Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Máster Universitario en Inteligencia Artificial

Trabajo Fin de Máster

Interpreting Bayesian Network-based Clustering

Author: Víctor Alejandre Jiménez
Tutor(a): Concha Bielza Lozoya y Pedro Larrañaga Múgica

Madrid, 10-2023

Trabajo Fin de Máster
Máster Universitario en Inteligencia Artificial

Title: Interpreting Bayesian Network-based Clustering

10-2023

Author: Víctor Alejandre Jiménez
Tutor: Concha Bielza Lozoya y Pedro Larrañaga Múgica
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

Clustering es el principal problema del paradigma de aprendizaje no supervisado dentro del *machine learning*. Este problema consiste en agrupar individuos de una población en base a sus características a partir de una muestra de dicha población. El objetivo principal es obtener información de la población únicamente a partir de los datos lo cual lo convierte en un problema muy complejo. Esto se debe a que, a diferencia de otros problemas como la clasificación o la regresión, dentro del *clustering*, y en general en aprendizaje no supervisado, no se tiene información sobre los posibles resultados.

El hecho de no tener ningún tipo de conocimiento respecto a los posibles resultados que se pueden obtener al resolver el problema no solo hace que este sea complejo, sino que también aparezca la necesidad de explicaciones pues pueden aparecer preguntas acerca de los resultados para su comprensión y validación. Un ejemplo de ello es el etiquetado de los *clusters*. Este problema está muy relacionado con la caracterización de los *clusters* que no es más que identificar información que diferencia e identifica cada grupo obtenido. Así una de las aproximaciones mas comunes para el etiquetado de los *clusters* pasa por encontrar una caracterización de los grupos y dotar de un nombre representativo a partir de esta. De esta forma el problema del *clustering*, al igual que muchos otros, interseca con el campo de las explicaciones en inteligencia artificial (IA), el cual ha adquirido una gran importancia en los últimos años aunque hay que destacar que aún no se ha desarrollado una teoría consolidada.

En este contexto, el uso de modelos interpretables es importante. Esto se debe a que estos modelos pueden aportar a resolver el problema de forma óptima y así obtener información relevante. Pero, ¿qué es un modelo interpretable?, ¿por qué en este caso pueden ser útiles? ¿qué modelo interpretable se debería utilizar?. A estas preguntas da respuesta el campo de las explicaciones en IA.

El objetivo de este trabajo es mostrar la versatilidad y utilidad que presenta el modelo de red Bayesiana dentro del contexto de las explicaciones en IA, y particularmente en el problema de *clustering*. Para ello se va a estudiar el estado del arte en el campo de las explicaciones haciendo hincapié en la falta de consolidación de la teoría desarrollada en este. Posteriormente, se ubicará a las redes Bayesianas dentro de este campo a través del estado del arte de las herramientas que ofrece este modelo dentro del contexto de las explicaciones en IA.

Una vez establecido el marco teórico del trabajo se presentará el estado del arte en la resolución del problema del *clustering* con redes Bayesianas. Finalmente, se establecerá una nueva metodología que aprovecha la interpretabilidad de las redes Bayesianas para poder caracterizar los *clusters* que se pueden obtener y con ello resolver el problema del etiquetado de los *clusters*.

Esta metodología tiene base en la aproximación habitual para la caracterización de *clusters* que consiste en encontrar individuos representativos de estos. Estos individuos permiten identificar de forma sencilla las características que diferencian unos *clusters* de otros. De esta forma ofrecen una visión compacta de los *clusters* que facilita su etiquetado. Además, aprovechando la interpretabilidad de las redes Bayesianas esta nueva metodología introduce una nueva medida de importancia para las variables. Esta medida es introducida con el objetivo de permitir discernir qué variables son importantes dado cada representante de cada *cluster*, de forma que se facilita la comparación y comprensión de estos.

Abstract

Clustering is the main problem of the unsupervised learning paradigm within machine learning. This problem consists of grouping individuals from a population based on their characteristics. This grouping is usually obtained from a sample of individuals of the population. The main objective is to obtain information about the population only from the data, which makes it a very complex problem. This is because unlike other problems such as classification or regression, within clustering and in general in unsupervised learning there is no information about the possible outcomes.

Not having any knowledge about the possible results that can be obtained by solving the problem, not only makes the problem complex, but also creates the need for explanations, as questions about the results may arise for the understanding and validation of the results. An example of this is the naming of clusters. This problem is closely related to the characterization of clusters, which is nothing more than identifying information that differentiates and identifies each group obtained. Thus, one of the most common approaches for labeling the clusters is to find a characterization of the groups and provide a representative name based on it. In this way the problem of clustering, like many others, intersects with the field of explanations in artificial intelligence (AI). This field has acquired great importance in recent years, although it should be noted that a consolidated theory does not exist yet.

In this context, the use of interpretable models is important. This is because these models can contribute to solve the problem in an optimal way and thus obtain relevant information. But, what is an interpretable model?, why can they be useful in this case?, which interpretable model should we use? These questions are answered by the field of explanations in AI.

The aim of this work is to show the versatility and usefulness of the Bayesian network model within the context of explanations in AI, and particularly in the problem of clustering. For this purpose, the state of the art in the field of explanations in AI will be studied, emphasizing the lack of a formal ground theory in this field. Subsequently, Bayesian networks will be placed within this field through the state of the art of the tools offered by this model for generating explanations in the context of AI.

Once the theoretical framework of the work has been established, the state of the art in the resolution of the clustering problem with Bayesian networks will be presented.

Finally, we present a new methodology that takes advantage of the interpretability of Bayesian networks in order to characterize the clusters that can be obtained and thus solve the problem of cluster naming.

This methodology is based on the usual approach for the characterization of clusters, which consists of finding representative individuals of these clusters. These individuals allow to identify in a simple way the characteristics that differentiate a cluster from the others. Therefore, the representatives offer a compact view of the clusters that facilitates their labeling. Furthermore, taking advantage of the interpretability of Bayesian networks, this new methodology introduces a new measure of feature importance. The objective of introducing this measure is to allow discerning which variables are important given each representative of each cluster in a way that facilitates the comparison and understanding of the clusters.

Contents

Chapter 1

Introduction to XAI

1.1 Why? That's the question

Why, a simple word which represents one of the biggest double-edged sword in human history. This straightforward question has been the basis to every development in society, regardless of its nature. This is due to the fact that it encapsulates the interaction of humans with their environment since our actions are based on our understanding and knowledge and that is acquired with the answer to "why". The other edge of the sword comes when we have to give an answer to the question, i.e., an explanation. Let's take the following example:

Imagine that your car engine stops working, so in order to solve this problem you need to know why it has stopped working. Thus you pop the hood and find that the engine coolant tank is empty; you know that without that component the engine overheats and stops. Given this situation, the answer to the question "*why did the engine stop working?*" would be because the engine coolant tank was empty. If there was just a single answer, with a simple refill of the tank we would solve the problem and there would not be another course of action.

With respect to the previous example now imagine that you also see the oil level below the minimum needed. Now think about the following scenarios:

- You know that oil is required to lubricate and refrigerate the engine and without it the engine might seize (its metal components expand and block each other impeding the movement). Now your answer can be "the engine has stopped because it had no oil", or "the engine has stopped because there was no engine coolant" or "the engine has stopped because it had neither oil nor engine coolant".
- You do not know the functionality of oil so you ignore it and only fill up the engine coolant tank. In this case your only answer is "*because there was no engine coolant*". Moreover, if you were given the answer "*the engine has stopped because it had neither oil nor engine coolant*" you would not accept it because, for you, the absence of oil does not matter. In the best case scenario you would ask why the absence of oil is part of the answer.

So, what is the answer to the question "*why did the engine stop working?*"? Is it "*because there was no engine coolant*"? That is a truth so it is a plausible answer, although one might argue that it is incomplete in the sense that the engine shutdown was caused by the low level of oil too. But if we talk about completeness we have to say complete regarding what. In this case it is completeness with respect to the information we have, which is the absence of oil and engine coolant. Consequently, the answer would be "*the engine has stopped because it had neither oil nor engine coolant* ".

We have to make this distinction because if we talk about completeness with respect to the knowledge of the user who needs the explanation in the second scenario, then the answer would be "*because the there was no engine coolant*". Furthermore, if we talk about completeness with respect to the whole issue, i.e., all plausible problems that make the engine stop and their causes, then we would need to discard every one of them but absence of oil and engine coolant so the answer would be "*the engine has stopped because it had neither oil nor engine coolant* ".

Now if we think about the complexity of establishing a complete explanation, the question that arises is: is it worth it? If you were in a life-and-death situation that would probably be the case. If you were in your garage on a Sunday afternoon probably the answer "*because there was no engine coolant*" would be good for you. And in the case that filling up the tank was not sufficient then you would try to find another one (the oil level is too low, for example).

Up to this point some questions can arise from the discussion just held. For example, one might ask if there are unique explanations. Also, this question brings to us the question: are there desired properties for an explanation? Furthermore, what is an explanation? Do different kinds of explanations exist?

So again, what is the answer to the question "*why did the engine stop working?*"? After the little discussion proposed, it has been shown that in order to answer that question one has to study the blurred concept of explanation. This is not new, as stated in Mittelstadt et al. (2019), this subject has been already studied in fields such as law, cognitive science, philosophy and social sciences.

But the purpose of this work is not to deepen the concept of "explanation" from the perspective of the fields mentioned. This discussion has been held to understand the big picture of explanations in artificial intelligence (AI), which we are going to take a peek at. AI and, in particular, machine learning (ML) has expanded and conquered its own space in a lot of environments related to humans, and so in the present scenario, the absolute need to answer the question Why? has arisen.

Discussions and examples of the need of explanations in AI can be found in Molnar (2020) and Rudin (2019). We propose here an image tumor detector as a little example of the vast variety that exists. Thus, when our model detects a tumor in an image, an explanation of why it has been detected can be used to gain insight into the study of tumors or just for the expert to trust the conclusion.

The purpose of this work is to take a peek at the big picture of explainable AI (XAI) (see Figure 1.1) in order to show the versatility and utility of Bayesian networks. Hence, the focus will be on the whole picture regardless of the critiques there are of some lines of work (Rudin (2019)). The reason for doing this is that this work will adopt in its approach a descriptive perspective. In other words, we are not trying to discern whether there is a best line of work; we are trying to describe what are the limitations and properties of the different lines of work that exist and the role that Bayesian networks play. Particularly, once we have analysed them, we will focus on the subject of clustering with Bayesian networks.

1.2 State of the art on XAI

As it can be seen in works mentioned in the previous subsection (Rudin (2019), Molnar (2020)) ML requires explanations for different purposes such as debbuging, trusting predictions, detecting possible bias, etc,... Does this mean that we always need them? As Doshi-Velez and Kim (2017) state, the need of explanations comes from incompleteness in problem formalization impeding optimization and evaluation. Thus, when there are no significant consequences to decisions or the problem is well studied there is no demand of explanations. But, as one can imagine that is not the usual scenario.

So it is said in this work to distinguish between incompleteness and uncertainty. Uncertainty can be measured in terms of variance and is present in the majority of problems and domains. Incompleteness can be seen as unknown knowledge manifested in terms of unquantified bias in our models. The perfect example here can be seen practically in every work on ML in the iterative process of creating its models.

This process comprises, among other things, the understanding of data and predictors in order to determine which are relevant to the problem at hand. Not knowing which variables are important is a form of incompleteness that can lead to bias independently of either the model used or the domain defined for the problem. Since we are learning models from data there is always uncertainty in the predictions of the models as not all the casuistic is represented in the data. In this sense, explanations are a key to differenciate between the uncertainty that comes from learning from data and incompleteness in domain definition.

As an example of this, Rudin (2019) claims that the use of explanations derived from interpretable models helped in improving performance and revealing false assumptions about the data generation process in a project predicting New York grid failures. This is a thought to bear in mind since we will see that Bayesian networks are an interpretable model that also allows us to model uncertainty in terms of probability.

Once stated the need of explanations in ML, which is something every author who has written about this topic agrees on, we are going to overview explanations in AI by talking about the what and the how. Through the discussion and at the end of this we will present some works on the relationship between these two subjects. Keep in mind that the goal here is a purely descriptive overview of the topic, so we will not discuss how appropriate the ideas showed are but rather what they express.

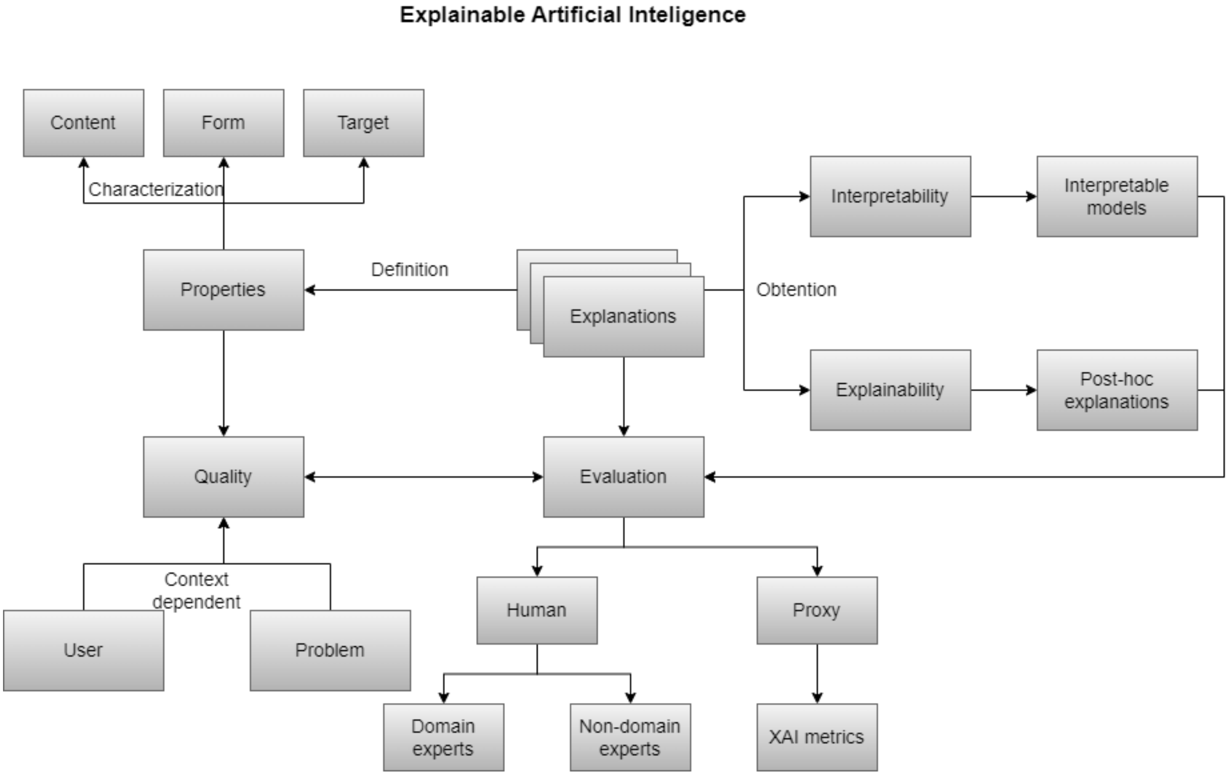


Figure 1.1: XAI overview.

1.2.1 What are explanations?

We start with the what. As shown in the previous section it is a difficult task to grasp the concept of explanation and that also translates into the field of AI. Most of the works done on this subject focus on the properties an explanation should or does fulfill directly, assuming the definition of explanation as an answer to a question why. Some of these works mention Miller (2019) to make this statement; others directly do not even make the effort to give this basis. This is not a surprise, since the concept of explanation lends itself to being defined with regard to the context it is being used in, always taking into account the underlying idea of being the answer to a question why. This is also no big news as the discussions held about explanations in a more "philosophical" way have taken this approach (Doshi-Velez and Kim (2017)).

We are going to distinguish between properties that characterize the explanations (see Figure 1.2) and properties that are established in order to measure their quality. When talking about the first ones, we reference Lacave and Díez (2002) and Burkart and Huber (2021). In both works they establish a form to differentiate types of explanations based on content (what to explain), form (how the explanation is presented), and target (to whom the explanation is given). This is done according to the idea that these are basic properties when trying to conceive an explanation in a specific context. So these two works focus on settling and describing these basic properties in the context of AI. We highlight that these works are developed with the objective of classifying/creating methods for generating explanations (Lacave and Díez (2002)) according to these elemental characteristics.

Based on content, Burkart and Huber (2021) discern that there exist five types of explanation: prototype (local), criticism (local), local, counterfactual (local) and global. On the other hand, Lacave and Díez (2002) dive deeper into this subject and establish different issues when talking about content: focus, purpose, causality and level of the explanation.

The focus of the explanation refers to which level: model, process of reasoning or evidence, is the explanation about. The purpose of an explanation can be description or comprehension. With respect to the level, the authors describe this section based on Bayesian networks talking about micro-level and macro-level. This is the analogue in Burkart and Huber's work when talking about global or local explanations. Finally, when talking about causality, the authors remark that there exists a tied relationship between explanation and cause. In certain domains causality is studied and needed; on the other hand not all models can deal with causality, hence the importance of establishing whether an explanation content is related to this concept.

When discussing about the form, Burkart and Huber (2021) establish that there exist three forms: textual description, graphical description and multimedia. The latter form consists of combining the first two. Lacave and Díez (2002) also make this distinction and add the user-system interaction criteria. Based on these criteria we talk about the way an explanation is presented in terms of how the user asks the system and how the latter responds. For example the user can ask by giving a natural language question or just some selected variables. Moreover, questions can be made during certain processes, as for example learning, whereas in some cases they can only be made until the final model is obtained.

Finally, in the work of Burkart and Huber (2021) they explain that, based on the knowledge the user may have, there exist different types of users who can be the target of the explanation. They state that there are different levels of user knowledge in both the domain of the task at hand and the model used, hence the explanations exposed and their complexity (we should highlight that they do not give a formal definition of complexity) must be in concordance with this, otherwise they are not useful. Based on this they define four different groups of users:

- **Non expert**. User with no knowledge of both domains (task and model). Thus, the explanation must be simple and easily comprehensible. We highlight that, as happens with the concept of complexity, no formal definition of a comprehensible explanation is given.

- **Domain expert**. User with knowledge of the domain of the task at hand. They clarify that this user needs more complex explanations in order to understand more deeply the system and task at hand. This will also help to gain trust in the model.
- **System developer**. A technical expert but with none or little task domain knowledge. Here the general purpose is to understand the system to guarantee its functionality.
- **AI developer**. Similar to system developer, although its general goal is to train the model. Its need for explanations comes with the goal of debbuging the system to improve it from a technical point of view (e.g., its performance).

On the other hand, Lacave and Díez (2002) expose similar ideas but with a totally different organization. They state that the effectiveness of an explanation partially comes from the ability to address user's specific needs and that essentially depends on their knowledge. They dictate that users's knowledge in the domain of the task at hand and the reasoning method must be taken into account independently as also exposed in Burkart and Huber (2021).

As stated above, these works are done in order to define these fundamental properties or dimensions of explanations in the context of AI with the goal of taking them into account when modeling a task where explanations are needed. But this characterization is not enough as it presents a great void when talking about the quality of an explanation. Even though these properties must be present when modeling a task which needs some kind of explanation, no one can assert whether an explanation is good or not based on them.

Furthermore, the only property which could be used to discern the quality of an explanation is the target, since part of the effectiveness of an explanation depends on the adaptation to the user. So a good explanation would be defined as the one which adapts to the target it is intended to. However, even with this consideration one could argue that there is still a lack of formalism (most works do not present a definition of complexity in order to address the intended target) and there are also aspects that are not contemplated such as uncertainty or generality. Hence the works that we are going to be discuss next, but first we are going to talk about what we define as the human factor.

What we contemplate as human factor is nothing else than humans being the center of the world of explanations in the sense that explanations are made by humans for humans. We will highlight that in the field of AI, models are the ones which give explanations but is the human who defines these models and hence the explanations they can elaborate. As a consequence, or more correctly as a tautology, the humans (the receivers of the explanation) are the final evaluator of the quality of an explanation. This makes the evaluation of the quality of explanations entirely subjective. Thus, the complexity when dealing with the concept of explanation, as the dependency on humans being the evaluators creates the dependency on the context.

At this point the proper question is: Can we define what is a good explanation in-

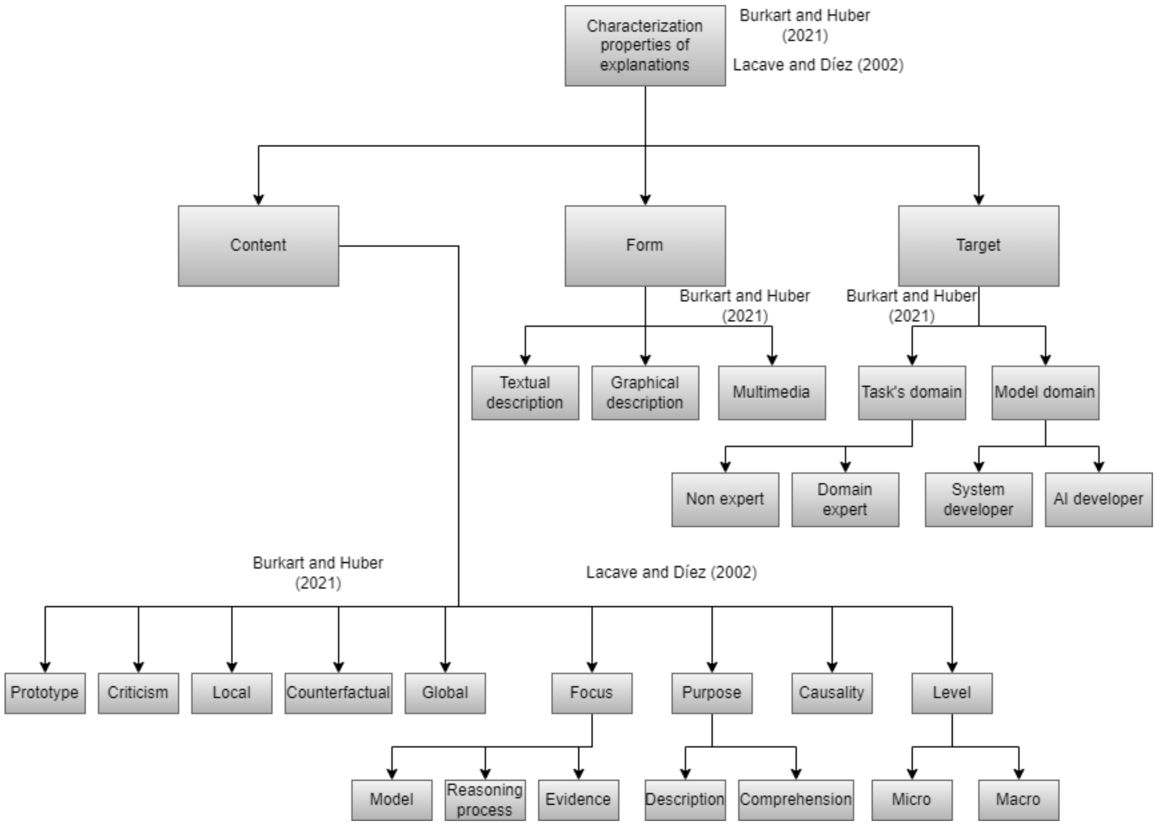


Figure 1.2: Characterization properties of explanations in the context of AI.

dependently of the context, i.e., in an objective way? If not, can it be defined in a specific context? Under the light of the ideas exposed the answer to the first question would be no, as just the different knowledge of humans would be enough to show the subjectiveness of the matter.

With respect to the second one, other questions arise such as what are the elements of the specific context that one has to take into account to define what is a good explanation? Are there key elements to differentiate in an objective way different types of contexts? Given these elements, would it be possible to objectively establish the properties a good explanation should have? And should the elements used to differentiate contexts be defined with respect to the properties or ideas of what a good explanation should be?

Up to this point it is enough to see where the complexity surrounding the concept of explanation comes from. As said before this concept has been studied in different fields and there is not yet a general and objective manner to answer the question "What is a good explanation?". We do not even understand how the human brain works which plays a key role on understanding how we communicate and give explanations to each other. But one probably has a better question: Why bothering establishing an objective approach to something that is apparently subjective?

A first thing is that having a formal and objective way of evaluating something sets up a systematic line of work for everybody with no room for error but the formalism being wrong. Secondly, as far as the human body of knowledge reaches, the evaluation of the quality remains subjective as we cannot answer in an objective manner, but also there isn't anything that states that this objective evaluation can not be accomplished. So trying to find an objective approach to these matters not only has the benefits of formalisms, but also can help to explore and broaden the big body of human knowledge, like for example understanding the human brain.

We are not going to dive deeper into this matter as the purpose of the discussion held is to show where the lack of formalism and uniformity and the variety of ideas in the works that are going to be presented come from. The ideas presented were taken from Doshi-Velez and Kim (2017). This work reflects the root of the problems in objectiveness when evaluating the quality of an explanation, i.e., the human factor, in the context of AI and more specifically ML. The authors establish, in a data-driven context, that there are three types of evaluation:

- **Application-grounded**. Involves evaluating the quality of explanations via domain experts in a particular real task.
- **Human-grounded**. Involves evaluating the quality of the explanations in simpler tasks without the need of domain experts. This is appropriate to evaluate general notions of explanations. The difficulty here is to create forms of evaluation without a specific end-goal. Examples are shown in Doshi-Velez and Kim (2017) such as giving a binary forced choice (pick between two explanations given).
- **Functionally-grounded**. The idea here is to use a proxy to evaluate explanation quality. A proxy is a formal definition of interpretability and the challenge is which one is to be used.

As for the last type presented we have not talked about interpretability yet, but the underlying idea is the key: use a particular formalism to evaluate quality. Bearing all of this in mind we present the work Molnar (2020). Here the formalism comes in the form of properties. In his work Molnar differentiates two groups of properties an explanation should fulfill: one comes from the humanities and social sciences and the other one is related to the ML field.

We start with the properties coming from the humanity and social sciences. Here the author bases his entire work on Miller (2019), which is a survey of publications on explanations. These properties are based on how humans explain things to each other so they measure the quality of what can be defined as "everyday"-type explanations. The following are presented:

- **Explanations are contrastive**. As stated by Lipton (1990), humans usually do not ask why a certain prediction was made, but why this prediction was made instead of another prediction. So a contrastive explanation is one which entails differenciating between the case of study and another reference case(s).
- **Explanations are selected.** This property takes basis on the Rashomon effect, which is the fact that an event can be explained by various and different causes (Anderson (2016)). As so, humans prefer/expect explanations to encapsulate some (usually a few) of the causes of the event and not all of them.

- **Explanations are social.** This property is related to the social context of the explainer and the receiver, similar to the target property stated in the works of Lacave and Díez (2002) and Burkart and Huber (2021). A good explanation must take into account the social context to determine its content and nature.
- **Explanations focus on the abnormal.** Here the idea is that humans focus on abnormal causes to explain things to each other. An abnormal cause is one which is rare, i.e., has low probability of happening, hence the fact of humans focusing on them.
- **Explanations are truthful.** The concept encapsulated by this property is that a given explanation has to be reliable in the sense that it must be trustworthy with respect to the real world. It does not make sense giving an explanation based on facts that do not make sense or happen in the real world. Neither is correct that what is expressed in the explanation does not see eye to eye the reality.
- **Explanations are consistent with prior beliefs of the explainee.** Here we have got what is called as confirmation bias: humans tend to devalue an explanation if it does not agree with their prior beliefs. As so, a good explanation must take into account the prior beliefs of the explainee.
- **Good explanations are general and probable.** The idea here is that a good explanation is one that applies to various and different events. This property creates a conflict with abnormal explanations as by definition they are not general and do not apply to different events. According to Molnar (2020), the abnormal property beats this one and in case the property of abnormality is not present is when this one can be good for measuring the quality of explanations.

The next group of properties are defined in an ML context, specifically in surpervised ML. Thus, it is not odd that some of the ideas share their basis with some of the properties presented before as they are defined in a general way. The properties are:

- **Accuracy**. This property states the idea of an explanation being good when unseen data can be predicted from it, i.e., once having an explanation one can predict the value of new points which are not present in the initial data used.
- **Fidelity**. In this case this property is defined within the context of *post-hoc* explanations (as this has not been discussed yet, later it will be reminded to ensure the understanding of it). The idea is similar to accuracy, but in this case a good explanation is one that allows to approximate the prediction of black-box models.
- **Consistency**. Here the idea is to measure the quality of an explanation based on how similar is compared to others obtained by different models used to solve the task at hand. Notice that the premise here is working on the same basis (data and variables). This property fails to measure the quality of the explanation if different models with different basis get similar predictions (Rashomon effect).
- **Stability**. The underlying concept of this property is robustness, i.e., slight changes on the input do not produce substantially changes on the output. In the context of ML this translates to similar explanations for similar instances. So, if we vary an instance producing a small change and this produces a very different explanation then there is a lack of stability.

- **Comprehensibility**. In this case this property makes reference to how well the explanation is understood. Again the basis of this property (as the social property) is to address properly the target of the explanation. We have a good explanation when the explainee understands it.
- **Certainty**. A good explanation is one that includes or makes a statement about the model's certainty of the prediction.
- **Degree of importance**. An explanation is good when it reflects the importance of the parts that constitute it. An example would be including the role of the variables involved in a prediction.
- **Novelty**. In this case a good explanation is one that expresses whether the data instance being explained comes from a part of the distribution that has been explored or not. This can be seen as a particular instance of the property of certainty. The underlying idea is that within regions of the distribution that have not been explored the model can be inaccurate, thus the explanation given is useless.
- **Representativeness.** This property makes reference to the extent of data instances that can be explained with the explanation. So, the quality of the explanation is measured in terms of its capacity to explain different data instances, i.e., the more that can be explained the better the explanation is.

We want to highlight three issues about the presented properties. Firstly, noticing that though the last properties are defined in the context of supervised learning, some of them share ideas with the ones presented from the works of humanity sciences, even with the previous work reviewed of characterization properties of explanations (Lacave and Díez (2002), Burkart and Huber (2021)). Furthermore, some of them can be seen as the particularization in the context of supervised ML of the properties that are defined in humanity sciences. For example, comprehensibility as a particularization of explanations being social or accuracy as a particularization of explanations being truthful. This is not odd as it can be seen as the perfect example of the lack of formalism. Bear in mind that the efforts are being made in order to evaluate in a formal and objective way something that is entirely subjective depending on humans and context. As so it is not strange to find similar underlying ideas even the same in different properties defined in different contexts and also find properties that make no sense in other contexts such as fidelity.

