Minimal algorithms for knowledge representation in clinical decision support systems research: a theoretical and empirical analysis

Anderson A. Eduardo*[1], Rafael M. Loureiro[2], Adriano Tachibana[2], Pedro V. Netto[2], Tatiana F. de Almeida[3], Luiz Henrique Alves Monteiro[4], and André P. dos Santos[1]

[1]TMDA, Hospital Israelita Albert Einstein
[2]Departamento de Imagem, Hospital Israelita Albert Einstein
[3]Relacionamento Médico, Hospital Israelita Albert Einstein
[4]Engineering School, Universidade Presbiteriana Mackenzie

April 20, 2022

Abstract

Clinical decision support systems (CDSS) figures out as one of the most promising technologies for data-centered and AI-prompted healthcare. Its current developments are mainly guided by two disparate mindsets, namely a machine learning- centered framework and a classical rule-based framework. These respective approaches presents contrastive pros and cons. In the present study we provide an analysis showing that these two mindsets are actually related to each other, and straightforward algorithms are feasible by combining current standards for machine learning and classic decision tables algorithms. A theoretical analysis are provided, as well a computational implementation (in python). A real case scenario on radiological immaging exam prescription is used to ilustrate the successfully application of our results. Future work on benchmarking the proposed algorithms embodied in a fully operational clinical decision support system could extend our findings towards daily used systems.

Keywords— Artificial intelligence; machine learning; decision table; digital health; health informatics.

1 Introduction

In the last decade, modern digital technology become pervasive in healthcare and medical systems. In terms of intellectual historical evolution, the ongoing disruption

*Corresponding author: act082@einstein.br

by the emergence of the so called fourth scientific paradigm reverbered in advancing health sciences, existing a long and apparent road of technological advances ahead [6, 8].

In this context, clinical decision support systems (CDSS) have transvased academic circles to daily life, with large operational systems affecting thousand of patients and professionals per day [7]. Today, CDSS is perceived by many authors as one of the most promising concept among emergent technologies in healthcare [5, 2, 7, 1].

In practical terms, CDSS are implemented as a software, generally integrated as a service in previously existing local systems [7]. The software is kept runing under the rotinetely use of a local system. But, internally, prescription of medical exams are processed by the CDSS and some type of signalization is shown to the user, sugesting the best options given patient information [7].

There are different and specific modes of implementation and operationalization of CDSS [4, 7]. Despite of that, they are built upon a same system architecture, comprising [7]: (i) an interface layer, where the user inputs patient data, receive prescription suggestions and queries are parsed; and (ii) the knowledge base, being an algorithm capable of modeling expert knowledge regarding a specific problem or intellectual domain.

The literature on the knowledge base component of CDSS discusses a variety of algorithms, from explicit IF-ELSE rules to deep learning. Naturally, every approach to knowledge modeling will have its pros and cons. But, in our interpretation, research advances have largelly been internally biased for each one of the respective lines of approach, with loosely interaction for theoretical synthesis and integrative developments. Interpretability of knowledge representation is a relevant example, with, on one hand, classic IF-ELSE approaches having clearer representation of internal extructure (directaly affecting the interpretability of systems suggestions), but at a high cost of implementation and maintainance, and, on the other hand, machine learning approaches, having low (or very low) internal interpretability, but with a lesser cost of implementation and maintainence and, importantly, with inherent uncertainity in outputs [7].

Thus, in present study we provide an analysis showing that straightforward algorithms are feasible, combining current standards in massive dataset formation for machine learning and classic decision tables algorithms. We show that there are place for mathematical theory bridging classic and modern approaches for knowledge representation in CDSS and we ilustrate it providing "white-box", graph-based minimal algorithms, with experimental results for real radiological imaging data.

2 The decision table formalism

Strictly speacking, a decision table is a table representing the exhaustive set of mutual exclusive conditional expressions, within a predefined problem area (see [9]).

There are two fundamental sets for a decision table, $\mathcal{C}$ and $\mathcal{A}$, namely the *condition set* and the *action set* [9]. The condition set is defined as follows (definition (2.1)).

Definition 2.1. The Condition set, $\mathcal{C}$, consists of n elements c_i, $1 \leq i \leq n$, which are, in turn, formed by the tuple $(c^{\langle S \rangle}, C^{\langle T \rangle})_i$, with $c_i^{\langle S \rangle} \in \mathcal{C}^{\langle S \rangle}$ and $C_i^{\langle T \rangle} \in \mathcal{C}^{\langle T \rangle}$, such that

$$\mathcal{C} = \{c_i \mid 1 \leq i \leq n\} = \{(c^{\langle S \rangle}, C^{\langle T \rangle})_i \mid 1 \leq i \leq n\},$$

where $\mathcal{C}^{\langle S\rangle}$ are the condition subjects, with each i^{th} subject having a domain $C_i^{\langle D\rangle} \in \mathcal{C}^{\langle D\rangle}$; $C_i^{\langle T\rangle} \in \mathcal{C}^{\langle T\rangle}$ is the set of possible logical states for the i^{th} subject, being stated in terms of $C_i^{\langle D\rangle} \in \mathcal{C}^{\langle D\rangle}$ and mapping to a one and only one item of the cartesian product $\mathcal{A}^{\langle V\rangle} \times \mathcal{A}^{\langle V\rangle}$ (which is definied bellow).

