Sur la complexité de l'apprentissage de préférences séparables de type *ceteris paribus*

Jérôme Lang[1] Jérôme Mengin[2]
lang@lamsade.dauphine.fr mengin@irit.fr

[1] LAMSADE
Université Paris-Dauphine
75775 Paris Cedex 16

[2] IRIT
Université de Toulouse
31062 Toulouse Cedex

Résumé :
Nous nous intéressons à l'apprentissage de relations de préférences sur des domaines multi-attributs (ou combinatoires), en faisant une hypothèse très simple d'indépendance entre les attributs: nous supposons que les préférences sur les différents attributs sont séparables. Etant donné un ensemble d'exemples, consistant chacun en une comparaison entre deux alternatives, nous voulons produire un CP-net séparable, consistant en une collection de préférences locales, une par attribut, qui soit compatible avec les exemples. Nous considérons trois formes de compatibilité entre un CP-net et un ensemble d'exemples ; pour chacune nous donnons une caractérisation ainsi que des résultats de complexité.

Mots-clés : Apprentissage de préférences ; CP-nets

Abstract:
We address the problem of learning preference relations on multi-attribute (or combinatorial) domains. We do so by making a very simple hypothesis about the dependence structure between attributes that the preference relation enjoys, namely separability (no preferential dependencies between attributes). Given a set of examples consisting of comparisons between alternatives, we want to output a separable CP-net, consisting of local preferences on each of the attributes, that fits the examples. We consider three forms of compatibility between a CP-net and a set of examples, and for each of them we give useful characterizations as well as complexity results.

Keywords: Preference learning ; CP-nets

1 Introduction

Pour de nombreuses applications, comme le commerce électronique, il est important d'apprendre les préférences d'un utilisateur sur un ensemble d'alternatives (au sens anglais du terme) qui a une structure combinatoire (ou multi-attributs) : chaque alternative est un n-uplet de valeurs, une pour chaque élément d'un ensemble donné d'attributs. Supposons par exemple qu'on observe une utilisatrice faire les choix suivants entre des billets d'avions : (a) elle préfère un vol KLM de nuit atterrissant à Heathrow à un vol Aeroflot de jour atterrissant à Gatwick ; (b) elle préfère un vol Aeroflot de nuit atterrissant à Heathrow à un vol KLM de jour atterrissant à Heathrom ; et (c) elle préfère un vol KLM de jour atterrissant à Heathrow à un vol KLM de nuit atterrissant à Heathrow. Une explication intuitvement correcte (parmi d'autres) de ses choix est qu'elle préfère (inconditionnellement) Aeroflot à KLM, les vols de jour aux vols de nuit (encore uncondtionnellement), et Heathrow à Gatwick, toujours unconditionnellement. Cette explication a la propriété cruciale de *séparabilité* : les préférences sur les valeurs de chaque attribut sont indépendantes des valeurs des autres attributs. Cette explication permet de prédire qu'elle préfère un vol Aeroflot de jour atterrissant à Heathrow à un vol Aeroflot de jour atterrissant à tout autre vol, et un vol Aeroflot de jour atterrissant à Heathrow à un vol Aeroflot de jour atterrissant à Gatwick (mais cela ne permet pas de prédire si elle préfère un vol Aeroflot de jour atterrissant à Gatwick ou un vol Aeroflot de nuit atterrissant à Heathrow).

Si on observe de plus que l'utilisatrice préfère un vol Aeroflot de nuit atterrissant à Gatwick plutôt qu'un vol Aeroflot de jour atterrissant à Gatwick, l'explication précédente n'est plus valable ; et plus généralement, aucune explication où ses préférences seraient séparables n'est valable. Une explication possible (parmi d'autres) pourrait être qu'elle préfère Aeroflot à KLM, Heathrow à Gatwick, les vols de jour aux vols nocturnes avec KLM et *vice-versa* avec Aeroflot.

Cela soulève plusieurs questions intéressantes : comment peut-on décider s'il est possible que les préférences de l'utilisatrice sont séparables ? Si oui, quelles sont les préférences locales sur chaque variable ? Sinon, peut-on trouver une relation de préférence séparable « maximalement compatible » avec l'ensemble d'exemples ? Cet article essaye de répondre à ces questions (mais nous n'abordons pas la recherche de relations de préférence *non séparables* compatibles avec un ensemble d'exemples).

Dans cet article, nous nous concentrons sur l'*apprentissage passif* : le système a en entrée un ensemble de préférences entre alternatives, que l'utilisatrice a déjà exprimées (sans nécessairement qu'on lui ait explicitement demandé de le faire), et on veut raisonner sur ces préférences afin de prédire ses choix sur d'autres alternatives. Cela est différent de l'*élicitation* de préférences (voir par exemple [Chen et Pu, 2004; Braziunas et Boutilier, 2006]), qui consiste à interagir avec l'utilisatrice en lui soumettant des requêtes spécifiques, ou en lui suggérant de nouvelles alternatives, afin de la conduire à exprimer ses préférences sur les valeurs de certains attributs [Viappiani *et al.*, 2006], jusqu'à ce qu'elle ait trouvé l'objet qu'elle préfère ou quitté le système.

Les préférences sur des domaines combinatoires ont été étudiées en détail en théorie de la décision multi-critères (notamment depuis Keeney et Raiffa 1976) et en intelligence artificielle. Les travaux en théroie de la décision multi-critères ont surtout porté sur la *modélisation* des préférences, alors qu'en intelligence artificielle les chercheurs ont plutôt cherché à définir des langages de *représentation* de préférences aussi succincts que possibles et à proposer des algorithmes permettant de trouver les alternatives optimales aussi rapidement que possible.

Les modèles et langages pour les préférences peuvent être différenciés d'abord selon qu'ils manipulent des préférences *ordinales* (des relations d'ordre entre alternatives) ou des préférences *numériques* (par exemple les fonctions d'utilité). Nous nous intéressons ici aux préférences ordinales. Elles ont l'avantage d'être souvent plus facile à obtenir d'un utilisateur.

Alors qu'un certain nombre de travaux récents ont porté sur l'apprentissage de fonctions d'ordre et de préférences sur des paires dans le cas de domaines non combinatoires (voir par exemple Cohen *et al.* 1999, Hüllermeier *et al.* 1999), et que l'apprentissage et l'élicitation de fonctions d'utilité sur des domaines multi-critères ont suscité quelques travaux (Ha et Haddawy 1997 ; Gonzales et Perny 2004,entre autres), il y a eu moins de tentatives d'apprentissage de préférences à partir de comparaisons ordinales, à savoir : (a) l'apprentissage de préférences lexicographiques [Schmitt et Martignon, 2006; Dombi *et al.*, 2007; Yaman *et al.*, 2008] ; (b) quelques travaux sur l'apprentissage de CP-nets [Athienitou et Dimopoulos, 2007; Koriche et Zanuttini, 2009], (c) des préférences *ceteris paribus* plus générales [Sachdev, 2007] ou (d) des fonctions d'utilité apprises à partir de comparaisons ordinales entre formules logiques [Domshlak et Joachims, 2007]. Nous reviendrons sur ces approches en Section 6.

