Artificial Intelligence, Machine Learning and Reasoning in Health Informatics —An Overview

Chapter · October 2020

DOI: 10.1007/978-3-030-54932-9_7

3 authors:

Mobyen Uddin Ahmed
Mälardalen University
150 PUBLICATIONS **2,034** CITATIONS
SEE PROFILE

Shaibal Barua
Mälardalen University
48 PUBLICATIONS **508** CITATIONS
SEE PROFILE

Shahina Begum
Mälardalen University
115 PUBLICATIONS **1,443** CITATIONS
SEE PROFILE

Artificial Intelligence, Machine learning and Reasoning in Health Informatics – An Overview

Mobyen Uddin Ahmed, Shaibal Barua, Shahina Begum

School of Innovation, Design and Engineering, Mälardalen University, Sweden

{mobyen.ahmed@mdh.se, shaibal.barua@mdh.se, shahina.begum@mdh.se}

Abstract. As humans are intelligent, to mimic or models of human certain intelligent behavior to a computer or a machine is called Artificial Intelligence (AI). Learning is one of the activities by a human that helps to gain knowledge or skills by studying, practising, being taught, or experiencing something. Machine Learning (ML) is a field of AI that mimics human learning behavior by constructing a set of algorithms that can learn from data, i.e. it is a field of study that gives computers the ability to learn without being explicitly programmed. The reasoning is a set of processes that enable humans to provide a basis for judgment, making decisions, and prediction. Machine Reasoning (MR), is a part of AI evolution towards human-level intelligence or the ability to apply prior knowledge to new situations with adaptation and changes. This book chapter presents some AI, ML and MR techniques and approached those are widely used in health informatics domains. Here, the overview of each technique is discussed to show how they can be applied in the development of a decision support system.

1. Introduction

In the medical/health informatics domain, the knowledge is today expanding so quickly to the extent that even experts have difficulties in following the latest new results, changes and treatments. Computers surpass humans in their ability to remember and such property is very valuable for a computer-aided system that enables improvements for both diagnosis and treatment. Decision Support System (DSS) are computer-based systems that can simulate expert human reasoning by understanding the principle of human intelligence, can be served as an assistant to a physician in the medical domain is increasingly important and. They are typically designed for medical knowledge, patient's data/information and an inference engine in to assist clinicians in their decision-making tasks namely diagnosis and treatment. In the medical domain diagnostics, classification and treatment are the main tasks for a physician. System development for such a purpose is also a popular area in Artificial Intelligence (AI) research. DSSs that bear similarities with human learning and reasoning have benefits and are often easily accepted by physicians in the medical domain [2-7]. Today many clinical DSSs are developed to be multi-purposed and often combine more than one AI methods and techniques. The multi-faceted and complex nature of the medical domain motivates researchers to design such multi-modal systems [6-8]. Several challenges associated with multi-modal machine learning framework [9] have been addressed, which are:

- Representation: Data processing requires for representing and summarizing heterogeneous data to achieve multiple complementary modalities.
- Translation: Identify and acknowledge the relationship among the multi-modal data.
- Alignment: Identify the relations between different health related conditions and the measuring parameters.
- Fusion: Apply data-level and feature-level fusion between two or more modalities to improve system performance.
- Co-learning: Explore the advantages and limitations of each modality and use that knowledge to improve the performances of models trained on a different modality.

This book chapter presents AI technique, such as Fuzzy Logic. Several supervised machine learning e.g. Support Vector Machine (SVM) and Random Forest (RF); unsupervised learning e.g. K-means clustering, Fuzzy C-means (FCM) clustering, Gaussian mixer model and Hierarchical clustering. Also, several machine reasoning algorithms such as Fuzzy Rule-Based Reasoning (RBR), Case-Based Reasoning (CBR) and Textual Case Retrieval.

2. Overview of Artificial Intelligence (AI)

Artificial Intelligence (AI) is an area of computer science that is used to develop intelligent machines to think and react like humans. As humans are intelligent, to mimic or models or simulation of humans' certain intelligent behaviour to a computer or a machine is called AI. Even though the concept of AI technology has been changed over time, the fundamental concept of AI is building machines that are capable of thinking (i.e., thought processes and reasoning) and reacting (i.e., behaviour and performance) like humans.

Arthur Lee Samuel coined the term “machine learning” in his 1959 paper [10], who was an American pioneer in computer gaming and artificial intelligence. He defined machine learning as the process of programming a digital computer that could behave similarly to the way that human beings or animals learn while doing some task. One of the popular AI algorithms e.g. fuzzy logic has been used in the case studies (discussed in another chapter later) of health informatics domain are presented.

2.1 Fuzzy Logic

Information can be incomplete, inconsistent, uncertain, or all of these three and it is often unsuitable for solving a problem. For example, “The temperature of the machine is *really hot,* or Anders is a *very tall* man.” Here, it can be seen that most of the time we rely on common sense when we solve problems. To deal with such vague and uncertain information exact mathematical techniques are not sufficient, we need a technique or approach that uses a much closer concept of human thinking. Fuzzy logic is specifically designed to mathematically represent this uncertainty and vagueness. So, fuzzy logic is not a logic that is fuzzy, but a logic that is used to describe fuzziness. It is a theory of fuzzy sets, sets that calibrate vagueness. Moreover, it is a form of multi-valued logic with more than two truth values to deal with reasoning i.e. an approximate value rather than an exact value. Opposite to the binary or crisp logic, it handles the perception of ‘Partially Truth’ i.e. the values between completely ‘true’ and completely ‘false’. The degree of truth of a statement can range between false (0) and true (1) and considers more than two truth values. Aristotle was the first to realize that logic based on “True” or “False” alone was not satisfactory. Plato left the foundation of a third region beyond the true and false [11]. Multi-valued logic was introduced by a Polish philosopher Jan Lukasiewicz in the 1930s. He presented logic that extended the range of truth values to all real numbers in the interval between 0 and 1 [12,13]. In 1965, Lotfi Zadeh a professor at the University of California at Berkley, available his famous paper “Fuzzy sets”. He prolonged the work on possibility theory into a formal system of mathematical logic and initiated a new concept for applying natural language terms. This new logic for demonstrating and manipulating fuzzy terms was called fuzzy logic [14,15]. The term “fuzzy logic” derives from the fuzzy set theory or the theory of fuzzy sets. The fuzzy set theory has successfully been applied in controlling uncertainties in various application domains [16] including the medical application areas.