For a set $C_i^{\langle T\rangle}$, its condition states follows the definition below (definition (2.2)).

Definition 2.2. Each set of condition state, $C_{(i)_k}^{\langle T\rangle} \in C_i^{\langle T\rangle}$, defines a subset $C_{(i)_k}^{\langle T\rangle *}$, such that $C_{(i)_k}^{\langle T\rangle *} \subset C_i^{\langle D\rangle}$ and

$$\bigcup_{k=1}^{n_k} \left(C_{(i)_k}^{\langle T\rangle *}\right) = C_i^{\langle D\rangle}$$

$$\bigcap_{k=1}^{n_k} \left(C_{(i)_k}^{\langle T\rangle *}\right) = \emptyset$$

Regarding the action set, it is defined as follows (definition (2.3)).

Definition 2.3. Similarlly to the Condition set, the Action set, $\mathcal{A}$, is formed of m actions a_j, $1 \leq j \leq m$, being $a_j = (a^{\langle S\rangle}, A^{\langle V\rangle})_j$, with $A_j^{\langle V\rangle} \in \mathcal{A}^{\langle V\rangle}$ the set of possible values for the j^{th} action of $\mathcal{A}$, and $a_j^{\langle S\rangle} \in \mathcal{A}^{\langle S\rangle}$ the j^{th} action item, such that

$$\mathcal{A} = \{a_j \mid 1 \leq j \leq m\} = \{(a^{\langle S\rangle} \in \mathcal{A}^{\langle S\rangle}, a^{\langle V\rangle} \in \mathcal{A}^{\langle V\rangle})_j \mid 1 \leq j \leq m\}.$$

In order to illustrate such concepts in the context of the present paper, we provide the following example.

Example. *Consider a decision table in a simplifyied context of medical immage prescription. The Condition set, in terms of* $\mathcal{C}^{\langle S\rangle} = \{age, clinical\ indication\}$, $\mathcal{C}^{\langle D\rangle} = \{\{0, 1, 2, ...\}, \{Indication\ 1, Indication\ 2\}\}$, $\mathcal{C}^{\langle T\rangle} = \{\{age \leq 17 \rightarrow children, age > 17 \wedge age \leq 50 \rightarrow adult, age > 50 \rightarrow senior\}, \{Indication\ 1, Indication\ 2\}\}$, *and* $\mathcal{C}^{\langle T\rangle *} = \{\{\{1, 2, ..., 17\}, \{18, 19, ..., 50\}, \{51, 52, ...\}\}, \{\{Indication\ 1\}, \{Indication\ 2\}\}\}$ *is defined as*

$$\mathcal{C} = \{(age, \{children, adult, senior\}), (clinical\ indication, \{Indication\ 1, Indication\ 2\})\}.$$

The Action set, in terms of $\mathcal{A}^{\langle S\rangle} = \{clinical\ exam\}$ *and* $\mathcal{A}^{\langle V\rangle} = \{exam\ 1, exam\ 2, exam\ 3\}$ *is defined as*

$$\mathcal{A} = \{(clinical\ exam, \{exam\ 1, exam\ 2, exam\ 3\})\}.$$

Given a decision table for which these definitions completely holds, we got a decision table function, as stated by the following theorem (theorem (2.1)).

Theorem 2.1. *(Decision table function) Considering the cartesian products* $\mathcal{C}^{\langle R\rangle} = \{(c_1^{\langle T\rangle}, c_2^{\langle T\rangle}, \ldots, c_k^{\langle T\rangle}) \mid c_1^{\langle T\rangle} \in C_1^{\langle T\rangle}, c_2^{\langle T\rangle} \in C_2^{\langle T\rangle}, \ldots, c_k^{\langle T\rangle} \in C_k^{\langle T\rangle}\}$ *and* $\mathcal{A}^{\langle R\rangle} = \{(a_1^{\langle V\rangle}, a_2^{\langle V\rangle}, \ldots, a_k^{\langle V\rangle}) \mid a_1^{\langle V\rangle} \in A_1^{\langle V\rangle}, a_2^{\langle V\rangle} \in A_2^{\langle V\rangle}, ..., a_l^{\langle V\rangle} \in A_l^{\langle T\rangle}\}$, *there is such a function*

$$F_{DT} : \mathcal{C}^{\langle R\rangle} \rightarrow \mathcal{A}^{\langle R\rangle},$$

mapping each action in $\mathcal{A}^{\langle R\rangle}$ *from one or more conditions in* $\mathcal{C}^{\langle R\rangle}$.

