Apprentissage de préférences à partir d'une ontologie formelle : méthodes et application en antibiothérapie

Jean-Baptiste Lamy, Karima Sedki, Rosy Tsopra

Apprentissage de préférences à partir d'une ontologie formelle : méthodes et application en antibiothérapie*

Jean-Baptiste Lamy[1], Karima Sedki[1], Rosy Tsopra[2]

[1] LIMICS, Université Paris 13, 93017 Bobigny, France, INSERM UMRS 1142, Sorbonne Universités, Paris
jean-baptiste.lamy@univ-paris13.fr, karima.sedki@univ-paris13.fr

[2] Université Paris 13, SMBH, Bobigny, France ; AP-HP, Assistance Publique des Hôpitaux de Paris, Paris, France
rosy.tsopra@nhs.net

Résumé : L'apprentissage des préférences est un problème de recherche qui a reçu beaucoup d'attention en intelligence artificielle ces dernières années. Il s'agit d'apprendre un modèle de préférences à partir de préférences observées. Ce modèle peut ensuite être utilisé pour obtenir une meilleure compréhension des préférences, et/ou pour effectuer des prédictions sur de nouvelles instances du problème. Les préférences sont généralement apprises à partir de données, par exemple sous la forme d'une matrice « instances × attributs ». Cependant, lorsque le domaine d'application est complexe, il peut être intéressant d'effectuer l'apprentissage à partir d'une ontologie formelle. Mais cela est plus difficile car (1) tous les attributs impactant les préférences n'ont pas nécessairement le même domaine, et (2) certains attributs peuvent correspondre à des propriétés *n*-aires (réifiées dans l'ontologie).

Dans cet article, nous proposons une méthode pour l'apprentissage d'un modèle de préférence à partir d'une ontologie, et prenant en compte notamment des attributs définis sur des domaines multiples, des propriétés *n*-aires, et des valeurs manquantes. Les préférences apprises peuvent alors venir enrichir l'ontologie. Elle « projette » tous les attributs sur une même classe d'individus, et ensuite des méthodes d'apprentissage de préférences plus classiques peuvent être employées. Nous avons appliqué cette méthode à une ontologie des antibiotiques. À partir de cette ontologie, nous avons explicité le modèle de préférence utilisé par les experts lorsqu'ils recommandent des antibiotiques dans les guides de bonnes pratiques, et en particulier quelle sont les propriétés importantes pour un antibiotique et comment les experts gèrent les informations manquantes.

Mots-clés : Ontologies, Préférences, Apprentissage, Antibiothérapie, Métaheuristique

1 Introduction

De nombreux consommateurs utilisent les sites de e-commerce pour acheter des livres, louer des DVD, choisir une voiture ou une recette de cuisine, *etc*. Afin de faciliter leur choix, des systèmes de recommandation (Adomavicius & Tuzhilin, 2005) sont apparus dans les sites de e-commerce. De manière similaire, dans de nombreux domaines professionnels, les systèmes de recommandation ont été proposés pour aider à la prise de décision, sur la base des préférences des experts. Ces systèmes ont aussi été appliqués en médecine (Sezgin & Ozkan, 2013), pour aider un patient à choisir un médecin (Han *et al.*, 2018) ou pour aider un médecin à choisir un traitement en s'appuyant sur les recommandations d'experts médicaux disponibles dans les guides de bonnes pratiques cliniques (Bouaud & Lamy, 2013). Ces systèmes nécessitent l'acquisition de préférences expertes ou utilisateurs. Celles-ci sont souvent coûteuses et longues à éliciter, notamment si le nombre d'instances (*e.g.* produits) est grand. Dans ce cas, il est intéressant d'apprendre les préférences à partir des données, car les données sont plus faciles à observer et à collecter : c'est l'*apprentissage des préférences* (Fürnkranz & Hüllermeier, 2010), qui est un domaine de recherche qui reçoit beaucoup d'attention ces derniers temps dans des disciplines telles que l'intelligence artificielle, l'aide à

⋆. Ce travail a été financé par l'ANSM (Agence Nationale de Sécurité du Médicament et des produits de santé) au travers du projet de recherche RaMiPA (AAP 2016).

la décision,... Il s'agit d'apprendre automatiquement un modèle de préférences à partir d'observations concrètes sur les préférences. Une fois le modèle appris, il peut être utilisé pour acquérir une meilleure compréhension du domaine et/ou pour aider à la décision.

Cependant, la qualité du modèle de préférences appris dépend du type de données fournies en entrée au système. De nombreuses approches d'apprentissage de préférences n'attachent que peu d'importance à la structure des données, et il s'agit souvent d'instances décrites par un ensemble d'attributs, par exemple dans une simple matrice « instance $\times$ attribut » à deux dimensions. L'utilisation d'ontologies pour structurer les données a été récemment proposée dans les systèmes de recommandations, et il a été montré que cela améliore les performances de ces systèmes (Bagherifard *et al.*, 2017; Werner *et al.*, 2014; Subramaniyaswamy *et al.*, 2013). Les ontologies ont aussi été utilisées pour structurer les modèles de préférences (Ngoc *et al.*, 2005). Cependant, l'apprentissage de préférences à partir d'ontologies est plus complexe, car tous les attributs impactant les préférences ne correspondent pas nécessairement à des propriétés définies sur le même domaine, et certains attributs peuvent correspondre à des propriétés *n*-aires, réifiées en plusieurs propriétés binaires dans l'ontologie.

Dans cet article, nous proposons une méthode pour apprendre un modèle de préférences à partir d'une ontologie, et prenant en compte les domaines multiples, les propriétés *n*-aires, et les valeurs manquantes. La méthode "projette" tous les attributs sur les individus d'une seule classe de l'ontologie, et ensuite les méthodes classiques d'apprentissage de préférences peuvent être employées. Nous décrivons aussi une méthode simple d'apprentissage de préférences, qui traduit l'apprentissage du modèle de préférences en un problème d'optimisation qui est résolu à l'aide d'une métaheuristique. Cependant, d'autres méthodes d'apprentissage pourraient être utilisées, et donc, la méthode que nous proposons pour appliquer l'apprentissage de préférences aux ontologies est générale, et non limitée à la méthode d'apprentissage présentée ici. Nous présentons également une étude de cas, dans laquelle nous utilisons la méthode proposée sur une ontologie des antibiotiques. Cette ontologie décrit les antibiotiques, les maladies infectieuses et les profils de patients concernés. Elle inclut aussi les recommandations de prescription issues de guides de bonnes pratiques cliniques et diverses propriétés liées aux antibiotiques et à leur utilisation (comme la présence d'effets indésirables importants, ou le risque de résistances bactériennes). A partir de cette ontologie, nous rendrons explicite le modèle de préférences implicite utilisé par les experts pour recommander un antibiotique lors de l'écriture des recommandations.

Dans un précédent travail (Tsopra *et al.*, 2018), nous avions utilisé l'apprentissage des préférences pour détecter des inconsistances dans les guides de bonnes pratiques en antibiothérapie. Ici, nous généraliserons la méthode d'apprentissage des préférences à partir d'ontologies, et nous l'étendrons pour gérer les difficultés mentionnées précédemment (domaines multiples, propriétés *n*-aires). De plus, la méthode publiée précédemment était limitée au classement bipartite (*i.e.* avec seulement deux niveaux de préférences, par exemple recommandé et non recommandé) et à des attributs Booléens. Nous généraliserons la méthode à plusieurs niveaux de préférences, et aux attributs catégoriels (ordonnés ou non). Nous améliorerons également le modèle de préférences et son apprentissage, de sorte à limiter les ambiguïtés dans le modèle et à rendre l'apprentissage plus reproductible. Enfin, nous nous intéresserons aux valeurs manquantes, et nous chercherons à comprendre comment elles influencent les préférences (par exemple comment un expert gère une valeur manquante).