In this context of lack of formalism and intersection of concepts, the ideas expressed in Doshi-Velez and Kim (2017) are very useful. They state that the three levels of evaluation (human experts, human non-experts and proxies) have to inform each other and be in concordance. Hence, one should establish which proxies are important in which real-world task (functionally-grounded to application-grounded) or what factors (latent dimensions of interpretability) one has to take into account when characterizing a proxy (human-grounded to functionally-grounded).

Doshi-Velez and Kim (2017) defines interpretability as the "ability to explain or to present in understandable terms to a human". Then, the latent dimensions of interpretability (see Figure 1.3) are factors that are relevant to characterize this ability and, as a consequence, explanations. These latent dimensions represent the under-

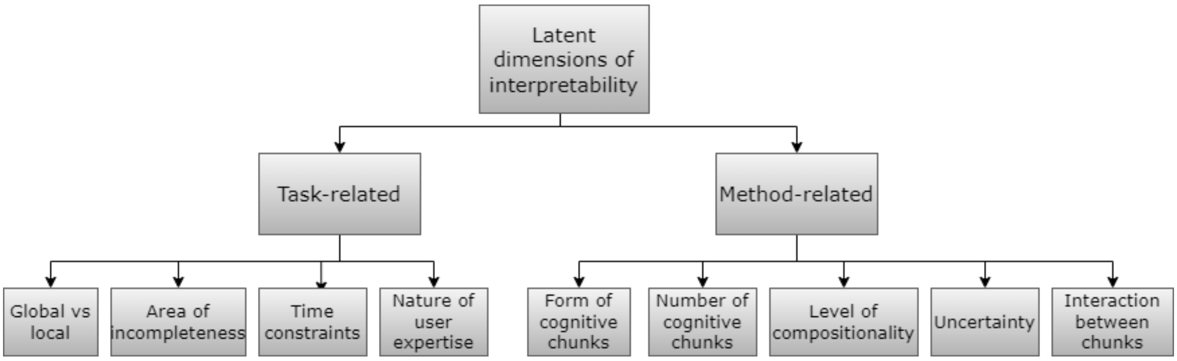


Figure 1.3: Latent dimensions of interpretability.

lying ideas of what should be taken into account to get a good explanation. Thus, we avoid definitions of particular properties for particular contexts. The authors establish some of them in a non-exhaustive list:

- **Task-related latent dimensions of interpretability**. Ideally, these factors should be isolated in human-grounded experiments to establish which methods work best when they are considered:
  - *Global vs. local.* This dimension is based on the idea that when seeking an explanation it should be considered whether the task at hand imposes a need of a global or more general explanation vs a local or more specific explanation related to the problem.
  - *Area of incompleteness.* This factor encapsulates the idea that within the task comes the incompleteness (Doshi-Velez and Kim (2017)) that generates the need of explanations. There are different causes for unquantified bias (incompleteness) that can provoke the necessity of an explanation.
  - *Time constraints.* This factor characterizes interpretability in terms of available time of the end-user to understand the explanation. It is related to the concept of complexity.
  - *Nature of user expertise.* This factor refers to the knowledge and communication skills that the end-user has.

- **Method-related latent dimensions of interpretability**. They define cognitive chunks to be the basic units of explanation.
  - *Form of cognitive chunks.* This factor states the idea that the form of the basic units of an explanation characterizes interpretability. We can find from raw features to symbolic representations as rules.
  - *Number of cognitive chunks.* This factor is again related to complexity, so the number of basic units of an explanation characterize interpretability. Authors highlight the relationship with the form of the cognitive chunk as some forms may contain more information than others, so this should be taken into account to define the number of units.
  - *Level of compositionality.* This factor expresses the importance of knowing if the cognitive chunks are structured and how in order to characterize

interpretability.

- *Uncertainty.* This factor characterizes interpretability with respect to the method in terms of including uncertainty measures and stochasticity. It should be taken into account how well people understand these terms.
- *Interactions between cognitive chunks.* This factor indicates that in interpretability it is important to know how the cognitive chunks are combined.

For the authors of Doshi-Velez and Kim (2017), the ideal world would be to establish and discover through human evaluation what factors are relevant to characterize interpretability within models and tasks. By doing this one would have a formal basis to evaluate the quality of explanations and construct and select proxies depending on the context and explainee. Notice the resemblance of some of the factors exposed to the properties discussed from the work Molnar (2020). This is not unusual due to the nature of these factors and shows that there is a certain degree of agreement in how to objectively evaluate quality of explanations.

Following with the issues of the properties presented from Molnar (2020), we notice that there was a contradiction between explanations focusing on the abnormal and explanations being general. Again, this shows the lack of formalism that exists within explanations as the properties formalize ideas about the quality of explanations but clash with each other. And again, the usage will depend on the context and explainee, but this is not the only thing it depends on. The other dependence is highly related to the third issue we are exposing next.

Notice that properties to evaluate the quality of explanations have been exposed, formalizing different ideas one could relate to that topic, but the thing is that nothing is said about how to measure these properties. This subject, which we call metrics of XAI, is one of the fields with more need of effort and investigation. It directly relates to how the explanations are obtained, which is the next topic we are going to discuss. As one could expect, the way an explanation is obtained influences how the properties and quality can be measured.

1.2.2 How do we obtain explanations?

There are two key concepts related to how explanations are obtained: interpretability and explainability. In the work of Barredo Arrieta et al. (2020) interpretability is defined as presented before (Doshi-Velez and Kim (2017)): "the ability to explain or to present in understandable terms to a human". Furthermore, they talk about explainability and the notion coming from Guidotti et al. (2018) stating "explainability is associated with the notion of explanation as an interface between humans and a decision maker which is, at the same time, both an accurate proxy of the decision maker and comprehensible to humans". After this, a discussion is made about the lack of formalism that surrounds these two concepts to try to end up defining explainability. Barredo Arrieta et al. (2020) defines, in the ML context, explainability as: "*Given a certain audience, explainability refers to the details and reasons a model gives to make its functioning clear or easy to understand*".

The concepts of interpretability and explainability, respectively, are generally related to the two forms of obtaining explanations: model interpretability and *post-hoc* explainability (Barredo Arrieta et al. (2020)) (see Figure 1.4). Model interpretability refers to the case where the model itself can be understood by humans, whereas *post-hoc* explainability relates to techniques that try to explain models that are not understandable by themselves. As so, explainability, which is a more general concept, is usually related to *post-hoc* explainability leaving model interpretability to be related to interpretability.

Furthermore, in the work of Barredo Arrieta et al. (2020), a division within model interpretability is presented. It is based on the domain in which the model is interpretable, distinguishing three levels:

- *Algorithmic transparency.* The underlying idea is that the process followed by a model to give an output has to be understandable. This is directly related to the predictions, but also to the training algorithm (Lipton (2018)).
- *Decomposability.* This level refers to models where each part of the model (input, parameters and calculation) can be explained. It encapsulates the previous level adding the constraint that every part of the model must be understandable.
- *Simulatability.* The underlying idea of this level is that the model can be simulated by a human, i.e., the model can be contemplated at once including everything: inputs, parameters and calculations. Lipton (2018) discerns two subtypes based on what renders the model not to be in this level: model size and computational time. This level encapsulates the previous ones.

Also a grouping of the different *post-hoc* techniques is presented. They state that *post-hoc* techniques are divided firstly by the intention of the author, then by the method used and lately by the type of data. However, their grouping does not exactly follows this division but rather presents a grouping with the goal of making it easy for researchers to look up for suitable techniques:

- *Text explanations.* The idea of these techniques is to learn how to generate text explanations of the model. This includes every type of symbolic representation of the functioning of the model.
- *Visual explanations.* The focus of these techniques is representing in a visual form what is trying to be explained from the model. It is a suitable group since it can be combined with other techniques.
- *Explanations by example.* These techniques are characterized for giving similar cases (data-examples) to the target given in order to understand how the model works.
- *Local explanations.* The underlying idea is to divide the solution space in subspaces and explain those ones, giving explanations that only hold for parts of the model's whole functioning.
- *Explanations by simplification.* In this case the idea is to generate new models that approximate the original model and are less complex, and explanations are extracted from those models. This is also known as surrogate models.

- *Feature relevance.* The idea here is to assign scores to the variables in order to determine their influence on the output, allowing the understanding of the model.

There is also another important classification of *post-hoc* techniques: model-agnostic and model-specific. The first one refers to those techniques that can be applied to any model. The second class comprises those that are defined for a specific model. In the work of Molnar (2020) one can find a review and explanation of the most known model-agnostic *post-hoc* techniques, and specific ones in the field of artificial neural networks.

The division of methods to obtain explanations and the established relation with the concepts of explainability and interpretability is rather a general and organizational theory than a formal one. This is not odd since this is directly related to explanations and the lack of formalism of explanations directly affects the concepts related to it. Next we present criticisms made of both ways of obtaining explanations derived from this lack of formalism.

On the one hand, critiques have been made about interpretability in Lipton (2018). The author states that the concept of interpretability is not formally defined; thus claims on model interpretability are wrong. In this work, the organizational view explained here is presented as model interpretability being model properties that comprise interpretations whereas *post-hoc* explainability are techniques that enable explanations in the form of "*What else can the model tell me?*".

From this point the argument made is that the claims made about linear models being more interpretable than neural networks are not based on a theoretical reason. This is due to the fact that for a similar performance, linear models usually need heavily engineered variables. This implies a trade-off between decomposability and algorithmic transparency when choosing between linear models or neural networks. As we obtain more complex engineered variables from the original ones to gain algorithmic transparency by using a linear model we lose decomposability since we are not usually able to understand these new features. On the other hand, neural networks do work with the raw features which can be directly understood. So, how is a linear model more interpretable if we lose decomposability?

Moreover, neural networks learn rich representations on raw features that can be represented and visualized so the interpretations made on this model with *post-hoc* techniques can prove useful too. So, despite the fact that in our heads we can conceive linear models being more interpretable, the truth is that there is no formality on the concept of interpretability that allows us to claim it. We highlight that the author concludes that claims on interpretability should be made based on a specific definition of interpretability. Also, claims on *post-hoc* techniques should fix an objective and demonstrate that the interpretations given allow to achieve it. These ideas will be important for the conclusions of this chapter.

On the other hand, we can find the work of Rudin (2019). Here the author states that

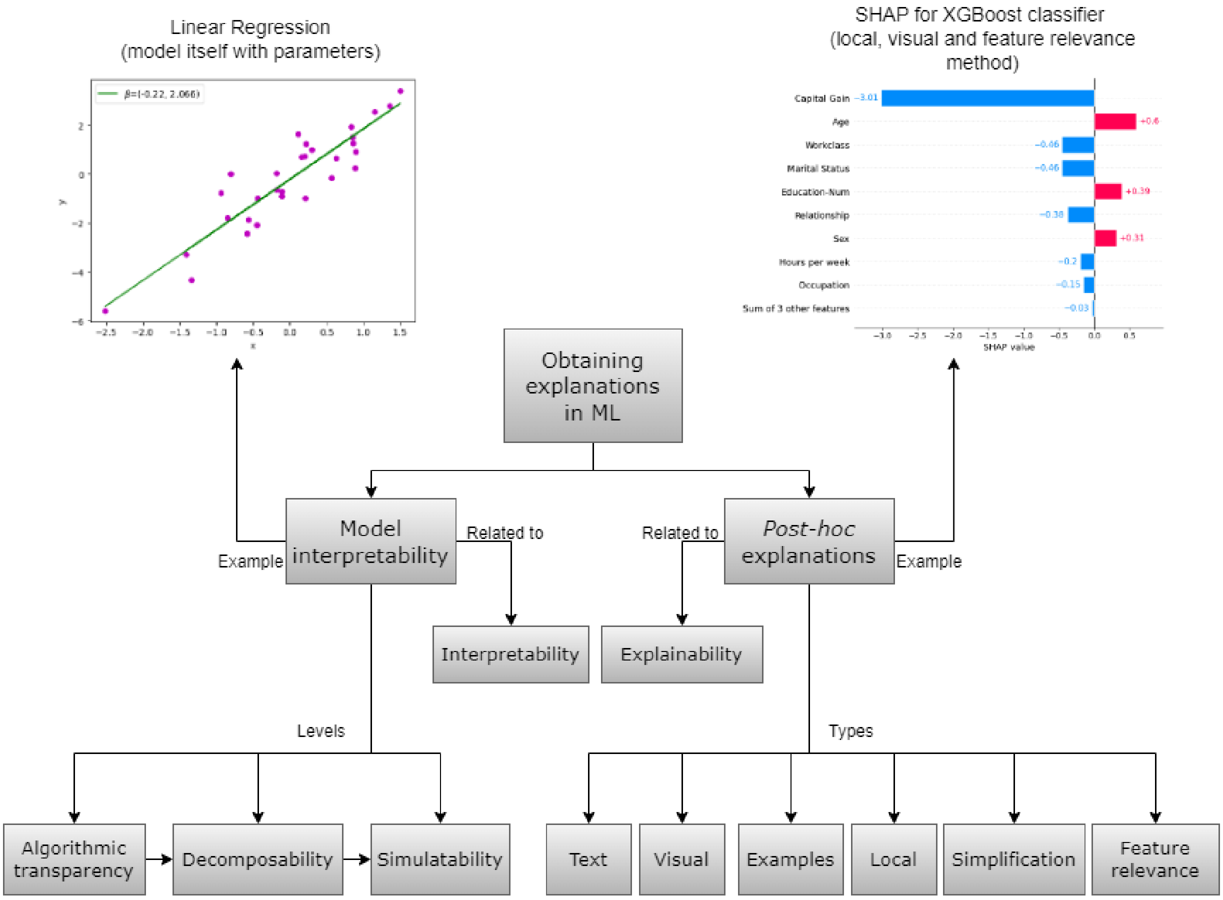


Figure 1.4: Obtaining explanations in the ML context.

explanations must come from interpretable models rather than from techniques that try to give an interpretation to black-box models (those which cannot be understood such as neural networks). The underlying idea of this critique is that interpretable models give explanations faithful to what they compute and do, whereas *post-hoc* explanations do not have this basis.

From this point, *post-hoc* explanations can be misleading and unreliable since they possibly do not present faithful explanations to what the black-box model is doing and learning. Other reasons centered around the black-box models are given and also a discussion on why it is that these explanations are used, but this is out of the scope of this work. Once all of this is presented, Rudin claims that interpretability is domain-specific and encourages to:

- Firstly, use interpretable models based on the needs of interpretability related to the problem.
- Secondly, expand the tool-box on interpretable models while also focusing on resolving some challenges that are already known such as constructing optimal logical models or constructing optimal sparse scoring systems. In Rudin et al. (2022) more of these existing challenges are presented and described more precisely.

Finally, based on the theory of knowledge representation and reasoning, we would like to propose an interpretation of the concepts of interpretability and explainability, which may be useful to understand the critique made by Rudin. In AI there are established three types of models: connectionist, symbolic and hybrid. Symbolic models use explicit symbols (mental models) to represent knowledge and syntactic rules to manipulate them, assuming that mental activity consist in the manipulation of those explicit symbols. In connectionist models we assume that knowledge is embedded into the connections and no symbols are used. This type of models exploits the stimulus-based learning and intelligent storage in form of connections or interactions among elements.

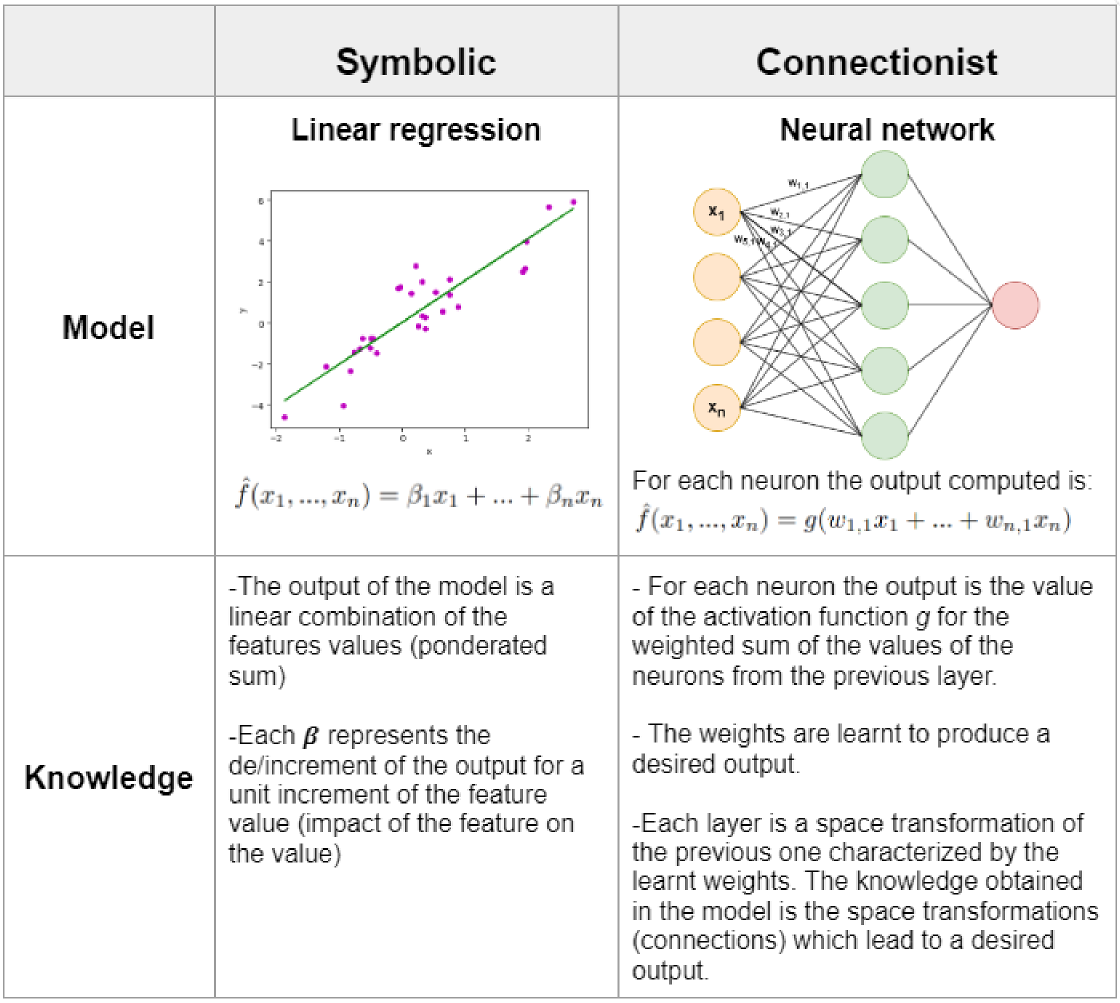
Furthermore, we present the concept of knowledge: a way to model in a structured manner experience obtained from a domain or that can emerge from the interpretation of information. Consequently, it allows interpretation of the information coming from the human senses, its representation, its storage and its organization.

Based on all of this, the theory of knowledge representation and reasoning studies the use of symbolic models (knowledge) and their manipulation (reasoning) to create knowledge systems that imitate humans in this matter. Up to this point it is neither new, nor odd the fact that this theory is related to XAI. We have got examples such as the use of rule-based expert systems or logic in argumentative XAI (Čyras et al. (2021)).

In the ML subfield of AI, the goal is to make computers learn and solve problems using data (information). Data is usually presented in the form of numerical variables although we have text, images and videos too. Here mathematical models are created in order to solve problems, hence they respond to the task they are thought, like for example classification. These models are conceived so that their inner workings (variables relationship, calculus, parameters, space of solutions, etc...) respond and allow them to resolve those tasks. Thus, what we are doing is embedding knowledge into the form of mathematical models. When given data about a problem that is needed to be solved, once the model is established we use learning algorithms that allow the model to learn the information contained in the data and combine it with the knowledge that it represents in order to solve the task.

Bearing all of this in mind the point of view here is that interpretable models are those which knowledge has been embedded into the mathematical model so that humans can manipulate it and reason about it. On the contrary, black-box models are those designed to learn from data and storage the knowledge in form of connections. So knowledge is not embedded into the design of the model, but rather the model is established to store the knowledge one can get from the data in form of connections. This makes it impossible to manipulate and reason about what is learned. Therefore, it is the *post-hoc* explainability that tries to interpret and express which knowledge embedded in the connections of the model. Figure 1.5 contains an example of each type of model.

From this perspective it is natural the critique made by Rudin since interpretable



	Symbolic	Connectionist
Model	Linear regression $\hat{f}(x_1, ..., x_n) = \beta_1 x_1 + ... + \beta_n x_n$	Neural network For each neuron the output computed is: $\hat{f}(x_1, ..., x_n) = g(w_{1,1}x_1 + ... + w_{n,1}x_n)$
Knowledge	-The output of the model is a linear combination of the features values (ponderated sum) -Each β represents the de/increment of the output for a unit increment of the feature value (impact of the feature on the value)	- For each neuron the output is the value of the activation function *g* for the weighted sum of the values of the neurons from the previous layer. - The weights are learnt to produce a desired output. -Each layer is a space transformation of the previous one characterized by the learnt weights. The knowledge obtained in the model is the space transformations (connections) which lead to a desired output.

Figure 1.5: Symbolic vs connectionist model example.

models have been designed and embedded with knowledge allowing interpretation and manipulation, whereas *post-hoc* explainability are techniques that try to interpret and explain knowledge embedded into connections.

So, what renders better explanations? It is true that interpretable models present a direct interpretation and explanation, and that with the idea of knowledge embedding more flexibility is presented to address interpretability for a specific context and explainee. Nevertheless, disregarding *post-hoc* explanations can be an error since those techniques relate input-output to interpret the knowledge embedded into the connections and this is a form of inductivism, one of the ways humans reason.

The point here is that there is not a formal ground theory that allows us to answer that question. Furthermore, it is not even clear what is a good explanation and, even if this is established, there is still the gap on how to obtain those good explanations in AI, which brings us the last subject: XAI metrics. Leaving the lack of formalism aside, it is clear that in order to evaluate the quality there is a need for measures.

1.2.3 XAI metrics

In the review of Burkart and Huber (2021) we can find a general summary on metrics for assessing explainability. The authors discuss the main lines of work taken in this field and implicitly expose the baseline problem of not having a ground objective theory. It is explained that the properties presented by Miller (2019), that we have also presented in this work, such as explanations being contrastive, social or selected, are usually difficult to measure.

Doshi-Velez and Kim (2017) introduces the idea of proxy to present the works done within interpretable models. A proxy is a formal, particular assumption on what is a good explanation. Particularly, in the context of interpretable models, it is a formal, particular definition of intepretability. Consequently, a model considered interpretable in some way is considered a proxy. As stated before, their idea is to study what proxies render good explanations in what context through human evaluation. Then, the interpretations that can be made from the model learned from the data create good explanations. One usual approach here is to consider as a proxy the combination of interpretable models with sparsity and complexity measures. We embed the sparsity and complexity measures defined over the parameters of a particular interpretable model within the optimization problem of the learning algorithm in order to establish a proxy (see Rudin (2019) challenges #1 and #2).

Complexity is tied to model comprehensibility (Guidotti et al. (2018)), a property we have presented here, and usually is approximated by measures related to model size. The concept of sparsity is related to density (antonym) and in this context makes reference to the model "not being dense" in order to generate explanations. For example in rule-based methods sparsity is measured (not the only way) in terms of the total number of rules in a decision set (Lakkaraju et al. (2016)).

More measures for the proxies of rule-based models and decision trees are given in Burkart and Huber (2021). Also, in this paper, works on measures for quality of explanations in recommendation systems are presented, all based on the idea that quality comes with the agreement of user's neighborhood on the recommendations. Finally, since there has been an explosion of works in the field of XAI that is not covered by Burkart and Huber (2021) we think that it is appropriate to mention some of them.

Hoffman et al. (2018) present some specific methods used to measure quality of explanations that respond to different subjects such as the goodness of the explanation, whether users are satisfied by explanations or how well users understand the AI systems. Hsiao et al. (2021) establish, through the help of the psychology field (human evaluation), subjective and objective measures within similar dimensions as in Hoffman et al. (2018), such as the user understanding or the goodness of the explanation.

Rosenfeld (2021) shifts the paradigm of evaluation metrics to one that considers the explanation itself and the appropriateness given the XAI goal and gives some measures according to this. This is due to the fact that users studies suffer from confirmation bias and methods explaining black-boxes do not follow agent's logical processes. Coroama and Groza (2022) present a review on evaluation metrics, giving a classification and a mapping between tools for explanation generation and those theoretically defined metrics. Biessmann and Refiano (2021) make a critique on formal automated metrics for evaluating *post-hoc* explainability within the field of computer vision. The key of this work is the comparison of those metrics with evaluations made with humans in the loop. Silva et al. (2023) present an attempt to evaluate objectively and subjectively different methods and metrics by means of humans.

Singh et al. (2022) work only in the subfield of *post-hoc* counterfactual explanations trying to analyze which methods apply and which allow comparison between methods. Akula and Zhu (2022) present a critique on saliency maps based on a human study of evaluation of this technique. Saliency maps is a *post-hoc* technique used in computer vision which gives back which parts of an image are the ones with more importance to the correspondent output. Li et al. (2022) present a taxonomy on evaluation metrics only for graph neural networks explainability methods.

Muñoz et al. (2023) present novel quantitative metrics which try to measure the interpretability factors based on global and local feature importance attributions, the variability of feature impact on the model output, and the complexity of feature interactions within model decisions. Belaid et al. (2022) try to create a benchmark for evaluating the quality of algorithms to generate explanations recovering some functional non-redundant tests from the literature. Palacio et al. (2021) try to establish a theoretical framework that provides concrete definitions for the terms "explanation" and "interpretation", and all steps necessary to produce explanations and interpretations, that is to say a unified framework to evaluate quality of explanations. Finally, Barkouki et al. (2023) and Barnard et al. (2022) are particular use-cases in XAI which entail particular metrics on evaluating the explanations within the use-case.

So far the work of Burkart and Huber (2021) and the little review presented here perfectly represent the variety and disparity of works in the subfield of XAI metrics. Furthermore, they faithfully show the non-existence of a common framework as we find works that aim to achieve that, works that only focus on particular techniques and critiques on different methods. We highlight the works done with humans in the loop that try to identify and evaluate methods as this follows the line of work proposed by Doshi-Velez and Kim (2017).

1.3 Conclusions

Now we have taken a peek at the big picture of XAI trying to understand both the quality and properties of explanations, in particular in the context of AI and ML, how those explanations are obtained, and the relationship between these two subjects. We have seen that the subject of explanations and their quality is something subjective. This is due to the fact that we do not entirely understand how the human brain works mixed with the fact that quality or adequacy of explanations depends on the context and end user. As a consequence, there is a lack of formalism and no ground general theory in the matter of explanations and their evaluation. Hence, the variety of works and ideas presented which sometimes are connected and sometimes are in contradiction.

Nonetheless, the connections between works shed light on having a formal and objective theory since they share same underlying ideas. In this landscape, the ideas of Doshi-Velez and Kim (2017) and Lipton (2018) bring a way to deal with the situation.

Ideally, the quality of explanations should be evaluated by humans. Since this is overly complicated, works should present how explanations are obtained, which factors of interpretability are being taken into account, properties of the explanation and how they are evaluated, task or problem and target audience, in short the proxy used to evaluate the quality of explanations. As works present their characteristics and human evaluation is done, we can discover factors of interpretability and proxies that work better and at the end, if possible, establish a ground formal theory.

Although human evaluation is not as easy as it sounds, for example, we have the confirmation bias (expected results for the individuals based on their own knowledge of the field) which can be a real problem within the unsupervised paradigm. Until now we have talked about XAI and explanations in ML, but particularly all the works presented are based on the supervised paradigm, being the explanations of predictions the main goal. In the case of unsupervised learning approach we are even more conditioned since there is no prior knowledge of the objective/results of the learning in contrast to supervised learning, where for example in classification, labels of the objects are given.

Under this paradigm the needs of explanations is even greater for this reason, which also makes the generation and quality evaluation of explanations more complex. Nonetheless, there is not much literature on explanations that regards this learning paradigm, though it is true that several of the ideas introduced in this section may apply in this case.

Chapter 2

Bayesian networks and explanations

2.1 Bayesian networks

In this section we present the Bayesian network model (Pearl (1988), Koller and Friedman (2009)) along with its inherent own tools and methods that allow us to generate explanations. The methods and ideas exposed will be based on the works of Lacave and Díez (2002) and Derks and De Waal (2020). Notice that we are not going to immerse ourselves into the subject of evaluating the quality of explanations since there is no canonical way to approach it, see Section 1.2.3.

A BN (see Definition 2.1.2) is a type of probabilistic graphical model which represents in a compact way a probability function $P(X_1, ..., X_p)$ taking advantage of the conditional independencies of the variables (see Definition 2.1.1) (Koller and Friedman (2009)). Thus, this model lives in the intersection of the fields of statistics, ML and AI. Moreover, its usage is quite useful in problems that present uncertainty, which are most life problems.

Definition 2.1.1 *Given a set of random variables* $\mathbf{Z}$ *and two random variables* A, B *then* A *is conditional independent of* B *given* $\mathbf{Z} \iff$

$$P(X, Y \mid \mathbf{Z}) = P(X \mid \mathbf{Z})P(Y \mid \mathbf{Z})$$

Definition 2.1.2 *Given a set of random variables* $\mathbf{X} = \{X_1, ..., X_p\}$ *following a distribution* $P(X_1, ..., X_p)$*, a BN is a tuple* $\mathcal{B} = (G, \theta)$ *where* $G = (\mathbf{V}, A)$ *is a directed acyclic graph (DAG), with* $\mathbf{V} = \{X_1, ..., X_p\}$ *and* $A \subset \mathbf{V} \times \mathbf{V}$ *a set of arcs.* $\theta = \{P(X_i \mid \mathbf{Pa}(X_i)) \mid i = 1, ..., p\}$ *is a set of conditional probability distributions (CPDs) of each random variable* X_i *conditioned to its parents* $\mathbf{Pa}(X_i)$ *in* G*.*

Each node of the graph represents each random variable of the system. The arcs denote probabilistic dependencies between children and parents. The graph encodes a set of conditional independencies between triplets of variables denoted $\mathcal{I}_l(\mathcal{G})$. This results in a factorization of $P(X_1, ..., X_p)$:

$$P(X_1, ..., X_p) = \prod_{i=1}^{p} P(X_i \mid X_1, ..., X_{i-1}) = \prod_{i=1}^{p} P(X_i \mid \mathbf{Pa}(X_i))$$

There exist three ways in order to model a CPD: parametric, non-parametric and hybrid (see Figure 2.1). In the parametric case, $P(X_i \mid \mathbf{Pa}(X_i))$ distributions are assumed to follow a parametric distribution such as a normal or a discrete probability distribution. On the contrary, in the non-parametric case no particular distribution is assumed. Finally, in the hybrid case both parametric and non-parametric approaches are mixed.