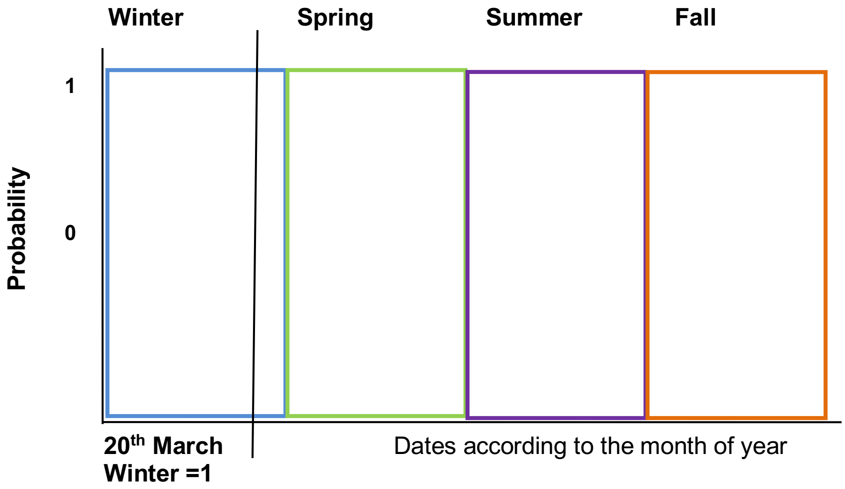

Figure 1. Binary or crisp logic representation for the season statement

The use of fuzzy logic in medical informatics has begun in the early 1970s. Figure 1 illustrated a binary logic with a crisp boundary of 4 different seasons in Sweden; where the X-axis corresponds to dates according to the month of the year and the Y-axis represents the probability between zero and one. In binary logic, the function that relates to the value of a variable with the probability of a judged statement is a 'rectangular' one. The output probability for any input will always be 'one' i.e. only one season and 'zero' for the rest of the seasons. The crisp boundary of the season winter drawn at 31st March and 20th March is winter with the probability of one. In fuzzy logic, the function can take any shape. As the season example illustrated, with the Gaussian curve in Figure 2, here, the X-axis is the universe of discourse which shows the range of all possible days for each month in a year for input. The Y-axis represents the degree of the membership function i.e. the fuzzy set of each season's day values into a corresponding membership degree. In fuzzy logic, the truth of any statement becomes a matter of degree. Considering the 20th March as an input in the fuzzy system, it is winter with the degree of truth 0.78 and at the same time spring with the degree of truth 0.22. So according to Zadeh [14], "*Fuzzy logic is determined as a set of mathematical principles for knowledge representation based on degrees of membership rather than on crisp membership of classical binary logic*".

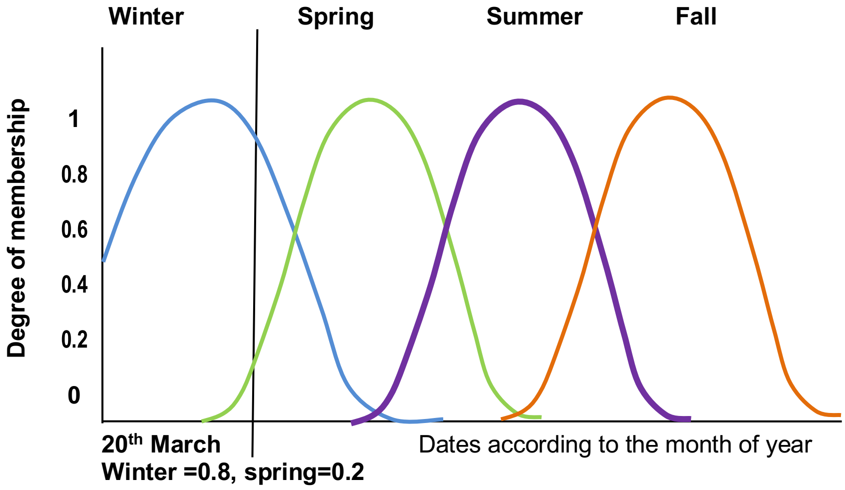

Figure 2. Fuzzy logic representation of the season statement

3. Overview of Machine Learning (ML)

As stated earlier, simulation of humans' certain intelligent behaviour is AI, there are several ways to do this. For example, it can be a set of 'if-then statements', rules explicitly programmed by a human hand. Machine learning is a subset of AI and thus all machine learning counts as AI. According to –Tom Mitchell, 'a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E'. Machine learning provides automated data analysis and automates analytical model building by detecting patterns in the data. Machine learning methods can also predict the patterns of future data and aid in decision making under uncertainty [17]. When there is no analytical solution exists, but data are available in that problem domain then machine learning is useful to build an empirical solution. Arthur Lee Samuel in his experiment was teaching a machine to play the game of checkers, subsequently, machine learning research diverse to many areas where often the focus is to find relationships in data and analysing the processes for extracting such relations [10]. This book chapter focuses on the applications of machine learning, here, both supervised and unsupervised (e.g. clustering) are considered.

3.1 Supervised Machine Learning Algorithms

Supervised classification problems involve an input space (i.e., the instances of χ) and an output space (e.g., the labelling of Υ). An unknown target function $f: \chi \rightarrow \Upsilon$ defines the functional relationship between the input space and output space. As mentioned above, a dataset D exists containing input-output pairs $(\chi_1, \Upsilon_1), \ldots \ldots, (\chi_n, \Upsilon_n)$ drawn as an independent and identical distribution (i.i.d) from an unknown underlying distribution $P(\chi, \Upsilon)$. The goal is to find a function $g: \chi \rightarrow \Upsilon$ that can approximate the solution of f with minimum errors. The function $g: \chi \rightarrow \Upsilon$ is called a classifier [18].

3.1.1 Support Vector Machine (SVM)

The SVM is a supervised machine learning method first developed by Vapnik [19] and is now commonly used in pattern recognition: it can be used for both classification and regression purposes [20,21]. An SVM finds the hyperplane that not only minimizes the empirical classification error but also maximizes the geometric margin of the classification [19]. SVM maps the original data points in the input space to a high dimensional feature space, making the classification problem simpler. Hence, SVM is suitable for classification problems with redundant datasets [22]. Consider an n-class classification problem with a training data set $\{\chi_i, \Upsilon_i\}_{i=1}^{n}$, where $\chi_i \in \mathbb{R}^d$ is the input vector, and Υ_i is the corresponding class label. The SVM maps the d-dimensional input vector space to a d_h-dimensional feature space and learns the separating hyperplane $\langle w, \chi \rangle + b = 0, b \in \mathbb{R}$ that maximizes the margin distance $\frac{2}{\|w\|_2^2}$, where w is a weight vector, and b is the bias. The SVM classifier obtains a new label $\widehat{\Upsilon}$ for the test vector by evaluating Equation (1):

$$\widehat{\Upsilon} = \sum_{i=1}^{N} w_i . K(\chi, \chi_i) + b \tag{1}$$

where N is the number of support vectors, w_i are the weights, b is the bias that is maximized during training and K is the kernel function.

The solid line represents the optimal hyperplane, dotted line denotes maximal margin; circles and diamonds on the margin are the support vectors [23]. Here, w is the weight vector and b is the threshold such that

$\Upsilon_i(\langle w, \chi_i\rangle + b) > 0\ (i = 1, \dots\dots, N)$. In this study, the Radial Basis Function (RBF) kernel was used for classification. The RBF can be denoted as $K(\chi, \Upsilon) = exp\left(-\frac{\|\chi-\Upsilon\|^2}{2\sigma^2}\right)$, where σ is the variance of the Gaussian. An SVM with RBF is a weighted linear combination of the kernel function computed between the data points and each of the support vectors. Figure 3 depicts an example of binary classification with linear separability.

