computational cost of our system can be reduced and distributed depending on the number of generated groups by doing the classification on separate clusters instead of the whole data set. Details of GSIC are described as the following.

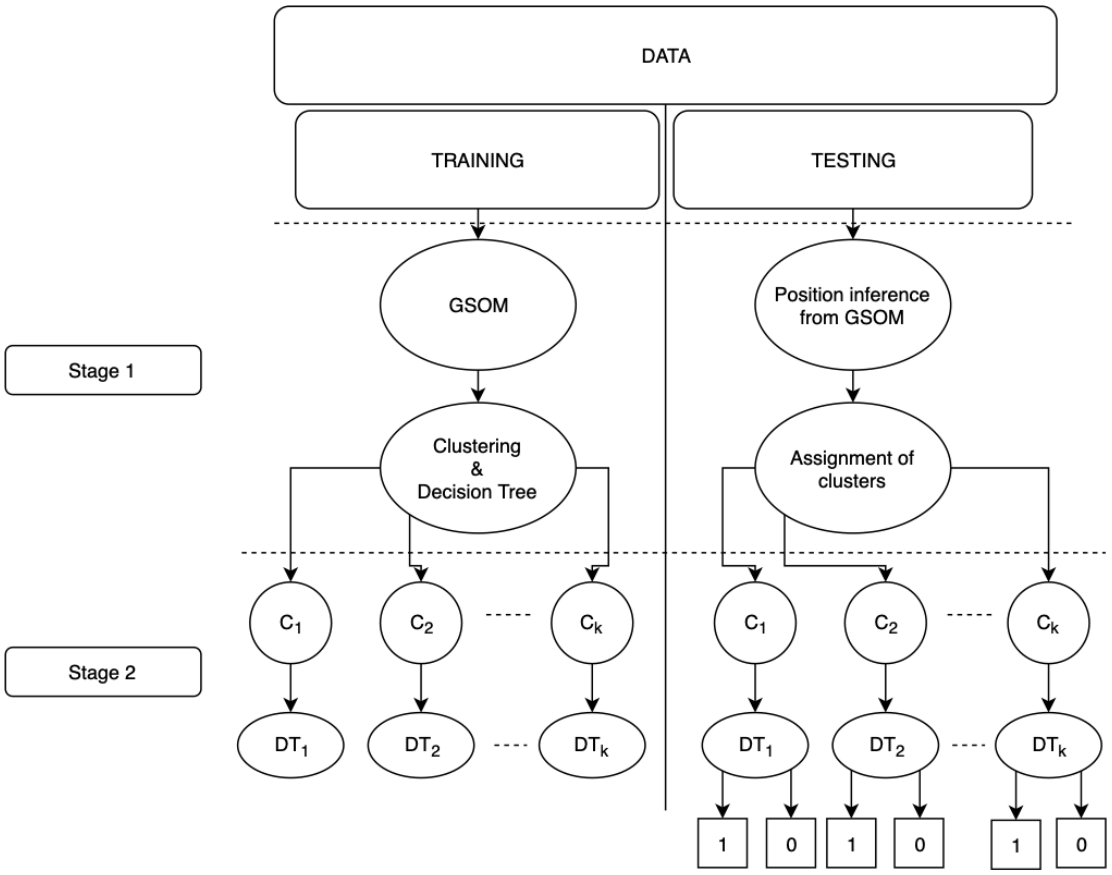


Figure 4.1: Illustration for the general structure of *GSIC*

In the first step of GSIC, a new representation of the original data will be generated for dimensionality reduction as well as revealing hidden relationships. Specifically, we implement the GSOM algorithm [32] - a variant of SOM (Self-Organizing Maps) [72] - for fulfilling the target as its capability to produce self-organizing semantic representations of input data dynamically. Particularly, in new representations, original similar data instances will be mapped to nearby points in 2D space. A notable example from GSOM can be found in Chapter 2. To formulate the problem, a set M containing four codebook vectors will firstly be initialized and denoted as $M = \{m_1, m_2, m_3, m_4\}$. Subsequently, the resemblance between each data instance x and codebook vectors will be estimated using Euclidean distance. Based on their similarity, data instances are assigned to their closest vector m_c (winner node) with $i, c \in \{1, ..., 4\}$.

$$m_c = \underset{i}{argmin}(Euclidean(x, m_i))$$

After the assigning step, the values of the winner node and its neighbors will be updated to reduce the quantization error (the distance between the data instance and the winner node) as described in Eq. (4.1). Consequentially, the winner node

and its neighbors will be closer to assigned data instances after the updating step. In the following formulation, t is the timestamp, α is the learning rate that decreases over each step and N is the set of neighbors of m_c.

$$m_i(t+1) = \begin{cases} m_i(t), & i \notin N_{t+1} \\ m_i(t) + \alpha(t) * (x - m_i(t)), & i \in N_{t+1} \end{cases} \tag{4.1}$$

In several situations, the so-called under-representation problem can happen when a large number of data instances concentrate on a single node in the map. To resolve the problem, new nodes will be grown from the overpopulated node so that data instances can be mapped more adequately. Practically, a value named maximum error H_E plays the role of tracking the highest error of generated nodes in the map M. The error E_c of node m_c is defined as the accumulated distance between it and the assigned data instance x.

$$E_c(t+1) = E_c(t) + ||x - m(t)|| \tag{4.2}$$

$$H_E = \underset{i}{argmax}(E_i)$$

The need of growing new nodes will be set through a threshold value so-called "Grow Threshold" (GT) which is computed in advance based on values of data set D. Specifically, new nodes will be created as direct neighbors of the node that satisfy the condition $H_E > GT$. Values of new nodes will be initialized to harmonize with their neighborhood regarding the smoothness of map M. The process of fitting the data is repeated for every instance in D until there are no changes in the map M (no new nodes are generated).

The result of GSOM is the self-organized map M which contains g codebook vectors $M = \{m_1, m_2, ..., m_g\}$ with $(g \geq i \geq 4)$. Each codebook vector is the representation of a Voronoi region in the original data space. Specifically, data instances assigned to the same codebook vector are similar to each other and adjacent codebook vectors share several close characteristics. To automatically identify those semantic groups of codebook vectors, the k-means method will be used to cluster the map M. Specifically, consider $C = \{C_1, ..., C_k\}$ as the set of k clusters extracted by k-means in M, for any two different clusters C_j and $C_{j'}$, we have

$$C_j \cap C_{j'} = \emptyset \text{ if } j \neq j' \text{and } M = \bigcup_{j=1}^{k} C_j \tag{4.3}$$

For each cluster C_j, the center of C_j is defined as

$$V_j = [v_{j1}, v_{j2}, ..., v_{jl}] \tag{4.4}$$

The resemblance between a codebook vector $m_i \in M$ and a cluster center V_j then can be estimated by utilizing the Euclidean distance as the following.

$$dis(m_i, V_j) = ||m_i - V_j|| \tag{4.5}$$

Based on Eq. (4.3) to (4.5), the k-means clustering algorithm aims to minimize the following objective function:

$$J(U, D) = \sum_{i=1}^{g} \sum_{j=1}^{k} u_{i,j} \times dis(m_i, V_j) \tag{4.6}$$

subject to

$$\begin{cases} \sum_{j=1}^{k} u_{i,j} = 1 & 1 \leq i \leq g \\ u_{i,j} \in \{0, 1\} & 1 \leq i \leq g,\ 1 \leq j \leq k \end{cases}$$

where $U = [u_{i,j}]_{g \times k}$ is the partition matrix.

In order to extract hidden characteristics inside each cluster, a decision trees algorithm is trained on data instances belonging to each cluster. In this work, we utilize the CART (Classification and Regression Tree) algorithm - a well-known binary decision tree learning algorithm proposed by [27]. The generation of decision trees on each cluster helps to reduce the computation cost as well as increasing the intelligibility of the model. Specifically, for a cluster $C_j \in C$, starting with all data instances belonging to C_j, the data will be split at the most informative feature by maximizing the information gain (IG) as in the Eq. (4.7).

$$IG(D_p, att) = I(D_p) - \left[\frac{N_{left}}{N_p} I(D_{left}) + \frac{N_{right}}{N_p} I(D_{right}) \right] \tag{4.7}$$

where att is the split feature, D_p and D_{left}, D_{right} are the data set of the parent, left and right child nodes respectively, I is the impurity measure, N_p is the total number of samples at the parent node, and N_{left}, N_{right} are the number of samples in the left and right child node.

The information gain indicates the difference between the impurity of the parent node and the sum of the child node impurities—the lower the impurity of the child nodes, the larger the information gain. The CART algorithm utilizes the Gini index (GI) as the impurity measure. The Gini index is intuitively a criterion to minimize the probability of misclassification.

$$I_{GI}(t) = \sum_{i=1}^{c} p(i|t)(-p(i|t)) = 1 - \sum_{i=1}^{c} p(i|t)^2 \tag{4.8}$$

where $p(i|t)$ is the proportion of the samples that belongs to class c for a particular node t.

The splitting process will be conducted iteratively at each child node until the leaves are pure where samples at each node belong to only one class. The pseudo-codes of GSIC is provided in Algorithm 3.

Algorithm 3. GSIC algorithm

Input: Data set $D = \{x_1, x_2, ..., x_n\}$.
Output: A set of decision trees for inference task $DT = \{DT_j | j \in \{1, ..., k\}\}$.
Step 1: Generate GSOM of training data
1: Initialize 4 codebook vectors of the GSOM map $M = \{m_1, m_2, m_3, m_4\}$.
2. Calculate the growing threshold GT of dataset D.
3: For each x in D:
4: $dis = Eucl(x, m_i)$
5: Assign x to $m_c = \underset{i}{argmin}(dis)$
6: Update m_c and its neighbors value using Eq.4.1
7: Update error value E_c according to Eq.4.2.
8: If $E_c > GT$:
9: Expand new nodes from m_c
10: Repeat loop 3 until M has no changes.
Step 2: Finding clusters from the generated map
11: Input: generated map $M = \{m_1, m_2, ..., m_g\}$ where $g \geq 4$.
12: Initialize k cluster centers $V = \{V_1, V_2, ..., V_k\}$.
13: For each m_i in M:
14: Compute distances between m_i and each cluster center in V.
15: Assign m_i to the its nearest cluster using Eq.5.5.
16: Repeat loop 13 until no changes.
Step 3: Generating segmented decision trees for each cluster
17: For each $C_j \in C$:
18: Generate decision tree DT_j for cluster C_j.
19: Extract and sort decision rules from DT_j.

4.3 Experimental Evaluation

4.3.1 Testing Data Sets

Nowadays, healthcare data sets have become more relevant that facilitate the surge of researches and applications in this field. Currently, MIMIC III (Medical Informa-

tion Mart for Intensive Care III) [73] is considered as one of popular and large-scale data sets in healthcare field. MIMIC III was developed by MIT Lab and contains almost 60000 ICU admissions in the Beth Israel Deaconess Medical Center. In this research, only a subset of MIMIC data set will be utilized for the testing purpose of the classification task that contains admissions suspected of sepsis [74]. Besides, 7 other real data sets are selected from the UCI Machine Learning Repository [75] to be evaluated with our proposed system to prove its effectiveness with a variety of data.

Table 4.1: Details of data sets collected from UCI and MIMIC III

No.	Name	Inst.	Attr.	Classes	Data types	Missing values
1	Adult	48842	14	2	Categorical, Integer	Yes
2	Banknote	1372	5	2	Real	No
3	Chess	3196	36	2	Categorical	No
4	German-credit	1000	20	2	Categorical, Integer	No
5	Haberman	306	3	2	Integer	No
6	Liver-disorder	345	7	2	Categorical, Real, Integer	No
7	Magic	19020	11	2	Real	No
8	Sepsis	11791	29	2	Categorical, Real	No

4.3.2 Experimental Setups and Final Results

To prove the merits of our proposed system GSIC, a binary classification experiment is conducted on a wide range of real data sets and popular baseline methods. Classifiers in the experiment are separated into two groups: black-box methods (Neural network, Random Forest, AdaBoost and RBF SVM) and interpretable methods (GA^2M, SBRL, CART, Logistic Regression). Generally, black-box methods are anticipated to be able to gain superior results than the interpretable classifiers due to their capability of capturing complex and diverse patterns inside the underlying data. Hyper-parameters of baseline classifiers are optimized using the GridSearch method. For GSIC, specific values for the set of parameters are determined as observed to be effective as well as improve the interpretability of our system. Classifiers are run with 5 folds cross-validation for each data set and final results are the average of results of 5 folds cross-validation run-times.