In the parametric approach, when all random variables are categorical we talk about discrete BNs. In this case the CPD is represented in a table (CPT) where we have a categorical distribution learnt for the conditioned variable for each possible combination of the values of the parents. When variables are continuous in the parametric modelling they are usually assumed to be Gaussian and we talk about Gaussian BNs. In this case $P(X_1, ..., X_p)$ is a multivariant Gaussian and dependencies are modelled with linear Gaussian CPDs. For each node of the network we have $P(X_i \mid \mathbf{Pa}(X_i)) \sim \mathcal{N}(\beta_0 + \sum_j \beta_j Pa(X_i)_j; \sigma)$. The parameters are obtained from the normal multivariate distribution properties. Given $\{\mathbf{X}, Y\}$ a set of features following a joint normal distribution:

$$P(\mathbf{X}, Y) = \mathcal{N}\left(\begin{pmatrix} \mu_{\mathbf{X}} \\ \mu_Y \end{pmatrix}; \begin{pmatrix} \Sigma_{\mathbf{XX}} & \Sigma_{Y,\mathbf{X}} \\ \Sigma_{\mathbf{X},Y} & \Sigma_{YY} \end{pmatrix}\right)$$

we have that $P(Y \mid \mathbf{X}) \sim \mathcal{N}(\beta_0 + \boldsymbol{\beta}^t\mathbf{X}; \sigma)$, with (Koller and Friedman (2009)):

$$\begin{aligned} \beta_0 &= \mu_Y - \Sigma_{Y,\mathbf{X}}\Sigma_{\mathbf{XX}}^{-1}\mu_{\mathbf{X}} \\ \boldsymbol{\beta} &= \Sigma_{\mathbf{XX}}^{-1}\Sigma_{Y,\mathbf{X}} \\ \sigma^2 &= \Sigma_{YY} - \Sigma_{Y,\mathbf{X}}\Sigma_{\mathbf{XX}}^{-1}\Sigma_{\mathbf{X},Y} \end{aligned}$$

One of the main advantages of assuming normally distributed data is that all conditioned distributions of features with respect to others are normal distributions. This also happens with the exponential family, where normal distribution belongs to. Tweedie Bayesian networks (TDB) (Masmoudi and Masmoudi (2019)) are BNs where CPDs are restricted to Gaussian, inverse-Gaussian, Gamma and Poisson distributions which belong to the exponential distribution family.

If mixed random variables (discrete and continuous) are present, we find conditional linear Gaussian BNs (CLGBNs), where discrete nodes cannot have continuous parents. Discrete variables are modelled as in the discrete BNs. Continuous variables are modelled with linear Gaussian CPDs, the main difference is that we learn as many linear Gaussian CPDs from the continuous parents as possible values take the discrete parents of the interest variable.

Moreover, we can find posterior models with mixed variables based on mixture of truncated exponentials (MTE) (Moral et al. (2001)), mixture of polynomials (MoP) (Shenoy and West (2011)) and mixture of truncated basis functions (MoTBFs) (Langseth et al. (2012)). It is also worth mentioning the augmented CLGBN, where discrete features can have continuous parents by the use of a softmax CPD.

One of the advantages of modeling in a parametric way is the existence of maximum likelihood estimators (unbiased and of minimum variance) of the parameters. Another advantage is the possibility of carrying out exact and approximate inference. The disadvantage of the parametric approach is that it is too restricted and when CPDs do not follow the assumed distribution large errors occur and the model loses efficiency. We highlight that all of this refers to continuous variables. Discrete variables are always modelled in a parametric way assuming a categorical probability distribution.

To avoid the problems of parametric modelling the non-parametric case shows up. Here no particular distributions for CPDs are assumed; instead the modelling is done by means of Gaussian processes, infinite mixtures and kernel density estimator (KDE). The idea of the non-parametric approach is to use particular instances of data $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ to approximate the original distribution instead of the parameters, where any distribution function may be fitted. We highlight the works of Atienza et al. (2022b) and Atienza et al. (2022c). These works consolidate the usage of KDE to build this type of BNs.

As the last paragraph implies, data is present in this model. In the case of the parametric approach, data is used to estimate the parameters, hence the ability to learn from particular data. In the non-parametric case, data is directly used to estimate the CPD, again learning from data. Furthermore, structure (graph representing conditional independencies) can also be learnt from data. There exist three types of algorithms for doing this: the ones based on testing conditional independencies, the ones based on search with maximization of a score such as the likelihood of the data and the ones that use both approaches (hybrid). More about the structure learning will be introduced in Chapter 4. We highlight that in both cases optimal solutions are hard to find and this should be taken into account when interpreting the model.

From the perspective of the theory of knowledge and reasoning, BNs can be seen as a model which represents knowledge in terms of probabilistic conditional independencies, allowing reasoning under uncertainty. This can be seen in the graph which represents the dependencies between variables and is translated into the mathematical model through the CPDs. Due to this fact, BNs are considered an interpretable model. Based on the knowledge embedded into the model, we can reason about it and obtain explanations. In the book of Darwiche (2009) we find a comparison between BNs and the classical knowledge-based approach to reasoning, but with the knowledge base being the BN and the reasoning engine based on the laws of probability.

Lacave and Díez (2002) and Derks and De Waal (2020) study the model of BNs and their relationship with explanations. To do so they distinguish three different fields

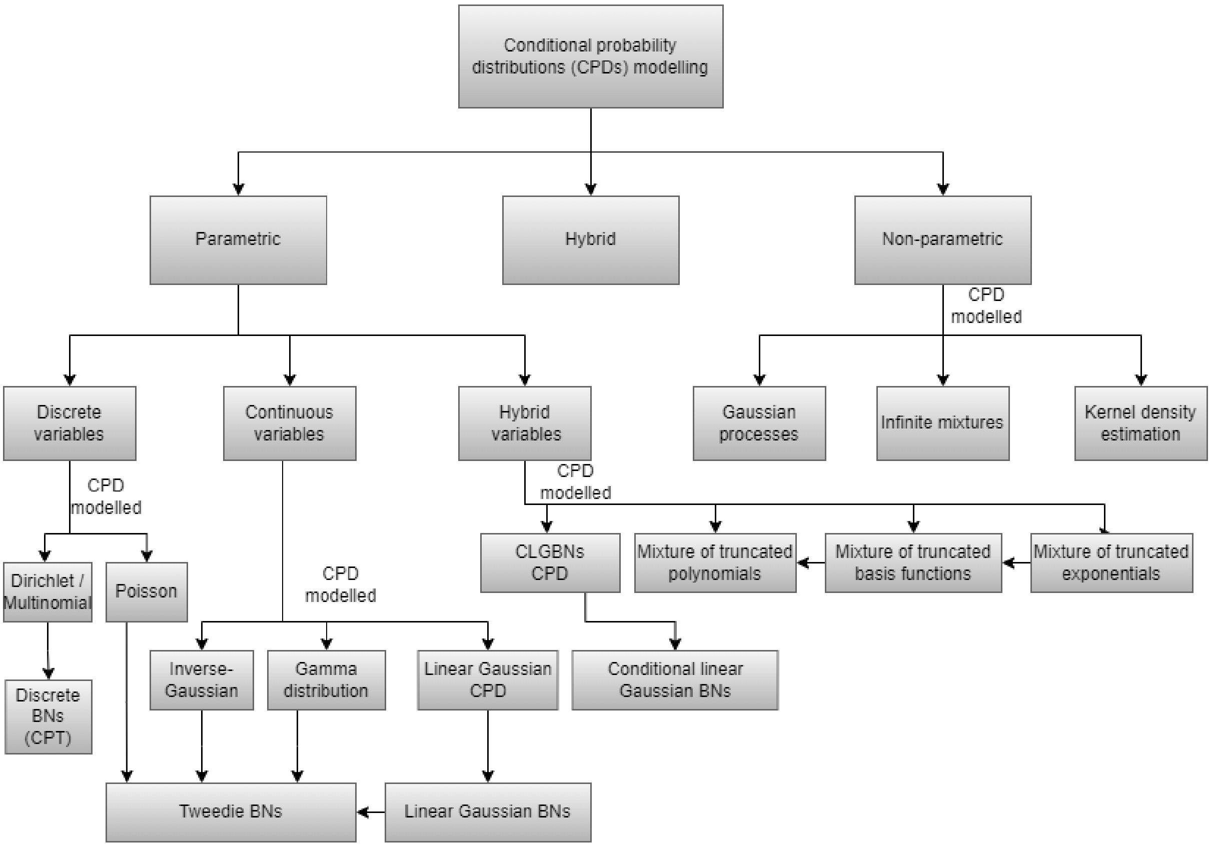


Figure 2.1: CPDs possible modelling in discrete and continuous BNs.

based on the focus of the explanation: model, evidence and reasoning (see Figure 2.2). Notice that in the figure there is also a node of decision since BN play a key role in decision systems, however this is out of the scope for this work.

Talking about the knowledge embedded into the design of the model in terms of conditional probabilistic relationships and their mathematical expressions (CPD) is what they considered the knowledge base. Interpretations can be given about this knowledge base, or in other words, model explanations (see Figure 2.3). This can be used to gain domain knowledge and assist application experts in the model-construction phase (Derks and De Waal (2020)).

The conditional dependencies and independencies have a direct interpretation coming from their definition (see Definition 2.1.2). This definition establishes that two variables are conditional independent given a set of variables $\mathbf{Z}$ when they are independent given an instantiation of $\mathbf{Z}$. In relation with model explanations, these must be interpreted just as it is, unless causality is introduced. If causality is introduced in the model we have that the encoded relationships of the graph G imply cause-effect relationships. In this case the graph still encodes probabilistic dependencies. The BNs which work with causality are known as causal BNs.

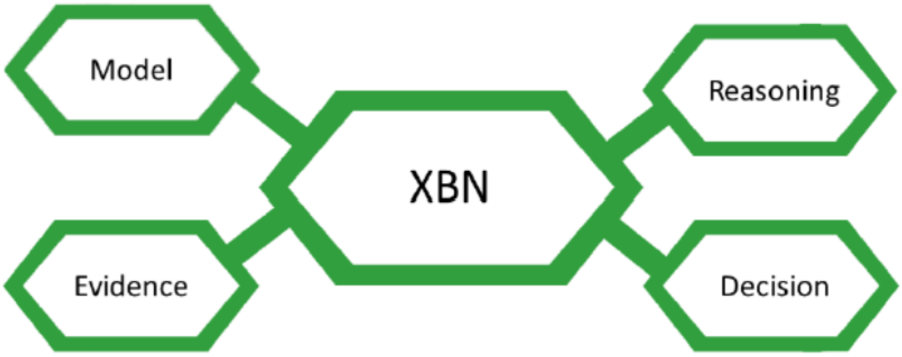


Figure 2.2: Image from Derks and De Waal (2020).

Since different solutions can be found in structure learning and optimal ones are hard to find, the identified conditional independencies may not represent reality. This does not affect interpretability at all, since error is always present in this and all other models, which is also a motivation for the need of explanations. The most important aspect here is to know what knowledge is embedded and how it is obtained and reasoned with. This also applies to the tools that will be presented. Notice that this directly relates to the Rashomon effect (see Section 1.2.1).

It is also important how the knowledge embedded into the model and the explanations extracted from this are presented to the user. In the work of Lacave and Díez (2002) papers are presented related to the various presentation forms: graphical display of the network, verbal description of the network and menu-driven navigation.

Menu-driven navigation makes reference to those works that allow navigation through the BN presenting knowledge about it, e.g., about the CPD of the nodes. Similarly does Elvira (Lacave et al. (2007)), which uses the CPD to add information to the conditional dependencies like positive or negative influence (posterior probabilities monotonically increase or decrease with respect to the conditioning variables). Notice that in the work of Lacave and Díez (2002), and some others, causal networks are considered. Since this is out of the scope we restrict this work to non-causal BNs. Thus, nothing is mentioned here about the particular methods and explanations that use the causality.

The reasoning that can be made with the knowledge embedded into the model does not end here, as we can see with the other two levels considered by the works used in this discussion (evidence and reasoning). On the one hand, evidence explanation makes reference to trying to explain particular data instances. On the other hand, reasoning explanation is tied to justifying the results obtained by the model and the reasoning process involved, the results not obtained by the model and hypothetical reasoning. All the tools that are used in these two levels fall under the field of inference. We have already seen the ones that can be used in explaining the model. Next we will review in Section 2.2 all the tools BNs offer that are useful for creating evidence and reasoning explanations.

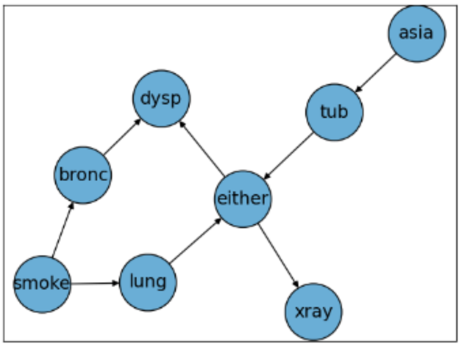
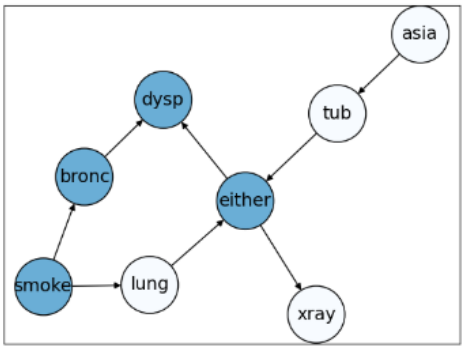
<table>
<tr><th>Visual</th><th>Verbal</th></tr>
<tr><td> </td><td>The probabilities of having bronchitis (‘bronc’) are directly influenced by the probability of being a smoker (‘smoke’)</td></tr>
<tr><td><table>
<tr><th></th><th colspan="2">bronc</th></tr>
<tr><th>smoke</th><th>no</th><th>yes</th></tr>
<tr><td>no</td><td>0.702</td><td>0.298</td></tr>
<tr><td>yes</td><td>0.391</td><td>0.609</td></tr>
</table></td><td>The probability of having bronchitis is higher if you are a smoker</td></tr>
<tr><td> </td><td>‘bronc’ is independent of the features ‘tub’, ‘asia’ , ‘xray’ and ‘lung’ if we know ‘dysp’, ‘smoke’ and ‘either’
(Markov blanket)</td></tr>
</table>

Figure 2.3: Model explanation examples.

2.2 Tools for creating explanations

To present the tools we are following the works of Lacave and Díez (2002) and Derks and De Waal (2020) differentiating between those related to evidence explanations and reasoning explanations. We first start reviewing those related to reasoning explanation since it covers the understanding of the basis of the knowledge embedded into the BN and how to manipulate and use it in order to later generate evidence explanations and more. Reasoning explanations are also themselves quite useful in particular contexts, as explaining the reasoning that can be made with the model directly entails the understanding of the task at hand. This is due to the fact that we are explaining the processes of the model to obtain outputs.

With respect to the taks's latent dimensions of interpretability (see Section 1.2.1 and Figure 1.3) the tools presented in this type of explanation create global explanations. Regarding the area of incompleteness (needs of explanations) of the task, since we are explaining different tools, each one with its own definition introduces what explanations it can give. With respect to time constraints, the user's understanding may vary depending on the method used, as for example a directly expressed relationship in natural language does not take the same amount of time as a graphic display of probabilities. The time that takes to produce the explanations may also vary from one method to another.Finally, user expertise has its minimum in probability knowledge since the reasoning made in BNs is all based on this discipline; time for understanding explanations is influenced by the skills of the user in this matter.

We now start reviewing Derks and De Waal (2020). In this case the authors only focus on explaining the type of reasoning that can be made based on the structure of the network and the probabilistic (in)dependencies between variables. Hence, here we do not find any method/tool that can be used to generate explanations about the reasoning. As expressed in this work, using the conditional (in)dependencies of the BN, which directly translate into the graph that represents the model, three types of reasoning can be made (see Figure 2.4):

- *Predictive reasoning.* This type of reasoning in BNs comprises going from parents to children, i.e., following the direction of the links of the DAG. The structure of the BN in terms of dependencies allows studying the influence of a known feature on its children, i.e., $A \to B$ permits a direct reasoning of the influence of A on B and this is what is called predictive reasoning.
- *Diagnostic reasoning.* This type of reasoning in BNs is made when given a particular feature (child), we reason and extract information about a feature that directly influences it (parent). In this case, the structure is used contrary to predictive reasoning: given $A \to B$ now we are extracting information about A knowing B, which passes through the calculation of $P(A \mid B)$. This is usually used when we want to extract knowledge from the BN on what went wrong.
- *Intercausal reasoning.* In this case we are reasoning and extracting knowledge and information of the influence of knowing a variable A on a variable B when both have influence on a third variable C. This type of reasoning appears in what we call a $v - structure$ ($A \to C \leftarrow B$).

There is another type of reasoning in BNs that is not covered in this work which is bidirectional reasoning. This type of reasoning is performed when two or more types of the ones presented are combined. So this is an explanation on the reasoning that allows the model, which is the basis for every other method. Although as said before, no method is presented which can be used to generate explanations about the reasoning.

If we want to explore methods within the explanation of reasoning, we have to review Lacave and Díez (2002). Particularly, the authors present a review on works divided by macro-level or micro-level explanations. The first group makes reference to explaining the reasoning of the model in a general way. Generally the idea is to explain

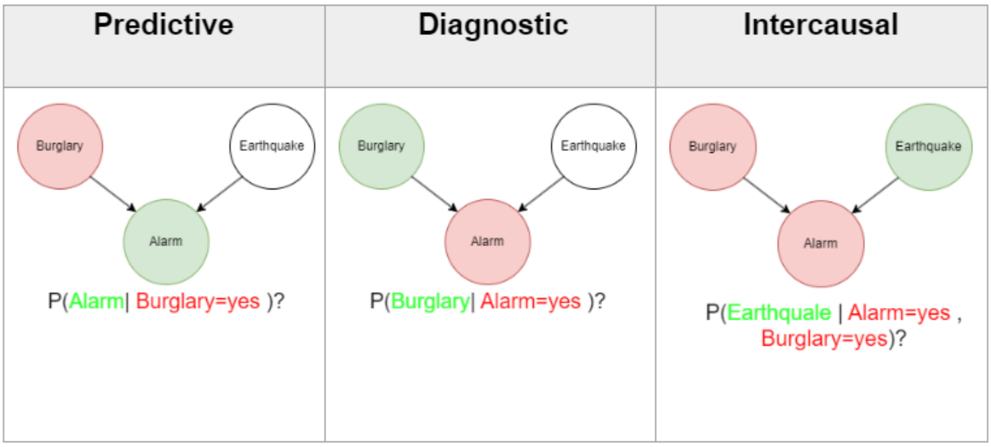


Figure 2.4: Example of three types of reasoning in BNs.

the paths of the BN which determine how the evidence flows and by doing this we characterize how the model reasons. The second one focuses on how the model reasons about a particular variable.

Micro-level:

- *Verbal description of variations in probabilities.* Here the idea is to present in natural language how the probabilities of a particular variable vary when other variables are observed. By doing this we represent how the model reasons about this variable, hence, it can be used as an explanation of the reasoning. The authors present the work of Elsaesser and Henrion (1990) as a representation of the idea of verbal description of variations in probabilities. For example, if p_1 changes to p_2 and $\frac{p_2}{p_1} \geq 5$ the authors use the expression: "a great deal more likely".

- *Graphical display of probabilities.* Again we find the idea of presenting graphically variations of probabilities of particular variables when others are observed. As a consequence, the explanation that can be obtained by this method is a plot such as bars, see Figure 2.5.

- *Analysis of the impact of evidence on a variable.* The idea here is to measure how evidence influences a variable, hence the measure captures the probabilistic relationships and gives information about them. As we are measuring how an evidence impacts a variable, the information obtained can be seen as an explanation of reasoning. Lacave and Díez (2002) present two works. The first is PATHFINDER (Heckerman (2019)), where the idea is to discriminate between particular evidences $\mathbf{d}_1, \mathbf{d}_2$ by their influence on a target variable X_i, given other evidence $\mathbf{e}$. Notice that this is developed for discriminating groups of diseases $(\mathbf{d}_1, \mathbf{d}_2)$. For doing so the evidence weight proposed by Good (1977) is used:

$$log(\frac{P(x_j \mid \mathbf{d}_1, \mathbf{e})}{P(x_i \mid \mathbf{d}_2, \mathbf{e})}) \quad x_i \in \Omega(X_i)$$

  where $\Omega(X_i)$ is the space where X_i lives. For each possible value x_i of X_i this

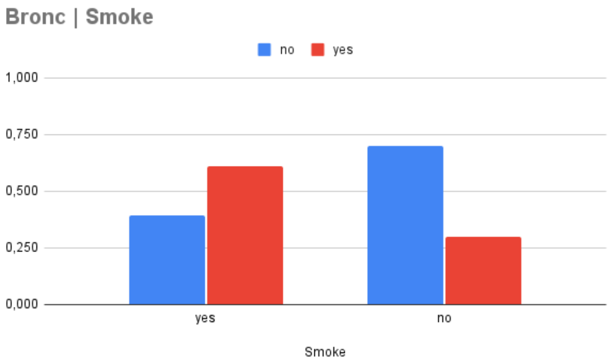


Figure 2.5: Example of graphical display of probabilities. Here we can see how the model reasons about the feature bronc as the probability of having bronchitis (red bar) increases when being a smoker.

measure is graphically displayed representing the proportionality on how the two different groups of variables influences X_i. The second work is Suermondt's INSITE (Suermondt (1992), Suermondt and Cooper (1993)) where given an evidence e, its influence on a variable X_i is measured through the cross-entropy H:

$$H(P(X_i \mid \mathbf{e}); P(X_i)) = \sum_i P(x_i \mid \mathbf{e}) log \left(\frac{P(x_i \mid \mathbf{e})}{P(x_i)} \right)$$

Macro-level:

- *Quantitative analysis of reasoning chains.* Lacave and Díez (2002) here present different works that appear under the same underlying idea: finding chains in the graph that relate an evidence to an interest variable and quantitatively assessing their importance. By doing this, it is shown how the reasoning in a BN is made at a macro-level since it is taking into account how the evidence propagates through all the net and affects the final objective variable.

  In Suermondt's INSITE we find an example of this method as they look for paths from a given evidence e to a target variable X_i, which are computationally related. Moreover, they analyse the importance of the chains in obtaining the inference result. This importance is obtained by removing arcs of the chains and studying the cross-entropy of the posterior distributions of the target variable given the evidence with and without the selected arc.

- *Probability of evidence.* The idea presented here is to use sensitivity analysis to present how changes in evidence affect the output. Generally any form of sensitivity analysis sheds light on the reasoning of the model. For example the study of how the probability function of the target variable, given an evidence, varies by using measures of distances between probability functions.

- *Qualitative explanations.* Although the works presented under these ideas are thought for causal BNs, the underlying idea can be passed on to non-causal

BNs with the reminder that influences should not be taken as cause-effect relationships. So the key here is to transform the BN into a qualitative BN which represents influences of adjacent nodes with postive, negative or unknown signs (Wellman (1990a), Wellman (1990b)).

After this, effects of evidences on interest variables can be studied through sign propagation (Druzdzel and Henrion (1993)). See Figure 2.6 for an example. This is similar to the quantitative study of chains of reasoning but transforming the information into qualitative concepts. Again, as we are representing how the evidence flows and affects the results of the target variables in some manner what we are representing is how the BN reasons, hence explanations can be obtained.

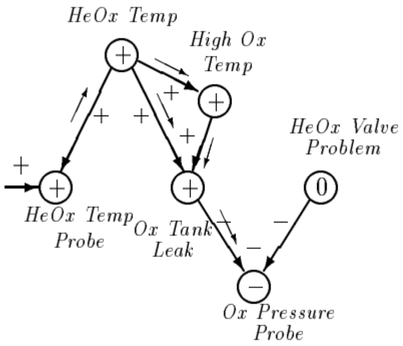


Figure 2.6: Example of sign propagation in a BN from Druzdzel and Henrion (1993).

- *Scenario-based explanations.* Methods under this flag try to establish scenarios to show how the model reasons. A scenario is an assignment of values to variables that are relevant to a certain conclusion, ordered in such a way that they form a coherent story. So given an evidence e if a target variable is set, then the idea is to show how variables are relevant to the target by building scenarios.

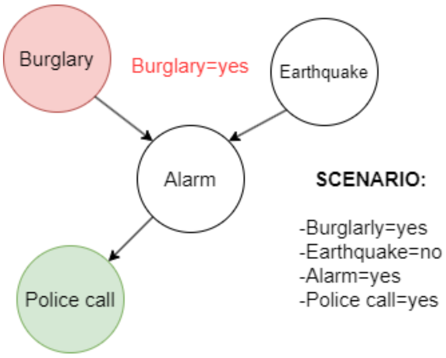


Figure 2.7: Example of a scenario in a BN. Given the evidence $Burglary = yes$ and the target $Police\ call$ this scenario shows us a plausible story that expresses how the model reasons about the target.

This is similar to studying the chains of the net, but by presenting ordered instances that show how the model is reasoning in a "sequential" way. In this case the explanation is usually conformed by the most probable scenarios obtained by partial abduction (we will define this concept later). Thus, the cognitive chunks are instances.

We continue now reviewing those explanations related to evidence explanations. Notice that there will be similarities with reasoning explanation as this is the basis for evidence explanation. The tools presented in this category make use of the knowledge embedded into BNs and the probabilistic reasoning that can be made to obtain knowledge about a given evidence, hence explanations can be built.

As we are dealing with particular instances, one of the task's latent dimensions of interpretability is local, i.e., we are reasoning about particular or specific points, not general patterns. In terms of time constraints, these methods can be used to produce explanations that usually are quick to understand. In the matter of time for the production this relates to the method established for inference and the complexity of the network. As for user expertise it should be noted that probability knowledge is a must since reasoning is made on this basis. Again, in terms of area of incompleteness of the task, each tool, with its own definition, introduces on what it can give an explanation. Finally, cognitive chunks in possible explanations are variables and instances since we are working with them.

Derks and De Waal (2020) distinguish two methods:

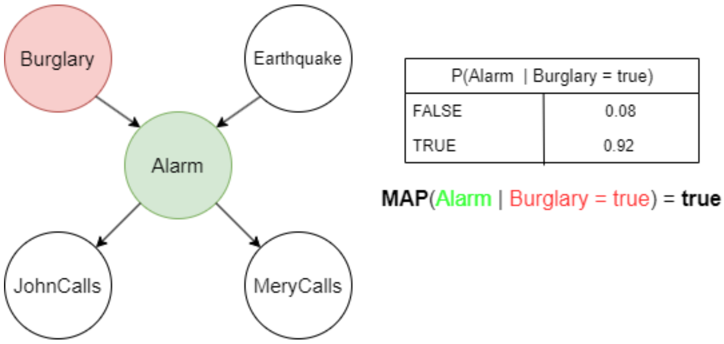
Maximum a posteriori (MAP). Given the evidence $\mathbf{e}$, the joint posterior distribution of a set of target variables $\mathbf{T}$ that are unobserved is used to determine the target variable's instance which is the most probable. This is also called partial abduction. A special case is when the set is constituted by all the unobserved variables, it is called most probable explanation (MPE). We can also find the k-most likely configurations where the first one is the maximum a posteriori (MAP):

$$\mathbf{t}^* = MAP(\mathbf{T} \mid \mathbf{e}) = argmax_{\mathbf{t} \in \mathbf{T}} P(\mathbf{t} \mid \mathbf{e})$$

The cognitive chunks then are instances of variables and the interpretation of this method is "$\mathbf{t}^*$ *is the most probable instance given* $\mathbf{e}$". With this we can answer what-if questions (hypothetical reasoning), we can make predictions, or we can directly explain an evidence based on the most probable configuration that happens when it is given, if this makes sense in the context. See Figure 2.8 for an example.

Most relevant explanation (MRE) (Yuan et al. (2011)). The basis of this method is the fact that in MPE and MAP the set of target variables is user-specified and can contain irrelevant variables with respect to the evidence. This leads to over-specified or under-specified explanations if we use them for such purpose. Yuan et al. (2011) show different pruning methods in order to deal with this problem. They are divided into pre-pruning and post-pruning.

Pre-pruning methods identify which variables are relevant with respect to the evidence by using the conditional independencies encoded in the BN. Post-pruning



P(Alarm | Burglary = true)	
FALSE	0.08
TRUE	0.92

MAP(Alarm | Burglary = true) = **true**

Figure 2.8: Example of MAP for feature Alarm given evidence "Burglary = true" in the Earthquake network from Scutari (2010).

methods first calculate MAP/MPE and then prune irrelevant variables. For doing so different approaches exist such as again using conditional independencies, greedily creating explanation trees with the most relevant variables or using measures for the explanatory power of the explanation, such as likelihood, in order to determine which target variables are relevant.

All of this does not mean that MAP/MPE cannot be used to produce explanations, but rather that MAP/MPE does not provide information on the relevancy of variables, so the explanations provided do not include this. Therefore what we are doing here can be seen as the introduction of a constraint of relevance, which adds another interpretation and hence other explanations. This falls in the direction of creating concise/selected explanations, but does not mean that one is better than the other, as again this is context-user dependent. The key is knowing the difference and the meaning of each method.

Nonetheless, the methods for pruning presented are criticized by Yuan et al. (2011). For them an explanation should be precise (high explanatory power) and concise (only contain the relevant variables). To account for this they translate the definition of explanation into an instantiation of *any* subset of the unobserved variables and provide a method that works under this assumption. By doing so, they provide the basis for developing an explanation method with its own the freedom to choose the target variables to include.