La suite de l'article est organisée comme suit. La section 2 décrit la méthode proposée pour générer une matrice pour l'apprentissage des préférences, à partir d'une ontologie où les attributs d'intérêts sont définis sur plusieurs domaines, dont certains sont des propriétés n-aires, et qui comprend des valeurs manquantes. La section 3 détaille le modèle de préférences que nous proposons, la manière dont l'apprentissage de ce modèle peut se traduire en un problème d'optimisation, et comment nous avons résolu ce problème. La section 4 présente une étude de cas utilisant la méthode proposée sur une ontologie des antibiotiques. Enfin, la section 5 discute la méthode proposée avant de conclure en section 6.

2 Générer une matrice simple à partir d'une ontologie

Avant l'apprentissage des préférences proprement dit, il est nécessaire d'extraire à partir de l'ontologie un jeu de données structuré selon un format plus simple, c'est-à-dire une matrice « instance $\times$ attribut ». Cette tâche n'est pas triviale, car les attributs d'intérêt pour l'apprentissage des préférences ne sont pas nécessairement tous définis sur le même domaine, et certains de ces attributs peuvent être des propriétés n-aires (réifiées dans l'ontologie).

Considérons une ontologie $\mathcal{O}$ dont les axiomes décrivent un ensemble d'individus $\mathcal{I}$, un ensemble de classes $\mathcal{C}$ et un ensemble de propriétés $\mathcal{R}$. Une classe particulière d'individus $\mathcal{X} \equiv \{x_1, ...\} \subseteq \mathcal{I}$ représente les *instances* pour le besoin de l'apprentissage des préférences. De plus, un sous-ensemble des propriétés $\mathcal{F} = \{p_1, ...\} \subseteq \mathcal{R}$ représente les attributs pris en compte par l'apprentissage des préférences. Les valeurs des attributs peuvent être assertées à différents niveaux dans l'ontologie : sur une instance, *e.g.* $p(x, v)$ où v est la valeur, mais aussi sur une classe d'instance, *e.g.* $c \sqsubseteq \exists p.\{v\}$ avec $c \sqsubseteq \mathcal{X}$, et même sur un individu ou une classe non-instance du problème d'apprentissage de préférences, *e.g.* $p(i, v)$ avec $i \in \mathcal{I} \setminus \mathcal{X}$ et $c \sqsubseteq \exists p.\{v\}$ avec $c \in \mathcal{C} \sqcap \neg\mathcal{X}$. Les attributs peuvent donc avoir n'importe quel domaine (pas nécessairement les instances). Enfin, nous considérons un ensemble de formules de préférences $\mathcal{P}$ observées entre les instances, chaque formule définissant un ordre partiel sur $\mathcal{X}$.

Example 1 : Une ontologie pour un site de *e*-commerce contient les classes suivantes : $Produit$, $Fabricant$ et $Pays$. Les attributs sont $estEnPromotion$ (domaine : $Produit$, range : Booléen) et $aPourPays$ (domaine : $Fabricant$, range : $Pays$). Une propriété non-attribut (vis-à-vis de l'apprentissage de préférence) est $fabriquéPar$ (domaine : $Produit$, range : $Fabricant$). Les formules de préférences expriment les préférences observées sur différents clients, *e.g.* $x_1 \succ x_2 \approx x_3$ (*i.e.* x_1 est préféré à x_2 et x_3 ; mais x_2 et x_3 sont indifférents et aucun n'est préféré à l'autre) si un client a regardé les produits x_1, x_2 et x_3, et a finalement acheté le produit x_1. L'objectif est de comprendre les raisons pour lesquelles un produit est préféré à un autre, et de pouvoir suggérer d'autres produits aux clients, sur la base des deux attributs (promotion et pays de fabrication). Ici, les individus de la classe $Produit$ sont les instances pour l'apprentissage des préférences. Cependant, les deux attributs ne sont pas définis sur le même domaine, alors que l'apprentissage des préférences considère généralement que les attributs portent tous sur les instances.

L'ontologie peut être utilisée pour « projeter » chaque attribut sur les instances, afin de produire cette matrice à partir d'une ontologie complexe avec des attributs de domaines hétérogènes.

2.1 Attributs définis sur un domaine non-instance

Pour chaque attribut p dont le domaine n'est pas les instances du problème d'apprentissage de préférences (*i.e.* $\exists p.\top \not\sqsubseteq \mathcal{X}$), nous définissons une composition de propriétés $q \circ ... \circ p$ qui commence par q (dont le domaine est $\mathcal{X}$, les instances), et qui se termine par p. Nous créons une nouvelle propriété p' avec pour domaine $\mathcal{X}$ et le même range que p. Ensuite, pour chaque valeur possible v du range de p, nous créons une classe c_v équivalente à toutes les instances indirectement reliées à v, et nous assertons une relation directe selon p'. D'un point de vue formel, $c_v \equiv \mathcal{X} \sqcap q \circ ... \circ p.\{v\}$ et $c_v \sqsubseteq \exists p'.\{v\}$. De plus, les valeurs de l'attribut p peuvent être assertées au niveau des classes et non des individus. Pour chaque classe $Y \sqsubseteq \exists p.V$ (où V est la classe de valeur), nous créons une classe C_V avec $c_V \equiv \mathcal{X} \sqcap q \circ ... \circ p.V$ et $c_V \sqsubseteq \exists p'.V$.

Example 1 (suite) : Nous créons la propriété $aPourPaysDeFabrication$ (domaine : $Produit$, range : $Pays$). Ensuite, nous définissons la classe de tous les produits fabriqués par un fabricant en France et nous assertons qu'ils sont fabriqués en France :

$$\begin{aligned} ProduitFaitEnFrance &\equiv Produit \\ &\sqcap \exists fabriquéPar \circ aPourPays.\{France\} \\ ProduitFaitEnFrance &\sqsubseteq \exists aPourPaysDeFabrication.\{France\} \end{aligned}$$

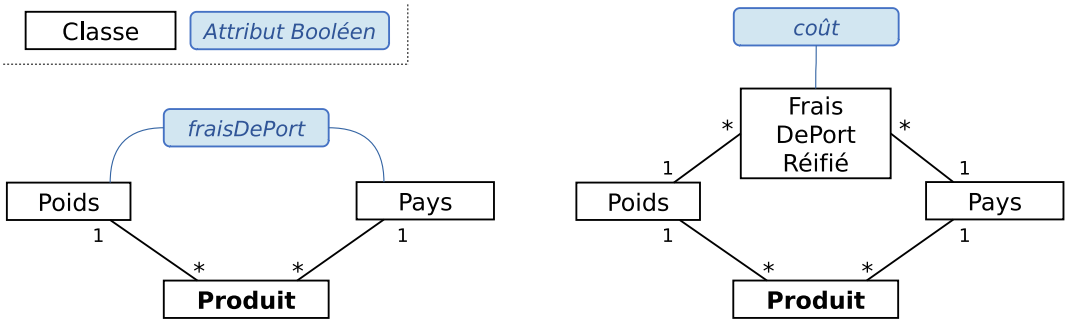

FIGURE 1 – Exemple d'attribut n-aire, avant réification (à gauche) et après (à droite).