For evaluating the results of the classification experiment, we used the AUC (Area Under The Curve) metric to assess the performance of classifiers. As can be observed in Table 4.2, black-box methods perform better for most of the testing data sets regarding their AUC score. Particularly, the neural network proved its

Table 4.2: AUC of classification results of 8 datasets from UCI and MIMIC III

Method	Sepsis	Magic	Banknote	Adult	Chess	Haberman	Liver-disorder	German-credit
Neural Net	**0.764 ± 0.021**	0.848 ± 0.007	0.996 ± 0.005	0.771 ± 0.008	0.994 ± 0.004	0.579 ± 0.063	**0.678 ± 0.052**	**0.683 ± 0.032**
Random Forest	0.713 ± 0.014	**0.856 ± 0.008**	0.991 ± 0.007	0.778 ± 0.004	0.989 ± 0.004	0.574 ± 0.047	0.662 ± 0.056	0.662 ± 0.02
AdaBoost	0.711 ± 0.012	0.821 ± 0.003	0.996 ± 0.004	0.783 ± 0.004	0.967 ± 0.008	0.603 ± 0.059	0.633 ± 0.026	0.663 ± 0.034
RBF SVM	0.664 ± 0.011	0.838 ± 0.005	**1.0 ± 0.0**	0.762 ± 0.004	**0.995 ± 0.002**	0.529 ± 0.03	0.629 ± 0.033	0.677 ± 0.013
CART	0.592 ± 0.066	0.711 ± 0.024	0.929 ± 0.011	0.74 ± 0.011	0.891 ± 0.033	0.606 ± 0.051	0.635 ± 0.066	0.562 ± 0.023
Logistic Regression	0.67 ± 0.016	0.745 ± 0.005	0.99 ± 0.003	0.767 ± 0.004	0.975 ± 0.004	0.543 ± 0.046	0.599 ± 0.054	0.676 ± 0.033
GA^2M	0.712 ± 0.015	0.825 ± 0.006	0.998 ± 0.002	**0.794 ± 0.008**	0.93 ± 0.049	0.55 ± 0.052	0.601 ± 0.041	0.639 ± 0.019
SBRL	0.54 ± 0.026	0.711 ± 0.014	0.933 ± 0.008	0.713 ± 0.019	0.945 ± 0.009	0.516 ± 0.019	0.57 ± 0.035	0.547 ± 0.057
GSIC	0.745 ± 0.025	0.812 ± 0.008	0.991 ± 0.002	0.758 ± 0.009	0.939 ± 0.018	**0.623 ± 0.055**	0.653 ± 0.063	0.641 ± 0.023

efficiency through prominent results. It is also noticeable that for a few data sets, CART still can achieve comparable results to other complex models. However, the trees generated by CART in those cases are mostly complicated and have large sizes which lower the interpretability and are easy to overfit the data. On the other hand, the proposed classifier GSIC performed competitively with black-box models while still preserves an acceptable degree of intelligibility. To that end, values of major hyper-parameters of the GSIC which are the number of groups and the deep of sub-tree inside each group are kept in an adequate range. A detailed analysis of the interpretability of GSIC will be provided in the next subsection with the sepsis data use case.

4.3.3 Interpretability Analysis for ICU Sepsis Use Case

According to [76], sepsis is a main cause of hospitalization, morbidity and mortality worldwide and has been listed as a healthcare priority by WHO [77]. The definition for sepsis given by The European Society of Intensive Care Medicine/ Society of Critical Care Medicine is a "life-threatening organ dysfunction caused by a dysregulated host response to infection" [78]. There are several criteria for assessing the severity and detection of sepsis have been suggested. Recently, a comparative study has been conducted for evaluating the effectiveness of relevant sepsis measurements that focused specifically on the data of patients stayed in ICU (Intensive Care Unit) [74].

In our experiment, the data set of ICU sepsis patients that was used in the above comparative study will be employed in the task of predicting mortality risk with the proposed GSIC classifier. Particularly, the data set originally contains retrospective data of 23620 admissions of adult patients in ICU with de-identified demographics information and medical data archived during hospital stays. The medical information comprises of Elixhauser index (the degree of comorbid burden for a patient), SIRS index (Systemic Inflammatory Response Syndrome), SOFA index (Sequential Organ Failure Assessment) and data informing mechanical ventilation need. Generally, the cohort is divided into two groups: survival and non-survival. Details of ICU sepsis data set are described in Table 4.3.

Moreover, values of various sepsis criteria are also provided in the work of [74] for the ICU sepsis data set. Specifically, seven of the included sepsis criteria are SOFA index, suspected of infection, Sepsis-3, CDC, Angus, Martin, CMS and Explicit criteria. As can be seen from Table 4.4, the number of patients who satisfied the condition of SOFA index larger than 2 surges up to over 75%. While the Sepsis-3 index constitutes nearly half of the cohort. The Explicit criterion accounts for only 9% of ICU patients. However, the number of in-hospital mortality for positive cases that satisfies the Explicit criterion is the highest while for Sepsis-3, it is only

Table 4.3: Summarized information of ICU sepsis data set [74].

Variables	All patients (N = 11,791)	Survivors (N = 10,514)	Non-Survivors (N = 1,277)
Age (y) [Q1-Q3]	64.5 [51.1, 78.5]	63.3 [50.0, 77.5]	74.9 [61.8, 83.7]
Male, n (%)	6478 (54.9%)	5,795 (55.1%)	683 (53.5%)
BMI (kg/m2), mean ± SD	28.7 ± 8.4	28.7 ± 8.2	28.1 ± 10.1
Race, n (%)			
White	8497 (72.1%)	7,630 (72.6%)	867 (67.9%)
Black	1110 (9.4%)	1,036 (9.9%)	74 (5.8%)
Hispanic	457 (3.9%)	424 (4.0%)	33 (2.6%)
Elixhauser index	1 [-1, 6]	0 [-1, 6]	5 [0, 10]
SIRS	3 [2, 3]	3 [2, 3]	3 [3, 4]
SOFA	3 [2, 5]	3 [1, 5]	6 [4, 10]
Mechanical ventilation, n (%)	4149 (35.2%)	3,273 (31.1%)	876 (68.6%)
ICU length-of-stay (d)	1.9 [1.1, 3.5]	1.9 [1.1, 3.2]	2.4 [1.1, 5.6]
30 day mortality, n (%)	1619 (13.7%)	375 (3.6%)	1,244 (97.4%)
Hospital mortality, n (%)	1277 (10.8%)	0 (0)	1277 (100%)

14.5%. The same situation can be observed from the table for in-hospital mortality of negative cases and composite outcomes for positive/negative cases.

Table 4.4: Statistics for sepsis criteria with patients and mortality rate [74].

Criteria	Patients who satisfy criteria (N, %)	In-hospital mortality for positive cases	In-hospital mortality for negative cases	Composite outcome for positive cases	Composite outcome for negative cases
SOFA≥2	8869, 75.2%	13.20%	3.60%	41.20%	19.20%
Suspected of infection	7061, 59.9%	12.50%	8.30%	46.30%	19.90%
Sepsis-3	5784, 49.1%	14.50%	7.30%	50.00%	21.90%
CDC	3761, 31.9%	18.60%	7.20%	61.10%	23.80%
Angus	3368, 28.6%	17.90%	8.00%	61.20%	25.50%
Martin	1734, 14.7%	22.70%	8.80%	60.10%	31.50%
CMS	1302, 11.0%	27.20%	8.80%	64.70%	32.10%
Explicit	1062, 9.0%	30.10%	8.90%	70.70%	32.20%

One common way to evaluate the interpretability of a system is based on the so-called *functionally-grounded method* proposed by [15]. Specifically, a system is considered as interpretable when it is proved to be well-improved than a proxy method. In our case, we would like to compare the results of our system the proxy method which is the CART algorithm as in Table 4.5. It is obvious that the decision rules generated by GSIC are much succinct and more understandable than the ones of CART. However, the rules generated by GSIC are conditionally applied to a

specific group of cohorts instead of the whole population.

Table 4.5: Comparison of decision rules generated by GSIC and CART

GSIC	CART
Rule 1: **if** lods > 5.5 **and** hosp_los ≤ 2.626 **and** lods > 7.5 **and** weight ≤ 97.7 **and** weight > 57.95 **then** *Class 1.* **Rule 2:** **if** lods > 5.5 **and** hosp_los ≤ 2.626 **and** lods ≤ 7.5 **and** vent > 0.5 **and** sirs > 2.5 **then** *Class 1.* **Rule 3:** **if** lods ≤ 5.5 **and** hosp_los ≤ 0.353 **and** age > 67.03 **and** weight ≤ 140.5 **then** *Class 1.*	**Rule 1:** **if** lods ≤ 0.425 **and** hosp_los > 0.002 **and** lods > 0.225 **and** hosp_los > 0.012 **and** icu_los > 0.037 **and** hosp_los > 0.033 **and** age > 0.693 **and** icu_los > 0.061 **and** hosp_los > 0.047 **and** elixhauser_hospital > 0.745 **and** sirs > 0.625 **and** race_black ≤ 0.5 **and** lods ≤ 0.325 **and** hosp_los > 0.125 **then** *Class 1.*

Furthermore, a model is also considered as interpretable if it comprises of understandable components (*model-based interpretability*) [4]. Particularly, GSIC practically implements transparent ML techniques including GSOM, k-means clustering and Decision trees. The descriptive accuracy as mentioned in [4] then can be improved with the employing of the aforementioned transparent methods. For each of the implemented techniques, a specific kind of insight about the underlying data can be extracted. For example, after the training of GSOM, a visual sentimental map of the original data can be obtained which reveals the relationships that are hidden inside the data. Specifically, as can be seen from Figure 4.2, a 2D mapping of the ICU sepsis data can be obtained from GSOM where mortality cases are marked with cross marks.

A closer look can be taken into the 2D map generated from GSOM for the ICU sepsis data set. From the map, we can observe the distribution of the original data (where similar cases are grouped next to each other). Particularly, more observation about critical cases (mortality cases are cross marked) can be made that provides practitioners with an overview of groups of the vulnerable cohort. Digging deeper into the new representation of the original sepsis data, various groups of cohorts (subcohorts) can be extracted by clustering the data points on GSOM map. Specifically, groups of mortality cases can be induced as in Figure 4.3 using k-means clustering

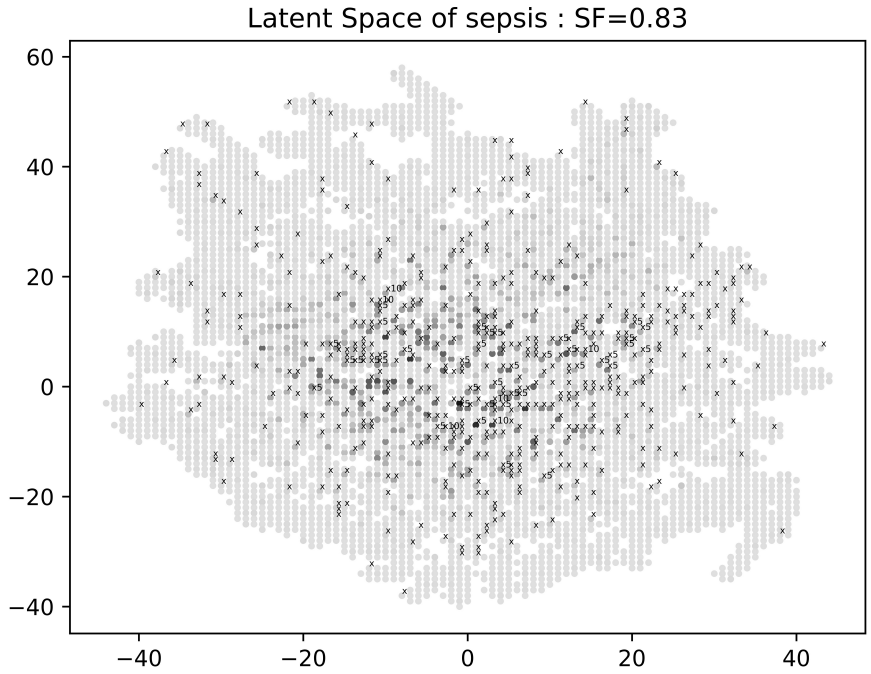


Figure 4.2: The 2D map generated by GSOM for the ICU sepsis data set.

