



UNIVERSITE GASTON BERGER

Master2 Informatique

Memoire de fin de cycle

Présenté par:

**PEKPASSI Digonaou**

29 Juillet 2015

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Aperçut des méthodes d'extraction de préférences</b>	<b>5</b>
2.1	Les types de classement . . . . .	5
2.1.1	Le classement de labels . . . . .	6
2.1.2	Le classement d'instances . . . . .	7
2.1.3	Classement d'objets . . . . .	7
<b>3</b>	<b>Les types de modèles de préférence</b>	<b>8</b>
3.1	Modèles de préférences quantitatifs . . . . .	8
3.2	Modèles de préférence qualitatifs . . . . .	8
3.2.1	Propriétés de base des relations binaires . . . . .	9
3.2.2	Formulations rencontrés dans les modèles de préférence qualitatives	10
3.2.3	Compositions des modèles de préférence qualitatives . . . . .	13
3.2.4	Modèle lexicographique . . . . .	13
3.2.5	Modèle à base de règles . . . . .	13
3.2.6	Modèle à base de CPNets . . . . .	13
3.2.7	CPRefMiner . . . . .	15
3.2.8	ProfileMining . . . . .	15
3.3	Méthodes de construction des profils de préférence . . . . .	15
3.3.1	Par la méthode de boosting . . . . .	15
3.3.2	Par la méthode SVM . . . . .	15
3.3.3	Par la descente de gradient . . . . .	15
3.3.4	Par les l'extraction de motifs fréquents . . . . .	15
<b>4</b>	<b>Définition et extraction des règles de préférence contextuelle</b>	<b>16</b>
4.1	Préférences utilisateur . . . . .	16
<b>5</b>	<b>Utilisation des motifs séquentiels dans le cadre des préférences contex- tuelles (approche Sprex)</b>	<b>19</b>
5.1	Représentation séquentielle des préférences . . . . .	20
5.2	Description de la construction du profil de préférence par Sprex-Build . . .	21
5.3	Description de l'approche utilisée pour la prédiction des prafarences utilia- teurs (Sprex-Predict) . . . . .	22

<b>6</b>	<b>Extraction des préférences contextuelles dans notre approche</b>	<b>23</b>
6.1	Formules de préférence suivant les travaux de ([?]) . . . . .	24
6.2	Formules de préférence suivant notre approche . . . . .	25
6.2.1	Séquences de formules de préférence élémentaires . . . . .	27
6.3	Construction du profil utilisateur . . . . .	32
6.4	Prédiction des préférences utilisateur . . . . .	35
6.5	Implémentation du travail . . . . .	35
6.5.1	Le programme Sprex . . . . .	36
6.5.2	Le programme Colico . . . . .	37
6.6	Evaluations expérimentales . . . . .	40
6.6.1	Interprétation . . . . .	41
6.6.2	Etapes de transformation des préférences utilisateurs en transaction de préférence . . . . .	42
6.6.3	Etape de l'extraction des règles interressantes minimales des tran- sactions de préférence . . . . .	42
6.7	Algorithme . . . . .	44
6.8	Exemple d'extraction des règles de préférence . . . . .	44
6.8.1	Cas où le schémas relationnel n'a que des attributs symboliques . .	44
6.9	Determination des itemsets interressants minimaux avec supmin=2 et conf- min=0.5 . . . . .	45
6.9.1	Cas où le schémas relationnel a des attributs numériques et symboliques	46
6.10	Determination des itemsets interressants minimaux avec supmin=2 et conf- min=0.5 . . . . .	48

# 1 Introduction

L'extraction des préférences est un domaine qui a pour but de prédire les préférences utilisateurs. Il trouve par exemple son application dans les systèmes de recommandation, dans la recherche personnalisée.

Les informations que les utilisateurs considèrent comme pertinents diffèrent d'un utilisateur à un autre ce qui est en partie dû au fait que les utilisateurs n'ont pas les mêmes préférences. C'est pour cela que des méthodes utilisant des données sur les préférences des utilisateurs ont émergé afin de permettre une meilleure personnalisation des services. Les préférences utilisateurs peuvent être recueillies de deux façon, la première en demandant aux utilisateurs de fournir explicitement des informations sur leurs préférences, la seconde en recueillant implicitement ces préférences à partir des interactions entre le système et l'utilisateur.

Dans notre travail, c'est cette dernière approche qui nous fournit les données de préférence, ce qui dans ce cas peut contenir des données incomplètes ou inconsistences qui neccessitent des méthodes d'extraction qui leur sont adaptées.

Dans la suite de ce document nous allons présenter un aperçut des formalisations proposées dans les travaux antérieurs, nous allons ensuite décrire l'approche que nous avons proposé puis, avec l'appuit de résultats expérimentaux, fournir une évaluation et une analyse sur l'efficacité de l'approche proposée.

