Incorporating Domain Experts' Knowledge into Machine Learning for Enhancing Reliability to Human Users

LI JIARUI

Abstract

Machine Learning (ML), as an application of Artificial Intelligence (AI) technology, is widely used in engineering practice. For evaluating the effectiveness of an ML technology, high accuracy and speed always obtain high praise. However, reaching the standard of accuracy and high speed is only the bottom-line requiring ML nowadays. Researchers are going after a higher realm, i.e., the reliability of ML to humans. The human users expect to understand the reasons under the decision made or actions took by an AI production, e.g., a robot, to judge whether they are reliable. Therefore, except for high accuracy and efficiency, ML models or algorithms used by such AI productions ought to be explainable so that human users can verify their reliability.

This thesis proposed that ML should use domain experts' knowledge to regulate or direct the learning procedures. Two types of knowledge are available. One is the knowledge summarized from human experience. The other is causal knowledge. In the first category, the knowledge is specified as the statistical rules summarized from an expert's experience. The experience knowledge can be used as "a priori" knowledge for regulating the learning procedure of an ML model. The expert experience knowledge plays a role of the complement before training the model when the ML model does not have sufficient explainability or the predicting accuracy is low. The causal knowledge involves the inferring logic of a domain expert that can be used for direct ML models to learn to simulate a human-like learning procedure. It is used in the training procedure of an ML model or post-training the model. We showed three novel technologies, including an explainable ML model and two ML algorithms involving expert knowledge as examples.

Acknowledgment

Throughout the writing of this dissertation, I have received a great deal of support and assistance.

Firstly, I would like to thank my supervisor, Professor Tetsuo Sawaragi, who continuously supported me in every activity as a researcher. His insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. It is he who has taught me the way to be a researcher with foresight and an open mind. Without his guidance and persistent help, this dissertation would not have been possible.

I would like to thank all staff and members of Sawaragi laboratory. Especially, Professor Yukio Horiguchi patiently guided every detail in my research and always provided keen comments to convince me of various perspectives of this work. It is he who has taught me the way to be a researcher with deep thinking and keeping pursuing perfection. Thanks to Secretary Mrs. Minato supported me by taking care of all the paperwork in the university. As an international student, I am not good at Japanese. Mrs. Minato patiently helped me with proofreading the Japanese documents. Besides, thanks to Researcher Toru Murase, kindly offered me advice in the regular meeting every month.

Gratitude is extended to the Otsuka Toshimi Scholarship Foundation, which has supported me with the funding for both research and life for two years.

In the end, I take this opportunity to express my profound gratitude to my family and friends both in China and Japan. Their respects and supports are the biggest warmness for me in a foreign country. Especially, thank you, my dear husband. It has been impossible for me to complete this work without his encouragement every day.

Contents

Chapter 1. Introduction

1.1 What is the reliability of Machine Learning?

Machine Learning (ML), as an application of Artificial Intelligence (AI) technology, is widely used nowadays to provide systems the ability to learn from the data automatically. For evaluating the effectiveness of an ML technology, high accuracy always obtains high praise. However, if an ML model's evaluation is an examination, the reaching standard of accuracy will be the cut-off score, which is the bottom line for ML. With the rapid development of the information age, the amount of data shows explosive growth. The fast computation speed is in urgent need as well. However, the essence of ML is to let machines or computers simulate the learning process of humans. Thus, an ML technology should be promoted if it can imitate humans as closely as possible, which requires the ML model to own the human-like inference logic. Thus, except for the high accuracy, high calculation speed, the reliable ML technologies should have explainability.

Why is explainability necessary? According to Samek, Wiegand, & Müller. (2017), first of all, explainable AI can be verified by domain experts. In relating to people's life and property safety, such as healthcare, law, and regulations, the AI model does not conform to common sense is invalid. According to Caruana, Lou et al. (2015), the authors show us an example of opposite conclusions of human experts and ML models in healthcare and conclude that the decisions must be given according to the domain expert's opinion. Secondly, explainability makes the models easily to be optimized. If we learn well about a model's decision procedure, its weakness will be easily found out simultaneously. The explainable models tell us the basis of the machine's thinking, which supplies the channel for judging its right or wrong.

On the other hand, an explainable model's essential function is the system's re-learning ability, which is the key to predict the future. The newest paper published by the Turing Award owner Judea Pearl (2018), discussed that the current machine learning theories are limited. Pearl thinks that machine learning systems operate almost entirely in statistics or blind models, which cannot be used as the basis for strong AI. Furthermore, he gave seven inspirations from causal reasoning. Pearl pointed out that "a human-level AI cannot emerge solely from model-blind learning machines; it requires the symbiotic collaboration of data and models." Thus, the ML model's explainability is the key for achieving strong AI.

Roscher (2020) and his team illustrate that the model's explainability should have three levels: transparency, interpretability, and understandability. The most basic level is that a model is transparent. Transparency requires the relations between data features and labels of a particular ML is shown in front of human eyes. For example, the binary choice structure is transparent of a decision tree (Breiman, Friedman, et al, 1984).

Furthermore, interpretability holds a higher level of requirements for ML models. According to Roscher, interpretability requires that the user can verify the reasonability of the learning process. For instance, Bayesian Networks (BNTs) are typical reasoning models. ML methods, such as Kutato 2 (K2) (Doguc & Ramirez-Marquez, 2009), are used for constructing the BNTs structure by learning from data. In the process of training the structure, K2 requires prior knowledge of the nodes' order, which are given from domain expert.

However, the understandability is at a higher level than the former two because it requires the model to tell the user of the human-understandable explanation science such as law, cognitive science, and causal relations. For example, the dark clouds, rain, and the wet floor are three events correlated closely with each other. Whether using the data of the dark clouds or the wet floor data, an ML model can output the probability of rain. Both of the models are interpretable, but only the clouds-rain model is with understandability if involving causality. The dark clouds cause the rain while the wet floor cannot. ML models with understandability can explain the scientific backgrounds of the inference to the user. In other words, understandability refers to human experts' domain knowledge and pursues that the AI performs more like human beings.

However, many existing explainable ML models are at the expense of accuracy. As shown in Fig.1.1 (Duval, A., 2019), the accuracy and the explainability of machine learning models are shown the anti-dependent relationship.

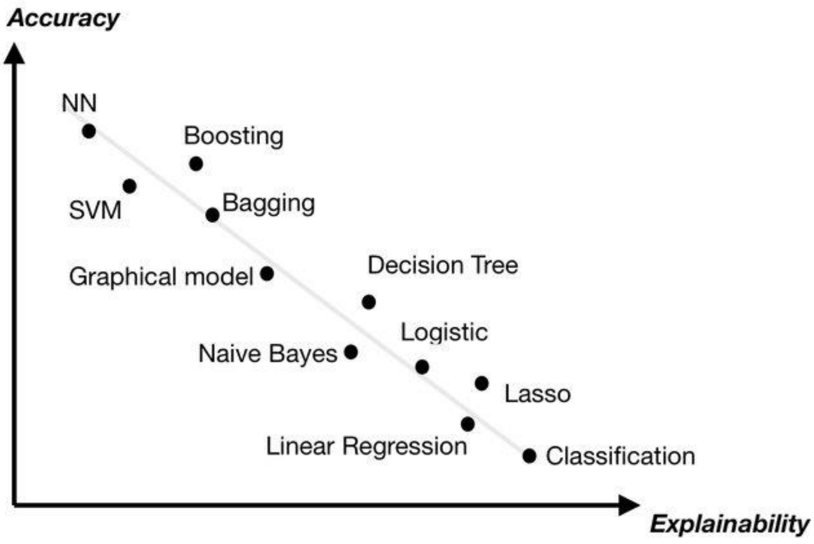


Fig.1.1. Models with high explainability are at the expense of accuracy (Duval, 2019)

Nevertheless, reaching the standard of high prediction accuracy is the bottom-line requiring an ML model. In this thesis, we define the reliability of ML, which involves both the accuracy and the explainability.

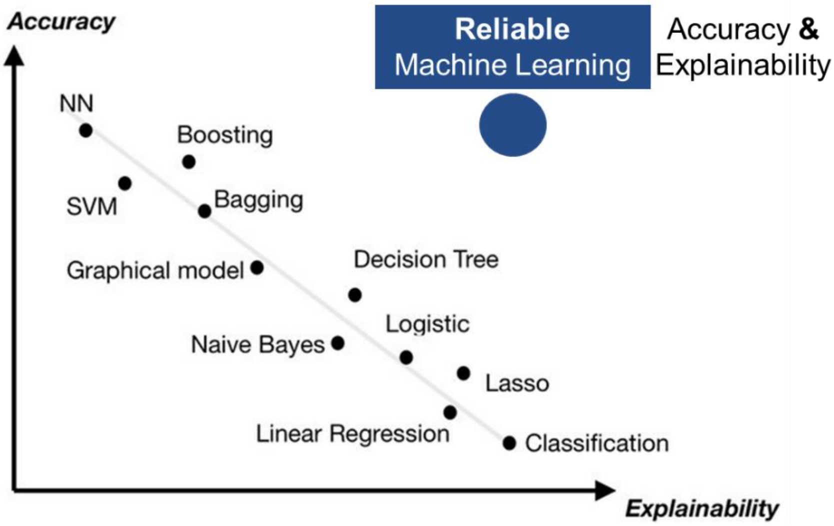


Fig.1.2. The reliability involves both the high accuracy and high explainability

For creating reliable ML models, domain expert knowledge plays an essential role. For one thing, domain experts help ML models by providing knowledge of explaining how or why the model is trained. For the other, involving domain experts' knowledge as "a priori" knowledge can help to enhance the accuracy of the performance. Different kinds of expert knowledge work for reliable Machine Learning in different ways.

1.2 What are useful domain experts' knowledge for reliable Machine Learning?

What expert knowledge is useful for ML, and how to utilize human expert knowledge in ML technologies to enhance their reliability to humans? This thesis proposed that ML should use domain experts' knowledge to regulate or direct the learning procedures and two types of knowledge are available. One is the knowledge summarized from human experience. The other is causal knowledge.

1.2.1 Knowledge summarized from experience

The process of human learning starts from observation. Then, we try to recognize a pattern to build up a relationship between the observed entities. The practical contact and the observation of facts and events is called as experience. Experience is the correct understanding of the world that human beings have acquired through long-term accumulation. Human expert knowledge is the rules and conclusions summarized from experience. The process of machine learning also starts from observation, but what the machine observed is data. Next, relations between entities are built up, and the process of choosing a suitable pattern to express the relations is called "fitting." It seems that there is no difference between human learning and machine learning, and the machine is even smarter. However, in some cases, the current ML models are powerless and make errors, while it is the human experience that can guide ML to correct errors.

The rapidly developmental ML technologies play essential roles in various fields, among which healthcare takes a piece of pie but should be discrimination from others. The extent to which machine learning should be relied on is one of the focuses in the medical field. The accuracy of an ML algorithm depends on the data collected in the past, and the quality of data impacts the learning result to a great degree. If biases or missing data mixes in the collected data, ML will lead user to the wrong direction. For example, Electric Health Recorders (EHR) is the common way of data collection in the modern medicine. Disease prediction can be achieved through learning the large amount of data representing symbols of the diseases using ML algorithms. However, one of the biases existing in the data collected through EHR is that most of the data only considers the unhealthy cases. The physiological data from healthy people is ignored or does not have chance to be collected (Ghassemi, Naumann, *et al.*, 2020). Besides, an ML algorithm makes decisions through a particular criterion, such as distance between data points or density. The criterions are always the inter-characteristic of the collected data while ML is not capable in identifying the bias in the data, i.e., the unbalanced number of sick/healthy examples. The bias in the collected data will lead ML to make incorrect forecast of the illness risk, particularly in the whole population including both healthy and unhealthy cases.

Healthcare problems involve human lives. The mistakes will cause irreversible harm to the individual and to the society. Nevertheless, the utilization of ML saves time and human labor, especially in the age of “big data”. It is a challenging subject for ML to make the greatest contribution to the medical field without making mistakes, for which human expert, e.g., doctors’ experience is sometimes vital for monitoring or improving the ML models, e.g., the knowledge of the ratio of healthy/unhealthy population in a particular age-group. Also, as for the knowledge from experience is what we believe as the absolute truth, the direct utilization will yield twice the result with half the effort and save much time and labor.

1.2.2 Causality

Except for the knowledge summarized from experience, causality is another human expert knowledge that a computer program cannot learn automatically from data.

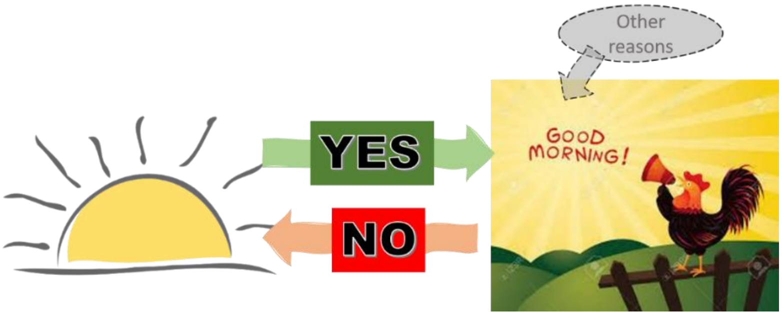
Causality is a complex philosophical concept. Thoroughly speaking, when means an event has a certain effect on the other one, the relationship between the two events is called causality. The event that happened earlier is the reason, and the latter is the result. There is much debate about the specific definition of causality, but it is not the point of our discussion in this thesis. The causality introduced to the field of ML is the so-called interventionism-causality.

In the interventionism-causality theory (Gebharter, 2017), the outlying intervention is the reason, and the corresponding change of the phenomenon is the result. Before introducing intervention, let us retrospect the flaw of the traditional concept of causality. The Scotland philosopher David Hume proposed that most people believe that if one thing always comes with another, there must be a correlation between them: “Post hoc ergo propter hoc,” which is the traditional cognition of causality. Hume defined the concept of “constant conjunction” and confirmed constant conjunction between them when one thing always caused another. However, Hume also thought that the observed constant conjunction could not predict that there would still exist constant conjunction between the two things in the future. The correlation is not causality. A simple example of rooster crowing and sunrise is that two events strongly correlated. However, everyone knows that the sunrise is not because of the rooster crowing. Thus, not all correlations can be expressed as causality, but all two events that have a causal relationship must be correlated. So how can we rise from correlation to the higher-level causal relationship? The intervention gives a hand.

The intervention is an outlying stimulation to a relationship, a model, or a system. With intervention, the status of the intervened object will change correspondingly. Assumes that there is an original status $Y_c(u)$ and an intervention (*T*). The causality can be expressed as $\delta(u) = Y_t(u) - Y_c(u)$. Only by evaluating $\delta(u)$ can the existence of causality be judged. For instance, if we artificially control the rooster keeping silent in the morning, the sun's status will still rise. $\delta(u) = 0$. There is no causality between two events.

Judea Pearl (2018) introduced interventionism-causality to the ML field. Correlation is the basis of ML. Through correlation, computer programs fit functions to express the correlation and make a prediction. Differently, humans learn from experience in the past and extract rules for making a judgment on something. When there is a new stimulation, humans' knowledge systems can be modified timely. The reason why humans can reply to the unknown change is that the causal models are built up. It is very different of causality and correlation, and the former is a higher level of the latter. An example of sunrise and cock singing shown in Fig. 1.3 helps describe why the ML will fail to make a prediction without considering causality.

The rooster singing in the morning and the sunrise are two events that often happen together. They are closely correlated with each other. By only considering the correlation that the ML does, the two events should be able to predict each other. For example, if sunrise is observed, then the ML will predict rooster will sing thereupon. Also, if a rooster sing is observed, the sun will rise correspondingly. However, As shown in Fig. 1.3 (b), if one day a rooster sings at night, would the sun rise at night? The answer is obviously no. Thus, why do our humans not make a mistake in making predictions like sunrise and rooster sing? That is because, besides correlation, humans also know the truth that the rooster singing in the morning is not the reason for sunrise, while sunrise is one of the reasons causing rooster to sing in the morning. In a causal model, the change in the reason will influence the result, while change of the result will not influence the reason. The direction of the arrow between the reason and the result is irreversible. According to a causal model, the corresponding changes in the result can be predicted by observing the changes in the reason. For instance, illuminate the rooster with a light in the evening that imitating the sunrise, the rooster will sing as well because of the stress response. However, even if all the roosters in the world stop singing in the morning, the sun will rise unaffected. The stimulus to the reason, such as illuminating the rooster, is called intervention. By observing the variation influenced by the intervention, a prediction of the change in the result can be made, called a counterfactual inference.



(a) Sunrise causes rooster sings in the morning, but rooster cannot call the sunrise in the morning

(b) Even though a rooster sings at night, the sun will not rise

Fig. 1.3. An example of rooster and sun for explaining interventionism-causality

The interventionism-causality is a useful tool for assisting ML to enhance reliability. For one thing, using human causal thinking to direct ML's learning procedure can endow the machine to own the prediction power like a human being. For the other, the guidance of causality enhances the robustness of ML, avoiding make mistakes like in the sun and rooster case.

This thesis discussed the ways of incorporating domain expert knowledge, especially experience and causality into ML. An overall proposal will be shown in Chapter 2, and Chapters 3-5 show three new technologies as cases.

1.3 Organization of this thesis

The remains of this thesis are organized as follows. The overall idea of this thesis is proposed in Chapter 2, in which the ways of incorporating human experience and causality into ML are illustrated.

Chapter 3 talks about a new clustering algorithm, the Cluster Size Constrained Fuzzy c-Means with Density Center Searching (CSCDFCM). The clustering procedure of CSCDFCM is guided by "a priori" knowledge that is summarized from human experience. The introduction of human knowledge improves the performance of the clustering algorithm when dealing with healthcare problems.

In Chapter 4, a new explainable ML model is described. The new explainable model is proposed based on Structural Equation Modeling (SEM). Data analysis, Machine Learning, and Causal analysis are three functions of the model.

The Relational Feature-Transfer Learning (RF-TL) based on causality is proposed in Chapter 5. It is an application of causality in ML. RF-TL can identify necessary data features in an unknown domain from a known domain, benefiting from the prediction ability of human knowledge, causality.

Conclusions and future expectations are summarized in Chapter 6.

Chapter 2. Incorporating domain experts' knowledge into machine learning-an overview

As mentioned in chapter 1, two kinds of domain expert knowledge are introduced in ML for enhancing the reliability. One is the knowledge extracted from experience, and the other is the causality. They function in different ways. In the first category, the knowledge is specified as the statistical rules summarized from an expert's experience. The experience knowledge can be used as "a priori" knowledge for regulating the learning procedure of an ML model. The expert experience knowledge plays a role of the complement before training the model when the ML model cannot make predictions accurately or does not have sufficient explainability. However, the causal knowledge involves the inferring logic of a domain expert that can be used for direct ML models to learn to simulate a human-like learning procedure. Fig.2.1 shows an overview of the idea of this thesis of how to utilize domain expert's knowledge to enhance the reliability of ML models.

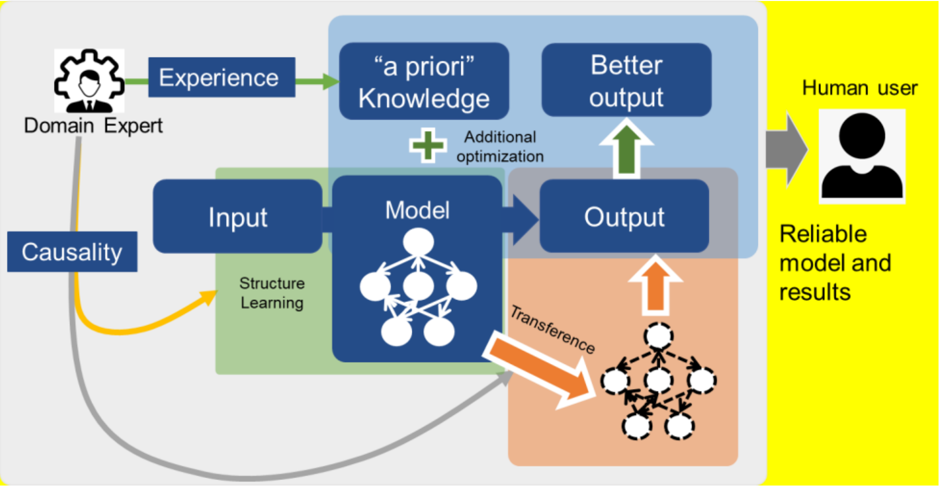


Fig. 2.1. An overview of using domain expert's knowledge to enhance the reliability of ML models to a human user.

2.1 Incorporating knowledge summarized from experience into Machine Learning

Traditional ML methods are classified as the supervised ML and unsupervised ML. Data with labels is the requirement for supervised ML, including data types, data attributes, and feature point locations. These marks are used as expected effects, and the prediction results of the machine are revised continuously. The supervised ML trains the machine through labeled data and compares the predicted result with the label. Furthermore, the machine modifies the model's parameters according to the comparison and repeats the procedure. The training will not stop until convergence. Finally, a particularly robust model is generated to achieve the ability of intelligent decision-making. The supervised ML methods include classification technologies and regression technologies. It is the most basic ML technology.

Nevertheless, not all machine learning problems have enough labeled data for training, spawning the unsupervised learning technologies. Clustering is a standard technology in the unsupervised learning category. It is a method to group data objects into clusters such that objects in the same cluster have similar characteristics, while those in different clusters are disparate (Nayak, Nik & Behera, 2015). Clustering methods can be classified as hard and soft (or fuzzy clustering methods). Methods in the first category, such as k-means, strictly allocate each object to only one cluster, aiming for hard clustering methods requiring clear cluster boundaries. However, in real life, the data objects' natural patterns may belong to single or multiple groups, limiting the hard-type clustering application. Therefore, fuzzy clustering techniques come into being (Banu & Andrews, 2015; Aulik & Bandyopadhyay, 2002). Fuzzy clustering is more flexible that allows every object belongs to more than one cluster. Fuzzy c-Means (FCM) is a widely used partitioning-fuzzy clustering method, which divides a set of data objects into fuzzy partitions by minimizing the intra-cluster variance and maximizing the inter-cluster variance to the objective function (Bezdek, Ehrlich & Full, 1984). The flexibility of acting on the overlapped data set, FCM, and its expansions are fashionable in diversified fields application, such as image processing and market segmentation (Mohamed, Ahmed & Farag, 1999; Ma, Tavares, et al., 2010; Hsu, 2000). However, a common failing of the partition-based clustering method is its inability to deal with varying cluster size datasets and data density. It is also called the "size-insensitive problem." Because of the absence of labels and the various limitations of objective functions, unsupervised ML methods' accuracy is tough to crack.

For overcoming the shortcomings of the traditional ML models, many advanced methods are developed. Especially the highly praised deep learning technologies let the accuracy of ML enhance to a new altitude. However, no matter the former methods or the outstanding deep learning technologies, there is one thing ignored, the human factors in ML. The utilization of human knowledge to direct ML's learning procedure can save much time and enhance accuracy. Although the machine can learn new things at high speed and deal with big data, human beings have accumulated rich knowledge and experience in a long time. It is no doubt that less time is needed to get something ready.

We call the human experience used for improving the performance as the "a priori knowledge." The "a priori knowledge" can be the data population in each cluster or the humanly labeled data. The machine's learning procedure is under the regulation of the knowledge to get better performance.

The blue block in Fig.2.1 shows the utilization of domain expert's experience knowledge for regulating the learning procedure of an ML model. Experiences are the correct understanding of the world that human beings have acquired through long-term observation and summarization. The process of ML also starts from observation, but what the machine observed is the data. Furthermore, models are learned by fitting the data according to the statistical dependence among the data. However, the fitting procedures of ML are not always capable of reflecting the bias in the data. In such a situation, expert knowledge will be useful for regulating the learning procedure of an

ML model. By doing so, the ML's prediction results are more in line with the ground truth, which becomes understandable for the user. Also, by considering a human expert's knowledge, the reasonability of the learning procedure can be explained to an extent. Chapter 3 of the presented studies described a clustering algorithm using a domain expert's experience as prior knowledge to regulate the learning procedure of FCM. It performs well in numerical experiments and practical applications.

2.2 Incorporating causality into Machine Learning

2.2.1 Using causality to design an explainable learning structure

There have been many explainable ML technologies are developed. These technologies can be divided into two groups, output-explainable models and design-explainable models.

The design procedure of an output-explainable model can be the black-box. After getting the trained model, the interpreter is built up for explaining the inference structure or the importance of the used features. The most commonly used "open black box" interpreters are rule extractors. The rule extraction technologies start with the trained complex models and use rule sets to generate interpretable symbol descriptions or models. For example, Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro, Singh & Guestrin, 2019) and Shapley Additive explanation (SHAP) (Tan, Caruana, et al., 2018) are two methods for weighting the feature importance after training a model. Four steps conduct LIME, Random sampling near the prediction sample, label the newly generated sample, calculate the distance between the newly generated sample and the predicted point needed to explain and get the weight, filter the features used for interpretation or fit the linear model. The SHAP method uses game theory to define a Shapley Value. It is the core to weigh data features. Besides, many graphical methods have been developed to explain the model, such as Partial Dependence Plots (PDP) (Greenwell, 2017), Accumulated Local Effects plot (Molnar, 2020), and Individual

Conditional Expectation plot (ICE) (Goldstein, Kapelner, *et al*., 2015). Apart from the explanation to feature weights, some output-explainable methods can explain the model structure as well. However, most of the methods in this category are approximate interpretations or partial interpretations. For example, the Activation Maximization (AM) method (Zhou, Cai, *et al*., 2017) finds the bounded norm's input pattern to maximize the given hidden unit's activation, thereby characterizing the hidden layer of DNN. Similar methods for finding out the meaning and logic of the black box are Sensitivity Analysis (Cortez & Embrechts, 2013), Back Propagation (Grathwohl, Choi, *et al*., 2017), and Feature Inversion (Du, Liu, *et al*., 2018).

The output-explainable methods are easy to be understood. However, this category's biggest problem is that it is also impossible to ascribe causal logic to the output-explainable ML models. The explanations are based on the output of the model. If the input and output are found illogical according to the human understandable theories, there is no way to optimize the model. A better choice is the design-explainable model. The design-explainable methods ask the model is transparent and with a simple model structure. Usually, they are developed based on the tree-type models, Bayesian Networks, or rule extraction models. Many researchers have applied BNT in ML to create explainable models. Constantinou, Fenton & Neil (2012) developed a rigorous and repeatable method for building effective Bayesian network (BNT) models for medical decision support from complex and unstructured data. They stress that their model can be used as an intervention model except for the high accuracy of interference. The intervention model is an important part of understanding and explaining a model. Nevertheless, this paper's whole procedure needs support from domain experts, which costs much time and labor. Other similar works like Keppens (2019) also connecting BNT with other ML methods, such as NNs. As is know that BNT is a kind of probability model, causing its decision procedure relies on probability dependence. In other words, the Bayesian network cannot explain the correlation and causality among training data. In contrast, SEM is a well-known data modeling method expressed by a series of regression functions, which can intuitively describe the relationship between data. The Structural

Causal Model (SCM), as a variant of the SEM, has been used in explaining an ML model. However, to our best knowledge, the published research on SCM or SEM application in the ML field is limited in applying them as an analysis model (Janzing, Rubenstein & Schölkopf, 2018; Holdefer & Skinner, 2020; Neto, 2020).

As mentioned in section 1.1, ML method with explainability requires the model to tell the user of the human-understandable explanation science. In this thesis, we proposed to involve the causal knowledge to post-explain the trained ML model for endowing the model explainability. The "post-explain" illustrated here is different from the "output explain" technologies used for "open the black box." The model that can be post-explained are required with a transparent and interpretable structure, i.e., models in the category of the design-explainable. Moreover, the human expert knowledge is involved after the model is trained and let the model be explainable by the explanation science.

In the green block of Fig.2.1, the procedure of designing the post-explaining ML model by using causality is shown. After the model is trained, the expert's causal knowledge is used for developing the ML model to a causal model, aiming for which an intervention procedure is taken into account. Intervention is one of the steps of interventionism-causality for identifying the causal model from a model described by correlations. In the interventionism-causality theory, the outlying intervention is the reason, and the corresponding change of the phenomenon is the result. The intervention is stimulation to a relationship, a model, or a system. With intervention, the states of the intervened object will change correspondingly. Fitting is the procedure that ML conducts for learning from data. Correlation is the basis that computer programs make a fitting process. Differently, humans learn from experience from the past and extract rules for making a judgment or a prediction. When there is a new stimulation of the changing in humans' knowledge base, the decision-making or prediction system structure will change correspondingly. That is how interventionism-causality works in human cognitive competence.

In Chapter 4, an ML model with interpretable structure was proposed, e.g., SEM-EML. The SEM-EML is constructed by latent factors with the domain expert's defined meaning and the arrows showing causal relationships between factors. Interventions can be conducted on the trained structure of SEM-EML. The output after the intervention is a causal model that can explain the causal inferring procedure to the human user.

2.2.2 Using causality to do transference

Another option of introducing causality into the development of explainable ML model is to use it during the process of designing the learning structure. The orange block in Fig.2.1 shows the procedure of using causal knowledge to direct the learning procedure of Transfer Learning.

Transfer Learning (TL) is an ML technology focusing on training a suitable model for a problem, transferred from the existing models of related problems. In a real-world application, sometimes the large numbers of labeled instances are hard to be collected. For solving the problem of the labeled data limitation, transfer learning uses the relationship between the features (X_S) in the source domain (D_S) and the features (X_T) in the target domain (D_T) and transfer the model from the source task (T_S) to the target task (T_T) (Torrey & Shavlik, 2010). There are several categories of transfer learning methods according to different division ways. For instance, TL methods can be divided into Inductive TL, Transductive TL, and Unsupervised TL. As the scenario is $T_S \neq T_T$ but labeled data are available in both the source domain and target domain, the method is categorized as Inductive TL. In contrast, the Transductive TL requires $T_S = T_{T,}$ $D_S \neq D_T$, and partially labeled data. Unlike the former two, there is no label in the Unsupervised TL's source domain and the target domain (Pan & Yang, 2009).

In another way, TL methods are classified into Homogeneous TL and Heterogeneous TL according to the feature space information. The Homogenous TL solves the problem that the source domain and target domain share the same feature space ($D_S = D_T$), while Heterogeneous TL gives the solution for the $D_S \neq D_T$ type problems (Zhuang, Duan, et al., 2020). Another classification criterion is the transfer approach. According to Zhuang, Duan et al. (2020), there are four groups of TL, Instance-based TL, Parameter based-TL, Feature-based TL, and Relation-based TL. Through the re-weighting procedure, Instance-based TL transfers the instance from the source domain to the target domain. Parameter-based TL extracts the standard parameters sharing by the source model and the target model (Weiss, Khoshgoftaar & Wang, 2016). Feature-based TL transfers the original features and creates new feature representations for the target, which is further classified into two sub-categories, i.e., asymmetric and symmetric feature-based TL. The difference between asymmetric and symmetric methods is similar to the relationship between Heterogeneous TL and Homogenous TL. In other words, the Heterogeneous feature-based TL is classified as an asymmetric method, and the Homogenous feature-based TL is the symmetric one. The last category, relation-based TL, is a new and hot topic in recent years. Unlike the other three groups, the relationships among data are considered, and the transfer objects are the logic networks in the source domain. Such methods are inspired by Knowledge Graphs (Odom, Porter & Natarahan, 2015). Relation-based TL assumes that the knowledge networks are the same between the source domain and target domain or can be transferred from the source to the target. Few technologies can be recognized as relational transfer learning to our best knowledge, and all these methods are based on the Probabilistic Logic Models (Omran, Wang & Wang, 2016; Kumaraswamy, Odom, et al., 2015). According to Kumaraswamy et al. (2020), an Interactive Transfer Learning in Relational Domains, called LTL, was created utilizing a tree-type inductive logic programming. The data structures and labels are required in the source and target domain, narrowing the LTL application range. However, the easy interaction with the domain expert is the bright spot comparing the previous study.