Pour (a) et (b), la relation de préférence sur un domaine multi-critères est construite à partir de relations de préférences « locales » sur les domaines de chaque attribut ; ces ordres locaux sont ensuite combinés soit de manière lexicographique pour (a), soit, pour (b), en faisant l'hypothèse que les comparaisons locales peuvent être généralisées à des comparaisons entre vecteurs de valeurs quand tous les autres attributs ont la même valeur (*ceteris paribus*). Les CP-nets permettent d'exprimer, à l'aide d'un graphe, des dépendances entres les variables, et ensuite de représenter des préférences locales sur les valeurs de chaque attribut, en fonction des valeurs d'autres attributs. Afin d'apprendre des CP-nets de structure quelconque, étant donné un ensemble d'exemples, il faudrait identifier la structure des dépendances entre attributs *et* les tables de préférences conditionnelles locales. On pourrait alors chercher le plus petit CP-net compatible avec les exemples, ou le meilleur compromis entre simplicité et compatibilité avec les exemples. Toutefois, dans cet article, nous nous restreignons à un problème plus simple en ne considérant que la classe des CP-nets les plus simples : les CP-nets séparables, dont le graphe des dépendances entre variables est vide. Il y a plusieurs raisons justifiant ce choix. D'abord, les préférences séparables forment une classe importante de relations de préférences sur des domaines multi-critères, par exemple en économie ou en choix social (voir par exemple [Bradley *et al.*, 2005]), et l'apprentissage de préférences séparables mérite une attention particulière. Deuxièmement, comme nous le verrons, le problème de l'apprentissage de préférences séparables a une complexité de calcul élevée – plus qu'on pourrait le penser *a priori* étant donné qu'on peut *raisonner* en temps linéaire sur des CP-nets séparables. Troisièmement, la mise au point de méthodes pour l'apprentissage de préférences non séparables est une question importante mais aussi très délicate, à laquelle on ne peut sûrement pas répondre en un seul article. Se concentrer d'abord sur des préférences séparables est une étape nécessaire et non triviale vers l'apprentissage de préférences plus complexes. Enfin, il est bien connu qu'une structure simple peut avoir de meilleures propriétés de généralisation qu'une structure plus compliquée qui serait plus en adéquation avec les exemples.

La section 2 rappelle des notions élémentaires sur les préférences sur des domaines multi-attributs, et sur les CP-nets (séparables). Nous nous intéressons ensuite successivement à trois formes de compatibilité entre un CP-net et un ensemble d'exemples : la *compatibilité faible* en section 3, la *compatibilité forte* en section 4, et la *compatibilité implicative* en section 5. Nous prouvons qu'alors qu'il est possible de tester en temps polynomial s'il existe un CP-net séparable qui implique un ensemble d'exemples, déterminer s'il en existe un qui soit faiblement (ou fortement) compatible avec les exemples est NP-complet. La section 6 discute des liens avec des travaux connexes, et la section 7 conclut l'article.

2 Le problème

On veut apprendre un ordre sur des objets, ou *alternatives*, définis par leurs valeurs pour les attributs d'un ensemble $\mathcal{V} = \{X_1, \ldots, X_n\}$. Chaque attribut X_i a un domaine fini D_i de valeurs possibles. On note $D = D_1 \times \ldots \times D_n$ l'ensemble des alternatives possibles. Si $\vec{x} \in D$, alors $(\vec{x})_i$ représente la valeur de $\vec{x}$ pour le i-ème attribut X_i. Un attribut X_i est *binaire* si D_i a deux éléments, que nous notons par convention x_i et $\overline{x_i}$.

Une *relation de préférence (stricte)* $\succ$ est un ordre partiel strict sur D, c'est-à-dire une relation binaire irréflexive et transitive (et donc asymétrique). Si de plus $\succ$ est totale (si $\vec{x} \neq \vec{y}$ alors $\vec{x} \succ \vec{y}$ ou $\vec{y} \succ \vec{x}$) alors $\succ$ est une *relation de préférence linéaire*, ou plus simplement un *ordre linéaire*. On peut remarquer que si on considère la relation transitive $\succ$ comme un graphe sur l'ensemble de sommets D (avec un arc de $\vec{x}$ à $\vec{y}$ si $x \succ y$), alors $\succ$ est irréflexive si et seulement si le graphe est acyclique.

Supposons maintenant qu'on a un ensemble fini d'*exemples* $\mathcal{E}$, où chaque exemple est une paire d'alternatives distinctes $(\vec{x}, \vec{y})$ (que nous noterons souvent $\vec{x} \succ \vec{y}$) telle que $\vec{x}$ est préféré à $\vec{y}$. Dans un système de recommandation, ces exemples pourraient avoir été enregistrés durant l'interaction d'une utilisatrice avec le système. Idéalement, on voudrait apprendre à ordonner toute paire non ordonnée d'alternatives distinctes $\{\vec{x}, \vec{y}\}$. Notre cible est donc une relation de préférence linéaire. [1]

Cet article étudie l'apprentissage de *structures de préférence ceteris paribus séparables*, ou *structure SCP* dans la suite. Les structures SCP sont des cas particuliers des *réseaux de préférences conditionnelles*, ou CP-nets, de [Boutilier *et al.*, 2004], lorsque le graphe de dépendance entre les attributs est vide, c'est-à-dire lorsque chaque attribut est préférentiellement indépendant de tous les autres attributs (les CP-nets permettent de faire dépendre les préférences sur les valeurs d'un attribut des valeurs d'autres attributs). Une structure SCP sur $\mathcal{V}$ est une collection $\mathcal{N}$ d'ordres linéaires $\succ_i$, un sur chaque D_i. Un *"swap"* est une paire d'alternatives $\{\vec{x}, \vec{y}\}$ qui diffèrent seulement par la valeur d'un attribut. Une structure SCP $\mathcal{N}$ induit une relation de préférence $\succ_{\mathcal{N}}$ de la manière suivante : on dit d'abord qu'un ordre linéaire D satisfait $\mathcal{N}$ si pour tout attribut X_i et pour tous $\vec{z}, \vec{z'}$ tels que $\vec{z}$ et $\vec{z'}$ coïncident sur tous les attributs excepté X_i, $\vec{z} \succ \vec{z'}$ si et seulement si $(\vec{z})_i \succ_i (\vec{z'})_i$. Ensuite, $\succ_{\mathcal{N}}$ est l'intersection de toutes les relations de préférences qui satisfont $\mathcal{N}$. On remarque que $\succ_{\mathcal{N}}$ est un ordre partiel sur D. Tout ordre linéaire $\succ$ qui satisfait une structure SCP est *séparable*. La séparabilité signifie que les attributs sont mutuellement indépendants en ce qui concerne les préférences. On peut remarquer que si $\mathcal{N}$ est une structure SCP, alors $\succ_{\mathcal{N}}$ est irréflexive, et peut donc être étendue à au moins un ordre linéaire sur D. [Boutilier *et al.* 2004, Th. 1].