Table 4.6: Inference rules of each cluster for predicting mortality cases

Cluster 1	Cluster 2	Cluster 3
if lods > 7.5 **and** hosp_los ≤ 3.877 **and** sepsis_cdc_simple > 0.5 **and** vent > 0.5 **then** *Class 1.* **if** lods > 7.5 **and** hosp_los > 3.877 **and** hosp_los ≤ 7.481 **and** icu_los > 4.303 **and** elixhauser_hospital > -2 **and** age > 59.269 **and** weight > 59.1 **then** *Class 1.* **if** lods ≤ 7.5 **and** hosp_los ≤ 1.619 **and** sepsis_cdc_simple > 0.5 **and** icu_los > 0.983 **then** *Class 1.*	**if** lods > 5.5 **and** hosp_los ≤ 2.626 **and** lods > 7.5 **and** weight ≤ 97.7 **and** weight > 57.95 **then** *Class 1.* **if** lods > 5.5 **and** hosp_los ≤ 2.626 **and** lods ≤ 7.5 **and** vent > 0.5 **and** sirs > 2.5 **then** *Class 1.* **if** lods ≤ 5.5 **and** hosp_los ≤ 0.353 **and** age > 67.03 **and** weight ≤ 140.5 **then** *Class 1.*	**if** lods > 7.5 **and** hosp_los ≤ 3.356 **and** hosp_los ≤ 1.685 **and** hosp_los > 0.320 **then** *Class 1.* **if** lods > 7.5 **and** hosp_los ≤ 3.356 **and** hosp_los > 1.685 **and** lods > 8.5 **and** hosp_los > 1.768 **then** *Class 1.* **if** lods > 7.5 **and** hosp_los > 3.356 **and** icu_los > 4.097 **and** hosp_los ≤ 6.392 **and** age ≤ 86.89 **and** elixhauser_hospital ≤ 19.5 **and** icu_los > 4.19 **then** *Class 1.*

method. For contrasting various groups of cohorts, we use the Decision trees method to extract representative rules between clusters. As can be seen in Figure 4.4,

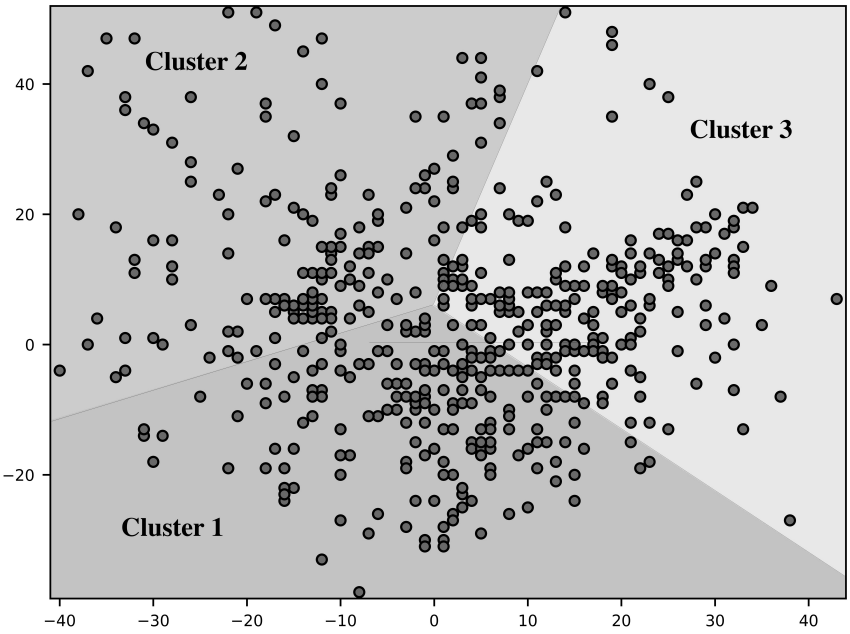


Figure 4.3: K-means clustering on generated GSOM .

sepsis_martin index can be considered as one of the main factors to distinguish group 3 of the cohort (cluster 3) from other groups.

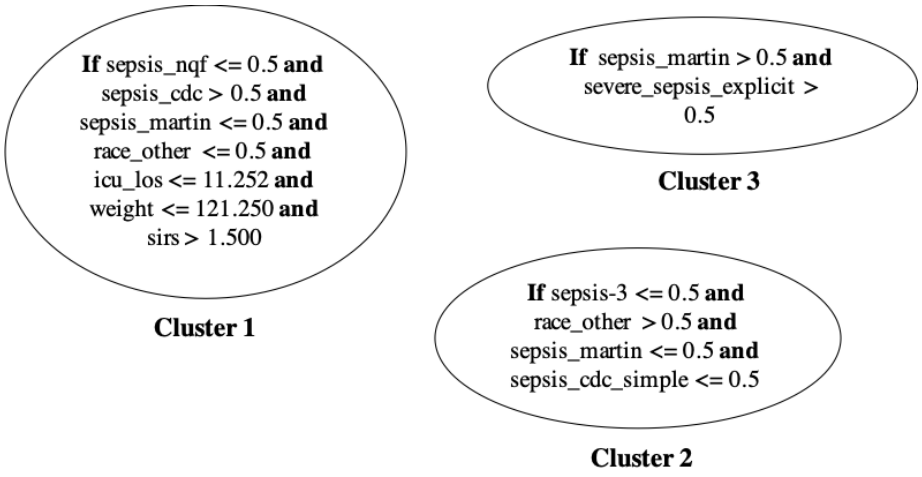


Figure 4.4: Representative rules for each sub-cohort.

After extracting groups of cohorts, in order to classify new instances, decision rules are generated for each cluster using the Decision trees algorithm that is trained on separate data instances belonging to each cluster. The generated rules can be sorted based on the number of mortality cases that satisfy those rules. The top-three rank rules that belong to each group can be seen in Table 4.6. Thanks to the simple form of decision rules that are generated from separated clusters, it is more under-

standable for practitioners to gain insights on significant factors that can impact the high risk of mortality due to sepsis. Specifically, as can be observed in Table 4.6, several factors can have major influence on the mortality risk with sepsis can be mentioned as sepsis-related indices (e.g. *lods* (logistic organ dysfunction system)), time of stay in hospital and ICU (*hosp_los*, *icu_los*) or patients' demographic (*age, weight*).

4.4 Summary

In this chapter, we introduced a new interpretable classification system named GSIC that achieved comparable results with common classification models. The proposed system has a transparent structure and its results can be interpreted and understandable by practitioners. Particularly, as demonstrated in the use case of predicting the high mortality risk of sepsis patients that stay in ICU, the information generated by GSIC can provide users with insights into the relationships hidden in the underlying data. Furthermore, those insights can be applied to the process of prioritizing patients for sepsis treatment. However, it needs to be further verified the effectiveness of the results generated by GSIC with experts in the related fields.

The proposed system still has several limitations such as long computation time for generating data mapping information or the lack of quantifying the uncertainty in the classifying process. For future work, we will pay more effort in improving on aforementioned limitations. Specifically, more focus will be put on enhancing the efficiency of the system by optimizing the learning process as well as employ a more fine-grained hyper-parameter tuning. Also, we will extend the evaluation experiments with more interpretable and black-box ML models to be able to provide a broader view of the performance of our proposed system when compared with a wide range of common techniques. Furthermore, the proposed system GSIC can be extended for handling multi-classes prediction tasks as well as working with unsupervised data.

Chapter 5

Enhancing Supervised Learning with Uncertainty Management

One of the main challenges for the supervised learning task is the uncertainty and ambiguity of the original data. There are many efforts that have been put into dealing with the challenge, however, there are still various remained problems, especially when the interpretability requirement is bundled together. Particularly, one of the popular supervised learning methods proposed for handling uncertainty data is the EKNN. The EKNN is an extended version of the famous k-NN (k-nearest neighbors) classifier that is developed based on the evidence theory. However, by remaining the original structure of the k-NN which is a distance-based technique, EKNN also exists the limitations when dealing with high dimensionality data as well as performs ineffectively with mixed distribution data where closed data points originated from different classes.

In this chapter, we introduce a new classification technique that can *"softly"* classify data points upon each separate cluster which can mitigate the overlapping data problem. The uncertainty introduced in the representation of data is managed by combining pieces of evidence induced from the trained clusters using Dempster's combination rule to generate final decisions. Moreover, the computational cost is improved by defining the mass function of evidence with the weight factor based on the distance between new data points and clusters' centers. In other to prove the merits of our proposed classification technique, we conduct an experiment on a wide range of real data and popular classifiers. The results have shown that our proposed technique can achieve comparable performance to state-of-the-art methods. The content of this chapter is a part of our work that has been published in [79].

5.1 Introduction of Uncertainty in Machine Learning

Along with the development of artificial intelligence applications, there is a newly rising field of managing uncertainty in machine learning and data mining tasks. Particularly, in several areas that require to make high-stakes decisions such as healthcare [80], bio-informatics [81] or even pure machine learning research [82, 83], it is essential to quantify and manage the uncertainty that arises from the data. According to [84], it is challenging when applying the traditional probabilistic framework on solving the problem of uncertainty, therefore several different approaches have been developed for managing uncertainty. Particularly, one of the most well-known theories in the field can be mentioned is the evidence theory proposed by [85].

Specifically, an example of the uncertainty in the field of supervised learning can be given is when several specific data instances cannot be correctly labeled due to the idea conflicts between experts. In order to resolve this specific type of uncertainty, several methods have been proposed that are developed based on the evidence theory. Specifically, a common approach is adopting partially supervised learning where labels of data are represented as the partial membership of available classes. Evidence-based induction trees [86] is a method that follows this approach by incorporating evidence-theoretic uncertainty measure to assess the impurity of nodes in decision trees. Another recent method is CD-EKNN [87] that can handle data that contain partially known labels by implementing the contextual discounting technique and utilizing the conditional evidential likelihood for optimizing the model's parameters.

Another common example of the uncertainty in the data is when the observed data is lack of information which can be caused by the sparse data problem or overlapping (mixed-distribution) data where data points belonging to different classes are close to each other. Several approaches have been proposed to mitigate this problem. One common method can be mentioned is EKNN [88]. EKNN is a distance-based method that classifies new data points based on the evidence of classes of their neighbors. EKNN works effectively with uncertainty data, however, it also has the limitation of computational complexity due to the induction of distances of data in the sample space. Several efforts have been made in order to mitigate this limitation by implementing feature selection or dimensional reduction techniques for downsizing the feature spaces such as ConvNet-BF [89] or REK-NN [90].

Another limitation of EKNN is the inflexibility when predetermining a number of nearby neighbors to extract evidence for the class information of a new data point. This problem can lead to errors when evidence from more neighbors is needed to collect (sparse distribution) or data points originating from different classes stay close to each other. Moreover, it is arguable when assigning hard-to-classified data

to only an ignorance group when those data points can be actually originated from different classes. Currently, several methods have been developed to improve these limitations. Specifically, a new approach is removing the need to defined in advance the number of neighbors by generating prototypes and extracting evidence of a new data point based on those prototypes such as ProDS [91], CCR [92]. Another approach instead of assigning hard-to-classified data to a single general unknown group, they will be assigned to the various classes-combined group named meta-class in addition to the original ignorance class such as BK-NN [93]. However, it is criticized that evidence extracted from the classes of prototypes is unreal due to their non-semantic representations. Also, information about the classes of new data points also has to be specified with a degree of certainty rather than merely assigned to some common groups of classes.

