

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

Jean-Baptiste Lamy, Karima Sedki, Rosy Tsopra

RaMiPA

Projet ANSM AAP 2016
Coordinatrice : Rosy Tsopra



LIMICS
Université Paris 13
Sorbonne Paris Cité
93017 Bobigny
Sorbonne Universités
INSERM UMRS 1142

Introduction

■ Apprentissage de préférences

◆ Exemple : système de recommandation des sites de e-commerce

- Les produits sont décrits par des propriétés (prix, catégories, provenance,...)
- Objectif : prédire quels produits préférerait l'utilisateur

◆ Apprendre un modèle de préférences

- L'apprentissage se fait à partir de préférences observées
 - L'utilisateur A a regardé les produits X, Y et Z et a acheté Z
- Modèle qualitatif : des règles de préférence
 - Exemple : les utilisateurs préfèrent les produits français aux produits pas chers
- **Modèle quantitatif** : des scores sont associés aux propriétés
 - Exemple : la provenance à un score de 3, le prix de 2

◆ Applicable à l'aide à la prescription médicale : choix d'un médicament par un médecin

Introduction

■ Apprentissage de préférences

- ◆ L'apprentissage se fait souvent à partir d'une matrice de la forme « instance x attribut »
- ◆ Dans des domaines complexes comme la médecine, il est nécessaire d'utiliser des données plus structurées, sous forme d'ontologie
 - les valeurs des propriétés peuvent être définies sur des classes
 - les attributs impactant les préférences n'ont pas le même domaine
 - présence de propriétés n-aires (réifiées)
 - présence de valeurs manquantes
- ◆ => **Objectif : proposer une méthode pour apprendre un modèle de préférences à partir d'une ontologie**
- ◆ Application en antibiothérapie

Méthode générale

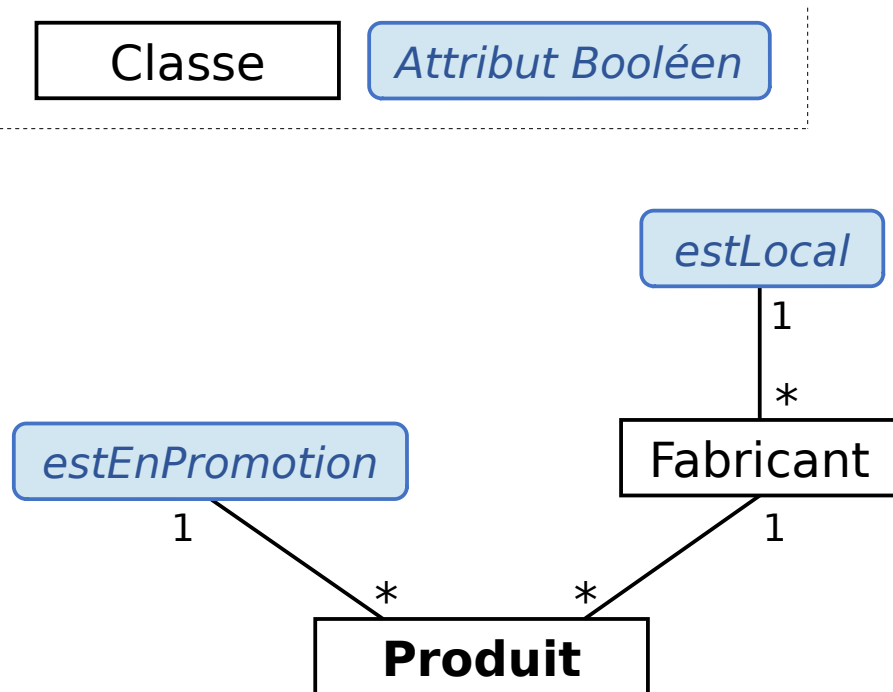
- 1) générer une matrice « instance x attribut » à partir de l'ontologie
 - ◆ Gérer l'hétérogénéité des domaines
 - ◆ Gérer les propriétés n-aires
- 2) réduire l'apprentissage des préférences à un problème d'optimisation
- 3) résoudre ce problème d'optimisation