Pour simplifier la présentation, on suppose dans la suite (sauf lorsqu'on indique explicitement le contraire) que tous les attributs sont binaires ; les résultats peuvent toutefois être généralisés à des attributs non binaires, comme on le montre à la fin de la section 3. Dans le cas d'attributs binaires, pour tout paire d'alternatives $(\vec{x}, \vec{y})$, on définit

$$\begin{aligned} \mathit{Diff}(\vec{x}, \vec{y}) &= \{x_i \mid (\vec{x})_i = x_i \text{ et } (\vec{y})_i = \overline{x_i}\} \\ &\cup \{\overline{x_i} \mid (\vec{x})_i = \overline{x_i} \text{ et } (\vec{y})_i = x_i\}. \end{aligned}$$

On a alors la caractérisation suivante de $\succ_{\mathcal{N}}$ (un corollaire des Théorèmes 7 et 8 de Boutilier *et al.*, 2004) :

[1] La méthode et les résultats présentés ici peuvent être facilement généralisés au problème de l'apprentissage de relations de préférences non strictes, lorsqu'on autorise l'indifférence entre deux alternatives. Nous nous restreignons à des relations strictes ici car la présentation est plus simple.

Lemme 1 *Soit $\mathcal{N}$ une structure SCP sur des variables binaires, et soit $\vec{y} \neq \vec{x}$. Alors $\vec{x} \succ_{\mathcal{N}} \vec{y}$ si et seulement si $\mathcal{N}$ contient $x_i \succ \overline{x_i}$ pour tout $x_i \in Diff(\vec{x}, \vec{y})$ et $\overline{x_i} \succ x_i$ pour tout $\overline{x_i} \in Diff(\vec{x}, \vec{y})$.*

Exemple 1 *Soient trois attributs binaires A, B et C, de domaines respectifs $\{a, \overline{a}\}$, $\{b, \overline{b}\}$ et $\{c, \overline{c}\}$, et soit $\mathcal{N} = \{a \succ \overline{a}, b \succ \overline{b}, \overline{c} \succ c\}$. On a $ab\overline{c} \succ_{\mathcal{N}} \overline{a}\overline{b}c$, alors que $a\overline{b}c$ et $\overline{a}bc$ sont incomparables pour $\succ_{\mathcal{N}}$.*

On a écrit plus haut que la cible de notre processus d'apprentissage devrait idéalement être un ordre linéaire sur D, mais nous venons de voir qu'une structure SCP $\mathcal{N}$ ne correspond en général pas à un tel ordre, puisque la relation de préférence $\succ_{\mathcal{N}}$ induite n'est en général pas complète. En fait, $\succ_{\mathcal{N}}$ peut être vue comme l'ensemble de toutes ses complétions, c'est-à-dire qu'une structure SCP exprime un *ensemble de relations de préférences linéaires*.

Si un exemple $(\vec{x}, \vec{y})$ est un swap, alors $\vec{x} \succ_{\mathcal{N}} \vec{y}$ ou $\vec{y} \succ_{\mathcal{N}} \vec{x}$: dans ce cas, on voudrait évidemment que $\mathcal{N}$ soit en accord avec l'exemple en ce qui concerne $\{\vec{x}, \vec{y}\}$. Mais si $(\vec{x}, \vec{y})$ n'est pas un swap, il peut y avoir des complétions $\succ$ et $\succ'$ de $\succ_{\mathcal{N}}$ telles que $\vec{x} \succ \vec{y}$ et $\vec{y} \succ' \vec{x}$. Il nous faut donc commencer par envisager les différentes manières d'estimer dans quelle mesure une structure SCP donné généralise un ensemble d'exemples donné.

Athienitou et Dimopoulos (2007) proposent des algorithmes pour apprendre un CP-net dont l'ordre partiel associé contient tous les exemples – nous dirons dans ce cas qu'il *implique* les exemples. Nous affirmons ici que cette exigence peut être trop forte, comme sur l'exemple suivant :

Exemple 2 *Soient deux attributs binaires X_1 et X_2, de domaines respectifs $\{x_1, \overline{x_1}\}$ et $\{x_2, \overline{x_2}\}$, et l'ensemble d'exemples $\mathcal{E} = \{x_1x_2 \succ x_1\overline{x_2},\ x_1\overline{x_2} \succ \overline{x_1}x_2,\ \overline{x_1}x_2 \succ \overline{x_1}\overline{x_2}\}$.*

Que peut-on espérer apprendre de l'ensemble d'exemples $\mathcal{E}$ ci-dessus ? La fermeture transitive de $\mathcal{E}$ est la relation de préférence complète $x_1x_2 \succ x_1\overline{x_2} \succ \overline{x_1}x_2 \succ \overline{x_1}\overline{x_2}$. Cette relation de préférence est *séparable* : l'agent préfère inconditionnellement x_1 à $\overline{x_1}$ et x_2 à $\overline{x_2}$. Le fait que $x_1\overline{x_2}$ soit préféré à $\overline{x_1}x_2$ indique simplement qu'à choisir entre X_1 et X_2, l'agent préfère abandonner X_2 (on peut par exemple interpréter X_1 comme signifiant « devenir riche » et X_2 comme signifiant « il fera beau demain »). Intuitivement, puisque $\mathcal{E}$ est séparable, on peut espérer renvoyer une structure $\mathcal{N}$ qui contienne $x_1 \succ \overline{x_1}$ et $x_2 \succ \overline{x_2}$. Pourtant, aucune structure SCP n'implique $\mathcal{E}$ (en fait, aucun CP-net n'implique $\mathcal{E}$, quelles que soient les dépendances). La structure $\mathcal{N}$ induit une relation de préférence partielle dans laquelle $x_1\overline{x_2}$ et $\overline{x_1}x_2$ sont incomparables. Plus généralement, aucune structure *ceteris paribus* ne peut « expliquer » que $x_1 \succ \overline{x_1}$ est « plus important que » $x_2 \succ \overline{x_2}$. Ainsi, si on cherche une structure qui *implique* tous les exemples, on renverra simplement « échec ». Par contre, si on cherche une structure SCP qui est simplement *consistante* avec les exemples, c'est-à-dire qui n'implique par le contraire des exemples, alors on renverra $\mathcal{N}$.

