Journal of Biomedical Informatics 104 (2020) 103407

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Explainable decision support through the learning and visualization of preferences from a formal ontology of antibiotic treatments

Jean-Baptiste Lamy[a,*], Karima Sedki[a], Rosy Tsopra[b,c,d]

[a] LIMICS, Université Paris 13, INSERM UMRS 1142, Sorbonne Universités, 93017 Bobigny, France
[b] INSERM, Université de Paris, Sorbonne Université, Centre de Recherche des Cordeliers, Information Sciences to support Personalized Medicine, F-75006 Paris, France
[c] Department of Medical Informatics, Hôpital Européen Georges-Pompidou, AP-HP, Paris, France
[d] INSERM, UMR 1099, LTSI Team Health Big Data, Université Rennes 1, Rennes, France

ARTICLE INFO

Keywords:
Clinical decision support system
Explainable artificial intelligence
Preference learning
Preference visualization
Ontologies
Antibiotics

ABSTRACT

The aim of eXplainable Artificial Intelligence (XAI) is to design intelligent systems that can explain their predictions or recommendations to humans. Such systems are particularly desirable for therapeutic decision support, because physicians need to understand rcommendations to have confidence in their application and to adapt them if required, *e.g.* in case of patient contraindication. We propose here an explainable and visual approach for decision support in antibiotic treatment, based on an ontology. There were three steps to our method. We first generated a tabular dataset from the ontology, containing features defined on various domains and *n*-ary features. A preference model was then learned from patient profiles, antibiotic features and expert recommendations found in clinical practice guidelines. This model made the implicit rationale of the expert explicit, including the way in which missing data was treated. We then visualized the preference model and its application to all antibiotics available on the market for a given clinical situation, using rainbow boxes, a recently developed technique for set visualization. The resulting preference model had an error rate of 3.5% on the learning data, and 5.2% on test data (10-fold validation). These findings suggest that our system can help physicians to prescribe antibiotics correctly, even for clinical situations not present in the guidelines (*e.g.* due to allergies or contraindications for the recommended treatment).

1. Introduction

Explainable Artificial Intelligence (XAI) [1–3] is a research field aiming to design intelligent systems capable of explaining their predictions or recommendations to humans. Various approaches have been proposed: (1) interpretable models make use of non-black box systems, such as a rule base or a formal ontology, (2) prediction interpretation and justification models produce explanations from a black-box algorithm [4], (3) hybrid approaches combine both types of model. Many XAI approaches are visual, either because the underlying Artificial Intelligence (AI) system focuses on image analysis and computer vision, or because they use visualization to convey large amounts of information in a small amount of space.

XAI has been studied in many applications including recommendation systems [5,6], classification [7] and in the military domain [8–10], but it is also of considerable interest in medicine, for explaining the predictions or recommendations of clinical decision support systems (CDSS) [11]. For diagnostic systems and medical image analysis, explanations are relatively easy to obtain: an image annotated with the contours of the detected anomalies can provide sufficient explanation. By contrast, for therapeutic systems, XAI is more difficult to achieve, due to the non-visual nature of patient data and the drug treatment. Some systems propose excerpts of clinical practice guidelines (CPGs) as an explanation [12]. However, the rationale underlying these recommendations is not always explicit in CPGs. The presence of detailed explanations can increase the trust and confidence of physicians in CDSS [13].

Many AI approaches can be used for therapeutic decision support, but they are not equally effective at providing explanations, particularly for situations in which patient and treatment data must be combined. One commonly used approach is classification (Fig. 1, top), which can be achieved through a formal ontology and a reasoner, or by machine learning on a prescription database. Classification approaches use patient data to classify a patient into one of several categories for treatment. This approach may, therefore, provide explanations based on these patient data. For example, it may explain that treatment *A* is

[*] Corresponding author.
E-mail addresses: jibalamy@free.fr (J.-B. Lamy), sedkikarima@yahoo.fr (K. Sedki), rosy.tsopra@nhs.net (R. Tsopra).

https://doi.org/10.1016/j.jbi.2020.103407
Received 7 October 2019; Received in revised form 3 March 2020; Accepted 4 March 2020
Available online 08 March 2020

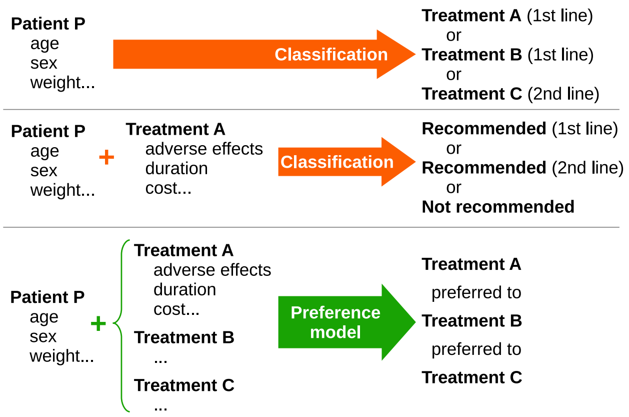

Fig. 1. Two approaches for therapeutic decision support: classification on patient data (top) and on patient and treatment data (middle) *vs* preference models (bottom).

recommended because patient *P* is a child. However, as classification is based exclusively on patient data, such systems cannot take treatment features into account in the explanations generated. In theory, the classification approach could be applied jointly to patient data and treatment features (Fig. 1, middle). However, in this context, classification approaches consider one patient and one treatment independently of each other, even if the treatments had been pre-sorted in rank groups (*e.g.*, first-line, second-line), whereas there is a need to compare treatments with each other based on patient information and treatment characteristics. For example, a “bad” treatment with many adverse effects may still be recommended if no better treatment exists.

Another approach is based on preference models. These models can be used to infer preference relationships for the ordering of treatments. For any two treatments, *A* and *B*, the model can be used to determine whether $A \succ B$ (*i.e.* *A* is preferred over *B*), $B \succ A$ or $A \approx B$ (*i.e.* no preference between *A* and *B*). Preference models can use both patient data and treatment features as input, considering all the possible treatments for each patient (Fig. 1, bottom). It can thus provide explanations that take treatment features into account. For example, treatment *A* may be recommended because patient *P* is a child and *A* has a nice taste (as children are known to refuse drugs that taste bad). This allows for more detailed explanations and thus a better understanding of the medical reasoning.

Moreover, preference models are also useful in situations in which the recommended option cannot be chosen (*e.g.* due to contraindications, allergies or patient refusal). The classification approach leaves a clinician with no useful advice if none of the treatment classes can be given for one reason or another. By contrast, the preference approach ranks all possible treatments. If the highest ranked treatment cannot be chosen, the second best can be considered, and so on.

Preference models can be elicited from experts, but their development is a difficult and time-consuming task. Another possibility is *preference learning*, in which a preference model is learned automatically from observed preference information, such as prescriptions in a database or expert recommendations in CPGs. However, the efficiency and quality of the learned preference model depend on the type of input provided to the learning system. Many preference learning approaches accord little importance to the structure of the input data, because these data often take the form of a simple dataset: instances described by a set of features (*i.e.* an “instance × feature” two-dimensional matrix).

Ontologies have recently been proposed as an appropriate technical choice for structuring the data used in preference learning. Ontologies are formal models that can be used to make knowledge accessible to a machine. In its narrow sense [14], an ontology contains all concepts and necessary relationships in a domain. Here, we adopt a broader sense, in which ontologies are knowledge bases encoded in a formal description-logic language like OWL. Such knowledge bases are particularly useful for the representation of highly hierarchical data (*e.g.* when the feature values are not defined for single instances but for classes of instances), a frequent situation in the medical domain. Ontologies have been shown to enhance the performance of preference-based recommendation systems [15–17]. Their use has also been proposed for the structuring of preference models [18]. However, it is more difficult to learn preferences from ontologies, for two reasons: (1) in preference learning, all features are defined on instances, whereas, in an ontology, features may have different domains (for example, some features may describe patient profiles, *e.g.* age class, and others may describe treatments, *e.g.* risk of adverse effects), and (2) some features may be *n*-ary properties, reified in the ontology.

In a recent study [19], we used preference learning to detect inconsistencies in CPGs for antibiotic treatment. We described the learning of a simple preference model from an ontology of antibiotic treatments [20,21] including CPG recommendations. This model distinguished between necessary and preference features, and associated a weight to each preference feature. It was sufficient for prediction purposes and for identifying CPG recommendations not matching predictions. However, it presented several limitations that made it less suitable for explaining the prediction to a physician, in an XAI perspective: (1) Learning was not fully reproducible, *i.e.*if the preference learning process is performed multiple times, different models are learned. These models had similar prediction performances, but could be very different, *e.g.* associating very different weights to the same feature. (2) The model could be ambiguous for humans to interpret. For example, it may give very similar weights to two features, *e.g.* 0.71 and 0.73, and it may not be clear to the physician whether these two weights are significantly different (*i.e.* should the feature with a value of 0.73 be regarded as more important than the one with a value of 0.71?). Moreover, if these weights are presented visually, *e.g.* on a bar chart, this difference may not be clearly visible. (3) Our preference learning method was limited to bipartite ranking (*i.e.* considering only two preference labels, “recommended” and “non-recommended”), whereas CPGs frequently have several levels of recommendation (*e.g.* recommended for first-line treatment, for second-line treatment, and not recommended). (4) Our ontology includes many missing values because of the specific features of the medical domain, in which much information may be missing. In accordance with the precautionary principle, these missing values were considered to take their worst possible value, but it would be interesting to explore how experts writing CPGs actually treat missing values.

Our objective here is to learn and visualize preferences from a knowledge base of antibiotic treatments. We present an improved and generalized method for learning and visualizing a preference model from an ontology, with a different objective from our previous work: clinical decision support and the development of recommendations that can be explained to humans. The method proposed solves the problems described above: (1–2) it considers additional optimization constraints and goals during learning, to facilitate the learning of a more reproducible and unambiguous model; (3) it supports multipartite ranking; and (4) it provides an understanding of the impact of missing values on the recommendations. Moreover, this method makes use of an ontology as the input for preference learning, and we describe the way in which it handles the difficulties encountered in this context, such as the possibility of features being defined on various domains, and the presence of *n*-ary features. We used a simple preference learning method, in which the learning process is reduced to an optimization problem resolved with a metaheuristic. Finally, we propose a method for visualizing the resulting preference model, using a set visualization technique called rainbow boxes [22]. This visualization can then be used to support and explain the decision provided.