Proof. Let $\mathcal{X} = \{x \mid x \in \mathcal{C}^{\langle R \rangle}\}$ and $\mathcal{Y} = \{y \mid y \in \mathcal{A}^{\langle R \rangle}\}$. Consider a relation $R \subset \{(x, y) \mid x \in \mathcal{X} \wedge y \in \mathcal{Y}\}$. As $\mathcal{C}^{\langle R \rangle}$ is defined in terms of $\mathcal{C}^{\langle T \rangle}$, the definitions (2.1) and (2.2) holds for any element or subset of $\mathcal{C}^{\langle R \rangle}$. Thus

$$\forall x_a, x_b \in \mathcal{X}, y_a, y_b \in \mathcal{Y} : x_a = x_b \rightarrow R(x_a) = R(x_b) \implies y_a = y_b,$$

meaning that R is a function with domain $\mathcal{X}$ and image in $\mathcal{Y}$. Consequently, $\mathcal{X} \subset \mathcal{C}^{\langle R \rangle}$ and $\mathcal{Y} \subset \mathcal{A}^{\langle R \rangle}$ implies that F_{DT} *is a function.* □

Traditionally, a decision table is represented in a (row-driven) augmented matrix, as follows (equations (1), (2) and (3)):

$$\mathbf{M}_{(C)} = \begin{pmatrix} c_{(1,1)} & c_{(1,2)} & \cdots & c_{(1,o)} \\ \vdots & \vdots & \ddots & \vdots \\ c_{(n,1)} & c_{(n,2)} & \cdots & c_{(n,o)} \end{pmatrix} \tag{1}$$

$$\mathbf{M}_{(A)} = \begin{pmatrix} a_{(1,1)} & a_{(1,2)} & \cdots & a_{(1,o)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{(m,1)} & a_{(m,2)} & \cdots & a_{(m,o)} \end{pmatrix} \tag{2}$$

$$\mathbf{M}_{(DT)} = (\mathbf{M}_{(C)}|\mathbf{M}_{(A)}) = \left(\begin{array}{cccc} c_{(1,1)} & c_{(1,2)} & \cdots & c_{(1,o)} \\ \vdots & \vdots & \ddots & \vdots \\ c_{(n,1)} & c_{(n,2)} & \cdots & c_{(n,o)} \\ \hline a_{(1,1)} & a_{(1,2)} & \cdots & a_{(1,o)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{(m,1)} & a_{(m,2)} & \cdots & a_{(m,o)} \end{array}\right) \tag{3}$$

Here, $c_{(i,l)} \in \mathcal{C}^{\langle T \rangle}$, with $i \in \{1, 2, ..., n\}$ and $l \in \{1, 2, ..., o\}$ refer to condition state values for the i^{th} condition subject and l^{th} rule. The values $a_{(j,l)} \in \mathcal{A}^{\langle V \rangle}$ with $j \in \{1, 2, ..., m\}$ and $l \in \{1, 2, ..., o\}$, refer to actions for the j^{th} action subject and l^{th} rule.

3 Linking decision tables and statistical machine learning datasets

Despite of both decision tables and machine learning be cirscunsribed whithin the wide field of artificial intelligence, these subjects has been, conceptually and temporally, relatively far from each other. The decision table formalism were developed in the age of symbolic approaches to artificial intelligence modelling. Machine learning, on the other hand, emerged as a broad theoretical mindset, which became in vogue in the recent decades, becaming the dominant paradigm for the current generation of artificial intelligence researchers and practioners. Despite of such historical bias, mathematical formalisms in the machine learning field and the decision table formalism are not disparate from each other, having place for theoretical synthesis and synergistic developments.

By the current formalism of statistical learning theory for supervised learning, a machine learning problem can be definied as follows (definition 3.1) (see [3]).

Definition 3.1. A machine learning problem consists of finding a function capable of mapping a random variable $\boldsymbol{Y} \in \mathbb{R}$ from a vector of $\boldsymbol{x} \in \boldsymbol{X}$, with $\boldsymbol{X} \subset \mathbb{R}^n$. Such a function, defined here as $h(x) : \boldsymbol{X} \rightarrow \boldsymbol{Y}$, belongs to a hipothetical space of functions, $\mathcal{H}$, and can be computed by the means of minimizing a loss function $L(h(.), S)$ with respect to a parameterized hypothesis, h, and considering S, a collection of observed instances of $\boldsymbol{X}$ and $\boldsymbol{Y}$.

By definition, and as stated by the definition (3.1), a machine learning problem explicitly depends on a collection of empirical observations. Thus, we state a more specific definition for S, as follows (definition 3.2).

Definition 3.2. Let Z be the catesian product $\boldsymbol{X} \times \boldsymbol{Y}$. A collection of empirical observations, S, is a statistical sample over the unknown probability distribution of Z, being constituted by tuples $(\boldsymbol{x}^{\langle observed \rangle}, y^{\langle observed \rangle})_i$, $1 \leq i \leq o$, with the vector $\boldsymbol{x}_i^{\langle observed \rangle} \in \boldsymbol{X}^{\langle observed \rangle}$, $\boldsymbol{X}^{\langle observed \rangle} = \{(x_{(1,1)}, x_{(1,1)}, \ldots, x_{(1,n)}), \ldots, (x_{(o,1)}, x_{(o,1)}, \ldots, x_{(o,n)}) \mid x_{(.,1)} \in X_1 \wedge x_{(.,2)} \in X_2 \wedge \cdots \wedge x_{(.,n)} \in X_n\}$, observing that $\forall X_i \subset \boldsymbol{X} : \cup_i(X_i) = \boldsymbol{X}^{\langle observed \rangle}$; and with $y_i^{\langle observed \rangle} \in \boldsymbol{Y}^{\langle observed \rangle}$. S is parsed by a learning algorithm which implements $L(h(.), S)$ and is capable of returning $h(.)_S$, which refers to a parameterized function regarding a particular sample dataset, S. Thus, the parameterization found by the algorithm is fundamentally determined by the sample dataset.