In this chapter, we put our effort to mitigate the aforementioned limitations. Specifically, we introduce a new classification method that can extract the evidence about the classes of new data objects from various *"fuzzy"* groups of data that represent different distributions that generate them. Generally, we make an assumption that a data set is generated by various distributions which can be represented in the form of heterogeneous clusters. In real-world applications, this kind of data can be observed from several fields such as healthcare or finance where data about users may be collected from different regions and periods. Basically, information about the distribution in each cluster can be extracted using decision trees on the set of data belonging to that cluster. Results from those decision trees could be considered as evidence for determining the classes of new data points. For making a final decision about the predicted class, Dempster's combination rule [85] will be used to fuse the evidence collected from previous steps.

The remaining part of this chapter is presented as follows. In section 5.2, backgrounds about evidence theory are introduced. In section 5.3, we elaborate on the details of our proposed classification system named *IEBC* (Inner Evidence-Based Classifier). In section 5.4, the experimental evaluation is conducted to prove the merits of our proposed system. Finally, in section 5.5, we provide a conclusion and mention about limitations of IEBC as well as our future work.

5.2 Evidence Theory

Evidence theory also known as Dempster-Shafer theory [85, 94] comprises a number of models for handling uncertainty. Dempster-Shafer theory generalizes probability theory and set-membership approaches. It plays an important role as an effective theoretical framework for reasoning with uncertain and imprecise information. In this part, we would like to provide a background of Dempster-Shafer theory in order to facilitate the formulation for our proposed classifier.

The frame of discernment defines a finite and unordered set Ω that contains relevant values of a variable x. Partial knowledge about the actual value of x can be represented by a *mass function* (also known as *the basic probability assignment*) m, which is defined as a mapping from 2^{Ω} to $[0, 1]$, satisfying that

$$m(\emptyset) = 0$$

$$\sum_{A \subseteq \Omega} m(A) = 1$$

For any subset $A \subseteq \Omega$, $m(A)$ can be interpreted as the belief that one is willing to commit to A. *Focal elements* of mass function m is defined as all A that satisfies $m(A) > 0$. There are two non-additive measures so-called *belief function* and *plausibility function* that can equivalently represent m as defined below.

For any subset $A \subseteq \Omega$, the belief function denoted by $Bel(A)$ is defined by

$$Bel(A) = \sum_{B \subseteq A} m(B)$$

and $Bel(A)$ can be interpreted as the degree of belief that the actual value of x is in A. The plausibility function $Pl(A)$ is then defined as

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B)$$

$Pl(A)$ can be seen as the degree to which the evidence is not contradictory to the proposition that the actual value of x is in A.

Furthermore, given two mass functions m_1 and m_2 on the same frame of discernment that are derived from independent pieces of evidence, they can be fused by using Dempster's combination rule [85] to obtain a new mass function $m_1 \oplus m_2$ which is defined as

$$(m_1 \oplus m_2)(A) = \frac{1}{1 - K} \sum_{B \cap C = A} m_1(B) m_2(C)$$

for all $A \subseteq \Omega$, where $K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C)$ is defined as the degree of conflict between m_1 and m_2. When the degree of conflict K between m_1 and m_2 is large, the combined evidence obtained by Dempster's combination rule may become unreliable and unintuitive which was pointed out in [95].

5.3 IEBC (Inner Evidence-Based Classifier)

A new classifier named *IEBC* will be introduced within this section in order to mitigate the problem of uncertainty with overlapping data that are generated by mixed distribution. The main intuition of our proposed system is that the overlapping data can be split by projecting them into a new representation and the classification task can be conducted on separated groups of data. To that end, we employ a fuzzy clustering technique on a new representation of the data and evidence about the class of new instances will be extracted and combined from those fuzzy clusters. Particularly, IEBC comprises of three major components: fuzzy clustering, decision trees induction and evidence combination. The general process of our proposed system is depicted in Figure 5.1.

In the first step of clustering new data representation, the fuzzy c-means clustering technique will be utilized for better grasping the uncertainty inside the data by allowing partial membership of data instances to each cluster instead of crisply assigning them to a specific one. The extracted fuzzy clusters are expected to contain instances that are generated by the same distribution and the characteristics of the distribution can be understood by analyzing instances belonging to that cluster.

In order to induce the characteristics of each cluster, decision rules generated by the CART algorithm will be applied. The CART algorithm will be trained on instances that belong to separate clusters and the results can also be used to classify new data instances. Those decision rules are considered as evidence about the "true" class of new data instances and will be combined across clusters using Dempster's combination rule to provide a final decision.

5.3.1 Notations

We would like to provide several notations to facilitate the formulation of our proposed classifier. Given a data set D that contains n instances and is characterized by m attributes. Then an instance x_i can be denoted as $x_i = \{x_{i1}, ..., x_{im}\}$ and data set $D = \{x_i | i = 1, 2, ..., n\}$. Values of label of an instance x_i belongs to the set $\Omega = \{1, ..., c\}$. We define a set of fuzzy clusters $\{A_j, j = 1, 2, ..., k\}$ that each data instance can belong to.

5.3.2 The Proposed Classification System

As mentioned in the previous part, IEBC comprises three main components which are fuzzy clustering, decision tree induction and evidence combination. In this sub-

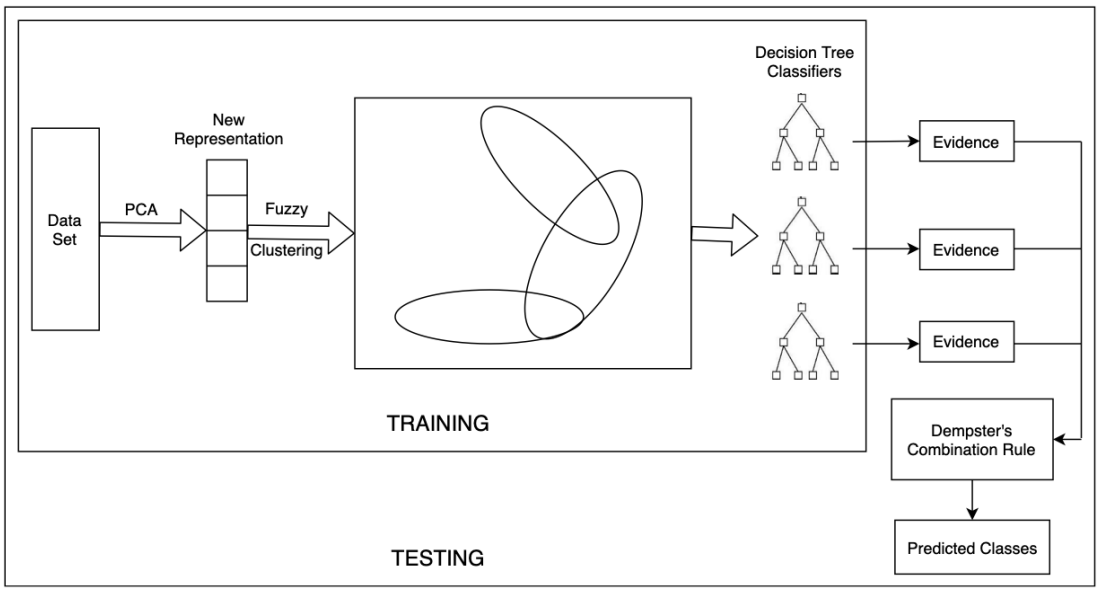


Figure 5.1: The overall process of the proposed classification system *IEBC*

section, we would like to provide a more detailed formulation of the proposed classification system. Particularly, in the first step of clustering new data representation, the uncertainty can also occur within the process [83]. Particularly, in several common clustering methods such as k-means, a data instance has to be assigned directly to a specific cluster even when it is possible to belong to other clusters as well. This problem reduces the effectiveness of the clustering process. In order to mitigate this limitation, we implement the fuzzy clustering approach where a data instance is allowed to have partial membership to many clusters as defined in the following [96].

$$\mu_{ij} = \mu_{A_j}(x_i) \in [0, 1] \tag{5.1}$$

where μ_{ij} is the membership value of a data instance x_i to cluster A_j and has to satisfy the following conditions:

- Sum of all membership values of a single data instance to all clusters is equal to 1.

$$\sum_{j=1}^{k} \mu_{ij} = 1, \text{ for all } i = 1, 2, ..., n$$

- There is no empty cluster or a cluster that contains all of the data instances in D.

$$0 < \sum_{i=1}^{n} \mu_{ij} < n$$

A family of fuzzy partition matrices, M_f, that contains membership values between k clusters and n data instances can be denoted as:

$$M_f = \left\{ U | \mu_{ij} \in [0,1]; \sum_{j=1}^{k} \mu_{ij} = 1; 0 < \sum_{i=1}^{n} \mu_{ij} < n \right\} \tag{5.2}$$

Any $U \in M_f$ is a fuzzy k-partition, and it follows from the overlapping character of the clusters and the infinite number of membership values possible for describing cluster membership that the cardinality of M_f is also infinity, that is, $\eta_{M_f} = \infty$.

To determine the fuzzy k-partition matrix U for grouping a collection of n data sets into k clusters, an objective function J for a fuzzy k-partition can be defined as:

$$J(U, v) = \sum_{i=1}^{n} \sum_{j=1}^{k} (\mu_{ij})^{\alpha} d_{ij}^2 \tag{5.3}$$

where α is the weighting parameter and d_{ij} is the Euclidean distance between the A_j cluster's center and the x_i data instance.

$$d_{ij} = d(x_i - v_j) = \left[\sum_{l=1}^{m} (x_{il} - v_{jl})^2 \right]^{1/2} \tag{5.4}$$

Particularly, the weighting parameter α adjusts the amount of fuzziness and has the value range of $[1, \infty)$ [97]. The center of cluster A_j is denoted as $v_j = \{v_{j1}, v_{j2}, ..., v_{jm}\}$. The value of v_j is computed as the following formula.

$$v_j = \frac{\sum_{i=1}^{n} (\mu_{ij})^{\alpha} \times x_i}{\sum_{i=1}^{n} (\mu_{ij})^{\alpha}} \tag{5.5}$$

In this work, the fuzzy c-means [98] will be implemented for the task of fuzzy clustering on the new representation of the original data. Details of the fuzzy c-means algorithm can be found in [99]. The generated fuzzy clusters are informative and flexible when assigning data instances to different groups. However, in order to extract insights from those clusters, it needs to harden (so-called *defuzzification*) the information contained in the fuzzy clusters.

There are mainly two methods to defuzzify the fuzzy clusters which are the *maximum membership* and *nearest center classifier* [99]. We use the maximum membership method for fulfilling the task as it is robust regarding its simplicity.

The main idea is straight-forward as assigning an instance to the cluster that has the highest membership values. However, by considering only the highest values, several fuzzy membership information can also be lost. For including that information, an adjustment is added to the maximum membership method so that not only the largest membership but also its nearly equal values will be assigned to the same cluster by defining a so-called fuzzy threshold β as the following.

$$\mu_{ij} \mapsto \overline{\mu_{ij}} = \begin{cases} 1 & \text{, if } \mu_{ij} = \max\limits_{j'}\{\mu_{ij'}\} \\ & \text{ or } |\mu_{ij} - \max\limits_{j'}\{\mu_{ij'}\}| \leq \beta \\ 0 & \text{, otherwise} \end{cases} \tag{5.6}$$

for $j, j' = 1, ..., k$ and $i = 1, 2, ..., n$.