We apply the proposed method to the design of a visual and explainable CDSS for empirical antibiotic treatment (*i.e.* antibiotic treatment prescribed without identification of the causal bacterium by microbiological culture). The ontology of antibiotic treatmentswas used as input for learning the implicit preference models used by medical experts in the writing of CPGs. The resulting CDSS [23] provides a visual display of the various antibiotics available on the market and their properties, weighted according to the preference model learned from the expert recommendations found in CPGs.

The rest of the paper is organized as follows. Section 2 presents related work on preference learning, set visualization and optimization. Section 3 describes the proposed method for generating a dataset for preference learning, starting from an ontology with features defined on various domains, *n*-ary features and missing values. Section 4 describes our preference model, and the way in which the learning of this model can be reduced to an optimization problem. Section 5 describes the visualization methods we applied to the preference model, and the additional constraint imposed by the visual approach. Section 6 explains how we solved the optimization problem. Section 7 presents the application of the proposed method to the ontology of antibiotic treatments. It also describes the resulting visual CDSS for antibiotic treatment and summarizes the results of the evaluation. Finally, Section 8 discusses the proposed method and the conclusion is presented in section 9.

2. Related work

2.1. Preference learning

Preference-based recommendation systems [24] recommend an option on the basis of a preference model. Such systems can be used on e-commerce websites to help clients to choose a book, a car, or a holiday destination, for example, but they can also be used for medical decision support, to help physicians to choose the most appropriate treatment option, such as the best drug to prescribe. In the medical domain, recommendation systems can be used to help the physician to provide personalized care [25], or to recommend a doctor for a patient [26], for example.

In these systems, the preference model can be elicited from experts, but this is often a difficult and time-consuming process. An attractive option is the learning of the preference model from observed preference information, because such data is easier to observe and collect. Examples of such data include customer transactions on an e-commerce website or medical expert recommendations in CPGs. This approach is known as preference learning [27], and is one of the research problems to have received considerable attention recently in disciplines such as AI, machine learning, data mining, and decision support. The aim is to learn preferences automatically and to construct a preference model from the observed preference information. Once the preference model has been learned, it can be used to improve our understanding of the domain and to provide decision support. In particular, the model can be used to explain the recommendations of the system to a human user.

An example of preference learning from ontology is provided by the work of Tsai and Wang et al. [28,29]. They proposed a learning objects recommendation model based on ontological approaches for e-learning systems. This model was based on course descriptions in SCORM (Sharable Content Object reference Model). The use of this ontology made it possible to infer course requirements, and to inherit instance feature values from their classes. By contrast to the work presented here, the authors did not consider features defined on heterogeneous domains and *n*-ary features. However, the use of the higher semantic richness provided by formal ontologies can be useful for the learning of preferences in complex domains, as we will show here for antibiotic treatment.

2.2. Set visualization

Set visualization involves the graphical representation of elements and sets, with each element possibly belonging to several sets. Many approaches have been proposed for set visualization [30]. We developed one of these approaches, rainbow boxes, a few years ago [22]. Rainbow boxes display the elements to be compared in columns, and the sets in labeled rectangular boxes that cover all the columns corresponding to the elements in the set. Larger boxes are placed at the bottom and two boxes can be side-by-side as long as they do not cover the same columns. A box can have holes, if the elements in the set are not displayed in consecutive columns. Finding the optimal column order that minimizes the number of holes is a combinatorial optimization problem with factorial complexity. The AFB metaheuristic (see below) is commonly used to solve this problem in a satisfactory time, up to about 50 columns in real time and more than 200 otherwise. Finally, we have also proposed a proportional version of rainbow boxes [31,32], in which the height of the boxes can vary, to represent a per set positive real value.

2.3. Metaheuristics

Several learning and visualization techniques, including rainbow boxes, require the resolution of optimization problems. Nature-inspired metaheuristics [33] are simple, but efficient and adaptable optimization algorithms. They are commonly used for both machine learning and visualization.

Here, we will use Artificial Feeding Birds (AFB) [34], a recently developed metaheuristic inspired by the behavior of pigeons. It considers a population of artificial birds (*e.g.* 20 birds). The position of each bird represents a candidate solution for the optimization problem. The algorithm performs several cycles. In each cycle, each bird performs one move, chosen from four: (1) walk to a random position close to the current one, (2) fly to a random position, (3) fly to the best position previously found by the same bird, and (4) fly to the current position of another random bird. Move #4 is allowed only for large birds, which represent 25% of the birds. Moves #3 and #4 are independent from the optimization problem, while moves #1 and #2 depend on the category of optimization problem. Consequently, AFB can be applied to any optimization problem defined by a triplet of functions (*cost*, *fly*, *walk*), where *cost* is the cost function to minimize, *fly* is a function that returns a totally random solution (corresponding to move #2) and *walk* is a function that returns a random solution close to another previous solution (move #1). AFB is currently the best option for optimizing rainbow boxes above 12 columns [34].

3. Generating a proper dataset from an ontology

Before learning preferences, it is necessary to extract from the ontology a proper dataset, *e.g.* an "instance × feature" matrix. This task is not trivial, because the interesting features for preference learning may not be defined on the same domain, and because they may be *n*-ary properties (reified in the ontology).

Let us consider an ontology $\mathcal{O}$ that consists of axioms describing a set of individuals $\mathcal{I}$, a set of classes C and a set of properties $\mathcal{R}$. A given class of individuals $\mathcal{X} \equiv \{x_1 \in \mathcal{I}, ...\}$ are the *instances* for preference learning. In addition, a subset of the properties $\mathcal{F} = \{p_1 \in \mathcal{R}, ...\}$ are the *features* for preference learning. Feature values can be asserted at various levels:

- On instances, *e.g.* a given drug in a drug-recommender system. This can be denoted $p(x, v)$ where x is the instance, p is the feature and v is the value.
- On classes of instances, *e.g.* on all drugs of a given therapeutic class. This can be denoted $c \sqsubseteq \exists\, p.\, V$ where c is a class of instances (*i.e.* $c \sqsubseteq \mathcal{X}$), p is the feature and V is the class of values p can take for instances belonging to class c.
- On non-instance individuals and classes, e.g. patient or patient categories (such as age classes, sex, *etc.). This can be denoted* $p(i, v)$

with $i \in \mathcal{I} \backslash \mathcal{X}$ *(for a non-instance individual) and* $c' \sqsubseteq \exists\, p.\, V$ *with* $c' \not\sqsubseteq \mathcal{X}$ *(for a non-instance class).*

Features can thus have any domain (not necessarily instances). Finally, we consider a set of preference formulas $\mathcal{P}$ observed between the instances, each formula defining a partial order on $\mathcal{X}$ of the general form $x_1 \approx x_2 \approx ... \succ x_3 \approx x_4 \approx ... \succ ...$, where $a \succ b$ means "a is preferred to b" and $a \approx b$ means "a and b are indifferent" (*i.e.* neither a nor b is preferred).

Example #1: A (trivial) ontology describes the drugs indicated for a given disorder. It contains the following classes: *Drug*, *PatientProfile* and *Prescription*. The features are *highCost* (domain: *Drug*, range: Boolean) and *isPregnant* (domain: *PatientProfile*, range: Boolean). The non-feature properties are *hasDrug* (domain: *Prescription*, range: *Drug*) and *hasPatient* (domain: *Prescription*, range: *PatientProfile*). Preference formulas express the observed preferences of various physicians or experts concerning prescriptions, *e.g.* $x_1 \succ x_2 \approx x_3$ (*i.e.* prescription x_1 is preferred to x_2 and x_3; but x_2 and x_3 are indifferent with neither preferred over the other), a given physician considered prescriptions x_1, x_2 and x_3, and finally prescribed x_1. The objective is to understand the reasons why one prescription is preferred over another, and to be in a position to recommend a prescription to physicians, on the basis of the two features (cost and pregnancy in this case). Here, individuals of the *Prescription* class are instances for preference learning. However, the two features are not defined in the same domain, whereas preference learning usually considers an "instance × feature" matrix.

The ontology can be used to "project" each feature onto the instances, to produce such a matrix from a complex ontology with features having heterogeneous domains. We use class definitions, rather than query languages such as SPARQL (SPARQL Protocol and RDF Query Language), because query languages are often limited when used at the class level and inference, in particular, may not be possible.

3.1. Features defined on non-instance domain

For each feature p the domain of which is not the instances (*i.e.* $\exists\, p.\, \top \not\sqsubseteq \mathcal{X}$), we define a property composition $q \circ ... \circ p$ that begins with q (the domain of which is $\mathcal{X}$, the instances), and ends with p. We create a new property p' with domain $\mathcal{X}$ and with the same range as p. Then, for each possible value v in the range of p, we create a class c_v equivalent to all instances indirectly related to v, and we assert a direct relationship using p'. Formally speaking, $c_v \equiv \mathcal{X} \sqcap q \circ ... \circ p.\, \{v\}$ and $c_v \sqsubseteq \exists\, p'.\, \{v\}$. In addition, the feature p may be defined at the class level. For each class $Y \sqsubseteq \exists\, p.\, V$ (where V is the class of values), we create a class C_V with $c_V \equiv \mathcal{X} \sqcap q \circ ... \circ p.\, V$ and $c_V \sqsubseteq \exists\, p'.\, V$.