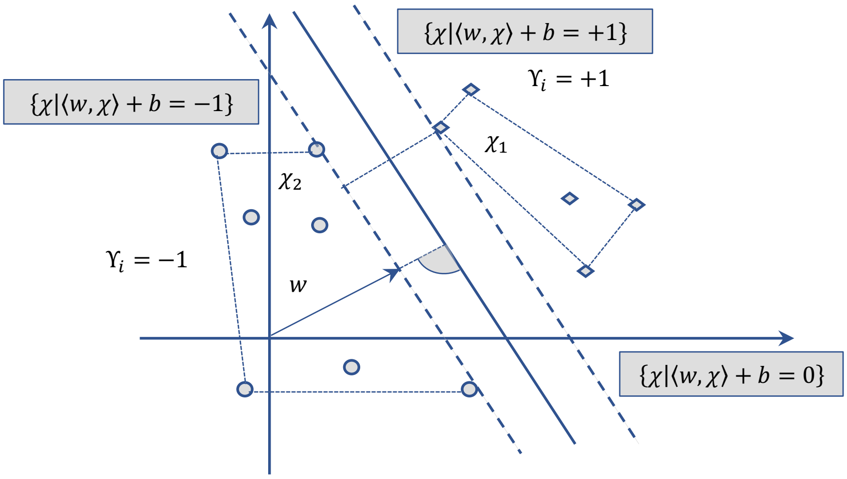
$\{\chi|\langle w,\chi\rangle + b = +1\}$
$\Upsilon_i = +1$
$\{\chi|\langle w,\chi\rangle + b = -1\}$
χ_1
χ_2
$\Upsilon_i = -1$
w
$\{\chi|\langle w,\chi\rangle + b = 0\}$

Figure 3. An example of SVM separation of 2-dimensional binary class problem

3.1.2 Random Forest (RF)

RF is a standard ensemble algorithm in machine learning that consists of a series of randomizing decision trees [24]. Each decision tree in the random forest is trained using bootstrap data samples, where bootstrapping is the process of creating samples with replacement. During the bootstrapping process, not all data are selected for training; the selected data are referred to as out-of-bag data, and these out-of-bag data are used to find the generalization error or the out-of-bag error. A generic architecture for a random forest classifier is shown in Figure 4.

During the tree-generation process, for the *k-th* tree, a random vector υ_k is generated, which is drawn from the same data distribution but independent of previous random vectors $\upsilon_1, \dots\dots, \upsilon_{k-1}$. For the given training dataset, the tree grows using the random vectors υ_k and creates a predictor $h(\chi, \mathrm{X}_k, \upsilon_k)$, where χ is the input data, X_k is the bootstrap sample, and υ_k

consists of a number of independent random variables m between 1 and K. Different generalizations can be achieved by varying the number of variables; it is recommended to start the search from $m = \lfloor \log_2 K + 1 \rfloor$ or $m = \sqrt{K}$ [25,24]. After generating a large number of trees, the output is the majority vote of all these decision trees. The important aspects of a random forest are that as the forest grows by adding more trees, it will converge to a limiting value that reduces the risk of overfitting and does not assume feature independence. RF is implemented using bagging, which is the process of bootstrapping the data plus using aggregation to make a decision.

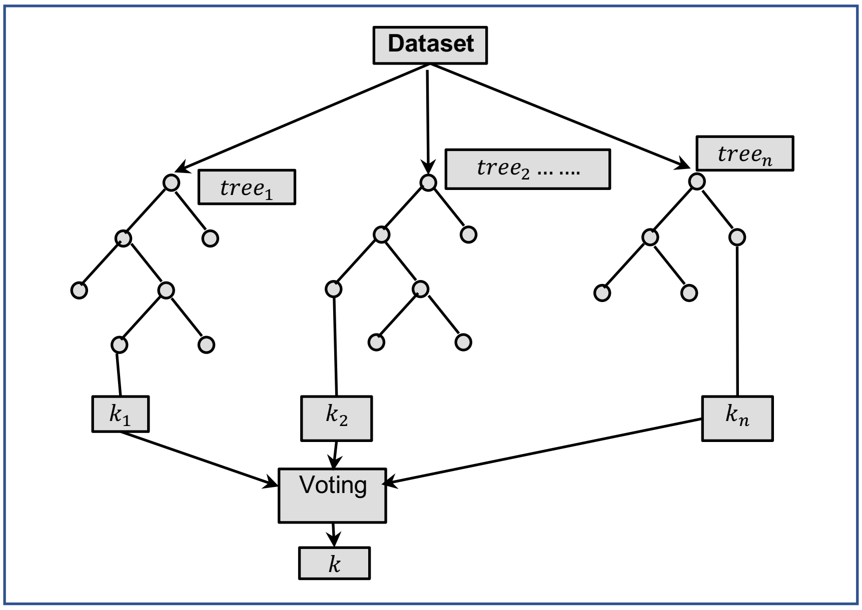

Figure 4. Generic structure of random forest

3.1.3 Artificial Neural Network (ANN)

In the artificial intelligence paradigm, one of the most popular learning methods is the Artificial Neural Network (ANN). ANN is a method that is vaguely inspired by the biological nervous system. It is composed of interconnected elements called neurons that work in unity to solve specific problems [26,27,21]. The neurons are connected through links, and numerical weight is assigned to each neuron. This weight represents the strength or importance of each neuron input and repeated adjustment of the weights are performed to learn from the input. Various types of neural networks are described in [21], and one of the most popular network architectures is the multilayer feedforward neural network methods using backpropagation to adapt/learn the weights and biases of the nodes. Such networks consist of an input layer that represents the input variables to the problem, an output layer consisting of nodes representing the dependent variables or the classification label, and one or more hidden layers that contain nodes to help capturing the nonlinearity of the data. The error is computed at the output layer and propagates backwards from the output layer to the hidden layer, then hidden layer to the input layer. A three-layer backpropagation neural network is shown in *Figure 5*.

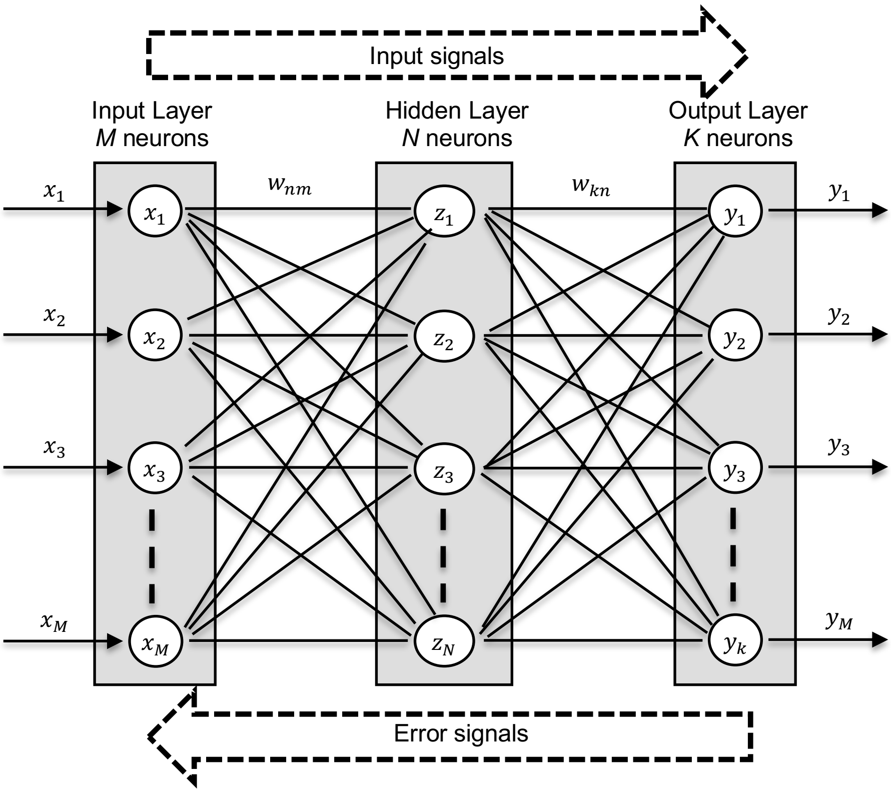

Figure 5. Multi-layer perceptron with backpropagation [28]

3.1.4 K-Nearest Neighbor (k-NN)

K-nearest neighbor (k-NN) is a non-parametric approach and one of the simplistic machine learning algorithms. It does not need to fit the data, which makes it flexible in the sense that k-NN is a memory-based algorithm which uses the observations in the training set to find the most similar properties of the test dataset [29]. In other words, k-NN classifies an unseen instance using the observations that have closest match similarity (k-number nearest neighbors) to it. In the statistical settings, k-NN does not make any assumptions about the underlying joint probability density, rather uses the data to estimate the density.