After this, the idea is to define a relevancy measure based on probability knowledge. This measure allows obtaining instantiations of the unobserved variable with regard to the evidence that, when used as an explanation accounts for precision and conciseness. Under this framework, the usage of maximum posterior probability as in MAP/MPE is not suitable since it does not account for conciseness. After a discussion of different possible relevance measures, the conclusion is the use of the Bayes factor (Jeffreys (1998)) in a general form, which they define as follows:

Definition 2.2.1 *The generalized Bayes factor (GBF) of an instance* x *for a given evi-*

dence $\mathbf{e}$ *is defined as:*

$$GBF(\mathbf{x};\mathbf{e}) = \frac{P(\mathbf{e} \mid \mathbf{x})}{P(\mathbf{e} \mid \bar{\mathbf{x}})} = \frac{P(\mathbf{x} \mid \mathbf{e})(1 - P(\mathbf{x}))}{P(\mathbf{x})(1 - P(\mathbf{x} \mid \mathbf{e}))}$$

where $\bar{\mathbf{x}}$ *denotes the set of all alternative hypotheses of* $\mathbf{x}$ *(all the potential alternative values of* $\mathbf{X}$ *which are not* $\mathbf{x}$*).*

Moreover GBF can be rewritten as:

$$GBF(\mathbf{x};\mathbf{e}) = \frac{\frac{P(\mathbf{x}|\mathbf{e})}{P(\bar{\mathbf{x}}|\mathbf{e})}}{\frac{P(\mathbf{x})}{P(\bar{\mathbf{x}})}} = \frac{\frac{P(\mathbf{x}|\mathbf{e})}{P(\mathbf{x})}}{\frac{P(\bar{\mathbf{x}}|\mathbf{e})}{P(\bar{\mathbf{x}})}}$$

Hence, it has a direct interpretation as the ratio of the posterior odds ratio given $\mathbf{e}$ and the prior odds ratio. It can also be interpreted as the ratio between the belief update ratios of $\mathbf{x}$ and the alternative hypothesis $\bar{\mathbf{x}}$. Taking the baseline of 1, a higher value of GBF means that the ratio that encapsulates the influence of $\mathbf{e}$ on the hypothesis $\mathbf{x}$ is bigger than the one on the alternative hypothesis $\bar{\mathbf{x}}$.

The GBF fulfills a variety of properties presented by the authors that make it suitable with respect to the objective of finding a relevant instance given an evidence that is both precise and concise. The first one relates to the consistency of the interpretation of the values of GBF with the decision-theoretic interpretation of extreme probability values. An example would be the GBF value of ∞ when $P(\mathbf{x}) < 1$ and $P(\mathbf{x} \mid \mathbf{e}) = 1$. This takes up on the idea that the evidence eliminates all uncertainty of the explanation/hypothesis $\mathbf{x}$. Hence, it is clear the value of the explanation with respect to the evidence and it is represented with the infinity value of the GBF.

We also find the monotonicity with regard to the difference between posterior and prior probability $P(\mathbf{x} \mid \mathbf{e}) - P(\mathbf{x})$. GBF increases as this difference increases, which indicates that GBF captures the influence of $\mathbf{e}$ on $\mathbf{x}$. Moreover, in extreme values of $P(\mathbf{x})$ (near 1 or 0), GBF is even bigger with respect to the same fixed difference of prior and posterior. This is quite interesting since it means that GBF captures the belief that the difference of prior and posterior in extreme values of $P(\mathbf{x})$ is more significant because of the theoretic interpretation of those values. Monotonocity appears also with respect to the belief update ratio $r(\mathbf{x};\mathbf{e}) = \frac{P(\mathbf{x}|\mathbf{e})}{P(\mathbf{x})}$. This is captured in the next theorem (Yuan et al. (2011)).

Theorem 2.2.1 *For an explanation* $\mathbf{x}$ *with a fixed belief update ratio* $r(\mathbf{x};\mathbf{e})$ *greater than* 1, $GBF(\mathbf{x};\mathbf{e})$ *is monotonically increasing as the prior probability* $P(\mathbf{x})$ *increases.*

If $r(\mathbf{x};\mathbf{e})$ is greater than 1, this means that $\mathbf{x}$ is more likely given $\mathbf{e}$, and given a big $P(\mathbf{x})$ means that $\mathbf{x}$ is likely to happen. Then if $\mathbf{e}$ happens, and $\mathbf{x}$ is even more likely this indicates that $\mathbf{x}$ has a high explanatory power given $\mathbf{e}$ (in the sense that it is likely to see that $\mathbf{x}$ happens after $\mathbf{e}$ happens). This should be captured by relevance measures which try to find and instantiation explanation given an evidence, and this happens with GBF. But the key properties are those which allow GBF to measure the relative importance of multiple variables and select the relevant ones, which is perfect to achieve the property of conciseness. First the CBF is defined (Yuan et al.

(2011)):

Definition 2.2.2 *The conditional Bayes factor (CBF) of explanation* y *for given evidence* e *conditioned on explanation* x *is defined as:*

$$CBF(\mathrm{y};\mathrm{e}\mid\mathrm{x}) = \frac{P(\mathrm{e}\mid\mathrm{y},\mathrm{x})}{P(\mathrm{e}\mid\bar{\mathrm{y}},\mathrm{x})}$$

With this definition not only do we have a factor decomposition of the GBF, which is great for incremental evidence, but we have the next theorem (Yuan et al. (2011)).

Theorem 2.2.2 *Let the conditional Bayes factor (*CBF*) of explanation* y *given explanation* x *be less than or equal to the inverse of the belief update ratio of the alternative explanations* $\bar{\mathrm{x}}$*, i.e.,* $CBF(\mathrm{y};\mathrm{e}\mid\mathrm{x}) \leq \frac{1}{r(\bar{\mathrm{x}};\mathrm{e})}$*, then we have:*

$$GBF(\mathrm{x},\mathrm{y};\mathrm{e}) \leq GBF(\mathrm{x};\mathrm{e})$$

As explained in Yuan et al. (2011), this provides a soft measure of the relevance of y with respect to e given x. On the contrary, the conditional independence of $\mathbf{Y}$ and $\mathbf{E}$ given $\mathbf{X}$ ($\mathbf{Y} \perp \mathbf{E} \mid \mathbf{X}$) provides a hard measure with answer yes or no to the relevance of variable/s $\mathbf{Y}$, thus the relevancy of y. This theorem implies that GBF encodes a decision boundary, which is the inverse of the belief update ratio of the alternative explanations $\bar{\mathrm{x}}$ given e. This ratio sets up the point of decision of the importance of the rest of variables, given x, for them to be included or not in the explanation. The idea is that if $CBF(\mathrm{y};\mathrm{e}\mid\mathrm{x}) \leq \frac{1}{r(\bar{\mathrm{x}};\mathrm{e})}$ then $GBF(\mathrm{x},\mathrm{y};\mathrm{e}) \leq GBF(\mathrm{x};\mathrm{e})$, therefore y should not be included in the explanation. Finally, Yuan et al. (2011) establishes the next corollaries:

Corollaries

- Let x be an explanation with $r(\mathrm{x};\mathrm{e}) > 1$, and $\mathbf{Y} \perp (\mathbf{X},\mathbf{E})$, then for any state y of $\mathbf{Y}$, we have $GBF(\mathrm{x},\mathrm{y};\mathrm{e}) < GBF(\mathrm{x};\mathrm{e})$.
- Let x be an explanation with $r(\mathrm{x};\mathrm{e}) > 1$, and $\mathbf{Y} \perp \mathbf{E} \mid \mathbf{X}$, then for any state y of $\mathbf{Y}$, we have $GBF(\mathrm{x},\mathrm{y};\mathrm{e}) < GBF(\mathrm{x};\mathrm{e})$.
- Let x be an explanation with $r(\mathrm{x};\mathrm{e}) > 1$, and y be a state of a variable/s $\mathbf{Y}$ such that $P(\mathrm{y}\mid\mathrm{x},\mathrm{e}) < P(\mathrm{y}\mid\mathrm{x})$, then we have $GBF(\mathrm{x},\mathrm{y};\mathrm{e}) < GBF(\mathrm{x};\mathrm{e})$.

The power of GBF as a soft measure for relevance on instantiation y with respect to e given x and the properties of consistency with respect the theoretical interpretation of probabilities make it a perfect measure for finding instantiations that are related to a given evidence in probabilistic terms (precision) and comprise the most relevant instances of variables (conciseness). Consequently, the MRE is defined as follows:

Definition 2.2.3 *Let* $\mathbf{T}$ *be a set of target variables, and* e *be the evidence of the remaining variables in a Bayesian network* $\mathcal{B}$*. MRE is the problem of finding an instantiation/-explanation* x *for* e *that has the maximum generalized Bayes factor score* $GBF(\mathrm{x};\mathrm{e})$*, i.e.,*

$$MRE(\mathbf{T};\mathrm{e}) = argmax_{\mathrm{x}\in\mathbf{X},\emptyset\subset\mathbf{X}\subseteq\mathbf{T}} GBF(\mathrm{x};\mathrm{e})$$

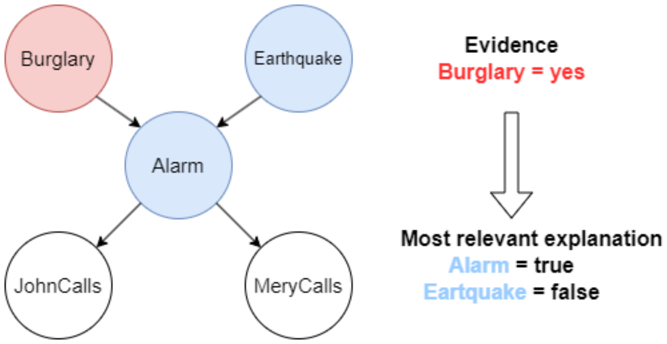


Figure 2.9: Example of MRE given evidence "Burglary = true" in the Earthquake network from Scutari (2010). Code can be found in **github**[2].

Finally, the cognitive chunks are instances of variables again, and in this case the interpretation of the method is: $\mathbf{x}$ is an instantiation of a subset of the target variables that maximizes the GBF. As a consequence of the properties of GBF we are capturing the minimal instantiation of variables that are relevant to the evidence. Again, its use with respect to explanations is creating explanations of an evidence by giving the most relevant and minimal instantiation of the target variables that relate to $\mathbf{e}$. We can also answer what-if questions (hypothetical reasoning). See Figure 2.9 for an example where, contrary to the example of Figure 2.8, the method explains the evidence, $Burglary = yes$, by giving back the instantiated target $(Alarm = yes, Earthquake = yes)$.

In Lacave and Díez (2002) similar material is introduced. In this case the authors only focus on MPE/MAP, i.e., total/partial abduction, and they present works on how to obtain MPE/MAP. In two of them, we can identify the preamble and basis for MRE. But these are not the only methods related to explaining evidence; we broaden these works by presenting the next works and methods.

Most relevance evidence (Meekes et al. (2015)). This method's purpose is to find relevant sets where computational efforts should focus. Nonetheless, the information and knowledge that it provides can be used as an explanation. Given a set of target variables $\mathbf{T}$ and an evidence set of evidence variables $\mathbf{E}$, Meekes et al. (2015) define the given-evidence sensitivity set for $\mathbf{T}$ given $\mathbf{E}$, the potential-evidence sensitivity set for $\mathbf{T}$ given $\mathbf{E}$, the evidence sensitivity set for $\mathbf{T}$ given $\mathbf{E}$ and the irrelevant evidence set for $\mathbf{T}$ given $\mathbf{E}$.

For all definitions the concept of d-separation is used. Moreover, relationships between the sets are defined and how Bayes-Ball algorithm (Shachter (2013)) may be used to calculate them. We focus on the irrelevant evidence set for $\mathbf{T}$ given $\mathbf{E}$, which is defined as the subset of $\mathbf{E}$ for which each variable X is d-separated (conditional independent) of $\mathbf{T}$ given any instance of $\mathbf{E} \setminus \{X\}$, i.e.,

$$IrrEv(\mathbf{T}, \mathbf{E}) = \{X \in \mathbf{E} : X \perp \mathbf{T} \mid \mathbf{E} \setminus \{X\}\}$$

Notice that here the idea is reversed to the one in MRE. Hence, given an evidence set $\mathbf{E}$ and a set of target variables $\mathbf{T}$, the idea is to use the conditional independencies to

[2] https://github.com/Victor-Alejandre/Bayesian-Network-based-clustering.git

find which evidence influences the target, thus it is important/relevant. In this case cognitive chunks of the explanation, if the method is used, are variables. Then, the knowledge given by the method is the subset of variables of $\mathbf{E}$ whose instances are going to affect the probabilities of the target, hence these are the explanations that can be obtained. See Figure 2.10 for an example.

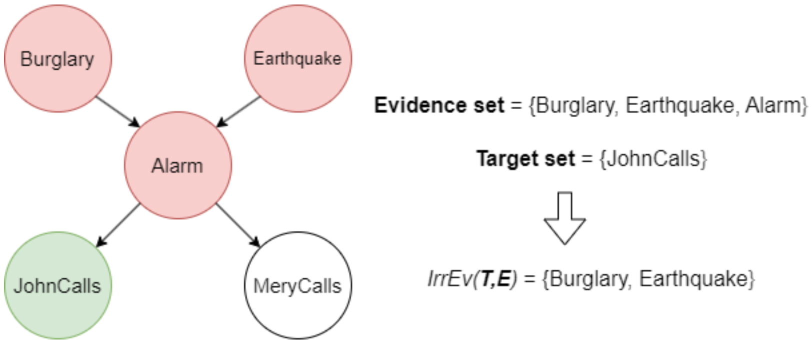

Burglary
Earthquake
Alarm
JohnCalls
MeryCalls
Evidence set = {Burglary, Earthquake, Alarm}
Target set = {JohnCalls}
*IrrEv(**T,E**)* = {Burglary, Earthquake}

Figure 2.10: Example of most relevance evidence in the Earthquake network from Scutari (2010).

MAP-independence (Kwisthout (2021)). This method again revolves around the concept of relevance in the context of MAP. The purpose is to try on clarifying the value of MAP by giving relevant variables that affect this value. In this case, it is argued that the model context relevance of variables to the MAP obtained through conditional dependencies is too strict. Consequently, the proposal here in order to find what information is important in a given MAP explanation/value is to find a set $\mathbf{I}^+$ of relevant variables and $\mathbf{I}^-$ of irrelevant variables in the calculation of MAP. Both sets shape the set of unobserved variables $\mathbf{I} = \mathbf{I}^+ \cup \mathbf{I}^-$ without counting the set of target variables $\mathbf{T}$ and the evidence $\mathbf{e}$, i.e., $\mathbf{I} = \mathbf{X} \setminus \{\mathbf{T} \cup \mathbf{E}\}$. Instead of using conditional dependencies, to define the set of irrelevant variables the idea is to determine which variables are irrelevant based on whether the output of MAP changes if the variables were to be observed. In order to find $\mathbf{I}^+$ the author poses and formalizes the following sub-problem: knowing $MAP = \mathbf{t}^*$ given an evidence $\mathbf{e}$ and $\mathbf{R} \subseteq \mathbf{I}$, is $\mathbf{t}^*$ MAP-independent from $\mathbf{R}$? The formalization is:

Given a BN $\mathcal{B} = (G, \theta)$, where $\mathbf{V}(G)$ is partitioned into a set of evidence nodes $\mathbf{E}$, with a joint value assignment $\mathbf{e} \in \mathbf{E}$, a non-empty explanation set $\mathbf{T}$, with a joint value assignment $\mathbf{t}^* \in \mathbf{T}$ (MAP), a non-empty set of nodes $\mathbf{R}$ for which we want to decide MAP-independence relative to $\mathbf{T}$, and a set of intermediate nodes $\mathbf{I}$ with $\mathbf{R} \subseteq \mathbf{I}$. In order to establish the sought MAP-independence, the following question must be answered: Is $\forall \mathbf{r} \in \Omega(\mathbf{R})$, $argmax_{\mathbf{t} \in \Omega(\mathbf{T})} P(\mathbf{t}, \mathbf{r} \mid \mathbf{e}) = \mathbf{t}^*$?

The author concludes that this problem is generally intractable, though if MAP calculation is tractable the problem may be tractable for a given set $\mathbf{R}$ not too big (Valero-Leal et al. (2023)). Hence, this method, given a set $\mathbf{R}$ of interest, solves the problem presented and determines whether $\mathbf{R}$ has influence on $\mathbf{t}^*$ given $\mathbf{e}$, if it were to be observed. Therefore the cognitive chunks of the explanation, if this method is used to generate one, are variables. Notice that here the explanation given combines with the explanations from MAP to enrich the understanding of the latter. It can also be used to answer what-if questions related to MAP.

Influence-driven (Albini et al. (2021)). This work focuses only on BN classifiers (single-labeled and multi-labeled). Particularly, the naive Bayes (single-labeled) and Bayesian network-based chain classifier (BCC) (Sucar et al. (2014)) with the leaves nodes being the observation features. Similar to Druzdzel and Henrion (1993) and Timmer et al. (2017) the idea is to find relevant variables for the classification of an instance given by means of the influences between variables. Consequently, the congnitive chunks of the explanation are variables and sets of influences.

A set of influences $\mathcal{I}$ is a set with pairs of variables, $\mathcal{I} = \{(X,Y) \in \mathbf{X} = \mathbf{O} \cup \mathbf{C}\}$, where different types of influences can be defined. The set $\mathbf{O}$ is the set of observable features, whereas the set $\mathbf{C}$ is the set of classification features. Both sets are disjoint and define the variables of the problem.

The influence types t_i are defined through a function $\pi : \mathcal{I} \times \mathcal{A} \to \{true, false\}$, where $\mathcal{A}$ is the set of all input assignments which contains all possible maps a. Each map a assigns a domain value for each variable that is not the target classification ($X \in \mathbf{O}$). Notice that the maps a are just instantiations of the observed variables but expressed as functions from the set of features to the feature space. For example, if we have a variable $Alarm$ with possible values $\{yes, no\}$ then $a(Alarm) = yes$ is a possible map.

As one can observe from their definition, the influences types are defined with regard to the inputs/evidence (maps a) we want to classify. Notice that given an influence type for the variables, the influence may exist only for some particular inputs (not all of them as it depends on a). Now an explanation kit is a set which, given the possible influences $\mathcal{I}$, contains pairs of the influence types t_i and the function that characterizes them π_i, $EK = \{\langle t_1, \pi_1\rangle, ..., \langle t_n, \pi_n\rangle\}$. Albini et al. (2021) define the method to obtain the relevant variables and influences which can be used for the explanation as:

Definition 2.2.4 *Given a set of influences $\mathcal{I}$ and an explanation kit $EK = \{\langle t_1, \pi_1\rangle, ..., \langle t_n, \pi_n\rangle\}$ for $\mathcal{I}$, an influence-driven explanation (IDX) drawn from EK for explanandum $C \in \mathbf{C}$ with input assignment map $a \in \mathcal{A}$, is a tuple $\langle \mathbf{X}_r, \mathcal{I}_{t_1}, ..., \mathcal{I}_{t_n}\rangle$ with:*

- $\mathbf{X}_r \subseteq \mathbf{X}$ *such that* $C \in \mathbf{X}_r$ *(we call* $\mathbf{X}_r$ *the set of relevant variables).*
- $\mathcal{I}_{t_1}, ..., \mathcal{I}_{t_n} \subseteq \mathcal{I} \cap (\mathbf{X}_r \times \mathbf{X}_r)$ *such that for any* $i \in \{1, ..., n\}, \forall (X,Y) \in I_{t_i}$, $\pi_i((X,Y), a) = true$.
- $\forall X \in \mathbf{X}_r$ *there is a sequence* $X_1, ..., X_k$, $k \geq 1$, *such that* $X_1 = X, X_k = C$, *and* $\forall 1 \leq i < k$, $(X_i, X_{i+1}) \in \mathcal{I}_{t_1} \cup ... \cup \mathcal{I}_{t_n}$.

As we can see, the method obtains the relevant variables for a specific classification variable given an instantiation (through mapping a). These variables are those for which a chain of influences reaching the explanation variable can be established, independently of the types of influences. This is represented with the elements of the tuple $(\mathcal{I}_{t_1}, ..., \mathcal{I}_{t_n})$. So this can be used to establish an explanation of the relationship of evidence and its classification. The explanation represents the important variables obtained based on the influences that they have. Thus, the influences defined and the influence types that can be defined are the key issue.

Monotonically Dialectal Influence Driven Explanations

$$EK = \{\langle t_1, \pi_1 \rangle, \langle t_2, \pi_2 \rangle\}$$

$$t_1 = - \ \ (monotonic\ attack)$$

$$t_2 = + \ \ (monotonic\ support)$$

For any $(X, Y) \in \mathcal{I}$ and input $a \in \mathcal{A}$

$$\pi_1((X,Y),a) = \pi_-((X,Y),a) = True \iff$$

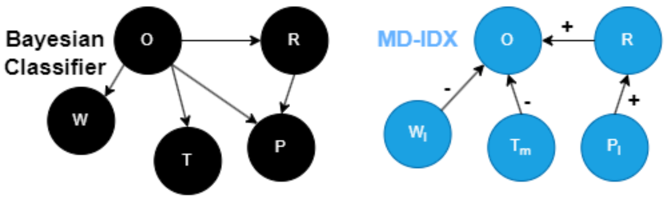
$$\forall x_i \in \Omega(X) \setminus \{MAP(X \mid a)\} \ \ P(MAP(Y \mid a) \mid a) < P(MAP(Y \mid a_{x_i}) \mid a_{x_i})$$

$$\pi_2((X,Y),a) = \pi_+((X,Y),a) = True \iff$$

$$\forall x_i \in \Omega(X) \setminus \{MAP(X \mid a)\} \ \ P(MAP(Y \mid a) \mid a) > P(MAP(Y \mid a_{x_i}) \mid a_{x_i})$$

- $a_{x_i}(Z) = a(Z) \ \ \forall Z \in \mathbf{X} \setminus \{X\}$
- $a_{x_i}(X) = x_i$

			MAP	
w	t	p	r	o
l	l	l	$+ (.98)$	$- (.99)$
m	l	l	$+ (.98)$	$- (.99)$
h	l	l	$+ (.98)$	$- (.99)$
l	l	h	$- (.79)$	$+ (.70)$
m	l	h	$- (.79)$	$- (.51)$
h	l	h	$- (.79)$	$- (.51)$
l	m	l	$+ (.98)$	$- (.95)$
m	m	l	$+ (.98)$	$- (.98)$
h	m	l	$+ (.79)$	$- (.91)$
l	m	h	$- (.79)$	$+ (.82)$
m	m	h	$- (.79)$	$+ (.82)$
h	m	h	$- (.99)$	$+ (.99)$
l	h	l	$+ (.99)$	$- (.99)$
m	h	l	$+ (.99)$	$- (.99)$
h	h	l	$+ (.94)$	$- (.99)$
l	h	h	$+ (.94)$	$- (.99)$
m	h	h	$+ (.94)$	$- (.99)$
h	h	h	$+ (.97)$	$- (.98)$



Features and feature spaces

$\mathbf{C} = \{PlayOutside\ (O),\ Raining\ (R)\}$

$\mathbf{O} = \{Wind\ (W),\ Temperature\ (T),\ Pressure\ (P)\}$

$\Omega(O) = \Omega(R) = \{-\ (no), +\ (yes)\}$

$\Omega(W) = \Omega(T) = \{low, medium, high\}$

$\Omega(P) = \{low, high\}$

$\mathcal{I} = \{(W,O), (T,O), (R,O), (T,R), (P,R)\}$

Example for a:

- $a(W) = low$
- $a(T) = medium$
- $a(P) = low$

Explanation *for* $O \in \mathbf{C}$ and a

- $\mathbf{X}_r = \{W, T, P, R, O\}$
- $\mathcal{I}_- = \{(W,O), (T,O)\}$
- $\mathcal{I}_+ = \{(R,O), (P,R)\}$

Figure 2.11: Example of influence driven explanation taken from Albini et al. (2021). The Bayesian network is the Play outside Bayesian classifier.

The authors here present three types of explanations based on the influence types defined: monotonically dialectical IDXs (see Figure 2.11), stochastically dialectical IDXs and attribution method-based dialectical IDXs. The three types of explanations have the same in common, only two types of influences types are defined which relate to the idea of attack and support (positive vs negative effects). The first two types are highly related to what is known as argumentative XAI (Čyras et al. (2021)), which is a discipline that studies particular frameworks that allow argumentation for explanations in some areas of AI. Here it is very important the property of dialectical monotonicity which is related to how humans think (monotonic or non-monotonic). The first two types fulfill this property. Also, the authors give relationships between these two types of explanations. Finally, the third type is introduced to show how this method also adapts to attribution methods (*post-hoc*).

See Figure 2.11 for an example with monotonically dialectical IDXs, where the defintion for the influence types is given in the upper part of the figure. Moreover, in the lower center and left part of the figure we can see the Bayesian classifier with the respective features and MAP classifications for the different combinations of the observable variables. On the lower right part of the figure it is shown in a graphical display of the network the different influences obtained for the input set of influences $\mathcal{I}$ and map a given below. Finally it is shown in the lower right corner the explanation obtained for those inputs.

Counterfactual (Koopman and Renooij (2021)). Counterfactual explanations answer the idea of contrastive (property) explanations (*Why this outcome instead of another one*?) (see Section 1.2.1). As expressed in Koopman and Renooij (2021) counterfactual explanations try to give a change in inputs which generates a change in the output. Hence, they answer the question by giving another instance that will generate another output. They explain that this idea has been translated into the field of ML in the form of different definitions. This is why we only present this work, which focuses on the definition within the field of BNs. However their ideas can be generalized.

The authors define the explanation given by their method as *persuasive contrastive explanations*. Their idea is to give answers to *Why t (MAP) instead of t'?* given an evidence $\mathbf{e}$. Their method obtains a counterfactual explanation based on the work of Wachter et al. (2017), i.e., an input assignment that generates t' instead of t. As they consider that this is not enough to answer the question *Why t instead of t'?*, they also provide an explanation based on giving the sufficient evidence that generates the output t (similar to most relevance evidence). Hence, their explanations also follow the idea of the property that explanations are selected. So their method obtains, given an evidence $\mathbf{e}$ and a target variable T, two instantiations: one that relates to the target as being the minimal evidence needed to obtain the MAP t and the other that relates to the target as a minimal instantiation that provokes a change in MAP to a value chosen t'. This is the definition presented in the work:

Definition 2.2.5 *Consider explanation context $\langle \mathbf{e}, t, t' \rangle$ where $\mathbf{e}$ is the evidence, t is the MAP of the target T given the evidence, and t' is another possible value of the target which we would like to obtain. A persuasive contrastive explanation is any pair of instances $(\mathbf{s}, \mathbf{c})$ where $\mathbf{s} \in \Omega(\mathbf{S}), \mathbf{c} \in \Omega(\mathbf{C})$, $\mathbf{S}, \mathbf{C} \subseteq \mathbf{E}$, and*

- $\mathbf{s} \subseteq \mathbf{e}$ *is a sufficient explanation for t, i.e., $MAP(T \mid (\mathbf{s}, \hat{\mathbf{e}})) = t$ for all $\hat{\mathbf{e}} \in \Omega(\hat{\mathbf{E}})$, with $\hat{\mathbf{E}} = \mathbf{E} \setminus \mathbf{S}$, and there is no $\mathbf{s}' \subset \mathbf{s}$ for which this property holds.*
- $\mathbf{c} \subseteq \mathbf{e}$ *is a counterfactual explanation for t' i.e., $MAP(T|(\bar{\mathbf{e}}, \mathbf{c})) = t'$ for $\bar{\mathbf{e}} \subseteq \mathbf{e}$, $\bar{\mathbf{e}} \in \Omega(\bar{\mathbf{E}})$, with $\bar{\mathbf{E}} = \mathbf{E} \setminus \mathbf{C}$, and there is no $\mathbf{c}' \subset \mathbf{c}$ for which this property holds.*

The authors also introduce a methodology to obtain these instantiations instead of using the brute-force approach. To sum up, the method proposed obtains instantiations; consequently the cognitive chunks of explanations obtained from these definitions are those. Moreover, the explanations obtained from this method can be seen as a combination of two explanations as the method proposes the combination of two evidences that follow different ideas but relate to each other. This is an example of a combination of methods/ideas that enrich explanations as they combine their knowledge and information. See Figure 2.12 for a particular example of the methodology applied to the Child network displayed on the left part of the figure. On the right part of the figure we find the evidence and target features that are instantiated for obtaining a particular explanation context. From this explanation context four persuasive contrastive explanations are obtained and shown in the form of a tuple that shows first the sufficient explanation s and second the contrastive explanation **c**.