2.2 Attributs n-aires

Pour chaque attribut n-aire, nous considérons sa réification en n propriétés binaires, p_1 à p_n. Nous distinguons l'une des entités reliées comme étant le range, c'est-à-dire la cible pour l'apprentissage des préférences, et les $n-1$ autres seront vues comme le domaine. Nous dénoterons arbitrairement le range avec l'index 1. Comme précédemment, nous créons une nouvelle propriété p', avec pour domaine $\mathcal{X}$ et le même range que p_1. Pour chaque valeur possible $(v_1, ..., v_n)$ pour les propriétés $(p_1, ..., p_n)$, nous créons une classe $c_{v_2,...,v_n}$, définie comme l'intersection des instances reliées à v_2,...,v_n *via* les propriétés p_2,...,p_n, et nous assertons une relation directe selon p'. D'un point de vue formel, $c_{v_2,...,v_n} \equiv \mathcal{X} \sqcap p_2.\{v_2\} \sqcap ... \sqcap p_n.\{v_n\}$ et $c_{v_2,...,v_n} \sqsubseteq \exists p'.\{v_1\}$. De plus, pour chaque classe $Y \sqsubseteq \exists p_1.V_1 \sqcap ... \sqcap p_n.V_n$, nous créons une classe $c_{V_2,...,V_n}$ avec $c_{V_2,...,V_n} \equiv \mathcal{X} \sqcap p_2.V_2 \sqcap ... \sqcap p_n.V_n$ et $c_{V_2,...,V_n} \sqsubseteq \exists p'.V_1$.

Exemple 2 : Nous pouvons étendre l'exemple précédent avec les frais de port, qui peuvent impacter la décision du client. Pour rester simple, nous considérerons seulement deux niveaux de frais : $Vrai$ (*i.e.* payant) et $Faux$ (frais de port gratuit). Les frais de port dépendent du poids du produit, avec deux poids possibles : $Léger$ et $Lourd$, et le pays de fabrication (en considérant que le fabricant envoie directement le produit au client). Les frais de port sont donc un attribut ternaire entre le coût, le poids et le pays (Figure 1). Le coût est le range pour l'apprentissage des préférences, tandis que le poids et le pays sont des domaines. Par exemple, la relation $fraisDePort(Faux, Léger, France)$ signifie que les produits légers fabriqués en France sont envoyés gratuitement. Cette relation ternaire peut être « projetée » sur les produits comme suit :

$$\begin{aligned} ProduitLégerFrançais &\equiv Produit \\ &\sqcap \exists aPourPoids.\{Léger\} \\ &\sqcap \exists aPourPaysDeFabrication.\{France\} \\ ProduitLégerFrançais &\sqsubseteq \exists aPourFraisDePort.\{Faux\} \end{aligned}$$

Quand le nombre d'instances de l'attribut n-aire est élevé, la création manuelle de ces classes peut être longue. Cependant, cela peut être automatisé avec un programme qui « pré-processe » l'ontologie. Nous avons utilisé des scripts Python avec Owlready 2 (Lamy JB, 2016, 2017), un module Open Source pour la programmation orientée ontologie.

Les deux difficultés traitées ci-dessus (domaines multiples et attributs n-aires) peuvent être rencontrées sur le même attribut (*i.e.* un attribut n-aire dont le domaine n'est pas les instances de l'apprentissage de préférences). Dans ce cas, les deux solutions proposées peuvent être utilisées conjointement.

Enfin, à l'aide d'un raisonneur comme HermiT (Motik *et al.*, 2009), nous reclassons les instances selon les classes définies par intention (c_v, c_V, $c_{v_2,...,v_n}$ et $c_{V_2,...,V_n}$), et nous obtenons donc pour chaque instance les valeurs des propriétés p'. Pour une instance et un attribut donné, si aucune valeur n'est obtenue, cela correspond à une valeur manquante (*e.g.* si le pays

de fabrication est inconnu). Si plus d'une valeur est obtenue, cela correspond à des valeurs conflictuelles. Dans ce cas, en fonction de l'attribut, il est possible de : (a) garder la plus mauvaise valeur, si les valeurs sont ordonnées, ou (b) conserver toutes les valeurs conflictuelles. La méthode d'apprentissage de préférences que nous décrirons par la suite autorise les deux options.

3 Apprentissage des préférences

L'apprentissage des préférences est réalisé sur les instances $\mathcal{X}$ et les attributs $\mathcal{F}' = \{p'_1, ...\}$, où p'_i sont les nouvelles propriétés créées à la section précédente, ou à défaut $p'_i = p_i$ pour les attributs binaires qui ont pour domaine les instances (pour lesquels aucune nouvelle propriété n'est nécessaire). Pour un attribut p', nous notons $\mathcal{V}_{p'}$ l'ensemble des valeurs possibles.

3.1 Modèle de préférences

De nombreuses méthodes d'apprentissage de préférences ont été proposées dans la littérature (Koriche & Zanuttini, 2010; Dekel *et al.*, 2003; Burges *et al.*, 2005). Celle que nous présentons ici vise à apprendre à la fois :

— **des contraintes nécessaires simples** $\mathcal{N}$, de la forme $p' = v$,
— **des préférences** $\mathcal{W}$, exprimées par des poids associés à chaque valeur possible de chaque attribut.

Les contraintes correspondent à des critères que doivent obligatoirement satisfaire les instances : par exemple, seuls les produits disponibles dans le pays du client peuvent être proposés à la vente. Au contraire, les préférences quantifient l'importance des attributs et de leurs valeurs. Nous pouvons donc formaliser notre modèle de préférences par un couple $\mathcal{M} = (\mathcal{N}, \mathcal{W})$ où $\mathcal{N}$ est un sous-ensemble de l'ensemble des contraintes possibles et $\mathcal{W}$ est une liste de poids, avec un poids pour chaque valeur possible pour chaque attribut :

$$\mathcal{M} = \begin{cases} \mathcal{N} \subseteq \{p' = v : \forall p' \in \mathcal{F}', \forall v \in \mathcal{V}_{p'}\} \\ \mathcal{W} = (w_{p',v} : \forall p' \in \mathcal{F}', \forall v \in \mathcal{V}_{p'}) \end{cases}$$

Notons que, si la contrainte $p' = v$ est présente dans $\mathcal{N}$, tous les poids $w_{p',v'}$ pour la propriété p' ne seront pas utilisés par la suite. Cependant, nous laissons ces poids présents dans le modèle pour faciliter l'apprentissage, car l'apprentissage des contraintes et des poids se fait simultanément.

Les valeurs manquantes sont associées à un poids de 0. Il s'agit d'une « origine arbitraire » pour la mesure des poids : comme les poids n'ont pas d'unité ni d'origine, nous pouvons définir cette origine à notre guise. Pour les valeurs conflictuelles, la somme des poids des différentes valeurs est considérée.

3.2 Réduction à un problème d'optimisation

Tout d'abord, pour les instances satisfaisant les contraintes nécessaires, nous définissons une fonction d'utilité u qui calcule son utilité. Les instances avec une plus grande utilité sont préférées aux autres. Nous avons choisi ici une fonction u très simple qui calcule la somme des poids pour chaque valeur d'attribut associée à l'instance :

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

Ensuite, nous définissons la fonction E (Algorithm 1) qui calcule le taux d'erreur obtenu lorsque l'on compare le modèle $\mathcal{M}$ aux formules de préférences observées $\mathcal{P}$ sur les instances $\mathcal{X}$. E calcule d'abord l'ordre total $\mathcal{T}$ sur $\mathcal{X}$, en utilisant le modèle : les instances qui satisfont les contraintes nécessaires sont préférées aux autres, et, parmi celles qui satisfont les

Algorithme 1 Algorithme de la fonction E qui calcule le taux d'erreur du modèle.

fonction $E(\mathcal{X}, \mathcal{P}, \mathcal{M})$:
soit $\mathcal{N}$ et $\mathcal{W}$ les deux composantes du modèle : $\mathcal{M} = (\mathcal{N}, \mathcal{W})$
soit $\mathcal{X}_{\mathcal{N}} = \{x \in \mathcal{X} : x \text{ satisfait les contraintes nécessaires } \mathcal{N}\}$
soit $\mathcal{X}_{\overline{\mathcal{N}}} = \mathcal{X} \setminus \mathcal{X}_{\mathcal{N}}$
soit $e = 0$ le nombre d'erreurs trouvés
soit $\mathcal{T}$ un ordre total sur $\mathcal{X}$, défini comme suit :
$\forall x_i \in \mathcal{X}_{\mathcal{N}}, \forall x_j \in \mathcal{X}_{\mathcal{N}}, u(x_i) > u(x_j) \Leftrightarrow x_i \succ x_j$
$\forall x_i \in \mathcal{X}_{\mathcal{N}}, \forall x_j \in \mathcal{X}_{\overline{\mathcal{N}}},\ x_i \succ x_j$
$\forall x_i \in \mathcal{X}_{\overline{\mathcal{N}}}, \forall x_j \in \mathcal{X}_{\overline{\mathcal{N}}}, x_i \approx x_j$
pour chaque instance $x \in \mathcal{X}$:
pour chaque formule de préférence $p \in \mathcal{P}$ impliquant x :
si p n'est pas compatible avec l'ordre total $\mathcal{T}$:
$e = e + 1$
break
retourne $\frac{e}{|\mathcal{X}|}$

contraintes, celles qui ont une plus grande utilité sont préférées. Enfin, la fonction compare $\mathcal{T}$ avec les formules de préférences pour chaque instance, et retourne le taux d'erreur E.