Relational TL enables the ML models to transfer the knowledge networks from one domain to another. How to transfer knowledge from one domain to another is a critical issue for relational TL technologies. Unlike the other types of TL technologies, such as instance-based, parameter-based, and feature-based TL, the difference between the source domain and the target domain is easily expressed mathematically, e.g., the distance between data features across domains, the difference in the relational structure between different domains is hard to describe statistically. That is, the transference of relation needs support from a domain expert.

In Chapter 5 of the presented thesis, causal knowledge is introduced to relational TL problems and proposed a Relational Feature Transfer Learning (RF-TL) method. RF-TL simulates the human expert-like inference procedure of causality in the model training process and directly outputs reliable inferring results to human users.

2.3 A brief summary

In this chapter, the ways of incorporating domain expert knowledge into ML are overviewed. The expert knowledge summarized from experience helps to regulate the learning procedure of ML, such that the learning result is more in line with the expectations of the user. Besides the experience knowledge, causality is another kind of valuable expert knowledge for ML. The causality can be used to direct the learning procedure or post-explain the model. In the following chapters, three novel technologies are introduced as examples showing the specific process of utilizing experience knowledge and causality in ML mentioned in this chapter.

Chapter 3. Cluster Size Constrained Fuzzy c-means with Density Center Searching

In this chapter, an example of using expert's experience as "a priori knowledge" to regulate ML's learning procedure is shown. The knowledge is introduced into a traditional ML algorithm, Fuzzy c-means (FCM). After utilizing the expert knowledge, FCM's accuracy is increased and the results conforms to the human understandable ground truth.

3.1 Backgrounds

FCM is a widely used partitioning-based fuzzy clustering method that divides a set of data objects into fuzzy partitions by minimizing the intra-cluster variance and maximizing the inter-cluster variance under the objective function. As the most commonly used partition-based clustering method, FCM has the problem of "size insensitivity."

For one thing, as a Euclidean distance-based method, FCM ignores the scale difference in different dimensions of the input data objects, resulting in weak discernment on diverse-distribution data structures. Mahalanobis distance has been added to the original FCM by many researchers (Liu, Jeng, et al., 2009; Huang, Lin et al., 2018; Smiti & Elouedi, 2016). Using the covariance matrix to evaluate the correlations among different dimensions makes such FCM extensions better at extracting data structures with various distributions. However, they sometimes fail when the difference in the distribution is low. Similarly, Gaussian-based models are also good at distinguishing the distinctions of data distributions (Zhuang, Huang, et al, 1996). Ichihashi et al. (2001) found that only when setting a fuzzifier as a particular value, FCM achieves the same effect as Gaussian Mixture Density Decomposition (GMDD) by regularizing the K-L information (KFCM) (Dempster, Laird & Rubin, 1977; Krishnan, Ng, et al, 1997). KFCM is a partition-based improvement that solves the size-insensitive problem to some degree, but it is limited by the boundary conditions and cannot deal with clusters firmly next to each other.

Moreover, because FCM uses a sum-of-squared-errors objective function to obtain solutions, it tends to drift centers of smaller clusters to more massive adjacent clusters and to equalize cluster populations. To overcome this problem, the following algorithms have been proposed: Fuzzy Maximum Likelihood Estimation (FMLE) (Gath, Geva & Anir, 1989), semi-supervised FCM (ssFCM) (Bensaid, Hall et al, 1996), Cluster Size Insensitive FCM (CSI-FCM) (Noordam, Broek, et al, 2002), and Size Insensitive Integrity-Based FCM (SIIB-FCM) (Lin, Huang, et al, 2014). Lin et al. (2014) proved SIIB-FCM is super over the other three, so we will not repeat the details here. However, SIIB-FCM is quite sensitive to the distances among clusters, which means it is only good at handling datasets with relatively clear cluster boundaries. This shortcoming leaves it stretched when dealing with practical issues, like removing tiny noise mixed in the raw signal.

Considering the weaknesses mentioned above, a more effective way is needed to solve the size-insensitivity problem of FCM. In the actual application, one can sometimes obtain background information of data objects, such as the expected number of data groups and the approximate size of each group from preliminary surveys, prior studies, or views of domain experts in actual applications of data clustering. The proportion of each cluster size and the number of clusters may well be available as a priori knowledge for clustering. Clustering is based on a similarity measure to group semblable data objects, so it is commonly utilized in market segmentation, vehicle routing selection, and healthcare problems. On these occasions, users usually attach certain present conditions for different classification purposes. As a result, if such additional information is introduced as a priori information, it will likely help clustering algorithms find the intrinsic structure behind observations.

In Chapter 3, we describe a wrapper Fuzzy clustering algorithm by defining convincing adjustment directions by introducing Mahalanobis distance.

The remainder of Chapter 3 is organized as follows. In section 3.2, we review the FCM. Section3.3 proposes a new algorithm to assimilate a priori knowledge of the cluster size proportions into fuzzy clustering. Section 3.4 presents experiments to show the advantages of the proposed method over other algorithms, and section 3.5 makes a discussion. Finally, concluding remarks are given in section 3.6.

3.2 An Overview of Fuzzy c-Means

Given a dataset that contains n objects, $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_n\}$, the FCM algorithm attempts to minimize the following objective function

$$J = \sum_{j=1}^{n} \sum_{i=1}^{c} u_{ij}^{m} \left\| \boldsymbol{x_j} - \boldsymbol{v_i} \right\|^2 \tag{3.1}$$

where c is the number of clusters, v_i is the centroid of the i^{th} cluster, u_{ij} is the degree of membership of x_j to the i^{th} cluster and m is the fuzzifier exponent ($m \in R$ and $m > 1$). The fuzzifier m results in fuzzier clusters with smaller values. For every cluster, there are boundary conditions of $\sum_{i=1}^{c} u_{ij} = 1$ and $u_{ij} \in [0,1]$. A solution of objective function (3.1) can be obtained by an iterative process that updates the memberships and the cluster centers alternately. The membership u_{ij} is obtained from a given set of cluster centers by

$$u_{ij} = 1 / \sum_{k=1}^{c} \left(\frac{\|x_j - v_i\|}{\|x_j - v_k\|} \right)^{\frac{2}{m-1}} \tag{3.2}$$

In contrast, the cluster center v_i is calculated from a given set of memberships by

$$v_i = \sum_{j=1}^{n} x_j * u_{ij}^{m} / \sum_{j=1}^{n} u_{ij}^{m} \tag{3.3}$$

As mentioned in section 1, FCM attempts to minimize the objective function (3.1) and thus tends to give data objects a lower membership grade of clusters with higher density. As a result, smaller clusters may take more objects than much larger clusters, which leads to balanced sizes, and may lead to misclassification of data objects. SIIB-FCM aims at solving this problem.

3.3 Algorithm proposal

3.3.1 “Size constrained” objective function

The proposed method aims at adjusting multi-dimensional fuzzy clusters to attain desired population sizes. We assume that the desired proportion of cluster sizes is available as a priori knowledge for the clustering problem in question. In this premise, it is hypothesized that the closer the calculated cluster size is to the given cluster population, the better the data structure that will be extracted. To this end, the proposed algorithm modifies fuzzy partitions generated by FCM-like soft clustering methods by optimizing a “size constrained” objective function shown as the function (3.4).

$$Js_i = \left|\sum_{j=1}^{n} u_{ij} - S_i\right| \tag{3.4}$$

$\mathrm{S_i}$ herein represents the target size of the cluster c_i, which is constrained by $\sum_{i=1}^{c} S_i = n$, where n is the total population of data. u_{ij} is the fuzzy membership of data object $\boldsymbol{x}_j$ to c_i. and $\sum_{j=1}^{n} u_{ij}$ gives the calculated size of the fuzzy cluster c_i. Js_i evaluates the difference of the generated cluster population from the given cluster size S_i for the i^{th} cluster.

The method is divided into two stages. The first stage adjusts the position of each cluster while maintaining its shape, and the second stage adjusts the shape of each cluster while maintaining its center position. For a clear understanding, we present the specific procedures of the adjustment in the next section.

3.3.2 The procedure of the algorithm

As mentioned above, the proposed method uses the two-stage adjustments intended to be applied to fuzzy partitions generated by other soft clustering methods. Both stages aim to optimize function (3.4). Fig. 3.1 shows the entire flow of the algorithm.

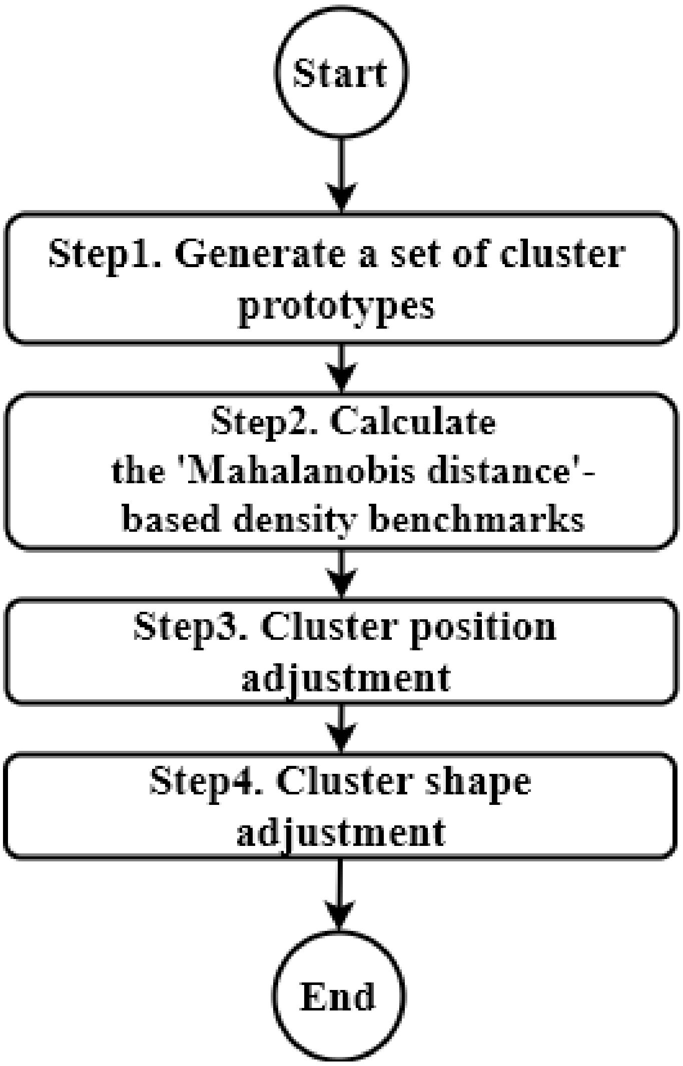

Start
Step1. Generate a set of cluster prototypes
Step2. Calculate the 'Mahalanobis distance'-based density benchmarks
Step3. Cluster position adjustment
Step4. Cluster shape adjustment
End

Fig. 3.1. The overall flow of the algorithm

After initialing the clustering solution, conduct the adjustment procedure for each cluster, respectively, and repeat steps 2 to 4. The order should be from the largest cluster to the smallest one. After completing the adjustment for one cluster, remove data belonging to the adjusted cluster from the input that obtained the highest membership to the target cluster and give the remainder of the dataset to step 2 as the new input for finding the density center of the next cluster. The memberships used for removing data are from the new membership matrix of the adjusted cluster.

For a more fundamental understanding, we use the following example to expound.

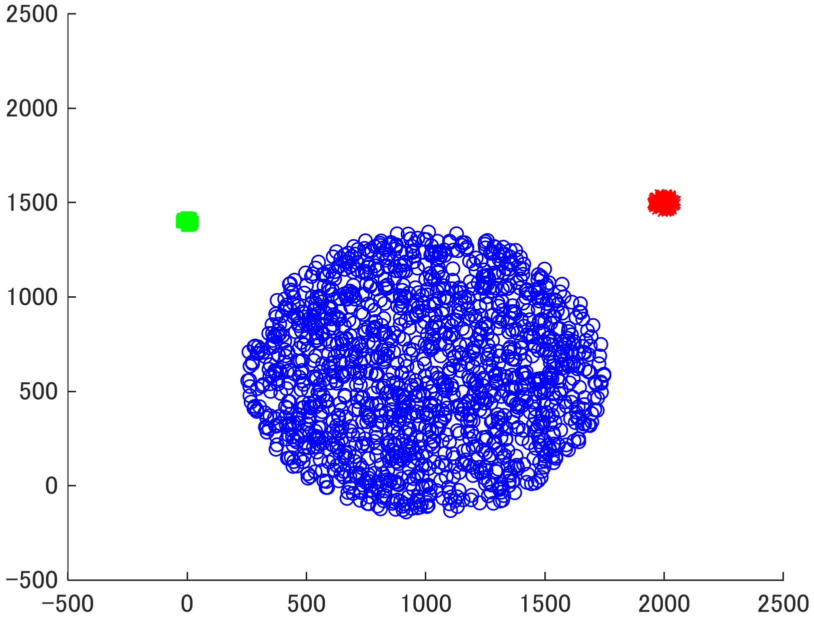


Fig. 3.2. Data distribution of the input dataset

As shown in Fig. 3.2, the input dataset contains three clusters with discrepant data populations of 100/1500/50. The following sections describe in detail the steps mentioned above using this example.

3.3.2.1 Cluster prototypes generation

Given a dataset that consists of n objects with p dimensions $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_n\}$, soft clustering methods can generate fuzzy partitions of the dataset. In the case of the FCM algorithm, a $p \times c$ matrix

$$V = (\boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \cdots \quad \boldsymbol{v}_c) = \begin{pmatrix} v_{11} & v_{21} & \cdots & v_{c1} \\ v_{12} & v_{22} & \cdots & v_{c2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1p} & v_{2p} & \cdots & v_{cp} \end{pmatrix} \tag{3.5}$$

expresses the cluster prototypes where c is the number of clusters and $\boldsymbol{v}_i$ is a p-dimensional vector that represents the centroid of the ith cluster c_i. With the cluster center matrix V, we can calculate the membership degree of each object $\boldsymbol{x}_j$ to each cluster c_i by function (3.2). In this paper, we assume m to usually be two when no domain knowledge is available.

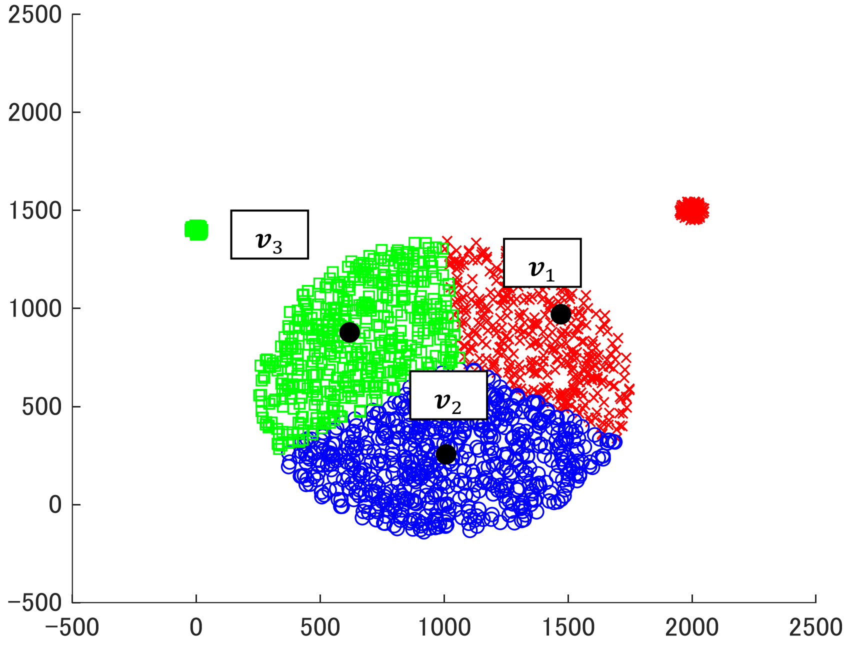


Fig. 3.3. FCM results and peak and floor of cluster 2

Also, according to function (3.2), every cluster c_i has the highest value of membership, i.e., $u_{ij} = 1$, at its center but the lowest value, i.e., $u_{ij} = 0$, at the centers of the other clusters. For the convenience of explaining the following processes, we name these positions the cluster peak and floor, respectively, for each cluster. As shown in Fig. 3.3, FCM classifies data points into three clusters with similar populations, and $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$ make up the cluster centers matrix. Using cluster 2 as an example, the position of $\boldsymbol{v}_2$ is the peak of cluster 2, while the positions of $\boldsymbol{v}_1$ and $\boldsymbol{v}_3$ are floors, which are the inputs to the following steps.

The subsequent stages of the proposed method regard the cluster center matrix V generated by FCM-like clustering algorithms as the initial solution.

3.3.2.2 Density prototypes

The "cluster center" is the arithmetic mean of all the points belonging to the cluster. Each point is closer to its cluster center than to other cluster centers. The original FCM uses Euclidean distance as the only measurement standard for the objective function, causing its driving smaller cluster centers closer to the bigger ones. That is why an equal tumble occurred. As shown in Fig. 3.4, it is obvious that point x is closer to the center of cluster 3 than cluster 2. Thus, FCM makes an erroneous judgment on data point x and divides it into cluster 3 but not the right one, cluster 2.

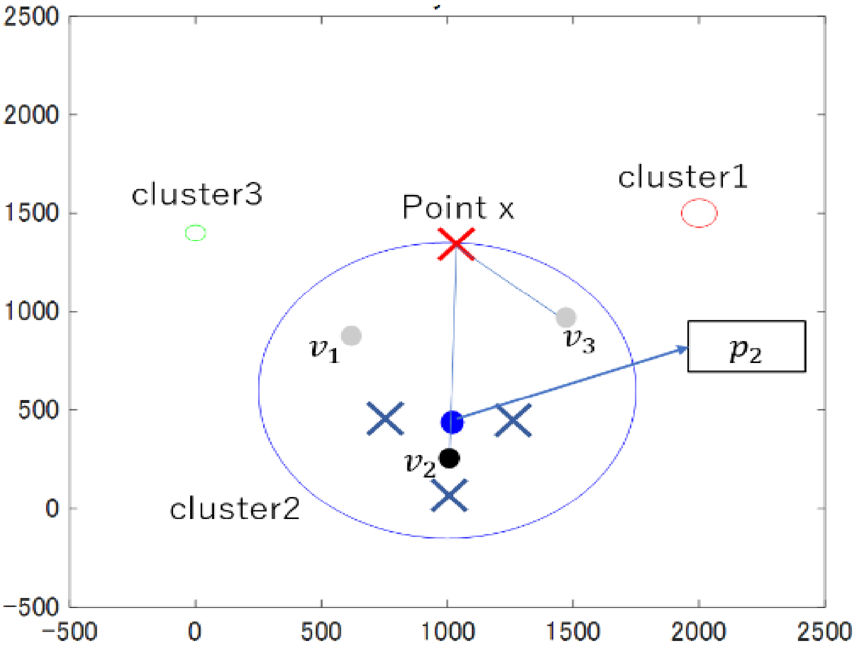


Fig. 3.4. Density center of cluster 2

The limited consideration of only one standard leading to a wrong answer for clustering. For finding out the most suitable positions of cluster centers, we introduce the cluster size S_i as another referring index. Under the constraints of the given "prior knowledge", we define a new concept, the density prototype p_i for each cluster.

The density reflects how close the data nearby each other in a certain area, and it can be calculated as $Size_i/Volume_i$, which expresses the number of data per unit volume. The cluster size S_i is a known knowledge, so as smaller the $Volume_i$ is, as higher the density will be. We use a sum of distances between each data point with its S_i closest neighbor points as the $Volume_i$. Furthermore, the density center p_i should be at the densest position of the i^{th} cluster. In another word, the $Volume_i$ center on p_i should be the smallest one. The density prototype of cluster i is generated by the following steps.

First, create a Mahalanobis distance matrix using all input objects

$$D = (\boldsymbol{dist}_1(k) \quad \boldsymbol{dist}_2(k) \quad \cdots \quad \boldsymbol{dist}_j(k)) =$$

$$\begin{pmatrix} 0 & d_{21} & \dots & d_{j1} \\ d_{12} & 0 & \dots & d_{j2} \\ \vdots & \vdots & \ddots & \vdots \\ d_{1j} & d_{2j} & \dots & 0 \end{pmatrix} (k = 1,2,\dots,j) \tag{3.6}$$

In function (3.6), each j-dimension vector $\boldsymbol{dist}_j$ presents the distance between each data object $\boldsymbol{x}_j$ with all the other objects. For a natural expression, d_{jj}, the distance to itself, is also presented in the matrix, which has a value of 0. The d_{kj} used here are Mahalanobis distances shown as the function (3.7).

$$d_{kj} = (x_k - x_j)^T S^{-1} (x_k - x_j) \tag{3.7}$$

where S is the covariance matrix of x_k and x_j.

For each distance vector $\boldsymbol{dist}_j$, we find the number of S_i smallest components that present $\boldsymbol{x}_j$ has S_i neighbor data objects nearest to it. Then, make a sum of these S_i components of $\boldsymbol{dist}_j$ and record it as $Volume_j$. Furthermore, finding out the density prototypes $p_i = \boldsymbol{x}_p$ for the i^{th} cluster by seeking the data object $\boldsymbol{x}_p$ that has the minimum value of $Volume_j, j = 1,2,\ldots,n$. As mentioned above, p_i is considered as the density prototype because comparing with other data objects, the S_i data objects around it are always the most compact regardless of the numerical value or distribution features. $p_i$is an appropriate symbol for defining the searching direction.

Here, what should be paid attention to is that the position of density prototype p_i is always different from the cluster center. The density prototype p_i is used as a benchmark in the next two steps, which guide the cluster centers moving towards a high-density area. Hence, the cluster center appears on the line of moving direction but may not coincide with the density prototype. As shown in Fig. 4.4, the cross marks present some typical data points in cluster 2. We assume that all the other data points distribute uniformly. The result of the “cluster center” position of cluster 2 given by the proposed method will not be p_2 but partial to data point x.

3.3.2.3 Cluster position adjustment

The first stage of adjustment is intended to find out the desired position of each cluster for obtaining the target cluster sizes. In this stage, the algorithm explores new positions of both peak and floor points for each cluster under constraints that limit the search directions.

We first duplicate the cluster center matrix V for each cluster. Let us define the cluster c_i's duplicate of the matrix V as $V^{(i)}$ here. The ith column of $V^{(i)}$ represents the peak of cluster c_i, and the other columns represent its floor points. On the other hand, the search direction of the cluster c_i is constrained as pointing from the peak to p_i. During the search, the cluster peak and floor positions will translate together in the same direction. The length between the peak and p_i determines the search interval. In our experience, the searching distance should be over 1.5 multiples of the

$peak{\sim}p_i$ distance to obtain a broader searching range, which depends on the fuzzy degree of the dataset. The fussier the data, the lower the searching distance should be. However, the multiple does not need to be set more than 3, which will waste calculating time. The multiple of the $peak{\sim}p_i$ distance is recommended to be 2 if there is no domain knowledge. Figure 5 shows the adjustment procedure of cluster 2 in this step.

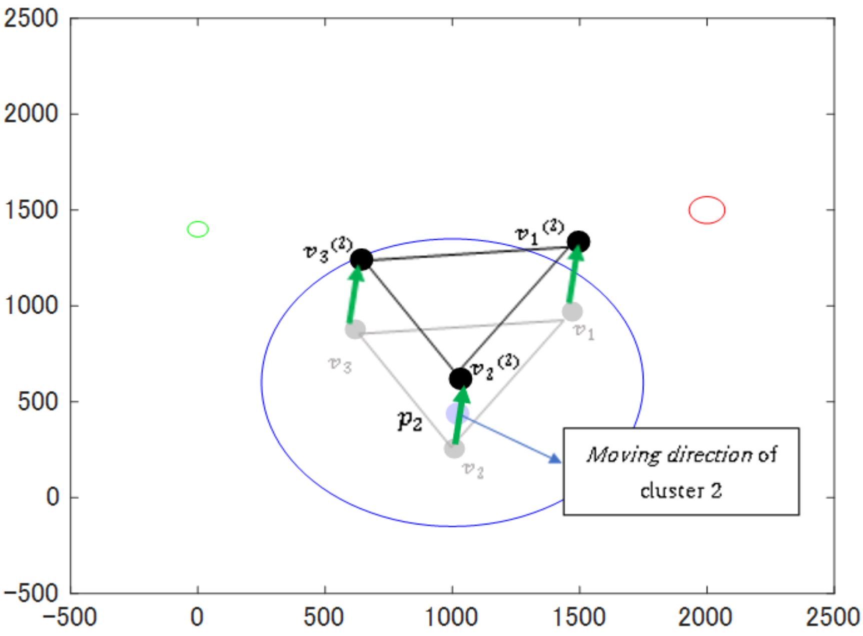


Fig. 3.5. Cluster position adjustment of cluster 2

In Fig. 3.5, the adjusting direction of cluster 2 is from v_2 pointing to p_2. The searching distance is double the distance between v_2 and p_2. The peak and floors are moved together in the same direction and searching distance here.

As the search direction and the interval are specified, the algorithm attempts to minimize objective function (3.4) for the chosen cluster. The search updates the cluster's peak/floor matrix $V^{(i)}$. As shown in Fig. 3.5, the black points are the results of this procedure for cluster 2.

3.3.2.4 Cluster shape adjustment

The second stage of adjusting the membership functions changes the shape of each cluster to minimize objective function (3.4) further by moving the clusters' floor positions while fixing their peak positions.

Given peak/floor matrixes $V^{(i)}$ gotten from the first stage, the algorithm attempts to search for a better floor point placement for each cluster c_i. During the search, all the floor points of the target cluster translate together. The searching direction and the interval are the same as in the first stage.

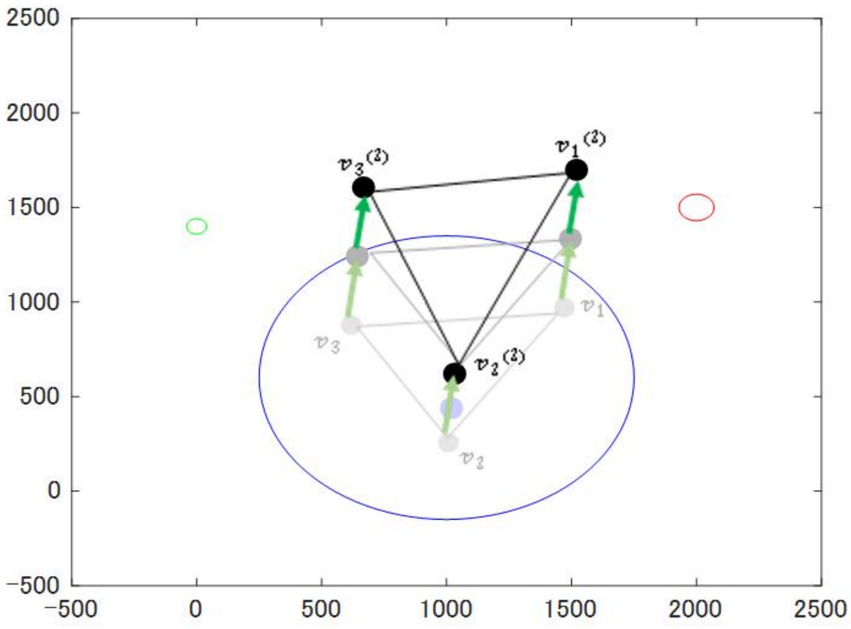


Figure 3.6. Cluster shape adjustment of cluster 2

As shown in Figure 3.6, during the shape adjustment procedure, the peak $v_2{}^{(2)}$ of cluster 2 is kept flexible and the floors $v_1{}^{(2)}$ and $v_3{}^{(2)}$ are adjusted in the same direction as in the position adjustment step.

3.3.2.5 Generate new membership functions

The result of the two-stage adjustment is a set of the updated peak/floor matrixes $V^{(i)}$ of all clusters. By applying function (3.2), we can obtain different membership matrixes $U^{(i)} = (u_{ij})_{1 \le i \le c,\ 1 \le j \le n}$ from each peak/floor matrix $V^{(i)}$. The proposed method creates the net membership matrix with the ith row out of $U^{(i)}$ arranged one over the other as below:

$$U = \begin{pmatrix} \boldsymbol{U}_1{}^{(1)} \\ \boldsymbol{U}_2{}^{(2)} \\ \vdots \\ \boldsymbol{U}_3{}^{(3)} \end{pmatrix} = \begin{pmatrix} u_{11}^{(1)} & u_{12}^{(1)} & \cdots & u_{1n}^{(1)} \\ u_{21}^{(2)} & u_{22}^{(2)} & \cdots & u_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ u_{c1}^{(c)} & u_{c2}^{(c)} & \cdots & u_{cn}^{(c)} \end{pmatrix} \tag{3.8}$$

At the same time, we combine the peak vectors of each $V^{(i)}$ as the new cluster center matrix:

$$V\_new = \begin{pmatrix} \boldsymbol{V}_1{}^{(1)} & \boldsymbol{V}_2{}^{(2)} & \cdots & \boldsymbol{V}_c{}^{(c)} \end{pmatrix} \tag{3.9}$$

Fig. 3.7 shows the results of V_new and the classification of the proposed algorithm that we used as the example.

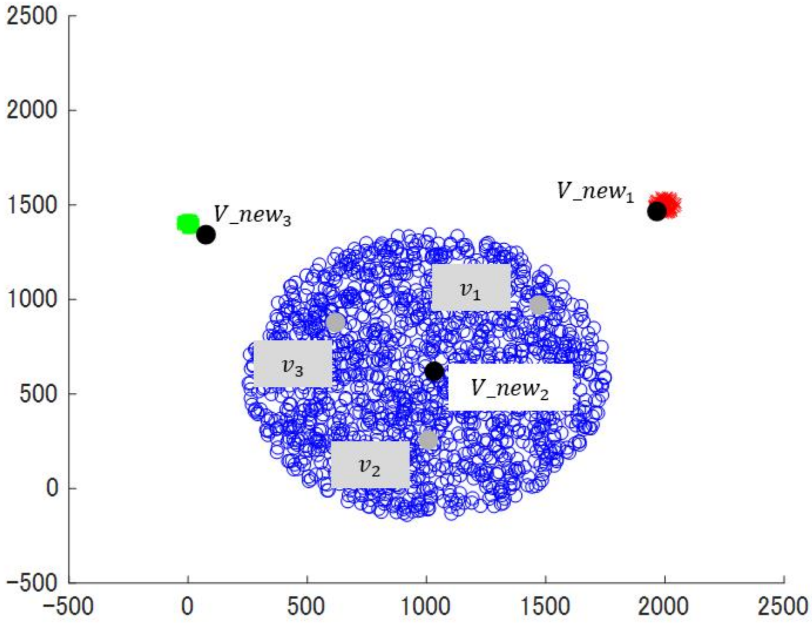


Fig. 3.7. Final results

3.4 Experiments

We use numerical and practical experiments to examine the effectiveness of the proposed method in comparison with FCM, SIIB-FCM, and KL-FCM. For testing the stronger points of the proposed algorithm, we conducted four experiments. The aims and the contents of the four experiments are shown in Table 3.1.