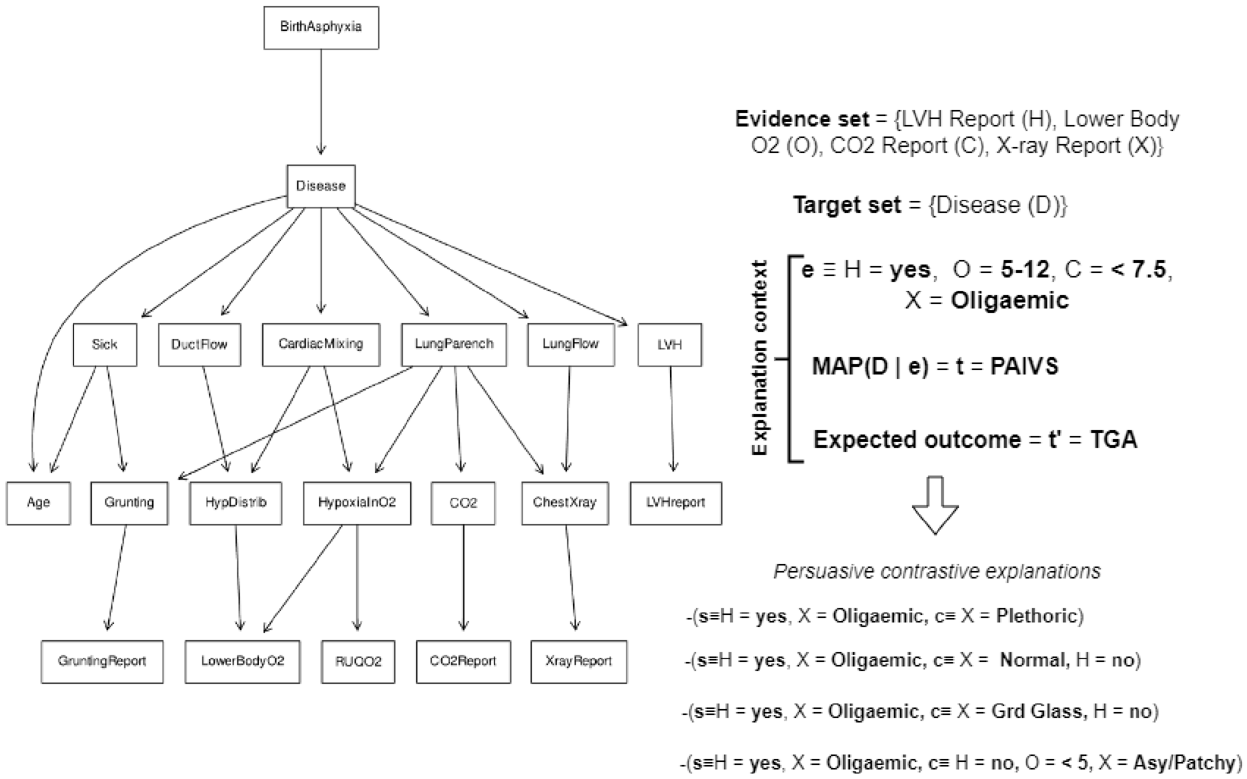

BirthAsphyxia
Disease
Sick
DuctFlow
CardiacMixing
LungParench
LungFlow
LVH
Age
Grunting
HypDistrib
HypoxiaInO2
CO2
ChestXray
LVHreport
GruntingReport
LowerBodyO2
RUQO2
CO2Report
XrayReport
Evidence set = {LVH Report (H), Lower Body O2 (O), CO2 Report (C), X-ray Report (X)}
Target set = {Disease (D)}
Explanation context
e ≡ H = yes, O = 5-12, C = < 7.5, X = Oligaemic
MAP(D | e) = t = PAIVS
Expected outcome = t' = TGA
Persuasive contrastive explanations
-(s≡H = yes, X = Oligaemic, c≡ X = Plethoric)
-(s≡H = yes, X = Oligaemic, c≡ X = Normal, H = no)
-(s≡H = yes, X = Oligaemic, c≡ X = Grd Glass, H = no)
-(s≡H = yes, X = Oligaemic, c≡ H = no, O = < 5, X = Asy/Patchy)

Figure 2.12: Example of persuasive contrastive explanations for the Child network (Scutari (2010)) taken from Koopman and Renooij (2021).

So we have presented a general view of the potential use of BNs to obtain explanations due to the knowledge embedded into the model. More ideas can be presented that make use of it, such as the use of estimators, simulation, graphical displays or even combinations of the methods presented as long as they represent useful information that can be considered an explanation. To close this section we would like to note that apart from all of this, the use of model-agnostic *post-hoc* methods can also be introduced in this model. Also in this matter, BNs present a great advantage to those methods which use the idea of perturbing variables to obtain explanations.

Perturbing variables consist of changing inputs and studying the change of outputs given by the model in order to elucidate how it works. Ideally, perturbation should be done using the probability distribution followed by the inputs ($P(\mathbf{x})$), since this captures the likelihood of how the variables take values in real life problems. BNs model and learn this function, thus the great advantage mentioned. Moreover, this instigates also, in some cases, the use of BNs as surrogate models as we present in the next section.

2.3 Solving the perturbation-based method's problem

Perturbation-based methods are those which use sampling of the feature space in order to obtain the results, like LIME (Ribeiro et al. (2016)) or anchors (Ribeiro et al. (2018)). Some model-agnostic methods like Shapley-values (Molnar (2020)) or partial dependence plots (Friedman (2001)) (there are more) make use of expectations of $\hat{f}(\mathbf{x})$ in order to give explanations about the inner working of the model. Notice that $\hat{f}(\mathbf{x})$ is the function learned by a specific model to solve a particular problem given a dataset D. Since computation of these expectations is hardly ever feasible or possible, the idea is to use estimators to approximate these values from samples (perturbation-based).

Most models used here do not work with the probability distribution function, and the usual procedure is sampling from the dataset, which is not wrong as long as the sampling is correctly made. However, the characteristics and definition of the BNs make this model ideal for its use as a surrogate model in these cases, not to give explanations but to sample and work with the probability distribution function. Nevertheless, it is worth mentioning that this would add computational effort and for some cases it would probably be infeasible, such as working with posterior probabilities in non-parametric BNs. In the following we exemplify this with Shapley-values and propose a variant that can be achieved through the use of BNs.

Shapley-values is a local model-agnostic method which comes from game theory (Shapley (1953)). In the original work the idea is to define a function φ that gives a share value of the game v (superadditive set-function) to each participant of the game. Shapley states three axioms that unequivocally define the game share function value φ. These axioms also wrap some of the natural properties expected from a function that gives the share of each player to the game: linearity/additivity, permutability and efficiency. The first one clears that for the sum of two games the share or value for each player should be the sum of the share for each game. The second one states that for any permutation of the players the share should be the same. Finally, the last one captures the idea that the sum of the values for all the players should be the total value of the game. The function φ obtained is defined as follows:

$$\varphi_i[v] = \sum_{\mathbf{S} \subseteq \mathbf{P}} \frac{(s-1)!(p-s)!}{p!}[v(\mathbf{S}) - v(\mathbf{S} - (i))]$$

where $\varphi_i[v]$ is the game share value for player $i \in \mathbf{P} = \{1, ..., p\}$ and game v. $\mathbf{S}$ are subsets of players from player set $\mathbf{P}$. Since all of this is defined in the context of game theory, efforts have been made in order to translate these ideas into the field of AI, particularly ML.

The idea in the ML context is to explain the contribution of each feature to the value $\hat{f}(\mathbf{x_e})$ given an instance $\mathbf{x_e}$. Therefore, the "players" here are the features. The key problem is to establish the analogue of the game value function v, since it is defined as a superadditive set-function in the context of game theory. Once defined v, the game share value function φ remains the same as defined before. For example notice that if we use the model function $\hat{f}(\mathbf{x})$ as v, then φ only fulfills the property of efficiency defined in the game theory framework if $\hat{f}(\mathrm{x})$ is linear. In the work of Sundararajan and Najmi (2020) we can find different definitions of the value function v in the context of ML (there are more such as intregated gradientes):

- **Conditional expectations Shapley (CES)**. The value function is defined in terms of the instance $\mathrm{x_e}$, model function $\hat{f}$ and probability distribution P followed by input variables:

  $$v(\mathbf{S}) = \mathbb{E}_{\mathrm{x}\sim P}[\hat{f}(\mathrm{x}) \mid \mathrm{x}_{\mathbf{S}} = (\mathrm{x_e})_{\mathbf{S}}]$$

  where $\mathrm{x}_{\mathbf{S}} = (\mathrm{x_e})_{\mathbf{S}}$ implies that the values of x for the subset of features $\mathbf{S} \subseteq \mathbf{X}$ are substituted by the values of $(\mathrm{x_e})$.

- **Baseline Shapley (BShap)**. Here the idea is to set a baseline point to measure contribution with respect to it. Hence, the value function is defined in terms of the instance $\mathrm{x_e}$, the model function $\hat{f}$ and a baseline point x':

  $$v(\mathbf{S}) = \hat{f}((\mathrm{x_e})_{\mathbf{S}}; (\mathrm{x}')_{\mathbf{X}\setminus\mathbf{S}})$$

- **Random baseline Shapley (RBShap)**. This idea is similar to BShap but with the use of marginal distributions. Thus the value function is defined in terms of the instance $\mathrm{x_e}$, model function $\hat{f}$, probability distribution P followed by input variables:

  $$v(\mathbf{S}) = \mathbb{E}_{\mathrm{x}'\sim P}[\hat{f}((\mathrm{x_e})_{\mathbf{S}}; (\mathrm{x}')_{\mathbf{X}\setminus\mathbf{S}})]$$

We state that defining the analogue of v in the ML context is the key problem since, as one would expect, the natural properties, which an attribution method should have, should appear in this context just like the axioms established in the work of Shapley. Since the game share value function is the same as in the game theory context, it is v the key to obtain a game share value function that fulfills the desired properties.

In this matter, Lundberg and Lee (2017) establish three properties/axioms similar to the ones defined in Shapley (1953) to characterize additive feature attributions methods (which include Shapley-values in the context of ML models). However, these properties are defined with respect to the equivalent of the game value function v in order to characterize unequivocally the game share value function φ in the ML context. Since we are working to measure the share of variables to a model function these properties of the game share value function should be established with regard to $\hat{f}$ instead of v. In Sundararajan and Najmi (2020) these axioms with respect to the model function are defined. Some of them correspond to axioms studied in the game context theory:

- **Dummy**. This axiom declares that when a feature is dummy it has zero attribution. Given a particular function f, a feature X_i is dummy if for any two values x_i, x_i' and every value x_{-i} of the other features then $f(x_i; \mathrm{x}_{-i}) = f(x_i'; \mathrm{x}_{-i})$. Thus, this property captures the idea of having zero attribution when having no effect on the function.

- **Efficiency**. This property is similar to the axiom of efficiency introduced in Shapley (1953), but in this case also relates to the introduction of a baseline. A value function is efficient if for every $\mathrm{x_e}$ and baseline x' the attributions add up to the difference $f(\mathrm{x_e}) - f(\mathrm{x}')$. This is natural since when establishing a baseline the value function should give the effect of the variable with respect to that baseline. Also, the fact that the sum of attribution values should take this value entails that we blame the features for the difference $f(\mathrm{x_e}) - f(\mathrm{x}')$.

- **Linearity**. This axiom states that the attributions of the linear combination of two given functions f_1, f_2 is the linear combination of the attributions of each function. This property encapsulates the idea that attributions are linear with respect to the function (efficiency), hence it is natural to ask that for sums of functions, attribution is the sum of the respective attributions for each function.

- **Symmetry**. This property states that for every function f that is symmetric in two variables X_i and X_j, i.e., for an instance $\mathrm{x_e}$ and baseline x' given with $x_i = x_j$ and $x'_i = x'_j$, then the attributions for features X_i and X_j should be the same.

- **Affine scale invariance (ASI)**. This property expresses that a game share value function φ is ASI if the attributions are invariant under simultaneous affine transformations of the function f and features. This is that for any given c, d, if $f_1(x_1, ..., x_p) = f_2(x_1, ..., (x_i - d)/c, ..., x_p)$ then for all the features the attributions should be the same given $(x_1, ..., x_p), f_1$ and the baseline point x', or given $(x_1, ..., cx_i + d, ..., x_p), f_2$ and the baseline point $(x'_1, ..., cx'_i + d, ..., x'_p)$

- **Demand monotonicity**. This property states that the game share value function φ should only increase when a feature X_i increases under the assumption that the function f is non-decreasing. A game share value function fulfills this property if this happens for any function f and any feature X_i given that satisfies the conditions.

- **Proportionality**. This axiom states that if the given function f can be rewritten as a function of the linear combination of the features $\sum X_i$, and the baseline x' is zero, then the attributions values are proportional to the values of the given instance $\mathrm{x_e}$.

Not only do we find these definitions in Sundararajan and Najmi (2020), but also a discussion about the different value functions v that they present and the fulfillment of these properties by the game share value function φ that these v define. It is shown that the CES φ does not fulfill the properties of symmetry, linearity, dummyness and demand monotonicity. It is also shown that BShap φ is the unique function which satisfies linearity, dummy, affine scale invariance (ASI), demand monotonicity (DM), and symmetry (plus minor technical conditions) for all attribution problems.

The problem of using BShap is that the results depend on the baseline point x' selection which is upon the user and problem context and may not be trivial. Also the share of the features is not the share of the feature to the function value $f(\mathrm{x_e})$ but rather the share to the difference $f(\mathrm{x_e}) - f(\mathrm{x}')$ (baseline point dependent). Notice that we seek the first one, not the second.

Furthermore, the game share value function φ defined by RBShap fulfills the properties of efficiency, symmetry, dummyness and linearity (Molnar (2020)). Also the use of the expectations with marginals renders the baseline as the expected value of f with respect to the inputs, which seems natural and is not baseline point dependent.

The problem with this methodology is that since we are using marginal expectations we may introduce unrealistic data instances in the calculations when features are not independent. This can cause misleading conclusions (Hooker et al. (2021)). Thus, we propose a variant of this method which takes the use of BNs to efficiently avoid this problem.

The idea is to use $\hat{f}(\mathbf{x_e})P(\mathbf{x_e})$ instead of $\hat{f}$ for a given instance $\mathbf{x_e}$ whose feature attribution is wanted. Hence, what we are doing is weighting the function value by the likelihood of the instance. This directly implies that, when calculating the value function, the marginal expectations take into account the likeliness of the instances and weight the function value with it, thus we avoid the problem of introducing unlikely data instances.

The downside of this idea is that we do not longer have the efficiency property with respect to the function of study. This directly translates into the interpretability of the attribution values: these ones represent the effect of each feature on the function value weighted by the likelihood of the instance and not only the function value, as one would like. Nonetheless, it is still interesting to know how the features contribute to the value function weighted by the likelihood of the instance and makes it perfect sense with also the benefit of avoiding the problem of unlikely data contributions. We implement these ideas in Algorithm 1.

```
Algorithm 1: Weighted Shapley-values
input  : $D, \hat{f}, \mathbf{x_e}$
Learn an adequate BN $\mathcal{B}$ from dataset $D$;
for $i = 1$ to $p$ do
    Sample from $\mathcal{B}$;
    Learn Shapley-values from the sample using
      $v(\mathbf{S}) = \mathbb{E}_{\mathbf{x}' \sim P}[\hat{f}((\mathbf{x_e})_{\mathbf{S}}; (\mathbf{x}')_{\mathbf{X} \setminus \mathbf{S}})P((\mathbf{x_e})_{\mathbf{S}}; (\mathbf{x}')_{\mathbf{X} \setminus \mathbf{S}})]$;
end
Result: Shapley-values computed
```

Finally, after this discussion one can see that Shapley-values present some problems and disadvantages which make them unreliable. Critiques about them, apart from the problems exposed here, are made in Kumar et al. (2020). Nonetheless, the purpose here was to show the advantage of introducing BNs. The ideas of using BNs for sampling and obtaining marginals and conditional probability functions can be extended to the other *post-hoc* methods mentioned here. As a matter of fact, they can be extended to any perturbation-based method.

Chapter 3

The clustering problem

3.1 Clustering

In this section we present the clustering problem, that falls into the unsupervised learning paradigm, and we review works on interpretability within the problem. The unsupervised learning paradigm wraps every ML problem involving unlabelled data. Therefore, in this context we do not have a target/objective given a particular input, but rather we simply have the inputs and the objective is to extract patterns from them. Hence, problems, methodologies and models under this paradigm have the ultimate goal of learning and extracting information from the data itself.

In unsupervised learning, clustering is the main problem. This problem consists of dividing a population into a set of groups (clusters) $\{c_1, ..., c_K\}$ with respect to a set of features $\mathbf{X} = \{X_1, ..., X_p\}$ that characterizes the population. In order to solve the problem we start from a sample of the population $\mathbf{x}_i = (x_{i1}, ..., x_{ip}), \forall i = 1, ..., N$, from which we infer how to group the individuals according to the features. Notice that the sets of groups where the population is divided into can be disjoint or not. In the last case, we are talking about overlapping clustering and its basis is that an individual does not have to belong just to one cluster. In this work we will consider both cases.

The clustering problem is not only important in fields such as computer sciences (web mining, spatial database analysis, textual document collection, image segmentation), life and medical sciences (genetics, biology, microbiology, paleontology, psychiatry, clinic, pathology), earth sciences (geography. geology, remote sensing), social sciences (sociology, psychology, archeology, education) or economics (marketing, business) (Xu and Wunsch (2005)) but is also related to other problems in the ML field like regression, prediction, data mining or pattern recognition (Saxena et al. (2017)). In Ghosal et al. (2020) we can find different fields where clustering has been successfully used to harness information, and the algorithms applied to do so.

As noted in Xu and Wunsch (2005), there is no agreed definition on what a cluster is, but rather a baseline idea that a cluster should comprise internal homogeneity and external separation. The idea is that similar individuals should be grouped together while being different to other grouped individuals. Consequently, how to measure similarity and dissimilarity between patterns should be established in a meaningful way, this being one of the keys of the complexity of clustering as there exist different ways to do so. These ways to solve the clustering problem entail different models and possible solutions and there is no general way to know what is the best approach to the problem at hand. Also, as this is unsupervised learning there is no universal way of measuring the quality of solutions. This usually leads to choosing the way to measure similarity subjectively. We also have that choosing the number of clusters is not a trivial problem.

All of this makes clustering analysis a non-sequential process (see Figure 3.1), i.e., repetition and trials are needed to correctly solve the problem, which usually needs expert knowledge to confirm the results (Xu and Wunsch (2005)). This gets even more complicated if we introduce explanations and model interpretability. Nonetheless, based on the works by Xu and Wunsch (2005), Ghosal et al. (2020), Saxena et al. (2017) and Madhulatha (2012), we present the different clustering techniques that have been developed and end up with a discussion and revision of interpretability within the problem of clustering.

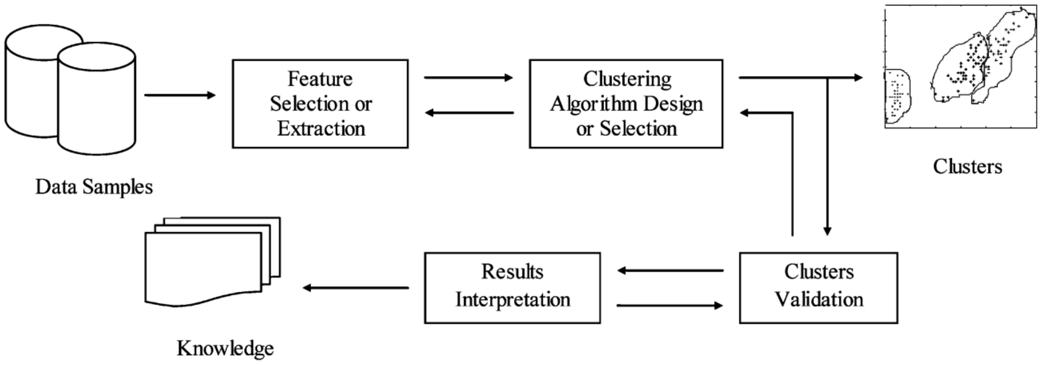

Data Samples
Feature Selection or Extraction
Clustering Algorithm Design or Selection
Clusters
Knowledge
Results Interpretation
Clusters Validation

Figure 3.1: Clustering process (image from Xu and Wunsch (2005)).

There are different ways in the literature for categorizing clustering techniques being the division into hierarchical and partitioning the most used one. To this division we can add six categories: model-based methods, density-based methods, grid-based methods, probabilistic clustering, graph-based methods and optimization-based methods. Here we present the idea and different algorithms of each type of technique starting with hierarchical clustering.

Given a sample of individuals of a population $\mathbf{x}_i = (x_{i1}, ..., x_{ip}), \forall i = 1, ..., N$ the idea of hierarchical clustering (Nielsen (2016)) is to group these individuals iteratively forming groups (agglomerative approach) or dividing groups (divisive approach). In the first case we start from all the individuals conceiving each one as a group and start making bigger groups by joining them until we have just a group containing all of them. On the other hand, the divisive approach is the reversed.

Both processes render information on how the instances group together through the iterative process, which is finally used to select the clusters. This is usually represented in a dendrogram (see Figure 3.2). The way of joining or dividing clusters is by using inter-cluster distances, which measure how similar/dissimilar are clusters in order to group/divide them. These distances are defined over previous distances defined in the space of features like the Euclidean distance, although there are various possible distances (Saxena et al. (2017)). This means that first we need to measure distances between individuals and then we generalize it to distances between clusters. Given a distance $d(x, y)$ defined in the feature space, there exist three typical distances between clusters:

- **Single-linkage clustering.** Given A, B two sets (clusters), their similarity is measured as :
$$min\{d(x, y) \mid x \in A, \;\; y \in B\}$$
- **Complete-linkage clustering.** Given A, B two sets (clusters), their similarity is measured as:
$$max\{d(x, y) \mid x \in A, \;\; y \in B\}$$
- **Average-linkage clustering.** Given A, B two sets (clusters), their similarity is measured as:
$$\frac{1}{\mid A \mid\mid B \mid} \sum_{x \in A} \sum_{y \in B} d(x, y)$$

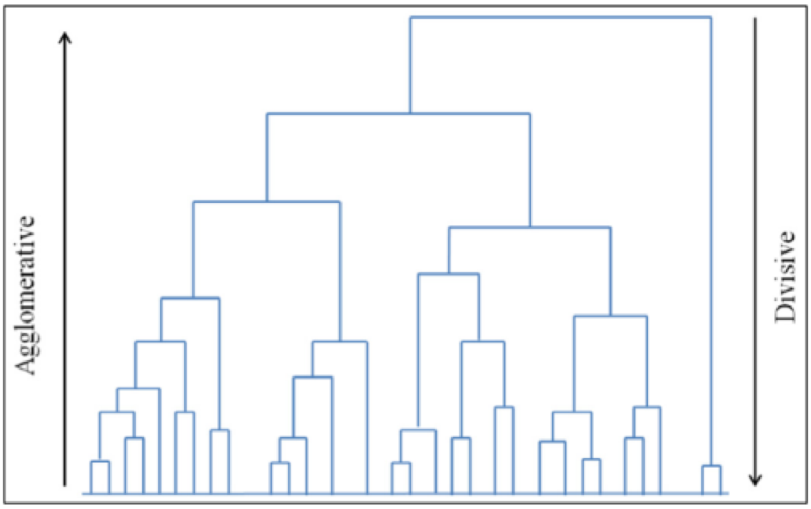


Figure 3.2: Example of dendrogram from Saxena et al. (2017).

So, in the case of agglomerative clustering, for each step we join the clusters of minimal distance between them and in the case of divisive clustering, we separate the clusters with the maximum distance.

There exist three main points of critique to this method. First we find that it is highly expensive in computational terms, especially the divisive approach. The second deficiency is that it lacks robustness, thus being sensitive to outliers. The last deficiency is that when a point is related to a cluster, it stays assigned to the cluster for the rest of the process with no room for reviewing its assignment. However, for the last deficiency, variants of the algorithm has been proposed such as BIRCH (Zhang et al. (1996)). Finally we highlight that it tends to form spherical clusters, which has to be taken into account when deciding which method to use.

In partitional clustering we do not find clusters iteratively. Instead, clusters are obtained simultaneously by optimizing an established criterion related to the distance (similarity) between the individuals of the sample $\mathbf{x}_i = (x_{i1}, ..., x_{ip}), \forall i = 1, ..., N$. The idea here is to assign points to clusters through the use of minimum distance, i.e., highest similarity. The distance to each cluster is calculated by means of the distance of the point to a cluster representative or centroid. The most known algorithms in this type of technique are k-means (MacQueen et al. (1967)) and fuzzy c-means (Bezdek (1973)), though the last one can be considered out of this group due to the introduction of fuzzy logic, but its basis is the same.

The idea of k-means is to define K centroids, each one for each cluster and optimize (minimize) an objective function $J = \sum_{j=1}^{K} \sum_{x_i \in c_j} \|\mathbf{x}_i - \mathbf{y}_j\|^2$ where we are calculating a chosen distance between individual $\mathbf{x}_i$ and a centroid $\mathbf{y}_j$ of cluster c_j to which the individual has been assigned. An individual is associated to a cluster c_j if the centroid $\mathbf{y}_j$ is the nearest to the individual with respect to the similarity distance established (usually the Euclidean distance is used in J).

In order to minimize the objective function there is an iterative process defined where initial centroids are established, then individuals are grouped based on these centroids and once grouped, centroids are recalculated as the mean of the individuals assigned to the clusters that they represent. This is repeated till centroids no longer move.

Notice that for each initialization of the clusters we obtain a final partition. Though this can be used to optimize J, these calculations are usually made for just one or various initializations and the best result is chosen. Also, there exists a variant called k-medoids using the median. This is used when the mean cannot be computed or in the presence of outliers (the median is more robust than the mean). Finally, we mention a variation called kernel k-means which uses kernels to transform the feature space into another one of higher dimension where there exists a possibility that data is separated linearly.

Fuzzy c-means proposes a very similar technique but fuzzy logic is introduced in order to model individual's degree of belonging to each cluster. In contrast, k-means is based on classical logic, i.e., each individual belongs to a cluster or not. As a consequence, here we are dealing with overlapping clustering. In this technique each individual has a degree $u_{i,j} \in [0, 1]$ of belonging to each cluster, which yields an optimization function:

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{K} u_{ij}^m \|\mathbf{x}_i - \mathbf{y}_j\|^2 \quad 1 < m < \infty$$

m is the fuzzy partition matrix exponent for controlling the degree of fuzzy overlap, hence controlling how fuzzy are the clusters represented by centroids $\mathbf{y}_j$ $j = 1, ..., K$. As in k-means, initial centroids are defined. Once this is established, distances from the point to centroids are calculated and used to compute belonging degrees and with them new centroids are calculated. This procedure iterates until a maximum number of iterations or convergence are achieved.

Both algorithms and in general partitional techniques present the advantages of being suited when clusters are spherical and separated. We also have that they are scalable and relatively simple, which is good for large datasets. In contrast, they present disadvantages such as reliance on the user to specify the initial number of clusters (there is no general way to select this parameter), poor performance when non-convex clusters are present and sensitivity to the initialization of centroids and outliers.

We continue with model-based clustering methods whose main characteristic is the use of ML and mathematical models to obtain the clusters. Though these models are used, the basic notion of clustering still remains, i.e., the objective is to group based on similarities and dissimilarities. Here the most known approaches are the use of neural networks and the use of decision trees. The idea applied in neural networks is to make neurons compete with each other and when a neuron is active it learns and reinforces its neighborhood. By doing this, neurons concentrate in different parts of the space representing the density function and we obtain the clusters. The most used algorithm is self organizing maps (Kohonen (1990)), which presents the disadvantages of needing the number of clusters and misrepresentation of certain areas (areas with real high density underrepresented and vice versa).

In the case of decision trees, there are different ways to approach the problem. One of them (more will be reviewed later) for example is based on the idea that if we have individuals which can be grouped differently, then the data points cannot be uniformly distributed. Thus, if we introduce some uniformly distributed points and conceive them as a class, while conceiving the sample of individuals a different class, we can make a classification problem. Decision trees in classification divide the feature space in hyper-rectangles where a class of the possible ones predominates. These hyper-rectangles are obtained by means of the leaf nodes. Therefore, by solving the classification problem created we obtain the clusters as the leaf nodes of the tree whose output is the class of the individuals (see Figure 3.3).

Under all of this we are grouping points through their similarity based on the Euclidean metric (in the case of continuous features). We highlight that there are variants which try to avoid the definition of the uniformly distributed points (Liu et al. (2000)) since it is not a trivial problem. Finally, we would like to mention that there are other models such as support vector clustering (Ben-Hur et al. (2001)) which has its basis on the theory of support vector machines (Boser et al. (1992)).

The main idea in density-based clustering methods is that a cluster is considered a region of the space where the density of the individuals sample exceeds a predefined threshold. The density of the individuals sample in a region is the proportion of the data points with respect to the area/volume of that region. The main advantage of these methods is the discovering of arbitrary-shaped clusters. The most remarkable algorithm here is $DBSCAN$ (Ester et al. (1996)).

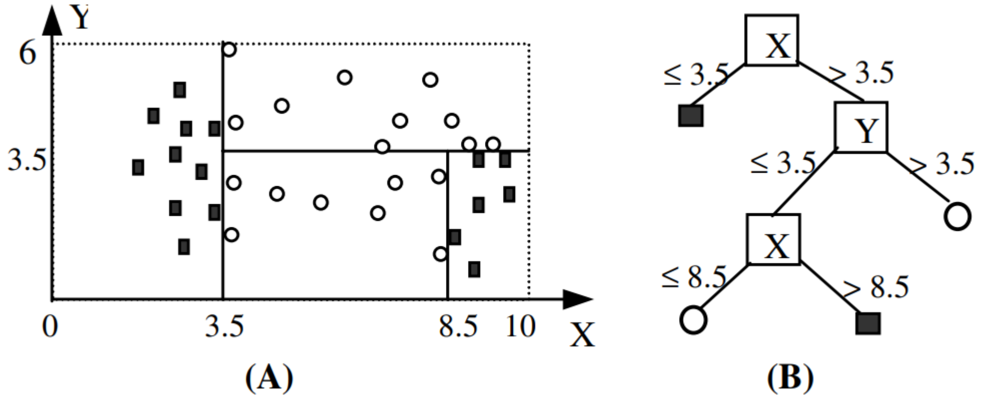


Figure 3.3: Example of decision tree-based clustering with introduction of uniformly distributed points (Liu et al. (2000)).

Next, we introduce grid-based clustering techniques. The main idea of the first one is to define a grid of cells in the space to work with, instead of the individuals of the sample. The idea is to compute the density of each cell based on the samples. After this we eliminate those cells whose density does not go above a certain threshold. Once all of this is done the remaining cells are used to form clusters based on similarity. The main advantage is the fast processing time, on the contrary disadvantages are the set of hyperparameters, such as the threshold, and the problems that a large dataset (in terms of features) can cause.

Next we show the mixture density-based clustering techniques which falls under what is known as probabilistic clustering, where BN clustering is placed. We would like to mention that in the literature probabilistic clustering can be seen classified as model-based. There are also authors that consider this type of methods to be simultaneously density-based and model-based (Fraley and Raftery (1998)). Nonetheless, the ideas of this type of methods make it worth of its own category.

The two main ideas of probabilistic clustering is to assume that individuals have a prior probability of belonging to each cluster $\pi_1, ..., \pi_K$ and that data follows a different distribution for each cluster (data is generated by different probability distributions). The classic approach to modelling the problem at hand is to assume that we have got a set of random variables $\mathbf{X} = \{X_1, ..., X_p\}$ (features of the population) and a random latent discrete variable H, which takes values $\{h_1, ..., h_K\}$ (clusters). Hence, we are dealing with a joint probability distribution $P(\mathbf{X}, H)$. Then, this probability function is assumed to decompose as $P(\mathbf{X}, H) = P(H)P(\mathbf{X} \mid H)$, thus we have a finite mixture model (FMM) (mixture density-based clustering techniques). With these assumptions we have that:

$$P(\mathbf{X}) = \sum_{j=1}^{K} \pi_j P_j(\mathbf{X} \mid h_j)$$

where $P_j(\mathbf{X} \mid h_j)$ for each j represents the different probability distributions from which individuals come. The problem is solved by obtaining the number of clusters, i.e., the value of K, which is usually assumed to be known, the parameters estimations if we are assuming $P_j(\mathbf{X} \mid h_j)$ to follow a specific distribution and the prior probabilities of individuals belonging to each cluster $\pi_1, ..., \pi_K$.