Définitions

- **Soit une ontologie O dont les axiomes décrivent :**
 - ◆ un ensemble d'individus I
 - ◆ un ensemble de classes C
 - ◆ un ensemble de propriétés R
- **Une classe d'individu $X \subseteq I$ représente les instances pour l'apprentissage des préférences**
- **Un ensemble $F \subseteq R$ représente les propriétés à considérer pour l'apprentissage des préférences**
- **Un ensemble P contient les formules de préférences observées**
 - ◆ Chaque formule est un ordre partiel sur X
 - ◆ Exemple de formule : $x_1 = x_2 > x_3$
- **Intérêt de l'ontologie :** définir les valeurs des propriétés aux niveaux des classes :
 - ◆ Le produit x_1 est en promotion
 - ◆ vs tous les produits du fabricant f sont en promotion

Domaines hétérogènes

■ Exemple : $X = \text{Produit}$, $R = \{\text{estEnPromotion}, \text{estLocal}\}$



Création d'une propriété :
« **estProduitLocal** » (**Produit** $\Rightarrow$ booléen)

Création de la classe **ProduitLocal :**

$\text{ProduitLocal} \equiv \text{Produit}$

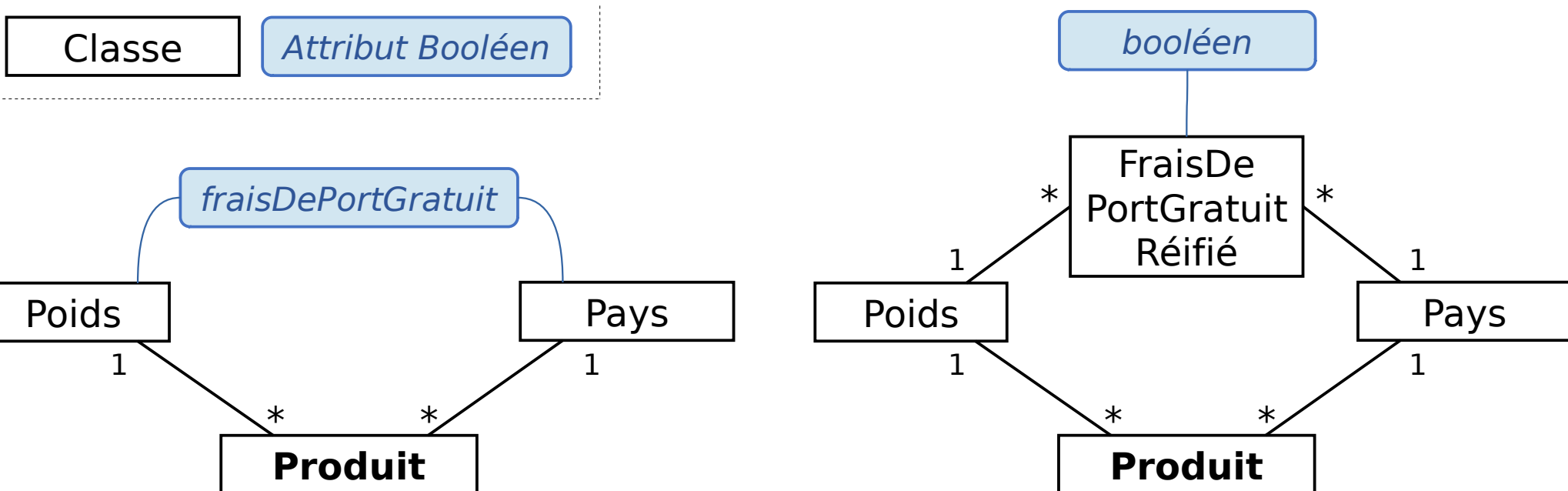
$\sqcap \exists \text{fabriquéPar} \circ \text{estLocal}.\{\text{True}\}$