Table3.1 Aims and the contents of the four experiments

No. of Experiment	Contents
Experiment 1	Data structures extraction test
Experiment 2	Distances tolerance test
Experiment 3	Robustness test
Experiment 4	Practical example of a healthcare problem

3.4.1 Capacity of correctly extracting data structures

In this experiment, a numerical dataset is prepared for testing the correct clustering power of the proposed method by evaluating the error rate of the data population classified to each cluster, the accuracy of the clustering results. Additionally, for the numerical dataset, we also evaluate the indexes typically used for soft clustering methods: Dunn's index (DI) and Xie-Beni index (XB).

The experiment used artificial datasets with different cluster populations. Section 3 has used a three clusters dataset as an example to explain the algorithm. Here, we created another different dataset to test the effectiveness of the proposed algorithm under the situation of different data distributions. Fig. 3.8 shows the original distributions of the input dataset.

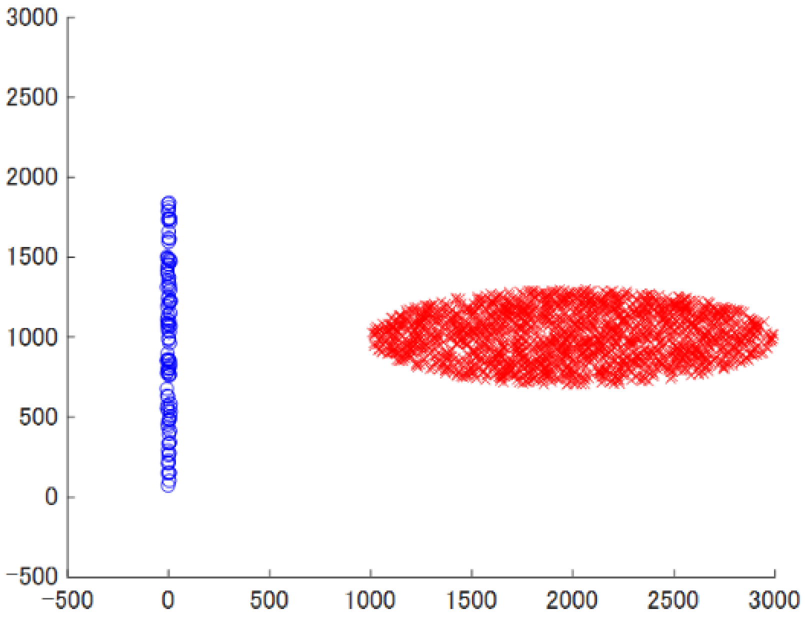
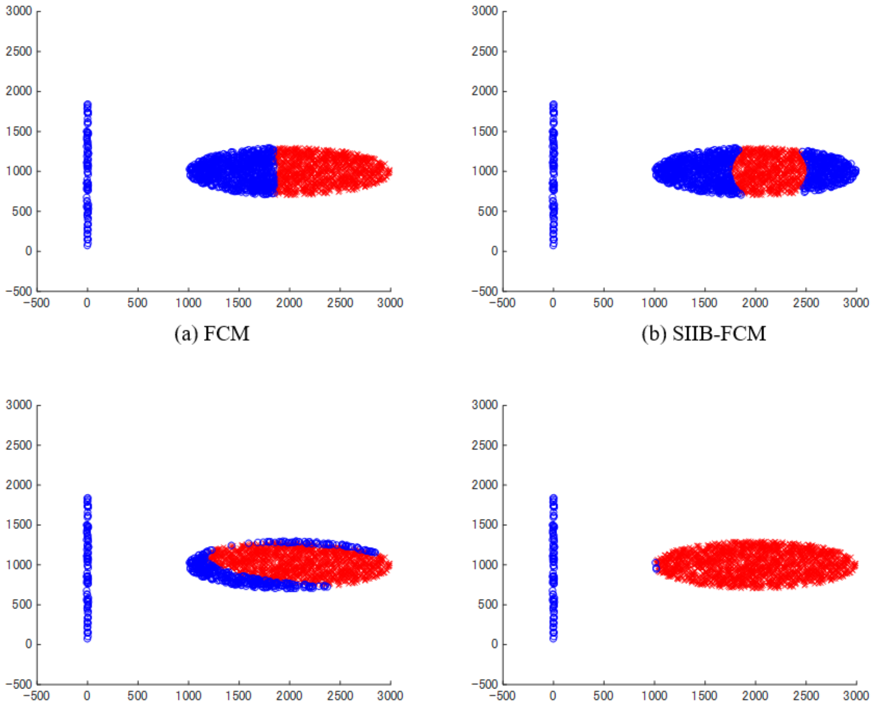


Fig. 3.8. Numerical Dataset with 2-clusters and different data distributions

For this test, there are three cross-test datasets, and the size of the cluster is set as 2000/50. Fig. 3.9 shows the results.



(a) FCM

(b) SIIB-FCM

(c) KL-FCM

(d) CSCD-FCM

Fig. 3.9. Clustering results of the four algorithms

KL-FCM, a segment-based clustering method using Mahalanobis distance, highlights its advantages when input data had diverse distributions. The proposed algorithm also draws on the advantage of Mahalonobis distance in the stage of finding the "density center." Thus, CSCD-FCM is tolerant to multi-distribution data to some degree.

Tables 3.2 and 3.3 show the related evaluation indexes.

Table 3.2 Cluster size results of two clusters with different distributions

Method	**C1 Size**	**C1 Difference**	**C2 Size**	**C2 Difference**
FCM	1177±26	823±26	923±26	-823±26
SIIB-FCM	807±105	1193±105	1293±105	-1193±105
KL-FCM	1582±13	418±13	518±13	-418±13
CSCD-FCM	**2000±1**	**0±1**	**100±1**	**0±1**

Table 3.3 Evaluation indexes of two clusters with different distributions

Method	**Accuracy**	**F1_score**	**DI**	**XB**
FCM	0.6080±0.01	0.6528±0.0033	0.0043±0.0016	0.1532±0.0032
SIIB-FCM	0.4321±0.05	0.6095±0.0119	0.0027±0.0013	0.0428±0.0031
KL-FCM	0.7171±0.13	0.6922±0.035	0.0016±0.0001	72.3254±58.074
CSCD-FCM	**0.9998±0.0003**	**0.9991±0.0015**	**0.303±0.2565**	**0.1868±0.0086**

Here, SIIB-FCM shows the lowest XB value on this occasion, which is mainly caused by the reduction of memberships for all the data. However, comparing with the accuracy of extracting the exact data structure, the lowering of the XB index does not make sense. Considering the size-insensitivity problem, CSCD-FCM performs best here.

3.4.2 Clustering capacity under different distances among clusters

SIIB-FCM and KL-FCM were proposed to improve FCM performance on a clustering dataset with unbalanced cluster sizes and different data distributions. Both methods perform well when clusters are far enough apart but not when clusters are close. We call this the “distance-sensitivity problem” here. This problem commonly occurs in unsupervised clustering methods, especially when the cluster sizes are unbalanced. The following experiments focus on the distance-sensitivity problem, and the results show that the proposed method has a quite high tolerance to the compactness of clusters.

In this experiment, different datasets are generated to two circle-shape clusters using normal-distribution data with the cluster size of 2000/50 with different cluster distances, and also a three-dataset validation is conducted to each kind of input. The center distance of the circle measures the distance between clusters.

We use accuracy and F1_score here to evaluate the clustering possibility of the four algorithms under close to far cluster distances. Figs. 4.10 and 4.11 show the accuracy and F1_scores of FCM, SIIB, KL-FCM, and CSCD-FCM under different cluster distances.

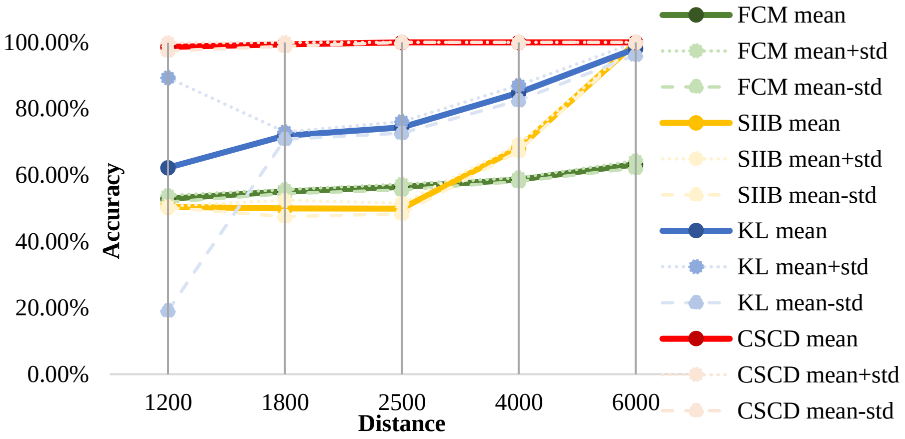


Fig. 3.10. Comparison of the Accuracy of four algorithms under different distances

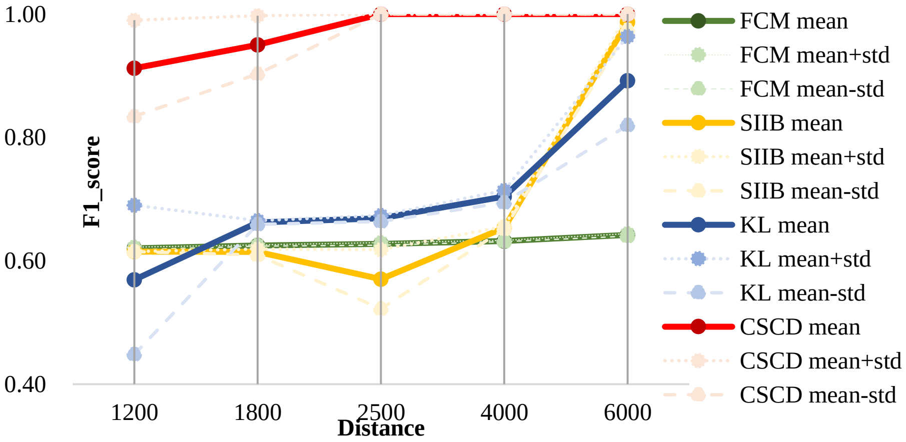


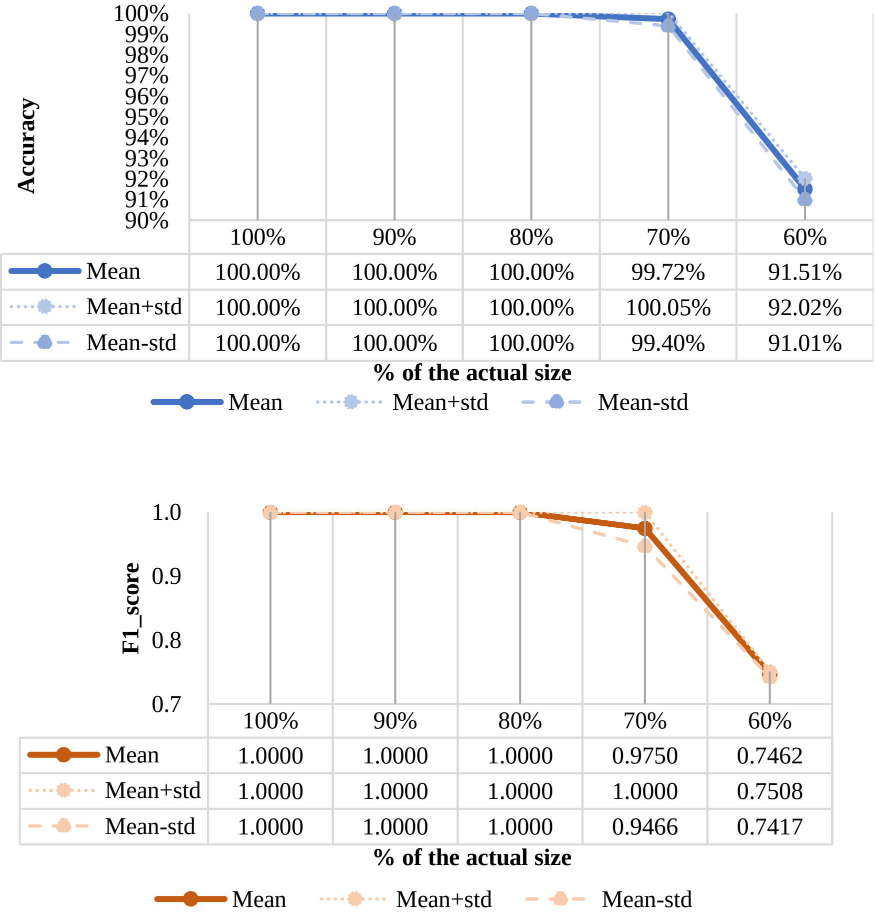
Fig. 3.11. Comparison of the F1_score of four algorithms under different distances

Accuracy and F1_score reflect the degree of the correct classification. Figures. 10 and 11 show that the size-insensitivity problem exists in FCM regardless of the distances among the clusters. When the interval between the two clusters is big enough, over 4000 shown in this example, KL-FCM and SIIB-FCM can obtain better results than FCM. However, when the clusters are closer together, KL-FCM and SIIB-FCM perform even worse than the original FCM sometimes.

In contrast, the proposed algorithm has almost no sensitivity to distance. That means that even the data distribution is very tight, so the clustering result is still good. This strong point may make CSCD-FCM more widely applicable than the other similar algorithms.

3.4.3 Robustness of the algorithm

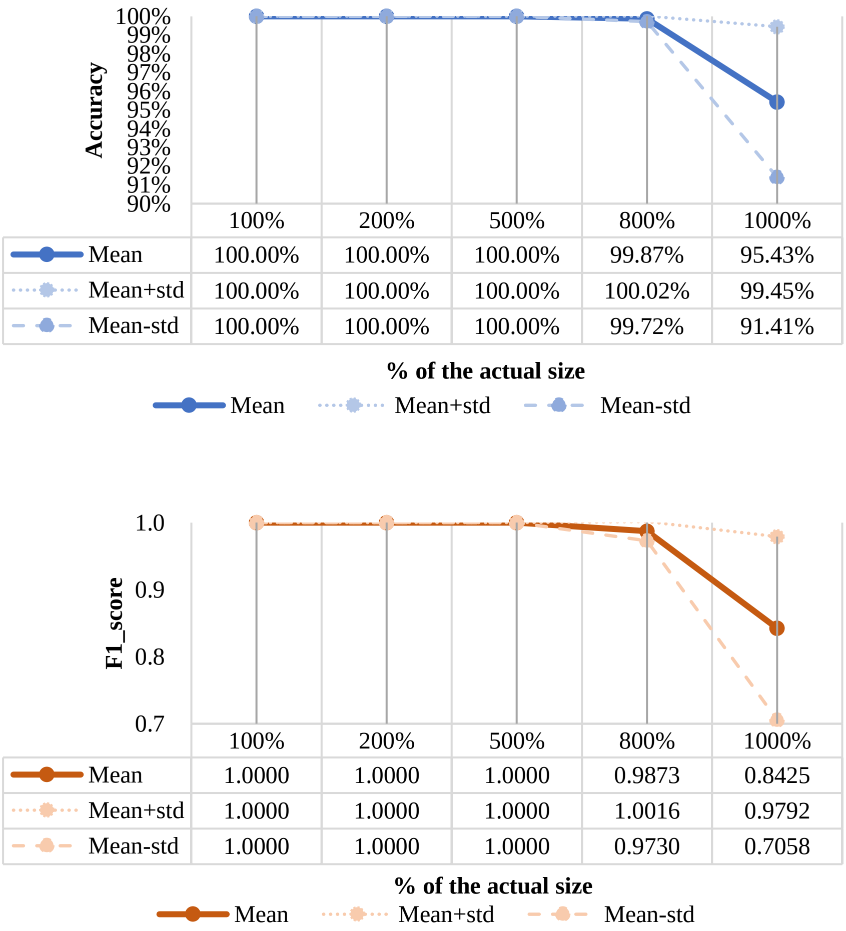
This experiment is aimed at testing whether the proposed algorithm can obtain adequate results when the cluster size information contains errors. Datasets used in this experiment are still generated in two circular clusters using normal-distribution data with the cluster size of 2000/50. We evaluate the accuracy and F1_score by fixing one cluster's size, which is the correct size information, and reducing or increasing the other clusters' sizes by a certain percentage of the actual size. Figs. 3.12 and 3.13 show the results.



	100%	90%	80%	70%	60%
Mean	100.00%	100.00%	100.00%	99.72%	91.51%
Mean+std	100.00%	100.00%	100.00%	100.05%	92.02%
Mean-std	100.00%	100.00%	100.00%	99.40%	91.01%

	100%	90%	80%	70%	60%
Mean	1.0000	1.0000	1.0000	0.9750	0.7462
Mean+std	1.0000	1.0000	1.0000	1.0000	0.7508
Mean-std	1.0000	1.0000	1.0000	0.9466	0.7417

Fig 3.12. Changes in indexes as cluster size decreases



	100%	200%	500%	800%	1000%
Mean	100.00%	100.00%	100.00%	99.87%	95.43%
Mean+std	100.00%	100.00%	100.00%	100.02%	99.45%
Mean-std	100.00%	100.00%	100.00%	99.72%	91.41%

	100%	200%	500%	800%	1000%
Mean	1.0000	1.0000	1.0000	0.9873	0.8425
Mean+std	1.0000	1.0000	1.0000	1.0016	0.9792
Mean-std	1.0000	1.0000	1.0000	0.9730	0.7058

Fig 3.13. Changes in indexes as cluster size increases

The results show that the proposed algorithm has certain robustness to the error input of size information. The algorithm still obtains both accuracy and F1_score over 0.95 when the given information reduces the bigger cluster to 70% of its actual size and increases the smaller cluster to 800% of its actual size. Thus, the proposed algorithm has a degree of tolerance to the “prior information error.” Considering this, when using the algorithm to deal with a practical problem, the requirement of the prior input knowledge of size information is not so strict. An inevitable ambiguous error is permitted.

3.4.4 A practical example

In this experiment, we conduct a practical application on a healthcare problem. The data comes from the questionnaires aiming for diagnosing Obstructive Sleep Apnea (OSA) disease. According to the most recent survey from the American Academy of Sleep Medicine (AASM), the sleep health research authority, 71.9% of the general people aged between 40~85 years old are suffering from this disease (Heinzer, Vat, et al, 2015). Because people are unconscious when sleeping, it is hard to discover OSA in the primary care stage. A commonly used means for self-diagnosing on OSA is the screening tools, and the STOP-Bang questionnaire (Chung, Abdullah & Liao, 2016) is the most famous one.

There are 8 questions in the STOP-Bang questionnaire. We collect 3,931 people's STOP-Bang data and Apnea-Hypopnea Index (AHI) from the SHHS, where AHI is regarded as a golden standard indicator to diagnose OSA. The participates age from 39 to 95 years old. None of these 3,931 people have been diagnosed as OSA by a medical doctor, nor has accepted an OSA treatment before collecting the data. An AHI ≥ 5 is a critical value to judge OSA, and there are 2,730 objects, occupying about 70% of the 3,931 people, that should be diagnosed as OSA. We applied the algorithm to 8-dimensions input data of the 8 questions from the 3,931 objects with the prior-knowledge of 2,730 objects for the positive cluster and 1,201 objects for the negative one. The clustering results were evaluated by comparing AHI value judgments. For a healthcare problem, medical doctors always concern more about the sensitivity of the positive rate. Hence this index is introduced as another evaluate indicator other than the accuracy and F1_score. The clustering results of the four methods are shown in Table 3.4.

Table 3.4 Evaluation indexes of the practical example

Method	**Accuracy**	**F1_score**	**Sensitivity**	**Cp Size**
FCM	0.3885	0.3531	0.4608	2190(-540)
SIIB-FCM	0.5785	0.5517	0.6150	2285(-445)
KL-FCM	0.4861	0.5523	0.3795	1362(-1368)
CSCD-FCM	**0.7202**	**0.6552**	**0.8359**	**2934(+204)**

The Cp in Table 3.4 presents the positive cluster, and Cp Size recorded as the data population classified to the positive cluster (differences with the true cluster size, 2,730). Table 3.4 clearly shows that with the constrained of the prior-knowledge, the difference between the data populations classified to each cluster of the proposed method and the given size is smallest, and the accuracy and the sensitivity is also much higher than the other methods.

3.5 Discussions

The size-insensitivity problem is common in objective function-based clustering methods like FCM. FCM clusters data objects by minimizing the sum of Euclidean distance errors among data, which causes the data object at the edge of bigger clusters to drag the center of the adjacent smaller clusters toward itself. The hauling leads to the clustering results of FCM often having balanced sizes.

To solve this size-insensitivity problem, improved methods such as SIIB and KL-FCM introduce extra-information like data populations and distribution characteristics to correct the shortcomings caused by the objective functions based on the Euclidean distance. Nevertheless, the only objective function that these methods try to optimize is still the sum of variance-errors of all data points, which leads to them being sensitive to the distance between neighboring clusters. This kind of simultaneous use of a variety of information causes the different types of information to interfere with each other, thus the algorithms cannot fully mine all the information carried by the data. The proposed method divides the optimization procedure into two stages: one for variance error optimization and the other for extra-knowledge optimization. The segmented use of the objective functions helps the proposed algorithm better utilize the necessary information hidden behind the data.

Additionally, attaching extra-knowledge as clustering data objects may bring about unnecessary biases to the original data structure. For one thing, the a priori knowledge utilized by the algorithm is not groundless rumors. It must be the knowledge carried by the input data objects, and the use of this knowledge should not affect the hidden information in the dataset that the algorithm hopes to extract. For another thing, the proposed algorithm has some tolerance to errors in input information of cluster size, which will help reduce the influence of the biases in application cases.

3.6 A brief summary

The present chapter proposes a new, improved fuzzy clustering method to solve the size-insensitivity problem. The proposed method is good at extracting data structure regardless of the cluster numbers or data dimensions and can deal with multi-distribution datasets. Unlike other similar algorithms, the proposed CSCD-FCM is not sensitive to cluster distance, which widens its application range. The proposed method can correct the error input of the priority information itself, so its application is soft and flexible.

The size-insensitivity problem is one of the significant shortcomings of traditional Fuzzy c-Means (FCM) and its variations, resulting in their week clustering probability when dealing with unbalanced datasets. This paper thus proposed an improved fuzzy clustering method on the basis of the assumption of obtaining “cluster sizes” as a priori information and subjoins a size objective function to take advantage of the information lost by traditional FCM. The algorithm contains a two-stage adjustment. One is the position adjustment to find the most suitable location for each cluster. The other is a further adjustment that continues to optimize the size objective function by changing the cluster shape. Additionally, utilizing Mahalanobis distance as defining the moving directions during the adjustment procedure enhances the capacity of the algorithm to deal with multi-distribution datasets. Compared with other algorithms aimed at solving the same problem, such as KL-FCM and SIIB-FCM, the proposed method can extract the actual data distribution more correctly and has a high tolerance to cluster distance. Additionally, the proposed CSCD-FCM can offer the right clustering solution.

The proposed CSCD-FCM not only improves the accuracy of prediction comparing with the original FCM. As a classification method, the explainability of it does not lose because of the utilization of the separated objective functions aligning to the original objective functions. The reliability of CSCD-FCM is high.

Chapter 4. Structural Equation Modeling-based Explainable Machine Learning model

Chapter 4 introduces an SEM-based explainable machine learning model. Six parts construct the model, and the model can realize the functions, including data analysis, machine learning, and causal analysis. The proposed model is design-explainable. Causality is introduced for post-explaining the model. The reminds of Chapter 4 is made up as follows.

In section 4.1, the background knowledge of the SEM is reviewed. Section 4.2 details the specific procedures of the proposed model. In Section 4.3, the model is applied to a healthcare problem, and Section 4.4 discusses the results. Finally, concluding remarks are given in Section 4.5.

4.1 Structural Equation Modeling

Structural Equation Modeling (SEM) is usually a two-step procedure. One is Exploratory Factor Analysis (EFA). The other is Confirmatory Factor Analysis (CFA).

EFA reliably classifies data items into corresponding factors without a specific hypothesis, which aims at identifying latent factors on the basis of the observed variables (Ulluman & Bentler, 2003). For a research topic, the result of EFA may not be unique. Researchers must balance the number of extracted factors avoiding both parsimony and plausibility. Hence, a repeated operation is necessary for EFA to obtain an excellent fitting model in the follow-up CFA procedure. A total explaining variance over 60% and a Kaiser-Meyer-Olkin (KMO) test result higher than 0.5 are the reference points of EFA.

In contrast to EFA, the hypothesis is necessary for the CFA procedure. Fig. 4.1 shows a conceptual model of CFA.

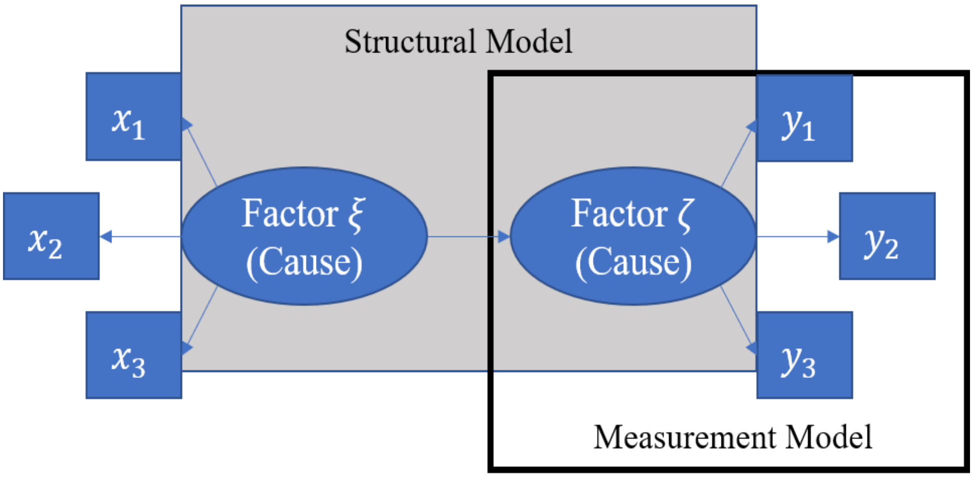


Fig 4.1. Conceptual model of CFA.

The measurement model and structural model make up the hypothesis for CFA to test. As mentioned above, EFA offers the results of extracted factors and their inclusive manifest variables, which builds up the measurement part. The structural part specifies the logic paths among factors. After constructing the model, the factor loadings between manifest items and latent factors and between every two factors are estimated according to the manifest items' covariance matrix. For example, the model shown as Fig. 4.1 can be expressed as

$$X = \Lambda_x \xi + \delta_x \tag{4.1}$$

$$Y = \Lambda_y \zeta + \delta_y \tag{4.2}$$

$$\zeta = \Gamma \xi + \varepsilon \tag{4.3}$$

where X and Y are 3-dimensions manifest variables. ξ and ζ are common factors measured by X and Y respectively. δ and ε are error terms. Using estimation methods, such as maximum likelihood estimation, the loading matrixes Λ_x and Λ_y are easy to calculate, which presents the factor loadings for each manifest variable to its latent factor. Moreover, Γ, the regression weight between two factors, can be estimated as well. The mark of a successful model is obtaining goodness of fit, proving that the hypothesis can express the structure of the data.

4.2 Proposal of the Structural Equation Modeling-Explainable Machine Learning model

4.2.1 Overall proposal

The procedure of the proposed method contains six steps: data preparation, data management, structure learning, parameter learning, model utilization, and model validation. The overall structure is shown in Fig. 4.2.

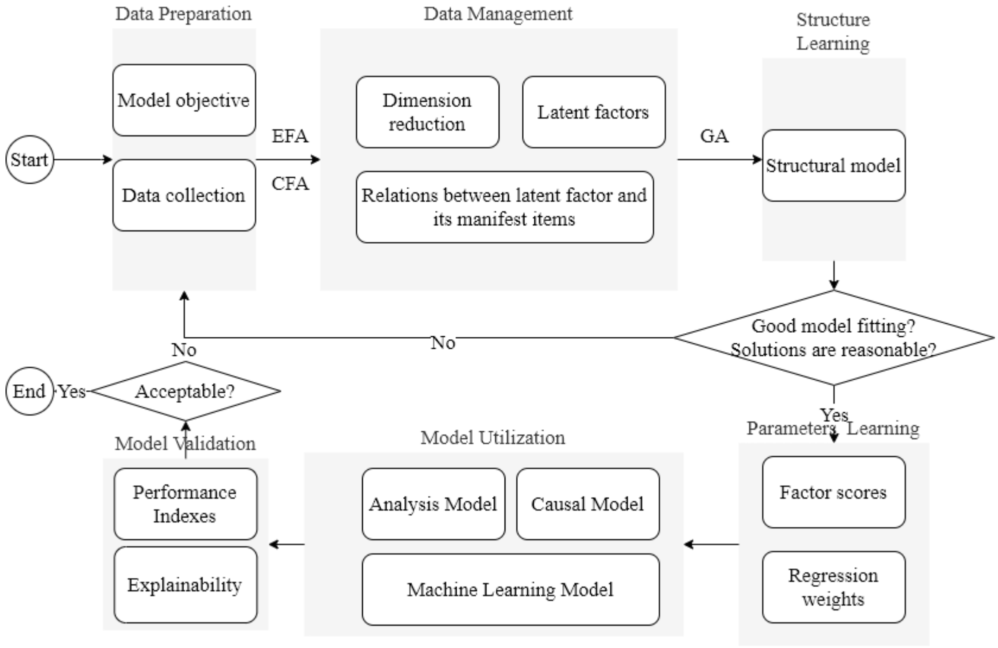


Fig. 4.2. Overall structure of proposed method

4.2.2 Data preparation

The starting point of the method is the preparation of data, before which the purpose of the model should be determined. Comprehensively considering all the possible related factors can save many resources for subsequent steps, such as the application fields, users' needs, and the quality of existing datasets. The necessary data should be collected corresponding to the experts' knowledge.

For easy illustration, in the following sections, we assume N-dimensions data have been collected for ML problem A.

4.2.3 Data management

Data management aims to simplify data dimensions, extract latent factors, and verify correlations between the latent factors and their manifest items. In the first step, the proposed method collects a large number of data features that relate to the learning target. However, the superfluous data dimensions inevitably cause a computational burden. Usually, not all collected characteristics contribute to the prediction goal. Thus, a filtering and dimensionality reduction process is necessary to extract the feature values closely related to the prediction goal and is sufficient to solve the ML problem.