Example #1 (continued): We create the *hasHighCostDrug* property (domain: *Prescription*, range: Boolean). We then define the class of all prescriptions for which the drug has a high cost, and we assert that all of them have a high cost drug:

$$\begin{aligned} HighCostPrescription \ &\equiv \quad Prescription \\ &\sqcap \ \exists\, hasDrug \circ highCost.\, \{True\} \\ HighCostPrescription \ &\sqsubseteq \ \exists\, hasHighCostDrug.\, \{True\} \end{aligned}$$

3.2. *n*-ary features

We consider the reification of each n-ary feature in n binary properties, p_1 to p_n. We distinguish one of the related entities as the range, which is the target of preference learning, and the $n-1$ others are viewed as the domain. We arbitrarily denote the range with index 1. As above, we create a new property p', with domain $\mathcal{X}$ and with the same range as p_1. For each possible value $(v_1, ..., v_n)$ of the n-ary feature $(p_1, ..., p_n)$, we create a class $c_{v_2,...,v_n}$, defined as the intersection of the instances related to $v_2, ..., v_n$ *via* properties $p_2, ..., p_n$, and we assert a direct relationship using p'. Formally speaking, $c_{v_2,...,v_n} \equiv \mathcal{X} \sqcap p_2.\, \{v_2\} \sqcap ... \sqcap p_n.\, \{v_n\}$ and $c_{v_2,...,v_n} \sqsubseteq \exists\, p'.\, \{v_1\}$. In addition, for each class $Y \sqsubseteq \exists\, p_1.\, V_1 \sqcap ... \sqcap p_n.\, V_n$, we create a class $c_{V_2,...,V_n}$ with $c_{V_2,...,V_n} \equiv \mathcal{X} \sqcap p_2.\, V_2 \sqcap ... \sqcap p_n.\, V_n$ and $c_{V_2,...,V_n} \sqsubseteq \exists\, p'.\, V_1$.

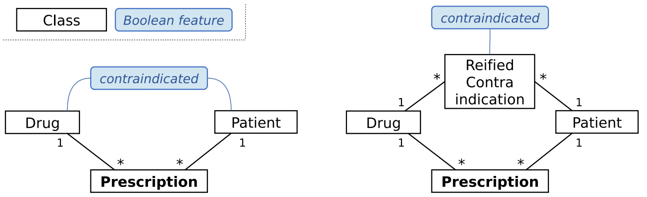

Fig. 2. Example of *n*-ary feature, before (left) and after (right) reification.

Example #2: We can extend the previous example with contraindications, which clearly affect the physician's choice. For the sake of simplicity, we consider only two levels of contraindication: *True* (*i.e.* contraindicated) and *False* (not contraindicated). Contraindications depend on both the patient profile and the drug. Contraindication is thus a ternary property between patient, drug and contraindicated status (Boolean) (Fig. 2). Contraindicated is the range for preference learning, whereas patient and drug are viewed as domains. For instance, the relation *contraindicated*(*pregnantPatientProfile*, *drugA*, *True*) means that *drugA* is contraindicated for pregnant women, with *pregnantPatientProfile* $\sqsubseteq$ *PatientProfile*. This ternary relationship can be "projected" onto prescription as follows:

$$\begin{aligned} &PrescriptionOfDrugAForPregnantPatient \\ &\quad \equiv Prescription \\ &\quad \sqcap \ \exists\, hasDrug.\, \{drugA\} \\ &\quad \sqcap \ \exists\, hasPatientProfile.\, \{pregnantPatientProfile\} \\ &PrescriptionOfDrugAForPregnantPatient \\ &\quad \sqsubseteq \exists\, contraindicated.\, \{True\} \end{aligned}$$

When the number of instances of the n-ary property is high, the creation of these classes is time-consuming and tedious. However, it can be automated by using a programming language to "preprocess" the ontology. We used Python scripts with Owlready 2 [35], an Open Source package for ontology-oriented programming.

The two difficulties addressed here (heterogeneous domains and n-ary properties) can be encountered on the same feature (*i.e.* an n-ary property having non-instance entities in its domain). In this case, the two solutions we propose can be combined.

Finally, using a reasoner such as HermiT [36], we can classify instances according to the classes defined by intention (c_v, c_V, $c_{v_2,...,v_n}$ and $c_{V_2,...,V_n}$) thereby obtaining, for each instance, the associated values of the p' properties. For a given instance and feature, if no value is obtained, the value is considered to be missing (*e.g.* if the presence of serious adverse effects is unknown for a given drug). If more than one value is obtained, the values are considered to be conflicting. In this case, depending on the feature, it can be decided: (a) to keep the worst value, if the values are ordered, or (b) to keep all the conflicting values. The preference learning method described below supports both missing and conflicting values.

4. Learning preferences from the generated dataset

Preference learning is performed on instances $\mathcal{X}$ and features $\mathcal{F}' = \{p_1', ...\}$, where p_i' are the new properties created in the previous section, defaulting to $p_i' = p_i$ for binary features having instances for domain (for which no new property is needed). For a given feature p', we use $\mathcal{V}_{p'}$ to denote the set of possible values.

4.1. Preference model

Many preference learning methods have been described. The method described here is designed to learn both:

- **Simple necessary constraints** $\mathcal{N}$, of the form $p' = v$ (*i.e.* the value of p' must be v),
- **Preferences** $\mathcal{W}$, expressed as weights, with one weight for each possible value of each feature.

The necessary constraints are mandatory: for instance, only drugs not contraindicated for the patient should be considered. On the contrary, preference weights quantify the importance of the various features and their value (higher values being preferred over lower values). We therefore formalize our preference model as $\mathcal{M} = (\mathcal{N}, \mathcal{W})$ where $\mathcal{N}$ is a subset of the set of all possible constraints and $\mathcal{W}$ is a tuple of weights, with one weight for each possible value of each feature[1]:

$$\mathcal{N} \subseteq \{p' = v \colon \forall\ p' \in \mathcal{F}',\ \forall\ v \in \mathcal{V}_{p'}\}$$
$$\mathcal{W} = (w_{p',v} \colon \forall\ p' \in \mathcal{F}',\ \forall\ v \in \mathcal{V}_{p'})$$

Note that, if the constraint $p' = v$ is present in $\mathcal{N}$, not all weights $w_{p',v'}$ for feature p' will subsequently be used. However, we leave these weights in the model to facilitate learning, because weights and constraints are learned simultaneously. Features not involved in the necessary constraints are called *preference features*.

Missing values are associated with a weight of 0 (arbitrary weight origin). For conflicting values (*i.e.* features with more than one value for a given instance, as defined at the end of the previous section), the sum of the weights of the values is used.

4.2. Reducing preference learning to an optimization problem

For instances satisfying the necessary constraints, we first define a utility function u that computes its utility. Instances with a higher utility are preferred over those with a lower utility, *i.e.* if $u(a) > u(b)$ then $a \succ b$. Function u computes the sum of the weights for each value associated with the instance:

$$u(x_i) = \sum (w_{p',v} \in \mathcal{W}\ such\ that\ p'(x_i, v))$$

Algorithm 1. Algorithm for the function E returning the error rate of a given model.

```
function $E(\mathcal{X}, \mathcal{P}, \mathcal{M})$:
  let $\mathcal{N}$ and $\mathcal{W}$ be the two parts of the model $\mathcal{M} = (\mathcal{N}, \mathcal{W})$
  let $\mathcal{X}_\mathcal{N} = \{x \in \mathcal{X} \colon x$ satisfies the necessary constraints $\mathcal{N}\}$
  let $\mathcal{X}_{\overline{N}} = \mathcal{X} \backslash \mathcal{X}_\mathcal{N}$
  let $e = 0$ be the number of errors found
  let $\mathcal{T}$ be a total order on $\mathcal{X}$, defined as follows:
    $x_i \succ x_j$ if and only if ($x_i \in \mathcal{X}_\mathcal{N}$ and $x_j \in \mathcal{X}_\mathcal{N}$ and $u(x_i) > u(x_j)$) or ($x_i \in \mathcal{X}_\mathcal{N}$ and
    $x_j \in \mathcal{X}_{\overline{N}}$)
    $x_i \approx x_j$ if and only if ($x_i \in \mathcal{X}_\mathcal{N}$ and $x_j \in \mathcal{X}_\mathcal{N}$ and $u(x_i) = u(x_j)$) or ($x_i \in \mathcal{X}_{\overline{N}}$ and
    $x_j \in \mathcal{X}_{\overline{N}}$)
  for each instance $x \in \mathcal{X}$:
    for each preference formula $pf \in \mathcal{P}$ involving $x$:
      if $pf$ is not compatible with the total order $\mathcal{T}$, then:
        $e = e + 1$
        break
return $\frac{e}{|\mathcal{X}|}$
```

We then define function E (Algorithm 1), which calculates the error rate obtained when model $\mathcal{M}$ is compared with the set of preference formulas $\mathcal{P}$ on instances $\mathcal{X}$. Function E first produces a total order $\mathcal{T}$ on $\mathcal{X}$, using the model: instances satisfying the necessary constraints are preferred over those that do not, and, among the instances satisfying the constraints, those with a higher utility are preferred. $\mathcal{T}$ is then compared with $\mathcal{P}$ to obtain the error rate E. For this comparison, we allowed the total order $\mathcal{T}$ (determined with the preference model) to be more precise than the observed preference formulas $\mathcal{P}$. Medically speaking, this means that, whenever two treatments are considered by experts to be equal, we allow the preference model to prefer one of them. This makes sense for recommendations, because, when two therapeutic options are recommended for first-line treatment, they may not be perfectly identical. By contrast, we did not allow the preference model to be less precise than the observed preference formulas. For example, we considered the total order $A \succ B \succ C$ to be compatible with the preference formulas $A \succ B \approx C$ and $A \approx B \succ C$. On the contrary, the total orders $A \succ B \approx C$ and $A \approx B \succ C$ were not considered to be compatible with the preference formula $A \succ B \succ C$, because in this case, the utility function u fails to predict some preferences.

Preferences can be learned by searching the model $\mathcal{M}^{best}$ that minimizes the error rate E. This is an optimization problem:

$$\mathcal{M}^{best} = \underset{\mathcal{M}}{\operatorname{argmin}}\,(E(\mathcal{X}, \mathcal{P}, \mathcal{M} = (\mathcal{N}, \mathcal{W})))$$

The resolution of this optimization problem yields the model with the best performance for prediction. However, this model may not be the most appropriate for visual explanation: it may be ambiguous (*e.g.* should very close weights be considered equivalent?) or difficult to represent visually. We therefore added additional constraints during the learning process, as described below.