## 2 Aperçut des méthodes d'extraction de préférences

Les méthodes d'extraction de préférences peuvent se distinguer par deux critères à savoir :

- le *type de classement* utilisé pour la représentation des préférences
- le *type de modèle de préférence* construit pour prédire les préférences

### 2.1 Les types de classement

Dans cette sous-section nous proposons une terminologie unifiée et claire pour les problèmes de classement les plus importants.

Dans le cadre des notations, nous allons utiliser une terminologie qui est souvent utilisée en apprentissage supervisé. Ainsi un objet caractérisant une donnée est appelé instance dénoté  $x$  et la classe à laquelle il est associé sera appelé label de classe. L'espace

caractéristique des instances sera noté  $\mathcal{X}$  et l'espace de sortie sera dénoté  $\mathcal{Y}$ . Les instances sont souvent représentées sous forme de vecteur caractéristique :

$$x = (x_1, x_2, \dots, x_m) \in \mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \dots \times \mathcal{X}_m$$

Nous distinguons trois types de classement à savoir :

- Le classement de labels
- Le classement d'instances
- Le classement d'objets

Nous allons décrire ci dessous ces différents types de classement.

### 2.1.1 Le classement de labels

Soit  $S_y$  l'ensemble des permutations de l'ensemble de labels  $\mathcal{Y} = \{y_1, y_2, \dots, y_k\}$  suivant un ordre total  $\succ$ .

Le *classement de labels* consiste à apprendre un classeur de labels qui est une fonction  $\mathcal{C}$  définie sur  $\mathcal{X} \rightarrow S_y$  et qui permet de fournir en sortie une permutation de l'ensemble  $\mathcal{Y} = \{y_1, y_2, \dots, y_k\}$  en fonction d'une instance  $x \in \mathcal{X}$  en entrée.

Si pour une instance  $x$  on définit par  $\pi_x$  une application de  $\{1, 2, \dots, k\} \rightarrow \{1, 2, \dots, k\}$  qui à chaque indice  $i$  d'un label associe la position  $\pi_x(i)$  de ce label dans la permutation  $\mathcal{C}(x)$ , nous obtenons :

$$y_{\pi_x^{-1}(1)} \succ_x y_{\pi_x^{-1}(2)} \succ_x \dots \succ_x y_{\pi_x^{-1}(k)}$$

L'ensemble d'apprentissage pour le classement de labels consiste souvent en un ensemble de préférences par paires de la forme  $y_i \succ_x y_j$  indiquant que  $y_i$  est préféré à  $y_j$  pour l'instance  $x$ .

#### Example 1

Comme exemple de situations où on a affaire à un classement de label, on peut parler du classement de la pertinence des genres de rubriques (sport, technologie, santé etc..) en fonction des différents journaux.

De même on peut faire un classement de labels sur l'ordre de préférence d'un ensemble de produits (exple : logements de vacances) en fonction des caractéristiques démographiques d'une personne.

### 2.1.2 Le classement d'instances

Dans le classement d'instances les éléments de  $\mathcal{Y} = \{y_1, y_2, \dots, y_k\}$  sont appelés des *classes* et se distinguent entre eux par un ordre naturel  $y_1 < y_2 < \dots < y_k$ . Chaque instance  $x$  de  $\mathcal{X}$  est lié à une classe de  $\mathcal{Y}$ . L'ensemble d'apprentissage pour le classement d'instance, consiste souvent à un ensemble d'instances labelisés par une des classes. Contrairement à la classification, le but n'est pas d'apprendre un classifieur mais une *fonction de classement*  $f(\cdot)$  qui à partir d'un sous ensemble  $X \in \mathcal{X}$  d'instances fournit en entrée, il fournit un classement de ces objets suivant la classe à laquelle est allouée chaque un des objets.

Dans le cas où on a  $k = 2$ , ce problème est connu comme un *problème de classement bipartite*. Dans le cas où  $k > 2$ , il est considéré comme un problème de classement *k-partite* ou *multipartite*.

#### Example 2

Comme exemple, on peut prendre le cas d'un reviewer qui cherche à classer les articles en fonction de leur qualité, en les répartissant en catégories rejetés, faiblement rejetés, faiblement acceptés, acceptés.

### 2.1.3 Classement d'objets

Dans le cas du *classement d'objets*, les instances de  $\mathcal{X}$ , appelés objets, ne sont pas liés à une caractéristique en sortie (label ou autre) comme les classements précédents. Le but est d'apprendre une fonction de classement  $f(\cdot)$  qui suivant un sous ensemble  $Z \in \mathcal{X}$  d'instances fournit en entrée, la fonction un classement de ces objets en sortie. Ceci est souvent effectué en assignant un score à chaque instance et en ordonnant ces instances en fonction de leur score. Comme ensemble d'apprentissage, un classeur d'objets a souvent accès à des exemples de classement entre des paires d'objets de la forme  $z \succ z'$  déterminant que l'objet  $z$  doit être classé au dessus de l'objet  $z'$ .