In terms of tabular datasets, $\boldsymbol{X}^{\langle observed \rangle}$ and $\boldsymbol{Y}^{\langle observed \rangle}$ can be represented in a matricial form, as shown below (equations (4) and (5)).

$$\mathbf{M}_{(X_{observed})} = \begin{pmatrix} x_{(1,1)} & x_{(1,2)} & \ldots & x_{(1,n)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(o,1)} & x_{(o,2)} & \ldots & x_{(o,n)} \end{pmatrix} \tag{4}$$

$$\mathbf{M}_{(Y_{observed})} = \begin{pmatrix} y_1 & y_2 & \ldots & y_o \end{pmatrix}^T \tag{5}$$

Thus, S can be represented by the augmented matrix given below (equation (6)).

$$\mathbf{M}_{(S)} = \left(\mathbf{M}_{(X_{observed})} | \mathbf{M}_{(Y_{observed})}\right) = \left(\begin{array}{cccc|c} x_{(1,1)} & x_{(1,2)} & \ldots & x_{(1,n)} & y_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{(o,1)} & x_{(o,2)} & \ldots & x_{(o,n)} & y_o \end{array}\right) \tag{6}$$

The matricial representation of decision tables is structuturally equivalent to such matricial representation of S, $\mathbf{M}_S$, as stated by the theorem (3.1).

Theorem 3.1. *(The equivalency theorem for $\mathbf{M}_{(DT)}$ and $\mathbf{M}_{(S)}$) A matricial representation of a decision table, $\mathbf{M}_{(DT)}$, given by equation (1), is equivalent to a matricial representation of a statistical machine learning dataset, $\mathbf{M}_{(S)}$, given by equation (4), if $\mathbf{M}_{(DT)}$ is transposed, such as*

$$\mathbf{M}_{(DT)}^T \equiv \mathbf{M}_{(S)}$$

Proof. Considering definitions (2.1), (2.2) and (2.3), each one of the n subjects in $\mathcal{C}^{\langle S \rangle} = \{c_1^{\langle S \rangle}, c_2^{\langle S \rangle}, \ldots, c_n^{\langle S \rangle}\}$ have a domain $\mathbf{dom}(c_i^{\langle S \rangle}) = C_i^{\langle D \rangle}, C_i^{\langle D \rangle} \subset \{C_1^{\langle D \rangle}, C_2^{\langle D \rangle}, \ldots, C_n^{\langle D \rangle}\}$, being assigned in a row-driven fashion in the matrix $\mathbf{M}_C$, given by equations (1) and (3). Each column l in $\mathbf{M}_C$ can be conveniently represented as a n-size tuple $\boldsymbol{t}_l$ composed by state values v_i, $1 \leq i \leq n$, each one row-wise assigned to its respective

subjects, $c_i^{\langle S \rangle} \in \mathcal{C}^{\langle S \rangle}$. Clearly, the o-columns in $\mathbf{M}_C$ is relative to the number of decision rules embeded in a decision table and, in turn, informing the number of such tuples, $\boldsymbol{t}_l$. Now, consider that each row in $\mathbf{M}_S$ groups the values of the n features (or traits, or variables, or dimensions) for each empirical observation, $\boldsymbol{x}_i^{\langle observed \rangle}$. Necessarily, each feature, X_i, have a proper domain, $\mathbf{dom}(X_i) \subset \mathbb{R}$, and each vector $\boldsymbol{x}_i^{\langle observed \rangle}$, being an empirical observation, by definition, can also be represented as a tuple $\boldsymbol{t}_l$ of observed state values v_i, column-wise assigned to its respective feature X_i, and showing that condition subjects are equivalent of features, or

$$\forall \boldsymbol{X}^{\langle observed \rangle} \subset \boldsymbol{X} : \boldsymbol{X}^{\langle observed \rangle} \subseteq \mathcal{C}^{\langle S \rangle} \implies \mathcal{C}^{\langle S \rangle} \equiv \boldsymbol{X}.$$

Thus, any hypothetical subject, C_a, with its own domain $\mathbf{dom}(C_a)$, can have its state value imputed in a tuple, t_C, comprised of n state values, along other $n-1$ state values for other subjects, such as $C_b, C_c, C_d, \ldots, C_z$. An arbitrary number of such tuples can be stacked either in a row-oriented fashion, forming a matrix $\mathbf{M}_1$, or in a column-oriented fashion, forming a matrix $\mathbf{M}_2$, without any loss of generality. It can be easilly seen that

$$\mathbf{M}_{(1)}{}^T - \mathbf{M}_{(2)} = 0$$
$$\mathbf{M}_{(1)}{}^T = \mathbf{M}_{(2)}$$