In the k-NN algorithm a distance function e.g., the Euclidean distance function is often used to find the k most similar instances. Then methods like majority voting is used on the k-neighbor instaces that indicates most commonly occurring classification to make the final classification. The bias-variance trade-off of k-NN depends of the selection of k, i.e. the number of nearest neighbors to be considered. As the value of k gets larger the estimation smoothed out more. Since k-NN is based on a distance function, it is straightforward to explain the nearest-neighbour model when predicting a new unseen data. However, it may be difficult to explain what inherent knowledge the model has learned. Further, the assumption of closest similarity that is similar data shares similar classification has a drawback in high dimensional feature spaces i.e., similar data may not be close together in the high dimensional spaces. Thus, it requires to ensure that there are enough data that rationalize the dimensionality of the feature space and the data density.

3.2 Unsupervised Machine Learning Algorithms

Unlike supervised learning, unsupervised learning does not have a defined output space (e.g., the labelling of ϒ). Here, the input space consists of a set of n observations $(\chi_1, \chi_2, \dots \dots \chi_n)$ of a random ρ-vector χ that has joint probability density $Pr(\chi)$. The goal of unsupervised learning is to directly infer this probability density i.e., the underlying structure in the data without the help of any supervisor or labelled data ϒ that can tell the degree-of-error for each observation [18].

Clustering is an approach of unsupervised machine learning that a set of data is divided into several subsets where the data within one subset are similar to each other and are different from the data of other subsets. The clustering approach or cluster analysis is not an algorithm itself rather it is a task to be solved by applying various algorithms. A formal definition can be presented as “These clusters should reflect some mechanism at work in the domain from which instances or data points are drawn, a mechanism that causes some instances to bear a stronger resemblance to one another than they do to the remaining instances" [30]. A mathematical definition of clustering as stated in [31], which can be express as follows: let $X \in R^{m \times n}$ in a set of data representing a set of m points x_i in R^n. The goal is to partition X into K groups and C_k so that all data that belongs to the same group are more "alike" than data in different groups. Each of the K groups are called a cluster and the result of the algorithm is an injective mapping $X \longmapsto C$ of data items X_i to clusters C_k. Several algorithms are available in literature with many different classifications. However, one simple classification of clustering can be divided into two classes as: 1) parametric and 2) non-parametric clustering. *Parametric* clustering helps to minimize a cost function where the main goal of this kind of algorithm is to solve an optimization problem in a satisfactory level imposed by the model. However, this algorithm requires a better understanding about data distribution and a proper probability distribution. This class can be further divided into two groups: a) generative or probability-based model and b) reconstructive models. In the probability-based model, the model relies on a guess that the data comes from a known distribution, but this is not true for many situations. So, this model cannot be usefully applied where the probability distribution is not known and/or the data are not numerical. The Gaussian mixture model is one example of such model. However, a proper probability distribution in data can be achieved using this algorithm. On the other hand, the reconstructive model aims to minimize the cost function. A most common and basic algorithm is K-means as an example of reconstructive models. For *non-parametric* clustering, the hierarchical algorithms or an agglomerative and divisive algorithm is a good example. The algorithm works based on dis-similarities among the current clusters for each iteration. The agglomerative algorithm merges and the divisive algorithm divides the clusters depending on similarities. Both of them also produce dendrograms, which presents clusters in a tree structure as bottom up or top down. A detailed elaboration of parametric and non-parametric clustering can be found in [32] and the difference between parametric and non-parametric clustering can be summarized in Table 1.

Table 1. Comparison of parametric and non-parametric clustering.

Criteria	Parametric	Non-parametric
Algorithm	Optimises a cost function	Density-based method
	Most costs are NP-hard problem	No cost functions
	Assumes more detailed knowledge of cluster	Does not depend on initialisation
	Assumes K is known	K and outliers selected automatically
	Gets harder with larger K	

When to use	Older, more widely used and studied Shape of clusters is known K not too large or known Clusters of comparable size	Requires hyper-parameters Shape of cluster is arbitrary K is large or has many outliers Cluster size in large range Lots of data

A summary of the 4 common and well-known clustering algorithms 1) K-means clustering, 2) Fuzzy C-means clustering, 3) Gaussian mixer model and 4) Hierarchical clustering are presented here:

3.2.1 K-means Clustering

K-means clustering expresses groups in a numeric domain and partitions data samples in disjointed groups. The main objective of the algorithm is to minimize the cost objective function and it requires the number of clusters and its initial centre points. The centre points can be given manually or randomly in the initial stage of the algorithm and later in each iteration the algorithm will automatically adjust in order to minimize the value of the distance matrix. Considering the distance matrix values, each iteration is repeated and as soon as the two distance values (previous and next) become the same, the algorithm stops. The Euclidean distance function is used in this algorithm in most of the cases and performance of the algorithm is strongly depends on the distance value. Although the algorithm is easy to implement and takes less time to compute compared to others, it has a drawback that it can be stuck in a local minimum since the algorithm depends on the provided initial centre point. The procedure starts work by giving a set of initial cluster numbers and the centre points for each cluster. Then the centre points are replaced by the mean point for each cluster. These steps are repeated until the two distances become the same. The algorithm can be illustrated as below [33]:

- Step 1. Choose K initial cluster centres $Z_1, Z_2, \dots\dots, Z_k$ randomly from the n points $\{X_1, X_2, \dots\dots, X_n\}$.

- Step 2. Assign point $X_i, i = 1, 2, \dots\dots, n$ to the cluster $C_j, j \in \{1,2, \dots\dots, K\}$, if

$\|X_i - z_j\| \leqq \|X_i - z_p\|, p = 1,2, \dots\dots, K$ and $j \neq p$

- Step 3. Calculate new cluster centers:

$$z_i^{new} = \frac{1}{n_i}\sum_{x_i \in C_i} X_j, i = 1,2, \dots, K$$

- Step 4. Continue step 2 and 3 and if $\|z_i^{new} - z_i\| < \varepsilon, i = 1, 2, \dots \dots, K$ then stop.

3.2.2 Fuzzy C-means (FCM) Clustering

FCM is also referred to as soft clustering; it is an unsupervised clustering algorithm that has been applied to a wide range of problems involving feature analysis, clustering and classifier designs. It is similar in structure to the K-means algorithm and also behaves in a similar way [34,35] except that fuzzy behaviour is also considered. It is a clustering algorithm where each data point belongs to a cluster to a degree specified by a membership grade whereas traditional clustering algorithms assign each data point to one and only one cluster. It is a clustering method that allows one piece of data to belong to two or more clusters. It associates each element that represents a set of membership levels. Thus, it creates the concept of fuzzy boundaries which is opposite from the traditional concept of well-defined boundaries. The algorithm is presented in several steps in Figure 6.