L'explication est que quand un agent exprime un CP-net, la relation de préférence induite par ce CP-net n'est pas censée représenter la relation de préférence de l'agent dans sa totalité, mais un sous-ensemble de celle-ci. En d'autres termes, quand un agent exprime le CP-net $\mathcal{N}$, il exprime simplement qu'il préfère x_1 à $\overline{x_1}$ *ceteris paribus* – c'est-à-dire pour une valeur fixée de X_2 – et qu'il préfère aussi x_2 à $\overline{x_2}$ *ceteris paribus* ; le fait que $x_1\overline{x_2}$ et $\overline{x_1}x_2$ soient incomparables dans $\mathcal{N}$ ne veut sûrement pas dire que l'agent les considère comme effectivement incomparables, mais provient du fait que, techniquement, les CP-nets ne sont pas suffisamment expressifs pour représenter la préférence manquante $x_1\overline{x_2} \succ \overline{x_1}x_2$.[2]

Cet exemples suggèrent que, plutôt que de chercher un CP-net qui implique les exemples, on devrait chercher un CP-net qui induit une relation de préférence consistante avec les exemples. Une première manière de définir cette notion de consistance est d'imposer que le CP-net appris $\mathcal{N}$ soit tel que les exemples soient consistants avec au moins une relation de préférence (totale) qui étend $\mathcal{N}$. Dans certains cas, exiger qu'au moins une complétion de $\mathcal{N}$ contienne tous les exemples peut être trop fort, notamment lorsque les exemples proviennent de plusieurs utilisateurs (lorsqu'on veut apprendre les préférences génériques d'un groupe d'utilisateurs), ou d'un seul utilisateur, mais dans plusieurs contextes:

[2] Des langages plus expressifs, comme les TCP-nets [Brafman *et al.*, 2006] ou les théories de préférences conditionnelles [Wilson, 2004] permettraient de représenter ces préférences.

Exemple 3 *Supposons qu'on observe que tous les utilisateurs d'un même groupe préfèrent inconditionnellement x_1 à $\overline{x_1}$ et x_2 à $\overline{x_2}$, alors que leurs préférences entre $x_1\overline{x_2}$ et $\overline{x_1}x_2$ diffèrent (on peut interpréter x_1 et x_2 comme signifiant respectivement « être invité à un bon dîner » et « recevoir une prime de 50 euros »). Alors $\mathcal{E} \supseteq \{\overline{x_1}x_2 \succ x_1\overline{x_2},\ x_1\overline{x_2} \succ \overline{x_1}x_2\}$: il est clair que $\mathcal{E}$ est inconsistent, et il ne peut donc pas y avoir de structure de préférence qui induise un ordre qu'on pourrait compléter en une relation de préférence linéaire qui contienne $\mathcal{E}$. Pourtant, si $\mathcal{N} = \{x_1 \succ \overline{x_1}, x_2 \succ \overline{x_2}\}$, alors chaque exemple de $\mathcal{E}$ est (individuellement) contenu dans au moins une complétion of $\succ_{\mathcal{N}}$.*

On définit maintenant trois notions de compatibilité entre une structure SCP et un ensemble d'exemples. (Ces définitions seraient aussi valables pour le cas de CP-nets de structures quelconques.)

Définition 1 *Soit $\mathcal{N}$ une structure SCP sur $\mathcal{V}$. Un exemple $(\vec{x}, \vec{y})$ est* impliqué par *$\mathcal{N}$ si $\vec{x} \succ \vec{y}$ pour toute complétion $\succ$ de $\succ_{\mathcal{N}}$; $(\vec{x}, \vec{y})$ est* consistant avec *$\mathcal{N}$ s'il existe une complétion $\succ$ de $\succ_{\mathcal{N}}$ telle que $\vec{x} \succ \vec{y}$. De plus, nous dirons qu'un ensemble d'exemples $\mathcal{E}$ est :*
- impliqué par *$\mathcal{N}$ si tout exemple de $\mathcal{E}$ est impliqué par $\mathcal{N}$;*
- globalement (ou fortement) compatible avec *$\mathcal{N}$ s'il existe une complétion $\succ$ de $\succ_{\mathcal{N}}$ telle que pour tout $(\vec{x}, \vec{y}) \in \mathcal{E}$, $\vec{x} \succ \vec{y}$;*
- faiblement compatible avec *$\mathcal{N}$ si tout exemple $(\vec{x}, \vec{y}) \in \mathcal{E}$ est individuellement consistent avec $\mathcal{N}$.*

Enfin, nous dirons que $\mathcal{E}$ est implicativement / fortement / faiblement séparable *s'il est impliqué par / fortement compatible avec / faiblement compatible avec au moins une structure SCP.*

Il est clair que la compatibilité forte implique la compatibilité faible, et que si $\mathcal{E}$ est impliqué par $\mathcal{N}$, alors $\mathcal{E}$ est aussi fortement compatible avec $\mathcal{N}$.

3 Compatibilité faible

On commence par décrire comment la recherche d'une structure SCP qui soit faiblement compatible avec un ensemble d'exemples donné peut être ramenée à un problème de satisfaisabilité propositionnelle. Dans le cas d'attributs binaires, on propose la traduction suivante d'un ensemble d'exemples en un ensemble de clauses propositionnelles, dont, nous le verrons ensuite, les modèles correspondent à des structures SCP faiblement consistantes avec les exemples.

À chaque exemple $\vec{x} \succ \vec{y}$ on associe la clause $C_{\vec{x} \succ \vec{y}}$ qui contient x_i si et seulement si $x_i \in \mathit{Diff}(\vec{x}, \vec{y})$ et $\neg x_i$ si et seulement si $\overline{x_i} \in \mathit{Diff}(\vec{x}, \vec{y})$. Par exemple, si $\vec{x} = x_1\overline{x_2}x_3x_4$ et $\vec{y} = \overline{x_1}x_2x_3\overline{x_4}$ alors $\mathit{Diff}(\vec{x}, \vec{y}) = \{x_1, \overline{x_2}, x_4\}$ et $C_{\vec{x} \succ \vec{y}} = x_1 \vee \neg x_2 \vee x_4$. Cette clause exprime le fait que x_1 est préféré à $\overline{x}_1$, ou $\overline{x}_2$ est preferred à x_2, ou x_4 est préféré à $\overline{x}_4$. (L'explication à cela est que si une structure SCP $\mathcal{N}$ contient $\{\overline{x}_1 \succ x_1, x_2 \succ \overline{x}_2, \overline{x}_4 \succ x_4\}$, alors d'après le lemme 1 on a $\vec{y} \succ_{\mathcal{N}} \vec{x}$, et donc $\mathcal{N}$ n'est pas consistent avec $\vec{x} \succ \vec{y}$.) Si $\mathcal{E}$ est un ensemble d'exemples, alors on pose $\Phi_{\mathcal{E}} = \{C_e \mid e \in \mathcal{E}\}$.

On peut maintenant définir une correspondance bijective entre les interprétations de $\{x_1, \ldots, x_n\}$ et les structures SCP sur $\mathcal{V}$. Si M est une telle interprétation, soit $\mathcal{N}_M$ la structure qui contient $x_i \succ \overline{x_i}$ pour tout i tel que $M \models x_i$ et la préférence $\overline{x_i} \succ x_i$ pour tout i tel que $M \models \neg x_i$. Ainsi, si $M(x_1) = \mathsf{vrai}$, $M(x_2) = \mathsf{faux}$, $M(x_3) = \mathsf{faux}$ et $M(x_4) = \mathsf{vrai}$ alors $\mathcal{N}_M$ contient les tables de préférence $\{x_1 \succ \overline{x_1},\ \overline{x_2} \succ x_2,\ \overline{x_3} \succ x_3,\ x_4 \succ \overline{x_4}\}$.