$\text{ProduitLocal} \sqsubseteq \exists \text{estProduitLocal}.\{\text{True}\}$

Propriétés n-aires

■ Exemple : $X = \text{Produit}$, $R = \{ \text{fraisDePortGratuit} \}$

- ◆ Les frais de port dépendent du poids du produit et de son pays d'origine
- ◆ `fraisDePortGratuit` est une propriété ternaire reliant Poids, Pays et booléen



Propriétés n-aires

Classe

Attribut Booléen

fraisDePortGratuit

Poids

1

Pays

1

Produit

*

*

booléen

FraisDe
PortGratuit
Réifié

*

*

Poids

1

1

Pays

1

1

Produit

*

*

Création d'une propriété : «aPourFraisDePortGratuit» (Produit => booléen)

Création de la classe ProduitLégerFrançais :

ProduitLégerFrançais $\equiv$ *Produit*

$\sqcap$ $\exists aPourPoids.\{Léger\}$

$\sqcap$ $\exists aPourPaysDeFabrication.\{France\}$

ProduitLégerFrançais $\sqsubseteq$ $\exists aPourFraisDePortGratuit.\{Faux\}$

etc.

Automatisation

- **La création manuelle des propriétés et des classes peut être fastidieuse**
 - ◆ => automatisation à l'aide de script Python avec Owlready2
 - ◆ => utilisateur du raisonneur HermiT pour reclasser les instances dans les nouvelles classes
- **Owlready2 :** [Artificial Intelligence in Medicine 2017]
 - ◆ Un module Python pour accéder aux ontologies de manière transparente : « Programmation orientée ontologie »
 - ◆ Les classes de l'ontologie sont des classes Python
 - ◆ Quadstore / base de données SQL optimisé et testé jusqu'à 1 milliard de triplets RDF
 - ◆ Traduit les restrictions SOME et ONLY en propriétés de classe
 - ◆ Intègre les raisonneurs HermiT et Pellet
 - ◆ Accès à UMLS pour les terminologies médicales

Owlready 2

Java + OWLAPI :

```
OWLDataProperty priceProperty = df.getOWLDataProperty(IRI.create("file.owl#has_price"));  
float price = ((Float) drug.getPropertyValue(priceProperty)).floatValue();
```

Python + Owlready2 :

```
drug.price
```



Modèle de préférences

■ **Nombreux modèles existants (qualitatif, quantitatif)**

■ **Modèle retenu (quantitatif) : $M = (N, W)$**

◆ Des contraintes nécessaires N

● De la forme « propriété = valeur »

◆ Des poids $w_{p,v}$, un pour chaque valeur de chaque propriété

● Choix d'une origine arbitraire : **valeur 0** pour les valeurs manquantes

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

Ensemble des propriétés

Ensemble des valeurs
possibles pour la propriété p

Modèle de préférences

■ La fonction d'utilité $u()$ calcule un « score » pour chaque instance

◆ On préfère les instances avec un score plus élevé

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

◆ Exemple d'un modèle sur 3 propriétés :

$$\mathcal{N} = \{p_{portGratis} = oui\}$$

$$\mathcal{W} = \{w_{promo,non} = -1, w_{promo,oui} = 2, w_{local,non} = -2, w_{local,oui} = 3\}$$

	portGratis	promo	local	
A	oui	non	oui	$\Rightarrow u(x) = -1+3 = 2$
B	oui	oui	oui	$\Rightarrow u(x) = 2+3 = 5$
C	oui	oui	non	$\Rightarrow u(x) = 2-2 = 0$
D	non	oui	oui	éliminé par la contrainte « portGratis = oui »

} **B > A > C > D**

Apprentissage des préférences

■ Le problème d'apprentissage du modèle peut se réduire à un problème d'optimisation

- ◆ Rechercher les contraintes N et les poids W qui minimisent le taux d'erreur E lorsque l'on compare les scores obtenus avec les formules de préférences observées P