5. Visualization of the preference model

One possible approach for decision support and the provision of explanations is the visualization of preferences. The visualization of the preference model $\mathcal{M}$ and its application to a set of instances $\mathcal{X}'$ (which may be $\mathcal{X}$, a subset of $\mathcal{X}$, or new instances not belonging to $\mathcal{X}$) can be seen as a set visualization problem. We consider instances as the elements of this visualization problem, and various sets representing the features of the instances and their impact on the preference model.

5.1. Reducing preference visualization to set visualization

We define three categories of sets: (1) sets of instances that do not satisfy a given necessary constraint, (2) sets of instances that satisfy all necessary constraints and have a given value for a given preference feature, associated with a strictly negative weight in the preference model (an argument against choosing this instance or a disadvantage), and (3) sets of instances that satisfy all the necessary constraints and have a given value for a given preference feature, associated with a strictly positive weight (an advantage). These sets can be formally defined as follows:

$$\mathcal{B}_{red} = \{\{x \in \mathcal{X}' \colon \neg\ p'(x, v)\}\ \forall\ (p' = v) \in \mathcal{N}\}$$
$$\mathcal{B}_{ora} = \{\{x \in \mathcal{X}' \colon x\ \notin\ \mathcal{B}_{red} \wedge p'(x, v)\}\ \forall\ p' \in \mathcal{F}',\ \forall\ v \in \mathcal{V}_{p'} such\ that\ p'\ not\ in\ \mathcal{N}\ and\ w_{p',v} < 0\}$$
$$\mathcal{B}_{gre} = \{\{x \in \mathcal{X}' \colon x\ \notin\ \mathcal{B}_{red} \wedge p'(x, v)\}\ \forall\ p' \in \mathcal{F}',\ \forall\ v \in \mathcal{V}_{p'} such\ that\ p'\ not\ in\ \mathcal{N}\ and\ w_{p',v} > 0\}$$

The resulting sets can be visualized with rainbow boxes. Fig. 3 shows a simple example, using the following preference model, instances and sets:

[1] In the definition, we use set-builder notation, i.e. $\{x \colon y\}$means "the set of all x such as y is true".

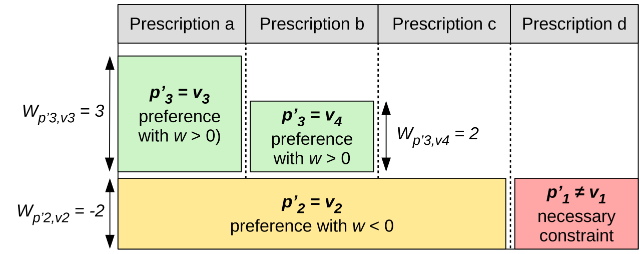

Fig. 3. General principles for the visualization of the preference model with rainbow boxes.

$$\mathcal{N} = \{p'_1 = v_1\}$$
$$\mathcal{W} = (w_{p'2,v2} = -2,\ w_{p'3,v3} = 3,\ w_{p'3,v4} = 2,\ w_{p'3,v5} = 0)$$
$$\mathcal{X}' = \{a, b, c, d\}$$
$$\mathcal{B}_{red} = \{B_{p'1 \neq v1} = \{d\}\}$$
$$\mathcal{B}_{ora} = \{B_{p'2=v2} = \{a, b, c\}\}$$
$$\mathcal{B}_{gre} = \{B_{p'3=v3} = \{a\},\ B_{p'3=v4} = \{b\}\}$$

In the rainbow boxes, instances are represented by columns, and sets by rectangular colored boxes. Each box covers the columns corresponding to the instances belonging to the set, and includes a label describing the feature and the associated value. Sets belonging to $\mathcal{B}_{red}$, $\mathcal{B}_{ora}$ and $\mathcal{B}_{gre}$ are colored in red, orange and green, respectively. The height of sets in $\mathcal{B}_{red}$ is arbitrary, whereas that in $\mathcal{B}_{ora}$ and $\mathcal{B}_{gre}$ is proportional to the absolute value of the corresponding weight $w_{p',v}$. Boxes are stacked vertically, with the largest ones at the bottom. Two boxes may be placed next to each other, as long as they have no common instances. Finally, columns are ordered such that instances belonging to the same sets are contiguous, with a combinatorial optimization algorithm (AFB, see Section 2.3). Whenever instances belonging to a given set cannot be placed in contiguous position, there is a "hole" is present in the box: the box consists of two rectangles, linked by a small stem. In addition, for ease of reading, all columns for antibiotics that do not satisfy the necessary constraints are grouped together, on the right.

This visualization can be used for rapid identification of the best instance: this is the column with no red boxes, the lowest total height of orange boxes and the greatest total height of green boxes. As the boxes are stacked, it is easy to determine total box height by eye. This visual computation corresponds directly to the utility function u defined above. In addition, the visualization explains why a given instance is preferred over another, by displaying their features and the learned weight for each. Finally, if the best instance cannot be chosen (*e.g.* due to a known patient allergy, in a medical context), the second best, and so on, can easily be determined. In Fig. 3, the best instance is a, and the overall ranking $\mathcal{T}$ is $a \succ b \succ c \succ d$ (as defined in Algorithm 1). Here, a is preferred over b because the value v_3 for feature p'_3 has a higher weighting than the value v_4.

5.2. Adapting the learning process for visualization

The visualization of the preference model with rainbow boxes imposes several constraints on the model. First, the boxes must be tall enough to include a label. The heights of the various boxes must also be sufficiently different for the difference to be readily detectable by the human eye. For example, two boxes with heights of 0.71 and 0.73 may be seen as having the same height. We therefore considered only integer values for weights, to prevent ambiguities caused by similar weights, and we forbade the values 1 and − 1 to prevent boxes from being too small, thus $w_{p',v} \in \{\ldots,-4, -3, -2, 0, 2, 3, 4\ldots\}$.

Second, the vertical space required by the visualization depends on weights: smaller weights are preferable to limit the amount of vertical space required. Thus, during the learning process, we try to minimize the error rate, but also the sum of the weights, denoted S_w. As previously noted, if the necessary constraint $p' = v$ is present in $\mathcal{N}$, none of the weights $w_{p',v'}$ have an impact (and are not displayed on rainbow boxes). We therefore excluded these weights when computing the sum, as follows:

$$S_w = \sum_{p'} \sum_{v} (w_{p',v}\ \textit{such that}\ \nexists v'\ \textit{with}\ (p' = v') \in \mathcal{N})$$

Algorithm 2. Algorithm for the *fly* and *walk* functions for optimization problems involving a mixture of combinatorial and global non-linear optimization.

```
function fly():
  let $\mathcal{N}$ be a random subset of the possible constraints
  let $\mathcal{W} = (w_{p',v} =$ random integer value between $-1$ and $9: \forall p' \in \mathcal{F}', \forall\, v \in \mathcal{V}_{p'})$
  return $\mathcal{M} = (\mathcal{N}, \mathcal{W})$
function walk($\mathcal{M}$):
  let $\mathcal{M}' = (\mathcal{N}', \mathcal{W}')$ be a copy of $\mathcal{M}$
  let $r$ be a random real number between 0 and 1
  if $r < 0.15$:
    add a random constraint in $\mathcal{N}'$
  else if $r < 0.3$:
    remove a random constraint in $\mathcal{N}'$
  else:
    modify a random weight in $\mathcal{W}'$ by $+1$ or $-1$
  return $\mathcal{M}'$
```

Beyond visualization, minimizing the sum of the weights has several advantages for rendering the preference model more understandable. First, by multiplying all weights by a given constant, it is possible to obtain a different, but equivalent, model. Minimizing the sum of the weights prevents this problem, thereby making the learning process more reproducible. Furthermore, we do not count weights associated with features involved in necessary constraints, and this favors models with a larger number of necessary constraints. This is desirable, because necessary constraints are usually easier to understand than preferences. For example, it is easier for a clinician to apply the recommendation "do not prescribe drugs with serious adverse effects", rather than "prefer drugs without serious adverse effects, but consider the tradeoffs with other properties such as efficacy".

We therefore search for the model $\mathcal{M}^{best}$ minimizing both the error rate E and the weight sum S_w, in lexicographic order (*i.e.* we minimize the error rate and, in cases of equality between two models, we prefer the model with the lowest weights). The optimization problem is now:

$$\mathcal{M}^{best} = \underset{\mathcal{M}}{\text{argmin}}\,(E(\mathcal{X}, \mathcal{P}, \mathcal{M}), S_w)$$

6. Solving the optimization problem

This optimization problem is complex, because it involves a mixture of combinatorial optimization (for optimizing $\mathcal{N}$) and global non-linear optimization (for optimizing $\mathcal{W}$). However, $\mathcal{N}$ and $\mathcal{W}$ need to be optimized simultaneously, because they are interdependent.

We solved this problem with Artificial Feeding Birds (AFB, see section 2.3). We chose this algorithm because of its generic nature and its ability to solve problems involving a mixture of combinatorial and global non-linear optimization. It can solve any optimization problem defined by a triplet of three functions (*cost*, *fly*, *walk*). Algorithm 2 shows the *fly* and *walk* functions we defined for optimizing the preference model $\mathcal{M}$. We used the default parameter values for the AFB metaheuristics, as described in [34].

Table 1
The 11 features in the knowledge base.