The proposed method assumes that each dimension of the collected data is a manifest item in SEM, which is the input for data management. Moreover, data dimensions are reduced through EFA and CFA.

For data management, EFA is used to simplify the observed variable and extract latent factors. CFA is used for further reducing items that have low factor loadings to the corresponding latent factors. The initial dataset contains N-dimensions data. EFA gets rid of the variables and extracts a suitable number of factors. Through a factor rotation process, the calculated factor loadings evaluate the variables' ability to explain each common factor. A factor loading over a threshold (>0.3 in the presented paper) presents the variable belonging to the corresponding factor. Factors that contain fewer than two items are inadvisable, and the final results are more convincing if every observed variable belongs to only one factor. Also, for different research purposes, researchers can reserve or remove factors in accordance with their experience. The final model should reach the reference points mentioned in Section 4.2.1. Let us assume that, for problem A, EFA extracts 15 items belonging to 5 factors.

Next, CFA is used for further confirming the factor loadings. In this step, the emphasis is to verify whether the extracted manifest items are suitable to explain the corresponding factor, and the complexity of the relations among latent factors is not considered here. Also, the differences in connections among latent factors do not affect the factor loadings between the manifest items and their corresponding factor. Thus, the hypothesis model is made with all factors correlated with each other in this step. In EFA, all the factors are compulsively assumed to be mutually independent. However, a structural model used in CFA considers the regressions or correlations among the factors. As a result, the factor loadings obtained from CFA are usually lower than those obtained from EFA. That is why CFA contributes to reducing data dimensions in this step further.

In the example of problem A, the CFA result shows that the factor loading of item 7 is 0.2, which is not suitable for measuring factor 3, so item 7 is removed from the dataset. Finally, the data management procedure extracts 14 items and 5 factors, shown as Fig. 4.3.

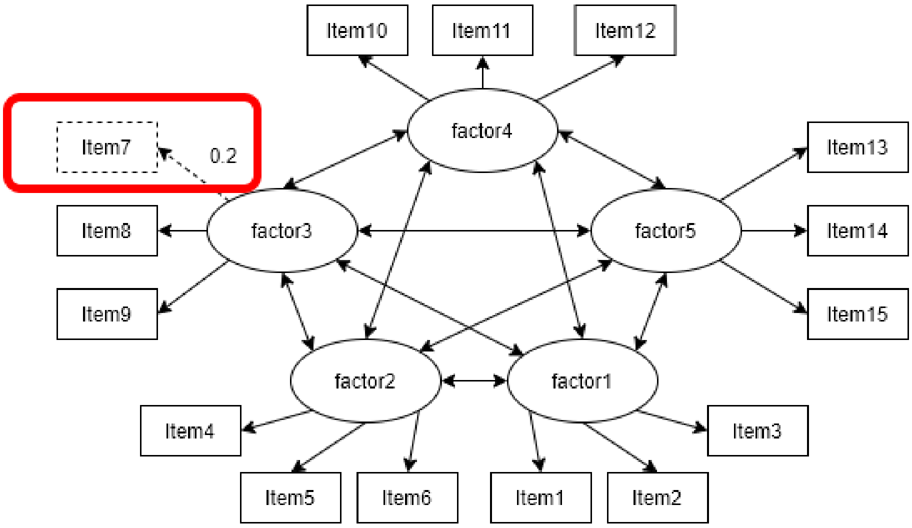


Fig. 4.3. Data management result

4.2.4 Structural learning

The structure learning procedure aims to specify the relations between every two latent factors and find out the best model fitting on the given data. When there is enough domain knowledge, the structure can be given by the experts. Nevertheless, a more automatic way is to use the heuristic method. In the proposed method, we use Genetic Algorithm (GA) to conduct the structure learning procedure, and the steps of applying GA in SEM are as follows.

Step 1. Determine the fitness indicators;

Step 2. Code the chromosomes and set evolution parameters;

Step 3. Generate the initial population and perform pre-evolution iterations for finding "suggestions";

Step 4. Add the "suggestions" to the initial population and conduct the evolution steps.

4.2.4.1 Fitness indicators

The goodness of fit indicators are the criteria for assessing whether SEM models stand or fall. The basic purpose of the indicators is to measure whether the theoretical model constructed by researchers reasonably explains actual observed data. In the proposed study, for obtaining a simple and clear explainable model, the complexity of the model is also noteworthy. As a result, apart from the commonly reported evaluation indexes, the Goodness of Fit Index (GFI), Chi-square (χ^2), and Comparative Fit Index (CFI), the indexes measuring the Degree of Freedom (DoF) are also considered by the proposed method, which are the Root Mean Square Error of Approximation (RMSEA) and the Adjusted Goodness of Fit Index (AGFI). When the number of factors is fixed, the higher the DoF, the simpler the model. The organized and used indicators in this research are illustrated as follows.

The different index evaluates the goodness of fit of a model from different aspects. Only choosing one index as the GA fitness function is not all-inclusive, combining all five indexes and defining a Comprehensive Evaluation Index (CEI).

$$CEI = GFI + AGFI + CFI + (1/\chi^2) + (1/RMSEA) \tag{4.4}$$

Also, every singular index is checked simultaneously as CEI changes to avoid the situation that a certain indicator does not meet the fitting requirements.

4.2.4.2 Chromosomes encoding and parameters setting

The corresponding GA terms to their meaning in SEM are shown in Table 4.1.

Table 4.1. GA-SEM terminology

GA term	Meaning in SEM
Gene	Hypothesis path among factors
Chromosome	Hypothesis model
Population	Group of chromosomes
Fitness Function	CEI

In the proposed method, each gene indicates one path from one factor to another. The gene will be coded as "1" if the relation is true and "0" if false. What should be paid attention to here is that the path has the direction, and the difference between the directions affects the results of model fitting. Thus, when "1" is given to the gene of factor A pointing to factor B, "0" should be given to the gene of factor B pointing to factor A at the same time. Also, a factor cannot point to itself. One chromosome contains n*(n-1) genes if n factors are used in the model.

Additionally, the double arrows connection in an SEM model means two factors are correlated, but the causal relationship remains unclear. One function of the proposed method is to do causal analysis, so a double-direction arrow and the circle structure are not permitted in the model. The population number is set in accordance with the number of factors, which should be higher when there are more latent factors in the model.

Because the gene in the proposed method is simply encoded in binary, it is not very strict in the choice of crossover, mutation, and selection methods. If there is no domain knowledge, the probability of the crossover rate is recommended to be set as 0.8. However, the mutation rate should be set at 0.3~0.5, which is higher than the commonly recommended mutation rate in many applications of GA. SEM cannot calculate all solutions of GA. When there are unreasonable relationships in the model, SEM will return an error message indicating that the model cannot be calculated. We think these solutions are invalid. On this occasion, we order GA to return to the minimum value. As a result, a relatively higher mutation rate is set to enhance the calculation effectiveness.

4.2.4.3 Initial population generation and pre-evolution for finding out suggestions

This step is conducted to avoid GA being caught in a local extremum. The procedure of structure learning is conducted after EFA and CFA. The factors extracted by EFA and CFA accord with the correlations of the manifest items. As long as SEM can calculate the model, it will not obtain a very low value in fitting indexes, such as GFI of almost all solutions ranging between 0.8~1. The changing range of CEI is small, causing GA to be caught in the local extremum if no pre-processing is operated. However, if the extracted factor is confirmed, the strong or weak relations among the factors will be determined. Besides, the stronger relations that are established, the higher the fitness value. Thus, we create random initial populations and conduct multiple but fewer iterations to extract these strong relations. Here, the factor loading higher than 0.3 is thought as a strong relationship between two factors. Then we give suggestions to the algorithm.

For a suggestion, the genes presenting the strong relations are coded as "1," and other genes as "0." The suggestions should be inherited as the dominant population. The crossover and mutation in the dominant population help GA escape from the local extremum. It is unnecessary to pour all possible solutions with strong relations into the initial population. The final solution is not always the same as one or several of the suggestions. If there is no domain knowledge, three suggestions are enough.

4.2.4.4 Evolution steps

After finding the suggestions, a new initial population containing the suggestions is given to GA. The evolution procedures will stop when CEI is not improved after several evolutions or the program meets a set maximum iteration criterion. The solution (or solutions) is decoded as the path between factors, and every fitness index should be checked.

If the goodness of fit is acceptable, the next step of parameter learning will begin. Alternatively, if the collected data is not sufficient for building a model, the procedure should go back to data collection. For problem A, one of the results of structure learning is as follows.

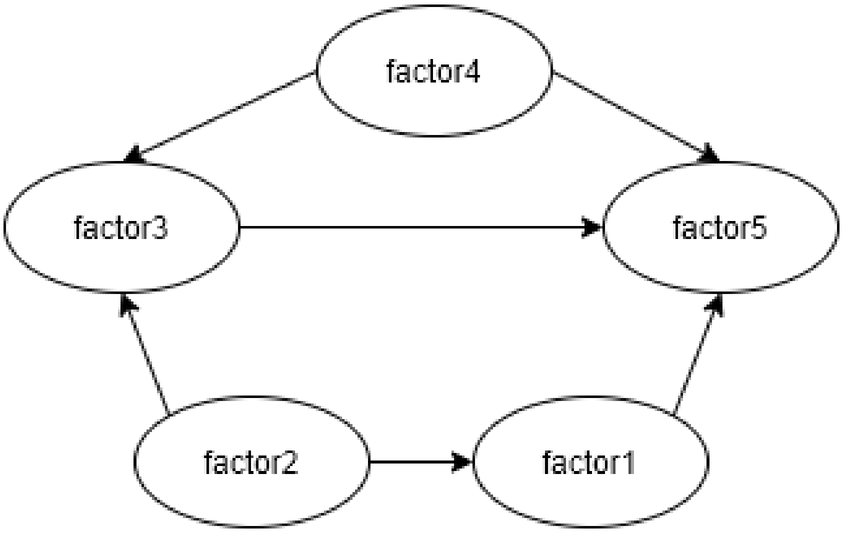


Fig. 4.4. Structure learning result

For a particular problem, there may be multi-solutions obtained from GA because CEI turns out to be the best fitness value of all these models. We call these possible solutions the candidate models. All the candidate models should be retained for the following steps.

4.2.5 Parameters learning

The parameter learning of the proposed model contains two parts. One is the structure simplification according to the factor loadings between factors. The other is a regression procedure for separating the learning target from the training data.

There are many methods for SEM to estimate the factor loadings, such as maximum likelihood estimation, general least squares, and asymptotically distribution-free methods. Different methods apply to different data distributions. For example, maximum likelihood estimation requires the data to approximate a normal distribution, whereas the general least squares method does not. The asymptotically distribution-free method can deal with missing data. Thus, before conducting SEM, a priori analysis of the normality of data is necessary. A suitable method should be selected accordingly. The same estimation method is used in the EFA procedure, structural learning procedure, and parameter learning procedure for maintaining consistency.

The factor loadings can be calculated using the estimation method, which represents the strong or weak relations among factors. The calculation is conducted using functions (4.1) ~ (4.3). In the proposed method, we define a factor loading ≥ 0.3 as showing two factors that have a relatively strong relationship. The factors that have factor loadings <0.3 with all the other factors are thought to have no efficacy for constructing the model. Furthermore, these factors and their contained items should be removed from the model. We call this procedure a structure simplification.

For example, in problem A, the factor loadings of factor 2 are lower than 0.3 regardless of other factors, so factor 2 and items 4, 5, and 6 ought to be removed from the model.

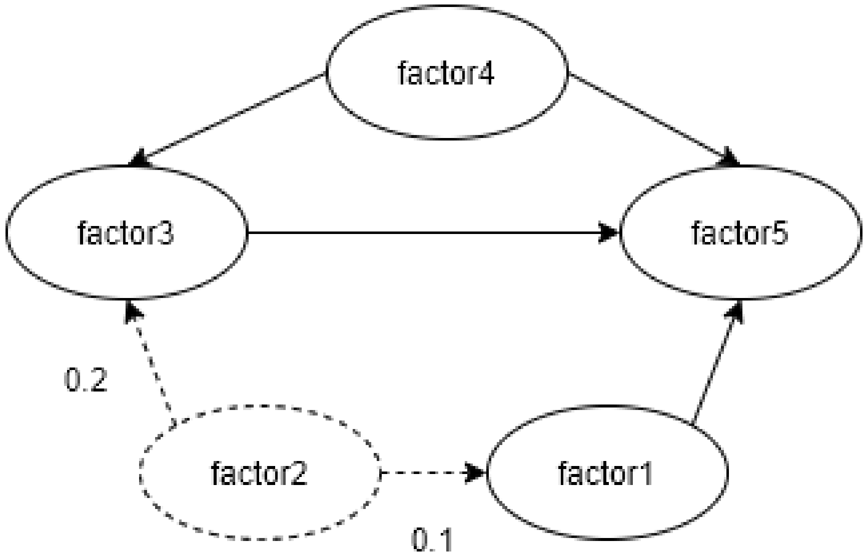


Fig. 4.5. Structure arrangement

As mentioned in Section 4.3.4, there may be multi-solutions obtained from the structure learning procedure. In this situation, the factor(s) in all the candidate models that have factor loadings <0.3 should be removed.

After the structure simplification, the selected estimation method is used once more to calculate the factor loadings, which can be used to analyze the relations between every two factors. However, for a ML problem, the purpose of the model is classification or prediction. The classification or prediction target is used as one of the manifest items in the built SEM model. Thus, a further step needs to be taken to extract the classification or prediction target and use other manifest items to estimate the target. For example, as shown in Fig. 4.6, item 15 is our classification target for problem A.

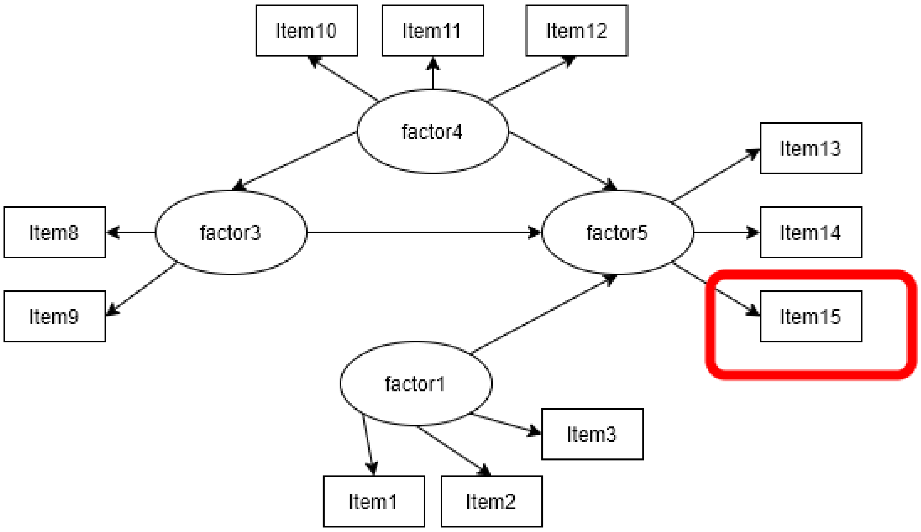


Fig. 4.6. Item 15 is the classification target

The estimation methods described above calculate the regression relations between factors and their contained items, which measures the measuring ability of each factor to its items. In contrast, SEM can also estimate the factor scores of each factor using the manifest items. In the shown example, the following function estimates the factor scores of the i^{th} factor.

$$FS\_i = \beta_i + \omega_{i\_1} * item_1 + \cdots + \omega_{i\_j} * item_j + \cdots + \omega_{i\_15} * item_{15} \qquad (4.5)$$

In function (4.5), β_i is the constant term, and ω_{i_j} is the regression weight of $item_j$ for $Factor\ i$. Maximum likelihood estimation is usually used here for estimating factor scores. As mentioned above, many candidate models may be obtained by the structure learning procedure. However, the models with the same CEI value turn out the same factor score calculation results. Thus, the parameter learning shows the same results of all the candidate models. Function (4.5) shows that for each factor score, the classification target, item 15, is used as one of the evaluation items for calculating factor scores. As a result, the SEM model cannot be used directly for a classification or prediction model. For using other items (training items) to learn the target item (predicting item), the proposed method conducts a multiple linear regression procedure using the training items on the factor scores. Then, the New estimated Factor Scores (NFS) are obtained in the presented example, as shown in function (4.6).

$$NFS\_i = N\beta_i + N\omega_{i\_1} * item_1 + \cdots + N\omega_{i\_j} * item_j + \cdots + \mathbf{0} * item_{15} \qquad (4.6)$$

Function (4.6) shows that only the training items estimate the NFSs. The target item 15 is released from all the factors. Also, the parameters, $N\beta_i$, the constant item for

$Estimated\ Factor\ score\ i$, and $N\omega_{i\_j}$ the regression weight for item j of $Estimated\ Factor\ score\ i$ can be obtained at the same time. Moreover, the final model can be built as shown in Fig. 4.7.

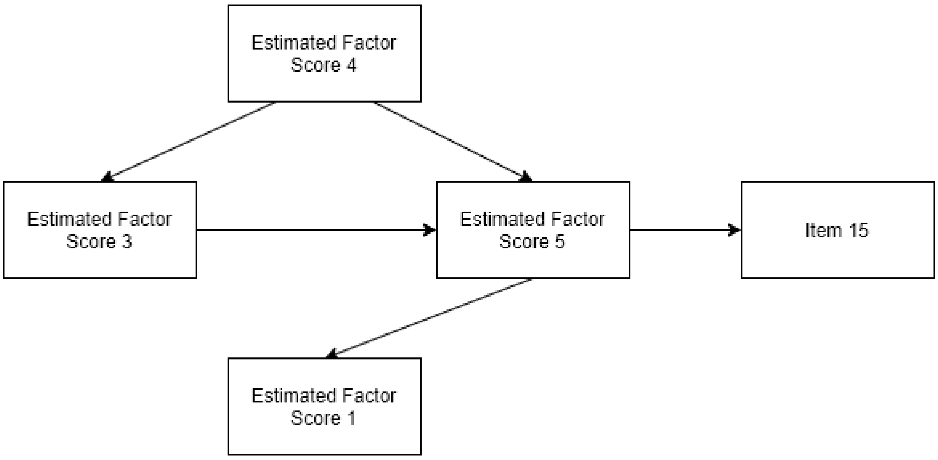


Fig. 4.7. Final model

4.2.6 Model utilization and validation

The model can be applied to different purposes, such as data analysis, machine learning, and causal analysis. A practical example showing the specific utilization of the proposed model will be presented in Section 4.4.

For different application purposes, the model should be validated from different aspects. For example, the goodness of fit is the most important evaluation index for the analysis model. The accuracy is the focal point for the ML model. The effectiveness of the intervention is the key to causal models. Besides, for an explainable and persuasive model, the model structure should be simple and easily understood by humans. Also, domain experts should accept its rationality. If the model cannot meet the mentioned requirements, data will need to be repeatedly collected.

4.3 Experiments

This section describes a practical application of the proposed method to data analysis, ML, and causal analysis on a common sleep disorder disease, Obstructive Sleep Apnea (OSA).

For testing OSA, the most precise device is Polysomnography (PSG) with a peripheral capillary oxygen saturation (SpO2) test. However, it is expensive and hard for people to use at home. Instead of professional devices, questionnaires are better choices to diagnose OSA in primary care and are self-diagnostic. Many kinds of questionnaires contain enormous amounts of questions about these three aspects, such as the Quality of Life (QoL) questionnaire, Epworth sleepiness scale, and Stop-Bang questionnaire. Much data is available, but it is impossible and not necessary to use all of these questionnaires at the same time.

On the other hand, the rationality of the model used by a healthcare problem must be recognized by the doctors. Thus, explainable models are necessary. A comprehensible model that humans can easily understand also enhances the ease of communication between doctors and patients. Considering the demands mentioned above, we explain how to apply the proposed method to provide a simple and useful analyzing, predicting, and causal analyzing model for the OSA problem.

4.3.1 Data preparation

Before collecting data, we review the factors relating to OSA. According to the recently published literature (Senaratna, Perret, *et al*, 2017; Mansukhanj, Kolla, *et al*, 2019; Mendelson, Bailly, *et al*, 2018; Chang, Baik, *et al*, 2018; Quan, Budhiraja & Kushida, 2018), OSA relates closely with the following aspects: age, gender, body mass index (BMI), sleep quality including daytime tiredness, snore, health status, and underlying diseases. Thus, we collected questionnaire data considering these factors—the data used for the analysis comes from the Sleep Heart Health Study (SHHS) database (Zhang, Cui, *et al*, 2018; Quan, Howard *et a*l, 1997). Apnea-Hypopnea Index (AHI) data can be made on the basis of PSG collection. Among all 5408 participants, 3931 subjects completed all data collection and had no history of OSA diagnosis. AHI ≥5 is an indicator of suffering from OSA. A total of 70% of subjects had an AHI≥5 in our study (3931 in total, 1863 males, 2068 females, age 63.7±11.3).

Additionally, there are 66 items collected from the self-rated questionnaires, including Anthropometrics (6 items), Health interview (11 items), Sleep habits and quality (41 items), and SF_36 questionnaires (8 calculated items). Besides, the AHI≥5 treated as undiagnosed OSA is the 67th item input to EFA explained by the next section.

4.3.2 Data management

EFA and CFA were conducted on the collected items. Table 4.2 shows the EFA results.

Table 4.2. EFA results

	Sn	SC	He	HBN	UD	UO
Ge	**-0.438**	0.172	-0.145	0.044	-0.199	-0.237
HoS	**0.866**	0.031	0.007	0.042	-0.099	0.119
HLD	**0.873**	0.012	0.024	0.056	-0.093	0.079
CS	**0.793**	-0.035	0.020	-0.007	0.000	-0.058
TFA	-0.098	**0.752**	-0.090	0.097	-0.012	0.032
WN	-0.014	**0.880**	-0.072	0.098	0.076	-0.024
WE	0.012	**0.817**	-0.030	0.058	0.064	0.000
RP	0.045	-0.025	**0.802**	-0.128	-0.206	-0.025
VT	-0.011	-0.190	**0.752**	-0.169	-0.012	-0.110
RE	0.083	-0.004	**0.787**	-0.058	-0.022	0.030
WC	0.043	0.097	-0.076	**0.699**	-0.007	0.041
CP	-0.019	0.077	-0.113	**0.811**	0.041	-0.036
SoB	0.025	0.070	-0.131	**0.825**	0.055	0.036
Age	-0.155	-0.040	-0.077	-0.066	**0.811**	-0.021
Hy	-0.046	0.008	-0.111	0.082	**0.605**	0.212
Nu	0.134	0.262	-0.032	0.080	**0.454**	-0.092
BMI	0.073	0.026	-0.122	0.058	-0.139	**0.812**
AHI	0.141	-0.012	0.032	-0.015	0.297	**0.709**

The meaning of the abbreviations in Table 4.2 are as follows: Sn: Snore, SC: Sleep Complaint, He: Health, HBN, Hard Breath at Night, UD: Underlying Disease, UO: Undiagnosed OSA, Ge: Gender, HoS: Snore Frequency, HLD: Loudness of the Snore, CS: Changes in the severity of the Snore over time, TFA: Frequency of having trouble falling asleep, WN: Frequency of Wake up at Night, WE: Frequency of Wake up Early and cannot go back to sleep, RP: Role-Physical index, VT: Vitality index, RE: Role-Emotion index, WC: Frequency of Woken by Cough, CP: Frequency of Waken by Chest Pain, SoB: Frequency of Woken by Short of Breath, Hy: Hypertension, and Nu: Nocturia.

The 18 items express a total variance of 62.33%, and the KMO test of 0.72. From Table 4.2, the EFA results show that 18 items are classified into six factors, and all variables have factor loadings higher than 0.3 to only one factor. Furthermore, we draw a hypothesis model using the extracted 18 items-6 factors and further evaluate the factor loadings using the CFA model, as Fig. 3.8 shows.

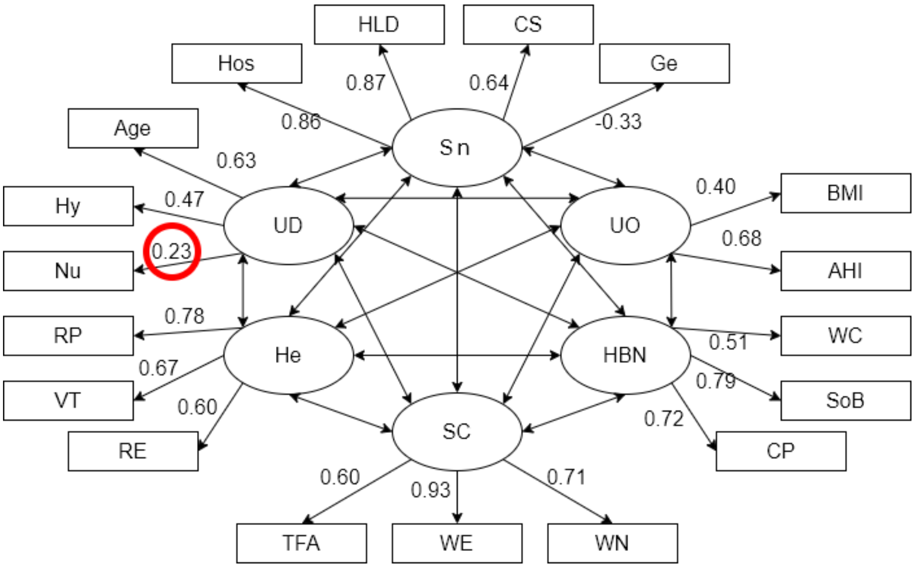


Fig. 4.8. CFA model

As shown in Fig. 4.8, the factor loading of Nocturia to the underlying disease is lower than 0.3, which is not favorable. After removing the Nocturia variable from the model, Table 4.3 shows the final factor loadings.

Table 4.3. Factor loadings

Measured Variable	←	Factor	Factor loadings
Ge	←	Sn	-0.328
HoS	←	Sn	0.861
HLD	←	Sn	0.867
CS	←	Sn	0.636
TFA	←	SC	0.600
WN	←	SC	0.931
WE	←	SC	0.708
RP	←	He	0.780
VT	←	He	0.667
RE	←	He	0.603
WC	←	HBN	0.511
CP	←	HBN	0.718
SoB	←	HBN	0.788
Age	←	UD	0.638
Hy	←	UD	0.459
BMI	←	UO	0.382
AHI	←	UO	0.708

The abbreviations in Table 4.3 have the same meanings as in Table 4.2.

4.3.3 Structure learning

The extracted six factors are used for structure learning. The GA procedure specifies the structural model. There are six factors, so every chromosome contains 30 genes encoded by "0" or "1." The crossover, mutation, and selection methods are chosen as Single-Point crossover, Uniform Mutation, and Linear Ranking Selection. Because there are only a few genes in each chromosome, the Single-Point crossover method is selected. For a binary encoding GA, there are not many kinds of mutation methods from which to choose, and Uniform Mutation is the most commonly used. Ranking Selection is mostly used when the individuals in the population have very close fitness values. The CEI is used as the fitness function in the presented application, which usually changes in a small range at the end of the run. Thus, Ranking Selection leads GA to better select parents in this situation.

After choosing the crossover, mutation, and selection methods, the pre-evolution is conducted for finding out suggestions. The result is shown in Fig. 4.9.

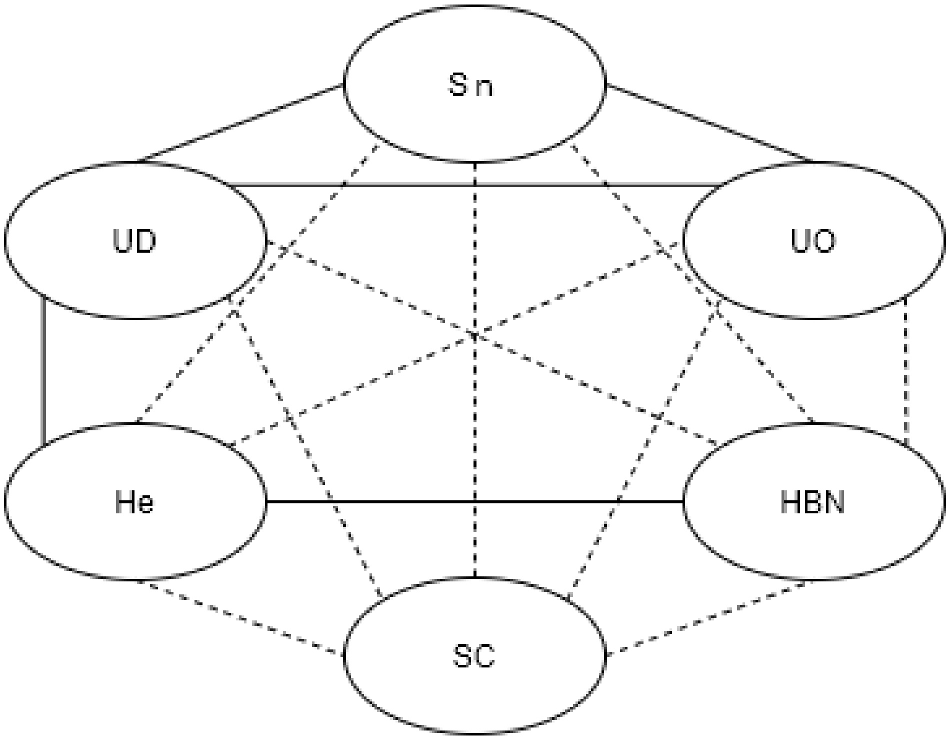


Fig. 4.9. Suggestions

From Fig. 4.9, three suggestions are chosen randomly with the full line parts coded as "1" and imaginary line coded as "0." Adding the suggestions to the initial populations with the parameters shown as Table 4.4 is given to GA.

Table 4.4. Parameters for the final evaluation

Population Size	Crossover Rate	Mutation Rate	Maxi	Max_Run
70	0.8	0.4	2000	300

GA is conducted 10 times, and three answers with the same CEI value, 23.925, are obtained. Fig. 4.10 shows the answers.

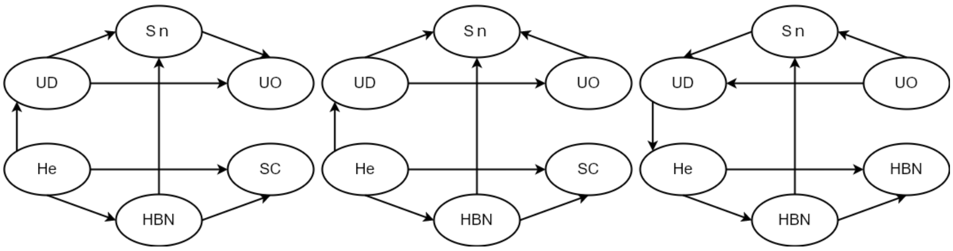


Fig. 4.10. Three candidate solutions

As shown in Fig. 4.10, the architectures of the models are the same, but parts of the arrow directions differ among the three candidates.