Here, one of the biggest problems of the maximum likelihood estimators for the parameters of the distributions is that they cannot be computed directly due to the latent variable. In this case, the EM algorithm (Dempster et al. (1977)) appears, which will be discussed more deeply later. One of the most known models assumes $P_j(\mathbf{X} \mid h_j)$ to be a multivariate Gaussian, hence we work with a Gaussian mixture (GM). One of the problems of this model is the formation of spherical and elliptical clusters. We also mention Autoclass (Cheeseman and Stutz (1996)) which covers different distributions such as Bernoulli or log-normal. Finally, we would like to note that we have distinguished between mixture density-based and probabilistic clustering since BNs allow more decompositions of $P(\mathbf{X}, H)$ than just the FMM.

Finally, we present graph clustering and techniques based on optimization algorithms. The graph clustering techniques share in common the introduction of graph theory in order to achieve the desired clusters. The basic idea here is to construct a graph where nodes are the individuals of the sample. Similarity is embedded into the edges of the graph with the ultimate goal of finding communites/clusters in the graph by means of graph theory, decomposing the network using topological properties.

One of the most remarkable algorithms here is spectral clustering (Donath and Hoffman (1973)) where the idea is to construct a similarity graph (adjacency matrix is weighted based on similarity between instances) and use the eigenvectors of the Laplacian matrix to perform k-means and obtain the clusters. The advantages of this algorithm are the lack of assumptions on cluster form, the scalability under sparse adjacency matrix and no local suboptima. This algorithm can be seen as a "starting algorithm" for k-means, though not everything is perfect since establishing a good similarity graph is not a trivial problem.

With respect to techniques based on optimization, the main idea here is to take advantage of optimization algorithms to solve the clustering problem. We would like to mention that methods can be divided into evolutionary approaches such as genetic algorithms and search-based approaches such as simulated annealing.

Now that we have reviewed the main clustering techniques, in the next section we discuss the needs of explanations when solving this problem. To sum up, the concepts presented in this section recall two main ideas. First, the clustering problem involves grouping individuals based on features that describe them, and the underlying idea for doing so is the use of similarity and dissimilarity, which can be established in different ways. Second, the clustering process is rather iterative since there are several ways of establishing similarity and obtaining clusters with no general way to determine the suitable one. This is due to the fact that there is no general way to measure the quality of solutions, and determining the number of clusters is not trivial.

3.2 Interpretable clustering

Leaving aside the complexity of the whole process of solving a clustering problem, we just have a ML model that learns from data in a specific way. Thus, it is natural that why-questions arise in this context, like for example: "*why does an individual belong to the cluster the model establishes?*" or "*why do two individuals with the same values for a subset of the features belong to different clusters?*". Although it is true that in the unsupervised learning paradigm we do not have an objective/problem with prior knowledge like for example in supervised learning classification, this does not exclude the need of explanations as exemplified.

Moreover, one could argue that no prior knowledge accentuates the need of explanations since we are trying to harness new and relevant information without knowing what we are seeking. This can be seen in one of the problems associated with clustering, clustering naming, which consists of giving meaningful names to each cluster obtained. Hence, understanding the model itself can be quite useful for the clustering problem and this just does not call for explanations but for model interpretability. The drawback here is that most techniques and models introduced are not interpretable. Next we present works done in this direction.

Yang et al. (2021) define a categorization of interpretable clustering techniques which comprises rule-based, rectangle box-based and decision trees-based methods, see Figure 3.4.

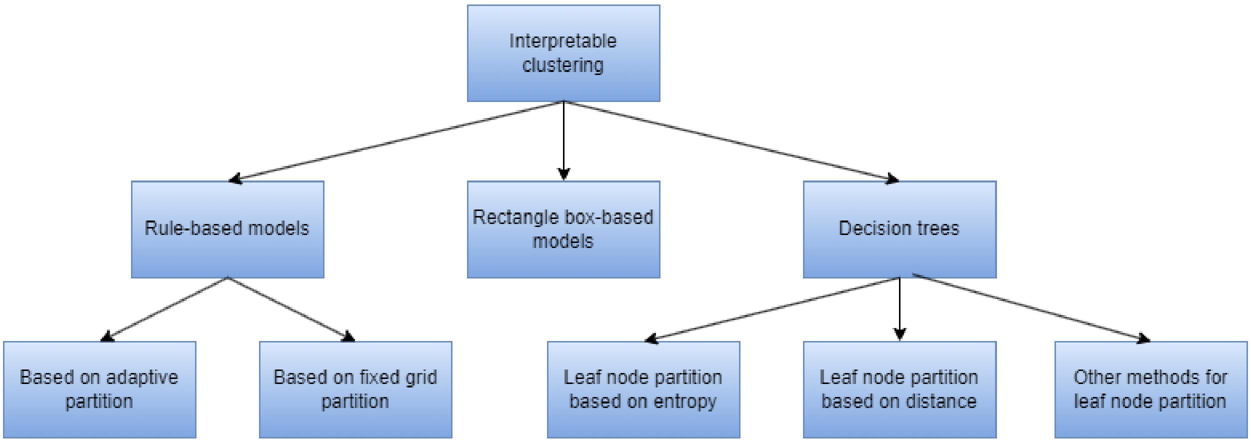

Interpretable clustering
Rule-based models
Rectangle box-based models
Decision trees
Based on adaptive partition
Based on fixed grid partition
Leaf node partition based on entropy
Leaf node partition based on distance
Other methods for leaf node partition

Figure 3.4: Classification of interpretable clustering methods (image from Yang et al. (2021)).

However, one could argue that this categorization is not complete since for example, it does not take into account BN-based clustering, that directly inherits the interpretability of BNs as we will see in Chapter 4. Nevertheless, this categorization is useful to show the little space that interpretable clustering occupies within the clustering framework. Next, we review some works in interpretable clustering.

Rule-based models come from knowledge representation theory and are based on generating rules of type $if - then$, where the clause if contains a condition which in the case of the ML context is defined in terms of the features of the problem. In the ML context it is an intepretable model due to the simplicity of the rules which can be directly explained in understandable terms for humans, thus interpretable. This is due to the fact that rules are designed to store knowledge in this case being the condition imposed on the features. We would like to notice that the design of rules does not only allow to store knowledge but also grants reasoning with a reasoning-engine based on induction. We would also like to mention that ruled-based models are highly related to decision trees since the latter can be interpreted in terms of rules. Also in the context of rectangle box-based clustering, methods are related to rules for the same reason.

Saisubramanian et al. (2020) present an adaptive partition-based method that creates a model that optimizes for interpretability by using k-means and rules. The idea is to predefine a set of features that signify interpretability for the user. The usage of this set is to measure the interpretability of clusters by the proportion of individuals that share the same feature value for these variables within the clusters. With this measure the idea is to set a minimum β value for this measure and find clusters that score above it. By doing this the clusters are interpretable as they share the same feature value for the interpretability features. For finding the clusters, a modified version of k-means is presented. Finally, once the clusters are obtained these values conform to a special set of rules to interpret the clusters.

Chen (2018) presents generative interpretable clustering model with feature selection, which is another adaptive partition-based method. The model generates a list of rules involving different sets of features for each cluster. The goal is to obtain a final model consisting of a list of rules for each cluster based on the idea that if a sample corresponds to a cluster, then the rules of the list tend to be satisfied. Each rule for each cluster contains only a feature. The rules are obtained via one-dimensional GM models that learn the partitioning of individual features. From this point the author establishes a method to obtain whether a feature characterizes a given cluster; if it is so then the feature is used to add a rule to the list of rules of the cluster.

Other works are based on fuzzy rules which are just an adaptation of rules to fuzzy logic. In Wang et al. (2014), a traditional membership function is used combined with a new fuzzy adaptive partitioning of the feature space to create the rules, making it more flexible. However, the method fails to make the individuals membership sum to one across the clusters and this can affect the interpretability. On the other hand, we can find models based on a fixed grid partition. Mansoori (2011) works with trigonometric membership functions and a totally different way of obtaining the rules. In this case just as in the methodology presented for decision trees, the idea is to randomly introduce data and make the clustering problem a classification problem. The $SGERD$ algorithm (Mansoori et al. (2008)) is used to obtain the rules. Finally, Hsieh et al. (2016) present a work where trapezoidal membership functions in combination with genetic algorithms are used to obtain the rules.

We continue with the rectangle box-based methods. The base of this type of models is the goal to create hyperrectangles in the feature space that represent the clusters. The next two methods that we are introducing are soft in the sense that they assume probability distributions. This implies that individuals have a "membership" or belonging degree (do not confuse with fuzzy logic, here we work with densities and probabilities) to the clusters.

In the work of Pelleg and Moore (2001), the main idea is that each of the clusters follows a Gaussian distribution for each dimension with its mean value being an interval $[L, H]$ where L, H are the respective lower bound and upper bound that define the hyperrectangle representing the cluster. Each of the boundaries are found by fixing one of them and maximizing the likelihood of the given sample in order to find the other. This is repeated iteratively until convergence.

The other work is Chen (2018) which introduces discriminative rectangle mixture, where the main idea again is to define a distribution for the hyperrectangle based on the lower and upper bounds that define it (L, H) and then obtain them based on the sample of individuals, thus, obtaining the clusters. One of the most remarkable features of this algorithm is that it works with two sets: in the first one cluster-preserving features are used to obtain the distributions (the hyperrectangles), and in the second one, the rule-generating features are used to create the rules based on the hyperrectangle for the clustering (this method can also be seen as ruled-based).

Finally, we tackle the decision tree-based methods. These methods belong to model-based clustering, as the base of all of them is to use decision trees models, which are interpretable. From Section 3.1 we have that there are different approaches using decision trees and we already have presented one; in the following we discuss the rest. We start with those methods which directly construct a decision tree, as in the supervised model, by splitting the nodes without transforming the problem into a classification one. In the case of clustering, the splitting of the nodes is made using (dis)similarity measures instead of the class label or mean squared error as in the supervised case. The clusters are obtained at the leaf nodes.

We first start with those methods within the leaf node partition based on distance category. In Fraiman et al. (2013) the cutting point is established through the use of an heterogeneity measure defined by the authors. This measure is calculated before splitting and then we make the difference with the sum of the left and right heterogeneity measure calculated for several splitting nodes and points. The highest value of these differences determines the cutting point. This process is followed iteratively until the measure of dissimilarity of the two new clusters obtained after each iteration reaches a certain threshold.

In Frost et al. (2020) the idea remains the same but in this case they use a measure to produce the splits called surrogate cost, which uses the result of a previously applied k-means. Hence the objective is to minimize the surrogate cost through the splits until a certain preselected number of clusters is reached. By doing this the algorithm tries to emulate the results of the k-means but with an interpretable model (the decision tree). We follow with the approach of Bertsimas et al. (2021). In this case the idea is to translate the problem into a mixed-integer optimization problem, where a quality measure of the clusters such as the silhouette metric is optimized (maximized). Then, an algorithm using a coordinate-descent procedure is used to obtain the optimal tree from the whole tree space.

With regard to the leaf node partition based on distance methodologies, we would like to present the work of Ghattas et al. (2017) which reproduces the ideas of Fraiman et al. (2013) but using an entropy-based measure to work with categorical data (leaf node partition based on entropy). The last approach we introduce is characterized by the introduction of fuzzy logic. This method belongs to the other methods for leaf node partition category. The main idea is to take advantage of fuzzy logic and the flexibility of fuzzy partition to construct an interpretable model as the decision tree is. We highlight the work of Jiao et al. (2022) where the algorithm proposed does not need prior knowledge about the number of clusters.

Finally, to close the section, we would like to mention that there also exist *post-hoc* approaches which try to render interpretable those clustering methods that are not. Moshkovitz et al. (2020) make efforts to implement a *post-hoc* interpretation of the clusters provided by the k-means model. The authors use the results provided by k-means to construct a classification tree. The classification labels are the clusters obtained by the k-means algorithm. This methodology is known as surrogate model (Section 1.2.2). On the other hand, we can find works such as Balabaeva and Kovalchuk (2020) and Dronov and Evdokimov (2018) which focus on obtaining model-agnostic methods. The first one uses Bayesian inference to study the differences between clusters through the comparison of posterior distributions of features. The second one gives a *post-hoc* analysis based on the degree of influence of the features towards the clustering partition obtained.

Chapter 4

Bayesian network-based clustering

In this chapter we will introduce the clustering problem with BNs. We will describe the underlying ideas of the model in the context of clustering and how to use it to solve the problem. As mentioned in the previous chapter, this approach falls under probabilistic clustering which is a soft method since we have probabilities of belonging to the clusters for each individual ("degrees of belonging"). This is natural since the BN is a probabilistic model which is designed to factorize the probability distribution function of a certain set of features by using the conditional independencies.

Thus, solving the problem of clustering with BNs consists of finding a factorization $P(H, \mathbf{X}) = P(H \mid \mathbf{Pa}(H)) \prod_{i=1}^{p} P(X_i \mid \mathbf{Pa}(X_i))$. Notice that this covers FMM models by simply adding the assumption that the latent variable H cannot have any parent. The factorization of the joint probability distribution (the structure G and CPDs) is usally learnt from data (Section 2.1).

Given a particular structure G of a BN, the learning of the CPDs from data takes three possible approaches: parametric, non-parametric, semiparametric/hybrid (Section 2.1). All of the methods presented in Section 2.1, independently of the approach, contemplate complete data, i.e., they do not work with missing values. Since in the problem of clustering we are introducing a latent variable, we do not have missing values but a whole variable with no available data. Hence, alternatives must be sought and this is where the Expectation Maximization (EM) algorithm (Dempster et al. (1977)) comes in.

The main idea with missing data is to work with the marginal loglikelihood of the observed dataset D to obtain the MLE of the parameters of the target distribution. In other words, the MLE are determined from the marginal loglikelihood of the data $log\mathcal{L}(D;\boldsymbol{\theta})$:

$$log\mathcal{L}(D;\boldsymbol{\theta}) = logP(D \mid \boldsymbol{\theta}) = \sum_{h \in H} logP(D, h \mid \boldsymbol{\theta}) = \sum_{h \in H} log \prod_{\mathbf{x} \in D} P(\mathbf{x}, h \mid \boldsymbol{\theta})$$

$$= \sum_{h \in H} \sum_{\mathbf{x} \in D} logP(\mathbf{x}, h \mid \boldsymbol{\theta})$$

Since this is hardly ever feasible, the EM algorithm allows approximating the maximum likelihood estimators with incomplete or missing data. In order to achieve this, the EM algorithm works with the expected loglikelihood with respect to the conditional distribution of H given the observed data points, $\mathrm{x} \in D$, under certain given parameters $\boldsymbol{\theta}_t$.

The process involves two steps: expectation and maximization. These steps are iteratively repeated until convergence or maximum number of iterations are achieved. In the expectation step the expected loglikelihood with respect to the conditional distribution of H is computed given $\mathrm{x} \in D$ under certain given parameters $\boldsymbol{\theta}_t$. In the maximization step the next value of the parameters is obtained, $\boldsymbol{\theta}_{t+1}$, that maximizes the expected likelihood computed.

We are given a set of the observed random features, $\mathbf{X}$, and a latent variable, H, which follow a joint parametric probability distribution, where $\boldsymbol{\theta} \in \Theta$ are the parameters of the distribution. Then, given a dataset D of the observed data and a particular value $\boldsymbol{\theta}_t \in \Theta$, the expected loglikelihood with respect to the conditional distribution of H given D under $\boldsymbol{\theta}_t$ is:

$$Q(\boldsymbol{\theta} : \boldsymbol{\theta}_t) = \mathbb{E}_{H|\mathbf{X},\boldsymbol{\theta}_t}[log\mathcal{L}(\boldsymbol{\theta}; D, H)] = \sum_{\mathrm{x} \in D} \sum_{h \in H} P(h \mid \mathrm{x}, \boldsymbol{\theta}_t) log P(\mathrm{x}, h \mid \boldsymbol{\theta})$$

With this in mind the EM algorithm is found in Algorithm 2.

Algorithm 2: EM algorithm

```
input  : $(\boldsymbol{\theta}_0, D)$
for $t = 0, 1, ...$ to convergence do
    E-step: compute $Q(\boldsymbol{\theta} : \boldsymbol{\theta}_t) = \mathbb{E}_{H|\mathbf{X},\boldsymbol{\theta}_t}[log\mathcal{L}(\boldsymbol{\theta}; D, H)]$ ;
    M-step: Find the parameter value $\boldsymbol{\theta}_{t+1} = argmax_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} : \boldsymbol{\theta}_t)$ ;
    if $Q(\boldsymbol{\theta}_t : \boldsymbol{\theta}_t) \geq Q(\boldsymbol{\theta}_{t+1} : \boldsymbol{\theta}_t)$ then
        Result: $\boldsymbol{\theta}_t$
    end
end
```

The key is that it is demonstrated that improvements on $Q(\boldsymbol{\theta} : \boldsymbol{\theta}_t)$ lead to improvements on the marginal loglikelihood ($log\mathcal{L}(\boldsymbol{\theta}; D)$). Also Wu (1983) demonstrates the convergence of the algorithm to a stationary point (does not imply the maximum) under some regularity conditions. The main drawback of this algorithm is the sensitivity to the parameter initialization $\boldsymbol{\theta}_0$.

It is worth to mention that there exist different variants of the algorithm from which we highlight the next: Generalized EM (Dempster et al. (1977)) and Monte Carlo-EM (Levine and Casella (2001)). The first one addresses the problem of an intractable maximization step as depending on the parametric distribution established this step can be problematic. The second one addresses the problem of intractable or infeasible expectation step and will be used later in Chapter 5 in the methodology proposed.

In the context of BNs, Lauritzen (1995) introduced the EM algorithm for discrete BNs. For networks working with also continuous variables, thus hybrid, we are not aware of any explicit development of the EM algorithm. In this case, problems with the expectation or maximization step can arise such as obtaining the latent variable posterior distribution $P(H \mid \mathrm{x}, \boldsymbol{\theta}_t)$. However, in these cases the usage of the generalized-EM or Monte Carlo-EM can help overcome those issues.

For learning the structure (DAG) G from the given data there are three main approaches: detecting conditional independencies, score+search methods and the hybrid methods (Section 2.1). Hybrid methods comprise the usage of both methods based on testing conditional independencies and score+search methods. The approach of detecting conditional independencies is quite natural since we are working with a model that tries to factorize the joint probability distribution function based on these conditional independencies of the features. The main idea is to study the conditional independencies of the features by means of statistical tests and then find the structure that represents most/all the relationships found. There are three main issues to deal with in this method: reliability and complexity of the tests, order of the variables and equivalent Markov DAGs.

With the increase of the conditional set of variables with respect to which we are testing, the reliability of the results decreases and the complexity increases. The order of the variables makes reference to the problem of establishing which conditional (in)dependencies to test. The main approach is to assume an order σ of the variables where each feature X_i can only depend on the features $X_1, ..., X_{i-1}$. Notice that finding the optimal order is a difficult problem. A notion of guidance in this problem is that a good order should respect causality, i.e., causes come before effects in the order. Finally, two DAGs are Markov equivalent if they represent the same conditional independencies. This can be identified by v-structures (see Section 2.2) of DAGs with the same edges (not arcs). Hence, we cannot find a unique solution if we work with the space of the DAGs. To solve this problem the key is to work with the partially directed acyclic graphs (PDAGs) space, where unique solutions can be found. This is what is done in the PC algorithm (Spirtes et al. (2000)).

The score+search approach takes a different perspective and tries to find the structure that best fits the data. Thus, the idea is to find the best structure by optimizing a fit measure that also involves the data at hand $f(structure, data)$. This measure should be meaningful with respect to the task at hand (learning the factorization of a probability distribution). In order to do this, a score (fit measure), a space of structures and a search method for that space are needed. There are two main types of score measures which are penalized loglikelihood such as BIC and Bayesian scores. With respect to the search space we have three spaces: the DAGs space, the equivalent classes of DAGs space and the ordering of features space.

The penalized loglikelihood scores are used in combination with the DAGs space since for their computation we need the structure G. They add, as its name says, a penalty term to the loglikelihood of the data in order to avoid overfitting, since the loglikelihood monotonically increases with the complexity of the model. Given a dataset D it is defined as follows:

$$log\mathcal{L}(D;\boldsymbol{\theta}) - dim(G)pen(N)$$

where $dim(G)$ makes reference to the dimension of the structure G measured through the number of parameters to estimate when working under parametric assumptions. The $pen(N)$ term is a non-negative penalization function, where N is the number of instances, such as for example $\frac{1}{2}log(N)$ which renders the BIC score. It is worth mentioning that $dim(G)$ is unmeasurable within a non-parametric framework, hence, an idea to overcome overfitting is using cross-validation (Atienza et al. (2022b)).

On the other hand, Bayesian metrics introduce a Bayesian framework with the use of prior distributions. Particularly, the idea is to maximize the posterior distribution of the structure given the data $P(G \mid D)$ over the structure space. Since this is infeasible, by means of Bayes theorem, we have that:

$$P(G \mid D) = \frac{P(D \mid G)P(G)}{P(D)}) \Rightarrow P(G \mid D) \propto P(D \mid G)P(G)$$

where $P(D \mid G)$ is the likelihood of the data given a structure G and $P(G)$ the prior distribution of the structures. Under a uniform prior distribution of the structures $P(G)$, we have that the structure which maximizes $P(G \mid D)$ is the same as the one that maximizes $P(D \mid G)$. Working in the DAGs space, this assumption is taken along with the assumption that the parameters follow a particular distribution in the parameter space Θ in order to define the most known Bayesian score K2 (Cooper and Herskovits (1992)). Here the mathematical expression of $P(D \mid G)$ is:

$$P(D \mid G) = \int P(D \mid G, \boldsymbol{\theta})P(\theta \mid G)d\boldsymbol{\theta}$$

where $P(\theta \mid G)$ is a prior distribution of the parameters given the structure and $P(D \mid G, \boldsymbol{\theta})$ is the likelihood of the data given a fixed structure and parameters. Finally, we would like to mention that in Chickering (2002) there is a score+search method defined to work in the equivalence class space and the work of Larrañaga et al. (1996), where the ordering variable space is used in combination with genetic algorithms.

As in the parameter learning, these methods do not work with missing data, thus they are not adequate for problems with latent variables. Efforts have been made to overcome this problem. The most remarkable work is Friedman (1998) with the introduction of the Bayesian structural expectation maximization (Bayesian-SEM) algorithm. In this work the author translates the EM algorithm to a score+search learning structure context under parametric assumptions. The score used is the Bayesian/K2 metric.

In the expectation step we are going to deal with the expected score and in the maximization step instead of maximizing the expected score with respect to the parameters, it is done with respect to the structures. Next we introduce some notation and assumptions.

We assume that we have a given a dataset D of observed data from random features $\mathbf{X}$, and a random latent variable H from which we do not have data at all, with a parametric joint distribution $P(\mathbf{X}, H)$. We also assume the hypotheses given when defining the K2 metric (Cooper and Herskovits (1992)) and conserve the notation given. We would like to mention that the algorithm is developed for discrete BNs, though it can be adapted to continuous features. Under this defined framework, the Bayesian-SEM is shown in Algorithm 3.

Algorithm 3: Bayesian-SEM algorithm

```
input : $(G_0, D)$
for $t = 0, 1, ...$ to convergence do
    Compute the posterior $P(\boldsymbol{\theta} \mid G_t, D)$ ;
    E-step: for each $G$ do
        $Q(G : G_t) = \mathbb{E}_{H|\mathbf{X}}[logP(H, D \mid G) \mid G_t, D]$
        $= \sum_{\mathbf{x} \in D} \sum_{h \in H} P(h \mid G_t, \mathbf{x}) logP(h, \mathbf{x} \mid G)$
    end
    M-step: Choose $G_{t+1}$ that maximizes $Q(G : G_t)$;
    if $Q(G_t : G_t) \geq Q(G_{t+1} : G_t)$ then
        Result: $G_t$
    end
end
```

$P(h \mid \mathbf{x}, G)$ is the posterior probability of the latent variable given the data point $\mathbf{x} \in D$ and a structure G. Since we are working with priors, we have $P(h \mid \mathbf{x}, G) = \int P(h \mid \mathbf{x}, \boldsymbol{\theta}) P(\boldsymbol{\theta} \mid G, D) d\boldsymbol{\theta}$. We also have that $P(h, \mathbf{x} \mid G) = \int P(h, \mathbf{x} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta} \mid G) d\boldsymbol{\theta}$.

Notice that contrary to EM, where the maximization step is obtained as a closed function, here this is infeasible. But this is not a problem since we have a finite structure space and with an exhaustive evaluation of the expected Bayesian score we can select the maximum. However, in practice this is computationally intractable and this is where the search procedure comes in.

Similar to the usual EM, Friedman (1998) proved that improvements in the expected score leads to improvements in the marginal score, which implies that we really do not need to choose the maximum but to choose G_{t+1} that fulfills $Q(G_{t+1} : G_t) > Q(G_t : G_t)$. This is quite useful to justify the introduction of a search method to ease the computational burden. It is also demonstrated that under regularity conditions, this algorithm converges to a stationary point.

But all of this is presented in a general setting with $P(\mathbf{X}, H)$. For what the author calls factored models (BNs), the algorithm has to be adapted since usually the evaluation of $P(h \mid \mathbf{x}, G)$ can not be done efficiently. The idea is to assume that the posterior probability of the parameters $P(\boldsymbol{\theta} \mid G, D)$ is sharply peaked, as a consequence $P(h \mid \mathbf{x}, G) \sim P(h \mid \mathbf{x}, G, \hat{\boldsymbol{\theta}})$ ($\hat{\boldsymbol{\theta}}$ can be obtained by performing inference), where $\hat{\boldsymbol{\theta}}$ is the MAP of the posterior probability of the parameters. This value $\hat{\boldsymbol{\theta}}$ is obtained by means of the EM algorithm, gradient ascent or variations of these methods. The algorithm for BNs is shown in Algorithm 4.

Algorithm 4: Factored-Bayesian-SEM algorithm

```
input : $(G_0, D)$
for $t = 0, 1, ...$ to convergence do
    Compute the MAP parameters $\hat{\boldsymbol{\theta}}$ for $G_t$ given $D$ ;
    E-step: for each model G found in the search procedure compute: do
        $Score(G : G_t) = \mathbb{E}_{H|\mathbf{X}}[logP(H, D \mid G) \mid G_t, D]$
        $= \sum_{\mathbf{x} \in D} \sum_{h \in H} P(h \mid G_t, \mathbf{x}, \hat{\boldsymbol{\theta}}) logP(h, \mathbf{x} \mid G)$
    end
    M-step: Choose $G_{t+1}$ from the search procedure that maximizes $Score(G : G_t)$;
    if $Score(G_t : G_t) \geq Score(G_{t+1} : G_t)$ then
        Result: $G_t$
    end
end
```

Here we need to compute $P(h, \mathbf{x} \mid G) = \int P(h, \mathbf{x} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta} \mid G) d\boldsymbol{\theta}$, which arises the same problem as in the computation of $P(h \mid \mathbf{x}, G)$. Different options are presented in Friedman (1998), where the easiest one is to perform again MAP estimation of the parameters for the structures G of the search procedure.

We have established Bayesian-SEM as the most remarkable since it sets a precedent and basis for most of the works done in BN-based clustering. One of the keys for this is that it presents the advantage of not restricting the structure (at least for discrete data). Moreover, different works have been presented for particular network structures (see Figure 4.1), some of them similar to the work of Friedman. In Table 4.1 we present a modified table from Keivani and Peña (2016) that summarizes most of them. In Table 4.2 we can find the respective notation.

Different choices for Bayesian network-based clustering				
Algorithm	Parameter search	Score	Structure	Structure search
Peña et al. (1999)	BC + EM	LML	ENB	FSS/BSS
Peña et al. (2002)	BC + EM	LML	RBMN	FSS/BSS
Peña et al. (2004)	EM	Marginal BIC	-	EDAs
Pham and Ruz (2009)	EM	MI	CL multinet	MWST
Pham and Ruz (2009)	EM	CMI	SNB	MWST
Pham and Ruz (2009)	EM	CMI	TAN	MWST
Santafé et al. (2006a)	EMA	LL	NB	-
Santafé et al. (2006b)	EMA-TAN	LL	NB	-

Table 4.1: BN-based clustering references

BC + EM	Bound and collapse EM
EMA	Expectation model averaging
EMA-TAN	Expectation model averaging tree augmented naive Bayes
LML	log marginal likelihood
LL	Loglikelihood
MI	Mutual information
CMI	Conditional mutual information
ENB	Extended naive Bayes
RBMN	Recursive Bayesian multinets
CL multinet	Chow-Liu multinet
SNB	Selective naive Bayes
TAN	Tree augmented naive Bayes
NB	Naive Bayes
FSS/BSS	Forward structure search/Backward structure search
EDAs	Estimation of distribution algorithms
MWST	Maximum weighted spanning tree

Table 4.2: Acronyms

Notice that Santafé et al. (2006a) and Santafé et al. (2006b) do not have structure search procedure. This is because they do not follow the scheme of SEM. In the first method it is assumed that the structure space is restricted to the selective naive bayes structures (SNB). SNB are naive Bayes structures where there is the possibility of some features being independent of the cluster and rest of variables. Under this assumption they present a variant of the EM algorithm for the learning of the parameters of the network, which refers to expectation model average (EMA) algorithm.

The core idea is to learn the parameters of a naive Bayes (NB) model but taking into account information about all possible SNB since a NB model is too strict. In order to do this, in the E-step all sufficient statistics are computed for all possible SNB. After this, we have the model average step (MA-step) instead of the maximization step.