Naturally,

$$\mathbf{M}_{(1)} \equiv \mathbf{M}_{(C)} \wedge \mathbf{M}_{(2)} \equiv \mathbf{M}_{(X_{observed})} \implies \mathbf{M}_{(C)}^T \equiv \mathbf{M}_{(X_{observed})}$$

The application of this same rationale for equations (2) and (5) is trivial, proving the theorem (3.1). □

Thus, any dataset as typically arranged in current machine learning practioning is ready to be used for a decision table implementation, as long as it is in accordance with definitions (2.1), (2.2 and (2.3)).

An interesting consequence of theorem (3.1) is stated by the following corollary.

Corollary 3.1. *Given a hypothesis $h \in \mathcal{H}$, any machine learning algorithm, in terms of the definitions (3.1) and (3.2), can learn and furnish a computational representation of any particular decision table function, as stated at theorem (2.1), provided that the loss function tends to zero. This is expressed by*

$$h_S := \arg\min_{h \in \mathcal{H}} L(h(.), S) \wedge L(h(.), S) \approx 0 \implies h_S \equiv F_{DT}$$

Proof. Let a dataset D be definied in terms of the equations (4), (5) and (6). Let a function $g_{Error} : \mathbb{R}^n \to \mathbb{R}$ be an error function, i.e., a function able to compute an arbitrary degree of correctedness for the output of functions $f_i \in \mathcal{F}$, being $\mathcal{F}$ a set of particular functions. Now, consider a function $f_{DT} \in \mathcal{F}$, which is specified in terms of the definitions (2.1), (2.2) and (2.3). Also, consider a function $f_{ML} \in \mathcal{F}$, specified in terms of definitions (3.1) and (3.2). By the definitions, we have $g_{Error}(f_{DT}, D) = 0$, i.e., when applying f_{DT} to the dataset D. Also by the definitions, we have that, for any situations in which $L(f_{ML}, D) \approx 0$ is achievable, we have

$$g_{Error}(f_{ML}, D) = 0 = g_{Error}(f_{DT}, D).$$

In this cases,

$$f_{DT} \models D \implies f_{ML} \models D$$

This result makes f_{DT} and f_{ML} equivalent in the representation of D, or

$$f_{ML} \equiv f_{DT}.$$

As f_{DT} is defined is the same terms of F_{DT} and f_{ML} is defined in the same terms of $h \in \mathcal{H}$, we got

$$\forall h \in \mathcal{H} : f_{ML} \equiv f_{DT} \Leftrightarrow F_{DT} \equiv h.$$

□

Corollary (3.1) pave the way for applying current machine learning algorithms in computational representation of decision tables in a straightforward fashion. This shows that, as we argued before, these are not rival formalisms, existing potential for theoretical and practical synthesis. Despite of corollary (3.1), we would like to remark that representing a decision table in terms of $h \in \mathcal{H}$ (see definition (3.1)) impels loss of interpretability of internal decision structure, being contingent to algorithm features.

4 Flexible decision tables

In the previous section we have shown that decision tables have a good fit with machine learning datasets, making them adherent to the canonical structure for modern datasets, widely used by machine learning practioners. Such result may help to bring it closer to the most palatable algorithms routinely used for artificial intelligence modelling.

However, as stated in the furnished definitions, only one output decision should be expected for any inputted query in a decision table model. Unfortunatelly, it can be too restrictive for many pratical applications. As an example, this is the case for prescription of radiology image exams. Profissionals in this field commonly ends up with more than only one well appropriate radiological immaging options for a same patient condition. So, in order to make simpler algorithms really usefull and adequate in suporting such cases, we should flexibilize decision table formalism.

Here we propose that a more flexible or generalizable background for decision tables can be obtained through a minor changing in the basal definitions at decision table formalism, as stated in the following definition (definition 4.1).

Definition 4.1. A decision table can be defined as a ordered triple $(\mathcal{C}^{\langle R \rangle}, \mathcal{A}^{\langle R \rangle}, G(V, E))$, with $\mathcal{C}^{\langle R \rangle} = \mathcal{C}^{\langle T \rangle} \times \mathcal{C}^{\langle T \rangle}$, $\mathcal{A}^{\langle R \rangle} = \mathcal{A}^{\langle V \rangle} \times \mathcal{A}^{\langle V \rangle}$ and $G(V, E)$ being a directed graph, with vertices $V \subseteq \mathcal{C}^{\langle R \rangle} \cup \mathcal{A}^{\langle R \rangle}$ and edges $E \subseteq \mathcal{C}^{\langle R \rangle} \times \mathcal{A}^{\langle R \rangle}$, mapping from $\mathcal{C}^{\langle R \rangle}$ to $\mathcal{A}^{\langle R \rangle}$. In such terms, a decision table consists in the binary relation, R_{DT}, over sets $\mathcal{C}^{\langle R \rangle}$ and $\mathcal{A}^{\langle R \rangle}$, being defined as

$$R_{DT} = G \subseteq \{(c, a) \mid c \in \mathcal{C}^{\langle R \rangle} \wedge a \in \mathcal{A}^{\langle R \rangle}\}$$