GA finds the best answer to CEI in the 560 generations. At the same time, AGFI and RMSEA also reach the extremum. The values of GFI, CFI, and Chi-square are the second-best ones, which is acceptable. As mentioned in Section 4.3.4.1, the goodness of fit is not the only target for structure learning in the proposed method, and we also hope a simpler structure can be obtained. AGFI and RMSEA consider the freedom degree of the model, and the better the two indexes are, the simpler the model will be. Thus, the results of GA in the presented example prove that utilizing CEI as the fitness function is effective. The value of the goodness of fitting is shown in Table 4.5.

Table 4.5. Goodness of fitting

GFI (>0.90)	**CFI (>0.90)**	χ^2	**AGFI (>0.90)**	**RMSEA (<0.06)**
0.967	0.941	1095	0.954	0.048

As shown in Table 4.5, all the indexes show that the three candidate models fit well.

4.3.4 Parameters learning

First, factor loadings are calculated to verify if any factors do not have strong enough relationships with others. The results are shown in Fig. 4.11.

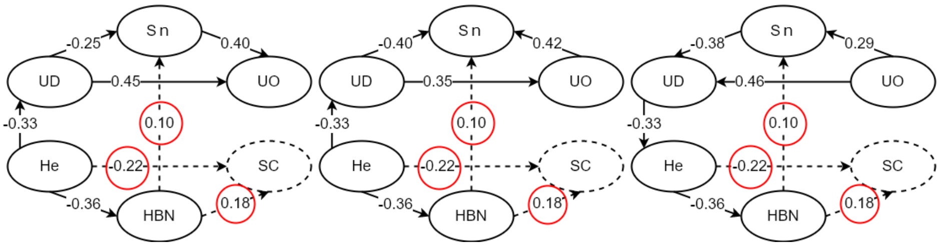


Fig. 4.11. Factor loading verification

As shown in Fig. 4.11, the relations in the red circles of all three candidates are lower than 0.3, which presents Sleep Complaint (SC) does not have strong relations with any other factors. As a result, SC and its contained manifest items are removed from the dataset. The remaining 14 items and their corresponding factors are shown in Table 4.6.

Table 4.6. Retained Items and Their Corresponding Factors

Items	Content	Factor
Age	-	Underlying disease
Gender	1: Men; 2: Women	Snore
BMI	Calculated by height and weight	Undiagnosed OSA
Snore Frequency	Snore frequency	Snore
Loudness of Snore	Snore loudness	Snore
Change in snore	Snore becoming stronger or weaker	Snore
Woken by Cough	Frequency of waking up due to a cough	Hard Breath at Night
Woken by Chest Pain	Frequency of waking up due to chest pain	Hard Breath at Night
Woken by Short of Breath	Frequency of waking up due to shortness of breath	Hard Breath at Night
Hypertension	Hypertension is present or undertreated by hypertension medicine	Underlying disease
Role-Physical	The role-physical score calculated from the SF_36 questionnaire	Health
Role-Emotion	The role-emotion score calculated by the SF_36 questionnaire	Health
Vitality	Vitality score calculated by SF_36 questionnaire	Health
AHI≥5?	**Apnea-Hypopnea Indexes calculated from PSG**	**Undiagnosed OSA**

As shown in Table 4.6, the item intended to be analyzed or predicted is AHI, one of Undiagnosed OSA's manifest items (UO). Thus, in the next step, a regression procedure is conducted using the other 13 items with their corresponding factor scores calculated by the candidate models. As mentioned above, all the candidate models have the same fitting results, so their parameter learning results are the same as well. By using the learned regression weights and the 13 items (items are shown in Table 6 except AHI), the estimated factor scores can be calculated. Furthermore, the final models made up by the estimated factors and AHI are shown in Figure 4.12.

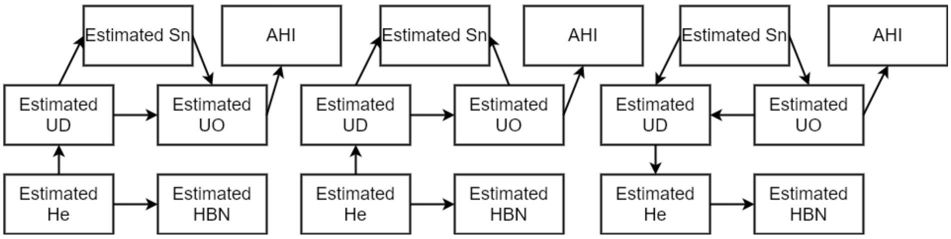


Figure 4.12. Final models

As shown in Fig. 4.12, three final candidate models are obtained. The validation of the fitting indexes are shown in Table 4.7.

Table 4.7. GFIs of final models

GFI (>0.90)	CFI (>0.90)	χ^2	AGFI (>0.90)	RMSEA (<0.06)
0.993	0.990	82	0.984	0.0453

4.3.5 Model utilization and validation

4.3.5.1 Data analysis

By using maximum likelihood estimation, the standard regression weights between every two factors are calculated, and results are shown in Fig. 4.13.

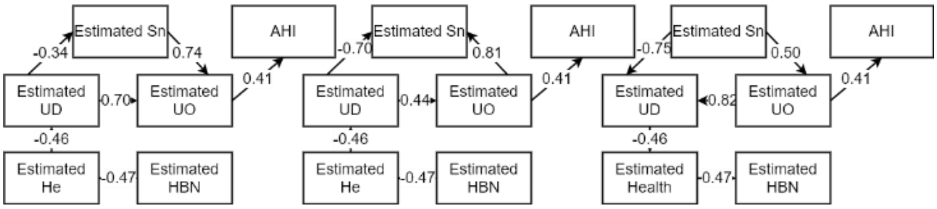


Figure 4.13. Analysis models

First, Snore and Underlying Diseases directly affect OSA, and Health and Hard Breath at Night affect OSA indirectly. Additionally, the factor loadings of Health factors with the other factors are negative, which indicates that health status indirectly reflects the probability of having OSA. The worse one's health, the higher the probability of suffering from OSA.

Considering the analysis described above, a new screening tool to evaluate the risk of having OSA has been created by our team.

4.3.5.2 Machine learning model

The purpose of this application is to predict whether AHI≥5. In the previous steps, 13 items were extracted, which can be used to estimate the factor scores. The proposed method uses the estimated factor scores to predict AHI. We validate the model from two aspects: prediction ability and structure effectiveness.

(1) Prediction Ability

An effective model with high prediction ability requires the model to extract useful features from the dataset accurately and classify the target with high accuracy. Decision Trees and its variances are commonly used methods that can simplify data dimensions and extract useful features. They also provide transparent models. In this part, we use three kinds of Decision Trees and its variants (the ordinary Decision Tree (DT), Bag-ensembled Random Forest (BRF) (Breiman, 2001), and AdaBoost-ensembled Random Forest (ARF) (Freund & Schapire, 1997) to make classification models for AHI and compare them with the proposed model.

As shown in Fig. 4.12, no matter which candidate model is used, Undiagnosed OSA is the only factor measuring AHI. We classify the estimated Undiagnosed OSA score to predict AHI. The unsupervised classification method, CSCDFCM, which will be described in Chapter 4. Simultaneously, we conducted Decision Trees to extract 13 items with the highest importance of the 66 items. The extraction results are different from those of the proposed method. Table 4.8 shows the extraction results of the three Decision Tree methods.

Table 4.8. Items extracted by Decision Trees

DT	BRF	ARF
Age	Age	Age
Height	Height	Height
Weight	Weight	Weight
BMI	BMI	BMI
Fall asleep while watching TV	Physical Function	Physical Function
Cups of coffee drunk every day	Mental Health	Mental Health
General Health	General Health	General Health
Vitality	Vitality	Vitality
Minutes to fall sleep	Minutes fall into sleep	Minutes fall into sleep
Time wake up on weekdays	Time wake up on weekdays	Time wake up at weekday
Time wake up at the weekend	Time wake up at the weekend	Time wake up at the weekend
Snore Frequency	Snore Frequency	Snore Frequency
Neck circumference≥40cm	Neck circumference≥40cm	Neck circumference≥40cm

Moreover, Table 4.9 shows the accuracy, F1_score, and the sensitivity of the positive of the three Decision Tree models and the classification result of the proposed model. All models conducted 5-fold cross validation.

Table 4.9. Comparison of the accuracy

Method	Accuracy	F1_score	Sensitivity
DT	67.7%	0.614 [0.46, 0.77]	77.7%
ADT	72.8%	0.657 [0.47, 0.82]	86.4%
BDT	74.1%	0.668 [0.47, 0.83]	89.4%
Proposed model	**74.5%**	**0.672 [0.48, 0.83]**	**90.0%**

As shown in Table 4.9, the proposed model obtained the best accuracy and F1_score, which proves it is more effective as a ML model than the similar explainable model, Decision Trees. Additionally, for a healthcare problem, doctors care about the sensitivity of the positive rate, and the proposed method reaches 90%, which is ideal.

(2) Structure effectiveness

This experiment aims to test the structure effectiveness of the proposed method. As shown in Fig. 4.12, three candidate models are built. Applying the candidate models to BNTs, six factors are the nodes, and the arrows are arcs building up the network. As the estimated factor scores are continuous numbers, CSCDFCM is conducted to the factor scores for discretizing the data. Furthermore, the estimated factor scores are the evidence used for interfering AHI.

There are three candidate models obtained from the proposed method. The above sections discussed that the estimated scores of the factors are the same in different candidates. Also, the structures of the three candidates are the same, and only a few directions of the arrows are different from each other, which does not affect the interference result of BNT. Thus, when applying the candidate models to BNT, the same result of prediction is obtained.

Besides, K2 is a commonly used method to train structures for BNTs. However, K2 requires domain knowledge to offer the order of nodes to the algorithm. Let us number the nodes of the factors as Hard Breath at Night: 1, Health: 2, Snore: 3, Underlying Disease: 4, Undiagnosed OSA: 5, and AHI: 6. We randomly put them in two orders, [1,6,3,2,4,5] and [6,1,4,2,5,3]. The structures trained by K2 are shown in Fig. 4.14.

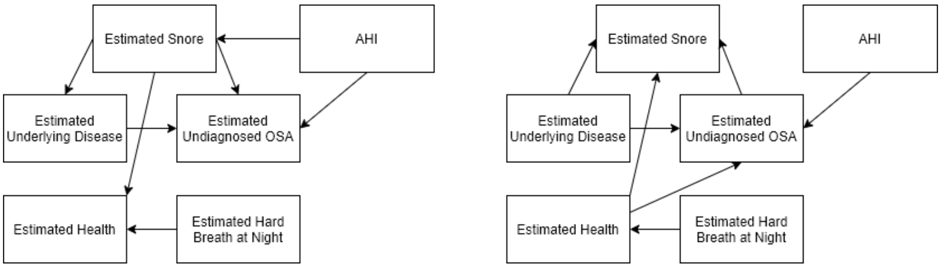

Estimated Snore
AHI
Estimated Underlying Disease
Estimated Undiagnosed OSA
Estimated Health
Estimated Hard Breath at Night
Estimated Snore
AHI
Estimated Underlying Disease
Estimated Undiagnosed OSA
Estimated Health
Estimated Hard Breath at Night

Fig. 4.14. Structures trained by BNTs

Fig. 4.14 shows that the structures trained by BNT under different orders of nodes are different from each other, and Table 4.10 compares the interference accuracy of AHI on the BNT trained structures and the proposed model structures.

Table 4.10. Comparison of proposed method and BNT

Method	**Accuracy**	**F1_score**
BNT structure 1	73.7%	0.655 [0.41, 0.83]
BNT structure 2	74.2%	0.663 [0.43, 0.83]
Proposed structure	**74.6%**	**0.675 [0.49, 0.83]**

The proposed model structure has the highest accuracy among the three. The results also show that the structures trained by BNT models only present the probability dependency of the nodes, but there is no way to train a reasonable BNT model without domain knowledge. For example, according to the analysis by the proposed model, there are no direct relations between Health and Snore (factor loading between them is lower than 0.3). However, there is a strong relationship between Health and Underlying Disease, and Health affects Snore indirectly through Underlying Disease. However, in the two models trained by BNT, wrong information is transferred by the structure.

This experiment shows that the proposed method can automatically apply a simple, reasonable, and effective model structure to BNT networks. There is no need for human experts to participate in the procedure of constructing the model, so much time and labor can be saved.

4.3.5.3 Causal models

Another function of the proposed model is to analyze the causal relationships among factors. Although statistical dependency between factors can be obtained from the models shown in Fig. 4.13, they cannot reflect the actual causal relationships for which model surgery is necessary. Introducing do(calculus) to the three candidate models, the intervention models can be obtained. We use one of the candidate models to illustrate the model surgery procedure. The other two are similar.

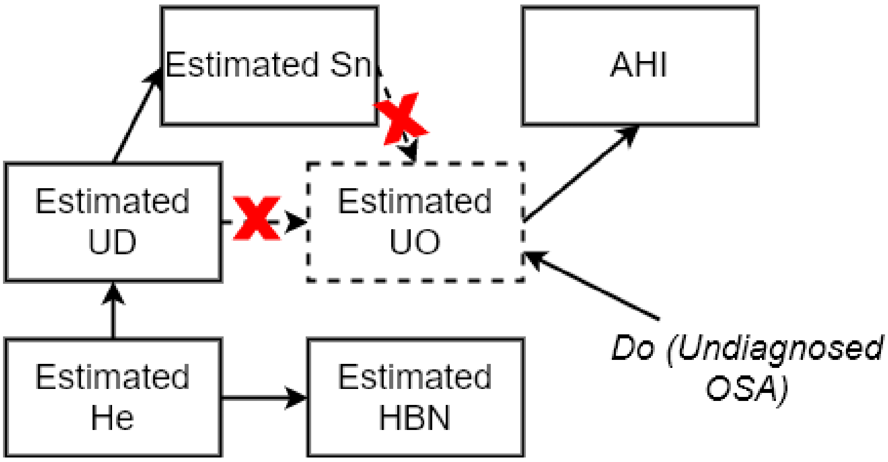


Figure 4.15. Invention model of Do (Undiagnosed OSA)

Fig. 4.15 conducts do (Undiagnosed OSA) for the OSA factor, so the connections between OSA and Snore and the Underlying Disease should be removed. Furthermore, the process of human intervention is conducted to OSA, such as medical treatment. If the causal relations in this model are true, no change will happen in Snore or Underlying Disease. Similarly, conducting do (Snore) and do (Underlying Disease) for the other two candidate models leads to different conclusions.

Doctors can determine the most suitable treatment plan for patients by analyzing the causal relationships, especially when the existing data is insufficient. The presented example shows three kinds of possible causal models. All three factors (Underlying Disease, Snore, and Undiagnosed OSA) can be reasons or results. However, fewer or more candidate models may be obtained from the other applications.

4.4 Discussions

With the development of ML technology and the accuracy of learning, the understandability between humans and machines is being paid more attention. Machines are hoped to imitate human behavior as closely as possible so that humans and machines can collaborate better or even mutually improve. For achieving human-machine understandability, the structures of the learning procedure have to be shown in front of the human eyes. In other words, the degree of explainability of a model is the premise for mutual understanding between humans and machines.

Several existing ML technologies were developed with the explainability, such as Decision Tree methods, BNTs, and their variants. However, some defects of these methods limit their application in practical cases. For example, Tree-type methods judge the necessity of the data features used for prediction by comparing the importance weight of the training data. The Trees cannot express the dependency relationship among the chosen data features, so the reasonability of the inference has no way to be estimated. The partial explainability makes the accuracy of the Tree-type methods dissatisfactory. In the other category, the BNTs methods, although the inference structures are clearly shown, the construction of the structure relies on the domain experts' knowledge. As shown in this paper's medical case, BNTs are incapable of creating the correct structure without prior knowledge. The inference structure's validity affects learning accuracy and relates to the further application, the causal analysis. In the field related to people's life and property, such as medicine and economy, causal analysis is an indispensable means to predict the future. The proposed method provides an explainable ML model from design to application. The structure is transparent, and rationality can be guaranteed, which endows the model with multi-functions with high quality, including data analysis, machine learning, and causal analysis.

4.5 A brief summary

The presented chapter proposed an explainable machine learning model by introducing Structural Equation Modeling to the problems. The model is transparent and interpretable from design to application. The human user can recognize the rationality of the model structure so that credible data analysis, ML, and causal analysis can be conducted simultaneously. An application example in the healthcare field shows the practice effectiveness of the model.

Although comparing with other explainable models, i.e., BNT, the proposed model depends much less on domain expert. The human expert knowledge is still critical in the data collection and EFA procedures in the proposed SEM-EML model. The learning outcomes of an ML depends on the quality of data collection to a great extent. Under the guidance of domain expert helps to save much time and reduce the risk of repeated optimization because of the incorrect information in the data. The utilization of expert knowledge makes it possible for ML improving the effectiveness, as well as for human understanding and trusting the learning model, which is so called as the reliability of ML.

In the next chapter, an application of SEM-EML is presented showing that except for experience, causality is another useful knowledge that guides ML to learn from insufficient data.

Chapter 5. A Relational Feature Transfer Learning Method led by causal knowledge from the domain expert

In chapter 4, an example of using causality to post-explain the ML model was shown. In this chapter, we present a Relational Feature Transfer Learning method, in which the causality directs the transferring procedure.

This chapter is organized as follows. Section 5.1 described the backgrounds of the Relational TL technologies. Section 5.2 gives technical background, including an overview of relations between the Knowledge Graphs (KGs) and SEM, and explains how SEM can contribute to constructing KGs for domain knowledge. It also gives a brief introduction to transfer component analysis (TCA) (Pan, et al, 2010) and CORelation Alignment (CORAL) (Sun, Feng & Saenko, 2016), which we selected as methods for comparison in the experiments. Section 5.3 describes the procedure of RF-TL. Section 5.4 shows an experiment we conducted to evaluate the effectiveness of RF-TL when it was applied to healthcare problems, and Section 5.5 discusses the results. Finally, we give concluding remarks in Section 5.6.

5.1 Backgrounds

Relational TL accounts for the relationships among data features, and the transferred objects are the logic networks in D_S. It assumes that the knowledge networks in D_S and D_T are the same or can be transferred from D_S to D_T. Two critical issues affect the development of relational TL: 1) how to extract knowledge networks from the data of the original domain and how to transfer knowledge networks from one domain to another.

Knowledge graphs (KGs) are useful tools for dealing with the first issue. KGs are knowledge bases that use data and logic to structure information. They are often used to store interlinked descriptions of entities with free-form semantics (IBM Cloud Education, 2021). KGs express not only statistical relationships among data but emphasize the human reasoning involved in the knowledge representation. According to (Bimba, Al-Hunaiyyan, et al, 2016; Chen & Luo, 2019), knowledge-based modeling manipulations are categorized into ontologies, cognitive knowledge bases, linguistic knowledge bases, and expert knowledge bases. Although expert-knowledge-based modeling methods have been criticized for their heavy reliance on expert experience, such experience and knowledge constitute an indispensable gold standard for validating models (Li, Wang, et al, 2020; Cheng, Zhang et al, 2018; Shi, Wang et al, 2020). Peng and his team (Peng, Wang, et al, 2019) proposed a hyper-network-based approach to retrieve data and reasoning with engineering design knowledge. Bayesian inference has been used for constructing the KGs. In (Rotmensch, Halpern, et al 2017), a Bayesian network with noisy OR gates was used to extract a health knowledge graph from Electronic Medical Records (EMRs). Bayesian-based technologies have been widely utilized for making KGs because of their intuitiveness and interpretability. However, Bayesian-based models depend on probabilistic inference, which cannot explain the correlations and causalities among data; this limits their application to KGs involving causal logic. In this chapter, we referred to the key procedures in SEM-EML described in chapter 4 and introduced SEM in TL technologies to extract KGs from data as a preparation for transfer learning.

Another essential issue with relational TL is the ways of transferring. Unlike instance-based, parameter-based, and feature-based TL, the difference between D_S and D_T is easily expressed mathematically, such as the distance between data features across domains. However, the difference in the relational structure between D_S and D_T is hard to describe statistically. That is, the transference of a relation needs support from a human expert. To the best of our knowledge, there are few algorithms for transferring “relations” (Omran, Wang, et al, 2016; Kumaraswamy, Odom, et al, 2015; Kumaraswamy, Ramanan, et al, 2020). Kumaraswamy et al. (2020) developed an interactive TL algorithm in relational domains, called language-bias transfer learning (LTL), that uses tree-type inductive logic programming. The transference procedure of LTL entirely depends on a human expert's experience assisting the algorithm to select appropriate relations to transfer, which is time-consuming and laborious. Instead of interacting with expert experience, a more efficient way is to teach the algorithms to imitate human cognition. A number of cognitive factors have been identified as being involved in the support of transferring empirical engineering knowledge (Wang, Jiang et al, 2021). In particular, causality, as a human inference logic, has attracted attention from researchers as ways for assisting and directing machine learning. Analogical reasoning, the well-known feature-mapping method proposed by Gentner and his team (2012), is a helpful tool for inferring relational structures from one domain to another. Gentner discussed that attention to the differences in objects between domains leads to the inference on the relationships among the objects. Through the procedure of analogy, features in one domain can be mapped to another one. Gentner's method stresses the similarities of relational structures in different domains. However, distinctions between domains were ignored. The mapping or transference should not be a static contrast but rather a dynamic process. In the presented study, we take advantage from another aspect of causality, counterfactual inference, which is able to guide the dynamic process of feature transference across domains.

A causal relationship is recognized as ground truth, and a change in the reason will cause a corresponding change in the result. In machine learning, the reason is a stimulus given to a model. The result is a change in the model produced by the stimulus. Furthermore, in causality theory, a prediction that if the same stimulus is experienced in the future, the model will change is called a counterfactual inference. The task of relational TL algorithms is to predict the unlabeled target in a domain by transferring a relational structure from another domain. If the relational structure changing rules from the source domain to the target domain can be inferred from a piece of particular causal knowledge, it will be feasible to predict a model in the target domain.

Using causality as a guide for the learning procedure in ML is efficient because the only information supplied by a domain expert is a piece of causal knowledge. The causal TL algorithm proposed by Rojas-Carulla et al. (2018) uses SEM for finding the invariant domain between D_S and D_T. Roughly speaking, the algorithm uses the invariance of the reasons in a causal relationship to find conjunct causal features in the two domains. However, it focuses on how to extract causalities, not how to transfer knowledge.

This paper proposes to use counterfactual inference to predict causal knowledge graphs from the source domain to the target domain for relational transfer learning. We name the algorithm we use for inference Relational Feature Transfer Learning (RF-TL). The counterfactual inference is made according to the causal knowledge provided by a domain expert, which predicts the relations among X_T from the relations in X_S. Moreover, other ML methods are used to label the data in D_T using the extracted features.

5.2 Technical background

5.2.1 Structural Equation Modeling for constructing knowledge graphs

The domain knowledge that is used for solving problems is expressed as rules in KGs. The rules are made up of IF and THEN parts. The IF part can include first-order logic expressions, e.g., the conjunction AND or disjunction OR. Nodes in KGs consist of linguistic objects and their values. Rules represent relations among nodes and can be classified as logical or fuzzy (Chen, Jia, et al, 2020). At present, domain knowledge is mostly acquired from domain experts, while automatic or semi-automatic methods have been proposed for saving labor and time (Kim & Raghavan, 2000; Tenorth & Beetz, 2013).

In the proposed RF-TL, we use SEM-EML mentioned in chapter 3 to obtain the structure of KGs from empirical data. As a way of measuring correlations among data points, the utilization of SEM for constructing KGs makes it possible to add properties to “edges,” i.e., IF A is a AND B is b, THEN A strongly (weakly) results in B, which extends the usable range of KG expressions on domain knowledge. Also, SEM's strong point over other information integration methods, e.g., Bayesian networks, is its ability to measure causalities between factors (corresponding to nodes in KGs). The notion of causality lets a static binary relationship between nodes, e.g., IF A is True, THEN B is true, acquire dynamic properties, e.g., IF A changes, THEN B will change. Dynamic properties are essential to KGs, without which KGs can only store and express "data from the past" but never predict the future. As the application in this study, we describe transference as a dynamic procedure that requires KGs to cope with change.

5.2.2 Feature-based transfer learning methods

Besides knowledge network extraction, another critical problem of transfer learning is how to transfer the relationships from the source domain to the target domain. As mentioned in section 5.1, several methods can be chosen depending on the transfer objects. In this study, we focus on feature-based transference. TCA and CORAL are representative feature-based TL methods and are briefly introduced here. Section 5.4 describes experiments that compared their performance with that of the proposed algorithm.

TCA maps data features in D_S and D_T into a high-dimensional reproducing kernel Hilbert space, where the distance between the data features in the marginal probability distributions over D_S and D_T is minimized while preserving their respective internal properties to the greatest extent. TCA extends the principal component analysis to TL, and TCA and PCA's core ideas are similar. In the transformed feature space, only the principal components are needed to be preserved. We call this idea dimensionality reduction. As mentioned in section 1, although the user can decide the number of dimensionalities that remains after TCA, it is hard to choose an appropriate number without prior knowledge. Also, the number of dimensions influences the accuracy of learning to a great extent. Our experiment in section 5.4 shows how the decision on the dimensionality number affects learning accuracy. Moreover, we show that RF-TL does not have this selection problem.

Different from TCA that transforms data features in both D_S and D_T into another space, CORAL transforms only X_S to D_T and uses the transformed X_S to train a model in D_T. The basis of CORAL is to extract correlations among data features and then transform the covariance matrix from the source domain to the target domain. On the one hand, while the distributions of data features are not so different from one domain to the other and they correlate strongly in each domain, CORAL fails to reduce the dimensionalities. On the other, two data features with strong correlations do show they have a particular relationship with each other while no causal relationships are interpreted. Correlations cannot tell us how one data feature changes in correspondence to a change in another data feature's change. While is not a problem to use data features with solid correlations to train a machine learning model, in TL, the transference is an automatic procedure. It is necessary in TL to predict the change in a model when data features in the source domain are changed to the target domain.

Thus, the learning structures expressing causal knowledge must be known in the source domain so that the correct transference of the model to the target domain can be conducted. Here, SEM is an excellent tool for extracting causal knowledge from data, and it is used in RF-TL.

5.3 Main proposal of Relational Feature Transfer Learning algorithm

The overall design of RF-TL is shown in Fig. 5.1. The core idea of RF-TL is to use causality to direct the counterfactual inference from D_S to D_T. An explainable model structure is necessary regardless of whether one is conducting the causal analysis or counterfactual inference. For training the source model, expert knowledge should be used as a measuring item(s) of the model so that in the next step, an intervention can be performed on the model. Furthermore, after extracting the knowledge network from the intervened sub-models, RF-TL uses counterfactual inference to predict the KG(s) carrying the information on features useful for D_T. The next sections illustrate the specific procedures of each step, including the role of the causal relationships among them.

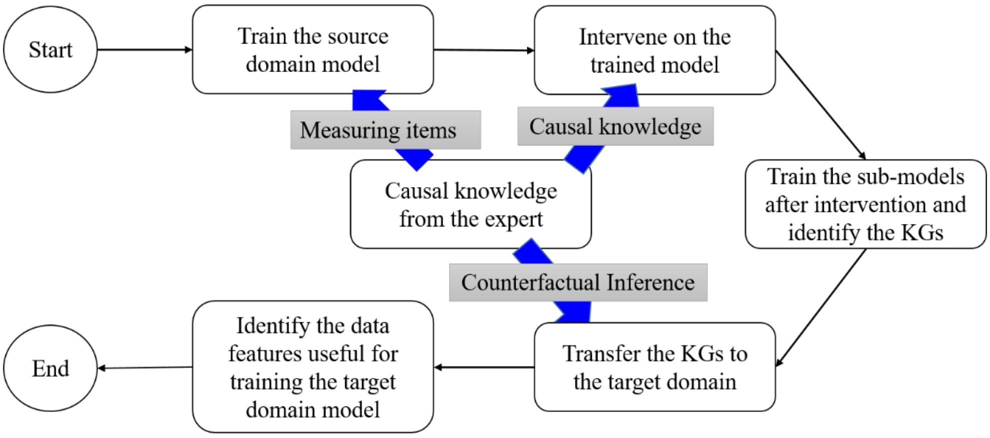
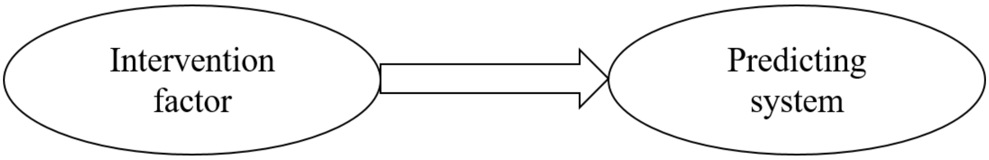

Start
Train the source domain model
Intervene on the trained model
Measuring items
Causal knowledge
Causal knowledge from the expert
Train the sub-models after intervention and identify the KGs
Counterfactual Inference
End
Identify the data features useful for training the target domain model
Transfer the KGs to the target domain

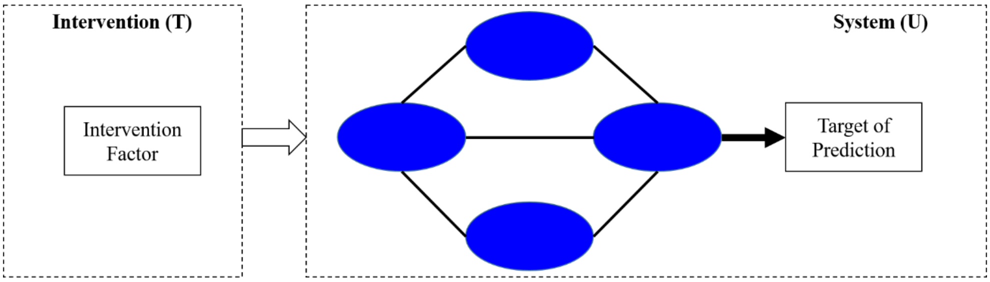
Fig. 5.1. Overall design of RF-TL

5.3.1 Causal relationships derived from expert knowledge

Causality is a philosophical concept. When two events occur in a certain time order, one event has an impact on the other. The event occurring earlier is the reason and the event occurring later is the result. An “Order” is very important for actual causality (Gebharter, 2017). Introducing causality theory to ML usually involves adopting the interpretation of interventionism. In interventionist-causality theory, an intervention is regarded as a reason, and the corresponding changes in the system are the results (Imben & Rubin, 2015). Fig. 5.2 (a) shows the concept of interventionism-causality.



(a) Concept of interventionism-causality



(b) Mechanism of interventionism-causality

Fig. 5.2. Intervention-causality theory

Causality is regarded as a factual truth in the real world. In a causal model, the direction of the arrow is non-reversible, which also clarifies the essential difference between causality and correlation. When we talk about two events being statistically correlated, we can only show that the two events have a particular relationship. However, there is no illustration about the "order" or which one impacts the other. In other words, causality is a ground truth or customary rule and is higher in some sense than the level of a statistical relation. In the interventionism-causality system, the intervening factor (the reason) is objectively variable and will lead to a corresponding change in the predicting system. There are many cases in real life where this theory applies. For example, the risk of getting a disease such as hypertension and diabetes becomes higher with increasing age. A change in a population will influence the economy. In a production line safety assessment system, the temperature of the environment is an essential factor affecting the safety risk.