Proposition 1 *Soit M une interprétation. Un ensemble d'exemples $\mathcal{E}$ est faiblement consistant avec $\mathcal{N}_M$ si et seulement si $M \models \Phi_{\mathcal{E}}$.*

Proof : On montre que $\mathcal{N}_M \not\models \vec{y} \succ \vec{x}$ si et seulement si $M \models C_{\vec{x} \succ \vec{y}}$. On peut supposer sans perte de généralité que $\vec{x} = x_1 \ldots x_n$ et $\vec{y} = \overline{x_1} \ldots \overline{x_i}x_{i+1} \ldots x_n$. Alors $M \not\models C_{\vec{x} \succ \vec{y}}$ si et seulement si $\mathcal{N}_M$ contient $\overline{x_1} \succ x_1, \ldots, \overline{x_i} \succ x_i$, ce qui, d'après le lemme 1, est équivalent à $\mathcal{N}_M \models \vec{y} \succ \vec{x}$. ∎

Corollaire 1 *$\mathcal{E}$ est faiblement séparable si et seulement si $\Phi_{\mathcal{E}}$ est satisfaisable.*

En conséquence, la séparabilité faible se ramène à un test de satisfaisabilité.

Exemple 4 *Soient les trois attributs binaires A, B, C, et l'ensemble d'exemples*

$$\mathcal{E} = \left\{ \begin{array}{c} abc \succ \overline{a}bc, \overline{a}\overline{b}c \succ ab\overline{c}, \\ \overline{a}b\overline{c} \succ a\overline{b}c, \overline{a}bc \succ \overline{a}\overline{b}\overline{c} \end{array} \right\}$$

$\Phi_{\mathcal{E}}$ a un modèle unique, qui correspond à la structure SCP $\mathcal{N} = \{a \succ \overline{a}, b \succ \overline{b}, c \succ \overline{c}\}$. $\mathcal{N}$ est donc l'unique structure SCP faiblement compatible avec $\mathcal{N}$, ce qui montre que $\mathcal{E}$ est faiblement séparable.

Proposition 2 *Décider si un ensemble d'exemples sur des attributs binaires est faiblement séparable est* NP*-complet.*

Proof : L'appartenance à NP est une conséquence directe du corollaire 1. Pour montrer la NP-difficulté, on utilise une réduction depuis 3SAT : soit Φ un ensemble de clauses à trois littéraux non tautologiques. Pour toute clause $C = l_1 \vee l_2 \vee l_3$ de Φ on crée un exemple $e_C = (\vec{x} \succ \vec{y})$ avec

- $\vec{x} = \varepsilon_1.x_1\varepsilon_2.x_2 \dots \varepsilon_n.x_n$ and
- $\vec{y} = \varepsilon'_1.x_1\varepsilon'_2.x_2 \dots \varepsilon'_n.x_n$,

où, pour tout i, $\varepsilon_i.x_i = \varepsilon'_i.x_i = x_i$, sauf si $l_j = \neg x_i$ pour un certain j, auquel cal $\varepsilon_i.x_i = \neg x_i$; et si $l_j = x_i$ pour un certain j, alors $\varepsilon'_i.x_i = \neg x_i$. Soit alors $\mathcal{E}_\Phi = \{e_C \mid C \in \Phi\}$. Par exemple, si $\Phi = \{x_1 \vee \neg x_2 \vee x_3, \neg x_1 \vee x_2 \vee x_4, x_2 \vee x_3 \vee x_4\}$ alors $\mathcal{E}_\Phi = \{x_1\overline{x_2}x_3x_4 \succ \overline{x_1}x_2\overline{x_3}x_4, \overline{x_1}x_2x_3x_4 \succ x_1\overline{x_2}x_3\overline{x_4}, x_1x_2x_3x_4 \succ x_1\overline{x_2}\overline{x_3}\overline{x_4}\}$. On vérifie facilement que $\Phi_{\mathcal{E}_\Phi} = \Phi$, et donc, d'après le corollaire 1 on conclut que Φ est satisfiable si et seulement si $\mathcal{E}_\Phi$ est faiblement séparable. ∎

La généralisation à des domaines non binaires n'est pas difficile. Au lieu d'avoir un symbole propositionnel par attribut, on a un symbole propositionnel par paire de valeurs de chaque attribut. Par exemple, dans le cas d'un attribut X de domaine $\{d_1, d_2, d_3\}$, on a trois symboles $d_1 \succ d_2$, $d_1 \succ d_3$ et $d_2 \succ d_3$ ($d_2 \succ d_1$ étant équivalent à $\neg(d_1 \succ d_2)$, etc.). Par exemple, si Y est un attribut binaire, la clause correspondant à l'exemple $d_1 y \succ d_2 \overline{y}$ est $(d_1 \succ d_2) \vee y$. La différence principale avec le cas binaire est qu'il faut maintenant des clauses pour imposer la transitivité des préférences locales. Soit $Trans = \bigwedge_{X_i \in \mathcal{V}} Trans_{X_i}$ la formule propositionnelle exprimant cette transitivité – par exemple, avec $D_1 = \{d_1, d_2, d_3\}$ on a

$$Trans_{X_1} = (d_1 \succ d_2 \wedge d_2 \succ d_3 \rightarrow d_1 \succ d_3) \wedge \dots$$

On remarque que $Trans$ est de taille polynomiale. On a maintenant une correspondance bijective entre les structures SCP et les interprétations qui satisfont $Trans$: $M \models \Phi_{\mathcal{E}} \wedge Trans$ si et seulement si $\mathcal{N}_M$ est faiblement consistent avec $\mathcal{N}_M$. En conséquence, $\mathcal{E}$ est faiblement séparable si et seulement si $\Phi_{\mathcal{E}} \wedge Trans$ est satisfaisable.

Cette traduction montre que la consistance faible pour des attributs non binaires est dans NP, et donc que le problème reste NP-complet.

Dès que $\mathcal{E}$ devient grand par rapport au nombre d'attributs n, la probabilité que $\mathcal{E}$ soit faiblement séparable devient faible. Dans ce cas, on pourrait vouloir déterminer une structure SCP qui soit faiblement compatible avec le plus grand nombre d'exemples de $\mathcal{E}$ possible. Le corollaire 1 appliqué à des sous-ensembles de $\mathcal{E}$ montre que ce problème revient à un problème de type MAXSAT. La structure SCP qui correspond le mieux à un ensemble d'exemples donné (au sens de la compatibilité faible) correspond à l'interprétation qui maximise le nombre de clauses de $\Phi_{\mathcal{E}}$ satisfaites. Ce résultat peut être étendu au cas de variables non binaires, avec la différence que les clauses de $Trans$ sont protégées. La propositions 1 indique aussi qu'on peut utiliser des méthodes d'approximation polynomiale ; et que, dans le cas binaire, si pour chaque exemple $(\vec{x}, \vec{y})$ de $\mathcal{E}$, $\vec{x}$ et $\vec{y}$ diffèrent sur au plus deux attributs, alors on peut tester si $\mathcal{E}$ est faiblement séparable en temps polynomial (mais le problème d'optimisation reste NP-difficile dans ce cas).