Step 1: U=[u_{ij}] matrix is initialize, $U^{(0)}$
Step 2: At k-step: calculate the centre vectors $C^{(k)}$=[c_j] with $U^{(k)}$
Step 3: Update $U^{(k)}$, $U^{(k+1)}$
Step 4: If || $U^{(k+1)}$ - $U^{(k)}$||< δ then STOP; otherwise return to the step 2,

where *m* is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster *j*, x_i is the i^{th} of d-dimensional measured data, c_j is the d-dimension centre of the cluster, and ||*|| is any norm expressing the similarity between any measured data and the centre. The equation of the cluster centres c_j and the updated membership matrix function u_{ij} are given bellow:

$$c_j = (\sum_{i=1}^{N} u_{ij}^m . x_i) / \sum_{i=1}^{N} u_{ij}^m \quad \textit{and} \quad u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\|x_i - c_j\| / \|x_i - c_k\| \right)^{2/(m-1)}}$$

Figure 6. Steps of the FCM clustering technique are taken from [97]

3.2.3 Gaussian Mixer Model

It is an example of a generative model where data are presented by a calculated Gaussian Mixture distribution of data points. Each distribution represents a different cluster and during clustering it computes the expectation maximisation of a data point. It is associated with fitting a set of data and identifies a set of Gaussian distribution that presents the highest probability for the data. The data is fitted using an expected maximisation algorithm that assigns probability to each component based on individual observations. This probability is sometimes also called as membership score or rank. Each data point has a membership score of belonging to each cluster. It appears to solve many problems related to other clustering techniques and has been identifying as yielding a more stable cluster especially when the requested number of clusters changes. The details of the model and the basic algorithm can be found in [32].

3.2.4 Hierarchical Clustering

The algorithm clusters data over a variety of scales by creating a hierarchical structure (tree) or dendrogram. The tree is not a single set of clusters, but rather a multilevel hierarchy, where clusters at one level are joined with clusters at the next level [36]. It is then further divided into two categories; bottom up i.e. agglomerative and the top down i.e. divisive clustering. This algorithm does not require any initialisation of centre points. To group the data together, a suitable proximity measure is used that estimates firstly the similarity between points and secondly the similarity between groups of points. It has several advantages, such as it starts with the number of clusters equal to the population of the initial data points then through an iterative process of grouping similar data points it finally ends up with a single cluster containing all the given data points. It makes it easy to catch the distance between clusters. If the agglomeration occurs between clusters at a greater distance than the previous agglomeration, one can decide whether to stop when the clusters are too far apart or when there is a sufficiently small number of clusters. However, it is not very efficient when it comes to dealing with large data. To perform agglomerative hierarchical clustering on a data set the algorithm uses the following procedures:

1. It calculates the distance between every pair of objects in the data set in order to find similarity or dissimilarity.

2. It collects or groups the objects into a binary, hierarchical cluster tree. Here, pairs of objects that are close to each other are linked. Since all the objects are paired into the binary clusters, newly formed clusters are grouped to larger clusters until a hierarchical tree is formed.

3. It determines the cutting position of the hierarchical tree into clusters. Here, it prunes the branches off at the bottom of the hierarchical tree and assigns all the objects below the cutting point to a single cluster.

In the hierarchical algorithm, the distance between pairs of objects is generally calculated using Euclidean distance. However, there are some other distance functions which can be used and available in MATLAB function ‘pdist’, such as Standardised Euclidean distance, cityblock, cosine, hamming, jaccard etc. Similarly, the linkage function applies ‘single’ (i.e. shortage distances) as a default parameter which determines the objects in the data set that should be grouped into clusters. However, other linkages can be used and available in MATLAB, such as average, centroid, complete etc. [32]. Finally, a cluster function is applied to group the sample data set into clusters where it specifies the cluster’s number.

4. Overview of Machine Reasoning (MR)

A program or software in a machine or a computer system that can perform reasoning to generate a conclusion form a knowledgebase using some logic or decision i.e. IF-THEN rules. This is an important area of AI approach to develop decision support systems [37]. According to Léon Bottou, machine reasoning is “A plausible definition of ‘reasoning’ could be algebraically manipulating previously acquired knowledge in order to answer a new question.” [38]. Similarly, Jerry Kaplan, describes the reasoning systems with two basic components including “knowledge base” – a collection of facts, rules and relationships and a “inference engine” that described how to manipulate [39].

4.1 Fuzzy Rule-Based Reasoning (RBR)

Fuzzy Rule-Based Reasoning is a combination of the fuzzy logic approach with traditional Rule Based Reasoning (RBR) which is also called Fuzzy

Inference Systems (FIS). Fuzzy inference is a computer paradigm based on fuzzy set theory, fuzzy if-then-rules and fuzzy reasoning. A traditional RBR system contains a set of if-then rules in a crisp format. A general form of a rule is "If <antecedent> then <consequence>". An example of such a rule is; "If speed is > 100 then stopping distance is 100 meters". In 1973, Lotfi Zadeh outlined a new approach to analyse complex systems, where human knowledge is captured as fuzzy rules [40,14]. A fuzzy rule is a linguistic expression of causal dependencies between linguistic variables in the form of if-then conditional statements. If we consider the previous example in a fuzzy format "If *speed* is ***fast,*** then stopping *distance* is ***long***". Here the term '*speed*' and '*distance*' are linguistic variables, while '***fast***' and '***long***' are linguistic values determined by fuzzy sets. Therefore '*speed* is ***fast***' is the antecedent and 'stopping *distance* is ***long***' is the consequence [14,15].

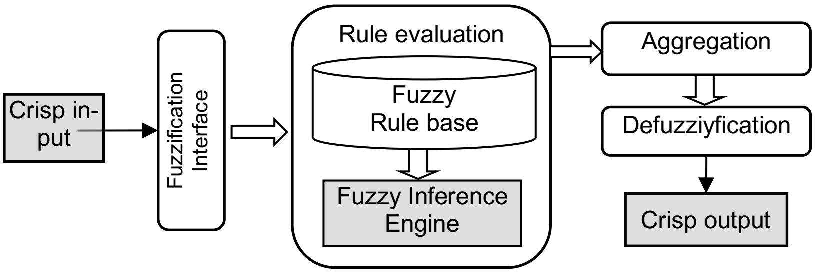

Figure 7. Steps in a Fuzzy Inference System (FIS).

Fuzzy decision making or inference systems can be defined as a process of mapping a given input to an output with the help of the fuzzy set theory i.e. fuzzification → fuzzy reasoning → defuzzification [16]. Well known inference systems are the Mamdani-style and Sugeno-style but both of them perform the 4-step process as described in Figure 7 which illustrates the steps of a fuzzy inference system for the Mamdani-style. Here, the *step1* is the fuzzification of an input variable i.e. crisp input is fuzzified against appropriate fuzzy sets. Given an input in a crisp format, *step1* computes the membership degree with respect to its linguistic terms. Consequently, each input variable is fuzzified over all the Membership Functions (*MF*s) used by the fuzzy rules. In a traditional rule-based system, if the antecedent part of a rule is true then the consequent part is also true. But in a fuzzy system, the rules are met to some extent. If the antecedent is true to some degree of membership, then the consequent is also true to that degree. *Step2* is the rule evaluation where it takes fuzzified inputs and applies them to the antecedent part of the fuzzy rules. So, it compares facts with the antecedents of the fuzzy rules to find degrees of compatibility. The value or firing strength is a single number from each rule represented in the result of the antecedent evaluation. This number is then applied to generate consequent MFs. Aggregation in *step3* is the process that merges all the output MFs for all the rules i.e. all outputs are combined into a single fuzzy set. The last and final phase (*step4*) in the inference process is defuzzification that determines a crisp value from the output membership function as a solution. The input for defuzzification is the aggregate fuzzy set and the output is a single number.