Afin d'augmenter la clarté du modèle (pour un humain), la capacité à le traduire visuellement par la suite, et la reproductibilité de l'apprentissage, nous avons ajouté deux contraintes au problème d'optimisation, détaillées dans les deux paragraphes suivants.

Premièrement, nous avons utilisé des valeurs entières pour les poids. Lors d'un précédent travail (Tsopra *et al.*, 2018), nous avions utilisé des poids réels. Cependant, cela conduisait à une certaine ambiguïté dans le modèle. Par exemple un poids appris de 0,71 pour un attribut doit-il être considéré comme significativement différent d'un poids de 0,73 pour un autre attribut ? Ou bien ces deux poids doivent-ils être considérés comme équivalents ? Dans la mesure où l'apprentissage n'est pas reproductible, une autre exécution peut éventuellement aboutir à des poids différents pour ces deux propriétés. Afin d'éviter ces ambiguïtés, dans le présent travail nous avons restreint les poids à des valeurs entières. De plus, afin de faciliter l'affichage graphique par la suite, nous avons interdit les valeurs 1 et -1 pour les poids. En effet, si l'on traduit visuellement les poids par la hauteur d'un rectangle, les valeurs trop petites conduiront à des rectangles de faible hauteur, dans lesquels il sera difficile d'écrire le nom de l'attribut correspondant. Par conséquent, $w_{p',v} \in \{..., -4, -3, -2, 0, 2, 3, 4...\}$.

Deuxièmement, lors de l'apprentissage, nous cherchons à minimiser le taux d'erreur, mais aussi la somme totale des poids, notée S_w. Notons que, si la contrainte $p' = v$ est présente dans $\mathcal{N}$, tous les poids $w_{p',v'}$ n'ont aucun impact. Par conséquent, nous calculons la somme des poids en excluant ceux qui, du fait des contraintes nécessaires, n'ont aucun impact, de la manière suivante :

$$S_w = \sum_{p'} \sum_{v'} \{w_{p',v} : \nexists v' \, such \, as \, (p' = v') \in \mathcal{N}\}$$

Minimiser la somme des poids présente plusieurs intérêts. Tout d'abord, en multipliant tous les poids par une valeur constante n, il est possible d'obtenir un modèle différent mais conduisant au même taux d'erreur. Ici, nous évitons ce problème et augmentons donc la reproductibilité de l'apprentissage. De plus, comme les poids ne sont pas comptés lorsqu'il existe une contrainte nécessaire, cela biaise l'apprentissage en faveur des contraintes : si deux modèles ont le même taux d'erreur mais que l'un contient davantage de contraintes nécessaires, il sera préféré. Cela est intéressant car les contraintes sont plus faciles que les poids à appréhender pour un humain. Enfin, si les poids sont présentés visuellement, par exemple par des hauteurs, cela permet de réduire l'espace occupé par la visualisation et de la rendre plus compacte.

Par conséquent, nous cherchons le modèle $\mathcal{M}^{best}$ qui minimise à la fois le taux d'erreur E et la somme des poids S_w, selon un ordre lexicographique (*i.e.* nous minimisons le taux

Algorithme 2 Algorithme pour les fonctions vol et $marche$ utilisées pour l'optimisation.

fonction $vol()$:
  soit $\mathcal{N}$ un ensemble de contraintes aléatoires
  soit $\mathcal{W} = \big(w_{p',v} =$ nombre entier différent de -1 et $1 : \forall p' \in \mathcal{F}', \forall v \in \mathcal{V}_{p'}\big)$
  retourne $\mathcal{M} = (\mathcal{N}, \mathcal{W})$
fonction $marche(\mathcal{M})$:
  soit $\mathcal{M}' = (\mathcal{N}', \mathcal{W}')$ une copie de $\mathcal{M}$
  soit r un nombre réel aléatoire entre 0 et 1
  si $r < 0.15$:
    ajouter une contrainte aléatoire à $\mathcal{N}'$
  sinon si $r < 0.3$:
    enlever une contrainte aléatoire de $\mathcal{N}'$
  sinon :
    modifier un poids de $\mathcal{W}'$ en utilisant la fonction de $marche$ proposée
    pour résoudre les problèmes d'optimisation global non-linéaire
    dans la métaheuristique AFB (Lamy JB, 2019)
  retourne $\mathcal{M}'$

d'erreur et, à taux d'erreur égal, nous préférons le modèle ayant la somme des poids la plus faible). Il s'agit d'un problème d'optimisation :

$$\mathcal{M}^{best} = \arg\min_{\mathcal{M}} \big(E(\mathcal{X}, \mathcal{P}, \mathcal{M} = (\mathcal{N}, \mathcal{W})), S_w\big)$$

3.3 Résolution du problème d'optimisation

Le problème d'optimisation décrit à la section précédente est complexe car il mélange optimisation combinatoire (pour optimiser $\mathcal{N}$) et optimisation globale non-linéaire (pour optimiser $\mathcal{W}$). Cependant, $\mathcal{N}$ et $\mathcal{W}$ doivent être optimisés simultanément, puisqu'ils sont interdépendants.

Nous avons résolu ce problème à l'aide de la métaheuristique des Oiseaux Picorant Artificial (OPA, *Artificial Feeding Birds* AFB) (Lamy JB, 2019), inspirée par le comportement des pigeons. Cette métaheuristique est très générique et peut résoudre n'importe quel problème d'optimisation défini par un triplet de trois fonctions $(coût, vol, marche)$, où $coût$ est la fonction de coût à minimiser, vol est une fonction qui retourne une solution aléatoire et $marche$ est une fonction qui modifie légèrement une solution existante. L'algorithme 2 décrit les fonctions vol et $marche$ que nous proposons pour optimiser le modèle de préférences $\mathcal{M}$.

4 Application à une ontologie de l'antibiothérapie

4.1 Contexte

Pour aider les médecins à prescrire le bon antibiotique, les autorités de santé publient des guides de bonnes pratiques cliniques. Dans ces guides, les experts recommandent la prescription de certains antibiotiques, selon les propriétés de chaque médicament, la maladie infectieuse et le profil du patient (adulte, enfant, femme enceinte,...). Par exemple, ils recommandent la fosfomycine-trométamol pour une cystite non compliquée chez la femme. Dans cette étude, nous cherchons à établir le modèle de préférences implicites qu'utilisent les experts pour choisir un antibiotique, lorsque la prescription d'antibiotique est justifiée.