Ordre total obtenu avec les scores $u()$: $B > A > C > D$

Formule de préférences observées P : $B > A, A = C, D > C$

On accepte que le modèle de préférence soit plus précis que P

2 formules satisfaites sur 3 $\Rightarrow E = 33\%$

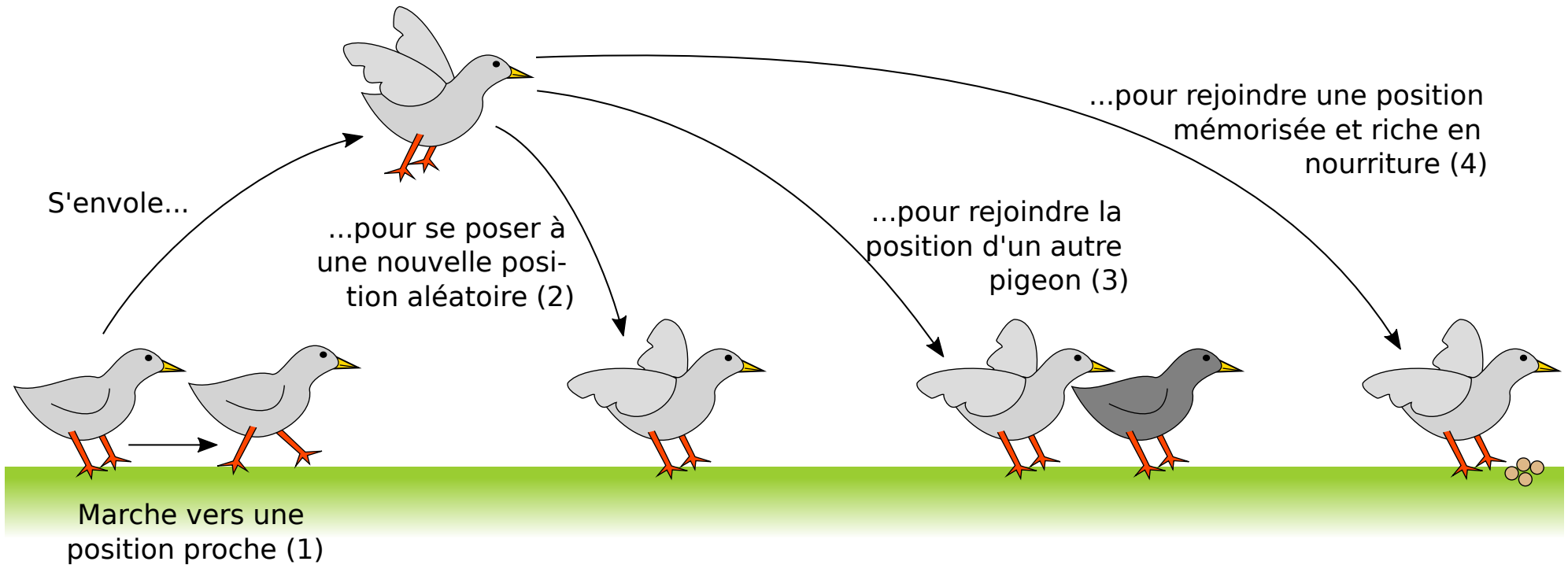
- ◆ Optimisation mixte : combinatoire (pour N) et numériques (pour W)
- ◆ Très peu d'algorithmes d'optimisation peuvent gérer ce type de problèmes
- ◆ C'est le cas de la métaheuristique des Oiseaux artificiels picorant (*Artificial Feeding Bird*)

Métaheuristique

Oiseaux artificiels picorant (Artificial Feeding Birds, AFB)

Inspiré du comportement du pigeon

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



- Simplicité
- Performance
- **Généricité** : optimisation combinatoire, optimisation global non-linéaire,...