First with the statistics obtained, for each SNB, MAP parameters $\hat{\boldsymbol{\theta}}$ are computed from $P(\boldsymbol{\theta} \mid G, D)$ following Thiesson (2013). With these MAP parameters we can approximate $P(H, \mathbf{X} \mid G, D)$:

$$P(H, \mathbf{X} \mid G, D) = \int P(H, \mathbf{X} \mid \boldsymbol{\theta}, G) P(\boldsymbol{\theta} \mid G, D) d\boldsymbol{\theta} \sim P(H, \mathbf{X} \mid \hat{\boldsymbol{\theta}}, G)$$

Then, under the constraint assumption of the SNB space, given a dataset D we have that:

$$P(H, \mathbf{X} \mid D) = \sum_{G \in SNB} P(G \mid D) \int P(H, \mathbf{X} \mid \boldsymbol{\theta}, G) P(\boldsymbol{\theta} \mid G, D) d\boldsymbol{\theta}$$
$$\sim P(G \mid D) P(H, \mathbf{X} \mid \hat{\boldsymbol{\theta}}, G) \propto P(D \mid G) P(G) P(H, \mathbf{X} \mid \hat{\boldsymbol{\theta}}, G)$$

On the other hand we have that for a NB model:

$$P(H, \mathbf{X} \mid D) = P(H) \prod_{i=1}^{p} P(X_i \mid H)$$

Santafé et al. (2006a) adapt the work of Cooper and Herskovits (1992) to the missing value framework in order to have a closed form for $P(D \mid G)$. Moreover, under the assumption of structure modularity (Santafé et al. (2006a)) for obtaining a closed form for $P(G)$, both equations for $P(H, \mathbf{X} \mid D)$ can be equalized. Since all of the terms are in a closed form we can obtain a closed form, which takes into account all SNB structures, for the next iteration parameters of the NB model. Notice that we obtain a proportional closed form for the parameters, for the explicit equations see Santafé et al. (2006a). In the second work we find the $EMA - TAN$ (TAN stands for tree augmented naive Bayes which is a type of BN with a particular structure). This works proposes a variant that works with all possible TAN structures instead of SNBs.

$BC + EM$ (Peña et al. (2000)) is a variant of the EM algorithm for BNs with the same goal of learning the parameters that maximize the marginal likelihood. The scores MI and CMI come from the main idea of the work by Pham and Ruz (2009), which is to assume that $P(H, \mathbf{X})$ is modeled by a mixture of k Bayesian networks with a particular structure. They contemplate: Chow-Liu multinets, selective naive Bayes, tree augmented naive Bayes. The parameters and structures are learned maximizing the CML criterion (Celeux and Govaert (1995)) developed for BNs classifiers. In order to do this with latent variables they use the CEM algorithm (Celeux and Govaert (1992)), a variant of the EM algorithm. It is the usage of the CML criterion that leads to the usage of CMI and MI scores and makes the maximum weighted spanning tree (MWST) (Chow and Liu (1968)) the perfect structure search procedure.

We would like to highlight the particular approach that uses CL multinets and two important aspects about it. The first one is that a mixture of CL multinets is not a BN; thus, the ideas proposed are adapted to this case. The second one is that with this approach we do not have the same structure for each cluster but a particular tree-structure for each one of them. This is quite useful since it allows learning under the assumptions that the conditional independence relationships of the features are context-specific for each cluster.

For the column of structure, ENB denotes extended naive Bayes introduced by Peña et al. (1999). This structure technically speaking is not a BN since it allows supernodes but still makes sense in this context. Supernodes are those nodes where we can have more than one feature. Furthermore, FSS/BSS stands for forward/backward structure search, since the idea to find the best structure is to start from the naive Bayes model (forward) or from the supernode with all features (backward) and combine or separate the nodes in the search structure step of Bayesian-SEM.

RBMN is the acronym for recursive Bayesian multinets, which are an extension of the Bayesian multinets (BMN), which are a generalization of BNs. The idea of BMN is to learn specific context (in)dependencies by instantiating a variable of interest and learning a BNs for the rest of the features. RBMN simply extends this idea to a tree shaped structure where internal nodes are the target instantiated variables and the leaf nodes are the BNs, allowing more than one variable to be instantiated. The idea proposed in Peña et al. (2002) is to use RBMNs and learn them through an iterative procedure based on the Bayesian-SEM where the final BMNs of the leaves are ENB.

In order for this to work we need the constriction that the cluster variable cannot be a target variable (contrary to the presented use of CL multinets). Finally, it is worth mentioning the work by Peña et al. (2004), where the authors introduce a variant of the Bayesian-SEM, which uses estimation distribution algorithms (EDAs) for the search strategy.

Please note, as pointed out in Keivani and Peña (2016) that all of these papers are developed for discrete BNs. Due to this fact it is possible to combine different methods of the parameter learning $(EM, BC + EM, EMA, EMA - TAN)$ and the structure search $(HC, FSS, BSS, MWST)$ (not all of them can be paired). The only constraint is to establish a score function with a closed and decomposable form, since we are working with BNs. For the case of continuous features this condition does not suffice as we also need to guarantee that the parameter search algorithm can be applied for the specific context.

Then, we have seen that the clustering problem can be solved using BNs. Therefore, we can take advantage of the interpretability of BNs for better understanding the clusters and problem at hand. Furthermore, all tools presented in Section 2.2 can be applied in this context. In the next chapter we show the potential usefulness of BNs within the clustering problem by giving a novel cluster characterization methodology for solving the clustering naming problem.

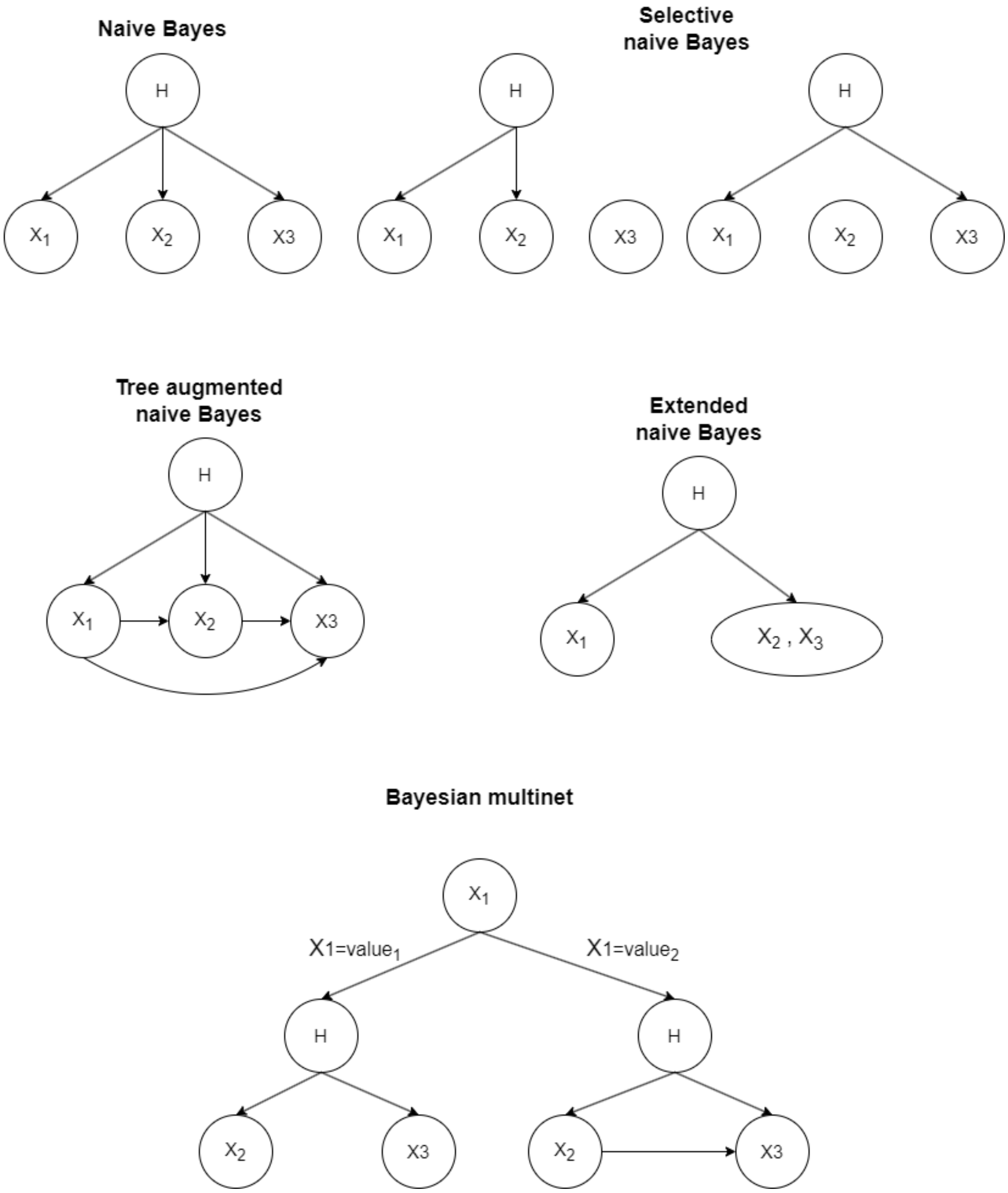


Figure 4.1: Different Bayesian network structures for clustering.

Chapter 5

Clustering naming with Bayesian networks

5.1 Clustering naming with Bayesian networks

As pointed out in Chapter 1, the unsupervised learning approach does not get away when it comes to explanations, since there are problems where those are needed. Particularly, the clustering problem as explained in Chapter 3 does not only lend itself for explanation needs but also, more specifically, interpretability is desired. Thus, we think that BNs can play a key role in this context for two reasons. First, it is an interpretable model in terms of knowledge embedded into the model. Second, since the clustering problem is more of an iterative and complex process with no ground basis for a granted solution, we think that the variety of tools of BNs for giving explanations for the model, the reasoning and the evidence (Chapter 2) can be very useful to ease this process.

As stated in Section 3.2, the clustering naming problem consists of giving meaningful and representative names to each cluster obtained. In this Chapter we introduce a method to characterize the clusters for solving the naming problem in order to show the mentioned potential of BNs within the clustering problem. The underlying question we are seeking to answer is: *why is this cluster different from the others?* By giving a characterization of each one of the clusters we harness close-packed and relevant information that can be useful to establish what makes each cluster different from the others. Thus, with the characterization of the clusters we can elucidate the representative names.

Before diving deeper into the method, we would like to mention that no quality assessment for explanations in terms of measures or properties are given for the method proposed for two reasons:

- As shown in Chapter 1 there is no ground theory for defining and evaluating explanations, which has led to a widespread literature nested in different specific contexts.
- Within the unsupervised learning framework, there is even less literature on how

> to define and assess the quality of explanations given by the models. One of the main reasons for this is that, contrary to supervised learning where everything revolves around the relationship of the features with respect to the target, in unsupervised learning there is no such target.

Hence, as concluded in Chapter 1, we are going to focus on how the explanation is obtained, which in this case is a characterization of the cluster, and which factors of interpretability are being taken into account. Ideally, we would use human evaluation but it is out of the reach of this work. We would also like to mention that no quality assessment of the cluster results, which is a part of the iterative process of clustering, has been made in the examples since the goal is just to show how the method works.

A usual approach for clustering characterization due to its simplicity is to find a representative individual of each cluster. The usage of a representative individual has two main advantages. First, we obtain in a compact way the representative values that the features take within the cluster. Second, with these representatives we can make comparisons between clusters. Therefore, this information can be useful in order to answer *why is this cluster different from the others?*, thus, it is helpful for giving names to each cluster.

For example a representative for continuous features can be found by averaging or calculating the median individual assigned for each cluster. Notice that in case of using k-means or k-medoids this is directly obtained with the results of the algorithm. Following this approach, our methodology finds a representative of each cluster but, in our case, the representative is found by MAP-estimation of $P(\mathbf{X} \mid h_i);\;\; h_i \in H$, for each $i = 1, ..., K$, where $\mathbf{X}$ are the features and H is the cluster variable.

We have a direct interpretation of representatives for this definition. A representative of a cluster is the most probable value of $\mathbf{X}$ given the cluster. Furthermore, each representative is the most probable value within each of the k different probability distributions from which data is assumed to be generated. One of the advantages of using this representative is that in the presence of multimodal probability distributions we are getting a better representative in terms of density value compared to the mean representative. This is due to the fact that in a multimodal distribution the mean can take values of low density value. On the contrary, the MAP representative is always one of the modes which implies a high density value.

Furthermore, in this case this idea could be extended to s-representatives for each cluster to capture the different modes, though interpretability and comparisons could be compromised. Another great advantage is that we do not have to distinguish between discrete or continuous data as long as MAP can be computed.

The methodology does not end here since we understand that to fully characterize the clusters via representatives in order to make comparisons, feature importance is needed as we could be making comparisons over features that are irrelevant for the cluster formation. Therefore, the idea is to establish feature importance for each of the representatives. One of the main highlights here is that we take advantage of the interpretability of BNs in order to define feature importance. Given an instance x and a distance $d(\cdot,\cdot)$, defined in the probability distribution space, we define the feature importance for each variable X_j as follows:

$$f_{X_j}(\mathbf{x}) = \mathbb{E}_{X_j|\mathbf{X}_{-j}=\mathbf{x}_{-j}}[d(P(H \mid \mathbf{x}), P(H \mid X_1 = x_1, ..., X_j, ..., X_p = x_p))]$$

The underlying idea of this new importance feature measure is to capture the effect of not having the information (value) of the target variable. This is achieved through the expected distance between the posterior probability of H, given x, and the posterior probability of H, given $\mathbf{x} = (x_1, .., x'_j, ..., x_p)$, where x'_j takes all possible values of feature X_j. Here different aspects must be analyzed in order to understand this measure:

- We are capturing the effect of the absence of the target variable by measuring how much the effect on the posterior probability of H of all the possible values that the target variable can take, differs with respect to the effect of the instance of interest (in our case the MAP-representative).
- We are using a distance $d(\cdot,\cdot)$ in the probability distribution space because we are measuring the difference of effects on the posterior probability of H. And we are measuring the effect of the instances on the posterior probability of H because this is the probability distribution that captures uncertainty of the belonging of an instance to the clusters. Consequently, by measuring distances of posteriors of H we seize the differences on the uncertainty the model has on the belonging of the instances to the clusters.
- Finally, the expected distance with respect to $P(X_j \mid X_1 = x_1, ..., X_{j-1} = x_{j-1}, X_{j+1} = x_{j+1}, ..., X_p = x_p)$ is taken due to the fact that given the partial instance $(X_1 = x_1, ..., X_{j-1} = x_{j-1}, X_{j+1} = x_{j+1}, ..., X_p = x_p)$ the values of X_j do not have the same probability. Thus, if each combination is weighted by this probability we are taking into account the plausibility of the instances to measure the difference in effects, which reflects reality better.

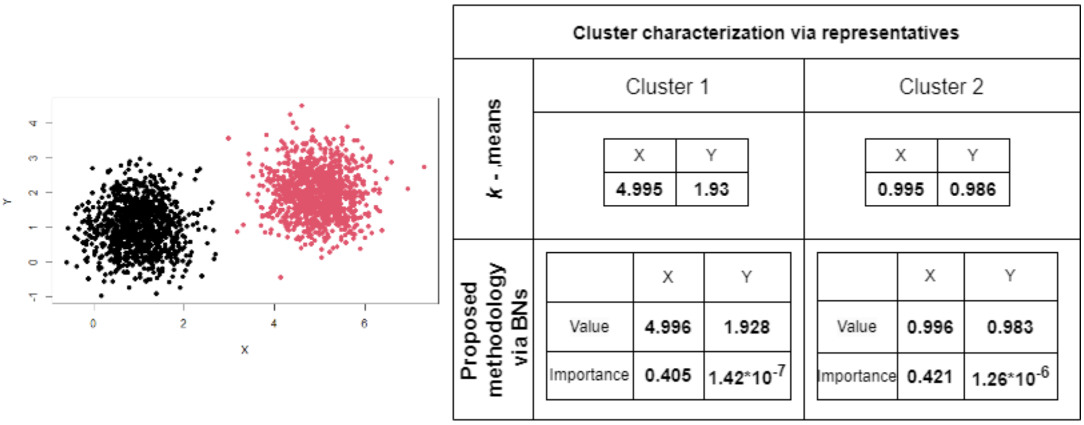
This shows the potential of the BNs since we have implemented an interpretable feature importance measure that allows better performance in the comparison of representatives by means of the interpretability of the model. Though, one could argue that comparison of the distances can be problematic as we do not know what is a big or small distance between distributions. This problem can be easily overcome using a bound distance such as the Hellinger distance:

Definition 5.1.1 *Given two discrete probability distributions $P = (p_1, ..., p_K)$ and $Q = (q_1, ..., q_K)$; the Hellinger distance is defined as:*

$$d(P, Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{K}(\sqrt{p_i} - \sqrt{q_i})} = \sqrt{2}\|\sqrt{P} - \sqrt{Q}\|_2$$

where $\|\cdot\|_2$ is the Euclidean distance. We have that the Hellinger distance is bounded which can be proved by the Cauchy-Schwarz inequality:

$$0 \leq d(P, Q) \leq 1$$



Cluster characterization via representatives

		Cluster 1		Cluster 2	
		X	Y	X	Y
k - means		4.995	1.93	0.995	0.986
Proposed methodology via BNs	Value	4.996	1.928	0.996	0.983
	Importance	0.405	$1.42*10^{-7}$	0.421	$1.26*10^{-6}$

Figure 5.1: Example with synthetic data for illustration. Code and synthetic data for the example can be found at **github**[2].

Finally, it is worth noticing that the importance feature measure can be applied to any other instance in any other context when needed. It can also be used with semi-supervised clustering and supervised classification with BNs. We present an example of the contribution of including the importance of the feature, see Figure 5.1. In this example synthetic data are created from two different 2-dimensional normal distributions (black and red points) which generate two distintic clusters as one can see in the left part of the figure.

In the example if an algorithm such as k-means was used, depending on the meaning of feature Y one could be misled when comparing the representatives and understand that the difference between them with respect to Y is relevant. Although, as we can see the value of Y for each representative does not allow to distinguish the clusters, hence does not characterize the clusters. Notice that in this case we can represent the features and clusters, but for higher dimensions (> 3) we would not be able to notice that feature Y is not important. On the other hand, we can see that our methodology captures the irrelevance of Y in both clusters.

We would also like to mention that other models such as k-means cannot implement this type of feature importance. Methods working with fuzzy logic could emulate this method, though it is not entirely clear how to measure in a meaningful way the differences between the membership degrees. Hence, only probabilistic clustering methods could adapt this methodology but only when posterior probabilities of the features can be computed.

Once introduced and discussed how explanations are obtained and shown the underlying natural ideas that can make them meaningful and useful, we are going to discuss the factors of interpretability taken into account. First of all the method does not take into account computational complexity. Thus, we are not considering time constraints and there can be different contexts where this method is not adequate for this reason. In terms of nature of user expertise, this method implies probability knowledge in order to process the results that it gives back. However, the form of the cognitive chunks are pairs of features values (MAPs) and importance, which makes

[2] https://github.com/Victor-Alejandre/Bayesian-Network-based-clustering.git

it really easy to understand even without probability theory knowledge. Nonetheless, in order to fully understand the result, it is needed.

In addition, it is a global explanation since we are characterizing the clusters produced by the model but it is important to notice that the feature importance measure is a local explanation generator. Also, the area of incompleteness we are attacking, as we have explained at the beginning, is the characterization of the clusters (which, is not known as we are dealing with unsupervised learning) in order to compare and understand them (*Why is a cluster different from the others?*). This can be used to achieve a solution for the clustering naming problem. Finally, it is important to mention that uncertainty has been dealt with since we are working with a probabilistic model and effects are also measured in the probability distribution space.

5.1.1 Implementation

We have implemented our methodology for categorical data in Python. The whole code can be found in **github**[3]. In this section we show an example applied to a customer dataset (Sharma (2021)), but before that, several issues about the implementation must be discussed.

First, the most important aspect to mention is that the code is implemented using *PyBNesian* (Atienza et al. (2022a)). This comes with some advantages and disadvantages, but in general it was the best option among the different libraries that exist to work with BNs, since each one of them is focused on particular aspects of this model. The main advantage that this library presents is that every type of BNs with regard to the CPD is implemented, with the exception of augmented networks (discrete nodes with continuous parents). This is a great advantage since it allows hybrid networks (needed if we are clustering continuous features) and also implements KDE approach for learning CPDs, which allows for the extension of this methodology to any type of distribution for future research.

On the other hand, as it happens with the rest of the libraries, *PyBNesian* is designed to only learn and represent the model for complete data. This has led to a lack of functionality towards implementing inference and unsupervised learning with it. For example there is no capability for being able to introduce desired CPDs parameters, which is needed. This has had a huge impact on the implementation developed here as we next show. Nonetheless, it still remains the best framework to develop the ideas of this work.

We first start with structure learning, since this functionality was not implemented with latent variables. We have restricted the implementation to TAN structures due to the fact that $P(H \mid \mathbf{X})$ can be easily computed and this is important for the Monte Carlo EM (MCEM), which is the variant of the EM implemented. We have used MCEM since we are not able to manually introduce CPDs in the model, which implies that CPDs must be learnt directly from data and MCEM is the only variant that allows

[3] https://github.com/Victor-Alejandre/Bayesian-Network-based-clustering.git

this, under the assumption that we can sample from $P(H \mid \mathbf{X})$. With respect to the score the expected log likelihood has been used with the penalization introduced by BIC.

Next, for the search procedure, a first best random search has been implemented with the operators: arc introduction, arc removal and arc direction change. The search is sequential. First, we check for arc introduction, then arc removal and at the end arc direction change. If the score improves at any stage we keep the BN with the change and go to the next stage. It is worth mentioning that in this case the number of clusters is considered to be known; nonetheless, another stage of clustering number changing could be introduced for automatic detection. Moreover, since prior information of parameters (Bayesian estimation) cannot be introduced and the library does not contemplate it, at each iteration completion of the total dataset is done to avoid zero probabilities. This must be taken into account since we are introducing bias.

Finally, for MAP computation we have implemented probabilistic logic sampling (Henrion (1988)). Moreover, for importance feature measuring, the inference is made by means of likelihood weighting (Fung and Chang (1990)). This is due to the fact that CPDs cannot be manually introduced, hence AIS-BN and EPIS-BN (Cheng and Druzdzel (2000) and Yuan and Druzdzel (2012)) cannot be implemented. Another option would be introducing exact inference but it has higher computational cost. Next, we show the results for the dataset mentioned before.

5.1.2 Customers dataset

The customers dataset (Sharma (2021)) is a hybrid dataset that contains information about 2000 individuals of a particular, anonymized fast-moving consumer goods (FMCG) store. Particularly we find the following features:

- *Sex*, a two-level factor with levels 0 (male) and 1 (female).
- *Marital status*, a two-level factor with levels 0 (single) and 1 (non-single: divorced/ married/ widowed).
- *Age*, a continuous feature with minimum value 18 years old and maximum value 76. Since we are dealing with categorical data this feature has been discretized into three factors with levels junior, adult and senior citizen.
- *Education*, a four-level factor with levels 0 (other/unknown), 1 (high school), 2 (university) and 3 (graduate school).
- *Income*, a continuous feature with minimum value 35832 and maximum value 309364. Since we are dealing with categorical data this feature has been discretized into a three-factor variable with levels low-income, medium-income and high-income according to the benchmark established by the Pew Research Center in 2020.
- *Occupation*, a three-level factor with levels 0 (unemployed/unskilled), 1 (skilled employee/official) and 2 (management/self-employed/high qualified employee/officer).

- *Settlement size* (size of the city customer lives in), a three-level factor with levels 0 (small), 1 (mid-sized) and 2 (big).

We have learnt the structure for a predefined value of $K = 2$ and applied the methodology proposed. The results obtained are presented in Table 5.1. We also find Figure 5.2 where the results are combined with a visual display (radar chart) in order to ease the comparisons and comprehension.

Representatives					
Cluster c1			**Cluster c2**		
Feature	**Value**	**Importance**	**Feature**	**Value**	**Importance**
Marital status	1	0.092	Sex	0	0.2422
Sex	1	0.0404	Marital status	0	0.1148
Age	Adult	0.0398	Education	1	0.0386
Education	1	0.0146	Occupation	0	0.0135
Settlement size	0	0.0121	Settlement size	0	0.0122
Occupation	1	0.0064	Income	Middle income	0.0098
Income	Middle income	0.0059	Age	Adult	0.0077

Table 5.1: Example for the methodology proposed for cluster characterization for customers dataset.

Firstly, we notice that for cluster $c1$ all features have low importance values. This implies that if any of the features were to take another value then the posterior probability of H would be quite similar to $P(H|MAP(\mathbf{X} \mid h_1))$. Therefore, our criteria for elucidating a name for the cluster has been focusing on features with the highest relative importance values which are Marital status, Sex and Age. Marital status and Sex take the value 1 whereas in cluster $c2$ these features take 0. Thus, there exists a difference between clusters. In the case of Age, we have that the representative of both clusters take the value $Adult$. But, in the case of cluster $c2$ this feature has a really low importance which implies that the value $Adult$ characterizes the cluster $c1$. From this analysis we have baptized the cluster $c1$ as *Adult non-single females*.

Following the criteria used for cluster $c1$ in cluster $c2$ we focus on features Sex and Marital status. As stated before, these features take value 0 different from the values that they take for cluster $c1$, thus, they do characterize the cluster. In this case we have baptized the cluster $c2$ as *Single males*.

To sum up, we have that the clients of the FMCG store are divided into two groups which are *Adult non-single females* and *Single males*. Before ending this example we would like to mention two details. Firstly, notice that the feature Occupation takes different values for each cluster. If we had not the importance measure one could be tempted to use this feature to discern and characterize the clusters for the naming. As we can see in both cases this feature has a really low importance which means that if this feature takes another value the posterior probability of H is quite similar to the one obtained with the representative. This has led us to ignore the feature for

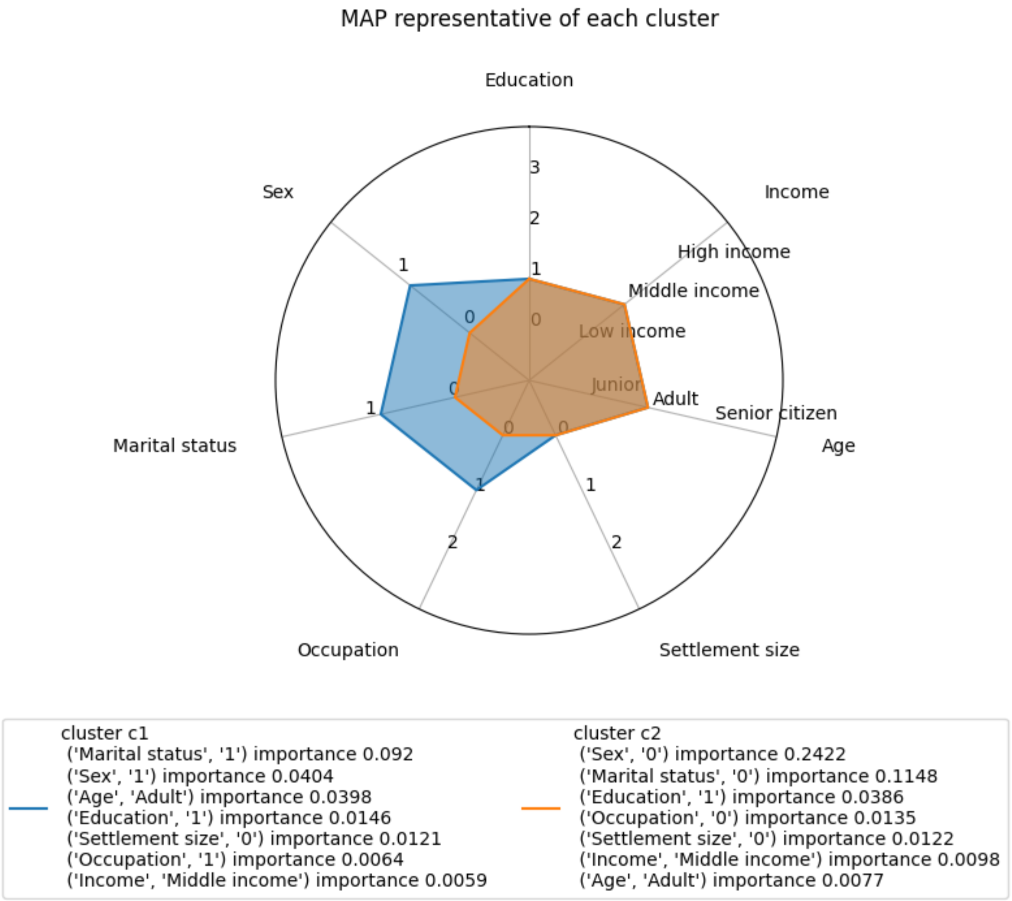


Figure 5.2: Radar chart example for the customers dataset.

solving the naming problem.

Moreover, in Figure 5.3 we have projected the features Sex, Marital status and Occupation for each possible individual of the customer dataset with the fixed values $Adult, 0, Middle\ income, 1$ for features Age, Settlement size, Income and Education. Each point has been colored according to the cluster they belong, which has been selected through map estimation. As we can see, once fixed the values of Marital status and Sex for each one of the cluster representatives ($single$, $male$ and $non - single$, $female$), then the feature Occupation does not affect on the belonging of the individual to the clusters. Notice that this is captured in a soft way by the importance feature measure proposed in this work. Furthermore, we can see that feature Occupation does not affect at all to which cluster belongs an individual given the fixed values $Adult, 0, Middle\ income, 1$.

Secondly, this methodology can be combined with other explanation methods or information in order to dive deeper into the matter. For example we can modify the radar chart from Figure 5.2 to include more information, such as the plausibility of the values taken by the representatives. This is what we find in Figure 5.4 where $P(MAP(\mathbf{X} \mid h_j)_i \mid h_j)$ for each $j = 1, 2$ and $i = 1, ..., p$ is shown with the respective representatives and importances in the legend.