Theorem 4.1. *(Flexible decision table) Given definition (4.1), F_{DT} is a special case of R_{DT}.*

Proof. The proof naturally emerge from the fact that F_{DT} is a function and R_{DT} is a mathematical relation. So,

$$\forall F_{DT} \forall R_{DT} P_1(F_{DT}) \wedge P_2(R_{DT}) \implies P_3(F_{DT}, R_{DT}),$$

being P_1, P_2 and P_3 the predicate symbol for *is a mathematical function*, *is a mathematical relation* and *is a special case*, respectivelly. □

With the definition (4.1), we are generalizing definition (2.1) and, consequently, we provide that F_{DT} (see theorem (2.1)) naturally be a particular case of R_{DT} (definition (4.1) and theorem (4.1)). This is a subtle change from the classic definition of decision tables, but with consequences making them more interesting for modern practical use, such as decision support systems for radiology exam prescription.

Also, it is important to note that, in the terms of definition (4.1) and theorem (4.1), any results obtained considering decision tables strictly as a mathematical function do not apply to R_{DT}. Here, the corollary (3.1) is the most relevant example.

Fortunately, in modeling a decision table as a mathematical relation and representing it as a graph object, we circunscribe our focal problem in the sound grounds of graph theory. This is very convenient, because algorithms and computational tools specific for such abstract objects are well developed and "white box" models are feasible. In the next sextion, we explore the simpler algorithms for computing R_{DT}.

5 Algorithms

In this section, we provide algorithms for the decision table graphs, in accordance to definition (4.1). It is assumed that the input dataset is structured in accordance to definitions (3.1), (3.2) and (4.1), as well as the theorem (3.1).

Given the nature of the dataset for a flexible decision table, it is very convenient to entirely load each vector of condition states and action values (i.e., the rows of $\mathbf{M}_{(S)}$), representing it as a single computational object (e.g., a n-tuple). By our definition of flexible decision table, such vectors constitute the paths of a graph G for the relation R_{DT}, thus the proper paths of the decision table.

The algorithm (1) shows that paths can be readly obtained from data arranged in terms of equation (6), providing a straightforward computational representation of G and, by definition, the relation R_{DT} for a given decision table.

Algorithm 1 The *Load data as graph paths* algorithm

$p \leftarrow$ *inputted string for CSV file path*
$q :=$ *string code for line break*
$c :=$ *string code for columns split*
$\mathcal{G} := \emptyset$
$b \leftarrow$ *pointer for* p
$l_{raw} \leftarrow$ *read(b)*
$l_{rows} \leftarrow$ *split*(l_{raw}, q)
for each $r \in \mathcal{S}, \forall \mathcal{S} = \{r \mid r \in l_{rows}\}$ **do**
    $t \leftarrow$ *split*(r, c)
    $\mathcal{G} \leftarrow \mathcal{G} \cup \{t\}$
end for
return $\mathcal{G}$

In the context of algorithm (1), *read* is a method for obtaining data in computer physical memmory through a pointer object, and *split* is a method to split a string at specific positions (given by c). These methods are commonly provided on most of currently used programming languages for artificial intelligence implementation (e.g., python).

Following, given a graph G, represented according to algorithm (1), querying for actions, $a^{\langle V \rangle} \in \mathcal{A}^{\langle V \rangle}$, given a particular set of condition states, $C_i^{\langle T \rangle} \in \mathcal{C}^{\langle T \rangle}$, should be possible. In algorithm (2), we show how a set of paths to actions, constrained to inputted condition states, can be efficiently maped.

Algorithm 2 The *Find paths* algorithm

$\mathcal{X} \leftarrow$ *set of items inputted by user*
$\mathcal{O} := \emptyset$
$\mathcal{G} :=$ *graph paths, or* R_{DT}
for each $p \in \mathcal{G}$ **do**
  if $\mathcal{X} \subseteq \mathcal{P}, \forall \mathcal{P} = \{p_i \mid p_i \in p\}$ **then**
    $\mathcal{O} \leftarrow \mathcal{O} \cup \{\mathcal{P}\}$
  end if
end for
return $\mathcal{O}$

Finally, our representation of G, in order to achieve a straightforward representation of R_{DT}, can easilly be rearranged into the classical representation for graph objects. This is especially relevant for the treatment of decision tables within the scope of long standing computational tools available for mathematical operations on graphs. Thus, in algorithm (3) we show that an edge list is readily obtainable for G.

Algorithm 3 The *From paths to edges* algorithm

$\mathcal{E} := \emptyset$
$\mathcal{G} :=$ *graph paths, or* R_{DT}
for each $p \in \mathcal{G}$ **do**
  for each $i \in \mathcal{N}, \forall \mathcal{N} = \{i \mid 1 \leq i \leq |p| - 1\}$ **do**
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{(p_i, p_{i+1})\}$
  end for
end for
return $\mathcal{O}$

It is easy to see that the temporal complexity of both algorithm (1) and algorithm (2) is $O(n)$. For algorithm (3), temporal complexity is $O(n \times m)$, being $m = |\boldsymbol{x}_i^{\langle observed \rangle}|$.