Au cours de travaux précédents (Tsopra *et al.*, 2014a,b), nous avons construit une base de connaissances décrivant 50 antibiotiques dans 66 situations cliniques, selon 11 attributs utilisés par les experts pour établir les recommandations (Table 1). Chaque attribut est Booléen, et sa valeur dépend de l'antibiotique mais également du profil patient, de la maladie infectieuse

#	**Attribut [**$nom\ court$**]** Définition
1	**Naturellement actif contre la bactérie pathogène [**$naturallement\,actif$**]** Est-ce que la bactérie pathogène est décrite comme naturellement sensible à l'antibiotique ? (*e.g.* l'amoxicilline est naturellement active contre les streptocoques du groupe A)
2	**Probablement actif contre la bactérie pathogène [**$probablement\,actif$**]** Est-ce que la fréquence de résistance de la bactérie pathogène pour l'antibiotique est décrite comme suffisamment basse pour permettre sa prescription ? (*e.g.* la ceftriaxone est probablement active contre E.coli)
3	**Efficacité clinique prouvée [**$prouvée$**]** Est-ce que l'antibiotique est décrit comme cliniquement efficace pour traiter l'infection OU est (ou a été) indiqué ou recommandé pour cette infection ?
4	**Absence de contre-indication [**$non\,contre\,indiqué$**]** Est-ce qu'il n'y a pas de contre-indication absolue à la prise de l'antibiotique pour le profil de patient concerné ?
5	**Protocole pratique [**$protocole$**]** Est ce que le protocole d'administration de l'antibiotique est décrit comme pratique pour le patient, c'est-à-dire administration par voie orale ET durée de traitement courte ?
6	**Classe non-précieuse [**$non\,précieux$**]** Est-ce que l'antibiotique appartient à une classe décrite comme « non précieuse », c'est-à-dire qui ne doit pas être préservée pour des infections plus graves ?
7	**Absence d'effets indésirables graves ou fréquents [**$peu\,d'effet\,ind$**]** Est-ce que l'antibiotique est décrit comme ne causant aucun effet indésirable grave et peu d'effets indésirables non-graves ?
8	**Haut niveau d'efficacité [**$très\,efficace$**]** Est-ce que l'antibiotique est décrit comme très efficace pour ce type d'infection ?
9	**Spectre antibactérien étroit [**$spectre$**]** Est-ce que l'antibiotique est décrit comme ayant un spectre antibactérien étroit ?
10	**Faible impact écologique sur les résistances bactériennes [**$risque\,éco\,bas$**]** Est-ce que l'antibiotique est décrit comme ayant un risque faible de provoquer l'émergence de nouvelle résistance bactérienne ?
11	**Goût [**$goût$**]** Est-ce que l'antibiotique a un goût acceptable pour le profil de patient (enfants notamment) ?

TABLE 1 – Les 11 attributs décrivant les antibiotiques dans l'ontologie.

(*e.g.* cystite) et/ou de la bactérie pathogène (*e.g.* Escherichia coli). Ses attributs peuvent donc être des attributs n-aires. La base de connaissances a été construite et peuplée par un médecin (RT) à partir de plusieurs guides de bonnes pratiques cliniques, et ensuite validée par un panel d'experts en antibiothérapie au cours d'un processus Delphi.

Cette base de connaissances a ensuite été formalisée dans une ontologie OWL 2.0 (Tsopra *et al.*, 2018). Elle contient 144,038 triplets RDF décrivant 5.696 classes, 19 propriétés et 34.483 axiomes, et appartenant à la famille $\mathcal{ALC}(\mathcal{D})$ [1] des logiques de description. La Fi-

1. $\mathcal{AL}$: langage d'attributs (incluant la négation atomique, l'intersection de concepts, la restriction univer-

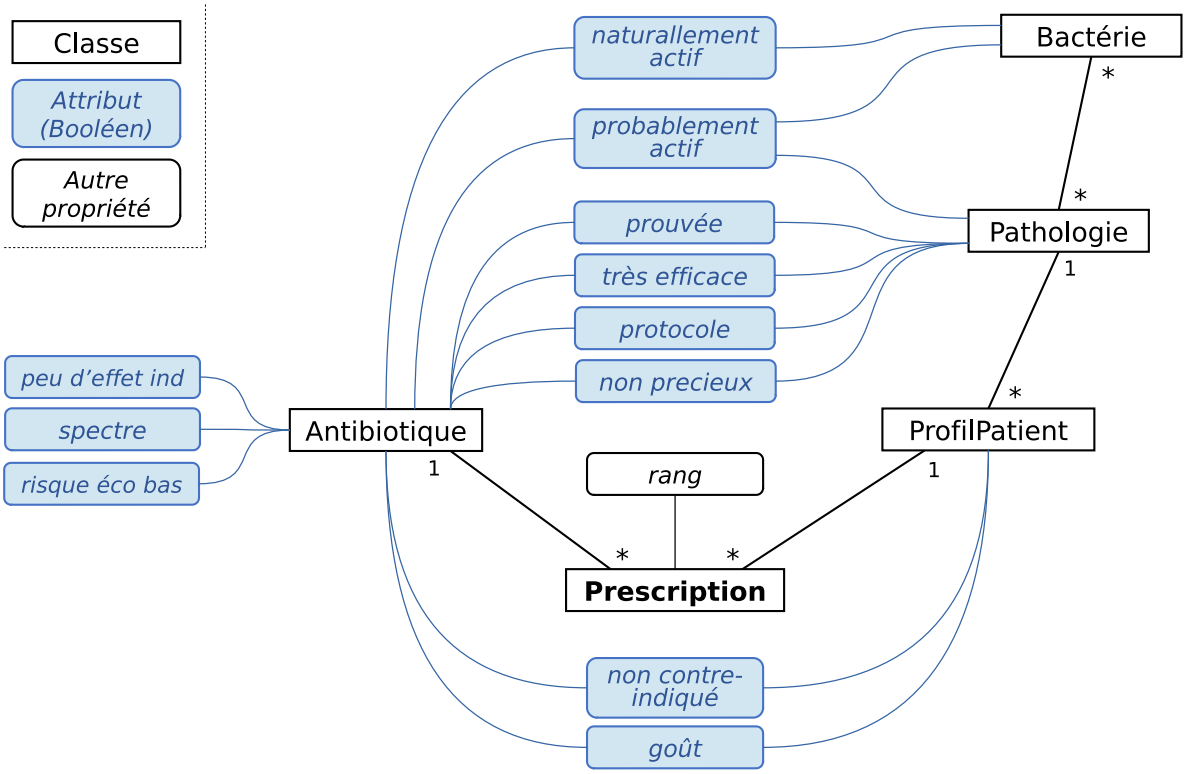

FIGURE 2 – Structure générale de l'ontologie de l'antibiothérapie. Les attributs ternaires et quaternaires (tous sauf les trois sur la gauche) sont réifiés dans l'ontologie.

gure 2 montre la structure générale de l'ontologie. Elle comprend cinq classes principales : $Antibiotique$, $ProfilPatient$ (*e.g.* adulte, enfant ou femme enceinte) associés à une maladie infectieuse ($Pathologie$) causée par une $Bactérie$ pathogène. Une $Prescription$ est l'association d'une $Antibiotique$ avec un $ProfilPatient$ (lui-même relié à une $Pathologie$ et une $Bactérie$). Les 11 attributs sont définis sur 5 domaines différents (aucun d'entre eux n'étant $Prescription$), et incluent 3 propriétés binaires, 7 ternaires et 1 quaternaire. De nombreuses valeurs manquantes sont présentes, à cause de la persistance importante d'information inconnue dans le domaine médical. De plus, pour les $Prescription$ recommandées dans les guides, le rang de recommandation est présent (1 à 4, le rang 1 étant prescrit de préférence au rang 2, *etc*).

Toutes les combinaisons médicaments-situations cliniques ont été créées dans l'ontologie (soit $66 \times 50 = 3300$), sous forme de classes OWL. Celles-ci ont ensuite été reclassées à l'aide du raisonneur HermiT.