#### Example 3

Comme exemple, nous pouvons considérer le problème de l'apprentissage du classement des résultats de recherche d'un moteur de recherche.

Le classement d'objet est le classement qui nous concerne dans le cadre de notre travail. Ainsi dans la suite du document nous allons nous restreindre à ce type de classement. A présent nous allons décrire les différents types de modèles de préférences que nous pouvons rencontrer dans le cas du classement d'objet.

### 3 Les types de modèles de préférence

Les modèles de préférence peuvent être principalement catégorisés en deux types :

- Les modèles de préférence quantitatifs. Ceux ci se basent sur le calcul de scores pour prédire les préférences entre les objets ;
- Le modèles de préférence qualitatifs. Ceux ci se basent sur la création de relation binaires pour prédire les préférences entre les objets.

Nous allons décrire plus explicitement ces deux types de modèles de préférence rencontrés mais nous mettront plus l'accent sur les modèles de préférence qualitatifs sur lesquels reposent notre travail.

#### 3.1 Modèles de préférences quantitatifs

Les modèles de préférences quantitatifs sont des modèles construits par apprentissage de préférence et qui calculent des scores sur des objets afin de pouvoir les comparer.

Comme exemple nous pouvons prendre le cas où on a deux tuples  $t_1$  et  $t_2$ , auxquels un modèle de préférence quantitatif alloue respectivement des scores de  $s_{t_1}$  et  $s_{t_2}$ . Si  $s_{t_1} > s_{t_2}$  alors le modèle détermine que  $t_1$  est préféré à  $t_2$ . Il y'a une panoplie de travaux qui ont été menés sur l'étude de modèles de préférences quantitatives, comme les travaux utilisant le boosting, le SVM, la descente de gradient etc. Nous n'allons pas explicité ces travaux vu que nous nous focaliseront sur les modèles de préférence qualitatifs qui sont le cadre dans lequel s'effectue notre travail.

#### 3.2 Modèles de préférence qualitatifs

Les modèles de préférence qualitatifs sont en quelque sorte des relations binaires qui à partir des objets à comparer et déterminent quel objet est plus préféré qu'un autre. Ainsi ces modèles de préférences peuvent se distinguer par les différentes propriétés caractérisant les relations binaires (réflexivité, transitivité, ..), les formes de structures utilisés dans ces modèles (formules logiques, contextes,..) et les différentes combinaisons qui peuvent être effectués entre ces modèles pour construire d'autres formes de modèles. Ainsi dans la suite de ce document nous allons tout d'abord faire un rappel sur les propriétés des relations binaires puis nous allons décrire les différentes structures rencontrés dans ces modèles et les différentes manière d'obtenir des compositions de ces modèles.



### 3.2.1 Propriétés de base des relations binaires

#### Definition 1. *Relation binaire*

Une relation binaire  $\mathcal{R}$  sur un ensemble  $E$  est un sous-ensemble du produit cartésien  $E \times E$  ie un ensemble de couples  $(x, y)$  d'éléments de  $E$ . Nous noterons  $x\mathcal{R}y$  pour indiquer que le couple  $(x, y)$  appartient à la relation  $\mathcal{R}$ .

Une relation binaire est :

- réflexive si  $\forall x \in E, x\mathcal{R}x$
- irreflexive si  $\forall x \in E, \neg(x\mathcal{R}x)$
- symétrique si  $\forall x, y \in E, x\mathcal{R}y \Rightarrow y\mathcal{R}x$
- antisymétrique si  $\forall x, y \in E, x\mathcal{R}y \wedge y\mathcal{R}x \Rightarrow x = y$
- asymétrique si  $\forall x, y \in E, x\mathcal{R}y \Rightarrow \neg(y\mathcal{R}x)$
- complète si  $\forall x, y \in E, x\mathcal{R}y \vee y\mathcal{R}x$
- transitive si  $\forall x, y \in E, x\mathcal{R}y \wedge y\mathcal{R}z \Rightarrow x\mathcal{R}z$
- négativement transitive si  $\forall x, y \in E, \neg(x\mathcal{R}y) \wedge \neg(y\mathcal{R}z) \Rightarrow \neg(x\mathcal{R}z)$

**Definition 2. *Relation d'indifférence*** Etant donnée une relation  $\mathcal{R}$  sur un ensemble  $E$ , la relation d'indifférence noté  $\tilde{\mathcal{R}}$  est définie pour tout  $x, y \in E$  par  $\tilde{\mathcal{R}}$  si  $x\mathcal{R}y$  et  $y\mathcal{R}x$ .