6 Experiments

In order to subdue our theoretical arguments to the scrutiny of reality, in this section we carry out an experiment exploring an application for a clinical decision support system. Focusing on a medical radiological context, we compiled a decision support dataset for

immaging exams for head and neck situations (see **Supplementary Material**). Our data were obtained from *Appropriateness Criteria* of American College of radiology (ACR), available online at **https://acsearch.acr.org/list**.

The algorithms (1), (2) and (3) were implemented in pure **python** programming language (version 3.9). Complementarily, we used the package **networkx** (version 2.6.3) to explore the output of algorithm (3).

Bellow, python coding for the mentioned algorithms are shown.

```
    def load_graph_paths(file_path, drop_first_line=True, sep=','):
        p = file_path
        q = '\n'
        c = sep
        G = set()
        b = open(file_path)
        l_raw = b.read()
        b.close()
        l_rows = l_raw.split(q)

        for index, r in enumerate(l_rows):
            # allowing drop CSV headers
            if drop_first_line:
                if index == 0:
                    continue
            else:
                pass
            t = tuple(r.split(sep))
            # cleaning accidental extra spaces
            t = tuple([t_i.strip() for t_i in t])
            G.add(t)

        return(G)
```

Listing 1: Python implementation of algorithm (1).

```
    def find_paths(X, graph):
        # cleaning accidental extra spaces
        X = set([unidecode(i.strip().lower()) for i in X])
        O = set()
        G = graph

        for p in G:

            # cleaning accidental extra spaces
            p = tuple([unidecode(p_i.strip().lower())
                                for p_i in p])
            if X.issubset(p):
                O.add(p)

        return O
```

Listing 2: Python implementation of algorithm (2).

```
    def from_paths_to_edges(graph):
        E = set()
        G = graph

        for p in G:
```

```
            for i in range(len(p) - 1):
                E.add((p[i], p[i+1]))

        return E
```

Listing 3: Python implementation of algorithm (3).

The complete implementation of our experiments is provided in `Supplementary Material`. Bellow, we provide the main results.

Experiment 1: *obtaining a representation of G for the experimental dataset*

Using the algorithm (1), implemented as the method `load_graph_paths`, the obtained output is shown in listing (4).

```
{
    ('Celulite','Celulite orbitaria','AngioRM de cranio','Geralmente Inadequado'),
    ('Celulite','Celulite orbitaria','AngioTC de cranio','Geralmente Inadequado'),
    ('Celulite', 'Celulite orbitaria', 'RM de cranio', 'Pode ser adequado'),
    ('Celulite', 'Celulite orbitaria', 'RM de face', 'Adequado'),
    ('Celulite', 'Celulite orbitaria', 'RM de orbitas', 'Adequado'),

    ...

    ('Vertigem','Vertigem persistente (vertigem central)','TC de cranio','Pode ser adequado')
}
```

Listing 4: Output for Experiment 1.

Experiment 2: *maping paths to actions, given input constrains*

Now, with the implementation `find_paths` for algorithm (2), the obtained output for the query `find_paths(['Vertigem', 'Vertigem episodica(vertigem periferica)'])` is shown in listing (5).

```
{
    ('vertigem', 'vertigem episodica (vertigem periferica)', 'angiorm de cranio arterial ou venosa', 'geralmente inadequado'),
    ('vertigem', 'vertigem episodica (vertigem periferica)', 'angiotc de cranio arterial ou venosa', 'geralmente inadequado'),
    ('vertigem', 'vertigem episodica (vertigem periferica)', 'rm de cranio', 'geralmente inadequado'),
    ('vertigem', 'vertigem episodica (vertigem periferica)', 'rm de cranio e orelhas internas', 'adequado'),
    ('vertigem', 'vertigem episodica (vertigem periferica)', 'rm de orelhas internas', 'adequado'),
```

```
        ('vertigem', 'vertigem episodica (vertigem periferica)',
        'tc de cranio', 'geralmente inadequado'),
        ('vertigem', 'vertigem episodica (vertigem periferica)', '
    tc dos ossos temporais', 'geralmente inadequado')
    }
```

Listing 5: Output obtained for Experiment 2.

Experiment 3: *obtaining canonical representation of graphs*

With the implementation `from_paths_to_edges` for algorithm (3), the obtained output is depicted on listing (6). This output were succefully parsed by `networkx` methods, being a renderization of G provided in figure (6).

```
    {
        ('AngioRM de cranio', 'Geralmente Inadequado'),
        ('AngioRM de cranio arterial ou venosa', 'Geralmente
    Inadequado'),
        ('AngioRM de pescoco', 'Adequado'),
        ('AngioRM de pescoco', 'Geralmente Inadequado'),
        ('AngioTC de cranio', 'Geralmente Inadequado'),
        ('AngioTC de cranio arterial ou venosa', 'Geralmente
    Inadequado'),
        ('AngioTC de pescoco', 'Adequado'),
        ('AngioTC de pescoco', 'Geralmente Inadequado'),

        ...

        ('Vertigem persistente (vertigem central)', 'TC de cranio')
    }
```

Listing 6: Output obtained for Experiment 3.