However, a commonality of the above-mentioned cases is the bias in data collection caused by objective facts. Sometimes, the collection of global data is impossible or inhumane. For example, data on diseases that occur more frequently in older age groups are scarce from young people. It is impossible to artificially make the young age quickly to get an age-wide predictive system. Similarly, it is unrealistic to change the population structure of a society in a short time. However, using existing data and by taking advantage of interventionism, we can observe a change in a system caused by an intervention factor. Furthermore, we can transfer the model constructed using existing data to the domain in which we want to predict. The details of the interventionism-causality mechanism are shown in Fig. 5.2 (b).

For the sake of illustration, suppose that we are to design an attendance forecasting system for baseball games. Baseball is usually not played in winter conditions, but the client wants to predict the attendance rate in winter. In this case, we define weather temperature as the intervening factor. Thus, the source domain D_S including data features in summer, and D_T represents the winter event.

Generally, in interventionism-causality, an intervention (*T*) is a stimulus applied to a system (U). The state (*Y*) of U changes in accordance with the stimulus. The intervention procedure is expressed as $\delta(u) = Y_t(u) - Y_c(u)$, where $Y_c(u)$ is the original state of U and $Y_t(u)$ is the state after the intervention. Using the baseball game prediction case mentioned above, we consider that temperature is the reason for the attendance rate. Then, if there is a system that can infer the attendance rate, the state of this system will respond accordingly to temperature intervention.

In practical applications, we would like to know the effect of an intervention on multiple systems, e.g., the effect of temperature on the decision to attend by a group of people. The following equation can be used to determine this effect

$$E[\delta(u)] = E[Y_t(u)] - E[Y_c(u)] \tag{5.1}$$

where $E[\cdot]$ represents the average state of a group of individuals.

However, in practice, it is difficult to obtain accurate information on the state Y of a group of people, which is called the fundamental problem of causal inference (FPCI) (Imbens & Rubin, 2015). In this case, it is impossible to ask every person in the world whether they would attend a game in winter. FPCI embodies the difficulty of determining $Y_t(u)$ and $Y_c(u)$ at the same time. In particular, three assumptions constrain the interventionism-causality (Imbens & Rubin, 2010): A) the stable unit treatment value assumption (SUTVA) regards every individual change as an independent event; B) the assumption of constant effect (CEA) supposes that the effects of an intervention are the same for every individual. That is, $\delta(u_i) = \delta(u_j)$ if i and j are different individuals in the same group; C) the assumption of homogeneity (HA) is such that $Y_t(u_i) = Y_t(u_j)$ for two individuals. Under these three assumptions, it is easy to estimate the effect of an intervention on a group of objects. Our RF-TL follows these three assumptions.

The next step after constructing a causal model is to carry out counterfactual inference. “Counterfactual” means the fact has not occurred but can be predicted according to certain evidence. The most important message conveyed from the causal model is that a change in reason will cause a change in the result, but the reverse is not true. Therefore, counterfactual inference can be made as if the “reason” will change in the future, changing the “result” correspondingly. Coming back to intervention-causation, we could say that “if a certain intervention is carried out on a model, the system will obtain $\delta(u)$”. Note that $\delta(u)$ only represents the change in the state, so it can be quantitative or qualitative. In the case of RF-TL, $\delta(u)$ is used as the transfer rule, which means it is qualitative. In the baseball game example, $\delta(u)$ can be obtained by intervening on temperature. Furthermore, counterfactual inference can be performed as “if there is an intervention on temperature, then the predicted attendance will change according to the rule(s).” Similar to the baseball game example, the main idea of RF-TL is to extract the “rule(s)” from the intervention conducted on the D_S model and make a counterfactual inference to transfer the knowledge network to D_T in accordance with the “rules”.

In the following sections, we will describe the approach for KGs extractions using an SEM-based method. Then we will show the specific steps of RF-TL from training the source domain model to the transference of KGs from D_S to D_T.

5.3.2 Translating Structural Equation Model into knowledge graph

As mentioned in section 5.1, SEM is a valuable tool for digging into statistical causal relations in data. SEM is usually framed as a two-step procedure. The first step is an exploratory factor analysis (EFA). The other is a confirmatory factor analysis (CFA). EFA is a reliable tool for classifying data items into corresponding factors without a specific hypothesis, which aims to identify latent factors based on the observed variables. The measurement model and structural model make up the hypothesis for CFA to test. EFA yields extracted factors and their inclusive manifest variables that constitute the measurement model. The structural model specifies the logic paths among factors. Once the model is constructed, the factor loadings between manifest items and latent factors and between every two factors are estimated in accordance with the covariance matrix of the manifest items

In this study, we use SEM to construct KGs. Because the original SEM is a data analysis model, in order to use it to extract KGs, it has to be modified with several further operations.

First, we need to transfer SEM into a predictive system. The main steps are shown in chapter 4 on SEM-EML. Roughly speaking, they include data collection, data management, structure management, and parameter learning. A common problem of SEM is that the validation of the model relies on a convincing hypothesis given by a domain expert, which is sometimes impossible or involves labor and time to obtain. In our approach, the strategy is adopted to optimize the structure of SEM. In the structure management procedure, to guarantee the model's validity, we use a genetic algorithm (GA) to identify the fittest model by setting goodness of fit (GoF) indexes. In the final step, the target of the prediction item is separated from other items by using a linear regression procedure.

Next, the obtained SEM-like predictive system is translated into KGs. The origin of using KGs can be traced back to the semantic network developed in the 1970s (Tao & Huakang, 2017). In particular, GOOGLE used a KG to enhance the performance of its search engine in 2012 (Singhal, 2012). There is no gold-standard definition for KGs, but they consist of a set for interconnected entities and their attributes (Pan, Vetere, et al, 2017). In other words, a KG is made up of pieces of knowledge and each piece can be represented as a subject-predicate-object relationship. The subjects and objects are the nodes in the graph and a predicate is an edge describing the relationship between two nodes. The elements of the KG are defined as follows.

Definition 5.1. Nodes: a) Body nodes are latent factors. b) An end node is the target item of the prediction, which also consists of a text description and label value.

Definition 5.2. Edges: a) Body edges are arrows connecting the body nodes and they represent the causal dependence between the nodes. An adjective word "Weak" or "Strong" is added to the edge as an attribute of the relationship. b) An end edge is an arrow pointing to an end node and it represents the predicate "predict", and it is not necessary to add the adjective pair.

In Definition 5.2, the adjective word "Weak" or "Strong" is added to edges. The choice between "Weak" or "Strong" depends on the path loading (standardized path coefficient) between the nodes. "Weak" is given to edges that have path loadings (absolute value) <0.3 between two nodes, while "Strong" is given to those with path loadings (absolute value) ≥ 0.3 (all the relationships should show statistical significance). In SEM, the path loadings evaluate the effect of one factor on the other. The factors that have a strong effect on each other are necessary for constructing the model. The path loadings are the standard regression coefficients between two nodes connected by an arrow, which relates to the (partial) correlation value. A model with a high goodness of fit means it can express the correlations among the factors comparably with the true relationships among the data, requiring the nodes connected by the arrows to have competing strong causal effects on each other. Although different researchers have different opinions on the reference point of the path loading (Hox & Maas, 2001; Steinmetz & Baeuerle, 2012), 0.3 is a safe choice. The effect of choosing different thresholds for the path loading is not a key point here. Users can choose a suitable number according to their application. The practical examples shown in this paper are medical cases, for which we chose 0.3 as a threshold for RF-TL to judge the “Weak” or “Strong” tags. If any factor has a low factor loading compared with all the other factors, it would be weak one in a prediction model. Fig. 5.3 shows the concept of a translated SEM-like KG.

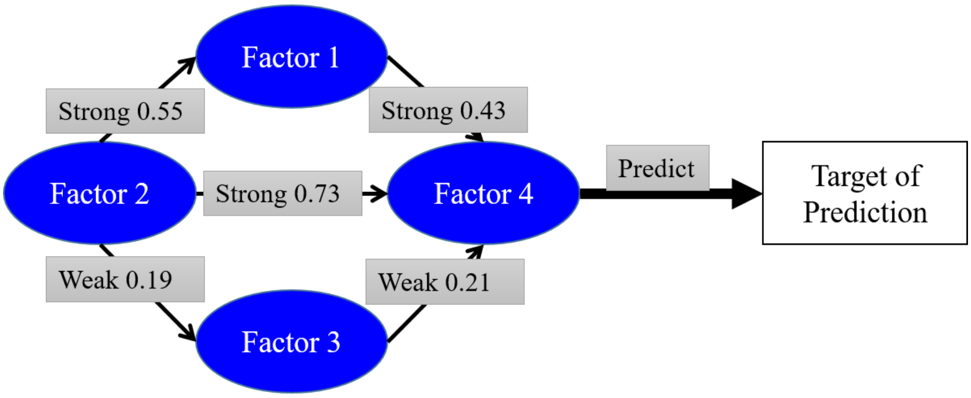


Fig. 5.3. SEM-like KGs. The model consists of latent factors, and measuring items belong to the factors (items except the target of prediction are omitted in the figure). Each latent factor represents a node in the KG made up of an ontology expression and statistical values regressed from the items. Between the nodes, the arrows are the edges of KG with an ontology expression of Weak or Strong and a path loading value. The end node is the target of the prediction item, and an edge pointing to it expresses the action of prediction.

As shown in Fig.5.4, each ellipse represents a latent factor, and the items for measuring the factor are represented as rectangles. Note that, except the target item of prediction, the other measuring items are not shown in the figure. The SEM-like KGs are made up of pieces of knowledge. For instance, in Fig. 5.4, Factor 3 is weakly related to Factor 4 and Factor 1 is strongly related to Factor 4.

Three predicate functions are used for expressing the knowledge in KGs:

$$S_i(x, y, fl_{x\to y}) \tag{5.2}$$

$$W_i(x, y, fl_{x\to y}) \tag{5.3}$$

$$N_i(x, y, 0) \tag{5.4}$$

Functions (5.2)–(5.4) represent three propositions. The subscript i in the functions represents the i^{th} sub-group, x and y are the nodes in the KG, $S_i(x, y, fl_{x\to y})$ means x results in y with a factor loading $fl_{x\to y,}$ and the relationship is strong, and $W_i(x, y, fl_{x\to y})$ means x results in y with a factor loading $fl_{x\to y}$, and the relationship is weak. The order of x and y cannot be changed in Functions (5.3) and (5.4). Function (5.5) means there is no relation between x and y, where there is no arrow between the two nodes in the graph (the standard regression coefficient approaches zero).

5.3.3. Model training in the source domain

The first step is to train the predictive model for *D_S*. RF-TL only cares about strong/weak relationships between nodes of the intervened models. Thus, when training the source domain part, the knowledge expressing relationships on the edges does not have to be shown in the figure. In other words, only the procedures described in section 5.3.2 that “transfer SEM to a predictive system” are conducted in the current step.

In this research, we only consider the situation in which the reason and result have a linear dependence. In the causal relation used by RF-TL, the “reason” is the intervention item. The “result” is the prediction target, and its target can be statistically expressed, such as the attendance rate of the baseball game.

In the source domain model, the item used as the intervening factor should be one of the measuring items of one of the latent factors, which ensures that the model and intervention are relevant. Once more using the baseball example, the temperature is the intervene factor, e.g., the “reason”.

A change in the intervening factor will cause a corresponding changing in the prediction system P, i.e., the attendance prediction system. Then, the trained source model is constructed, as shown in Fig. 5.4.

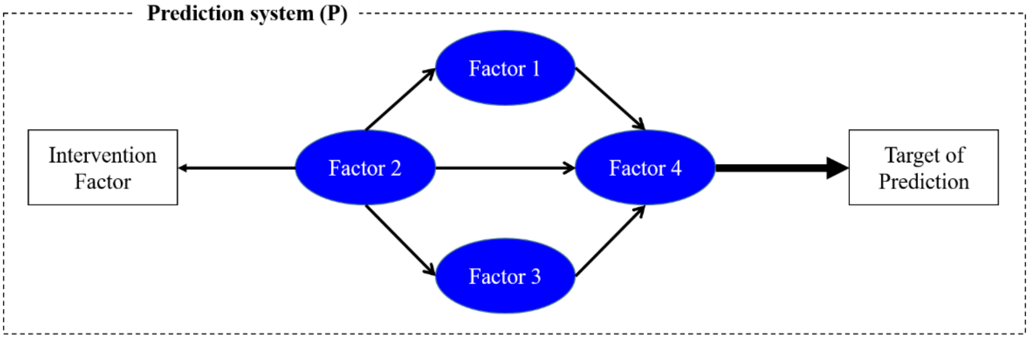


Fig. 5.4. The prediction system in the source domain. The intervening factor is a measuring item of factor 2, which is used as the intervention item, e.g., the “reason”.

Many feature-based TL algorithms have the function of data-dimension simplification. RF-TL is no exception. As mentioned in section 5.3.2, in the data management step, items that do not have a strong ability to measure the model will be removed. Compared with other data-dimension reduction methods, the distinct advantage of using SEM is that the extracted dimensions are all meaningful in practice; e.g., Item 1 represents temperature and Item 2 represents weather. The meaningfulness of the item is a key point for the causal analysis. We use expert causal knowledge in the intervention step. “Knowledge” means something explainable; thus, it is impossible to do a further causal intervention in the succeeding steps without revealing the explanations of the data features.

5.3.4 Interventions on the source model

Intervention stimulates a model by artificial means, and the stimulation constrains the intervention item to being a constant state. Under the three assumptions of FPCI, when the analysis object is a group, causality can be represented by the expectation of the difference between the intervened state and original state, i.e., function (5.1). In addition, an intervention item, such as the intervening factor in the model in Fig. 5.4, is a measuring item that can be regarded as a characteristic for describing one of the factors of the model, e.g., factor 2 in Fig. 5.4. Here, we will give the following definitions:

Definition 5.3. Intervention T: Classify objects into different sub-groups in accordance with the characteristic, i.e., the intervening factor. The intervening factor of each group is labeled by a constant number, such as 1 for the first group, 2 for the second group, etc.

Definition 5.4. The state $E[Y_i(u)]$ of the i^{th} group: The new prediction model trained by the data from sub-group i.

Definition 5.5. The original state $E[Y_c(u)]$: Assuming there are n sub-groups, $E[Y_c(u)]$ is the $(n-1)^{th}$ sub-group.

There are a few caveats regarding these definitions. The first is about the division of the sub-groups. Data in D_S should be divided into sub-groups in accordance with the scale of the intervening factor in D_S. The division must have scale invariance. As in the baseball game example, if the temperature range of D_T is 5°C and D_S is 15°C, there will be three sub-groups, each having a scale of 5°C. Second, RF-TL is based on the linear dependence between the reason and result. Thus, the division of sub-groups is not random but in accordance with the increase or decrease in the intervention item. Furthermore, if the mean value of the intervention item of D_T is on the lower side of D_S, the intervention item of the sub-group is labeled in a descending way, i.e., winter is colder than summer, while if the mean value of the intervention item of D_T is on the higher side of D_S, they will be labeled in an ascending way. Here, if we suppose that D_T ranges from 0°C to 5°C and D_S ranges from 25°C to the 40°C, the 40°C–35°C sub-group can be labeled 1, the 35°C–30°C sub-group can be labeled 2, and the 30°C–25°C sub-group can be labeled 3. Third, the original state is needed for the causal analysis. The original state should be a group without any interventions. Nevertheless, it is difficult to find an ideal state without any intervention; thus, in practice, one of the intervened states is often chosen as the original one. We define a group with label $n-1$ as the original state for convenience of evaluating its intervening scale relative to sub-group n, which is "nearest" D_T. The upper portion of Fig. 5.5 illustrates the intervention procedure.

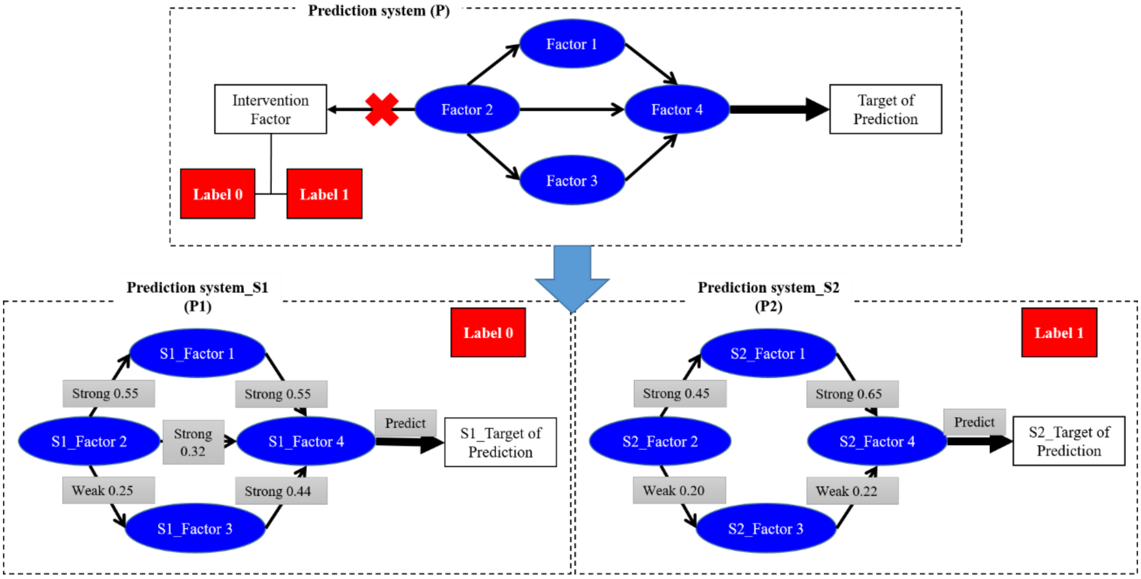


Fig. 5.5. Case of an intervention performed on the intervening factor and two trained sub-models. In the upper part of the figure, the red cross represents that the intervening factor is constrained to be constant label values, e.g., 0 and 1 in the example. After the intervention, the intervening factor is removed from the figure, and the data are divided into sub-groups, e.g., group 0 and group 1. Then, sub-models are trained using the respective sub-group data. Finally, after training the predictive system for each sub-group, edge descriptions of strong/weak relationships with path loadings are added to the figure.

After the intervention, the data in D_S are divided into sub-groups. Because the intervention item has been labeled with a constant number, which means the objects in the sub-group with such a label have the same attributes as the intervention item, the intervention item will no longer be a measuring item of the sub-models. After the intervention, SEM-like KGs are extracted using the data of each sub-group. The training procedure begins by preparing the data. The data features that are used as input for creating the i^{th} sub-prediction system P_i are those used by the prediction system before the intervention P. Unlike in the original *D_S* model, the items belonging to the same factor with the intervention item may be classified into another common factor in the EFA procedure because the intervention item is not used in the sub-models. It is also possible for the number of items or factors to decrease if the item does not have enough power to evaluate the system. Removal of items will affect the transfer process. The specific operations for handling this situation are discussed in the section about the transfer rules. However, the abstract concepts of the common factors should not be changed. Also, the meaning and number of latent factors should be the same in each sub-model; this is necessary for the following transfer procedure. If necessary, the common factors can be forced to be a certain number in accordance with the reference points. The procedure of creating sub-models is shown in the lower portion of Fig. 5.5. After creating the sub-predictive systems, the edge labels, i.e., weak/strong relationships with path loadings, are translated and added to the KGs.

5.3.5 Transferring knowledge graphs to the target domain

The purpose of RF-TL is to find suitable features for predicting the target in D_T through the transfer of the relationships of the D_S model. As mentioned in Section 2, a path loading (absolute value) ≥ 0.3 is the reference point for the predictive power of a factor. As a result, the transfer rules are defined for predicting the predictive power of the factors in the model of D_T. First order logic programming (FOLP) (Lavrac & Dzeroski, 1994) is used to create RF-TL, and the following pseudo-code shows the procedure. For a clear illustration, we have numbered the edges in the sub-models. As mentioned above, the number of latent factors remains the same in each sub-model. Assuming there are m factors in the model, if all the factors are connected to each other and the direction of the arrow is taken into account, there will be $m \times (m-1)$ edges. Also, as mentioned, if the path loading between nodes is extremely small, then no edge will be added to the KGs, i.e., $N_i(x, y, 0)$. If $N_i(x, y, 0)$ is true in all sub-models, this edge is considered to be useless for constructing the model. Thus, it is not necessary to input it to the transferring algorithm. In practice, most of the unnecessary data features are removed in the EFA step, and the remaining ones are classified into few latent factors. As a result, the time cost of RF-TL is usually acceptable. Assuming there are k such edges, they will be ignored when labeling the edges. As a result, the labels from 1 to $m \times (m-1) - k$ are given to the (potential) edges of each model. The order does not matter, but it should be the same in each sub-model.

Table 5.1. Pseudo-code of RF-TL algorithm

RF-TL: Transfer

```
Function EXECUTE TRANSFER ($Q_{s1}$, $Q_{s2}$, $M_T$, $m$, $k$, $w$)
  $M_T = \emptyset$
  for $i$ in the range $(1, m \times (m-1) - k)$ do:
    $N_{s1}(edge_i, 0) \vee W_{s1}(edge_i, fl_{i\_s1}) \wedge S_{s2}(edge_i, fl_{i\_s2}) \Rightarrow M_T(edge_i)$
    $W_{s1}(edge_i, fl_{i\_s1}) \vee S_{s1}(edge_i, fl_{i\_s1}) \wedge N_{s2}(edge_i, 0) \Rightarrow M_T(\neg edge_i)$
    $S_{s1}(edge_i, fl_{i\_s1}) \wedge W_{s2}(edge_i, fl_{i\_s2}) \Rightarrow M_T(\neg edge_i)$
    $N_{s1}(edge_i, 0) \wedge W_{s2}(edge_i, fl_{i\_s2}) \Rightarrow$ LOADING TRANSFER $(0, fl_{i\_s2}, M_T, w)$
    $W_{s1}(edge_i, fl_{i\_s1}) \wedge W_{s2}(edge_i, fl_{i\_s2}) \Rightarrow$ LOADING TRANSFER $(fl_{i\_s1}, fl_{i\_s2}, M_T, w)$
    $S_{s1}(edge_i, fl_{i\_s1}) \wedge S_{s2}(edge_i, fl_{i\_s2}) \Rightarrow$ LOADING TRANSFER $(fl_{i\_s1}, fl_{i\_s2}, M_T, w)$
  end
  return $M_T$
```

RF-TL: Path-loading calculation

```
function LOADING TRANSFER $(fl_{i\_s1}, fl_{i\_s2}, M_T, w)$
  $fl_T = |fl_{i\_s2}| + (|fl_{i\_s2}| - |fl_{i\_s1}|) * |w|$
  $|fl_T| \geq 0.3 \Rightarrow M_T(edge_i)$
  $|fl_T| < 0.3 \Rightarrow M_T(\neg edge_i)$
  return $M_T$
```

The inputs of the algorithm are Q_{s1}, Q_{s2}, M_T, m and w. Q_{s1} is the set of edges of the sub-model labeled $n-1$, and Q_{s2} is the set of edges of the sub-model labeled n. The edges are expressed using Functions (5.2)–(5.4). M_T is the transferred edges in $\mathrm{D_T}$. m is the number of factors, and w is the transfer weight. The principal part of the transferring algorithm is performed according to FOLP. Specifically, whether an edge should be added to the KG of D_T is decided by comparing the "strengths" of the edges in the neighboring sub-models, model $n-1$ and model n. If the edge in model $n-1$ is weak or none and in model n is strong, then the edge is added to the target KG. In contrast, if the edge in model $n-1$ is strong or weak and in model n is none, then the edge is not added to the target KG. Similarly, if the edge in model $n-1$ is strong and in model n is weak, then the edge is not added to the target KG. Moreover, the other situations need the path loadings to be calculated using the path loading calculation algorithm.

In Section 5.2, the "reason" and "result" in the causal relationship used with RF-TL are defined as an intervening factor, such as "temperature" and a statistically expressible model's target, such as "attendance rate". The reason and result are assumed to have a linear relationship, so the causal relation can be expressed as

$$Result = \beta_r \times Reason \tag{5.5}$$

where β_r is the standard regression coefficient.

Furthermore, the transfer weight w is defined as

$$w = multiple \times \beta_r \tag{5.6}$$

Here, $multiple$ depends on the "distance" between the highest (lowest) value of the intervention item $x_I^{(\mathrm{S})}$ in D_S and the lowest (highest) value of the intervention item $x_I^{(\mathrm{T})}$ in D_T. Although the sub-groups are divided up following the rules of the same scale of the intervention item in D_T, there may be a difference between the highest (lowest) value of it in D_S and lowest (highest) value of it in D_T. As in the baseball game example, data in D_S range from 25°C to 40°C, but the data in D_T range from 0°C to 5°C. There is a gap of 20°C between the two domains. Here, $multiple$ is used for filling the gap as shown in function (5.7). As mentioned in section 3.4, we labeled the sub-groups according to the mean value of the intervention item of D_T and D_S. Similarly, when the mean value of the intervention item of $\mathrm{D_T}$ is higher than $\mathrm{D_S}$, the distance is calculated by lowest value of $x_I^{(\mathrm{T})}$ and the highest value of $x_I^{(\mathrm{S})}$, vice versa. The *scale* mentioned here corresponds to the range of $x_I^{(\mathrm{T})}$, which is also the basis for dividing sub-groups.

$$multiple = \begin{cases} 1 + \frac{\left|\min x_I^{(\mathrm{T})} - \max x_I^{(\mathrm{S})}\right|}{scale}, & \text{if } \operatorname{mean}\left(x_I^{(\mathrm{T})}\right) > \operatorname{mean}\left(x_I^{(\mathrm{S})}\right) \\ 1 + \frac{\left|\max x_I^{(\mathrm{T})} - \min x_I^{(\mathrm{S})}\right|}{scale}, & \text{if } \operatorname{mean}\left(x_I^{(\mathrm{T})}\right) < \operatorname{mean}\left(x_I^{(\mathrm{S})}\right) \end{cases} \tag{5.7}$$

The transfer weight w calculates the changing scale between the source domain and the target domain but not regards to the increase or decrease dependence that decided by the label order of the sub models mentioned in section 5.4. Thus, in the transfer algorithm, the absolute value of w was used.

After calculating the path loadings, it is determined whether to add an edge to the target KG by comparing with the threshold of 0.3.

5.3.6 Identifying data features for training models in the target domain

RF-TL returns a set of edges M_T, and all the edges are marked "strong". If there are nodes that do not connect to any other nodes, the items belonging to the nodes are unnecessary for the D_T model and will be removed. For example, for problem A, the final model in D_T is shown in Fig. 5.7.

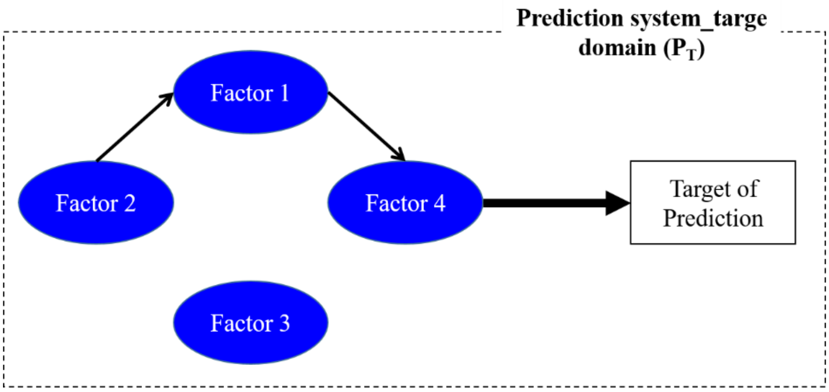


Fig. 5.7. The predictive system in DT. After the transference, in the KG of DT, only edges between Factors 1 and 2 and between Factor 1 and 4 are added. Factors 2, 3, and 4 are separated from each other. As Factor 4 directly points to the target of prediction and Factor 3 does not directly or indirectly connect to Factor 4, the data features that belong to Factor 3 will not be considered when the predicting system in DT is constructed.

As shown in Fig. 5.7, Factor 3 does not connect to any other factors in the prediction model after the transference. Thus, the items belonging to Factor 3 are removed and the items belonging to Factors 1, 2, and 4 are extracted for the prediction system in D_T.

Finally, the unlabeled target of prediction in D_T can be labeled by using a missing-data estimation method, such as the expectation–maximum (EM) algorithm.

5.4 Experiments

To evaluate the effectiveness of RF-TL, we conducted two experiments related to healthcare problems. One was on predicting the obstructive sleep apnea (OSA). The other was on prediction of ICU utilization in the COVID-19 pandemic. There is a common characteristic between these experimental cases, which is the higher the age of the patient is, the higher the risk will be (Gabbay & Lavie, 2012; Krieger, Sforza, et al, 1997; Ayalon, Ancoli-Israel & Drummond, 2010; Zhang, Cui, et al, 2018). Thus, expert causal knowledge in each case is the effect of age on the risk of disease. The causal model is shown as Fig. 5.8.

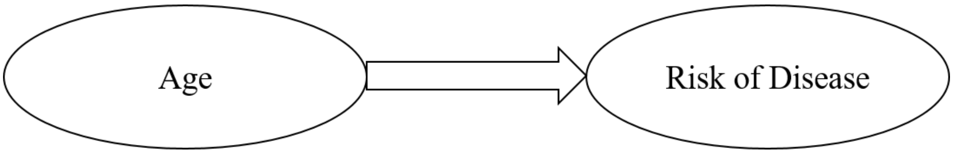


Fig. 5.8 Causal model of age increasing the risk of a particular disease, i.e., OSA and COVID-19

The hardware and software configurations of the experimental environment are shown in Table 2.

Table 5.2 Hardware and software configurations of the experimental environment

Hardware	
CPU	Core (TM) i7-7700HQ CPU @ 2.80GHz 2.80 GHz
Memory	16 GB
Operation system, software and programing language	
OS	Windows 10 x64
Software	IBM SPSS Statistics v26
Programming Language	R-3.6.2
Key package in R	Lavaan-0.6-9, semPlot-1.1.2, GA-3.2.1,

The primary development environment for the experiments is based on R. The essential package for SEM analysis is Lavaan, and the GA learns the structure of the graph. Finally, KGs are drawn using the semPlot package. Before building the structural model for SEM, we run the EFA in SPSS statistics software and the EM for predicting missing labels. Although the EFA and EM procedures can also be done in R, we took advantage of the user-friendly interaction of SPSS. The realization of the proposed algorithm is not limited to the configurations shown in Table 5.2. To the best of our knowledge, the mentioned packages can be used in other development environments, i.e., Python. The data sizes of the experiments were relatively small. When RF-TL is applied to big data, GPU-based packages can be used to speed up the calculation.

Table 5.3 Parameter settings of RF-TL in the two experiments

	OSA	**COVID-19**
m	5	2
k	16	1
w	0.975	0.994

There are three parameters that need to be pre-set before running the RF-TL algorithm, m: number of nodes; k: number of “no branch in sub-models;”

and w: transfer weight. The procedure for obtaining these parameters is shown in Section 5.3. In the respective experiments, these parameters were set as shown in Table 5.3.