**Definition 3. *Relation d'incompatibilité*** Etant donnée une relation  $\mathcal{R}$  sur un ensemble  $E$ , la relation d'incompatibilité notée  $||_{\mathcal{R}}$  est définie pour tout  $x, y \in E$  par :  $x||_{\mathcal{R}}y$  si  $\neg(x\mathcal{R}y)$  et  $\neg(y\mathcal{R}x)$

**Definition 4. *Relation d'ordre*** On appelle relation d'ordre sur un ensemble  $E$ , toute relation binaire sur  $E$  qui est réflexive, antisymétrique et transitive.

Si la relation d'ordre est complète, elle est dite ordre total sinon l'ordre est partiel.

**Definition 5. *Préordre*** Un préordre sur un ensemble  $E$  est une relation binaire réflexive et transitive.

À une relation d'ordre on peut associer une relation obtenue en ôtant de celle-ci les couples d'éléments identiques.

**Definition 6. (*Relation d'ordre strict*)** Une relation d'ordre strict sur un ensemble  $E$  est une relation binaire irreflexive et transitive. L'ordre strict est dit faible, s'il est partiel et négativement transitif.

### 3.2.2 Formulations rencontrés dans les modèles de préférence qualitatives

Nous appelons par structures les formes de représentations utilisées dans la constitution d'un modèle de préférence qualitatif.

Ainsi nous allons définir par la suite les formulations les plus couramment rencontrées.

1. **Formulation à base de formules logiques** Les travaux de [?] nous montrent comment peuvent être utilisées les formules logiques pour décrire les modèles de préférence. En effet dans ces travaux on parle de formules de préférences qui sont des formules logique de premier ordre appliquées à des couples de transactions afin de déterminer lequel est plus préféré. Elles sont sous la forme :

$$C(t_1, t_2) = \bigvee_{i=1, \dots, k} \left( \bigwedge_{j=1, \dots, k_i} f_{ij} \right)$$

2. **Formulations à base de contextes** Les préférences entre les objets peuvent varier en fonction du contexte dans lequel ils se trouvent. Par exemple lorsque l'on prend des films, dans le contexte où on a affaire à des films de comédie, on peut préférer les fins de certains acteurs alors que si on a affaire à des films d'action, on peut préférer des films ou jouent certains autres acteurs. Tout dépend du contexte qui est ici la catégorie du film.

Différents travaux ont recours à l'utilisation de contextes dans la définition de leur modèles. Ainsi nous pouvons parler :

- Des **CP-nets**. Soit  $\mathcal{A} = \{A_1, \dots, A_n\}$  un ensemble d'attributs d'objets. Un CP-nets sur  $\mathcal{A}$  est un graphe direct dans lequel il existe un noeud pour chaque attribut  $A_i$  de  $\mathcal{A}$ . Si pour deux attribut  $A_i, A_j$  on a une flèche du noeud de  $A_i$  vers le noeud de  $A_j$ , alors  $A_i$  est appelé un parent de  $A_j$ . L'ensemble de tous les parents d'un attribut  $A_i$  est noté  $P(A_i)$ . Pour chaque attribut  $A_i$ , son noeud dispose d'une table de préférence conditionnelle  $CPT(A_i)$  qui contient un ensemble de règles de préférence conditionnelles de la forme  $z_i : a_{i_1} \succ a_{i_2}$  où  $z_i \in \text{dom}(Pa(A_i))$  et  $a_{i_1}, a_{i_2}$  appartiennent à  $\text{dom}(A_i)$ . Ces règles signifient : étant donné  $z_1$  (dans notre cas ici c'est une forme de contexte),  $a_{i_1}$  est strictement préféré à  $a_{i_2}$  ceteris paribus (toute autre chose étant égale : c'est à quelque soient les autres valeurs fixées au niveau des attributs autre que  $A_i$  et  $P(A_i)$ ).

En des termes plus simples, cette règle de préférence indique qu'un tuple conte-

nant  $z_i$  et  $a_{i_1}$  est préféré à un tuple contenant  $z_i$  et  $a_{i_2}$  en considérant que les valeurs des tuples sur les autres attributs sont les mêmes.

#### Example 4

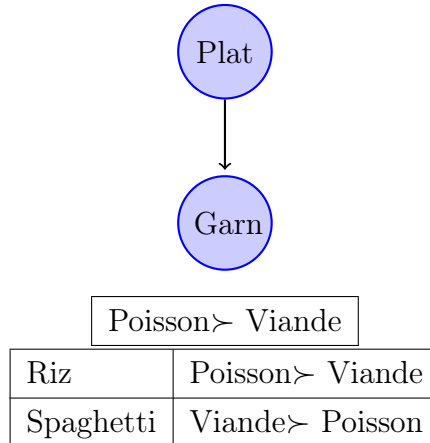
Cas des préférences au niveau des plats de riz. L'utilisateur préfère accompagner le riz au poisson plutôt qu'à la viande, et dans le cas du spaghetti, il préfère la viande au poisson.