Un problème d'optimisation = (*coût()*, *vol()*, *marche()*)

Apprentissage des préférences

function $E(\mathcal{X}, \mathcal{M})$:

let $\mathcal{N}$ and $\mathcal{W}$ be the two parts of the model $\mathcal{M} = (\mathcal{N}, \mathcal{W})$

let $\mathcal{X}_{\mathcal{N}} = \{x \in \mathcal{X} : x \text{ satisfies the necessary constraints } \mathcal{N}\}$

let $\mathcal{X}_{\overline{\mathcal{N}}} = \mathcal{X} \setminus \mathcal{X}_{\mathcal{N}}$

let $e = 0$ be the number of errors found

let $\mathcal{T}$ be a total order on $\mathcal{X}$, defined as follows:

$\forall x_i \in \mathcal{X}_{\mathcal{N}}, \forall x_j \in \mathcal{X}_{\mathcal{N}}, u(x_i) > u(x_j) \Leftrightarrow x_i \succ x_j$

$\forall x_i \in \mathcal{X}_{\mathcal{N}}, \forall x_j \in \mathcal{X}_{\overline{\mathcal{N}}}, x_i \succ x_j$

$\forall x_i \in \mathcal{X}_{\overline{\mathcal{N}}}, \forall x_j \in \mathcal{X}_{\overline{\mathcal{N}}}, x_i \approx x_j$

for each instance $x \in \mathcal{X}$:

for each preference formula p involving x :

if p is not compatible with the total order $\mathcal{T}$:

$e = e + 1$

break

return $\frac{e}{|\mathcal{X}|}$

Application en antibiothérapie

■ Aide à la décision thérapeutique

- ◆ Aider le médecin à mieux prescrire
- ◆ Particulièrement complexe en antibiothérapie en médecine générale
 - Problème des résistances bactériennes
 - Prescription « probabiliste » : sans connaître la bactérie causale

■ Les guides de bonnes pratiques donnent des recommandations de traitement mais ne les justifient que rarement

- ◆ => Utiliser l'apprentissage de préférences pour découvrir le modèle de préférences utilisé par les experts qui ont rédigé les guides

Matériel :

l'ontologie des antibiotiques

■ Décrit les propriétés des antibiotiques

◆ Les propriétés peuvent dépendre :

- de l'antibiotique
- du profil patient (enfant,...)
- de la pathologie infectieuse
- et/ou de la bactérie causale

=> des propriétés binaires,
ternaires et quaternaires

■ Contenu :