7 Conclusion

In this study we have explored an integrative approach bridging aspects of classical decision tables and modern machine learning, yielding simpler and sound approach to knowledge modeling for clinical decision support systems. We provide straightfoward mathematical results and computational algorithms, enforcing that valuable theoretical and practical findings can be obtained by intersectioning well-maturated artificial intelligence research and current machine learning formalisms. Through a python implementation, a real case scenario is used to ilustrate the application of our results in a knowledge modeling problem for medical radiology immaging exam prescription based on guidelines data. Future work on benchmarking the proposed algorithms embodied in a fully operational clinical decision support system could extend our findings towards daily used systems.

References

[1] Cemal Akturk. "Bibliometric analysis of clinical decision support systems". In: *Acta Informatica Pragensia* 10 (1 2021), pp. 61–74. DOI: 10.18267/J.AIP.146.

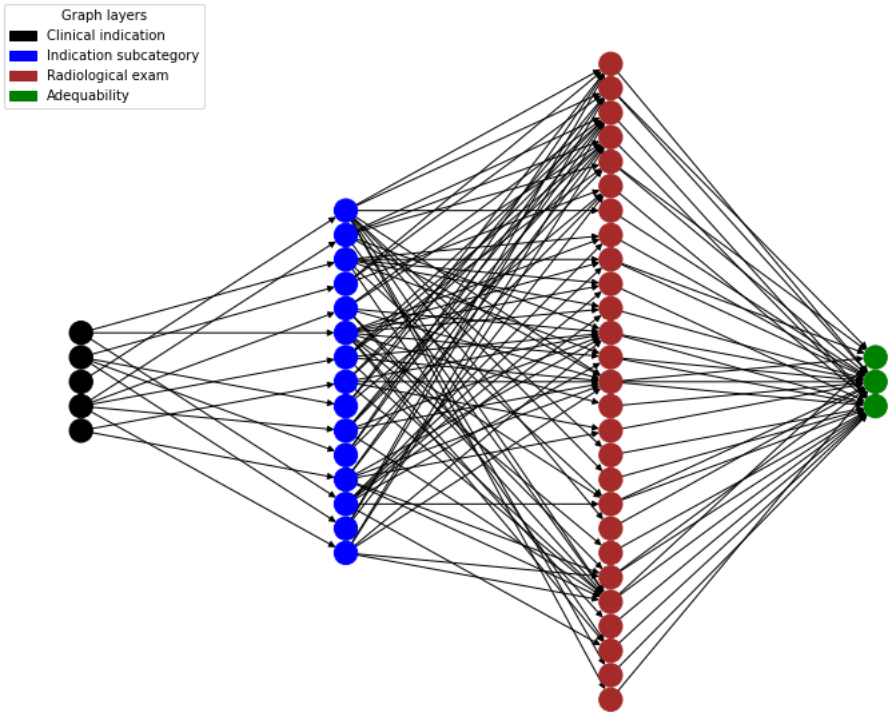

Figure 1: Graph representation for our experimental decision table, obtained using networkx package for the output of *from_paths_to_edges*.

[2] Joseph Doyle et al. "Clinical decision support for high-cost imaging: A randomized clinical trial". In: *PLOS ONE* 14 (3 2019), e0213373. DOI: 10.1371/journal.pone.0213373.

[3] Rafael Izbicki and Tiago Mendonça dos Santos. *Aprendizado de máquina: uma abordagem estatística*. Rafael Izbicki, 2020. ISBN: 6500024109.

[4] Ajit Kumar. "Stakeholder's Perspective of Clinical Decision Support System". In: *Open Journal of Business and Management* 04 (01 2016), pp. 45–50. DOI: 10.4236/ojbm.2016.41005.

[5] Vaishali Parsania and Nalin Jani. "Reviewing and Modeling Clinical Decision Support System". In: *International Journal of Technology and Science* 7 (Dec. 2015), pp. 15–17.

[6] Md Ileas Pramanik et al. "Healthcare informatics and analytics in big data". In: *Expert Systems with Applications* 152 (2020), p. 113388. DOI: 10.1016/J.ESWA.2020.113388.

[7] Reed T. Sutton et al. "An overview of clinical decision support systems: benefits, risks, and strategies for success". In: *npj Digital Medicine* 3 (1 2020), p. 17. DOI: 10.1038/s41746-020-0221-y.

[8] K M Tolle, D S W Tansley, and A J G Hey. "The Fourth Paradigm: Data-Intensive Scientific Discovery [Point of View]". In: *Proceedings of the IEEE* 99 (8 2011), pp. 1334–1337. DOI: 10.1109/JPROC.2011.2155130.

[9] J A N Vanthienen and Elke Dries. *Developments in decision tables: Evolution, applications and a proposed standard*. 1992, pp. 1–41.