Riz	Poisson $\succ$ Viande
Spaghetti	Viande $\succ$ Poisson

L'ensemble de ces tables de préférences conditionnelles vont former les CP-nets comme on peut le voir au niveau de l'exemple ci dessous.

#### Example 5

Meme genre que l'exemple précédent.



Soit un CP-nets  $N$  sur  $\mathcal{A}$ . Soit  $A_i$  un attribut de  $\mathcal{A}$  et  $Y = \mathcal{A} \setminus (Pa(A_i) \cup A_i)$ . Pour toute instance  $z_i$  de  $Pa(A_i)$  on associe l'ordre  $\succ_{z_i}^i$  induit sur  $dom(A_i)$  par la table de préférence  $CPT(X_i)$ .

Une relation de préférence  $\succ$  satisfait  $\succ_{z_i}^i$  ssi pour  $y \in dom(Y)$  et  $a_{i_1}, a_{i_2} \in dom(A_i)$ , si  $ya_{i_1}z_i \succ ya_{i_2}z_i$  alors  $a_{i_1} \succ_{z_i}^i a_{i_2}$ .

Une relation de préférence satisfait  $CPT(A_i)$  ssi elle satisfait  $\succ_{z_i}^i$  pour chaque  $z_i \in dom(Pa(A_i))$ .

Une relation de préférences  $\succ$  satisfait le CP-net  $N$  ssi elle satisfait  $CPT(A_i)$  pour chaque  $A_i$ .

Ainsi un modèle de préférence basé sur le CP-net est une relation de préférence qui satisfait le CP-net.

- Des **règles de préférence contextuelles** Une paire de transaction est une paire sous la forme  $\langle t_1, t_2 \rangle$  avec  $t_1$  et  $t_2$  des transactions caractérisant des objets. Cette paire décrit le fait que  $t_1$  est préféré à  $t_2$ . Nous les utiliseront pour décrire les règles de préférence contextuelles.

Les règles de préférence contextuelles ont été étudiées dans les travaux de [?],[?]. Elles ont une formulation similaire aux règles de préférence conditionnelles (dans les CP-nets) suivant leur formulation. Exemple avec des ensembles  $C, X, Y$  disjoints la règle  $C \rightarrow X \succ Y$  veut dire que pour toutes transactions  $t_1$  et  $t_2$  qui ont en commun le contexte  $C$ , et  $t_1$  contient  $X$  et ne contient pas  $Y$  et  $t_2$  contient  $Y$  et ne contient pas  $X$ , alors  $t_1$  est préféré à  $t_2$ . La différence avec les CP-nets est la manière dont les règles sont utilisées dans le modèle de préférence. En effet la caractéristique des règles de préférences contextuelles est qu'elles sont disposées sous forme de liste ordonnée dans les modèles de préférences. Différentes règles de tri peuvent être utilisées pour le tri de la liste. Par exemple la méthode de tri avec *best rule* permet de classer les règles suivant la valeur de leur confiance (taux de paires de transactions vérifiant la règle parmi les paires de transactions contenant le contexte de la règle) ensuite si la valeur de la confiance est égale elles sont triées suivant la valeur du support (taux de paires de transactions vérifiant la règle parmi toutes les transactions de la base).

3. **Formulations lexicographiques** Dans le modèle de préférence lexicographique, les préférences sont définies sur les attributs et sur un ordre d'importance entre les attributs. Ainsi si on a deux attributs munis de leur relation de préférence  $(A_i, \succ_i)$  et  $(A_j, \succ_j)$  de telle sorte que  $A_i$  est plus important que  $A_j$ , alors pour deux transactions  $t_1, t_2$ ,  $t_1 \succ t_2$  ssi  $(t_1.a_i \succ_i t_2.a_i) \vee (t_1.a_i = t_2.a_i \wedge t_1.a_j \succ_j t_2.a_j)$  avec  $t_1.a_k, t_2.a_k \in A_k, k = 1, 2$ .
4. **Modèle de pareto** Dans le modèle de préférence pareto c'est les préférences sur les attributs des objets qui sont mis en avant. Ainsi il y'a une relation de préférence  $\succ_i$  sur chaque attribut  $A_i$  de telle sorte que pour le modèle de préférence de pareto  $\succ$ ,  $t_1 \succ t_2$  ssi  $t_1.a_i \succ_i t_2.a_i$  avec  $t_1.a_i, t_2.a_i \in dm(A_i)$  pour tout attribut  $A_i$  de l'ensemble des attributs  $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$ . On note  $\succ \equiv \succ_1 \otimes \succ_2 \otimes \dots \otimes \succ_n$
5. **Formulations à base de constructeurs de base**