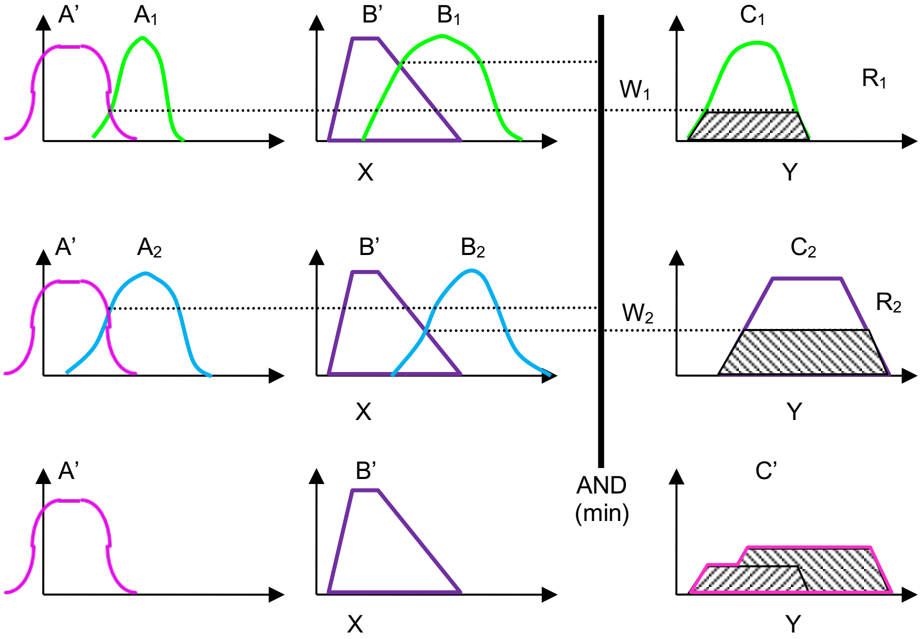

Figure 8. Graphical representation of an example of fuzzy inference

A simple example of fuzzy inference with multiple rules and multiple antecedents is illustrated in Figure 8. The rules and inputs are as follows: *Rule 1*: if x is A_1 and y is B_1 then z is C_1 and *Rule 2*: if x is A_2 and y is B_2 then z is C_2; Inputs: x is A and y is B then z is C (?). First the inputted crisp values (A and B) are converted into the fuzzy sets A' and B'. Then for the rule R_1 and $R_{2,}$ A' and B' are fuzzified with the fuzzy sets A_1, B_1 and A_2, B_2. The dotted line in Fig. 6 presents the clipped area of the membership functions in the antecedent part of the rules. As the rules contain multiple antecedents with AND operators, fuzzy intersection is used to obtain a single number that represents the evaluation result of the antecedents. W_1 and W_2 are the evaluation results applied to the MFs in the consequent part of the rules. Upward and downward diagonal patterns in the fuzzy sets C_1 and C_2

show the firing strengths for the rule's evaluation. After aggregation, the clipped fuzzy set C_1 and C_2, and the new fuzzy set C' are obtained. A defuzzification algorithm could convert this fuzzy set into a crisp value which is a single number that represents the final output [16,40].

4.2 Case-Based Reasoning (CBR)

Case-Based Reasoning (CBR) is a problem-solving method that gives priority to past experiences for solving current problems (solutions for current problems can be found by reusing or adapting the solutions to problems which have been solved in the past). Riesbeck & Schank presented CBR as, "A case-based reasoner solves new problems by adapting solutions that were used to solve old problems" [41]. The CBR method in a problem solving context can be described as follows: 1) given a particular problem case, the similarity of this problem with the stored problems in a case library (or memory) is calculated 2) one or more most similar matching cases are retrieved according to their similarity values 3) the solution of one of the retrieved problems is suggested for reuse by doing revision or possible adaptation (if needed e.g. due to differences in problem descriptions) 4) finally, the current problem case and its corresponding solution can be retained as a new solved case for further use [42]. The root of CBR can be traced from the work of Schank and his student at Yale University in the early 1980s but Watson presented in [43] that the research of CBR began in 1977. CYRUS [44,45] developed by Janet Kolodner, is the basic and earliest CBR system. She employed knowledge as cases and used an indexed memory structure. Other early CBR systems such as CASEY [46] and MEDIATOR [47] have been implemented based on CYRUS.

In the medical domain around the 1980s, early CBR systems were developed by Konton [46], and Braeiss [48,49]. According to Kolodner in [50] a case is a "contextualised piece of knowledge representing experience that teaches a lesson fundamental to achieving the goals of the reasoner". The *problem* part describes the condition of a case and the *solution* part presents advice or a recommendation to solve a problem. A comprehensive case structure has been proposed by Kolodner in [45] as follows: 1) a state with goal, 2) the solution 3) the outcome 4) explanations of results and 5) lessons learned. Further ahead, Bergmann et al. [51] classified case representation in the following three categories: a) feature vector representations or propositional cases b) structured representations or relational cases, and c) textual representations or semi-structure cases [51].

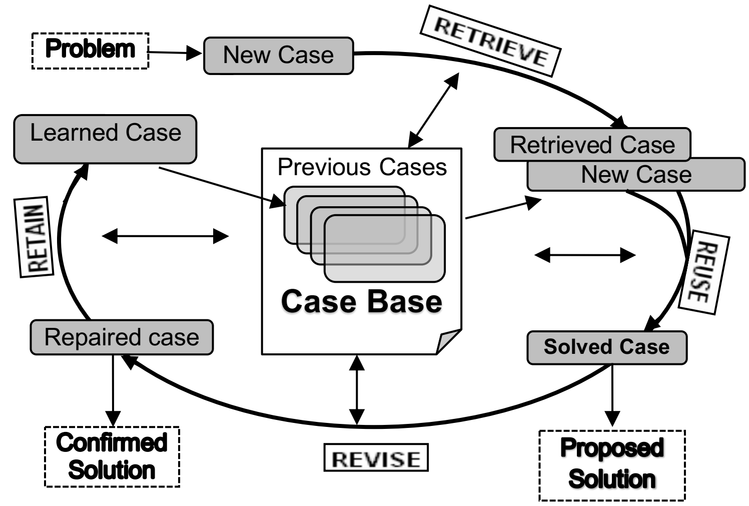

Figure 9. CBR cycle. The figure is introduced by Aamodt and Plaza [1].

A schematic or lifecycle that presents the key processes involved in the CBR method is shown in Figure 9. Aamodt and Plaza [1] have introduced a four-step model of CBR in a cyclical process comprising the four REs: Retrieve, Reuse, Revise and Retain. Retrieval is essential since it plays a vital role for calculating the similarity of two cases. The similarity value between cases is usually represented as 0 to 1 or 0 to 100, where "0" means no match and "1 or 100" means a perfect match. One of the most common and well-known retrieval methods is the *nearest neighbour* (or kNN) [52] which is based on the matching of a weighted sum of the features. For a feature vector, local similarity is computed by comparing each feature value and a global similarity value is obtained as a weighted calculation of the local similarities. A standard equation for the nearest-neighbour calculation is illustrated in Equation (2):

$$Similarity(T,S) = \frac{\sum_{i=1}^{n} f(T_i, S_i) \times w_i}{\sum_{i=1}^{n} w_i} \qquad (2)$$

In Equation 2:

T is the target case
S is the source case
n is the number of attributes in each case
i is an individual attribute from *1 to n*

f is a similarity function for attribute *i* in cases *T* and *S*

w is the importance for weighing of attribute *i*.