4 Compatibilité forte

Il est moins facile de caractériser la compatibilité forte. La différence entre compatibilité faible et forte tient à ceci : alors que pour la compatibilité faible on cherche une structure SCP consistante avec chaque exemple de $\mathcal{E}$ pris individuellement, pour la compatibilité forte on cherche une structure SCP consistante avec tout l'ensemble d'exemples $\mathcal{E}$, pris globalement.

Exemple 4, suite *$\mathcal{E}$ n'est pas fortement compatible avec $\mathcal{N}$, parce que $\mathcal{E} \cup \succ_{\mathcal{N}}$ possède le cycle suivant :*

$$a\overline{b}c \succ_{\mathcal{N}} \overline{a}\overline{b}c \succ_{\mathcal{E}} ab\overline{c} \succ_{\mathcal{N}} \overline{a}b\overline{c} \succ_{\mathcal{E}} a\overline{b}c$$

Puisque $\mathcal{E}$ n'est fortement compatible avec aucune autre structure SCP que $\mathcal{N}$ (et ceci parce que $\mathcal{N}$ est la seule structure SCP faiblement compatible avec $\mathcal{N}$), $\mathcal{E}$ n'est pas fortement séparable.[3]

[3]Remarquons que $\mathcal{E}$ est à la fois faiblement consistant et sans cycles, comme on a vu à l'example 3, et pourtant il n'est pas fortement consistent.

Remarquons que toutes les alternatives contenues dans le cycle de l'exemple ci-dessus apparaissent dans $\mathcal{E}$. Plus généralement, si on note $D(\mathcal{E})$ l'ensemble des alternatives de D qui apparaissent dans $\mathcal{E}$, alors on a la caractérisation suivante de la compatibilité forte :

Proposition 3 *$\mathcal{E}$ est fortement compatible avec $\mathcal{N}$ si et seulement si la restriction de $\succ_{\mathcal{N}} \cup \mathcal{E}$ à $D(\mathcal{E})$ est acyclique.*

Proof : Par définition, si $\mathcal{E}$ est fortement compatible avec $\mathcal{N}$ alors $\succ_{\mathcal{N}}$ peut être complété en un ordre linéaire contenant $\mathcal{E}$, donc $\succ_{\mathcal{N}} \cup \mathcal{E}$ est acyclique, et sa restriction à $D(\mathcal{E})$ aussi. Réciproquement, supposons que $\mathcal{E}$ n'est pas fortement compatible avec $\mathcal{N}$: $\succ_{\mathcal{N}} \cup \mathcal{E}$ ne peut pas être étendu en un ordre linéaire, ce qui signifie qu'elle n'est pas irreflexive, donc qu'elle contient un cycle Γ. Puisque les structures SCP sont consistantes, Γ doit contenir au moins un arc de $\mathcal{E}$. On peut transformer Γ en un cycle Γ' dont les sommets sont tous dans $D(\mathcal{E})$: il suffit de remplacer, tant que c'est possible, toute séquence de sommets dans Γ de la forme $x \succ_{\mathcal{N}} y \succ_{\mathcal{N}} z$ par $x \succ_{\mathcal{N}} z$ (ce qui est possible parce que $\succ_{\mathcal{N}}$ est transitive). Or, tout arc dans Γ' est soit dans $\mathcal{E}$, soit adjacent à deux arcs de $\mathcal{E}$ (il ne peut pas être réduit à un arc réflexif simple de $\succ_{\mathcal{N}}$, parce que $\succ_{\mathcal{N}}$ est irreflexive), donc ses extrémités sont dans $D(\mathcal{E})$. Donc Γ' est un cycle de la restriction de $\succ_{\mathcal{N}} \cup \mathcal{E}$ à $D(\mathcal{E})$. ∎

Puisque la restriction de $\succ_{\mathcal{N}} \cup \mathcal{E}$ à $D(\mathcal{E})$ a au plus $2.|\mathcal{E}|$ sommets, on peut déterminer s elle possède un cycle en temps polynomial, donc déterminer si $\mathcal{E}$ est fortement compatible avec $\mathcal{N}$ est dans P. Puisque la taille d'une structure SCP est linéaire par rapport au nombre de variables, déterminer si $\mathcal{E}$ est fortement séparable est dans NP.

Proposition 4 *Déterminer si $\mathcal{E}$ est fortement séparable est NP-complet.*

Proof : La difficulté vient de la réduction suivante depuis le problème de SÉPARABILITÉ FAIBLE. Soit $\mathcal{E} = \{\vec{x}_1 \succ \vec{y}_1, \ldots, \vec{x}_m \succ \vec{y}_m\}$ un ensemble d'exemples, construit sur un ensemble de symboles propositionnels X. À $\mathcal{E}$ on associe l'ensemble d'examples $\mathcal{E}'$, construit sur l'ensemble d'attributs $X \cup \{P_1, \ldots, P_m\}$. Pour tout $e_i = \vec{x}_i \succ \vec{y}_i$ in $\mathcal{E}$ on crée l'exemple e'_i suivant:

$$\vec{x}_i\overline{p}_1 \cdots \overline{p}_{i-1}p_i\overline{p}_{i+1} \cdots \overline{p}_m \\ \succ \\ \vec{y}_i\overline{p}_1 \cdots \overline{p}_{i-1}p_i\overline{p}_{i+1} \cdots \overline{p}_m.$$

On a alors que $\mathcal{E}' = \{e'_i \mid e_i \in \mathcal{E}\}$ est fortement séparable si et seulement si $\mathcal{E}$ est faiblement séparable. Supposons que $\mathcal{E}$ est faiblement séparable : il existe alors une structure SCP $\mathcal{N}$ consistante avec chaque e_i in $\mathcal{E}$. Soit $\mathcal{N}' = \mathcal{N} \cup \{p_1 \succ \overline{p}_1, \ldots, p_n \succ \overline{p}_n\}$. Pour tout i, parce que $\mathcal{N}$ est consistante avec $\vec{x}_i \succ \vec{y}_i$, il existe un ordre linéaire $\succ_i$ étendant $\succ_{\mathcal{N}}$ et contenant $\vec{x}_i \succ \vec{y}_i$. Soit $\succ$ la relation de préférence définie par: $\vec{x}\vec{p} \succ \vec{y}\vec{p'}$ si et seulement si l'une de ces conditions est vérifiée :

(a) $\vec{p}$ est lexicographiquement plus grand que $\vec{p'}$ (ce que nous notons $\vec{p} >_{lex} \vec{p'}$), c'est-à-dire, il existe un $i \leq m$ tel que $\vec{p}$ contient p_i, $\vec{p'}$ contient $\overline{p}_i$, et pour tout $j < i$, $\vec{p}$ contient p_j si et seulement si $\vec{p'}$ contient p_j.