4.2 Application de la méthode

Nous avons appliqué la méthode proposée à cette ontologie. $Prescription$ correspond à la classe des instances pour l'apprentissage des préférences. Les rangs de recommandations ont été traduits en formules de préférences $\mathcal{P}$; par exemple si un guide recommande la prescription d'un antibiotique A en rang 1 et la prescription de B ou C en rang 2 pour un $ProfilPatient$ donné, cela a été traduit par la formule de préférence $A \succ B \approx C \succ D \approx E \approx ...$ (où $D, E, ...$ sont les prescriptions correspondant à tous les autres antibiotiques disponibles mais non recommandés par le guide). La Figure 3 montre la structure générale de la matrice obtenue à partir de l'ontologie. Les rangs de recommandations correspondent aux

selle et les qualifieurs existentiels limités à la classe Thing), $\mathcal{C}$: négation complexe, $(\mathcal{D})$: type de données (Baader *et al.*, 2007).

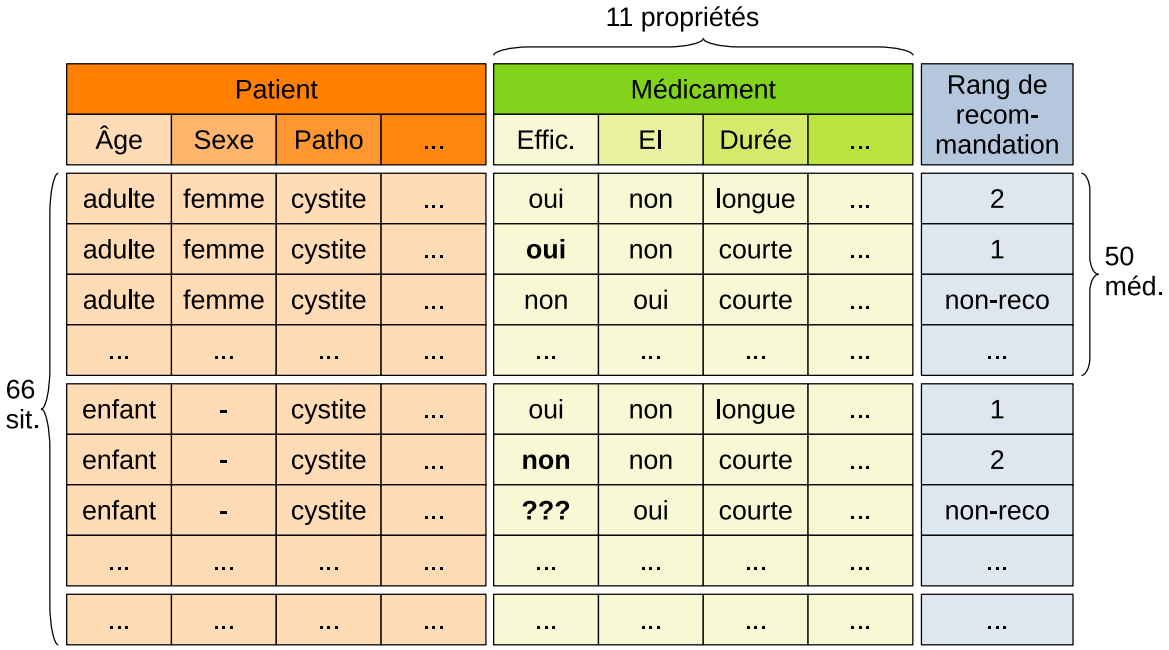

FIGURE 3 – Structure générale de la matrice « instances × attributs » obtenue (EI : effets indésirables).

recommandations trouvées dans les guides de bonnes pratiques cliniques. Pour une situation clinique donnée, la partie « patient » (en orange) reste la même, tandis que la partie « médicament » en vert varie selon les 50 médicaments. Cette partie « médicament » dépend du médicament mais aussi de la partie patient (certaines propriétés du médicament change selon le patient pour lequel il est prescrit, par exemple la durée du traitement n'est pas forcément la même pour un enfant et un adulte).

Lorsque des valeurs conflictuelles ont été rencontrées (*e.g.* une $Pathologie$ associée à deux $Bacteries$ pathogènes putatives, l'une étant résistante à l'antibiotique et l'autre ne l'étant pas), nous avons retenu la pire valeur (principe de précaution, ici la valeur $Faux$). De plus, tous les attributs de l'ontologie correspondent à des avantages potentiels d'un antibiotique (*e.g.* peu d'effets indésirables, grande efficacité, *etc.*). Par conséquent, nous avons restreint les poids associés à la valeur $Faux$ à des nombres négatifs (ou nul), et ceux associés à la valeur $Vrai$ à des nombres positifs (ou nul ; *i.e.* $w_{p',False} \leq 0$ et $w_{p',True} \geq 0$). Ceci interdit d'apprendre un modèle médicalement absurde, par exemple dans lequel les antibiotiques avec beaucoup d'effets indésirables seraient préférés à ceux en ayant peu.

La méthode a été implémentée en Python et l'ontologie a été manipulée avec Owlready (Lamy JB, 2016, 2017). Nous avons utilisé les valeurs par défaut des paramètres pour la métaheuristique AFB (valeurs figurant dans (Lamy JB, 2019)). Nous avons exécuté la métaheuristique sur 3,000 itérations et nous avons retenu le meilleur modèle trouvé.

4.3 Évaluation du processus d'apprentissage

Afin d'évaluer le processus d'apprentissage et la capacité à généraliser les résultats au-delà des données d'apprentissage, nous avons effectué une validation croisée par dixième (*10-fold cross-validation*). La base de données a été découpée en 10 sous-ensembles aléatoires. Pour chaque sous-ensemble, nous avons effectué l'apprentissage sur les 9 autres sous-ensembles, et utilisé le dixième sous-ensemble comme jeu de test. Le taux d'erreur moyen est de 3.5% sur le jeu d'apprentissage et de 5.2% sur le jeu de test. Le taux d'erreur restant faible sur le jeu de test, cela suggère que le modèle de préférences est suffisamment générique pour pouvoir être appliqué à des situations qui ne figurent pas dans le jeu d'apprentissage.

$$\begin{array}{lcc} & naturallement\,actif & = Vrai \\ \wedge & probablement\,actif & = Vrai \\ \wedge & prouvée & = Vrai \\ \wedge & non\,contre\,indicaqué & = Vrai \end{array}$$

	Attribut	$w_{p',Faux}$	$w_{p',Vrai}$
5	$protocole$	-7	4
6	$non\,précieux$	-2	2
7	$peu\,d'effet\,ind$	-5	2
8	$très\,efficace$	-2	2
9	$spectre$	-2	2
10	$risque\,éco\,bas$	-3	2
11	$goût$	-3	2

TABLE 2 – Les contraintes nécessaires $\mathcal{N}$ (en haut) et les poids de préférence $\mathcal{W}$ appris (en bas ; les poids ne figurent pas lorsqu'ils n'ont aucun impact, du fait des contraintes nécessaires).

4.4 Résultats

La Table 2 montre le meilleur modèle obtenu, après apprentissage sur l'ensemble de la base de connaissances. Celui-ci conduit à un taux d'erreur de 3,5%. Le modèle de préférences montre l'importance de chaque attribut pour le choix d'un antibiotique à prescrire.

Les contraintes apprises montrent que les prescriptions doivent avoir nécessairement la valeur $Vrai$ pour quatre attributs ($naturallement\,actif$, $probablement\,actif$, $prouvée$, $non\,contre\,indiqué$) pour être recommandées. Ceci est pertinent d'un point de vue clinique car seuls les antibiotiques actifs d'un point de vue microbiologique et clinique doivent être prescrits, et les contre-indications doivent être respectées. En effet, les attributs impliqués dans les contraintes nécessaires sont ceux considérés comme les plus importants par le médecin de notre équipe (RT). En particulier, trois d'entre eux ($naturallement\,actif$, $prouvée$, $probablement\,actif$) sont liés à l'efficacité du traitement et le troisième ($non\,contre\,indiqué$) à la sécurité.