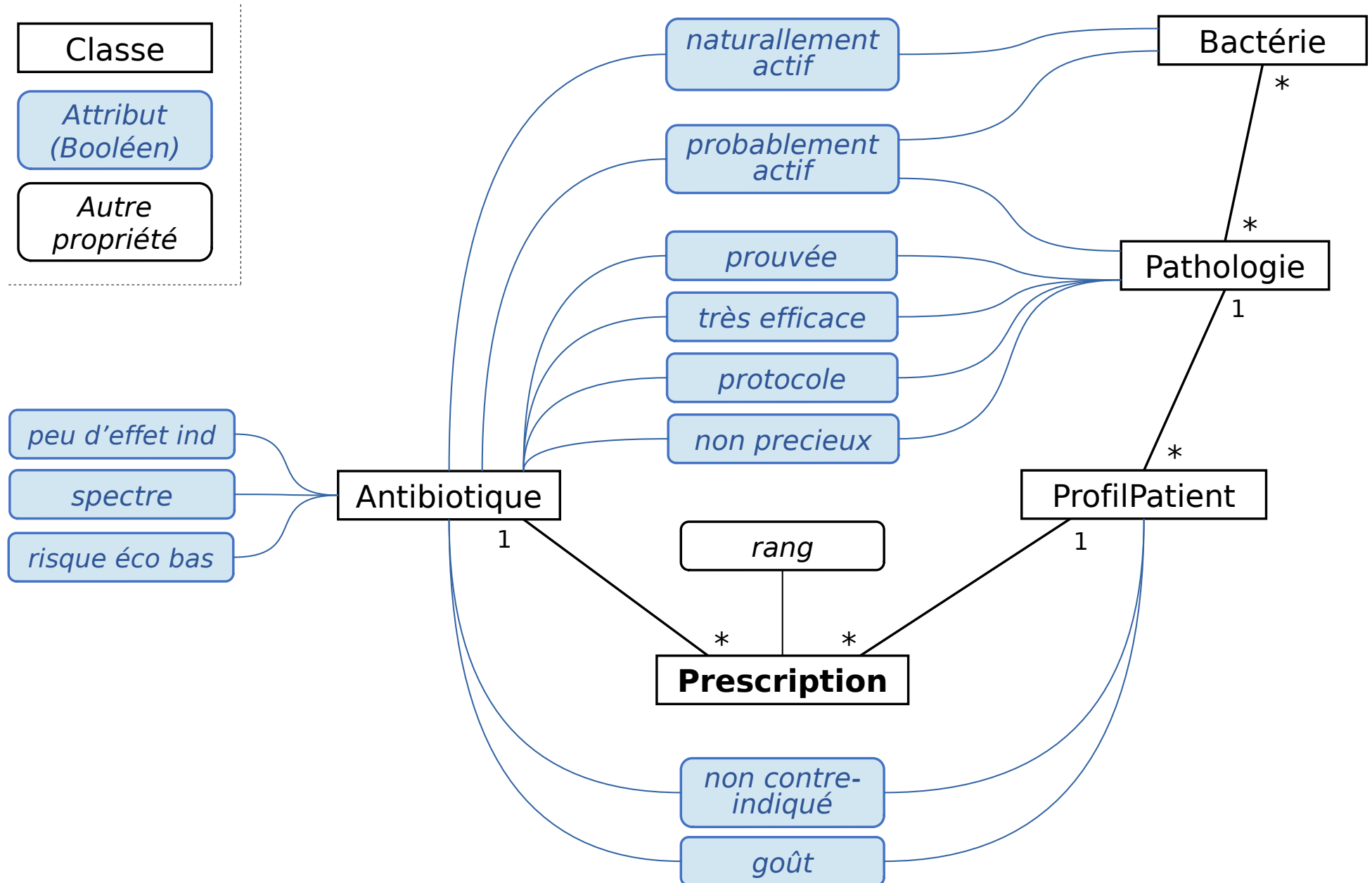
- ◆ 11 propriétés
- ◆ 50 antibiotiques
- ◆ 66 situations cliniques (patient + pathologie)
 - Nombreuses valeurs manquantes / inconnues
- ◆ + le niveau de recommandation dans les guides (1ère intention, 2ème,...)

■ Format

- ◆ À l'origine : 24 Fichiers Excel reliés entre eux... :-)
- ◆ => ontologie OWL (Famille logique ALC(D), 144.038 triplets RDF)

Matériel :

l'ontologie des antibiotiques



Extraction d'un tableau

■ L'ontologie contient...

- ◆ ...des propriétés n-aires (réifiées)
- ◆ ...des propriétés définies sur différents domaines (Antibiotique, Patient, Pathologie, Bactérie...)

■ => Projeter toutes les valeurs des propriétés sur la classe Prescription

- ◆ Script Owlready
- ◆ Raisonneur HermiT pour reclasser les Prescriptions et projeter les valeurs des propriétés

HighCostPrescription $\equiv$ *Prescription*
 $\sqcap \exists \text{hasDrug} \circ \text{highCost}.\{\text{True}\}$
HighCostPrescription $\sqsubseteq \exists \text{hasHighCostDrug}.\{\text{True}\}$

PrescriptionOfDrugAToPregnantPatient
 $\equiv$ *Prescription*
 $\sqcap \exists \text{hasDrug}.\{\text{drugA}\}$
 $\sqcap \exists \text{hasPatientProfile}.\{\text{pregnantPatientProfile}\}$
PrescriptionOfDrugAToPregnantPatient
 $\sqsubseteq \exists \text{contraindicated}.\{\text{True}\}$

Extraction d'un tableau

11 propriétés

Patient				Médicament				Rang de recommandation
Âge	Sexe	Patho	...	Effic.	EI	Durée	...	
adulte	femme	cystite	...	oui	non	longue	...	2
adulte	femme	cystite	...	oui	non	courte	...	1
adulte	femme	cystite	...	non	oui	courte	...	non-reco
...	...	...	...	...	...	...	...	...
enfant	-	cystite	...	oui	non	longue	...	1
enfant	-	cystite	...	non	non	courte	...	2
enfant	-	cystite	...	???	oui	courte	...	non-reco
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

66 sit.