(b) $\vec{p}=\vec{p'}=\overline{p}_1 \cdots \overline{p}_{i-1}p_i\overline{p}_{i+1} \cdots \overline{p}_m$ pour un i, et $\vec{x} \succ_i \vec{y}$;

(c) $\vec{p} = \vec{p'}$, $\vec{p}$ n'est pas de la forme $\overline{p}_1 \cdots \overline{p}_{i-1}p_i\overline{p}_{i+1} \cdots \overline{p}_m$, et $\vec{x} \succ^* \vec{y}$, où $\succ^*$ est une relation de préférence étendant $\succ_N$.

On peut vérifier que $\succ$ est une relation de préférence complète $\succ$ étendant $\succ_{\mathcal{N}'}$, et contenant $\mathcal{E}'$, ce qui implique que $\mathcal{E}'$ est fortement consistant avec $\mathcal{N}'$, et que $\mathcal{E}'$ est fortement séparable .

Réciproquement, supposons que $\mathcal{E}'$ soit fortement séparable . Soit $\mathcal{N}'$ fortement consistant avec $\mathcal{E}'$, et $\succ$ étendant $\succ_{\mathcal{N}'}$ et contenant $\mathcal{E}'$. Pour tout $i \leq m$, définissons $\succ_i$ comme suit : $\vec{x} \succ_i \vec{y}$ *iff* $\vec{x}\overline{p}_1 \cdots \overline{p}_{i-1}p_i\overline{p}_{i+1} \cdots \overline{p}_m \succ \vec{y}\overline{p}_1 \cdots \overline{p}_{i-1}p_i\overline{p}_{i+1} \cdots \overline{p}_m$. Alors, pour tout i: $\succ_i$ est un ordre linéaire sur 2^X ; $\succ_i$ étend $\succ_{\mathcal{N}}$; et $\succ_i$ contient $\vec{x}_i \succ \vec{y}_i$, parce que $\succ$ contient $\vec{x}_i\overline{p}_1 \cdots \overline{p}_{i-1}p_i\overline{p}_{i+1} \cdots \overline{p}_m \succ \vec{y}_i\overline{p}_1 \cdots \overline{p}_{i-1}p_i\overline{p}_{i+1} \cdots \overline{p}_m$. Donc, pour tout i on a trouvé une extension $\succ_i$ de $\succ_{\mathcal{N}}$ contenant $\vec{x}_i \succ \vec{y}_i$, ce qui implique that $\mathcal{E}$ est faiblement compatible avec $\mathcal{N}$, et que $\mathcal{E}$ est faiblement séparable . ∎

Remarquons que bien que la séparabilité faible et la séparabilité forte aient la même complexité, la séparabilité faible a la propriété intéressante qu'il existe une traduction simple en SAT préservant les solutions (les modèles de $\Phi_{\mathcal{E}}$ correspondent bijectivement aux structures SCP-nets faiblement consistantes avec $\mathcal{E}$), ce qui permet à la consistance faible d'être calculée en pratique en utilisant des algorithmes pour SAT. Pour calculer une structure SCP fortement consistante avec $\mathcal{E}$, on peut générer des structures $\mathcal{N}$ faiblement consistantes avec $\mathcal{E}$, et tester ensuite l'acyclicité de $\succ_{\mathcal{N}} \cup \mathcal{E}$.

5 Compatibilité implicative

Il est facile de déterminer s'il existe une structure SCP qui implique $\mathcal{E}$. Dans le cas de variables binaires, à chaque exemple $\vec{x} \succ \vec{y}$ on associe la conjonction de littéraux $\Gamma_{\vec{x}\succ\vec{y}} = \neg C_{\vec{y}\succ\vec{x}}$. Soit $\Gamma_{\mathcal{E}} = \bigwedge\{\Gamma_e \mid e \in \mathcal{E}\}$. À l'aide du Lemme 1, on peut prouver que $M \models \Gamma_{\mathcal{E}}$ si et seulement si $\mathcal{N}_M$ implique $\mathcal{E}$. La recherche d'un modèle d'une conjonction de clauses se fait en temps polynomial, donc déterminer si un ensemble d'exemples est implicativement séparable est dans P.

Bien que la séparabilité implicative soit plus facile que les séparabilités forte et faible, comme nous l'avons expliqué plus haut, nous pensons que c'est une notion bien trop forte.

6 Travaux connexes

L'apprentissage de préférences est un champ de recherches actif, qui s'intéresse à divers aspects de l'apprentissage de préférences, notamment l'arrangement de labels [Hüllermeier *et al.*, 2008] ou l'arrangement d'objets [Cohen *et al.*, 1999], mais jusqu'à présent ces recherches se sont focalisées sur des domaines simples, non combinatoires[4].

L'apprentissage de fonctions numériques sur des domaines combinatoires a été considéré, notamment dans [Ha et Haddawy, 1997; Gonzales et Perny, 2004]. Récemment, Domshlak et Joachims [2007] ont proposé une méthode pour apprendre une fonction d'utilité à partir de préférences entre formules logiques. En ce qui concerne l'apprentissage de préférences *ordinales* sur des domaines multiattributs, une approche qui suit une méthodologie proche de la nôtre (mais pour une classe très différente de structures de préférences) est le travail sur l'apprentissage de préférences lexicographiques sur des domaines multiattributs [Schmitt et Martignon, 2006; Dombi *et al.*, 2007; Yaman *et al.*, 2008]. Une différence importante avec notre travail est qu'ils apprennent la relation de préférence en entier, alors que nous n'apprenons qur les préférences locales. Bien entendu, apprendre la relation de préférence en entier donne une information plus riche, mais cela a un prix : l'hypothèse que les préférences sont lexicographiques est extrêmement restrictive. Les préférences lexicographiques sont une forme très spécifique de préférences, que l'on ne rencontre que rarement en pratique. Il faut remarquer que toute relation de préférence lexicographique est séparable, mais que l'ensemble des relations de préférence séparables est bien plus grand. Il est intéressant de noter que les résultats de complexité sont similaires.

Koriche et Zanuttini [2009] montrent qu'il est possible d'éliciter efficacement des CP-nets en utilisant des requêtes simples de la forme $\vec{x} \succ^? \vec{y}$, où $\vec{x}$ et $\vec{y}$ diffèrent seulement sur la valeur d'une variable. L'apprentissage passif de CP-nets a été considéré par Athienitou et Dimopoulos [2007]. Comme nous l'avons expliqué plus haut, ils ne s'intéressent qu'à la compatibilité implicative, ce qui limite considérablement l'applicabilité de leur approche. Sachdev [2007] propose d'apprendre des "théories de préférence" (au sens de Doyle *et al.* [1991]) consistantes avec un ensemble d'exemples. Il suppose qu'on a accès à l'ensemble complet des exemples correspondant à une théorie de préférence existante, ce qui limite l'applicabilité pratique de son approche.