Parmi les attributs de préférence, l'attribut $protocole$ semble avoir des poids plus importants que les autres. Ceci est logique car le protocole détermine la facilité de prise du médicament, et donc l'observance du patient : les antibiotiques trop compliqués à prendre sont oubliés par le patient. En particulier, si le traitement est trop long, le patient interrompt souvent celui-ci, ce qui conduit à des rechutes et peut favoriser l'émergence de résistances bactériennes.

Ce modèle de préférences permet aussi de comprendre comment les experts traitent les informations manquantes, pour chaque attribut. Les valeurs manquantes correspondent à un poids de 0 (origine arbitraire). Par conséquent, la position du 0 par rapport aux deux bornes w_{Faux} et w_{Vrai} indique si les valeurs manquantes sont plutôt traitées comme des $Faux$ ou des $Vrai$. Ici, dans la Table 2, nous pouvons constater que, pour l'attribut $peu\,d'effet\,ind$, la valeur absolue de $w_{peu\,d'effet\,ind,Faux}$ est beaucoup plus grande que celle de $w_{peu\,d'effet\,ind,Vrai}$. Cela signifie que les valeurs manquantes sont traitées comme étant plus proches de $Vrai$ que de $Faux$, c'est-à-dire que, lorsque l'expert ne sait pas si un antibiotique présente ou non des effets indésirables graves ou fréquents, il aura plutôt tendance à faire comme s'il n'y avait pas de tels effets.

Globalement, nous constatons dans la Table 2 que les valeurs absolues des poids associés aux valeurs $Faux$ sont toujours supérieures ou égales aux poids associés aux valeurs $Vrai$. Les experts interprètent donc les valeurs manquantes comme plutôt « en faveur » du médicament. Cela est particulièrement vrai pour les effets indésirables ($peu\,d'effet\,ind$, comme

vu au paragraphe précédent) et la facilité de prise ($protocole$). En revanche, pour le spectre ($spectre$) et l'efficacité ($très\ efficace$), les valeurs manquantes sont traitées par les experts comme un « entre-deux » situé à mi-chemin entre $Vrai$ et $Faux$.

5 Discussion

Dans cet article, nous avons proposé une méthode générale pour l'apprentissage de préférences à partir d'une ontologie formelle. La méthode permet l'apprentissage de préférences à partir d'attributs qui peuvent être définis sur des domaines hétérogènes, ainsi que d'attributs n-aires. Elle permet aussi d'étudier la manière dont les valeurs manquantes sont traitées. Nous avons aussi décrit un modèle de préférences et une méthode d'apprentissage, fondée sur l'optimisation. Cependant, d'autres méthodes d'apprentissage pourraient être utilisées, telles que l'intégrale de Choquet (Tehrani *et al.*, 2012). Nous avons illustré notre approche sur une ontologie de l'antibiothérapie, et nous avons présenté le modèle de préférences résultant. Comparé à nos travaux antérieurs (Tsopra *et al.*, 2018), ce modèle permet une meilleure compréhension des critères utilisés par les experts pour recommander les antibiotiques, et en particulier de la manière dont ils gèrent les valeurs manquantes. Le modèle présente également moins d'ambiguïtés liées aux valeurs de poids très proches. Les contraintes et les poids appris peuvent ensuite venir enrichir l'ontologie, par exemple sous forme d'annotations associées aux différentes propriétés.

Dans la section 2, notre méthode requiert la création de nombreuses classes et propriétés pour prendre en compte les domaines multiples et les attributs n-aires. Nous avons suggéré l'utilisation d'un langage de programmation pour "prétraiter" l'ontologie et créer ces classes et propriétés. La principale limite de cette approche est qu'elle ne prend en compte que les faits assertés, mais pas les faits qui pourraient être inférés. Une autre solution aurait été l'utilisation de règles fondées sur la logique du premier ordre. Si nous reprenons l'exemple 2 (les frais de port), la règle suivante aurait pu être utilisée :

$$\begin{aligned}\forall(r,w,c,p), & FraisDePortRéifié(r) \wedge aPourValeur(r,Faux)\\ & \wedge Poids(w) \wedge aPourPoids(r,w)\\ & \wedge Pays(c) \wedge aPourPays(r,c)\\ & \wedge Produit(p) \wedge aPourPoids(p,w) \wedge aPourPaysDeFabrication(p,c)\\ & \Rightarrow aPourFraisDePort(p,Faux)\end{aligned}$$

Cette règle est plus générique que la classe $ProduitLégerFrançais$ que nous avons proposé dans l'exemple 2, car elle couvre tous les produits avec des frais de port gratuit, et pas seulement les produits français légers. Cependant, ces règles ont une limitation importante : elles ne fonctionnent que sur les individus, mais pas sur les classes (*e.g.* dans la règle précédente, r, w, c et p doivent être des individus, et non des classes). Par conséquent, l'approche « prétraitement » est préférable si les attributs sont assertés au niveau des classes (sous forme de restrictions), tandis que la logique du premier ordre est préférable si les faits inférés doivent être considérés. Ici, l'ontologie de l'antibiothérapie que nous avons utilisée inclut de nombreux attributs dont les valeurs sont assertés au niveau des classes. Par exemple, toutes les pénicillines A sont contre-indiquées chez les patients allergiques à l'amoxicilline, où « pénicillines A » est une classe d'antibiotiques. C'est pourquoi nous avons choisi l'approche « prétraitement ».

Au contraire, les logiques de description ne peuvent pas être utilisées pour définir des classes générales ciblant chaque valeur d'un attribut n-aire, parce que les logiques de description sont sans variable (*variable-free*) et correspondent en fait à des formules avec une seule variable libre (Baader *et al.*, 2007). Dans l'exemple précédent, il serait tentant de considérer la classe $ProduitÀFraisDePortGratuit$ (une classe correspondant à la règle précédente en logique du premier ordre, plus générale que $ProduitLégerFrançais$), et de la définir de la

sorte :

$$
\begin{aligned}
ProduitÀFraisDePortGratuit \;\equiv\; & Produit \\
\sqcap\; & \exists aPourPoids \\
& \quad \circ aPourFraisDePortRéifié \\
& \quad \circ\, aPourValeur.\{Faux\} \\
\sqcap\; & \exists haPourPaysDeFabrication \\
& \quad \circ aPourFraisDePortRéifié \\
& \quad \circ\, aPourValeur.\{Faux\} \\
ProduitÀFraisDePortGratuit \;\sqsubseteq\; & \exists aPourFraisDePort.\{Faux\}
\end{aligned}
$$

Cependant, cette définition ne fonctionnera pas comme souhaité. Plus précisément, les deux $FraisDePortRéifié$ (celui relié au $Poids$ et celui relié au $Pays$) ne sont pas nécessairement les mêmes. Corriger cela demanderait une seconde variable libre (pour asserter que les deux $FraisDePortRéifié$ sont les mêmes), et cela n'est pas possible en logiques de description.

Dans la littérature, l'apprentissage des préférences a rarement été appliqué aux ontologies. Tsai *et al.* (Tsai *et al.*, 2006) et Wang *et al.* (Wang *et al.*, 2007) ont proposé un modèle d'apprentissage fondé sur une approche ontologique pour les systèmes de e-learning. Il s'appuie sur des cours décrits en SCORM (*Sharable Content Object reference Model*). L'ontologie permet d'inférer les prérequis pour chaque cours, et d'hériter les valeurs des attributs à partir de leur classe. Contrairement au travail que nous avons présenté ici, les auteurs n'ont pas pris en compte les attributs définis sur des domaines différents ni les attributs n-aires. Cependant, l'utilisation d'une sémantique plus riche offerte par l'ontologie formelle peut être utile pour l'apprentissage de préférences dans des domaines complexes, comme l'antibiothérapie.