The weights allocated to each feature/attribute provide them a range of importance. But determining the weight for a feature value is a problem and the easy way is to calibrate this weight by an expert or user in terms of the domain knowledge. However, it may also be determined by an adaptive learning process i.e. learning or optimizing weights from the case library as an information source. The *Reuse* step is reusing one of the retrieved cases from the case library and returning it as the proposed solution for a current case. But in some cases, this phase can become more difficult, especially when there are notorious differences between the current case and the closest one retrieved. An *adaptation* of the obtained solution is required in order to provide a solution for the current problem. For adaptation, it could calculate the differences between the retrieved case and the current case. Then it is possible to apply algorithms or rules that take the differences into account to suggest a solution. This adaptation could be done by an expert/user in the domain. The expert determines if it is a reasonable solution to the problem and they can modify the solution before approval. After that the case is sent to the *Revise* step where the solution is verified and evaluated for the correctness and presented as a confirmed solution to the new problem case [52]. The term *Retain* becomes the final stage which functions as a learning process in the CBR cycle, and it incorporates the new solved case into the case library for future use. The most common way to retain a case is to simply record the information concerning the target problem specification and its final solution (assuming that the solution given was accurate and correct) [42]. If the solution retrieved is not as reliable as it should be, additional information might be stored into the case library such as the changes made to the retrieved solution. So, the information to be saved has to be considered carefully [53]. CBR is applied in a large number of medical applications as presented in [54].

4.3 Textual Case Retrieval

Bergmann et al. [51] have proposed that a case could be represented as a textual or semi-structural format. Textual case retrieval could be defined as matching a user query against a bunch of free-text cases. Text retrieval is a branch of Information Retrieval (IR) if the information is stored in the form of text. IR is a science used for searching documents and/or for information within a document or metadata about the document. In this research the knowledge of IR is used to search and retrieve cases with features containing information in a textual format. The idea of this process begins when a query is entered by a user into the system through a user interface. Then the system extracts information from the query. The extracted features may match with several objects (cases) in the collection (case library) with different degree of relevance. The degree of relevance can be computed by the system as a numerical value that shows how well each case is matched with the query. Finally, according to this numerical value, all the cases will be sorted, and the top ranked cases will be presented to the user [55]. There are several ways to find a match between a user query and the stored cases, such as Boolean model, fuzzy retrieval, vector space model, binary retrieval etc. [55]. The Vector Space Model (VSM) [56] is the most common and well-known method that has been used in information retrieval.

VSM or term vector model is an algebraic model that represents textual cases in a vector of terms. It identifies similarity between a query case *Q* and the stored cases *C*_i. One of the best-known schemes is the *tf-idf* (term frequency – inverse document frequency) [57] weighting used together with cosine similarity [58] in the vector space model [56] where the word "document" is treated as a case. The *tf-idf* is a traditional weighting algorithm and is often used in information and/or textual retrieval. The similarity/relevancy is measured from the cosine angle between a query case *Q* and the stored cases C_i inside a vector i.e. a deviation of angles between the case vectors. "$\cos\theta = \frac{Q.C_i}{\|Q\|\|C_i\|}$" is a general equation to calculate the cosine similarity where $Q.C_i$ is the dot product and $\|Q\|\|C_i\|$ is the magnitude of the vectors (a query and the stored case), i is the index of the cases in the case library. The value of the similarity lies in the range of -1 to +1, where -1 means no matching and +1 means exactly the same. In terms of IR, the cosine similarity of two cases will range from 0 to 1, since the *tf-idf* weights cannot be negative. The final result 1 is a full match and 0 means no words match between Q and C_i. To measure the similarity we need two things, the weight of each term in each case and the cosine similarity between the cases inside a vector space.

The terms are words, keywords, or long phrases in a case and the dimension of the vector is the number or frequency of each term in the vocabulary of cases. If a term occurs in a case the value will be non-zero in the vector. Each word *tf* is the relative frequency of the word in a specific case (document represent as a case) and it presents the importance of the word inside the case. *idf* is the inverse proportion of the word over the whole case corpus which presents the importance of the word over the entire case pool. The weight vector for a case c is $V_c = \left[w_{1,c}, w_{2,c}, \ldots, w_{N,c}\right]^T$ and $w_{t,c} =$

$tf_t . \log \frac{|C|}{|\{t \in c\}|}$ where tf_t is the term frequency or the number of times a term/word *t* occurs in a case *c* and is the inverse case frequency. The symbol "$|C|$" is the total number of cases in the case library and $|\{t \in c\}|$ is the number of the cases containing the term *t* i.e. case frequency.

5. Summary and discussion

This book chapter presents some methods on AI, ML and MR those can be applied in health informatics domain to develop DSS. Here, fuzzy logic can be used to handle uncertainty issues in decision-making tasks, an example of a Parkinson disease application domain with fuzzy logic with rules is presented in the case study chapter. Some well-known machine learning methods both considering supervised and unsupervised are also presented, for example, kNN, SVM, ANN and RF are presented as supervised machine learning algorithms. For unsupervised learning, several clustering approaches are presented, such as K-means, Fuzzy C-means, Gaussian mixer model and Hierarchical. Also, the definition of parametric and non-parametric clustering together with their comparison is presented. Three different approaches to machine reasoning is discussed, such as fuzzy RBR, CBR, and textual case retrieval. Applications of such ML and MR in health informatics domains are presented in the use-case study chapter. It has been observed not a single method is enough to develop a DSS and most common is to combine more than one method as in a hybrid manner. Due to the multifaceted and complex nature of the health informatics domains, it is necessary to design such hybrid systems that can handle several challenges associated with multi-modal machine learning.

Reference

1. Aamodt A, Plaza E (1994) Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Commun 7 (1):39-59

2. Begum S, Ahmed MU, Funk P, Xiong N, Von Schéele B (2009) A Case-based Decision Support System for Individual Stress Diagnosis Using FuzzyY Similarity Matching. Computational Intelligence 25 (3):180-195. doi:10.1111/j.1467-8640.2009.00337.x

3. Corchado JM, Bajo J, Abraham A (2008) GerAmi: Improving Healthcare Delivery in Geriatric Residences. IEEE Intelligent Systems 23 (2):19-25. doi:10.1109/MIS.2008.27

4. Marling C, Whitehouse P Case-Based Reasoning in the Care of Alzheimer's Disease Patients. In, Berlin, Heidelberg, 2001. Case-Based Reasoning Research and Development. Springer Berlin Heidelberg, pp 702-715

5. Marling C, Shubrook J, Schwartz F Case-Based Decision Support for Patients with Type 1 Diabetes on Insulin Pump Therapy. In, Berlin, Heidelberg, 2008. Advances in Case-Based Reasoning. Springer Berlin Heidelberg, pp 325-339

6. Marling C, Wiley M, Cooper T, Bunescu R, Shubrook J, Schwartz F The 4 Diabetes Support System: A Case Study in CBR Research and Development. In, Berlin, Heidelberg, 2011. Case-Based Reasoning Research and Development. Springer Berlin Heidelberg, pp 137-150

7. Montani S (2008) Exploring new roles for case-based reasoning in heterogeneous AI systems for medical decision support. Applied Intelligence 28 (3):275-285. doi:10.1007/s10489-007-0046-2

8. Nilsson M, Funk P, Olsson EMG, von Schéele B, Xiong N (2006) Clinical decision-support for diagnosing stress-related disorders by applying psychophysiological medical knowledge to an instance-based learning system. Artificial Intelligence in Medicine 36 (2):159-176. doi:https://doi.org/10.1016/j.artmed.2005.04.004