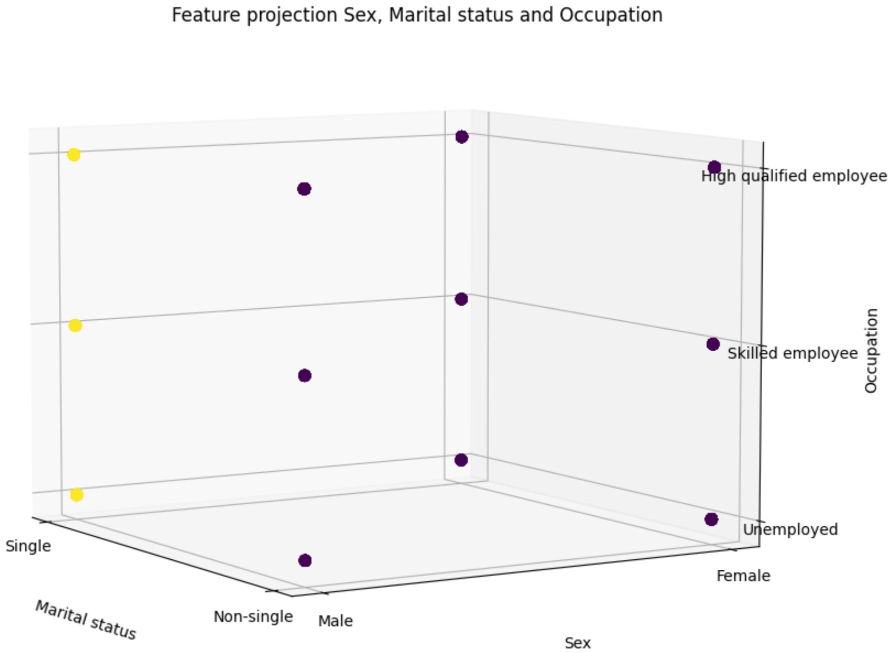


Figure 5.3: Individuals cluster belonging through map estimation for the projection of Sex, Marital status and Occupation given the fixed values $Adult, 0, Middle\ income, 1$ for features Age, Settlement size, Income and Education.

Another example is the combination of the information given by the representatives and the feature importance measure, with a global model explanation method like the representation of the network. This can be useful as we are at the same time watching what characterizes the clusters and the global information of the BN about the dependencies of the variables. Following we show an example for this case in Figure 5.5.

Finally, notice that quality measure of the clusters is not performed since, as mentioned before, this relates to the clustering process and here we are trying to show the methodology proposed. Nonetheless, it is worth mentioning that this can compromise the analysis and comparisons, since the model could not be obtaining good clusters with respect to the notion of quality selected.

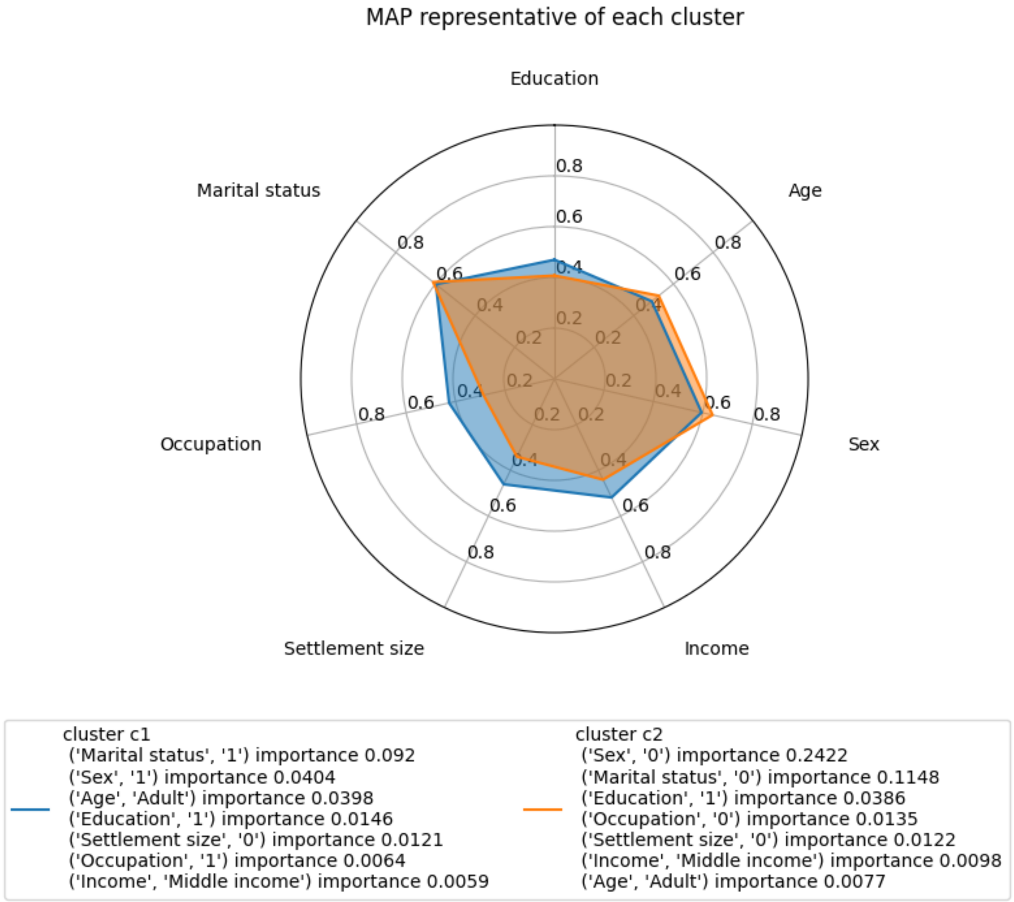


Figure 5.4: Modified radar chart example for the customers dataset.

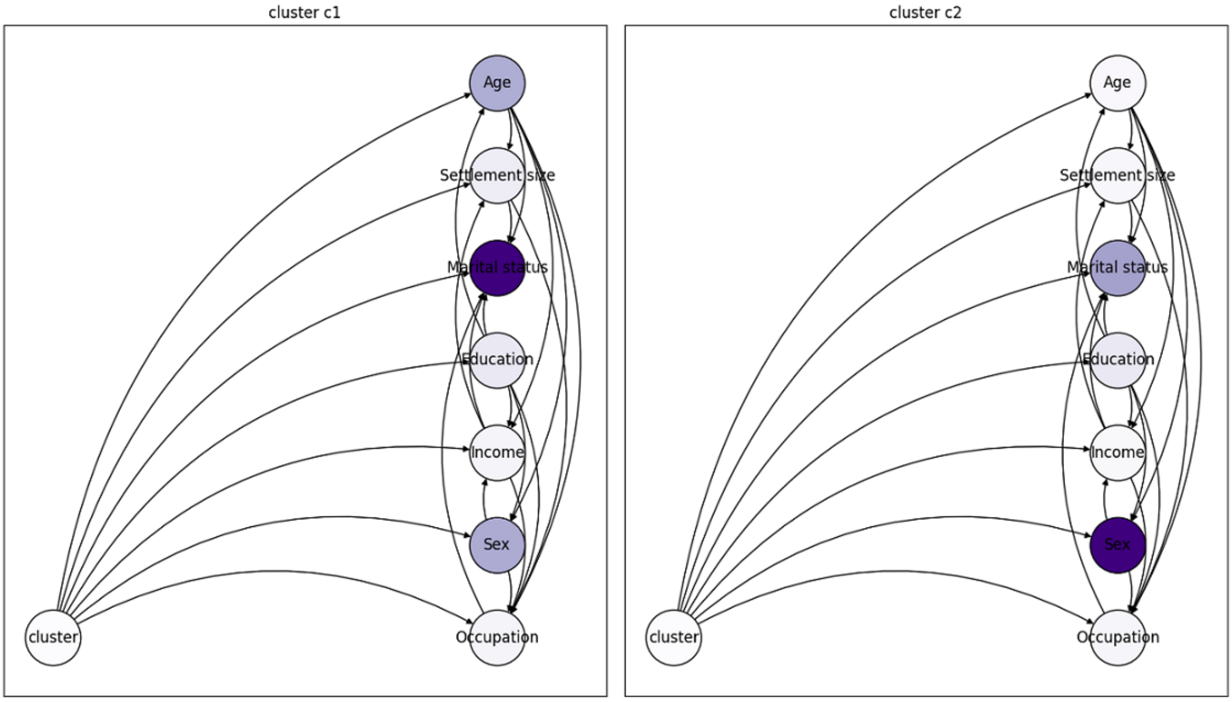


Figure 5.5: Example of combination of BN representation for each cluster and feature importance measure for the customers dataset. Node's color intensity indicates the importance of the feature for each representative of each cluster.

Chapter 6

Conclusions and future research

We present the conclusions reached along this work and the future research lines that it opens.

6.1 Conclusions

In Chapter 1 we have seen that explanations in AI is an area still under development, which has no ground and general theory to define and measure quality of explanations. The main cause here is the difficulty to measure objectively something that is subjective and context dependent. Nonetheless, we review several works which present useful ideas and leave a hope for objectiveness. We stand out the work of Doshi-Velez and Kim (2017), which sets insights into the human dependent evaluation of explanations and provides an "objective" approach.

From this Chapter we have also concluded that the most important aspect when developing methodologies for obtaining explanations is to state how they are obtained and which interpretability factors related to how the explanation is obtained are taken into account. However, in some contexts quality measures and properties for the explanation can be defined and studied, but there is no guarantee that this reflects reality.

In Chapter 2 BNs are reviewed. Also the usefulness of this model with regard to the framework of ML and explanations has been shown. From this Chapter we can conclude that BNs are a versatile, interpretable model that allows for several tools for generating explanations due to the probabilistic knowledge embedded into it. It has also the advantage of dealing with uncertainty, which most of the ML models do not have. However, this is a double-edged sword since receivers of explanations must have probability knowledge. This usefulness has been exemplified as we have shown the advantages a BN can introduce when used in some of the most relevant *post-hoc* methodologies for obtaining explanations in Section 2.3.

In Chapter 3 we have shown that the clustering problem which belongs to the unsupervised learning framework is a complex iterative process due to the lack of prior information about the problem, which implies no general quality measure of the results. As a consequence, this rises the need of explanations and since we do not have prior information about the results to compare with, interpretability is sought in order to understand what the model does and comprehend the results. One of the goals of understanding results is clustering naming, which is the problem dealt with later in Chapter 5.

In Chapter 4 the main conclusion is that BNs can deal with incomplete data in terms of structure and parameter learning, which implies that the clustering problem can be modeled with them. As a consequence, BNs can be potentially useful in clustering since they are an interpretable model with several tools (Chapter 2) for generating explanations and obtaining results interpretation.

In Chapter 5 a methodology based on the usage of BN for clustering characterization in order to solve the clustering naming problem is proposed. This methodology entails an interpretable and natural way to characterize clusters in order to compare and understand them, showing the potential of BNs within the clustering problem. The most remarkable characteristic of the methodology is the defined feature importance measure for a given instance. As it is out of reach for this work, the evaluation of the quality of clusters and explanations have not been dealt with. Hence, assertions on the usefulness of BN cannot be presented. Nevertheless, the ideas expressed show potential for solving the clustering naming problem.

6.2 Future research

The first future research line is the study of the quality of the explanations given by the methodology proposed in Chapter 5. Quality should be evaluated in order to determine if that is a correct way to characterize clusters and is useful for cluster naming. Ideally, this would be determined with human expert evaluation in particular contexts and clustering problems. This can also be approached with the current literature by studying if the properties to determine quality of explanations are suitable/make sense to measure the quality of the methodology. Also quality explanation measures could be used if they are adequate for the context. Moreover, the feature importance measure for the MAP representative should be evaluated separately since it is likely to be also appropriate for other problems. This is due to the fact that this measure is a methodology that finds local explanations that can be used with any given instance of interest (not only MAP representatives) in any type of problem.

Another interesting line of future research is trying to expand the application of BNs explanations tools to the clustering problem in order to show the usefulness of the model in this context. As shown in Chapter 2, there is a large toolbox of methods that can be used to generate explanations starting from posterior probability plots, to visualize uncertainty on the target variable, to more sophisticated methods such as MAP-independence or influence-driven.

A particular line would be studying the usage of MAP-independence and influence-driven in multipartition clustering where more than one latent (clusters) variable exists. For example MAP-independence can be used to determine which latent variables affect a latent variable of interest, given an evidence e. This gives a set of the different forms of grouping the population (latent variables) that affect the grouping made with respect to the latent variable of interest.

Finally, a third line of future research would be studying the feasibility of the methodology presented for different structures and types of data, since here we have only dealt with TAN structures and discrete data. In order to be able to apply the ideas presented in here, two main issues must be solved:

- First we have to guarantee that the network structure and parameters can be learnt in the presence of latent variables to model the cluster problem. As seen in Chapter 4, in order to adapt/apply Bayesian-SEM algorithm to any structure we need a way to obtain the CPDs parameters (mainly EM and variants) and the score function for the search to be in a closed and decomposable form to be able to compute it. For both things we mainly work with expectations with respect to the latent variable conditioned distribution; thus it is key to be able to compute $P(H \mid \mathrm{x}, G)$ given an instance x of the dataset D and a structure G. Nevertheless, if we are not able to compute this posterior probability, importance sampling can be employed, though this has its drawbacks as for example a low convergence rate.
- Second, we have to be able to make inference in the BN in order to compute feature importance and MAPs. Inference is a highly computationally expensive problem, thus the main problem here is to achieve good results in a reasonable time. Furthermore, we may also find that for some structures and variable types, some posteriors cannot be computed.

With all of this in mind we can see for example that generalizing the methodology for any structure with discrete features is quite easy. For continuous normal features with the TAN structure, the methodology can be applied and effortlessly implemented, also with hybrid data with normal continuous features (CLGBN).

In addition, with restrictions to the TAN structure with continuous non-normal features with a non-parametric approach for the CPDs (CKDE), the methodology can be implemented. Here for the structure learning generalized-EM plus Monte Carlo-EM must be used, which take advantage of the fact that $P(H \mid \mathrm{x}, G)$ can be easily computed due to the TAN structure. For the expectations used in the feature importance measure, approximation methods must be used such as importance sampling, since exact inference cannot be performed.

Lastly, the framework with no structure restrictions and continuous variables is the most difficult case to translate the ideas into, since we are working with augmented networks (softmax CPDs for discrete features with continuous parents) and it is not trivial how to make inference.

Bibliography

Akula, A. R. and Zhu, S.-C. (2022). Attention cannot be an explanation. *arXiv preprint arXiv:2201.11194*.

Albini, E., Rago, A., Baroni, P., and Toni, F. (2021). Influence-driven explanations for Bayesian network classifiers. In *The Proceedings of the 18th Pacific Rim International Conference on Artificial Intelligence, Part I 18*, pages 88–100. Springer.

Anderson, R. (2016). The Rashomon effect and communication. *Canadian Journal of Communication*, 41(2):249–270.

Atienza, D., Bielza, C., and Larrañaga, P. (2022a). PyBNesian: An extensible Python package for Bayesian networks. *Neurocomputing*, 504:204–209.

Atienza, D., Bielza, C., and Larrañaga, P. (2022b). Semiparametric Bayesian networks. *Information Sciences*, 584:564–582.

Atienza, D., Larrañaga, P., and Bielza, C. (2022c). Hybrid semiparametric Bayesian networks. *TEST*, 31(2):299–327.

Balabaeva, K. and Kovalchuk, S. (2020). Post-hoc interpretation of clinical pathways clustering using Bayesian inference. *Procedia Computer Science*, 178:264–273.

Barkouki, T., Deng, Z., Karasinski, J., Kong, Z., and Robinson, S. (2023). Xai design goals and evaluation metrics for space exploration: A survey of human spaceflight domain experts. In *Proceedings of the AIAA Science and Technology Forum 2023*, page 1828.

Barnard, P., Macaluso, I., Marchetti, N., and DaSilva, L. A. (2022). Resource reservation in sliced networks: An explainable artificial intelligence (XAI) approach. In *ICC 2022-IEEE International Conference on Communications*, pages 1530–1535. IEEE.

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., and Herrera, F. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115.

Belaid, M. K., Hüllermeier, E., Rabus, M., and Krestel, R. (2022). Compare-XAI: Toward unifying functional testing methods for post-hoc XAI algorithms into an interactive and multi-dimensional benchmark. *arXiv preprint arXiv:2207.14160v2*.

Ben-Hur, A., Horn, D., Siegelmann, H. T., and Vapnik, V. (2001). Support vector clustering. *Journal of machine learning research*, 2:125–137.

Bertsimas, D., Orfanoudaki, A., and Wiberg, H. (2021). Interpretable clustering: An optimization approach. *Machine Learning*, 110:89–138.

Bezdek, J. C. (1973). *Fuzzy-mathematics in pattern classification.* Cornell University.

Biessmann, F. and Refiano, D. (2021). Quality metrics for transparent machine learning with and without humans in the loop are not correlated. *arXiv preprint arXiv:2107.02033.*

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

Burkart, N. and Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317.

Celeux, G. and Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics & Data Analysis*, 14(3):315–332.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793.

Cheeseman, P. C. and Stutz, J. C. (1996). Bayesian classification (Autoclass): Theory and results. *Advances in Knowledge Discovery and Data Mining*, 180:153–180.

Chen, J. (2018). *Interpretable Clustering Methods.* PhD thesis, Northeastern University.

Cheng, J. and Druzdzel, M. J. (2000). AIS – BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research*, 13:155–188.

Chickering, D. M. (2002). Learning equivalence classes of bayesian-network structures. *The Journal of Machine Learning Research*, 2:445–498.

Chow, C. and Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467.

Cooper, G. F. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347.

Coroama, L. and Groza, A. (2022). Evaluation metrics in explainable artificial intelligence (XAI). In *International Conference on Advanced Research in Technologies, Information, Innovation and Sustainability*, pages 401–413. Springer.

Čyras, K., Rago, A., Albini, E., Baroni, P., and Toni, F. (2021). Argumentative XAI: A survey. *arXiv preprint arXiv:2105.11266.*

Darwiche, A. (2009). *Modeling and Reasoning with Bayesian Networks.* Cambridge University Press.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Derks, I. P. and De Waal, A. (2020). A taxonomy of explainable Bayesian networks. In *Artificial Intelligence Research: the First Southern African Conference for AI Research*, pages 220–235. Springer.

Donath, W. E. and Hoffman, A. J. (1973). Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Dronov, S. V. and Evdokimov, E. A. (2018). Post-hoc cluster analysis of connection between the forming characteristics. *Model Assisted Statistics and Applications*, 13(2):183–195.

Druzdzel, M. J. and Henrion, M. (1993). Efficient reasoning in qualitative probabilistic networks. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 548–553.

Elsaesser, C. and Henrion, M. (1990). Verbal expressions for probability updates. How much more probable is “much more probable”? In *Machine Intelligence and Pattern Recognition*, volume 10, pages 319–328. Elsevier.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the International Conference on Knowledge Discovery & Data Mining*, volume 96, pages 226–231.

Fraiman, R., Ghattas, B., and Svarc, M. (2013). Interpretable clustering using unsupervised binary trees. *Advances in Data Analysis and Classification*, 7:125–145.

Fraley, C. and Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Friedman, N. (1998). The Bayesian structural EM algorithm. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, UAI’98, page 129–138. Morgan Kaufmann Publishers Inc.

Frost, N., Moshkovitz, M., and Rashtchian, C. (2020). ExKMC: Expanding explainable k-means clustering. *arXiv preprint arXiv:2006.02399*.

Fung, R. and Chang, K.-C. (1990). Weighing and integrating evidence for stochastic simulation in Bayesian networks. In *Machine Intelligence and Pattern Recognition*, volume 10, pages 209–219. Elsevier.

Ghattas, B., Michel, P., and Boyer, L. (2017). Clustering nominal data using unsupervised binary decision trees: Comparisons with the state of the art methods. *Pattern Recognition*, 67:177–185.

Ghosal, A., Nandy, A., Das, A. K., Goswami, S., and Panday, M. (2020). A short review on different clustering techniques and their applications. *In Proceedings of International Conference on Emerging Technology in Modelling and Graphics 2018*, pages 69–83.

Good, I. (1977). Explicativity: A mathematical theory of explanation with statistical applications. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 354(1678):303–330.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, 51(5):1–42.

Heckerman, D. (2019). Probabilistic similarity networks. *CoRR*, abs/1911.06263.

Henrion, M. (1988). Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In *Machine Intelligence and Pattern Recognition*, volume 5, pages 149–163. Elsevier.

Hoffman, R. R., Mueller, S. T., Klein, G., and Litman, J. (2018). Metrics for explainable AI: Challenges and prospects. *arXiv preprint arXiv:1812.04608*.

Hooker, G., Mentch, L., and Zhou, S. (2021). Unrestricted permutation forces extrapolation: Variable importance requires at least one more model, or there is no free variable importance. *Statistics and Computing*, 31:1–16.

Hsiao, J. H.-W., Ngai, H. H. T., Qiu, L., Yang, Y., and Cao, C. C. (2021). Roadmap of designing cognitive metrics for explainable artificial intelligence (XAI). *arXiv preprint arXiv:2108.01737*.

Hsieh, C.-H., Yan, C.-H., Mao, C.-H., Lai, C.-P., and Leu, J.-S. (2016). GMiner: Rule-based fuzzy clustering for Google drive behavioral type mining. In *2016 International Computer Symposium*, pages 98–103. IEEE.

Jeffreys, H. (1998). *The Theory of Probability*. OuP Oxford.

Jiao, L., Yang, H., Liu, Z.-G., and Pan, Q. (2022). Interpretable fuzzy clustering using unsupervised fuzzy decision trees. *Information Sciences*, 611:540–563.

Keivani, O. and Peña, J. M. (2016). Uni-and multi-dimensional clustering via Bayesian networks. In *Unsupervised Learning Algorithms*, pages 163–192. Springer.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press.

Koopman, T. and Renooij, S. (2021). Persuasive contrastive explanations for Bayesian networks. In *Proceedings of the 16th European Conference, Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 229–242. Springer.

Kumar, I. E., Venkatasubramanian, S., Scheidegger, C., and Friedler, S. (2020). Problems with Shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pages 5491–5500. PMLR.

Kwisthout, J. (2021). Explainable AI using MAP-independence. In *Proceedings of the 16th European Conference, Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 243–254. Springer.

Lacave, C. and Díez, F. J. (2002). A review of explanation methods for Bayesian networks. *The Knowledge Engineering Review*, 17(2):107–127.

Lacave, C., Luque, M., and Díez, F. J. (2007). Explanation of Bayesian networks and influence diagrams in Elvira. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(4):952–965.

Lakkaraju, H., Bach, S. H., and Leskovec, J. (2016). Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1675–1684.

Langseth, H., Nielsen, T. D., Rumí, R., and Salmerón, A. (2012). Mixtures of truncated basis functions. *International Journal of Approximate Reasoning*, 53(2):212–227.

Larrañaga, P., Kuijpers, C. M., Murga, R. H., and Yurramendi, Y. (1996). Learning Bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics-Part A (Systems and Humans)*, 26(4):487–493.

Lauritzen, S. L. (1995). The EM algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis*, 19(2):191–201.

Levine, R. A. and Casella, G. (2001). Implementations of the Monte Carlo EM algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439.

Li, Y., Zhou, J., Verma, S., and Chen, F. (2022). A survey of explainable graph neural networks: Taxonomy and evaluation metrics. *arXiv preprint arXiv:2207.12599*.

Lipton, P. (1990). Contrastive explanation. *Royal Institute of Philosophy Supplements*, 27:247–266.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Liu, B., Xia, Y., and Yu, P. S. (2000). Clustering through decision tree construction. In *Proceedings of the 9th International Conference on Information and Knowledge Management*, pages 20–29.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30:4765–4774.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Madhulatha, T. S. (2012). An overview on clustering methods. *CoRR*, abs/1205.1117.

Mansoori, E. G. (2011). FRBC: A fuzzy rule-based clustering algorithm. *IEEE Transactions on Fuzzy Systems*, 19(5):960–971.

Mansoori, E. G., Zolghadri, M. J., and Katebi, S. D. (2008). SGERD: A steady-state genetic algorithm for extracting fuzzy classification rules from data. *IEEE Transactions on Fuzzy Systems*, 16(4):1061–1071.

Masmoudi, K. and Masmoudi, A. (2019). A new class of continuous Bayesian networks. *International Journal of Approximate Reasoning*, 109:125–138.

Meekes, M., Renooij, S., and van der Gaag, L. C. (2015). Relevance of evidence in Bayesian networks. In *Proceedings of the 13th European Conference, Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 366–375. Springer.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Mittelstadt, B., Russell, C., and Wachter, S. (2019). Explaining explanations in AI. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 279–288.

Molnar, C. (2020). *Interpretable Machine Learning*. Lulu.com.

Moral, S., Rumí, R., and Salmerón, A. (2001). Mixtures of truncated exponentials in hybrid Bayesian networks. In *Proceedings of the 6th European Conference, Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 156–167. Springer.

Moshkovitz, M., Dasgupta, S., Rashtchian, C., and Frost, N. (2020). Explainable k-means and k-medians clustering. In *International Conference on Machine Learning*, pages 7055–7065. PMLR.

Muñoz, C., da Costa, K., Modenesi, B., and Koshiyama, A. (2023). Local and global explainability metrics for machine learning predictions. *arXiv preprint arXiv:2302.12094*.

Nielsen, F. (2016). *Hierarchical Clustering*, pages 195–211. Springer International Publishing, Cham.

Palacio, S., Lucieri, A., Munir, M., Ahmed, S., Hees, J., and Dengel, A. (2021). XAI handbook: Towards a unified framework for explainable AI. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3766–3775.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Pelleg, D. and Moore, A. (2001). Mixtures of rectangles: Interpretable soft clustering. In *Proceedings of the 18th International Conference on Machine Learning*, pages 401–408. Morgan Kaufmann.

Peña, J. M., Lozano, J. A., and Larrañaga, P. (1999). Learning Bayesian networks for clustering by means of constructive induction. *Pattern Recognition Letters*, 20(11-13):1219–1230.

Peña, J. M., Lozano, J. A., and Larrañaga, P. (2000). An improved Bayesian structural EM algorithm for learning Bayesian networks for clustering. *Pattern Recognition Letters*, 21(8):779–786.

Peña, J. M., Lozano, J. A., and Larrañaga, P. (2002). Learning recursive Bayesian multinets for data clustering by means of constructive induction. *Machine Learning*, 47:63–89.

Peña, J. M., Lozano, J. A., and Larrañaga, P. (2004). Unsupervised learning of Bayesian networks via estimation of distribution algorithms: An application to gene expression data clustering. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(supp01):63–82.

Pham, D. T. and Ruz, G. A. (2009). Unsupervised training of Bayesian networks for data clustering. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2109):2927–2948.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " 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, pages 1135–1144.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Rosenfeld, A. (2021). Better metrics for evaluating explainable artificial intelligence. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, pages 45–50.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., and Zhong, C. (2022). Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistic Surveys*, 16:1–85.

Saisubramanian, S., Galhotra, S., and Zilberstein, S. (2020). Balancing the trade-off between clustering value and interpretability. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 351–357.

Santafé, G., Lozano, J. A., and Larrañaga, P. (2006a). Bayesian model averaging of naive Bayes for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(5):1149–1161.

Santafé, G., Lozano, J. A., and Larrañaga, P. (2006b). Bayesian model averaging of TAN models for clustering. In *Proceedings of the Third European Workshop on Probabilistic Graphical Models*, pages 271–278.

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., Er, M. J., Ding, W., and Lin, C.-T. (2017). A review of clustering techniques and developments. *Neurocomputing*, 267:664–681.

Scutari, M. (2010). Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22.

Shachter, R. D. (2013). Bayes-ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). *arXiv preprint arXiv:1301.7412*.

Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games (AM-28), Volume II*, pages 307–317.

Sharma, D. (2021). *Customer Clustering, Version 1. Retrieved July, 2023 from https://www.kaggle.com/datasets/dev0914sharma/customer-clustering.*

Shenoy, P. P. and West, J. C. (2011). Inference in hybrid Bayesian networks using mixtures of polynomials. *International Journal of Approximate Reasoning*, 52(5):641–657.

Silva, A., Schrum, M., Hedlund-Botti, E., Gopalan, N., and Gombolay, M. (2023). Explainable artificial intelligence: Evaluating the objective and subjective impacts of XAI on human-agent interaction. *International Journal of Human–Computer Interaction*, 39(7):1390–1404.

Singh, V., Cyras, K., and Inam, R. (2022). Explainability metrics and properties for counterfactual explanation methods. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 155–172. Springer.

Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, Prediction, and Search*. The MIT Press.

Sucar, L. E., Bielza, C., Morales, E. F., Hernandez-Leal, P., Zaragoza, J. H., and Larrañaga, P. (2014). Multi-label classification with Bayesian network-based chain classifiers. *Pattern Recognition Letters*, 41:14–22.

Suermondt, H. J. (1992). *Explanation in Bayesian Belief Networks*. Ph.D. thesis, Department of Computer Science, Stanford University, CA STAN-CS–92–1417.

Suermondt, H. J. and Cooper, G. F. (1993). An evaluation of explanations of probabilistic inference. *Computers and Biomedical Research*, 26(3):242–254.

Sundararajan, M. and Najmi, A. (2020). The many shapley values for model explanation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 9269–9278. PMLR.

Thiesson, B. (2013). Score and information for recursive exponential models with incomplete data. *arXiv preprint arXiv:1302.1571*.

Timmer, S. T., Meyer, J.-J. C., Prakken, H., Renooij, S., and Verheij, B. (2017). A two-phase method for extracting explanatory arguments from Bayesian networks. *International Journal of Approximate Reasoning*, 80:475–494.

Valero-Leal, E., Bielza, C., Larrañaga, P., and Renooij, S. (2023). Efficient search for relevance explanations using map-independence in bayesian networks. *International Journal of Approximate Reasoning*, 160:108965.

Wachter, S., Mittelstadt, B., and Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the general data protection regulation. *Harvard Journal of Law & Technology.*, 31:841.

Wang, X., Liu, X., and Zhang, L. (2014). A rapid fuzzy rule clustering method based on granular computing. *Applied Soft Computing*, 24:534–542.

Wellman, M. P. (1990a). Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44(3):257–303.

Wellman, M. P. (1990b). Graphical inference in qualitative probabilistic networks. *Networks*, 20(5):687–701.

Wu, C. F. J. (1983). On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95 – 103.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678.

Yang, H., Jiao, L., and Pan, Q. (2021). A survey on interpretable clustering. In *2021 40th Chinese Control Conference*, pages 7384–7388. IEEE.

Yuan, C. and Druzdzel, M. J. (2012). An importance sampling algorithm based on evidence pre-propagation. *arXiv preprint arXiv:1212.2507*.

Yuan, C., Lim, H., and Lu, T. (2011). Most relevant explanation in Bayesian networks. *Journal of Artificial Intelligence Research*, 42:309–352.

Zhang, T., Ramakrishnan, R., and Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. *ACM Sigmod Record*, 25(2):103–114.