### 3.2.3 Compositions des modèles de préférence qualitatives

1. **Composition de préférence prioritaires** La composition de préférence prioritaire prend en compte la notion de priorités entre les relations de préférence. Ainsi si on a deux relations  $\succ_i$  et  $\succ_j$ , alors la préférence prioritaire  $\succ_i \& \succ_j$  est définie par : pour deux transactions  $t_1, t_2$ ,  $t_1 \succ_i \& \succ_j t_2 \Leftrightarrow (t_1 \succ_i t_2) \vee (t_1 \sim_i t_2 \wedge t_1 \succ_j t_2)$ .
2. **Composition de pareto** La composition de pareto de deux relations de préférences  $\succ_l$  et  $\succ_m$  est une composition définie par :

$$\forall t_i, t_j \in R, t_i \succ_l \otimes \succ_m t_j \text{ssi} (t_i \succ_l t_j \wedge \neg(t_j \succ_m t_i)) \vee (t_i \succ_m t_j \vee \neg(t_j \succ_l t_i))$$

### 3.2.4 Modèle lexicographique

### 3.2.5 Modèle à base de règles

### 3.2.6 Modèle à base de CPNets

Un CP-nets est une approche de représentation des préférences fondée sur une représentation graphique. Ceci se fait à l'aide de deux notions à savoir :

- la notion de préférence conditionnelle ;
- la notion de ceteris paribus.

Soit  $V = \{X_1, X_2, \dots, X_i\}$  La notion de préférence conditionnelle est d'indiquer pour chaque attribut  $X_i$  ses attributs parents  $Pa(X_i)$ , c'est à dire les autres attributs dont il dépend par rapport aux préférences de l'utilisateur. Ainsi lorsque l'on a une instantiation  $u$  de  $Pa(X_i)$ , on sait quel valeur de  $X_i$  est préférée à l'autre suivant  $u$ . La notion de ceteris paribus dans le cas d'une préférence conditionnelle est que pour un attribut  $X_i$ , le choix au niveau de cet attribut dépend de ses parents  $Pa(X_i)$  indépendamment des valeurs fixées au niveau des autres attributs  $Y = V - Pa(X_i) \cup X_i$ .

En se basant sur ces notions, nous formons au niveau de chaque attribut  $X_i$ , ce qu'on appelle **Table de Préférence Conditionnelle CPT** (sur la condition du ceteris paribus). Celle ci énumère les différentes préférences au niveau des valeurs de  $X_i$  suivant les valeurs des attributs de  $Pa(X_i)$ .

#### Exemple 6

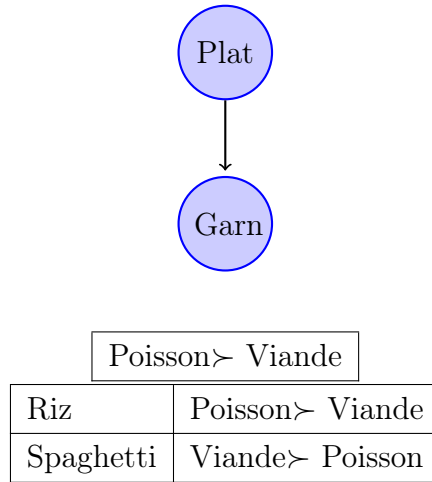
Cas des préférences au niveau des plats de riz. L'utilisateur préfère accompagné le riz au poisson plutôt qu'à la viande, et dans le cas du spaghetti, il préfère la viande au poisson.

Riz	Poisson $\succ$ Viande
Spaghetti	Viande $\succ$ Poisson

L'ensemble de ces tables de préférences conditionnelles vont former les CP-nets comme on peut le voir au niveau de l'exemple ci dessous.

### Example 7

Meme genre que l'exemple précédent.



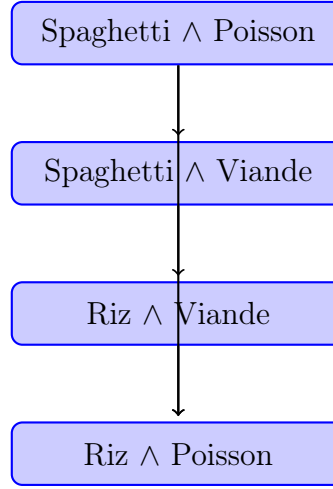
Ainsi nous définissons un CP-nets de la manière suivante :

**Definition 7.** *Un CP-net sur les attributs  $V = \{X_1, X_1, \dots, X_n\}$  est un graphe orienté sur les attributs  $X_1, X_1, \dots, X_n$  où chaque nœud  $X_i$  est annoté avec une table de préférence conditionnelle,  $CPT(X_i)$ . Chaque table de préférence conditionnelle  $CPT(A_i)$  associe un ordre total  $\succ_u^i$  pour chaque instanciation  $u$  de  $Pa(X_i)$ .*

A partir d'un CP-net on peut alors schématiser le graphe de préférence induit.