5.4.1 Questionnaire diagnosis of Obstructive Sleep Apnea

OSA is a common sleep disorder. The most effective method of diagnosing OSA is using polysomnography with a peripheral capillary oxygen saturation test. However, it is expensive and difficult for people to use at home. Here, questionnaires are better than methods that require professional supervision as a means of diagnosing OSA in primary care and are self-diagnostic. There are many types of questionnaires containing numerous questions, such as the Quality of Life questionnaire, Epworth sleepiness scale, and Stop-Bang questionnaire. We collected 60 items for predicting the risk of getting OSA from the self-rated questionnaires of the Sleep Heart Health Research dataset, which includes anthropometrics (6 items), health interviews (11 items), sleep habits, and quality (35 items), and 36-Item Short Form Survey (SF_36) questionnaires (8 calculated items). Additionally, an Apnea Hypopnea Index (AHI) ≥ 5 is treated as undiagnosed OSA.

In the experimental dataset, there were a total of 3821 patients aged from 40 to 80. The patients in their 50s and 60s had labeled AHI data and those in their 40s and 70s did not have any label. The tasks began with constructing an OSA-prediction model for patients in their 50s and 60s. Features for the young group (40s) and old group (70s) were transferred from the 50s~60s model. D_{SO} was the source domain that included the features X_{SO} (in their 50s and 60s with the label of AHI), D_{TO1} was the target domain that included the features X_{TO1} (in their 40s without the label of AHI), and D_{TO2} was the target domain that included the features X_{TO2} (in their 70s without the label of AHI). There were two tasks. T_{TO1} was to extract the data feature for predicting OSA in D_{TO1}, and T_{TO2} was to extract the data feature for predicting OSA in D_{TO2}.

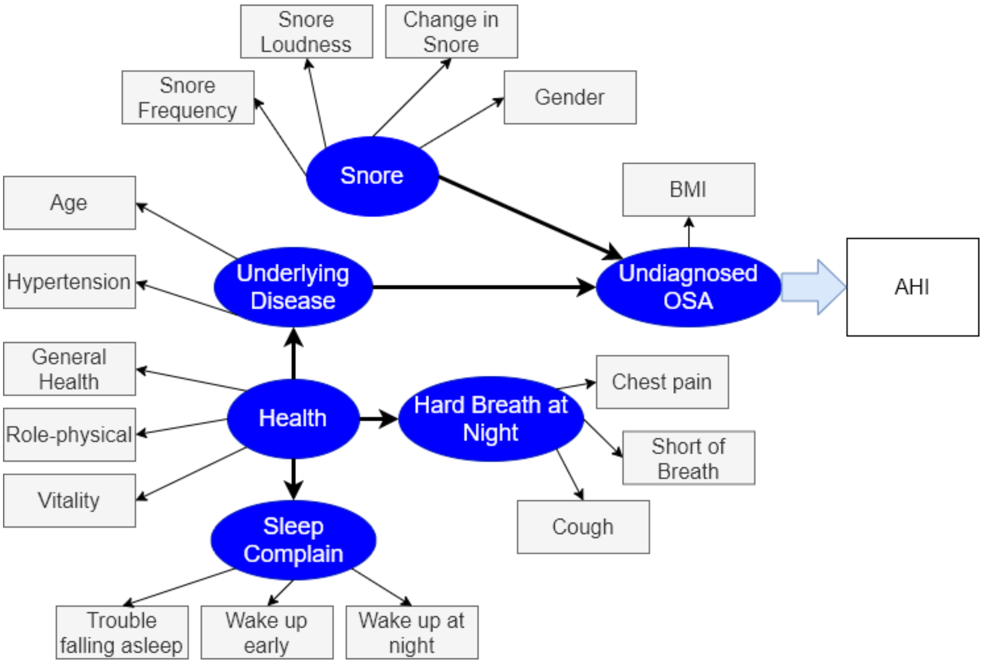
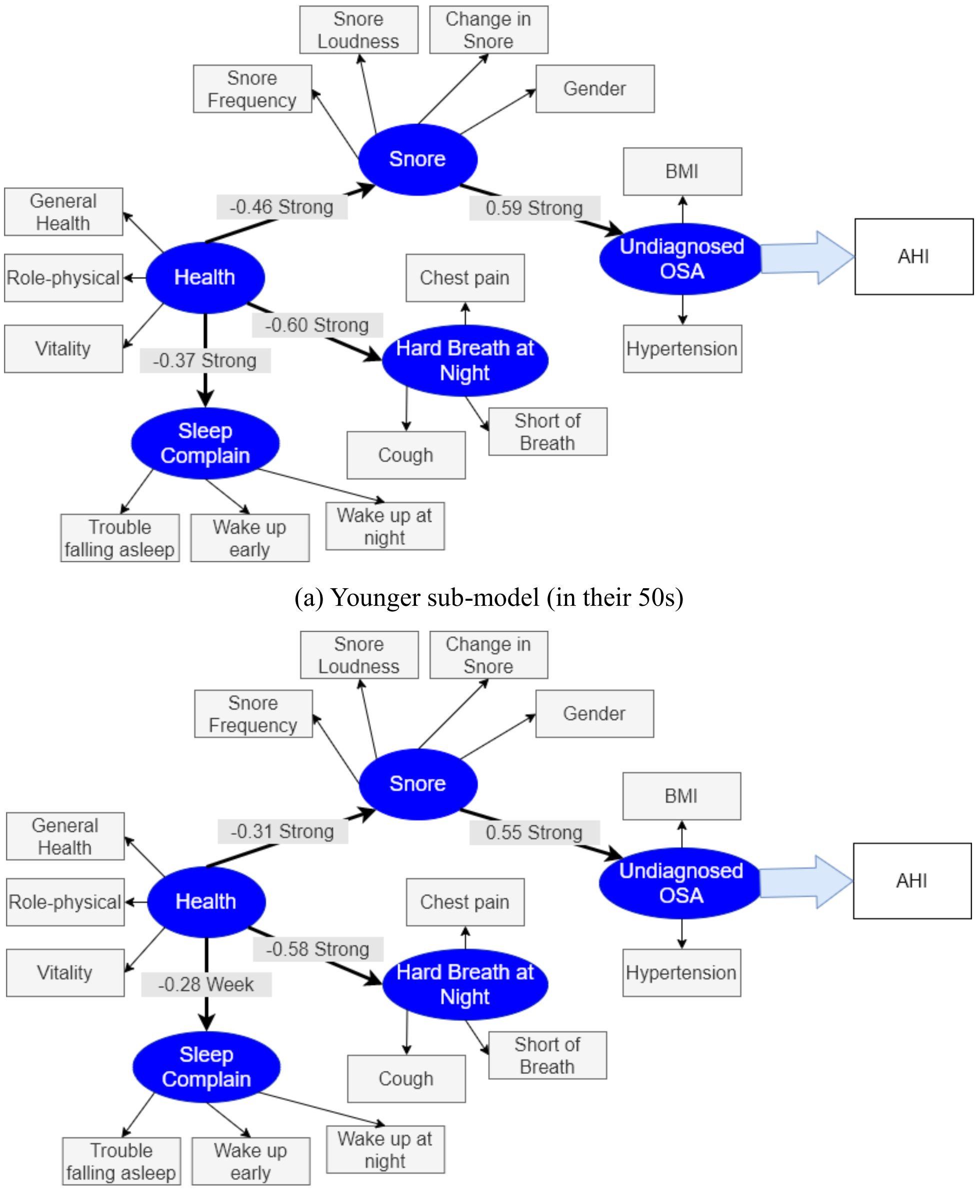


Fig. 5.9. Model of D_{so} for predicting OSA

Fig. 5.9 shows the constructed model for predicting AHI in D_{SO}, which consists of 16 questionnaire item variables that are classified into six factors. Age was one of the measuring items for the factor “underlying disease.” The age intervention produced two groups. One was a sub-group of patients in their 50s labeled Younger and the other was a sub-group of patients in their 60s labeled Older. Two sub-models were trained using the 16 items with the corresponding data in each sub-group. The trained sub-models are shown in Fig. 5.10.



(a) Younger sub-model (in their 50s)

(b) Older sub-model (in their 60s)

Fig. 5.10. Sub-models for predicting OSA in the divided source domains

As shown in Fig. 10, the age intervention removed the factor “underlying disease” from the model, and classified the Hypertension item into the factor “undiagnosed OSA.” This re-classification is reasonable and will not influence the result of the transfer. The factor loadings are marked on the path and have been translated into “weak” or “strong” labels.

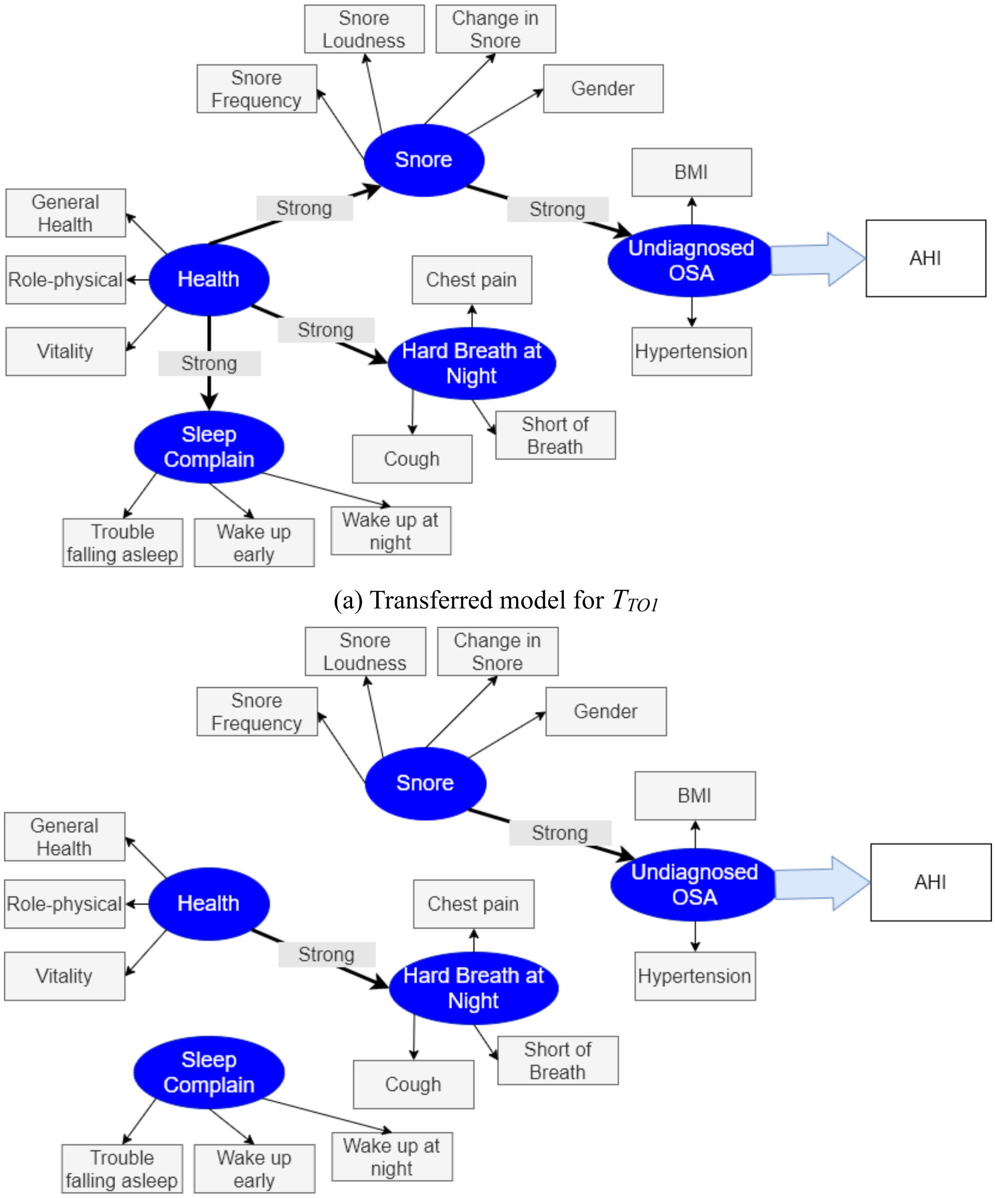
Next, RF-TL was used to transfer the knowledge network from $\mathrm{D_{SO}}$ to $\mathrm{D_{TO}}$. As mentioned above, there were two transfer tasks. One was to transfer from D_{SO} to $D_{TO1,}$ (Young). Here, the Younger sub-model was labeled 1 and Older sub-model was labeled 2. The other task was to transfer from D_{SO} to $D_{TO2,}$ (Old). Here, the Younger sub-model was labeled 2 and Older sub-model was labeled 1. Next, counterfactual inference was performed on the basis of the causal knowledge. For example, as shown in Fig. 5.10 (a), the relationship between factor “health” and factor “snore” is strong with a factor loading of -0.46. In Fig. 5.10 (b), the relationship is still strong but with a factor loading of -0.31. Additionally, the “reason” we used in this example is “age.” Thus, from Fig. 5.10, we can obtain the following information:

“In the prediction system of AHI, as age increases (decreases), the relationship between Health and Snore becomes weaker (stronger).”

Furthermore, the counterfactual inference yielded,

“If age is older (younger), then the relationship between health and snore is weaker (stronger).”

The counterfactual inference of the RF-TL algorithm is quantified depending on which the relationship between factors in the target domain is predicted (such as the weak relationship between “health” and “snore” in T_{TO2} as shown in Fig. 5.10 (b)). The transferred models for $\mathrm{T_{TO1}}$ and $\mathrm{T_{TO2}}$ are shown in Fig. 5.11.



(a) Transferred model for T_{TO1}

(b) Transferred model for T_{TO2}

Fig. 5.11. Transferred models for predicting OSA in the two different target domains

For the transferred model of T_{TO1} (Young), all the relations in the model were marked as strong and connected. As well as the items shown in the model, we used age as the expert knowledge. As a result, age was found to be the ground truth that changes OSA and that should be used as one of the data features for predicting AHI. Thus, there were 15 items, plus the item age, which was used for predicting AHI for the objects aged in their 40s.

Different from the model for T_{TO1}, the relationship starting from Health and pointing to Snoring and Sleep Complaints was “Weak” for D_{TO2}. Thus, an edge was not added to the graph, and the model was divided into two parts: one part with the factors “Health”, “Sleep Complaints”, and “Difficult to Breathe at Night” and one part for predicting AHI. Only the six items contained in the factors “Snoring and Undiagnosed OSA” and the age item were used for predicting AHI of T_{TO2}.

To evaluate the effectiveness of the feature transfer, the EM algorithm was used for predicting the label for the target domains. As mentioned above, we extracted 16 items for T_{TO1} and 7 items for T_{TO2} from the 60 items originally collected. We used the EM algorithm for labelling AHI in D_{TO1} and D_{TO2} with 60 items, 16 items, and 7 items. We used accuracy and F1 score as evaluation indexes. Table 5.4 lists the results.

Table 5.4. Comparison of OSA prediction using different numbers of items

D_{TO}	**Young (D_{TO1})**		**Old (D_{TO2})**	
No.	**Accuracy**	**F1_score**	**Accuracy**	**F1_score**
60 items	72.88%	0.7319	75.16%	0.5669
16 items	**74.53%**	**0.7519**	76.14%	0.5801
7 items	73.71%	0.7441	**76.41%**	**0.5804**

The prediction results indicate that the use of 16 items in D_{TO1} resulted in the highest accuracy and F1_score and that 7 was the most suitable number of extracted items for predicting AHI of the old group.

The CORAL algorithm and TCA algorithm are two commonly used TL algorithms for transferring features from the source to the target domain with no labels. We compared these two algorithms with RF-TL. For TCA, it is necessary to determine the number of previously transferred features; thus, 16 is given to T_{TO1} and 7 is given to the T_{TO2} to maintain consistency with RT-TL. On the other hand, CORAL does not need to define the number of previously transferred features and transferred 59 items from the source domain to both target domains. The number of features it extracted was much greater than the number extracted by RF-TL, which highlights the advantage of RF-TL. The EM algorithm was used again for predicting AHI by using the items extracted with TCA and CORAL, and the results were compared with those of RF-TL. Table 5.5 lists the results.

Table 5.5. Comparison of results with TCA and CORAL for predicting OSA

D_{TO}	**Young (D_{TO1})**		**Old (D_{TO2})**	
Methods	**Accuracy**	**F1_score**	**Accuracy**	**F1_score**
TCA	55.28%	0.6098	77.49%	-
CORAL	56.84%	0.6573	72.74%	0.5934
RF-TL	**74.53%**	**0.7519**	**76.41%**	**0.5804**

For T_{TO1}, RF-TL had the highest accuracy and F1_score. For T_{TO2}, although the accuracy of TCA was higher, the precision of negative (AHI is labeled as 0) was zero, so there was no F1_score and it failed to make a prediction. The accuracy of RF-TL was higher than that of CORAL. The F1_score was a little lower due to the unbalanced number of objects contained in the negative and positive groups. Also, there were 59 items used for CORAL and only 7 items used for RF-TL. These results show that RF-TL outperformed CORAL.

5.4.2 ICU-candidate prediction for COVID-19 patients

The novel coronavirus started spreading across the world in early 2020. Millions of people have been infected, and the number is still increasing. Because of the large number of patients, medical collapse threatens many countries. Predicting severe cases requiring an intensive care unit (ICU) is an important task. The “COVID-19 - Clinical Data to assess diagnosis” dataset has been published online (Hospital Sírio-Libanês, 2020); it contains 189 items, including the ICU item (0 for No, 1 for Yes) collected from the patients diagnosed with COVID-19. There are 430 objects with no missing items of the patients ranging from 20s to 90s. The distribution of the age groups and the ICU-positive rate are listed in Table 5.6.

Note that the data were collected before mutated virus started to spread. The example only considered age as the factor in the counterfactual inference. Note as well that the current situation of viral spread is different due to mutations (such as widespread transmission of mutated virus among young people), which may cause differences from the results of this example. RF-TL only considered single-factor causality. The limitations of this point will be explained in the discussion section.

Table 5.6. Information on patients infected by COVID-19

Age group	**No. of patients**	**ICU-positive rate [%]**
20s and 30s	113	36.28
40s and 50s	110	43.64
60s and 70s	108	55.56
80s and 90s	218	62.63

The ICU-positive rate has a positive correlation with age. Also, in accordance with current knowledge, age is one of the factors of infection and severe cases (Wu, Leung, et al, 2020), which conforms to the age-disease causal model shown in Fig. 5.8. We assumed that only the data of the 40s–70s age groups were labeled with ICU tags (D_{SI}) and that the two target domains D_{TI1} of the 20s and 30s groups and D_{TI2} of the 80s and 90s groups did not have ICU tags. RF-TL was used for transferring data features from D_{SI} to D_{TI1} and D_{TI2}. The two tasks, T_{TI1} and T_{TI2}, aimed at extracting suitable data items for predicting ICU candidates in D_{TI1} and D_{TI2}. A prediction model was first constructed for D_{SI}, as shown in Fig. 5.12.

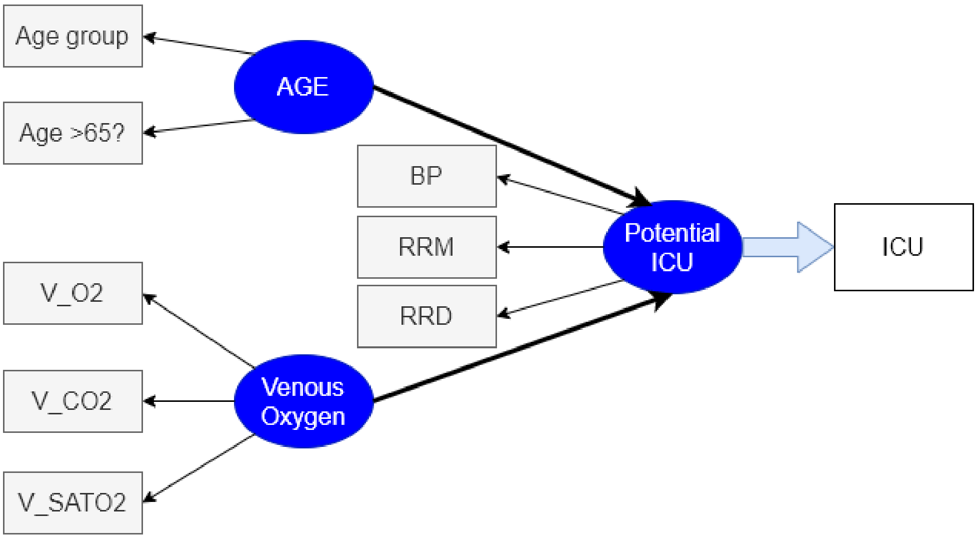
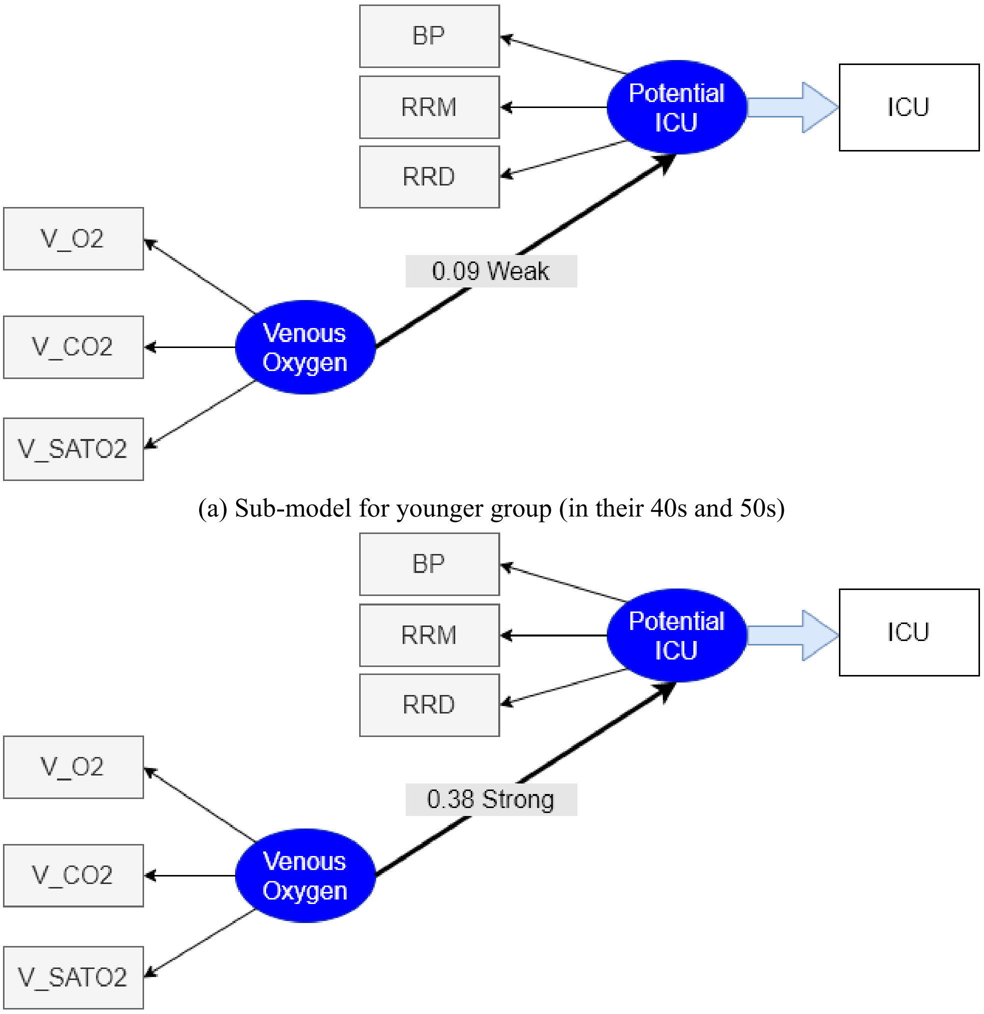


Fig. 5.12. Prediction model for D_{SI}

The notation V_O2 denotes the partial pressure of venous oxygen (minimum); V_CO2 denotes the partial pressure of venous carbon dioxide (maximum); V_SATO2 denotes blood oxygen saturation (mean); BP denotes diastolic blood pressure (range); RRM denotes respiratory rate (mean); and RRD denotes respiratory rate (range/median).

Although there were 189 items available, only 8 items were extracted for predicting whether the object needs to be sent to ICU. An intervention was conducted on the age factor. Different from the OSA case, age in the ICU model is a factor, not an item. Thus, the invention procedure removes the age factor from the model and divides the source domain into two sub-domains; one containing patients in their 40s and 50s and the other containing patients in their 60s and 70s. The sub-models are shown in Fig. 5.13.

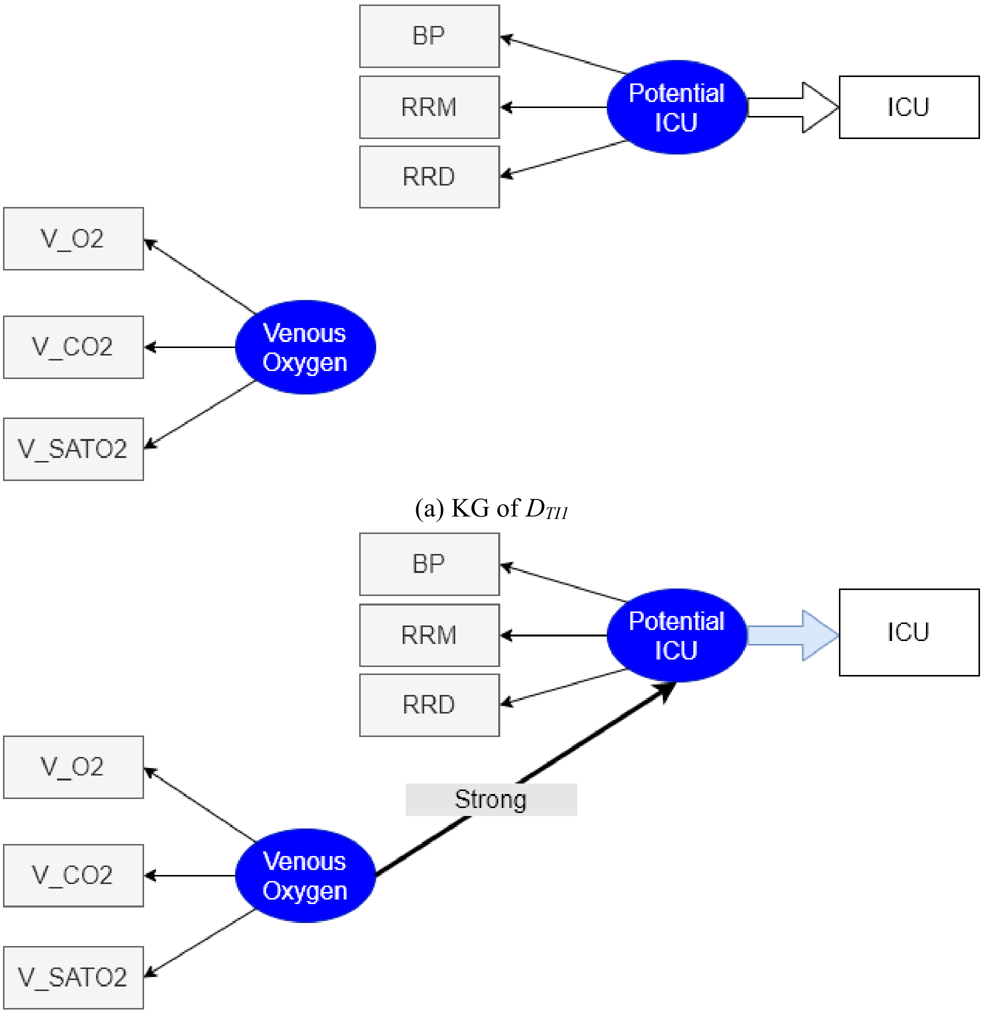


(a) Sub-model for younger group (in their 40s and 50s)

(b) Sub-model for older group (in their 60s and 70s)

Fig. 5.13 Sub-models for ICU-candidate prediction in the divided source domains

The rules were used to transfer the models to the target domains. The KGs for T_{TI1} and T_{TI2} are shown in Fig. 5.14.



(a) KG of D_{TI1}

(b) KG of D_{TI2}

Fig. 5.14. Transferred models for ICU-candidate prediction in the two different target domains

From the information in Fig. 5.14, only the 3 items belonging to the Potential ICU factor and the age item were used for T_{TI1}. Six items with age, a total of 7 data features were used for T_{TI2}. The EM algorithm was used for labeling the ICU data. Table 4 compares the prediction results for T_{TI1} and $\mathrm{T}_{\mathrm{TI2}}$ with 189, 7, and 4 items.

Table 5.7. Comparison of ICU prediction using different numbers of items

D_{TO}	Young (D_{TO1})		Old (D_{TO2})	
No.	**Accuracy [%]**	**F1_score**	**Accuracy [%]**	**F1_score**
189 items	84.96	0.8353	79.80	0.8018
7 items	84.07	0.7519	**88.98**	**0.8898**
4 items	**89.38**	**0.8895**	77.78	0.7687

As expected, the 4 items of T_{TI1} and 7 items of T_{TI2} yielded the highest prediction performance.

Similarly, we compared the results of the prediction with TCA and CORAL. For the COVID-19 ICU case, 186 items were transferred from the source domain to the two target domains with CORAL and 4 items were fixed for the younger domain and 7 items for the older domain with TCA. The results are listed in Table 5.8.

Table 5.8. Comparison of results of predicting OSA with TCA and CORAL

D_{TI}	Younger (D_{TI1})		Older (D_{TI2})	
Algorithms	**Accuracy**	**F1_score**	**Accuracy**	**F1_score**
TCA	36.28%	-	62.63%	-
CORAL	81.42%	0.8124	80.81%	0.8198
RF-TL	**89.38%**	**0.8895**	**88.89%**	**0.8898**

RF-TL performed the best in each target domain.

5.5 Discussions

The transferred element of RF-TL is the relationship in the model, which is different from other TL algorithms. Apart from accuracy, researchers of ML technologies are beginning to focus their attention on the inferring logic inside the model. Their goal is to build ML models with the ability to interpret human cognitive and reasoning processes. KGs are excellent tools for showing human knowledge networks in which domain experts' inference logic can be demonstrated. Relational TL algorithms are applications of KGs. For relational TL, only by clarifying the learning structure of the source-domain model can the relationships be transferred to target domains.

A causal relationship is a higher level of statistical dependency between two data items and is ground truth based on expert knowledge or experience. The reason and the result in a causal model are correlated with each other. However, the causality between them cannot be determined only by clarifying the correlation between two items. Causality needs a "time order". That is, the reason occurs before the result, and a change in the reason inevitably causes a corresponding change in the result. In contrast, correlation is only an expression of the data at a certain time point and it does not express the time order. Future prediction needs to clarify the development of one thing along with the time stream. This is why counterfactual inference can be done only in accordance with the causal relationship.