By defuzzying partition matrix U into a hard partition matrix U' using Eq. (5.6), we can obtain k clusters $\{C_1, ..., C_k\}$ where

$$C_j = \{x_i | \overline{\mu_{ij}} = 1, i \in \{1, ..., n\}\} \text{ with } j \in \{1, ..., k\}$$

With each cluster C_j, a decision tree classifier dt_j is generated using *CART* algorithm [100] for extracting insights from the cluster as well as classifying new data instances. Specifically, for a data instance x_i, the results of each classifier is noted as $P_i^j = [p_{i1}^j, ..., p_{ic}^j]$ where p_{il}^j ($l \in \Omega$ and $i \in \{1, ..., n\}$) is the probability of the data instance x_i classified as class l by the classifier dt_j.

In the final step of IEBC, the results of each classifier are considered as evidence given by an independent expert on the "true" class of a data instances x_i. Based on this assumption, the probability matrix P_i^j generated by a classifier dt_j along with the weighting factor $u_{ij} \in U$ can be utilized to define the basic probability assignment (*bpa*) m_j as the following.

$$m_j : 2^\Omega \to [0, 1]$$

$$m_j(\{l\}) = p_{il}^j \times u_{ij} \tag{5.7}$$

$$m_j(A) = 0 \text{ if } |A| \geq 2,\ A \neq \Omega \tag{5.8}$$

$$m_j(\Omega) = 1 - \sum_{l=1}^{c} p_{il}^j \times u_{ij} \tag{5.9}$$

In IEBC, only the mass functions of focal elements which are singletons will be considered. All other imprecision will be assigned to the whole frame of discernment $\Omega = \{1, ..., c\}$. Specifically, given the evidence $E_j = \{m_j(\{l\}), m_j(\{\Omega\})\}$ with $j \in \{1, ..., k\}$ and $l \in \{1, ..., c\}$ that is provided by the classifier dt_j, we can combine the set of evidence $E = \{E_1, ..., E_j\}$ by using Dempster's combination rule (denoted as $\bigoplus$).

$$m = \bigoplus_{j=1}^{k} m_j$$

Specifically, two mass functions m_j and m_{j+1} derived from two evidence sources E_j, E_{j+1} can be combined using Dempster's rule to obtain a new mass function $m_j \oplus m_{j+1}$, defined as

$$m_j(A) \oplus m_{j+1}(A) = \frac{1}{1-K} \sum_{B \cap C = A} m_j(B) \times m_{j+1}(C)$$

for all nonempty $A \subseteq \Omega$, where $K = \sum_{B \cap C = \emptyset} m_j(B) \times m_{j+1}(C)$ is the degree of conflict between m_j and m_{j+1}.

The following table to describe the detailed combination of two evidence sources E_j, E_{j+1}.

Table 5.1: Intersection of pieces of evidence from two evidence sources

E_{j+1} \ E_j	$m_j(\{1\})$	$m_j(\{2\})$	...	$m_j(\{c\})$	$m_j(\Omega)$
$m_{j+1}(\{1\})$	$m(\{1\})$	$\emptyset$	...	$\emptyset$	$m(\{1\})$
$m_{j+1}(\{2\})$	$\emptyset$	$m(\{2\})$	...	$\emptyset$	$m(\{2\})$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
$m_{j+1}(\{c\})$	$\emptyset$	$\emptyset$	...	$m(\{c\})$	$m(\{c\})$
$m_{j+1}(\Omega)$	$m(\{1\})$	$m(\{2\})$	...	$m(\{c\})$	$m(\Omega)$

Finally, the pignistic probability transformation is usually utilized to transform the combined mass function m into a probability function for making final decisions

[101]. Due to the consideration of focal sets including only singletons and the ignorant set Ω, the probability function m' generated using the pignistic probability transformation is as follows.

$$m'(\{l\}) = m(\{l\}) + \frac{m(\Omega)}{|\Omega|} \tag{5.10}$$

From Equation (5.10), the portion of $\frac{m(\Omega)}{|\Omega|}$ is a constant value with $\forall l \in \Omega$. Consequently, the combined mass $m(\{l\})$ can be used directly to determine the class of an instance x_i by selecting *bpa* of the class that has the highest mass value. The pseudo-code of IEBC is provided in Algorithm 4.

$$predicted\_class = c^\star \tag{5.11}$$

$$c^\star = \underset{l}{argmax}(m(\{l\})) \tag{5.12}$$

where $c^\star, l \in \{1, ..., c\}$.

5.4 Experimental Evaluation

5.4.1 Testing Data Sets

For evaluating the proposed classification system IEBC, a subset of the MIMIC III data set that contains admissions suspected of sepsis in ICU are selected [74]. The MIMIC III (Medical Information Mart for Intensive Care III) is currently one of the most popular and biggest healthcare data sources developed by MIT Lab [73]. Particularly, nearly 60000 ICU admissions with their de-identified medical data are recorded. Besides, 9 data sets from UCI [75] are also selected to evaluate the generalizability of our proposed system for a variety of data. Summarized information of the selected data sets is included in Table 5.2.

5.4.2 Experimental Setups and Final Results

To prove the merits of our proposed system IEBC, a classification experiment will be conducted on binary-labeled data sets. Particularly, the proposed system will be contrasted with common classification methods including robust supervised learners (Neural network, Random Forest, AdaBoost and RBF SVM), evidence-based classification techniques (EKNN [88], ProDS [91]) and CART. Generally, robust classifiers such as the neural networks or random forests typically can achieve prominent performance (accuracy) compared with other techniques thanks to their proven high level

Algorithm 4. IEBC algorithm

Input: Data set $D = \{D_{train}, D_{test}\}$

Output: Predicted classes of D_{test}

Training:

Step 1: Transforming data

1. $D'_{train} = PCA(D_{train})$

Step 2: Clustering

2. Set k, α values
   Initialize partition matrix U^0
   Set step counter $r = 1$
3: **While** $||U^{r+1} - U^r|| > \epsilon$:
4: Calculate cluster centers v_j using Eq.(5.5) on D'_{train}
5: Updating partition matrix U^r:
6:

$$\begin{cases} \mu_{ij}^{(r+1)} = \left[\sum_{j'=1}^{k} \left(\frac{d_{ij}^r}{d_{ij'}^r}\right)^{2/(\alpha-1)}\right]^{-1} \\ \qquad\qquad \text{, if } I = \emptyset \\ \mu_{ij}^{(r+1)} = 0 \\ \qquad\qquad \text{, if } j \in \neg I \end{cases}$$

where

$I = \{j | 2 \leq k \leq n;\ d_{ij}^r = 0\}$

$\neg I = \{1, 2, ..., k\} - I$

7: Defuzzy clusters using Eq.(5.6)
8: **Return:** $U = \{u_1, ..., u_k\}$, $C = \{C_1, ..., C_k\}$

Step 3: Build classifiers

9: **For each** C_j **in** C:
10: $dt_j = Decisiontree(C_j)$
11: **Return:** $DT = \{dt_1, ..., dt_k\}$

Testing:

Step 4: Cluster referencing

12: **For each** x **in** D_{test}:
13: **For each** u **in** U:
14: Compute distances $d = Euclid(x, u)$
15: Update U as in Step 2 without recalculating v_j
16: Assign $W_x = U_x$
17: **Return:** $W_x = [w_1, ..., w_k]$

Step 5: Assigning mass of evidence

18: **For each** x **in** D_{test}:
19: **For each** dt_j **in** DT:
20: $l = dt_j(x)$
21: Assign mass $m(\{l\}) = 1$
22: Apply weighting factor $m_w(\{l\}) = w_j \times m(\{l\})$
23: **Return:** $M_w^x = [m_{w\_1}, ..., m_{w\_k}]$

Step 6: Combining evidence

24: **For each** x **in** D_{test}:
25: $m_{w\_1}(\Omega) = 1 - \sum_{l=1}^{c} m_{w\_1}(\{l\})$
26: Initialize combined mass $m_{comb} = m_{w\_1}$
27: **For** i **in range** $(2, k)$:
28: $m_{w\_i}(\Omega) = 1 - \sum_{l=1}^{c} m_{w\_i}(\{l\})$
29: $m_{comb} = Dempster(m_{comb}, m_{w\_i})$
30: $y_x = \underset{l}{argmax}(m_{comb}(\{l\}))$
31: **Return:** Predicted classes $Y = [y_1, ..., y_{|D_{test}|}]$

of competence in learning complex and diverse patterns from the training data. For a fair comparison, we optimize parameters of baseline methods applying the Grid-Search method. However, for our proposed method, in order to preserve a proper level of interpretability, we restraint the ranges of main parameters to low values. The classification experiment is run with 5 folds cross-validation for each model and data set. The final results are the averaged value of 5 folds cross-validation run-times. A part of the results is reused from our previous work [70].

The main metric used to evaluate the classification results is AUC (Area Under The Curve) score. It is an effective and simple metric to evaluate the binary

Table 5.2: Characteristics of 10 datasets collected from UCI and MIMIC III

No.	Name	Inst.	Attr.	Classes	Data types	Missing values
1	Chess	3196	36	2	Categorical	No
2	Credit-approval	690	15	2	Categorical, Integer, Real	Yes
3	Diabetes	769	9	2	Real	No
4	Haberman	306	3	2	Integer	No
5	Heart	303	75	2	Categorical, Integer, Real	Yes
6	Heart-statlog	270	13	2	Categorical, Real	No
7	Inosphere	351	34	2	Integer, Real	No
8	Liver-disorder	345	7	2	Categorical, Real, Integer	No
9	Magic	19020	11	2	Real	No
10	Sepsis	11791	29	2	Categorical, Real	No

Table 5.3: AUC of classification results of 10 testing datasets - part 1

Method	Sepsis	Haberman	Magic	Chess	Diabetes
Neural Net	**0.764 ± 0.021**	0.579 ± 0.063	0.848 ± 0.007	0.994 ± 0.004	0.703 ± 0.024
Random Forest	0.713 ± 0.014	0.574 ± 0.047	**0.856 ± 0.008**	0.989 ± 0.004	0.733 ± 0.031
AdaBoost	0.711 ± 0.012	0.603 ± 0.059	0.821 ± 0.003	0.967 ± 0.008	0.708 ± 0.031
RBF SVM	0.664 ± 0.011	0.529 ± 0.03	0.838 ± 0.005	**0.995 ± 0.002**	0.722 ± 0.046
CART	0.592 ± 0.066	0.606 ± 0.051	0.711 ± 0.024	0.891 ± 0.033	0.698 ± 0.039
ProDS	0.526 ± 0.013	0.571 ± 0.044	0.789 ± 0.007	0.940 ± 0.009	0.723 ± 0.050
EKNN	0.567 ± 0.013	0.546 ± 0.053	0.821 ± 0.004	0.957 ± 0.005	0.714 ± 0.016
IEBC	0.761 ± 0.013	**0.665 ± 0.039**	0.833 ± 0.007	0.978 ± 0.016	**0.748 ± 0.026**

Table 5.4: AUC of classification results of 10 testing datasets - part 2

Method	Liver-disorder	Credit-approval	Heart-statlog	Ionosphere	Heart
Neural Net	0.678 ± 0.052	0.86 ± 0.024	**0.835 ± 0.031**	0.913 ± 0.024	0.83 ± 0.049
Random Forest	0.662 ± 0.056	**0.874 ± 0.019**	0.823 ± 0.065	**0.93 ± 0.037**	0.808 ± 0.032
AdaBoost	0.633 ± 0.026	0.865 ± 0.008	0.782 ± 0.049	0.905 ± 0.042	0.817 ± 0.046
RBF SVM	0.629 ± 0.033	0.863 ± 0.025	0.827 ± 0.042	0.929 ± 0.049	0.812 ± 0.083
CART	0.635 ± 0.066	0.749 ± 0.037	0.776 ± 0.057	0.861 ± 0.046	0.741 ± 0.05
ProDS	0.669 ± 0.023	0.867 ± 0.019	0.823 ± 0.041	0.937 ± 0.015	**0.845 ± 0.022**
EKNN	0.637 ± 0.053	0.864 ± 0.037	0.813 ± 0.031	0.920 ± 0.042	0.796 ± 0.051
IEBC	**0.689 ± 0.031**	0.842 ± 0.098	0.822 ± 0.042	0.909 ± 0.047	0.83 ± 0.043

classification task. As can be observed from the final results in Table 5.3 - 5.4, robust classifiers such as the neural network or random forests almost dominate other methods regarding AUC metric. Moreover, our proposed classifier IEBC is out-

performed state-of-the-art methods while achieved the highest AUC score for three data sets. With other testing data sets, its performance is competitive with other robust learning techniques. It should be mentioned that IEBC also has a good degree of interpretability and their hyper-parameters are not optimized yet. Particularly, IEBC well-performed with medical data sets such as Sepsis, Haberman, Diabetes, Liver-disorder and Heart which proved the effectiveness of our proposed method over the overlapping data problem (medical data generated by mixed distribution).