### Example 8

Le graphe de préférence induit par rapport à l'exemple précédent nous donne :



Sur cette figure, un arc partant d'un élément  $a$  vers un élément  $b$  indique que la préférence de  $b$  sur  $a$  put être déterminée directement par les CPTs du Cp-net. On voit que l'élément le plus en haut (Spaghetti  $\wedge$  Poisson) est l'élément le moins préféré tandis que l'élément le plus en bas (Riz  $\wedge$  Poisson) est l'élément le plus préféré.

### Comparaison de deux transaction suivant un CP-net

**Definition 8.** Prenons un CP-net  $N$  sur  $V$ ,  $X_i$  un attribut de  $V$  et  $Pa(X_i)$  les attributs parents de  $X_i$  dans  $V$ . Soit  $Y = V \setminus U \cup \{X_i\}$ . Soit  $\succ_u^i$  l'ordre induit par  $CPT(X_i)$  sur  $Dom(X_i)$  pour toute initialisation  $u \in Ass(U)$  de  $Pa(X_i)$ . Soit  $\succ$  une relation de préférences sur  $Asst(V)$ .

On peut dire qu'une relation de préférences  $\succ$  satisfait  $\succ_u^i$  ssi on a, pour tout  $y \in Asst(Y)$  et tout  $x, x' \in Dom(X_i)$ ,  $yxu \succ yx'u$  chaque fois que  $x \succ_u^i x'$ .

Une relation de préférence  $\succ$  satisfait  $CPT(X_i)$  ssi elle satisfait  $\succ_u^i$  pour tout  $u \in Ass(U)$ . Donc un CP-net  $N$  est satisfait par  $\succ$  ssi celui ci satisfait chacun des préférences conditionnelles exprimées dans les CPT de  $N$  suivent la notion de *ceteris paribus*.

**Definition 9.** Soit  $N$  un CP-net sur l'ensemble des attributs  $V$ . Soit  $a, b \in Asst(V)$ .  $N$  entraine  $a \succ b$  et noté  $N \models a \succ b$  ssi  $a \succ b$  pour toute relation de préférence qui satisfait  $N$ .

### 3.2.7 CPRefMiner

### 3.2.8 ProfileMining

## 3.3 Méthodes de construction des profils de préférence

### 3.3.1 Par la méthode de boosting

### 3.3.2 Par la méthode SVM

### 3.3.3 Par la descente de gradient

### 3.3.4 Par les l'extraction de motifs fréquents

---

## 4 Définition et extraction des règles de préférence contextuelle

Soit  $\mathcal{R}(R_1, R_2, \dots, R_n)$  un schémas relationnel tel que pour chaque attribut  $R_i$  on note son domaine de valeurs par  $Dom(R_i)$ . Ainsi notons  $Dom(\mathcal{R}) = Dom(R_1) \times Dom(R_2) \times \dots \times Dom(R_n)$ . On appelle une **transaction**, tout élément de  $Dom(\mathcal{R})$ . Soit  $\mathbf{I} = \bigcup_1^n Dom(R_i)$ , tout élément  $i$  tel que  $i \in \mathbf{I}$ , est appelé **item**. En prenant une transaction  $T = i_1, i_2, \dots, i_n$ , tout élément  $I$  tel que  $I \in T$  est appelé **itemset**. Une **base de transactions** est un ensemble de transactions, chacun associé à un identifiant unique. Une **paire de transactions**  $\langle T_1, T_2 \rangle$  est un vecteur de transactions tel que  $\langle T_1, T_2 \rangle \neq \langle T_2, T_1 \rangle$

### Exemple 9

Soit un schémas relationnel  $\mathcal{R}(\text{Genre}, \text{Acteur}, \text{Annee})$ . Leur domaines de valeurs peuvent être les suivants :  $Dom(\text{Genre}) = \{\text{Action}, \text{Aventure}, \text{Guerre}, \text{Comedie}\}$ ,  $Dom(\text{Acteur}) = \{\text{PierceBrozman}, \text{SylvesterStalone}, \text{HarrisonFord}\}$  et  $Dom(\text{Annee}) = [1900; 2020]$ .  $Dom(A) = Dom(\text{Genre}) \times Dom(\text{Acteur}) \times Dom(\text{Annee})$  Ainsi  $T = \{\text{Action}, \text{PierceBrozman}, 2010\}$  est une transaction de  $Dom(A)$ .  $T' = \{\text{Action}, 2010\} \in T$  est alors un itemset.

## 4.1 Préférences utilisateur

**Definition 10.** (*Préférence utilisateur*)

Une **préférence utilisateur** est une paire de transactions  $\langle T_1, T_2 \rangle$  qui spécifie que l'utili-



sateur préfère  $T_1$  à  $T_2$ . Elle est aussi écrite sous la forme  $T_1 \succ T_2$ .

**Definition 11.** (*Base de préférence*)

Soit  $\mathcal{D}$  un ensemble de transactions. Une **base de préférence**  $\mathcal{P} = \mathcal{D} \times \mathcal{D}$  d'un utilisateur est un ensemble de préférences utilisateur.