Nous avons utilisé la métaheuristique AFB, à cause de sa généricité et sa capacité à résoudre des problèmes mélangeant optimisation globale non-linéaire et optimisation combinatoire. En particulier, et contrairement aux autres métaheuristiques existantes (Yang XS, 2010), l'AFB ne nécessite pas le calcul de distance entre solutions du problème. Cela rend l'AFB plus facile à adapter à des problèmes nouveaux pour lesquels le calcul d'une telle distance n'est pas trivial. De plus, nos travaux précédents (Tsopra *et al.*, 2018) ont montré que l'AFB était plus performant pour l'apprentissage de préférences dans ce contexte, comparé à d'autres métaheuristiques.

Nous avons présenté une étude de cas en antibiothérapie. À notre connaissance, nos travaux sont les premiers à essayer d'appliquer l'apprentissage des préférences aux guides de bonnes pratiques cliniques. Dans ce domaine, les valeurs de certains attributs (tels que les résistances des bactéries aux antibiotiques) évoluent rapidement. La méthode que nous proposons est automatique, et donc le modèle de préférences peut être mis à jour facilement lorsque l'ontologie est modifiée.

Les experts qui écrivent les guides s'appuient sur leurs connaissances propres, qui sont essentiellement tacites (Smith MK, 2003). La présentation du modèle de préférences explicite à ces experts pourrait leur permettre de mieux appréhender leurs connaissances et leur processus de raisonnement, et de produire des recommandations plus fiables et plus consistantes.

Dans le domaine thérapeutique, très peu d'approches fondées sur les préférences ont été proposées. La majorité des système d'aide à la décision clinique (Bouaud & Lamy, 2013) s'appuie sur l'implémentation des recommandations des guides et sur les contre-indications des médicaments, *e.g.* sous forme de systèmes à base de règles. Cependant, ils ne cherchent pas à construire un modèle de préférences ni à comprendre les raisons sous-jacentes aux recommandations, comme nous l'avons fait ici. Un système d'aide à la décision s'appuyant sur un modèle de préférences serait donc une approche radicalement nouvelle en médecine. Un tel système permettrait en particulier d'effectuer des prédictions sans se limiter aux recommandations présentes dans les guides. En effet, nous avons vu (section 4.3) que le modèle pouvait être généralisé au-delà des données d'apprentissage. Il pourrait donc servir à produire des recommandations lorsque les recommandations sont absentes (souvent les guides

ne donnent pas des recommandations pour tous les cas possibles (Bouaud *et al.*, 2009)) ou lorsqu'elles ne peuvent pas être appliquées (par exemple à cause de contre-indications ou d'allergie).

6 Conclusion

Nous avons proposé une méthode générale pour l'apprentissage de préférences à partir d'une ontologie formelle, prenant en compte les domaines hétérogènes, les attributs n-aires et les valeurs manquantes. Nous avons montré comment cette approche permettait de mieux comprendre le raisonnement des experts en infectiologie. Les perspectives de ce travail incluent l'application de la méthode à d'autres ontologies dans le domaine médical ou au-delà, ainsi que la visualisation du modèle de préférences appris en antibiothérapie et son utilisation pour l'aide à la décision thérapeutique auprès des médecins généralistes (Tsopra *et al.*, 2019).

Références

ADOMAVICIUS G. & TUZHILIN A. (2005). Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *IEEE Transactions of Knowledge and Data Engineering*, **17**, 734–749.

BAADER F., CALVANESE D., MCGUINNESS D. L., NARDI D. & PATEL-SCHNEIDER P. L. (2007). *The description logic handbook : theory, implementation and applications*. Cambridge University Press.

BAGHERIFARD K., RAHMANI M., NILASHI M. & RAFE V. (2017). Performance improvement for recommender systems using ontology. *Telematics and Informatics*, **34**(8), 1772–1792.

BOUAUD J. & LAMY J. B. (2013). *Yearbook of medical informatics*, volume 8, chapter A medical informatics perspective on clinical decision support systems. Findings from the yearbook 2013 section on decision support, p. 128–31.

BOUAUD J., SÉROUSSI B., FALCOFF H., JULIEN J., SIMON C. & DENKÉ D L. (2009). Consequences of the verification of completeness in clinical practice guideline modeling : a theoretical and empirical study with hypertension. *AMIA Symposium*, **2009**, 60–4.

BURGES C. J. C., SHAKED T., RENSHAW E., LAZIER A., DEEDS M., HAMILTON N. & HULLENDER G. N. (2005). Learning to rank using gradient descent. In *ICML*, volume 119 of *ACM International Conference Proceeding Series*, p. 89–96 : ACM.

DEKEL O., MANNING C. D. & SINGER Y. (2003). Log-linear models for label ranking. In *NIPS*, p. 497–504 : MIT Press.

FÜRNKRANZ J. & HÜLLERMEIER E. (2010). *Preference learning : An introduction*.

HAN Q., MARTINEZ DE RITUERTO DE TROYA I., JI M., GAUR M. & ZEJNILOVIC L. (2018). A collaborative filtering recommender system in primary care : Towards a trusting patient-doctor relationship. In *IEEE International Conference on Healthcare Informatics (ICHI)*, p. 377–379.

KORICHE F. & ZANUTTINI B. (2010). Learning conditional preference networks. *Artificial intelligence*, **174**(11), 685–703.

LAMY JB (2016). Ontology-Oriented Programming for Biomedical Informatics. *Studies in health technology and informatics (STC)*, **221**, 64–68.

LAMY JB (2017). Owlready : Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. *Artif Intell Med*, **80**, 11–28.

LAMY JB (2019). *Advances in nature-inspired computing and applications*, chapter Artificial Feeding Birds (AFB) : a new metaheuristic inspired by the behavior of pigeons, p. 43–60. Springer.

MOTIK B., SHEARER R. & HORROCKS I. (2009). Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research*, **36**, 165–228.

NGOC K. A. P., LEE Y. K. & LEE S. Y. (2005). OWL-based user preference and behavior routine ontology for ubiquitous system.

SEZGIN E. & OZKAN S. (2013). A systematic literature review on health recommender systems. In *E-Health and Bioengineering Conference (EHB)*, p. 1–4.

SMITH MK (2003). Michael Polanyi and tacit knowledge. *The encyclopedia of informal education*.

SUBRAMANIYASWAMY V., VARADHARAJAN V. & INDRAGANDHI V. (2013). A review of ontology-based tag recommendation approaches. *International Journal of Intelligent Systems*, **28**(11), 1054–1071.

TEHRANI A. F., CHENG W. & HULLERMEIER E. (2012). Preference learning using the Choquet integral : The case of multipartite ranking.

TSAI K. H., CHIU T. K., LEE M. C. & WANG T. I. (2006). A learning objects recommendation model based on the preference and ontological approaches.

TSOPRA R., LAMY J. B. & SEDKI K. (2018). Using preference learning for detecting inconsistencies in clinical practice guidelines : methods and application to antibiotherapy. *Artif Intell Med*, **89**, 24–33.

TSOPRA R., SEDKI K., COURTINE M., FALCOFF H., DE BÉCO A., MADAR R., MECHAÏ F. & LAMY J. B. (2019). 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*, **accepted**.

TSOPRA R., VENOT A. & DUCLOS C. (2014a). An algorithm using twelve properties of antibiotics to find the recommended antibiotics, as in CPGs . In *AMIA Annu Symp Proc*, volume 1115-24.

TSOPRA R., VENOT A. & DUCLOS C. (2014b). Towards evidence-based CDSSs implementing the medical reasoning contained in CPGs : application to antibiotic prescription. In *Stud Health Technol Inform*, volume 205, p. 13–7.

WANG T. I., TSAI K. H., LEE M. C. & CHIU T. K. (2007). Personalized learning objects recommendation based on the semantic-aware discovery and the learner preference pattern. *Educational technology & society*, **10**(3), 84–105.

WERNER D., SILVA N., CRUZ C. & BERTAUX A. (2014). 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.

YANG XS (2010). *Nature-inspired metaheuristic algorithms (second edition)*. Luniver Press.