5.5 Summary

In this chapter, we introduced a new evidence-based classification system named IEBC for handling the uncertainty that is generated from the data. Specifically, IEBC is developed to handle the problem when classifying data that are generated by various distributions. To that end, a fuzzy clustering technique is implemented in order to extract information of heterogeneous clusters inside the underlying data. Then, the characteristics of each cluster will be captured using the decision trees technique. Finally, evidence about the class of new data points will be collected and integrated by utilizing Dempster's combination rule to generate the final result.

The effectiveness of our proposed method has been proved by a classification experiment conducted on a wide range of real data (especially with healthcare data set) and popular baseline methods. Furthermore, as IEBC comprises of interpretable machine learning methods such as fuzzy clustering and decision trees as well as an intuitive evidence combination method, it can also provide users with insights about relationships that are hidden in underlying data. The extracted knowledge can support the decision-making process in fields which require to make high-stakes decisions such as healthcare, medicine or finance. For future work, we will conduct the multi-classification task with more evidence-based methods. Moreover, a deeper investigation of applications of our method in healthcare or business fields will be conducted to explores its potential.

Chapter 6

General Discussion

In this chapter, we would like to discuss problems and challenges in the field of interpretable ML and XAI. Firstly, we bring up the original problem of defining interpretability, explainability and how to be able to claim an ML model interpretable. As currently, there is no concrete definition for those concepts that leads to the hardness of evaluating interpretable ML models. Secondly, the future and potential of transparent ML models will be discussed. Specifically, we attempt to find answers to the question posed that how and which directions interpretable models could be developed among the widespread of black-box models such as deep neural networks. Finally, we would like to have a brief discussion on trends and applications of post-hoc models which are also a promising and powerful solution in the field besides developing intrinsically interpretable models.

What is interpretability in ML context and how to claim a ML model to be interpretable?

In many research, interpretability is considered as a broad and poorly defined concept with little consensus on its definition and evaluation [4, 15]. Therefore, instead of implementing a general meaning of this concept, researchers usually try to place it into a concerned context. Specifically, in [4], they considered interpretability in the context of machine learning as a part of the data science life circle. Interpretable machine learning is defined as the use of ML models to extract relevant knowledge about domain relationships contained in data. Where relevant knowledge is defined as available insights for a specific audience within a chosen domain problem. On the

other hand, in the context of ML systems, [15] defined interpretability as the ability to explain or to present in understandable terms to human.

Moreover, in many of XAI-related research, interpretability and explainability are two concepts which are usually used interchangeably. However, according to [102], interpretability is a concept that falls under the umbrella of explainability. If a system is explainable then it is interpretable but maybe not correct in a reverse way. The goal of interpretability is merely to describe the internal of a system in a way that is understandable to humans. Therefore, it depends on cognition, knowledge and bias of users to determine whether a system is interpretable. It also mentioned that it should be cautious not to develop persuasive systems rather than transparent systems.

More generally, in [103], interpretability was observed as not a monolithic concept but reflects several distinct ideas. They argued that interpretations serve objectives that are deemed important (such as ethics, legality) but struggle to model formally. Similarly, in [15], interpretability serves as a tool to confirm other important desiderata of ML systems besides task performance. Those objectives (desiderata) may include trust, causality, transferability, informativeness, fair and ethical decision-making.

Due to the lack of consensus on the definition of interpretability concepts, there is also no only benchmark for evaluating interpretable ML models. In chapter 2, we introduced details of several notable evaluation methods for interpretable models. In this discussion part, we would like to summarize those approaches and mention their limitations. Currently, there are two common ways to evaluate the interpretability of ML models [15]. The first way is placing the models in its application, if the system is useful in a practical application or a simplified version of it, then it will be considered interpretable. The second way to evaluate the interpretability is via a quantifiable proxy by firstly claiming the interpretability of a specific class of ML methods such as linear models, rule lists, then follows the proposed models optimized on those classes.

Obviously, those aforementioned evaluation methods are vague, simple and subjective. New evaluation approaches as mentioned in chapter 2 are developed with more firm background therefore more reliable. Specifically, in [15], they proposed a data-driven way to derive operational definitions and evaluations for explanations. The proposed evaluation method creates a general framework for evaluating the interpretability of ML systems. However, there is a lack of specific instructions for various kinds of applications. Also, human-related evaluations have to be designed carefully to avoid confounding factors and user-related bias. In [4], they clarified the desiderata for interpretations when placing the concepts in the general data-science life cycle which including predictive accuracy, descriptive accuracy and relevancy. However, also no detailed instructions are given to improve the quality of concerning

objectives.

In summary, to the best of my knowledge, there is a need for more detailed definitions of interpretability in the machine learning context. Also, the evaluation methods for interpretable ML models have to be defined more clearly in both vertical and horizontal directions. In the vertical direction, there is the need to define the abstract of evaluation methods for interpretable ML while in the horizontal direction, application-specific evaluations have to be described.

Future and potential of transparent models

Currently, there are two major trends towards implementing interpretable ML as introduced in Chapter 2 which are intrinsically transparent and post-hoc models. While post-hoc interpretation models seem promising while providing explanations for black-box models as the results benefit from both the high performance of black-box models as well as the capability to provide explanations for their decision-making process. However, in the work of [2], they argued about the limitations of this approach compared to building transparent models. In this part, we would like to bring up their arguments as the background for the debate about the potential of transparent models and their future. Specifically, post-hoc interpretation models suffer from several limitations including the fidelity of the provided explanations. It comes from the fact that many explanation models do not mimic the computations of the original models, therefore those explanation models may not use the same features as original ones and thus not faithful to the working mechanism of the black-box models.

The second problem related to post-hoc models is that the provided explanations are usually incomplete. In [2], they gave an example of saliency maps used in the image processing field and argued that the relationship of final labels and the focus parts of the network is not clarified which leads to confusion when encountering misclassified cases. Moreover, the incorporation of outside information is hard compared with transparent models. From those limitations, the development of explanation techniques for black-box models could be troublesome and potentially cause catastrophic harm to society.

However, the development of intrinsically transparent models also encounters several major challenges. When black-box models such as neural networks or random forest can automatically learn non-linear relationships, most well-known transparent models such as decision trees or association rule lists struggle to capture that information. Moreover, the bigger the size of the learning problem, the more complex the rules generated by transparent models. Those characteristics limit transparent models from the applications with big data where a large volume of data and features

are entailed. Most of the current approaches for building transparent models involve the incorporation of expert knowledge into intrinsic interpretable models that make the process ineffective and costly in both computational and financial perspectives. Also, due to no concrete definition of interpretability, it is challenging to develop general interpretable models. Instead, application or domain-specific interpretation is usually developed.

Currently, due to the limitation in performance generally, interpretable ML models are usually developed for specific data. In particular, instead of being optimized for all available data which generated by various distributions and posses different characteristics, it is reasonable to focus on specific datasets which are concerned by a specific application such as patients' medical record in healthcare application or material structure-properties in material discovery application. By focusing on the data structure to build interpretable models, the proposed models can be more effective and useful. The second trend in developing transparent models is efforts to reduce the complexity of generated rules, explanations or representations. For example, in the case of decision trees, several techniques could be applied to reduce the size of the generated trees such as trimming techniques which can generalize the results as well as reduce the complexity. Finally, instead of generating global explanations, transparent models could focus on generating local interpretation with the aim of reducing the overwhelming of problem space while still can provide accurate explanations for a specific decision. Several techniques can be applied to fulfill the goal can be listed as case-based learning or prototype learning.

Future trends for black-box models with explanations

Today, black-box models are becoming ubiquitous with a wide range of applications because of their prominent performance. Black-box models such as deep neural networks or random forests generally perform better than other intrinsically transparent methods in many applications due to their capability of capturing non-linear relationships in the data. In several fields such as image processing or text processing, deep neural networks can equal or even outperform human which lead to the wide-spreading of those applications from the business section to private life.

A common application of black-box models is recommendation systems for products in business sites where new products can be introduced to users based on their historical checking items or similarity between users. Another popular application can be listed as machine translation services such as Google translate which apply text processing techniques, or face and object recognition applications used for photo-video capturing or security control purposes. Also, a notable application of deep learning is in automatic driving cars where ML models collect and process

multi-sources data for supporting safe driving.

For some of the aforementioned applications such as recommendation system or automatic language translation, the information and results provided by those applications for users is not significant for high-stakes decisions and allow room for mistakes without a deep investigation into the reason. However, in other applications such as security control using face recognition or in automatic driving cars, errors happen in those kinds of applications could lead to life-threatening consequences. When problems happen to those applications, careful diagnostic needs to be carried out for debugging the origin of the problems as well as providing customers with reasonable explanations for those failures. However, if the applications are integrated with black-box models which are opaque and hard to explain then it is a non-trivial task to debug the problem as well as provide clues for their decision-making process.

Even with a full acknowledgment of the above-mentioned limitations of black-box models when being applied to high-stakes decision-making applications, one still cannot totally negate the role and potential of black-box models in those applications [2]. Therefore, many methods have been proposed to provide explanations for black-box models' decisions or integrate the explainability to those models. Specifically, as described in the literature review part of Chapter 2, an example could be listed as the saliency maps for the recurrent neural network using in the image classification task. In that case, a separate model is used to estimate the focus pixels in the images which the network focuses on. However, as pointed out by [2] where it does not provide intuitive information for users about the relations of the focus pixels with the outcome label of the image, especially in wrongly detected cases. Another notable effort in embedding explainability to black-box models is the attention model with LSTM (Long Short-Term Memory) - a variation of the recurrent neural network. In this technique, an attention mechanism is combined with the original model to extract textual based explanations for the process images. Also a notable approach is providing local explanations for a single result of black-box models such as LIME (Local Interpretable Model-Agnostic Explanations) which is described in detail in Chapter 2.

In summary, in spite of the substantial limitations of black-box models for high-stakes applications, black-box models are still attractive to this field due to their prominent performance and well-developed tools as well as the abundant environment. However, one should consider carefully about limitations of those models when intend to develop for critical applications in fields such as healthcare, medicine or self-driving car. Especially, for the generation of explanations using a second model, the fidelity of explanations to the actual processing mechanism of the original model has to be maintained in order to provide trustworthy explanations. It is recommended to embed directly the explainability mechanism into the original black-box models to better ensure the fidelity, however, it can cause a significant increase in computation cost as well.

Chapter 7

Conclusion

Machine learning and data mining techniques increasingly play a significant role in society under various forms of hardware and software applications. Those applications can exist in phones, smartwatches, cars or even at your home and have profound impacts on our private life as well as our work. One of the main reasons for their development is the improvement in the performance with complex models built on the base of a huge amount of collected data. Those models are designed to learn non-linear relationships between the underlying data and provide the predictions based on those learn information. However, little knowledge about what they have learned can be revealed for their users - which is us - due to the complexity of their working mechanism. This situation leads to several limitations including the hardness when debugging a certain problem produced by the systems or gaining the trust of users by proving their fidelity and fairness. Especially, in many domains that require transparency and interpretability such as medicine or healthcare, the characteristics of those models become an important limitation when considering their adoption in those fields. In our research, we make an effort to mitigate the aforementioned problems for promoting the applications of ML techniques in critical decision-making fields.