7 Conclusion et travaux futurs

Notre contribution est double: nous avons montré que la recherche d'une relation de préférence *ceteris paribus* séparable (ou de manière équivalente, la recherche d'un CP-net séparable) compatible avec un ensemble d'exemples n'est pas immédiate à définir en raison du fait que, au contraire des préférences lexicographiques, les préférences séparables ne permettent pas d'induire une relation de préférence complète sur le domaine multiattributs à partir des relations de préférence locales. Nous avons proposé et

[4]Des domaines multiattributs sont parfois considérés, mais avec un rôle très différent : par exemple, en arrangement de labels on apprend une relation de préférence sur un domaine simple, étant donnée une information sous la forme d'un vecteur de valeurs d'attributs (par exemple, on cherche à prédire la façon dont un utilisateur va ordonner un ensemble de candidats à une élection, étant donnés son sexe, son âge et sa profession).

discuté trois formes de compatibilité entre un CP-net séparable et un ensemble d'exemples. Pour chacune d'entre elles, nous avons identifié sa complexité et donné des caractérisations qui peuvent être utilisées pour calculer un CP-net séparable qui correspond au mieux à un ensemble d'exemples. Bien que la séparabilité implicative soit plus facile à déterminer que la séparabilité faible et la séparabilité forte, elle est toutefois bien moins intéressante que les deux autres notions.

Plus généralement, nos définitions de compatibilités implicative, forte et faible sont pertinentes pour n'importe quel CP-net. Jusqu'à présent nous n'avons des caractérisations et des résultats de complexité pour ces trois formes de consistance que dans le cas séparable, c'est-à-dire le cas où le graphe de dépendence n'a pas d'arc. Une question à long terme consisterait à trouver des méthodes pour apprendre des CP-nets avec n'importe quelle structure, en se focalisant d'abord sur les structures simples, par exemples les hyperarbres. Une première étape consisterait à apprendre des CP-nets avec un graphe *fixé*: étant donné un ensemble d'exemples $\mathcal{E}$ et un graphe de dépendance G, renvoyer le CP-net maximisant la G-compatibilité (pour chacune des trois formes de compatibilité) avec $\mathcal{E}$. Malheureusement, cela n'est pas si simple. Notre traduction d'exemples en clauses propositionnelles peut être généralisée, en utilisant pour chaque attribut binaire une variable propositionnelle pour toute combinaison de valeurs de ses parents. Par exemple, si A n'a aucun parent et B n'a comme parent que A, on utilise trois variables propositionnelles : a, $(a{:}b)$ et $(\overline{a}{:}b)$, où, par exemple, l'affectation de $(a:b)$ à vrai correspond à l'entrée $a:b \succ \overline{b}$ dans la table de préférence de B. La compatibilité faible peut alors être exprimée en utilisant, pour chaque exemple $\vec{x} \succ \vec{y}$, une clause pour chaque séquence de swaps de $\vec{y}$ à $\vec{x}$, exprimant qu'au moins une des étapes de la séquence est bloquée. Ceci garantit qu'on n'a pas $\vec{y} \succ_{\mathcal{N}} \vec{x}$. Mais il peut y avoir un nombre exponentiel de telles clauses, même dans le cas où le nombre de parents dans le graphe de dépendance est borné, ce qui limite considérablement l'applicabilité de cette traduction.

Remerciements

Un grand merci à Richard Booth, Yann Chevaleyre, Kevin Garcia, Peter Haddawy, Frédéric Koriche, Mathieu Serrurier, Chattrakul Sombattheera et Bruno Zanuttini pour de nombreuses discussions au sujet de l'apprentissage de CP-nets. Cette recherche a partiellement bénéficié du soutien du projet ANR-05-BLAN-0384 "Preference Handling et Aggregation in Combinatorial Domains".

References

F. Athienitou et Y. Dimopoulos. Learning CP-networks: a preliminary investigation. In *Proc. 3rd Multidisciplinary Work. on Advances in Preference Handling (PREF'07)*, 2007.

C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, et D. Poole. CP-nets: a tool for representing et reasoning avec conditional ceteris paribus préférence statements. *JAIR*, 21:135–191, 2004.

W. J. Bradley, J. K. Hodge, et D.M. Kilgour. Separable discrete preferences. *Mathematical Social Science*, 49(3):335–353, 2005.

R. Brafman, C. Domshlak, et S. Shimony. On graphical modeling of préférence et importance. *JAIR*, 25:389–424, 2006.

D. Braziunas et C. Boutilier. Elicitation of factored utilities. *AI Magazine*, 29 (4), 2008.

L. Chen et P. Pu. Survey of préférence elicitation methods. Tech. Rep. 200467, EPFL, 2004.

W. Cohen, R. Schapire, et Y. Singer. Learning to order things. *JAIR*, 10:243–270, 1999.

J. Dombi, C. Imreh, et N. Vincze. Learning lexicographic orders. *Eur. J. of Operational Research*, 183:748–756, 2007.

C. Domshlak et T. Joachims. Efficient et non-parametric reasoning over user preferences. *User Modeling et User-Adapted Interaction (UMUAI)*, 17(1-2):41–69, April 2007.

J. Doyle, Y. Shoham, et M. Wellman. A logic of relative desire (preliminary report). In *Proc. ISMIS '91*, 16–31. 1991.

C. Gonzales et P. Perny. GAI networks for utility elicitation. In *Proc. KR'04*, pages 224–233. AAAI Press, 2004.

V. Ha et P. Haddawy. Problem-focused incremental elicitation of multi-attribute utility models. In *Proc. UAI-97*, pages 215–222, 1997.

E. Hüllermeier, J. Fürnkranz, W. Cheng, et K. Brinker. Label ranking by learning pairwise preferences. *AIJ*, 172:1897–1917, 2008.

R. Keeney et H. Raiffa. *Decision avec Multiple Objectives: Preferences et Value Trade-offs*. Wiley, 1976.

F. Koriche et B. Zanuttini. Learning conditional préférence networks avec queries, 2009. In *Proc. IJCAI'09*.

M. Sachdev. On learning of ceteris paribus préférence theories. Master's thesis, Graduate Faculty of North Carolina State University, 2007.

M. Schmitt et L. Martignon. On the complexity of learning lexicographic strategies. *J. Mach. Learn. Res.*, 7:55–83, 2006.

P. Viappiani, B. Faltings, et P. Pu. Preference-based search using example-critiquing avec suggestions. *JAIR*, 27:465–503, 2006.

N. Wilson. Extending CP-Nets avec Stronger Conditional Preference Statements. In *AAAI-04*, pages 735–741, 2004.

F. Yaman, T. Walsh, M. Littman, et M. desJardins. Democratic approximation of lexicographic préférence models. In *Proc. ICML '08*, pages 1200–1207, 2008.