#	Feature [*short name*] Definition
1	**Naturally active against the causal bacterium [*naturally active*]** Whether the causal bacterium is described as sensitive or of intermediate sensitivity to the antibiotic (*e.g.* amoxicillin is naturally active against group A streptococci)
2	**Probably active against the causal bacterium [*probably active*]** Whether the frequency of resistance in the causal bacterium is considered low for the antibiotic (*e.g.* ceftriaxone is probably active against *E.coli*)
3	**Proven clinical efficacy against the disease [*proved*]** Whether the antibiotic is described as clinically effective for treating the infection OR is (or has been) indicated/recommended for treatment of the infection (*e.g.* penicillin G has proven clinical efficacy against pharyngitis)
4	**Absence of contraindications for the patient [*not contraindicated*]** Whether there is no absolute contraindication of the antibiotic for the patient profile (*e.g.* Ppristinamycin is not contraindicated for children over the age of six years)
5	**Convenient protocol [*protocol*]** Whether the antibiotic is prescribed for oral administration AND for a short duration (*e.g.* fosfomycin trometamol has a convenient protocol in uncomplicated cystitis)
6	**Non-precious class [*not precious*]** Whether the antibiotic does not belong to a class of drugs that must be preserved for more serious infections (*e.g.* amoxicillin is a non-precious class in sinusitis)
7	**Absence of serious and frequent side effects [*no side ef*]** Whether there is no serious side effects mentioned AND the frequency of side effects is sufficiently low for antibiotic prescription to be allowed (e.g. fosfomycin trometamol gives no serious side effects, and the side effects reported are rare)
8	**High level of efficacy [*efficacy level*]** Whether the antibiotic is described as very effective (high clinical cure rate, *e.g.* levofloxacin is very effective in prostatitis)
9	**Narrow antibacterial spectrum [*spect*]** Whether the antibiotic is described as having a "narrow" antibacterial spectrum (*e.g.* nitrofurantoin has a narrow activity spectrum)
10	**Low level of ecological adverse effects [*low eco risk*]** Whether the antibiotic is described as having a low risk of promoting bacterial resistance (*e.g.* Ppivmecillinam has a low level of ecological risk)
11	**Taste [*taste*]** Whether the antibiotic has an acceptable taste for the patient (*e.g.* Ccefuroxime axetil has a bad taste and thus is not acceptable for children)

7. Application to the design of a visual CDSS based on an ontology of antibiotic treatments

7.1. Context

National health authorities publish CPGs to help physicians to prescribe the correct antibiotic. In these CPGs, experts recommend prescribing particular antibiotics on the basis of drug properties, the infectious disease and the patient's condition. For example, they recommend fosfomycin trometamol for uncomplicated cystitis in women, because of its particular properties, such as its activity against *E. coli*. However, the preference model used by the experts for recommending antibiotics is not explicit in CPGs, and this can lead to a misunderstanding of the recommendations by GPs, resulting in the poor adoption of recommendations. In this study, we aimed to explain the implicit preference model used by the experts for recommending antibiotics and to use this model for clinical decision support in primary care.

In previous studies [20,21,19,37], we built a knowledge base describing antibiotics in terms of 11 features used by experts to establish recommendations (Table 1). Each feature is Boolean, and its value depends on the antibiotic, the patient profile (*e.g.* child or pregnant woman), the infectious disease (*e.g.* cystitis) and/or the likely causal bacteria (*e.g. Escherichia coli*). The True value corresponds to an advantageous property, the False value to a disadvantageous property, and unknown values are considered as missing values. The knowledge base was built and populated by a medical doctor (RT) using data from multiple CPGs, and was then validated by a panel of experts in antibiotics through a Delphi Process.

This knowledge base was formalized as an OWL 2.0 ontology [19]. It contains 144,038 RDF triples describing 5696 classes, 19 properties and 34,483 axioms, and it belongs to the $\mathcal{ALC}(\mathcal{D})$ family[2] of description logics (DL). Fig. 4 shows the general structure of the ontology. It has five main classes: *Antibiotic*, *PatientProfile* associated with an infectious *Pathology* caused by likely *Bacteria*. A *Prescription* is the association of an *Antibiotic* with a given *PatientProfile*. The 11 features are defined on five different domains (none of which is *Prescription*), and include three binary properties, 7 ternary properties and 1 quaternary property. Most feature values are not specified at the individual level but at the class level, due to the inheritance relationships for both antibiotic families and patient profiles. For example, all fluoroquinolones are contraindicated in children. Ofloxacin is a fluoroquinolone, and it therefore inherits this contraindication. This important use of inheritance was our main motivation for using an ontology. Many missing values are present, due to the large amount of unknown knowledge in the medical field. In addition, for each *Prescription* recommended in CPGs, the rank of recommendations is presented (1 to 4, with lower values for rank preferred). This ontology is not publicly available, but can be made available on request.

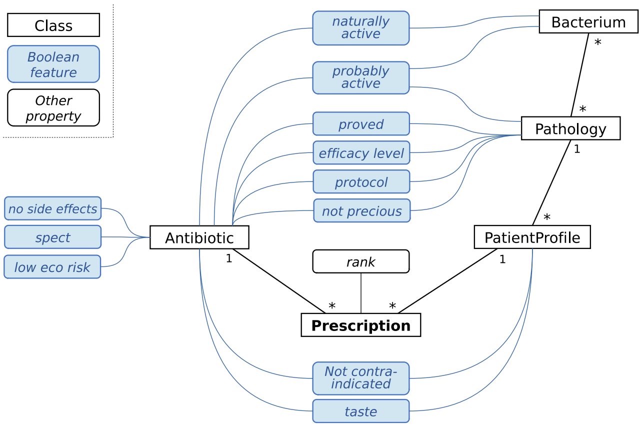

Fig. 4. General structure of the antibiotic treatment ontology. Ternary and quaternary properties (all but the three on the left) are reified in the ontology.

7.2. Application of the proposed method

We applied the proposed method to this ontology. Prescriptions are the instances for the purpose of preference learning. Ranks of recommendations were translated into a set of preference formulas $\mathcal{P}$; for example, if a CPG recommends prescription A in rank 1 and

[2] $\mathcal{AL}$: attribute language (including atomic negation, concept intersection, universal restriction, existential qualification limited to class Thing), $\mathcal{C}$: complex negation, $(\mathcal{D})$: use of datatypes [38].

prescriptions B and C in rank 2 for a given *PatientProfile*, this was translated as $A \succ B \approx C \succ D \approx E \approx$ ...(where D, E, ...are all the other possible prescriptions).

All features in the ontology correspond to potential advantages of antibiotics (*e.g.* low frequency of adverse effect, high efficacy). We therefore restricted the weights for *False* values to negative numbers, and those for *True* values to positive numbers (*i.e.* $w_{p',False} \leq 0$ and $w_{p',True} \geq 0$). This prevents the learning of medically absurd models, such as a model in which the prescription of antibiotics with many adverse effects is preferred over that of antibiotics with fewer adverse effects. When conflicting values were encountered (*e.g.* a *Pathology* associated with two likely causal *Bacteria*, one of which is resistant to a given *Antibiotic* whereas the other is not), we retained the worst value (*i.e. False*).

7.3. Evaluation of the learning process

We performed a 10-fold cross-validation to evaluate the learning process. The available data was randomly split in 10 subsets. One subset was reserved for use as the test set, and the other nine were used for learning. This process was then repeated with a different subset as the test set, until all the subsets had been used as the test set (for a total of 10 iterations). For each of the 10 subsets, we ran AFB for 3,000 iterations and retained the best model (*i.e.* $\mathcal{M}^{best}$ in Section 5.2). The mean error rate was 3.5% on the learning set and 5.2% on the testing set. The low error rate for the test set suggests that the learned preference model is sufficiently generic and can be generalized well to situations not present in the learning data.

7.4. Resulting preference model

We produced the final preference model from the entire dataset. We ran AFB for 3,000 iterations, we performed 10 runs, and we retained the best model. The best model has an error rate of 3.5%. The best model was found in five runs, at iteration 2,378 on average.

The best model is described in Table 2. The constraints learned show that antibiotic prescriptions must have a *True* value for four features (*naturally active*, *probably active*, *proved* and *not contraindicated*) to be recommended. This is clinically relevant, because only microbiologically and clinically effective antibiotics should be prescribed to guarantee that the patient is cured, and contraindications should be avoided to guarantee patient safety. Indeed, the features included in the necessary constraints were considered the most relevant for prescriptions by our medical doctor (RT). The other models identified were very similar to the best model. In particular, the necessary constraints were the same and the *protocol* and *no side ef* feature were associated with higher weights. These models were associated with a higher weight sum S_w or (more rarely) a slightly higher error rate.

Table 2
Necessary constraints $\mathcal{N}$ (top) and preference weights $\mathcal{W}$ learned (bottom; weights are not shown when, due to necessary constraints, they have no impact).

$$\begin{aligned} & \textit{naturally active} = \textit{True}\\ \wedge\ & \textit{probably active} = \textit{True}\\ \wedge\ & \textit{proved} = \textit{True}\\ \wedge\ & \textit{not contraindicated} = \textit{True} \end{aligned}$$

#	Feature	$w_{p',False}$	$w_{p',True}$
5	*protocol*	−7	4
6	*not precious*	−2	2
7	*no side ef*	−5	2
8	*efficacy level*	−2	2
9	*spect*	−2	2
10	*low eco risk*	−3	2
11	*taste*	−3	2

The preference model shows the importance of each feature for the choice of antibiotic. The features *naturally active*, *probably active*, *proved* and *not contraindicated* are very important because they are associated with the necessary constraints. The feature *protocol* and *no side ef* are important too, but less so, because they are associated with the highest weights (in absolute values).

In addition, the model provides an understanding of how experts interpret missing values when writing CPGs. For preference properties, missing values correspond to a weight of 0. Consequently, the position of the 0 between the two weights of a given property gives an idea of how missing values are interpreted. For example, the weights for the *efficacy level* feature are $w_{\textit{efficacy level},False} = -2$ and $w_{\textit{efficacy level},True} = 2$. Thus, missing values are interpreted as exactly "in between" high and low efficacy. By contrast, for the*no side ef* feature, the weights are $w_{\textit{no side ef},False} = -5$ and $w_{\textit{no side ef},True} = 2$. This suggests that experts tend to consider antibiotics with unknown serious/frequent adverse effects as being more like those with no such effects than those with adverse effects. This may appear surprising, because it might seem to violate the principle of precaution. However, in practice, a drug is usually considered to have no serious/frequent adverse effect until such effects are discovered in medical practice. From this standpoint, the expert behavior identified here makes sense.

7.5. Visual and explainable CDSS

We automatically generated the proposed visualization for all clinical situations modeled in the ontology, for a total of 66 situations, each corresponding to a given patient profile and a given infectious disease. We integrated the visualization into a CDSS, AntibioHelp®. This CDSS displays all the antibiotics present in the ontology in columns, and their properties in colored boxes. Fig. 5 shows an example, for uncomplicated cystitis in women. The colored boxes give the properties of each antibiotic and their importance. Properties are identified with icons and labels; for small boxes, the entire label can be obtained by hovering the mouse over the box.