Specifically, in the field of unsupervised learning, clustering is a fundamental task that has been utilized in many scientific fields. Clustering groups data into clusters. For each cluster, objects in the same cluster are similar between themselves and dissimilar to objects in other clusters. The k-means is a popular interpretable method for the clustering task. However, it suffers from the problem of underfitting data with simple dissimilarity measures - a key part in formulating clusters of k-means. In our work, we proposed a new k-means-based clustering method with a novel dissimilarity measure that can better fit with the underlying data. The effectiveness of the proposed clustering algorithm is proven by a comparative study conducted on popular clustering methods for categorical data.

In the field of supervised learning, we proposed a two-stage binary classification system named GSIC that is applicable for healthcare (or general) data. The proposed system still preserves a proper level of interpretability and can also achieve comparative results with popular classification techniques. The motivation behind the proposed system is the lack of effective classification methods for handling data generated by various distributions (such as healthcare or finance data) that can harmonize both performance and interpretability perspectives. The experimental evaluation with a use case in sepsis patients staying in ICU has shown the merits of our proposed classification system.

On the other hand, we realized the limitation of our proposed classification system when dealing with uncertainty. Specifically, real data with high uncertainty and ambiguity is challenging for the classification task. E-KNN is a popular evidence theory-based classification method developed for handling uncertainty data. However, as a distance-based technique, it also suffers from the problem of high dimensionality as well as mixed distribution data where closed data points originated from different classes. Based on that motivation, we enhanced our proposed classification system GSIC with the capability of handling the uncertainty existing in the underlying data. The classification experiment conducted on various real data and popular classifiers has shown that the proposed technique has competitive results compared with state-of-the-art methods.

For our future work, limitations on our proposed models will be carefully investigated and improved. Particularly, more efforts will be put into the new k-means-based clustering technique for reducing computational cost as well as provide visualization tools to represent discovered mutual relationships between features. For the improved version of our proposed classification method GSIC, we will conduct further testing based on extracted insights from the underlying data with the collaboration of experts in healthcare or material science fields. Furthermore, we would like to investigate post-hoc models as a potential research direction for explaining the decisions of black-box models with common types of data such as text or images. We hope that with the knowledge gained from developing intrinsically transparent methods, we can make contributions towards the improvement of the explainability and interpretability of post-hoc models.

Publications

1. T.-P. Nguyen, D.-T. Dinh, and V.-N. Huynh, "A New Context-Based Clustering Framework for Categorical Data" in *PRICAI 2018: Trends in Artificial Intelligence*, 2018, pp. 697–709.

2. T.-P. Nguyen and V.-N. Huynh, "A New Interpretable System for Knowledge Exploration and Classification: ICU Sepsis Data Case Study" in *AHFE 2020: The Human Side of Service Engineering*, July 2020.

3. T.-P. Nguyen, S. Nguyen, D. Alahakoon and V.-N. Huynh, "GSIC: A New Interpretable System for Knowledge Exploration and Classification" in *IEEE Access*, vol. 8, pp. 108544-108554, 2020, doi: 10.1109/ACCESS.2020.3001428.

4. T.-P. Nguyen and V.-N. Huynh, "A New Classification Technique Based on The Combination of Inner Evidence" in *IUKM 2020: Integrated Uncertainty in Knowledge Modelling*, 2020, pp. 174-186.

Bibliography

[1] Johansson, U., Sönströd, C., Norinder, U., Boström, H.: Trade-off between accuracy and interpretability for predictive in silico modeling. Future Medicinal Chemistry 3(6), 647–663 (Apr 2011)

[2] Rudin, C.: Please Stop Explaining Black Box Models for High Stakes Decisions (Nov 2018)

[3] Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining Explanations: An Overview of Interpretability of Machine Learning. arXiv:1806.00069 [cs, stat] (May 2018)

[4] Murdoch, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R., Yu, B.: Interpretable machine learning: definitions, methods, and applications. arXiv:1901.04592 [cs, stat] (Jan 2019)

[5] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv:1312.6199 [cs] (Dec 2013)

[6] Nguyen, A., Yosinski, J., Clune, J.: Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. arXiv:1412.1897 [cs] (Dec 2014), arXiv: 1412.1897

[7] Jia, R., Liang, P.: Adversarial Examples for Evaluating Reading Comprehension Systems. arXiv:1707.07328 [cs] (Jul 2017)

[8] Vellido, A.: The importance of interpretability and visualization in machine learning for applications in medicine and health care. Neural Computing and Applications (Feb 2019)

[9] Luo, Y., Tseng, H.H., Cui, S., Wei, L., Ten Haken, R.K., El Naqa, I.: Balancing accuracy and interpretability of machine learning approaches for radiation treatment outcomes modeling. BJR|Open 1(1), 20190021 (Mar 2019)

[10] Tu, J.V.: Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. Journal of Clinical Epidemiology 49(11), 1225–1231 (Nov 1996)

[11] Ravì, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., Yang, G.: Deep Learning for Health Informatics. IEEE Journal of Biomedical and Health Informatics 21(1), 4–21 (Jan 2017)

[12] Cabitza, F., Rasoini, R., Gensini, G.F.: Unintended Consequences of Machine Learning in Medicine. JAMA 318(6), 517–518 (Aug 2017)

[13] Goodman, B., Flaxman, S.: European Union regulations on algorithmic decision-making and a "right to explanation". AI Magazine 38(3), 50–57 (Oct 2017)

[14] Moraffah, R., Karami, M., Guo, R., Raglin, A., Liu, H.: Causal Interpretability for Machine Learning - Problems, Methods and Evaluation. ACM SIGKDD Explorations Newsletter 22(1), 18–33 (May 2020), https://dl.acm.org/doi/10.1145/3400051.3400058

[15] Doshi-Velez, F., Kim, B.: Towards A Rigorous Science of Interpretable Machine Learning (Feb 2017)

[16] Du, M., Liu, N., Hu, X.: Techniques for Interpretable Machine Learning. arXiv:1808.00033 [cs, stat] (May 2019), http://arxiv.org/abs/1808.00033, arXiv: 1808.00033

[17] Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 58, 82–115 (Jun 2020)

[18] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A Survey of Methods for Explaining Black Box Models. ACM Comput. Surv. 51(5), 93:1–93:42 (Aug 2018), http://doi.acm.org/10.1145/3236009

[19] Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (Mar 1986), https://doi.org/10.1007/BF00116251

[20] Piatetsky-Shapiro, G.: Discovery, analysis, and presentation of strong rules. In: Piatetsky-Shapiro, G., Frawley, W.J. (eds.) Knowledge Discovery in Databases, pp. 229–248. AAAI/MIT Press (1991)

[21] Volovoi, V.: System Reliability at the Crossroads. ISRN Applied Mathematics 2012, 850686 (Dec 2012), https://doi.org/10.5402/2012/850686, publisher: International Scholarly Research Network

[22] Stiglic, G., Kocbek, P., Fijacko, N., Zitnik, M., Verbert, K., Cilar, L.: Interpretability of machine learning based prediction models in healthcare. arXiv:2002.08596 [cs, stat] (Feb 2020), http://arxiv.org/abs/2002.08596, arXiv: 2002.08596

[23] Ahmad, M.A., Eckert, C., Teredesai, A., McKelvey, G.: Interpretable Machine Learning in Healthcare p. 7 (2018)

[24] Molnar, C.: Interpretable Machine Learning. Lulu.com (Feb 2020), google-Books-ID: RHjTxgEACAAJ

[25] Freitas, A.A., Freitas, A.A.: Comprehensible Classification Models – a position paper 15(1), 10

[26] Quinlan, J.R.: Simplifying decision trees. International Journal of Man-Machine Studies 27(3), 221–234 (Sep 1987)

[27] Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Chapman and Hall/CRC, Boca Raton, 1 edition edn. (Jan 1984)

[28] Kasim, K.: Basic Concepts of Modern Epidemiology: Epidemiology and Research. LAP Lambert Academic Publishing (Oct 2012), google-Books-ID: BVbMMgEACAAJ

[29] Esene, I.N., Ngu, J., El Zoghby, M., Solaroglu, I., Sikod, A.M., Kotb, A., Dechambenoit, G., El Husseiny, H.: Case series and descriptive cohort studies in neurosurgery: the confusion and solution. Child's Nervous System 30(8), 1321–1332 (Aug 2014), https://doi.org/10.1007/s00381-014-2460-1

[30] Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (Jan 1967), conference Name: IEEE Transactions on Information Theory

[31] Kramer, O.: K-Nearest Neighbors. In: Kramer, O. (ed.) Dimensionality Reduction with Unsupervised Nearest Neighbors, pp. 13–23. Intelligent Systems Reference Library, Springer, Berlin, Heidelberg (2013)

[32] Alahakoon, D., Halgamuge, S.K., Srinivasan, B.: Dynamic self-organizing maps with controlled growth for knowledge discovery. IEEE Transactions on Neural Networks 11(3), 601–614 (May 2000)

[33] Kohonen, T.: Self-Organizing Maps. Springer Series in Information Sciences, Springer-Verlag, Berlin Heidelberg, 3 edn. (2001)

[34] Carvalho, D.V., Pereira, E.M., Cardoso, J.S.: Machine Learning Interpretability: A Survey on Methods and Metrics. Electronics 8(8), 832 (Jul 2019)

[35] Ribeiro, M.T., Singh, S., Guestrin, C.: "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 - KDD '16. pp. 1135–1144. ACM Press, San Francisco, California, USA (2016), http://dl.acm.org/citation.cfm?doid=2939672.2939778

[36] Lundberg, S.M., Lee, S.I.: A Unified Approach to Interpreting Model Predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 4765–4774. Curran Associates, Inc. (2017)

[37] Lundberg, S.M., Nair, B., Vavilala, M.S., Horibe, M., Eisses, M.J., Adams, T., Liston, D.E., Low, D.K.W., Newman, S.F., Kim, J., Lee, S.I.: Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. Nature Biomedical Engineering 2(10), 749–760 (Oct 2018), https://www.nature.com/articles/s41551-018-0304-0, number: 10 Publisher: Nature Publishing Group

[38] Simonyan, K., Vedaldi, A., Zisserman, A.: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv:1312.6034 [cs] (Apr 2014), arXiv: 1312.6034

[39] Pearl, J.: Theoretical Impediments to Machine Learning With Seven Sparks from the Causal Revolution. arXiv:1801.04016 [cs, stat] (Jan 2018), http://arxiv.org/abs/1801.04016, arXiv: 1801.04016

[40] Chattopadhyay, A., Manupriya, P., Sarkar, A., Balasubramanian, V.N.: Neural Network Attributions: A Causal Perspective. arXiv:1902.02302 [cs, stat] (Jul 2019), http://arxiv.org/abs/1902.02302, arXiv: 1902.02302

[41] Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., Lee, S.: Counterfactual Visual Explanations. arXiv:1904.07451 [cs, stat] (Jun 2019), http://arxiv.org/abs/1904.07451, arXiv: 1904.07451

[42] Yang, F., Du, M., Hu, X.: Evaluating Explanation Without Ground Truth in Interpretable Machine Learning. arXiv:1907.06831 [cs, stat] (Jul 2019), http://arxiv.org/abs/1907.06831, arXiv: 1907.06831

[43] Letham, B., Rudin, C., McCormick, T.H., Madigan, D.: Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. The Annals of Applied Statistics 9(3), 1350–1371 (Sep 2015), arXiv: 1511.01644

[44] Chugh, S.S., Havmoeller, R., Narayanan, K., Singh, D., Rienstra, M., Benjamin, E.J., Gillum, R.F., Kim, Y.H., McAnulty, J.H., Zheng, Z.J., Forouzanfar, M.H., Naghavi, M., Mensah, G.A., Ezzati, M., Murray, C.J.L.: Worldwide Epidemiology of Atrial Fibrillation: A Global Burden of Disease 2010 Study. Circulation 129(8), 837–847 (Feb 2014), https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4151302/

[45] Kamel Hooman, Okin Peter M., Elkind Mitchell S.V., Iadecola Costantino: Atrial Fibrillation and Mechanisms of Stroke. Stroke 47(3), 895–900 (Mar 2016),

https://www.ahajournals.org/doi/10.1161/strokeaha.115.012004, publisher: American Heart Association