9. Baltrušaitis T, Ahuja C, Morency L-P (2018) Multimodal Machine Learning: A Survey and Taxonomy. IEEE Transactions on Pattern Analysis and Machine Intelligence

10. Samuel AL (1959) Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development 3 (3):210-229. doi:10.1147/rd.33.0210

11. Salmani D, Akbari M (2008) Fuzzy and research paradigms relationship: a mutual contribution. Academic Leadership: The Online Journal 6 (2):7

12. Lukasiewicz J (1970) Logical foundations of probability theory. Jan Lukasiewicz, Selected Works:16-63

13. Łukasiewicz J (1953) A System of Modal Logic. Proceedings of the XIth International Congress of Philosophy 14:82-87

14. Zadeh LA (1965) Fuzzy sets. Information and Control 8 (3):338-353. doi:https://doi.org/10.1016/S0019-9958(65)90241-X

15. Marks-II R, Zadeh L (1994) Fuzzy logic technology and applications. IEEE Technological Activities Board

16. Jang J-SR, Sun C-T (1997) Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. Prentice-Hall, Inc.,

17. Robert C (2014) Machine learning, a probabilistic perspective. Taylor & Francis,

18. Friedman J, Hastie T, Tibshirani R (2001) The elements of statistical learning, vol 1. vol 10. Springer series in statistics New York,

19. Vapnik VN (1992) Principles of Risk Minimization for Learning Theory. Advances in Neural Information Processing Systems 4:831-838. doi:citeulike-article-id:431737

20. Jain AK, Duin RPW, Jianchang M (2000) Statistical pattern recognition: a review. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22 (1):4-37. doi:10.1109/34.824819

21. Basheer IA, Hajmeer M (2000) Artificial neural networks: fundamentals, computing, design, and application. Journal of Microbiological Methods 43 (1):3-31. doi:http://dx.doi.org/10.1016/S0167-7012(00)00201-3

22. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene Selection for Cancer Classification using Support Vector Machines. Machine Learning 46 (1):389-422. doi:10.1023/a:1012487302797

23. Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B (1998) Support vector machines. IEEE Intelligent Systems and their Applications 13 (4):18-28. doi:10.1109/5254.708428

24. Breiman L (2001) Random forests. Machine learning 45 (1):5-32

25. Breiman L (1996) Bagging Predictors. Machine Learning 24 (2):123-140. doi:10.1023/a:1018054314350

26. Drew PJ, Monson JRT (2000) Artificial neural networks. Surgery 127 (1):3-11. doi:http://dx.doi.org/10.1067/msy.2000.102173

27. Zhang GP (2000) Neural networks for classification: a survey. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 30 (4):451-462. doi:10.1109/5326.897072

28. Negnevitsky M (2001) Artificial Intelligence: A Guide to Intelligent Systems. Addison-Wesley Longman Publishing Co., Inc.,

29. Larose DT (2005) k-Nearest Neighbor Algorithm. In: Discovering Knowledge in Data. John Wiley & Sons, Inc., pp 90-106. doi:10.1002/0471687545.ch5

30. Witten IH, Frank E, Hall MA, Pal CJ (2016) Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann,

31. Graepel T (1998) Statistical physics of clustering algorithms. Technical Re-port 171822

32. Fung G (2001) A comprehensive overview of basic clustering algorithms.

33. Guldemır H, Sengur A (2006) Comparison of clustering algorithms for analog modulation classification. Expert Systems with Applications 30 (4):642-649

34. Chuai-Aree S, Lursinsap C, Sophasathit P, Siripant S (2001) Fuzzy c-mean: A statistical feature classification of text and image segmentation method. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9 (06):661-671
35. Velmurugan T, Santhanam T (2010) Performance evaluation of k-means and fuzzy c-means clustering algorithms for statistical distributions of input data points. European Journal of Scientific Research 46 (3):320-330
36. Chen G, Jaradat SA, Banerjee N, Tanaka TS, Ko MS, Zhang MQ (2002) Evaluation and comparison of clustering algorithms in analyzing ES cell gene expression data. Statistica Sinica:241-262
37. Wos L, Overbeck R, Lusk E, Boyle J (1984) Automated Reasoning: Introduction and Applications. Prentice Hall Professional Technical Reference,
38. Bottou L (2014) From machine learning to machine reasoning. Machine learning 94 (2):133-149
39. Kaplan J (2016) Artificial Intelligence: What everyone needs to know. Oxford University Press,
40. Zadeh LA (1973) Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. IEEE Transactions on Systems, Man, and Cybernetics SMC-3 (1):28-44. doi:10.1109/TSMC.1973.5408575
41. Riesbeck CK, Schank RC (1989) Inside Case-Based Reasoning. L. Erlbaum Associates Inc.,
42. Lopez De Mantaras R, Mcsherry D, Bridge D, Leake D, Smyth B, Craw S, Faltings B, Maher ML, Cox MT, Forbus K, Keane M, Aamodt A, Watson I (2005) Retrieval, reuse, revision and retention in case-based reasoning. The Knowledge Engineering Review 20 (03):215-240. doi:doi:10.1017/S0269888906000646
43. Watson I, Perera S (1997) Case-based design: A review and analysis of building design applications. AI EDAM 11 (01):59-87. doi:doi:10.1017/S0890060400001840
44. Kolodner JL (1983) Reconstructive memory: A computer model. Cognitive Science 7 (4):281-328. doi:https://doi.org/10.1016/S0364-0213(83)80002-0
45. Kolodner J (2014) Case-based reasoning. Morgan Kaufmann,
46. Koton PA (1988) Using experience in learning and problem solving. Massachusetts Institute of Technology,
47. Simpson Jr RL (1985) A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation. AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH,
48. Bareiss R (2014) Exemplar-based knowledge acquisition: A unified approach to concept representation, classification, and learning, vol 2. Academic Press,
49. Bareiss E (1989) Protos: A unified approach to concept representation, classification, and learning.
50. Lindgaard G Intelligent decision support in medicine: back to bayes? In: Proceedings of the 14th European conference on Cognitive ergonomics: invent! explore!, 2007. ACM, pp 7-8
51. Bergmann R, Kolodner J, Plaza E (2005) Representation in case-based reasoning. Knowl Eng Rev 20 (3):209-213. doi:10.1017/s0269888906000555
52. Watson I (1998) Applying case-based reasoning: techniques for enterprise systems. Morgan Kaufmann Publishers Inc.,
53. Ontañón S, Plaza E (2003) Collaborative case retention strategies for CBR agents. Paper presented at the Proceedings of the 5th international conference on Case-based reasoning: Research and Development, Trondheim, Norway,
54. Begum S, Ahmed MU, Funk P, Ning X, Folke M (2011) Case-Based Reasoning Systems in the Health Sciences: A Survey of Recent Trends and Developments. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 41 (4):421-434. doi:10.1109/TSMCC.2010.2071862
55. Singhal A (2001) Modern information retrieval: A brief overview. IEEE Data Eng Bull 24 (4):35-43
56. Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Commun ACM 18 (11):613-620. doi:10.1145/361219.361220
57. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Information Processing & Management 24 (5):513-523. doi:https://doi.org/10.1016/0306-4573(88)90021-0
58. Weber RO, Ashley KD, BrÜNinghaus S (2005) Textual case-based reasoning. The Knowledge Engineering Review 20 (3):255-260. doi:10.1017/S0269888906000713