In this example, the pathology considered is "uncomplicated cystitis", and the patient profile is "Adult woman (non-pregnant)". In empirical antibiotic treatment, the probable casual bacterium for this pathology is *E. coli*. The *naturally active* feature is a ternary property that depends on the antibiotic and the bacterium. Therefore, any antibiotic A (such as clarithromycin) that is not *naturally active* against *E. coli*, *i.e.* ternary property value (A, *E. coli*, *True*) does not hold, will belong to the red box "Bacterium with natural resistance". Other features involved in necessary constraints are treated similarly.

For each preference feature, two boxes (one green and one orange) are shown. Let us consider the *protocol* feature (labeled "convenient"/ "not convenient" in Fig. 5). It is a ternary property that depends on the antibiotic and the pathology. Thus, any antibiotic A_1 (such as ceftriaxone) that does not have a convenient *protocol* for uncomplicated cystitis, *i.e.* ternary property value (A_1, *uncomplicated cystitis*, *False*), is assigned to the orange box "Not convenient". Any antibiotic A_2 (such as fosfomycin trometamol) that has a convenient *protocol* for uncomplicated cystitis, *i.e.* ternary property value (A_2, *uncomplicated cystitis*, *True*), is assigned to the green box "Convenient". Any antibiotic A_3 (such as moxifloxacin) for which the value is missing, *i.e.* there is no ternary property value (A_3, *uncomplicated cystitis*, X), $X \in \{True, False\}$, is assigned to neither the green box nor the orange box.

In Fig. 5, 12 antibiotics (of the 42 considered) satisfy the necessary constraints and could therefore be prescribed. At a glance, we can see that one of them, fosfomycin trometamol, has the highest total heights of green boxes and lowest total weights of orange boxes. It is, therefore, the most appropriate antibiotic in this clinical situation. Furthermore, in the column header, we can see that this is the antibiotic

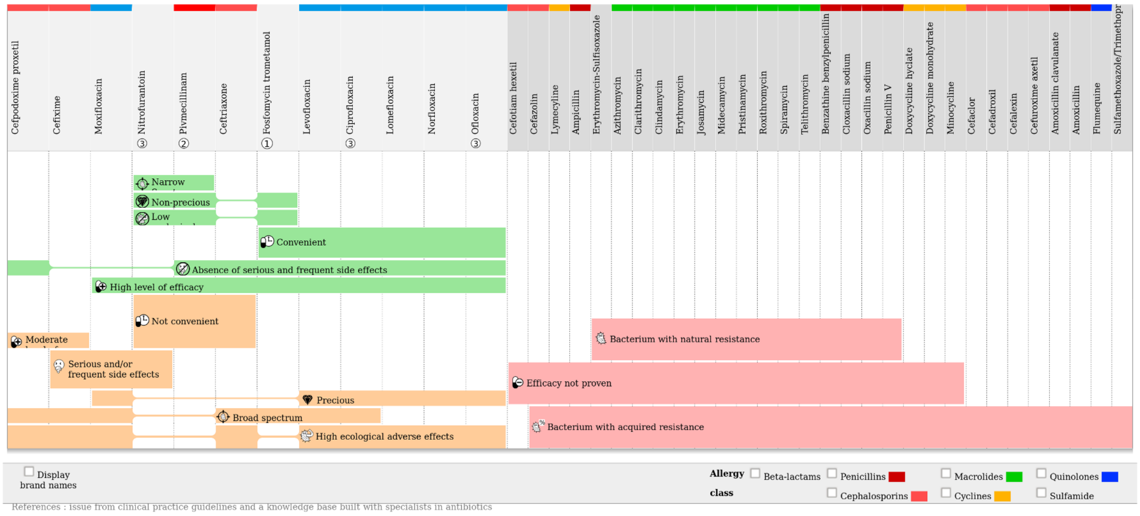

Fig. 5. Example of the proposed visual CDSS, showing the 42 antibiotics (columns) and their properties (boxes) when used for the treatment of uncomplicated cystitis in women. Red boxes correspond to unsatisfied necessary constraints, orange boxes to negative weights (*i.e.* disadvantages) and green boxes to positive weights (*i.e.* advantages). Height of orange and green boxes is proportional to the weight learned (Table 2). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

recommended in rank #1 in CPGs.

The interface can be used to identify the recommended antibiotic, but it also aims to explain why it is recommended and preferred over other antibiotics. For example, in Fig. 5, the physician can easily understand that fosfomycin is the most appropriate because it has many advantages in terms of protocol, side effects, efficacy and ecological risk. In addition, the interface is interactive: it allows the physician to filter out antibiotics on the basis of therapeutic classes, using the checkboxes at the bottom of the screen. When the patient has allergies (*e.g.* allergy to beta-lactams), this makes it possible to filter out the antibiotics to which the patient is allergic.

7.6. Evaluation of the CDSS

In a recent study [23], we investigated whether displaying the weighted preference properties could increase the confidence of General Practitioners (GPs) in CPG recommendations, and help them to extrapolate recommendations to patients for whom CPGs provide no explicit recommendations (*e.g.* because the recommended antibiotic cannot be prescribed due to allergies or contraindications). With this goal in mind, we carried out a two-stage crossover online study. GPs were asked to respond to clinical cases using CPG recommendations, either alone or with explanations displayed through the interface. We compared their responses with a gold standard derived blindly from the responses of two medical doctors. GP confidence was measured for each clinical case, on a seven-point percentage-based scale (2, 10, 25, 50, 75, 90, 98% certainty).

In total, 64 GPs were enrolled in the study. The display of the weighted preference properties significantly decreased the error rate ($-$ 41%, $p\,value = 6 \times 10^{-13}$), and significantly increased GP confidence ($+$8%, $p\,value = 0.02$) for situations for which there were no explicit recommendations. By contrast, no significant effect was found for situations in which there were explicit recommendations. GPs found the interface usable (SUS score $= 64$). Thus, these results suggest that the proposed CDSS can improve antibiotic prescription in situations for which there are no explicit recommendations, through visualization of the weighted antibiotic properties.

8. Discussion

We describe here a general method for learning preferences from a formal ontology, and for visualizing the resulting preference model with rainbow boxes. We demonstrate the use of the proposed method on an ontology in antibiotic treatment, and we present the learned preference model. Having developed this model, we then used it to design a visual CDSS for antibiotic prescriptions in primary care.

8.1. Preference learning

Preference learning made it possible to build a preference model, whereas the direct elicitation of the model from experts would have been difficult. An expert may recommend drug A rather than drug B in a given clinical situation, and he or she may argue his or her decision, but can hardly quantify and weight each argument precisely. By contrast, it is easier for an expert to populate a knowledge base describing the qualitative properties of drugs, such as efficacy, or antibiotic resistance.

Our method is based on an ontology. It can learn preferences from features that are defined with heterogeneous domains, from features that are reified *n*-ary properties, and it takes into account missing values. The use of ontologies in preference learning has two clear advantages. First, feature values can be expressed at class level rather than instance level, which is useful in highly hierarchical domains such as the medical domain. For example, “macrolides are not active against *E. coli*” involves a ternary property (*naturally active*), an antibiotic class (macrolides, which includes several instances of antibiotics) and a bacterium instance (*E. coli*). Moreover, the various features hold for heterogeneous domains (see Fig. 4): antibiotic, (antibiotic, patient) pairs (considering ternary properties), (antibiotic, pathology) pairs, (antibiotic, bacterium) pairs, and (antibiotic, pathology, bacterium) triples. Second, it makes it possible to use all the tools developed for OWL ontologies, including the Protege editor and the Owlready ontology-oriented programming module, which can translate OWL ontologies into SQL databases, thus having the advantages of both ontologies and relational databases.

In Section 3, the proposed method requires the creation of many classes for dealing with heterogeneous domains and *n*-ary features. We suggested the use of a programming language to “preprocess” the ontology and create these classes. The main limitation of this approach is that it takes only asserted facts into account, ignoring facts that can be inferred. Another solution would have been the use of rules based on First-Order Logic. However, such rules have an important limitation: they work on individuals but not on classes. The “preprocessing” approach should therefore be chosen when features are asserted at the class level, whereas First-Order Logic should be chosen when there is a need to consider inferred facts. The antibiotic treatment ontology considered here includes many features for which values are provided at the class level. For example, all penicillins A are contraindicated for patients allergic to amoxicillin, but "penicillins A" is a class of several antibiotics rather than a single antibiotic. We therefore chose to use the "preprocessing" approach.

After generating the tabular dataset, we performed preference learning with an optimization algorithm. However, many other preference learning techniques, such as Choquet integral [39], could have been used.

We performed several experiments to evaluate the impact on preference learning of adding a new feature. In these experiments, we ran the learning process after removing one feature. When the *spect* feature, associated with low weights, was removed, the error rate increased to 3.9% and the weights of the other features were modified, but the changes were limited: no weights were modified by more than $+1/-1$, and the *protocol* and *no side ef* features were still associated with the highest weights. Removal of the *protocol* feature, which is associated with high weights, led to the error rate increasing to 3.8% and larger changes in the weights of the other features. The *efficacy level* feature had the highest weights, potentially reflecting possible redundancy between these two features: more effective antibiotics may require a shorter treatment period and, therefore, a simpler protocol. The impact of adding a new feature therefore depends on the importance of the feature added. However, as our knowledge base was based on CPGs and validated by several medical doctors, including the referent doctor for infectious diseases at our hospital, we are confident that it contains the most important features for prescribing antibiotics.

The proposed preference model had an error rate of 3.5% (115 errors in 3,300 prescriptions). In our previous study [19], we used preference learning to identify candidate inconsistencies in CPGs. Our previous preference model had slightly higher error rates (3.8% for first-line treatment and 4.0% for any line), and about half these errors were manually classified as being due to inconsistencies in CPGs. Here, 51 of the 55 errors previously considered to be due to inconsistencies in CPGs were still present. Consequently, at least $51/115 = 44\%$ of errors are related to inconsistencies in CPGs. The others errors may be related to limitations of the preference model or the knowledge base.