[46] Wolf, P.A., Dawber, T.R., Thomas, H.E., Kannel, W.B.: Epidemiologic assessment of chronic atrial fibrillation and risk of stroke: the Framingham study. Neurology 28(10), 973–977 (Oct 1978)

[47] Gage, B.F., Waterman, A.D., Shannon, W., Boechler, M., Rich, M.W., Radford, M.J.: Validation of Clinical Classification Schemes for Predicting Stroke: Results From the National Registry of Atrial Fibrillation. JAMA 285(22), 2864–2870 (Jun 2001), https://jamanetwork.com/journals/jama/fullarticle/193912, publisher: American Medical Association

[48] Roy, K., Kar, S., Das, R.N.: A Primer on QSAR/QSPR Modeling. SpringerBriefs in Molecular Science, Springer International Publishing, Cham (2015), http://link.springer.com/10.1007/978-3-319-17281-1

[49] Ghasemi, F., Mehridehnavi, A., Pérez-Garrido, A., Pérez-Sánchez, H.: Neural network and deep-learning algorithms used in QSAR studies: merits and drawbacks. Drug Discovery Today 23(10), 1784–1790 (Oct 2018), http://www.sciencedirect.com/science/article/pii/S1359644617304762

[50] Mikulskis, P., Alexander, M.R., Winkler, D.A.: Toward Interpretable Machine Learning Models for Materials Discovery. Advanced Intelligent Systems 1(8), 1900045 (2019), https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.201900045, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aisy.201900045

[51] Grath, R.M., Costabello, L., Van, C.L., Sweeney, P., Kamiab, F., Shen, Z., Lecue, F.: Interpretable Credit Application Predictions With Counterfactual Explanations. arXiv:1811.05245 [cs] (Nov 2018), http://arxiv.org/abs/1811.05245, arXiv: 1811.05245

[52] Managing Disruption | Moody's Analytics, https://www.moodysanalytics.com/risk-perspectives-magazine/managing-disruption

[53] Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. arXiv:1711.00399 [cs] (Mar 2018), http://arxiv.org/abs/1711.00399, arXiv: 1711.00399

[54] Nguyen, T.P., Dinh, D.T., Huynh, V.N.: A New Context-Based Clustering Framework for Categorical Data. In: Geng, X., Kang, B.H. (eds.) PRICAI 2018: Trends in Artificial Intelligence. pp. 697–709. Lecture Notes in Computer Science, Springer International Publishing (2018)

[55] Berkhin, P.: A survey of clustering data mining techniques. In: Grouping multidimensional data, pp. 25–71. Springer (2006)

[56] Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM computing surveys (CSUR) 31(3), 264–323 (1999)

[57] MacQueen, J.: Some methods for classification and analysis of multivariate observations. The Regents of the University of California (1967)

[58] Huang, Z.: Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. Data Mining and Knowledge Discovery 2(3), 283–304 (Sep 1998)

[59] San, O.M., Huynh, V.N., Nakamori, Y.: An alternative extension of the k-means algorithm for clustering categorical data. International Journal of Applied Mathematics and Computer Science 14, 241–247 (2004)

[60] Chen, L., Wang, S.: Central clustering of categorical data with automated feature weighting. In: Twenty-Third International Joint Conference on Artificial Intelligence (2013)

[61] Nguyen, T.H.T., Huynh, V.N.: A k-Means-Like Algorithm for Clustering Categorical Data Using an Information Theoretic-Based Dissimilarity Measure. In: Foundations of Information and Knowledge Systems. pp. 115–130. Springer, Cham (Mar 2016)

[62] Cohen, J., Cohen, P.: Applied multiple regression/correlation analysis for the behavioral sciences. L. Erlbaum Associates, Hillsdale, N.J. (1983)

[63] Ralambondrainy, H.: A conceptual version of the k-means algorithm. Pattern Recognition Letters 16(11), 1147–1157 (1995)

[64] Ienco, D., Pensa, R.G., Meo, R.: From Context to Distance: Learning Dissimilarity for Categorical Data Clustering. ACM Trans. Knowl. Discov. Data 6(1), 1:1–1:25 (Mar 2012)

[65] Lichman, M.: UCI machine learning repository (2013), `http://archive.ics.uci.edu/ml`

[66] Huang, J.Z., Ng, M.K., Rong, H., Li, Z.: Automated variable weighting in k-means type clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(5), 657–668 (May 2005)

[67] Au, W.H., Chan, K.C.C., Wong, A.K.C., Wang, Y.: Attribute Clustering for Grouping, Selection, and Classification of Gene Expression Data. IEEE/ACM Trans. Comput. Biol. Bioinformatics 2(2), 83–101 (Apr 2005)

[68] MacKay, D.J.C.: Information Theory, Inference & Learning Algorithms. Cambridge University Press, New York, NY, USA (2002)

[69] Hall, M.A., Holmes, G.: Benchmarking attribute selection techniques for discrete class data mining. IEEE Transactions on Knowledge and Data engineering 15(6), 1437–1447 (2003)

[70] Nguyen, T.P., Nguyen, S., Alahakoon, D., Huynh, V.N.: GSIC: A New Interpretable System for Knowledge Exploration and Classification. IEEE Access 8, 108544–108554 (2020), conference Name: IEEE Access

[71] Seger, C.A., Miller, E.K.: Category Learning in the Brain. Annual review of neuroscience 33, 203–219 (2010)

[72] Kohonen, T.: Essentials of the self-organizing map. Neural Networks 37, 52–65 (Jan 2013)

[73] Johnson, A.E.W., Pollard, T.J., Shen, L., Lehman, L.w.H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., Mark, R.G.: MIMIC-III, a freely accessible critical care database. Scientific Data 3, 160035 (May 2016)

[74] Johnson, A.E.W., Aboab, J., Raffa, J.D., Pollard, T.J., Deliberato, R.O., Celi, L.A., Stone, D.J.: A Comparative Analysis of Sepsis Identification Methods in an Electronic Database. Critical Care Medicine 46(4), 494–499 (Apr 2018)

[75] Dua, D., Graff, C.: UCI machine learning repository (2017), `http://archive.ics.uci.edu/ml`

[76] Fleischmann, C., Scherag, A., Adhikari, N.K.J., Hartog, C.S., Tsaganos, T., Schlattmann, P., Angus, D.C., Reinhart, K.: Assessment of Global Incidence and Mortality of Hospital-treated Sepsis. Current Estimates and Limitations. American Journal of Respiratory and Critical Care Medicine 193(3), 259–272 (Sep 2015)

[77] Reinhart, K., Daniels, R., Kissoon, N., Machado, F.R., Schachter, R.D., Finfer, S.: Recognizing Sepsis as a Global Health Priority — A WHO Resolution. New England Journal of Medicine 377(5), 414–417 (Aug 2017)

[78] Singer, M., Deutschman, C.S., Seymour, C.W., Shankar-Hari, M., Annane, D., Bauer, M., Bellomo, R., Bernard, G.R., Chiche, J.D., Coopersmith, C.M., Hotchkiss, R.S., Levy, M.M., Marshall, J.C., Martin, G.S., Opal, S.M., Rubenfeld, G.D., Poll, T.v.d., Vincent, J.L., Angus, D.C.: The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). JAMA 315(8), 801–810 (Feb 2016)

[79] Nguyen, T.P., Huynh, V.N.: A New Classification Technique Based on the Combination of Inner Evidence. In: Huynh, V.N., Entani, T., Jeenanunta,

C., Inuiguchi, M., Yenradee, P. (eds.) Integrated Uncertainty in Knowledge Modelling and Decision Making. pp. 174–186. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020)

[80] Begoli, E., Bhattacharya, T., Kusnezov, D.: The need for uncertainty quantification in machine-assisted medical decision making. Nature Machine Intelligence 1(1), 20–23 (Jan 2019), number: 1 Publisher: Nature Publishing Group

[81] Vluymans, S., Cornelis, C., Saeys, Y.: Machine Learning for Bioinformatics: Uncertainty Management with Fuzzy Rough Sets p. 2

[82] Huynh, V.N.: Uncertainty Management in Machine Learning Applications. International Journal of Approximate Reasoning 107, 79–80 (Apr 2019), https://linkinghub.elsevier.com/retrieve/pii/S0888613X19300672

[83] Hüllermeier, E.: Uncertainty in Clustering and Classification. In: Deshpande, A., Hunter, A. (eds.) Scalable Uncertainty Management. pp. 16–19. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2010)

[84] Jousselme, A.L., Maupin, P., Bosse, E.: Uncertainty in a situation analysis perspective. In: Sixth International Conference of Information Fusion, 2003. Proceedings of the. vol. 2, pp. 1207–1214 (Jul 2003)

[85] Shafer, G.: A Mathematical Theory Of Evidence. Princeton University Press (1976)

[86] Denoeux, T., Bjanger, M.: Induction of decision trees from partially classified data using belief functions. In: Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0. vol. 4, pp. 2923–2928 vol.4 (Oct 2000), iSSN: 1062-922X

[87] Denœux, T., Kanjanatarakul, O., Sriboonchitta, S.: A new evidential K-nearest neighbor rule based on contextual discounting with partially supervised learning. International Journal of Approximate Reasoning 113, 287–302 (Oct 2019)

[88] Denoeux, T.: A k-nearest neighbor classification rule based on Dempster-Shafer theory. IEEE Transactions on Systems, Man, and Cybernetics 25(5), 804–813 (May 1995), conference Name: IEEE Transactions on Systems, Man, and Cybernetics

[89] Tong, Z., Xu, P., Denœux, T.: ConvNet and Dempster-Shafer Theory for Object Recognition. In: Ben Amor, N., Quost, B., Theobald, M. (eds.) Scalable Uncertainty Management, vol. 11940, pp. 368–381. Springer International Publishing, Cham (2019), series Title: Lecture Notes in Computer Science

[90] Su, Z., Hu, Q., Denaeux, T.: A Distributed Rough Evidential K-NN Classifier: Integrating Feature Reduction and Classification. IEEE Transactions on Fuzzy Systems pp. 1–1 (2020), conference Name: IEEE Transactions on Fuzzy Systems

[91] Denoeux, T.: 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 (Mar 2000), conference Name: IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans

[92] Liu, Z.g., Pan, Q., Dezert, J., Mercier, G.: Credal classification rule for uncertain data based on belief functions. Pattern Recognition 47(7), 2532–2541 (Jul 2014)

[93] Liu, Z.g., Pan, Q., Dezert, J.: A new belief-based K-nearest neighbor classification method. Pattern Recognition 46(3), 834–844 (Mar 2013)

[94] Dempster, A.P.: Upper and Lower Probabilities Induced by a Multivalued Mapping. In: Yager, R.R., Liu, L. (eds.) Classic Works of the Dempster-Shafer Theory of Belief Functions, pp. 57–72. Studies in Fuzziness and Soft Computing, Springer, Berlin, Heidelberg (2008)

[95] Zadeh, L.A.: Review of books: A mathematical theory of evidence. The AI Magazine 5(3), 81–83 (1984)

[96] Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics 3(3), 32–57 (Jan 1973), publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01969727308546046

[97] Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, USA (1981)

[98] Bezdek, J.C., Ehrlich, R., Full, W.: FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences 10(2), 191–203 (Jan 1984)

[99] Ross, T.J.: Fuzzy logic with engineering applications. John Wiley, Chichester, U.K, 3rd ed edn. (2010), oCLC: ocn430736639

[100] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA (1984)

[101] Smets, P., Kennes, R.: The transferable belief model. Artificial Intelligence 66(2), 191–234 (Apr 1994)

[102] Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining Explanations: An Overview of Interpretability of Machine Learning. arXiv:1806.00069 [cs, stat] (Feb 2019), http://arxiv.org/abs/1806.00069, arXiv: 1806.00069

[103] Lipton, Z.C.: The Mythos of Model Interpretability. arXiv:1606.03490 [cs, stat] (Mar 2017), http://arxiv.org/abs/1606.03490, arXiv: 1606.03490