Transferring the information from a known domain to an unknown domain can be treated as a prediction; thus, the level of statistical dependency is not sufficient. Traditional TL algorithms, such as TCA and CORAL, only take into account the statistical relationships among data features. The two algorithms do not perform well because of the non-significant difference in the data-feature distribution between the source domain and the target domain. Nevertheless, RF-TL uses causality to direct the transfer procedure by predicting how the relationships between data features change in the source domain. The relations among the features are considered, and the inference is performed in accordance with explainable human causal knowledge. As a result, good performance can be obtained in practical applications.

However, the two experiments had limited data sizes, so the calculating time is not long. Two parts of RF-TL take up most of the calculation time. One is the comparison of the edges between sub-models. The more edges are added the higher the time cost becomes. To deal this problem, RF-TL uses a pre-pruning step before transferring. As shown in the OSA experiment, 16 edges are removed from 20 edges before the transference steps. The other time-consuming step is the GA procedure for identifying the structure of KGs. Referring to SEM-EML, the proposed SEM-like KGs conducts a two-step

GA. In step 1, correlations are estimated between each pair of nodes. It can be easily performed in SEM by adding double-direction arrows to all nodes. Edges connecting nodes with a correlation coefficient higher than 0.1 are labeled. Then, the labeled edges that are suggested solutions to be input to the GA are constrained to be “1”. The Goodness of Fit (GoF) indexes are used as fitting functions in the GA, and the suggested edges let GA iterations start from a relatively high GoF which helps to reduce the number of iterations and save calculation time. Although various factors that influence time costs are considered, RF-TL needs a further test to determine its effectiveness and feasibility.

Additionally, the causal model used in RF-TL is a single-factor causality. In other words, only parts of the causal structure are taken into account. For a practical case, one result is usually linked by multiple reasons. As in the COVID-19 example, except for age, mutation of the virus would be another factor influencing the prediction of the severe-case rate. When considering multiple factors in a causal model, the degree of influence of each factor should be weighted accordingly, which we will do in our future work.

5.6 A brief summary

In this chapter, we proposed RF-TL. In accordance with causal analysis and counterfactual interference directions, RF-TL transfers the relationships between data features from a source domain to the target domain. Feature extraction is then conducted in accordance with the information in the transferred KG. Because RF-TL considers the links between different data items and the prediction function of causality, it performs better in practical cases than other TL algorithms.

The proposed RF-TL applied SEM-EML to train the explainable ML models, based on which the utilization of causal knowledge from a domain expert becomes possible. The features-choosing for ML is complex and challenging. The completely depending on algorithms choosing by themselves is time consuming and inexplicable. In this chapter, an example of using causality to identify useful data features in a learning domain was described. The proposed RF-TL in this chapter take advantage of the causality, which is a convenient and valid way for enhancing the reliability of the ML.

Chapter 6. Conclusions and Future Perspectives

6.1 Conclusions

This thesis researched on incorporating human expert knowledge into Machine Learning for enhancing reliability. Except for the accuracy, reliability also evaluates the explainability of an ML technology. For enhancing reliability, human expert knowledge is an effective way of guiding the learning procedure of ML. In this study, we discussed the effect of two kinds of human expert knowledge, the knowledge extracted from experience and the causality, on the improvement of ML technologies' reliability. Three novel ML models and algorithms are proposed as cases and validated that human expert knowledge's introduction helps ML enhance the accuracy, learning efficiency, and explainability.

A cluster size constrained Fuzzy c-Means directed by density information (CSCD-FCM) was shown in Chapter 3. FCM is a commonly used data classification method. However, limited by the drawbacks of the objective function, each cluster's data populations tend to be equal. CSCD-FCM solved this problem by giving a priori-knowledge about the cluster size to the algorithm. The utilization of knowledge extracted from human experience as "a priori knowledge" regulates the learning procedure of FCM, which improved the performance of the algorithm and made the clustering results according to human expectation.

Chapter 4 described an explainable machine learning model based on the Structural Equation Modeling (SEM-EML) method. SEM-EML clearly shows the dependency relationship among the data features and the machine's inference procedure, which unfolds the new knowledge hidden in the data before human users' eyes. By using the causal analysis function, users can predict the developing tendency of the analysis target reasonably.

Chapter 5 illustrated a Relational Feature-Transfer Learning (RF-TL) algorithm. RF-TL used SEM-EML to extract the critical information from data and transfer the knowledge networks to the target domain from another related domain by the causality direction. RF-TL is proposed based on the interventionism-causality theory. The causal knowledge used for transferring the knowledge network is given by the domain expert, which is assumed as the ground truth. According to the causal knowledge, the “Strong/Weak” relation between the target domain’s data features can be predicted from the known information in the source domain.

Machine Learning is the ability for machines to mine useful information from data and help humans predict a target. Various machine learning models are expressions of the learning procedure. The data comes from human beings and human lives. In essence, machine learning is a process that comes from humans and is applied to humans. However, nowadays, to pursue high-precision learning models, researchers mostly focus on the process of “learning from data” while ignoring the role of the human factor in machine learning. This thesis stressed the usefulness of human expert knowledge assisting the learning of machines. In the future, we would like to apply the technologies mentioned in this thesis to assist the development of humanoid robots, especially in the application of cognitive development in Artificial Intelligence.

6.2 Future perspectives

First of all, the practical cases mentioned in the dissertation have not involved the “big data”. Although the Machine Learning models based on big data, e.g, deep learning NNs, always get very high accuracy, most of the algorithms in this category are complete black boxes. The explanation to the relationships among big data is complex. However, the methods, such as SEM-EML proposed in this thesis has a high data dimensions reduction ability. It can be used for reducing the number items. Then, the explainable models can be constructed using the less amount of data features. The methods mentioned in this thesis are hoped to be extended to big data occasions in future.

Additionally, the reliability of ML built up on the extent of how much human users can understand the learning structures and predicting results made by an algorithm or a model. What can be straightly and easily accepted and comprehended is the knowledge that already mastered by our human beings. As shown in the presented studies, the utilization of expert knowledge benefits ML in the aspects of both effectiveness and explainability. Nevertheless, the human exploration of ourselves and the world is extremely limited. For instance, the perception and cognition procedure of the human brain still remain large blank. The learning procedure of ML is to imitate human brain making the decision. The inexplicable of ML partially comes out from the cognitive deficiency of the thinking procedure of a human brain, which limited the development of ML at the same time. The further exploring of human and nature and explaining the “model in the brain” is the premise of explaining the ML, as well as improving the ML technologies.

For the other, human beings’ capability is finitude comparing with the super power of Machine Learning from the “big data”. The “knowledge system” should not only strict in “human knowledge”, while it ought to be the information network in the natural world we lived in. The powerful ML technologies are functional tools to mine information from data. In the future, probably currently already, ML will become the teacher to rich the knowledge system of our humans. In return, the new knowledge learned by human can once more used as “expert knowledge” to improve the learning ability of ML. The reciprocal symbiosis of human and ML can be achieved through the “Learning from the Knowledge”, and where the knowledge located in should be the “truth in the natural world”. As shown in Figure. 6.1, in a “Data-Knowledge-Practice” cycle, Data are representations of the information in the actual world. Knowledge can be learned from Data, which guide to improve the practice of the user (Human/Machine). The improved practice produces new data to the system and new knowledge is obtained continuously.

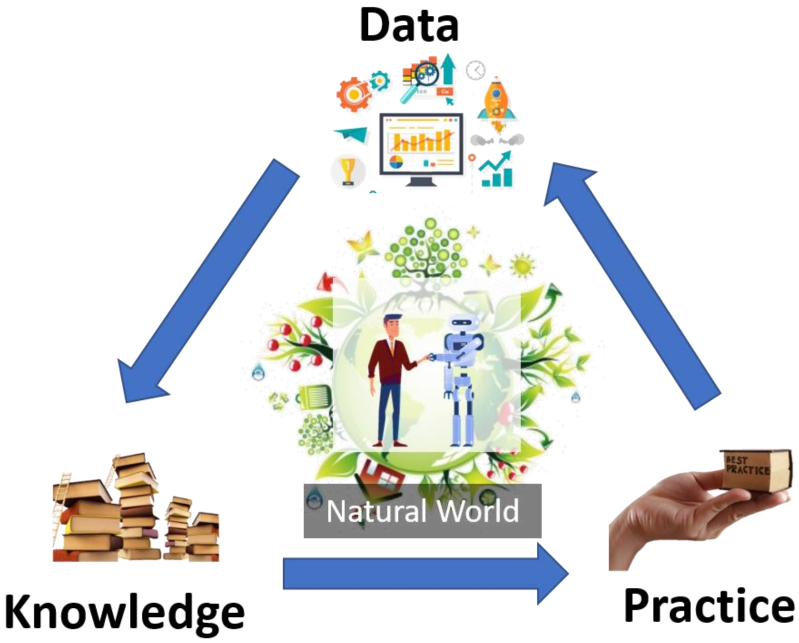
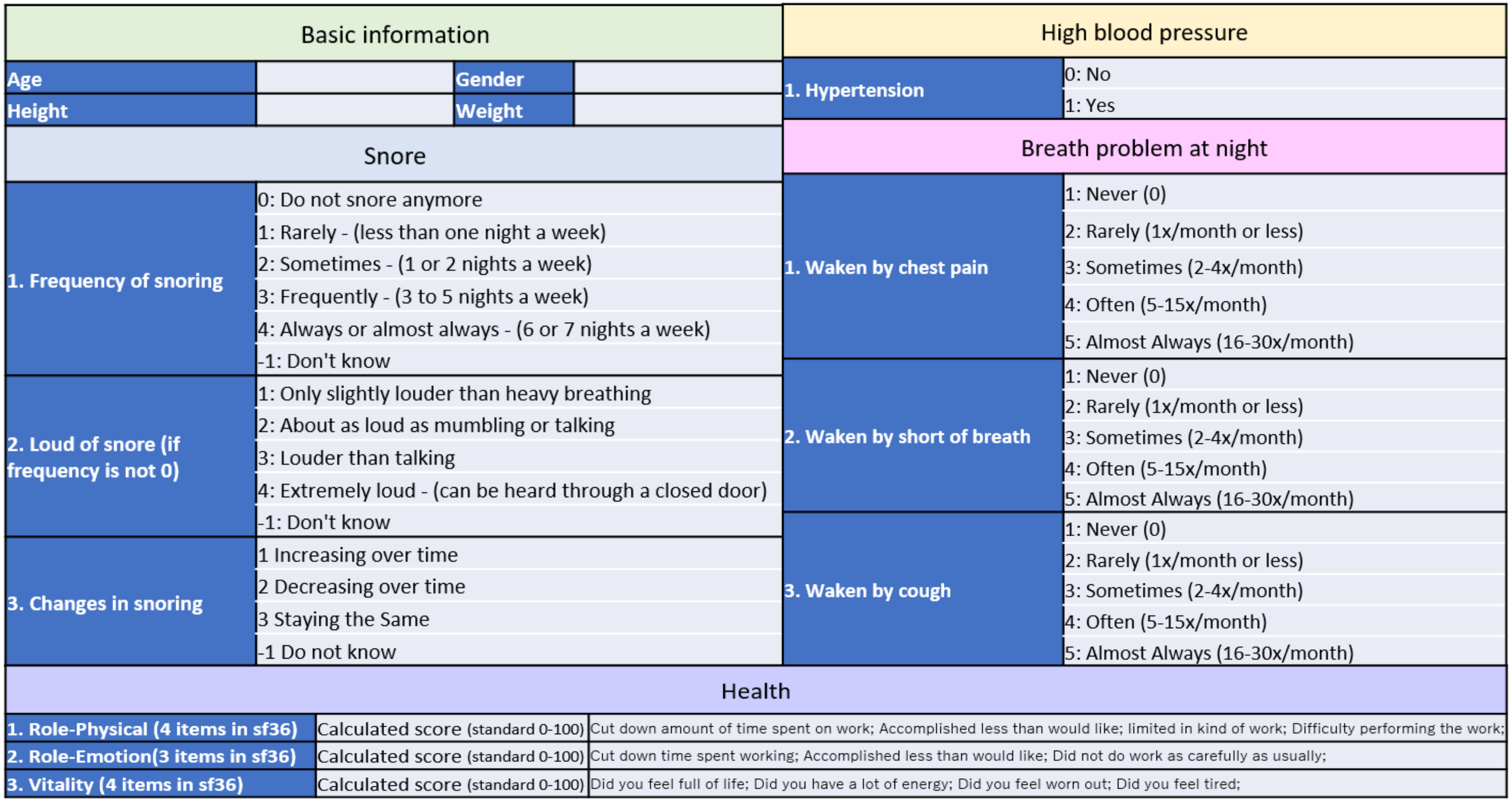


Fig. 6.1. Human and ML learn knowledge from Natural World and achieve mutual improvement

The reliability between humans and ML is the guaranty of not only improving the performance of ML but also human progress. It is a start point of mutual improvement through the human-machine corporation. This study serves as a modest spur to induce the Human-ML symbiotic development and mutual improvement. The part of the picture that ML helped by human knowledge was painted and the left of the painting is expected to be drawn in the future.

Appendix 1. Sleep Obstructive Apnea Risk Assessment Questionnaire



Basic information			
Age		Gender	
Height		Weight	

Snore	
1. Frequency of snoring	0: Do not snore anymore
	1: Rarely - (less than one night a week)
	2: Sometimes - (1 or 2 nights a week)
	3: Frequently - (3 to 5 nights a week)
	4: Always or almost always - (6 or 7 nights a week)
	-1: Don't know
2. Loud of snore (if frequency is not 0)	1: Only slightly louder than heavy breathing
	2: About as loud as mumbling or talking
	3: Louder than talking
	4: Extremely loud - (can be heard through a closed door)
	-1: Don't know
3. Changes in snoring	1 Increasing over time
	2 Decreasing over time
	3 Staying the Same
	-1 Do not know

High blood pressure	
1. Hypertension	0: No
	1: Yes

Breath problem at night	
1. Waken by chest pain	1: Never (0)
	2: Rarely (1x/month or less)
	3: Sometimes (2-4x/month)
	4: Often (5-15x/month)
	5: Almost Always (16-30x/month)
2. Waken by short of breath	1: Never (0)
	2: Rarely (1x/month or less)
	3: Sometimes (2-4x/month)
	4: Often (5-15x/month)
	5: Almost Always (16-30x/month)
3. Waken by cough	1: Never (0)
	2: Rarely (1x/month or less)
	3: Sometimes (2-4x/month)
	4: Often (5-15x/month)
	5: Almost Always (16-30x/month)

Health		
1. Role-Physical (4 items in sf36)	Calculated score (standard 0-100)	Cut down amount of time spent on work; Accomplished less than would like; limited in kind of work; Difficulty performing the work;
2. Role-Emotion(3 items in sf36)	Calculated score (standard 0-100)	Cut down time spent working; Accomplished less than would like; Did not do work as carefully as usually;
3. Vitality (4 items in sf36)	Calculated score (standard 0-100)	Did you feel full of life; Did you have a lot of energy; Did you feel worn out; Did you feel tired;

Bibliography

Ayalon, L., Ancoli-Israel, S., & Drummond, S. P. (2010). Obstructive sleep apnea and Age: a double insult to brain function?. *American journal of respiratory and critical care medicine*, 182(3), pp. 413-419.

Banu, P. N., & Andrews, S. (2015). Performance analysis of hard and soft clustering approaches for gene expression data. *International Journal of Rough Sets and Data Analysis (IJRSDA)*, 2(1), pp. 58-69.

Bensaid, A.M., Hall, L.O., Bezdek, J.C., Clarke, L.P.(1996). Partially supervised clustering for image segmentation. *Pattern Recognition*, 29(5), pp. 859–871.

Bimba, A. T., Idris, N., Al-Hunaiyyan, A., Mahmud, R. B., Abdelaziz, A., Khan, S., & Chang, V. (2016). Towards knowledge modeling and manipulation technologies: A survey. *International Journal of Information Management*, 36(6), 857-871.

Breiman, L. "Random Forests." *Machine Learning*, 45, pp. 5–32, 2001.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. *Boca Raton*, FL: Chapman & Hall, 1984.

Chang, E. T., Baik, G., Torre, C., Brietzke, S. E., & Camacho, M. (2018). The relationship of the uvula with snoring and obstructive sleep apnea: a systematic review. *Sleep and Breathing*, 22(4), pp. 955-961.

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

Chen, H., & Luo, X. (2019). An automatic literature knowledge graph and reasoning network modeling framework based on ontology and natural language processing. *Advanced Engineering Informatics*, 42, 100959.

Chen, X., Jia, S., & Xiang, Y. (2020). A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141, 112948.

Cheng, B., Zhang, Y., Cai, D., Qiu, W., & Shi, D. (2018, August). Construction of traditional Chinese medicine knowledge graph using data mining and expert knowledge. *In 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC)* pp. 209-213. IEEE.

Chung, F., Abdullah, H. R., & Liao, P. (2016). STOP-Bang questionnaire: a practical approach to screen for obstructive sleep apnea. *Chest*, 149(3), pp. 631-638.

Cialdini, R. B., & Sagarin, B. J. (2005). Principles of Interpersonal Influence. In T. C. Brock & M. C. Green (Eds.), *Persuasion: Psychological insights and perspectives* pp. 143–169. Sage Publications, Inc.

Constantinou, A. C., Fenton, N. E., & Neil, M. (2012). pi-football: A Bayesian network model for forecasting Association Football match outcomes. *Knowledge-Based Systems,* 36, pp. 322-339.

Duval, A. (2019). Explainable Artificial Intelligence (XAI). *MA4K9 Scholarly Report*, Mathematics Institute, The University of Warwick.

Doguc, O., & Ramirez-Marquez, J. E. (2009). A generic method for estimating system reliability using Bayesian networks. *Reliability Engineering & System Safety*, 94(2), pp. 542-550.

Freund, Y. and R. E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting." *J. of Computer and System Sciences,* Vol. 55, 1997, pp. 119–139.

Gath, I., Geva, A.B., Amir, B. (1989). Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), pp. 773–780.

Gebharter, A. (2017). Causal nets, interventionism, and mechanisms. Cham: *Springer*.

Gentner, D., & Smith, L. (2012). Analogical reasoning. *Encyclopedia of human behavior,* 2, pp. 130-136.

Ghassemi, M., Naumann, T., Schulam, P., Beam, A. L., Chen, I. Y., & Ranganath, R. (2020). A review of challenges and opportunities in machine learning for health. *AMIA Summits on Translational Science Proceedings*, pp. 191.

Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1), pp. 44-65.

Grathwohl, W., Choi, D., Wu, Y., Roeder, G., & Duvenaud, D. (2017). Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *arXiv* preprint arXiv:1711.00123.

Greenwell, B. M. (2017). pdp: An R Package for Constructing Partial. *Dependence Plots*. R J., 9(1), pp. 421.

Heinzer, R., Vat, S., Marques-Vidal, P., Marti-Soler, H., Andries, D., Tobback, N., ... & Vollenweider, P. (2015). Prevalence of sleep-disordered breathing in the general population: the HypnoLaus study. *The Lancet Respiratory Medicine*,3(4), pp. 310-318.

Holdefer, R. N., & Skinner, S. A. (2020). Motor evoked potential recovery with surgeon interventions and neurologic outcomes: A meta-analysis and structural causal model for spine deformity surgeries. *The Clinical Neurophysiology*, 131(7), pp. 1556-1566.

Hospital Sírio-Libanês (2020.7). COVID-19 - Clinical Data to assess diagnosis. Retrieved from https://www.kaggle.com/S%C3%ADrio-Libanes/covid19

Hox, J. J., & Maas, C. J. (2001). The accuracy of multilevel structural equation modeling with pseudo balanced groups and small samples. *Structural equation modeling*, 8(2), pp. 157-174.

Hsu, T. H. (2000). An application of fuzzy clustering in group-positioning analysis. *Proceedings National Science Council of the Republic of China*, 10(2), pp. 157-167.

Huang, S. F., Lin, Y. H., Yih, J. M., & Tseng, J. C. (2018). Fuzzy clustering algorithm based on Mahalanobis Distances with recursive process. *International Journal of Intelligent Technologies & Applied Statistics*, 11(4), pp. 221-239.

IBM Cloud Education (2021.4). Knowledge Graph. Retrieved from https://www.ibm.com/cloud/learn/knowledge-graph

Imbens, G. W., & Rubin, D. B. (2010). Rubin causal model. *In Microeconometrics,* pp. 229-241. Palgrave Macmillan, London.

Imbens, G. W., & Rubin, D. B. (2015). Causal inference in statistics, social, and biomedical sciences. *Cambridge University Press*.

Keppens, J. (2019). Explainable Bayesian Network Query Results via Natural Language Generation Systems. *In Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pp. 42-51.

Kim, M., Lu, F., & Raghavan, V. V. (2000). Automatic construction of rule-based trees for conceptual retrieval. *In Proceedings Seventh International Symposium on String Processing and Information Retrieval*. SPIRE 2000, pp. 153-161. IEEE.

Krieger, J., Sforza, E., Boudewijns, A., Zamagni, M., & Petiau, C. (1997). Respiratory effort during obstructive sleep apnea: the role of Age and sleep state. *Chest*, 112(4), pp. 875-884.

Krishnan, G. J., Ng, T., Ng, S. K., Krishnan, T., & Mclachlan, G. J. (1997). *The EM algorithm*. In Wiley Series in Probability and Statistics: Applied Probability and Statistics, WileyInterscience.

Kumaraswamy, R., Odom, P., Kersting, K., Leake, D., & Natarajan, S. (2015, November). Transfer learning via relational type matching. *In 2015 IEEE International Conference on Data Mining*, pp. 811-816. IEEE.

Kumaraswamy, R., Ramanan, N., Odom, P., & Natarajan, S. (2020). Interactive Transfer Learning in Relational Domains. *KI-Künstliche Intelligenz*, pp.1-12.

Lavrac, N., & Dzeroski, S. (1994). Inductive Logic Programming. *In WLP*, pp. 146-160.

Lin, P.L., Huang, P.W., Kuo, C.H., Lai, Y.H. (2014). A size-insensitive integrity-based fuzzy c-means method for data clustering. *Pattern Recognition*, 47(5), pp. 2042–2056

Li, L., Wang, P., Yan, J., Wang, Y., Li, S., Jiang, J., et al. (2020). Real-world data medical knowledge graph: construction and applications. *Artificial intelligence in medicine*, 103, 101817.

Liu, H. C., Jeng, B. C., Yih, J. M., & Yu, Y. K. (2009). Fuzzy C-means algorithm based on standard Mahalanobis distances. *In Proceedings. The*

2009 International Symposium on Information Processing (ISIP 2009) pp. 422. Academy Publisher.

Ma, Z., Tavares, J. M. R., Jorge, R. N., & Mascarenhas, T. (2010). A review of algorithms for medical image segmentation and their applications to the female pelvic cavity. *Computer Methods in Biomechanics and Biomedical Engineering*, 13(2), pp. 235-246.

Mansukhani, M. P., Kolla, B. P., & Somers, V. K. (2019). Hypertension and Cognitive Decline: Implications of Obstructive Sleep Apnea. *Frontiers in cardiovascular medicine*, 6(9).

Maulik, U., & Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on pattern analysis and machine intelligence*, 24(12), pp.1650-1654.

Mendelson, M., Bailly, S., Marillier, M., Flore, P., Borel, J. C., Vivodtzev, I., ... & Pépin, J. L. (2018). Obstructive sleep apnea syndrome objectively measured physical activity and exercise training interventions: a systematic review and meta-analysis. *Frontiers in Neurology*, 9(73).

Mohamed, N. A., Ahmed, M. N., & Farag, A. (1999). Modified fuzzy c-mean in medical image segmentation. *In 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*. ICASSP99 (Cat. No. 99CH36258) Vol. 6, pp. 3429-3432. IEEE.

Molnar, C. (2020). Interpretable machine learning. *Lulu. com*.

Nayak, J., Naik, B., & Behera, H. S. (2015). Fuzzy C-means (FCM) clustering algorithm: a decade review from 2000 to 2014. *In Computational intelligence in data mining*, 2, pp. 133-149. *Springer*, New Delhi.

Neto, E. C. (2020). Towards causality-aware predictions in static machine learning tasks: the linear structural causal model case. *arXiv* preprint arXiv:2001.03998.

Noordam, J.C., van den Broek, W.H.A.M., Buydens, L.M.C. (2002) Multivariate image segmentation with cluster size insensitive Fuzzy C-means. *Chemometrics and Intelligent Laboratory Systems*, 64(1), pp. 65–78.

Odom, P., Khot, T., Porter, R., & Natarajan, S. (2015). Knowledge-based probabilistic logic learning. *In Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Omran, P. G., Wang, K., & Wang, Z. (2016). Transfer learning in probabilistic logic models. *In Australasian Joint Conference on Artificial Intelligence*, pp. 378-389. Springer, Cham.

Pan, J. Z., Vetere, G., Gomez-Perez, J. M., & Wu, H. (2017). Exploiting linked data and knowledge graphs in large organization, pp. 281. Heidelberg: *Springer*.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), pp. 1345-1359.

Pan, S. J., Tsang, I. W., Kwok, J. T., & Yang, Q. (2010). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2), pp. 199-210.

Pearl, J. (2018). Theoretical impediments to machine learning with seven sparks from the causal revolution. *arXiv* preprint arXiv:1801.04016.

Peng, G., Wang, H., Zhang, H., & Huang, K. (2019). A hypernetwork-based approach to collaborative retrieval and reasoning of engineering design knowledge. *Advanced Engineering Informatics*, 42, 100956.

Quan S F, Howard B V, Iber C, et al (1997). The sleep heart health study: design, rationale, and methods. *Sleep*, 20(12), pp.1077-1085.

Quan, S. F., Budhiraja, R., & Kushida, C. A. (2018). Associations between sleep quality, sleep architecture and sleep disordered breathing and memory after continuous positive airway pressure in patients with obstructive sleep apnea in the Apnea Positive Pressure Long-term Efficacy Study (APPLES). *Sleep Science*, 11(4), pp. 231.

Ribeiro, M., Singh, S., & Guestrin, C. (2019). Local Interpretable Model-Agnostic Explanations (LIME): An Introduction.

Roscher, R., Bohn, B., Duarte, M. F., & Garcke, J. (2020). Explainable machine learning for scientific insights and discoveries. *IEEE Access*, 8, pp. 42200-42216.

Rotmensch, M., Halpern, Y., Tlimat, A., Horng, S., & Sontag, D. (2017). Learning a health knowledge graph from electronic medical records. *Scientific reports*, 7(1), 1-11.

Samek, W., Wiegand, T., & Müller, K. R. (2017). Explainable artificial intelligence: Understanding, visualizing, and interpreting deep learning models. *arXiv* preprint arXiv:1708.08296.

Senaratna, C. V., Perret, J. L., Lodge, C. J., Lowe, A. J., Campbell, B. E., Matheson, M. C., ... & Dharmage, S. C (2017). Prevalence of obstructive sleep apnea in the general population: a systematic review. *Sleep medicine reviews*, 34, pp.70-81.

Shi, D., Wang, T., Xing, H., & Xu, H. (2020). A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning. *Knowledge-Based Systems*, 195, 105618.

Singhal, A. (2012). Introducing the knowledge graph: things, not strings. *Official Google blog*, pp. 5.

Smiti, A., & Elouedi, Z. (2016, September). Fuzzy density-based clustering method: Soft DBSCAN-GM. *In 2016 IEEE 8th International Conference on Intelligent Systems (IS)*, pp. 443-448. IEEE.

Steinmetz, H., Isidor, R., & Baeuerle, N. (2012, April). Testing the circular structure of human values: A meta-analytical structural equation modeling approach. *Survey Research Methods* 6(1), pp. 61-75.

Sun, B., Feng, J., & Saenko, K. (2016, March). Return of frustratingly easy domain adaptation. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Tan, S., Caruana, R., Hooker, G., Koch, P., & Gordo, A. (2018). Learning global additive explanations for neural nets using model distillation. *arXiv* preprint arXiv:1801.08640.

Tao, L., Cichen, W., & Huakang, L. (2017). Development and construction of knowledge graph. *J Nanjing Univ Sci Technol*.

Tenorth, M., & Beetz, M. (2013). KnowRob: A knowledge processing infrastructure for cognition-enabled robots. *The International Journal of Robotics Research*, 32(5), 566-590.

Torrey, L., & Shavlik, J. (2010). Transfer learning. *In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pp. 242-264. IGI global.

Ullman, J. B., & Bentler, P. M. (2003). Structural equation modeling. *Handbook of psychology*, pp. 607-634.

Wang, F., Jiang, Z., Li, X., & Li, G. (2021). Cognitive factors of the transfer of empirical engineering knowledge: A behavioral and fNIRS study. *Advanced Engineering Informatics*, 47, 101207.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). *A survey of transfer learning. Journal of Big data*, 3(1), pp. 9.

Wu, J. T., Leung, K., Bushman, M., Kishore, N., Niehus, R., de Salazar, P. M., ... & Leung, G. M. (2020). *Estimating the clinical severity of COVID-19 from the transmission dynamics in Wuhan, China*. Nature Medicine, 26(4), pp. 506-510.

Zhang G Q, Cui L, Mueller R, et al (2018). The National Sleep Research Resource: towards a sleep data commons. *Journal of the American Medical Informatics Association*, 25(10), pp.1351-1358.

Zhou, Z., Cai, H., Rong, S., Song, Y., Ren, K., Zhang, W, & Wang, J. (2017). Activation maximization generative adversarial nets. *arXiv* preprint arXiv:1703.02000.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE.*

Zhuang, X., Huang, Y., Palaniappan, K., & Zhao, Y. (1996). Gaussian mixture density modeling, decomposition, and applications. *IEEE Transactions on Image Processing*, 5(9), pp. 1293-1302.

Publications related to this thesis

Original Papers

Li, J., Horiguchi, Y., & Sawaragi, T. (2020). Cluster Size-Constrained Fuzzy c-Means with Density Center Searching. *International Journal of Fuzzy Logic and Intelligent Systems*, 20(4), pp. 346-357.

Li, J., Sawaragi, T., & Horiguchi, Y. (2021). Introduce structural equation modelling to machine learning problems for building an explainable and persuasive model. *SICE Journal of Control, Measurement, and System Integration*, 14(2), pp. 67-79.

Li, J., Sawaragi, T., & Horiguchi, Y. (2022). Counterfactual Inference to Predict Causal Knowledge Graph for Relational Transfer Learning by Assimilating Expert Knowledge--Relational Feature Transfer Learning Algorithm. *Advanced Engineering Informatics.* (Second time review completed)

International Conference

Li, J., Horiguchi, Y., & Sawaragi, T. (2018). Refining Fuzzy c-Means Membership Functions to Assimilate A Priori Knowledge of Cluster Sizes. *In 2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)* pp. 654-659. IEEE;

Li, J., Horiguchi, Y., & Sawaragi, T. (2019). A Wrapper Algorithm for Fuzzy Cluster Membership Modification Based on Size Constraints. *ISIS2019& ICBAKE 2019*;

Li, J., Horiguchi, Y., & Sawaragi, T. (2020). Data Dimensionality Reduction by Introducing Structural Equation Modeling to Machine Learning

Problems. *In 2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)* pp. 826-831. IEEE.