8.2. Visualization

We used rainbow boxes to visualize preferences. Preference learning is a well-established topic in computer science, but very few studies have focused on the visualization of a preference model. D Bogdanov et al. [40] visualized a preference model in music, using set visualization: Euler diagrams drawn on top of a two-dimensional semantic projection. In a previous work [41], one of the authors (JBL) used rainbow boxes for XAI, but following a totally different approach, using case-based reasoning rather than preference learning. This suggests that set visualization, and rainbow boxes in particular, can be of particular interest for XAI. This is not particularly surprising, given the frequent use of set theory in AI.

One of the perspectives opened up by this work is a comparison of the proposed visual tool with other possible presentation of the explanations, such as a simpler textual list of advantages and disadvantages, or more complex visual datamining tools such as the one proposed by H Ltifi et al. [42] for monitoring nosocomial infections.

8.3. Optimization method

We used the AFB metaheuristic, because of its generic nature and its suitability for use in global non-linear optimization and combinatorial optimization.One of the drawbacks of AFB is that, due to its metaheuristic nature, it is partly random and cannot therefore be entirely reproducible (unless a fixed random seed is used). This can be problematic for the generation of explanations. However, for complex optimization problems, metaheuristics are often the only option available. In particular, as stressed above, the learning of the preference model requires a mixture of global non-linear optimization and combinatorial optimization, and the optimization of rainbow boxes has a factorial complexity and, with 42 columns, the solution space is huge. Moreover, the addition of constraints to the learning process (as detailed in Section 5.2) made the learning almost reproducible. Finally, in Algorithm 2, we used arbitrary threshold values for r (0.15 and 0.3) and the default values for the other AFB parameters. Optimal values could be sought, *e.g.* using another (or the same) metaheuristic. However, the entire learning process takes less than three minutes on a state-of-the-art laptop computer, and we did not, therefore, feel the need to spend time optimizing these parameters.

8.4. Ontologies in antibiotic treatment

We chose to create our ontology from scratch, because we found no resources entirely suitable for our purposes: the existing resources (DrugBank[3], RxNorm[4], [43,44]) describe marketed products (*e.g.* Clamoxyl®) rather than substances (*e.g.* amoxicillin), they do not contain all the features required for our system (*e.g.* ecological risk), and they do not categorize each feature as an advantage or a disadvantage with regard to prescription. Our literature review identified two ontologies developed for antibiotic prescriptions in hospital. The first, IDDAP, [43] allows proposing a list of "appropriate" antibiotics according to relationships between infectious diseases/antibiotics, antibiotics/bacteria, antibiotics/patient and antibiotics/drugs. However, the medical content of this ontology was based partly on non-validated resources (such as Wikipedia), it was not checked by antibiotic specialists, and some important features (*e.g.* narrow spectrum) were missing. The second ontology [44] was populated from robust sources (CPG, knowledge of antibiotic specialists) and allows generating a list of antibiotics based on matches between antibiotic/bacteria, antibiotic/patients and antibiotic/diseases. However, both ontologies produce recommendations different from those found in CPGs.

We therefore constructed our own knowledge base and ontology manually, from medical textual resources and expert knowledge. In the future, we plan to update the ontology automatically through external resources. For example, we could use microbiological observatories to update microbiological properties (*e.g.* for bacteria, natural sensitivities), drug databases for updating drug properties (*e.g.* for contra-indications), pharmacovigilance databases for updating side effects, and Medline databases for updating properties relating to efficacy (*e.g.* evidence of clinical efficacy). The incorporation of regularly updated data should improve the quality of healthcare and increase the adoption of this system by physicians [45].

8.5. Comparison with other CDSS for antibiotic prescription

Very few preference-based approaches have been proposed for medical decision support. In the antibiotic domain [46–49], other types of reasoning have been proposed: production rules (*e.g.* MYCIN [50], ADVISE [51]), fuzzy logic (*e.g.* FCM-uUTI DSS [52,53], Terap-IA [54]), causal probabilistic networks (*e.g.* TREAT [55]), and logistic regression models (*e.g.* HELP [56]). However, these types of reasoning may be difficult for physicians to understand, potentially impeding the adoption of these systems. Furthermore, the criteria considered by these systems and the recommendations they produce do not match those of CPGs.

Our approach aims to overcome these limitations. We made the reasoning used by clinical experts for recommending antibiotics explicit, through a preference model. This preference model is displayed

[3] https://www.drugbank.ca

[4] https://www.nlm.nih.gov/research/umls/rxnorm

in the form of rainbow boxes presenting the recommended antibiotics, but also the non-recommended antibiotics, with their weighted properties. This (i) helps physicians to deal with situations not described in CPGs (*i.e.* if the physician cannot prescribe the recommended antibiotic *e.g.* because it has been poorly tolerated by the patient in the past, he or she can easily select another option from the list displayed in the interface); (ii) provides explanations for physicians in the form of weighted antibiotic features: physicians can easily understand the reason why one antibiotic is preferred over others (*e.g.* because it has fewer adverse effects). The provision of recommendations that can be adjusted to any clinical situation [57], accompanied by convincing explanations, understandable by physicians, should improve physicians' knowledge about antibiotics [58], their critical analysis capacities [58], and their confidence in the CDSS [59], increasing the chances of its adoption.

Our approach is implemented in AntibioHelp®[23,60], a CDSS for antibiotic prescriptions. However, this approach could also be useful during the process of CPG formulation. Indeed, the preference model could be presented to experts during the writing of CPGs to help them to produce better guidelines and for the automatic detection of possible inconsistencies [19]. Experts could also visualize and compare antibiotics easily, according to their features weighted by degree of importance.

9. Conclusion

In conclusion, preference learning is a very promising approach for analyzing medical reasoning that can also be used for clinical decision support. In particular, it can combine patient data and treatment features to generate explanations. We implemented our approach in AntibioHelp®, which provides guideline recommendations and justifications, to help physicians to extrapolate the recommendations to situations for which no explicit recommendations exist. The feasibility of extrapolating the proposed method to explainable decision support in other medical domains, such as the treatment of chronic disorders (type 2 diabetes, hypertension, *etc.*) should be investigated.

CRediT authorship contribution statement

Jean-Baptiste Lamy: Methodology, Conceptualization, Software, Visualization, Writing - original draft, Writing - review & editing. **Karima Sedki:** Methodology, Conceptualization, Software, Writing - review & editing. **Rosy Tsopra:** Funding acquisition, Methodology, Conceptualization, Data curation, Visualization, Validation, Writing - review & editing. All authors agreed with all aspects of the study and the manuscript.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the French drug agency (ANSM, Agence Nationale de Sécurité du Médicament et des produits de santé) through the RaMiPA project (Raisonner pour Mieux Prescrire les Antibiotiques or Reasoning for a better antibiotic prescription), AAP 2016.

References

[1] O. Biran, C. Cotton, Explanation and justification in machine learning: A survey, in: Workshop on Explainable AI (XAI), 2017, pp. 8–13.

[2] F.K. Došilovic, M. Brcic, N. Hlupic, Explainable artificial intelligence: A survey, in: International convention on information and communication technology, electronics and microelectronics (MIPRO), 2018, pp. 0210–0215.

[3] J.M. Schoenborn, K.D. Althoff, Recent Trends in XAI: A Broad Overview on current Approaches, Methodologies and Interactions, in: Case-Based Reasoning for the Explanation of intelligent systems (XCBR) Workshop, 2019.

[4] W. Samek, T. Wiegand, K.R. Müller, Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models, Arxiv preprint arxiv:1708.08296.

[5] P. Kouki, J. Schaffer, J. Pujara, J. O'Donovan, L. Getoor, User preferences for hybrid explanations, Proceedings of the 11th ACM Conference on Recommender Systems (RecSys), Como, Italy, 2017, pp. 84–88.

[6] F. Gedikli, D. Jannach, M. Ge, How should I explain? A comparison of different explanation types for recommender systems, Int. J. Hum Comput Stud. 72 (4) (2014) 367–382.

[7] A. Shih, A. Choi, A. Darwiche, A symbolic approach to explaining bayesian network classifiers, Proceedings of the 37th International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 2018, pp. 5103–5111.

[8] H.C. Lane, M.G. Core, M. Van Lent, S. Solomon, D. Gomboc, Explainable artificial intelligence for training and tutoring, 2005.

[9] M. Van Lent, W. Fisher, M. Mancuso, An explainable artificial intelligence system for small-unit tactical behavior, Proceedings of the National Conference on Artificial Intelligence, 2004, pp. 900–907.

[10] M.G. Core, H.C. Lane, M. Van Lent, D. Gomboc, S. Solomon, M. Rosenberg, Building explainable artificial intelligence systems, in: AAAI, 2006, pp. 1766–1773.

[11] E. Tjoa, C. Guan, A Survey on Explainable Artificial Intelligence (XAI): Towards Medical XAI(2019) , arXiv: 1907.07374.

[12] J.B. Lamy, V. Ebrahiminia, C. Riou, B. Séroussi, J. Bouaud, C. Simon, S. Dubois, A. Butti, G. Simon, M. Favre, H. Falcoff, A. Venot, How to translate therapeutic recommendations in clinical practice guidelines into rules for critiquing physician prescriptions? Methods and application to five guidelines, BMC Med. Informatics Decis. Making 10 (2010) 31.

[13] A. Bussone, S. Stumpf, D. O'Sullivan, The role of explanations on trust and reliance in clinical decision support systems, 2015 International Conference on Healthcare Informatics, 2015, pp. 160–169.

[14] A. Rector, Axioms & templates: distinctions & transformationsamongst ontologies, frames & information models, Proceedings of the Seventh International Conference on Knowledge Capture, 2013, pp. 73–80.

[15] K. Bagherifard, M. Rahmani, M. Nilashi, V. Rafe, Performance improvement for recommender systems using ontology, Telematics Inform. 34 (8) (2017) 1772–1792.

[16] D. Werner, N. Silva, C. Cruz, A. Bertaux, An ontology-based recommender system using hierarchical multiclassification for economical e-news, in: Proceedings of the International Conference on Informatics in Economy (IE 2014), Bucharest, Romania, 2014.