50 méd.

Apprentissage des préférences

- Les rangs de recommandation des guides peuvent être traduits en formules de préférences

	Rang
A	2
B	1
C	2
D	non-reco

=> **B > A=C > D**

Résultats

naturellement actif = *Vrai*
 $\wedge$ *probablement actif* = *Vrai*
 $\wedge$ *prouvée* = *Vrai*
 $\wedge$ *non contre indiqué* = *Vrai*

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

Taux d'erreur **$E = 3,5 \%$**

Validation croisée par dixième :
 $E = 5,2 \%$

■ 4 propriétés nécessaires

◆ preuve clinique, absence de contre-indication, sensibilité naturelle, absence de résistance

■ Rôle prépondérant de la propriété protocole

◆ Correspond à la durée de traitement

■ Permet de voir comment les experts gèrent les valeurs manquantes

- ◆ Valeur manquante : poids = 0
- ◆ Pour les effets indésirables, le 0 est plus proche de 2 que de -5
- ◆ => les experts traitent les valeurs manquantes davantage comme des absences d'effets indésirables gênants

Discussion

- => **Une méthode générale pour apprendre des préférences à partir d'une ontologie**
 - ◆ Les approches proposées dans la littérature ne géraient pas les domaines multiples et les propriétés n-aires
- **Permet une meilleure compréhension des recommandations en antibiothérapie**
 - ◆ L'apprentissage de préférences est une approche originale en médecine
 - ◆ Les experts s'appuient sur leurs connaissances tacites pour la rédaction des guides
 - ◆ La quasi-totalité des systèmes d'aide à la décision clinique implémentent les recommandations mais ne cherchent pas à les comprendre