[17] V. Subramaniyaswamy, V. Varadharajan, V. Indragandhi, A review of ontology-based tag recommendation approaches, Int. J. Intell. Syst. 28 (11) (2013) 1054–1071.

[18] K.A.P. Ngoc, Y.K. Lee, S.Y. Lee, OWL-based user preference and behavior routine ontology for ubiquitous system, 2005.

[19] R. Tsopra, J.B. Lamy, K. Sedki, Using preference learning for detecting inconsistencies in clinical practice guidelines: methods and application to antibiotherapy, Artif. Intell. Med. 89 (2018) 24–33.

[20] R. Tsopra, A. Venot, C. Duclos, An algorithm using twelve properties of antibiotics to find the recommended antibiotics, as in CPGs, in: AMIA Annu Symp Proc, vol. 1115-24, 2014.

[21] R. Tsopra, A. Venot, C. Duclos, Towards evidence-based CDSSs implementing the medical reasoning contained in CPGs: application to antibiotic prescription, Stud Health Technol Inform, 205 2014, pp. 13–17.

[22] J.B. Lamy, H. Berthelot, C. Capron, M. Favre, Rainbow boxes: a new technique for overlapping set visualization and two applications in the biomedical domain, J. Visual Lang. Comput. 43 (2017) 71–82.

[23] R. Tsopra, K. Sedki, M. Courtine, H. Falcoff, A. De Béco, R. Madar, F. Mechaï, J.B. Lamy, Helping GPs to extrapolate guideline recommendations to patients for whom there are no explicit recommendations, through the visualization of drug properties. The example of AntibioHelp® in bacterial diseases, J. Am. Med. Inform. Assoc. 26 (10) (2019) 1010–1019.

[24] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (2005) 734–749.

[25] E. Sezgin, S. Ozkan, A systematic literature review on health recommender systems, E-Health and Bioengineering Conference (EHB), 2013, pp. 1–4.

[26] Q. Han, I. Martinez de Rituerto de Troya, M. Ji, M. Gaur, L. Zejnilovic, A collaborative filtering recommender system in primary care: Towards a trusting patient-doctor relationship, in: IEEE International Conference on Healthcare Informatics (ICHI), 2018, pp. 377–379.

[27] J. Fürnkranz, E. Hüllermeier, Preference learning: An introduction, 2010.

[28] K.H. Tsai, T.K. Chiu, M.C. Lee, T.I. Wang, A learning objects recommendation model based on the preference and ontological approaches, in: Sixth International Conference on Advanced Learning Technologies, 2006. 2006, pp. 36–40.

[29] T.I. Wang, K.H. Tsai, M.C. Lee, T.K. Chiu, Personalized learning objects recommendation based on the semantic-aware discovery and the learner preference pattern, Educ. Technol. Soc. 10 (3) (2007) 84–105.

[30] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, P. Rodgers, The state-of-the-art of set visualization, Comput. Graphics Forum 35 (1) (2016) 234–260.

[31] J.B. Lamy, R. Tsopra, RainBio: Proportional visualization of large sets in biology, IEEE Trans. Visualisation Comput. Graphics accepted. doi:10.1109/TVCG.2019.2921544.

[32] J.B. Lamy, R. Tsopra, Translating visually the reasoning of a perceptron: the weighted rainbow boxes technique and an application in antibiotherapy, in: International Conference Information Visualisation (iV), London, United Kingdom, 2017, pp. 256–261.

[33] X.S. Yang, Nature-inspired Metaheuristic Algorithms, second ed., Luniver Press, 2010.

[34] J.B. Lamy, Advances in nature-inspired computing and applications, Springer, 2019, Ch. Artificial Feeding Birds (AFB): a new metaheuristic inspired by the behavior of pigeons, pp. 43–60.

[35] J.B. Lamy, Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies, Artif. Intell. Med. 80 (2017) 11–28.

[36] B. Motik, R. Shearer, I. Horrocks, Hypertableau reasoning for description logics, J. Artif. Intell. Res. 36 (2009) 165–228.

[37] J. Krishnakumar, R. Tsopra, What rationale do GPs use to choose a particular antibiotic for a specific clinical situation? BMC Family Pract. 20 (1) (2019) 178, https://doi.org/10.1186/s12875-019-1068-7.

[38] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.L. Patel-Schneider, The Description Logic Handbook: Theory, Implementation and Applications, Cambridge University Press, 2007.

[39] A.F. Tehrani, W. Cheng, E. Hullermeier, Preference learning using the Choquet integral: the case of multipartite ranking, Trans. Fuz Sys. 20 (6) (2012) 1102–1113.

[40] D. Bogdanov, M. Haro, F. Fuhrmann, A. Xambó, E. Gómez, P. Herrera, Semantic audio content-based music recommendation and visualization based on user preference examples, Inf. Process. Manage. 49 (1) (2013) 13–33.

[41] J.B. Lamy, B. Sekar, G. Guezennec, J. Bouaud, B. Séroussi, Explainable artificial intelligence for breast cancer: a visual case-based reasoning approach, Artif. Intell. Med. 94 (2019) 42–53.

[42] H. Ltifi, E. Ben Mohamed, M. Ben Ayed, Interactive visual knowledge discovery from data-based temporal decision support system, Inf. Visualization 15 (1) (2016) 31–50.

[43] Y. Shen, K. Yuan, D. Chen, J. Colloc, M. Yang, Y. Li, K. Lei, An ontology-driven clinical decision support system (IDDAP) for infectious disease diagnosis and antibiotic prescription, Artif. Intell. Med. 86 (2018) 20–32, https://doi.org/10.1016/j.artmed.2018.01.003.

[44] T.J. Bright, E. Yoko Furuya, G.J. Kuperman, J.J. Cimino, S. Bakken, Development and evaluation of an ontology for guiding appropriate antibiotic prescribing, J. Biomed. Inform. 45 (1) (2012) 120–128, https://doi.org/10.1016/j.jbi.2011.10.001.

[45] A. Moxey, J. Robertson, D. Newby, I. Hains, M. Williamson, S.A. Pearson, Computerized clinical decision support for prescribing: provision does not guarantee uptake, J. Am. Med. Inform. Assoc. 17 (1) (2010) 25–33, https://doi.org/10.1197/jamia.M3170.

[46] M. Rawson T, P. Moore L S, B. Hernandez, E. Charani, E. Castro-Sanchez, P. Herrero, B. Hayhoe, W. Hope, P. Georgiou, H. Holmes A, A systematic review of clinical decision support systems for antimicrobial management: are we failing to investigate these interventions appropriately? Clin. Microbiol. Infection: Off. Publ. Eur. Soc. Clin. Microbiol. Infect. Dis. 23 (8) (2017) 524–532, https://doi.org/10.1016/j.cmi.2017.02.028.

[47] M.T. Baysari, E.C. Lehnbom, L. Li, A. Hargreaves, R.O. Day, J.I. Westbrook, The effectiveness of information technology to improve antimicrobial prescribing in hospitals: a systematic review and meta-analysis, Int. J. Med. Inf. 92 (2016) 15–34, https://doi.org/10.1016/j.ijmedinf.2016.04.008.

[48] J. Holstiege, T. Mathes, D. Pieper, Effects of computer-aided clinical decision support systems in improving antibiotic prescribing by primary care providers: a systematic review, J. Am. Med. Inform. Assoc. 22 (1) (2015) 236–242, https://doi.org/10.1136/amiajnl-2014-002886.

[49] B. Cánovas-Segura, A. Morales, J.M. Juarez, M. Campos, F. Palacios, Impact of expert knowledge on the detection of patients at risk of antimicrobial therapy failure by clinical decision support systems, J. Biomed. Inform. 94 (2019) 103200, https://doi.org/10.1016/j.jbi.2019.103200.

[50] E. Shortliffe, A rule-based approach to the generation of advice and explanations in clinical medicine, 1977.

[51] K.A. Thursky, M. Mahemoff, User-centered design techniques for a computerised antibiotic decision support system in an intensive care unit, Int. J. Med. Inf. 76 (10) (2007) 760–768.

[52] E.I. Papageorgiou, J.D. Roo, C. Huszka, D. Colaert, Formalization of treatment guidelines using Fuzzy Cognitive Maps and semantic web tools, J. Biomed. Inform. 45 (1) (2012) 45–60, https://doi.org/10.1016/j.jbi.2011.08.018.

[53] E.I. Papageorgiou, Fuzzy cognitive map software tool for treatment management of uncomplicated urinary tract infection, Comput. Meth. Prog. Biomed. 105 (3) (2012) 233–245, https://doi.org/10.1016/j.cmpb.2011.09.006.

[54] P. Barrufet, J. Puyol-Gruart, C. Sierra, Terap-IA, a Knowledge-Based System for Pneumonia Treatment, Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems (EIS), 1998.

[55] L. Leibovici, M. Paul, A.D. Nielsen, E. Tacconelli, S. Andreassen, The TREAT project: decision support and prediction using causal probabilistic networks, Int. J. Antimicrob. Agents 30 (Suppl 1) (2007) S93–102.

[56] L. Pestotnik, C. Classen, S. Evans, P. Burke, Implementing antibiotic practice guidelines through computer-assisted decision support: clinical and financial outcomes, Ann. Intern. Med. 124 (10) (1996) 884–90.

[57] M. Lugtenberg, J.S. Burgers, C.F. Besters, D. Han, G.P. Westert, Perceived barriers to guideline adherence: a survey among general practitioners, BMC Family Pract. 12 (2011) 98, https://doi.org/10.1186/1471-2296-12-98.

[58] D. Shankar, B. Martins, W. Tu, K. Goldstein, A. Musen, Building an explanation function for a hypertension decision-support system, Stud. Health Technol. Inform. 84 (Pt 1) (2001) 538–42.

[59] L.R. Ye, The value of explanation in expert systems for auditing: An experimental investigation, Expert Syst. Appl. 9 (4) (1995) 543–556.

[60] R. Tsopra, J.P. Jais, A. Venot, C. Duclos, Comparison of two kinds of interface, based on guided navigation or usability principles, for improving the adoption of computerized decision support systems: application to the prescription of antibiotics, J. Am. Med. Inform. Assoc. 21 (e1) (2014) e107–e116.