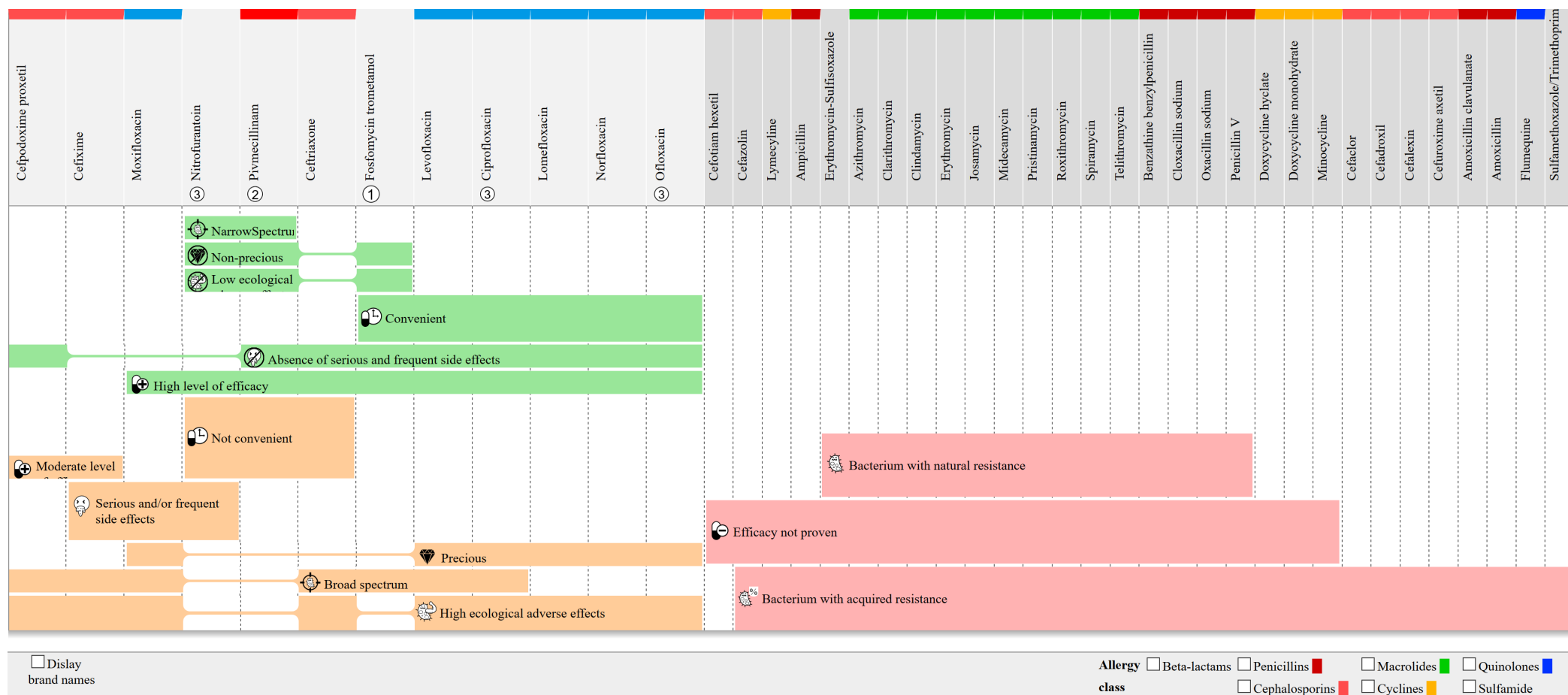
Discussion

- **Le modèle permet d'extrapoler les recommandations lorsqu'elles ne peuvent pas être appliquées**
 - ◆ Taux d'erreur $E = 5,2 \%$ en validation croisée par dixième (*10-fold cross-validation*)
 - ◆ => bonne généralisation
- **Modélisation du raisonnement ?**
 - ◆ Le modèle de préférences permet de retrouver et d'expliquer les recommandations
 - ◆ **MAIS** il ne correspond pas forcément à la manière dont raisonnent médecins et experts (sciences cognitives)

Perspectives

- **Visualisation du modèle de préférences pour expliquer visuellement les recommandations**
 - ◆ Ajout de contrainte visuelle au modèle de préférences, pour faciliter la lecture
 - ◆ Visualisation avec la technique des boîtes arc-en-ciel

Interface AntibioHelp® : Cystite non compliquée chez la femme



References : issue from clinical practice guidelines and a knowledge base built with specialists in antibiotics

Perspectives

- **Intégrer cette interface dans un système d'aide à la décision complet : AntibioHelp®**
 - ◆ Page 1 : saisie du cas patient (âge, pathologie,...)
 - ◆ Page 2 : faut-il prescrire un antibiotique ? + critères d'hospitalisation
 - ◆ Page 3 : quel antibiotique prescrire ? à l'aide des boîtes arc-en-ciel
- **Réaliser une évaluation du système complet en pratique réelle**
- **Application de la méthode à d'autres spécialités ?**
 - ◆ Maladies chroniques ? Cancer du sein ?
- **Utilisation de modèles de préférences plus sophistiqués ?**
 - ◆ Le modèle actuel est linéaire pour la partie des préférences (W)
 - ◆ Des modèles plus complexes seront-ils visualisables ?

Références



- ◆ Tsopra R, Lamy JB, Sedki K.
Using preference learning for detecting inconsistencies in clinical practice guidelines: methods and application to antibiotherapy.
Artificial Intelligence in Medicine 2018;89:24-33
IMIA yearbook best « decision paper » 2019
- ◆ Lamy JB.
Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies.
Artificial Intelligence in Medicine 2017;80:11-28
- ◆ Lamy JB.
Artificial Feeding Birds (AFB): a new metaheuristic inspired by the behavior of pigeons.
Advances in nature-inspired computing and applications 2019;43-60, Springer
- ◆ Tsopra R, Sedki K, Courtine M, Falcoff H, De Béco A, Madar R, Mechaï F, Lamy JB.
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.
Journal of the American Medical Informatics Association 2019