**Definition 12.** (*Règle de préférence contextuelle*)

Soit des itemsets  $C, X, Y$ . Une **règle de préférence contextuelle** est une représentation de la forme  $C \rightarrow X \succ Y$  signifiant que lorsque l'itemset  $C$  est observé, l'utilisateur préfère l'itemset  $X$  à l'itemset  $Y$ .

$C$  est appelé itemset contextuel,  $X$  est appelé itemset préféré et  $Y$  itemset non préféré

**Example 10**

La règle de préférence contextuelle  $\{Action\} \rightarrow \{1990\} \succ \{2000\}$  signifie que dans le cas de films d'action, l'utilisateur préfère les films de 1990 aux films de 2000.

La règle de préférence contextuelle  $\{MemeGenre\} \rightarrow \{Avant\ 2000\} \succ \{Après\ 2000\}$  signifie que dans le cas où deux films sont de même genre, l'utilisateur préfère les films produits au delà de l'année 2000.

**Definition 13.** Deux transactions sont **comparables** suivant la règle  $C \rightarrow X \succ Y$  si les deux contiennent  $C$  et si une des deux contient  $X$  et l'autre  $Y$ .

**Definition 14.** On dit qu'une préférence  $\langle T_1, T_2 \rangle$  **supporte** une règle de préférence contextuelle  $\mathcal{R} = C \rightarrow X \succ Y$  (noté  $R \vdash^+ \langle T_1, T_2 \rangle$ ) si  $C \in T_1 \cap T_2$ ,  $X \in T_1 \setminus T_1 \cap T_2$  et  $Y \in T_2 \setminus T_1 \cap T_2$ .

**Definition 15.** On dit qu'une préférence  $\langle T_1, T_2 \rangle$  **contredit** une règle de préférence contextuelle  $\mathcal{R} = C \rightarrow X \succ Y$  (noté  $R \vdash^- \langle T_1, T_2 \rangle$ ) si  $C \in T_1 \cap T_2$ ,  $Y \in T_1 \setminus T_1 \cap T_2$  et  $X \in T_2 \setminus T_1 \cap T_2$ .

**Example 11**

Avec  $T_1 = \{Action, Pierce Brosman, 1990\}$  et  $T_2 = \{Action, Sylvester Stalone, 2000\}$ , on constate que  $\langle T_1, T_2 \rangle$  supporte la règle de préférence  $\{Action\} \rightarrow \{1990\} \succ \{2000\}$ .

De même on dit qu'une transaction  $T_1$  est **préférée** à une transaction  $T_2$  suivant une règle contextuelle  $\mathcal{R} = C \rightarrow X \succ Y$  et noté  $T_1 \succ_{\mathcal{R}} T_2$  si si  $C \in T_1 \cap T_2$  (resp  $T_1$  et  $T_2$  vérifient  $C$ ) et si  $Y \in T_1 \setminus T_1 \cap T_2$  (resp seul  $T_1$  vérifie  $Y$ ) et  $X \in T_2 \setminus T_1 \cap T_2$  (resp seul  $T_2$  vérifie  $X$ ).

Soit une règle de préférence  $r$  et une base de données de préférences  $\mathcal{P}$ .  
Nous définissons par :

—  $agree(r, \mathcal{P})$  l'ensemble des préférences qui supportent  $r$  ;

$$agree(r, \mathcal{P}) = \{\langle T_i, T_j \rangle \in \mathcal{P} / R \vdash^+ \langle T_i, T_j \rangle\}$$

—  $contradict(r, \mathcal{P})$  l'ensemble des préférences qui contredisent  $r$  ;

$$contradict(r, \mathcal{P}) = \{\langle T_i, T_j \rangle \in \mathcal{P} / R \vdash^- \langle T_i, T_j \rangle\}$$

—  $cover(i, \mathcal{P})$  l'ensemble des préférences qui supportent et contredisent  $r$

$$cover(r, \mathcal{P}) = agree(r, \mathcal{P}) \cup contradict(r, \mathcal{P})$$

**Definition 16.** Le **support** d'une règle de préférence dans une base de préférence  $\mathcal{P}$  est le rapport entre le nombre de préférences qui supportent la règle de préférence et le nombre total des préférences de la base de préférence :

$$supp(r) = \frac{|agree(r, \mathcal{P})|}{|\mathcal{P}|}$$

**Definition 17.** La **confiance** d'une règle de préférence quand à elle est le taux de transactions vérifiant la règle par rapport à la somme du nombre de transactions vérifiant et du nombre de transactions contredisant la règle.

$$conf(r) = \frac{|agree(r, \mathcal{P})|}{|cover(r, \mathcal{P})|}$$

**Definition 18.** (Règle de préférence fréquente) Une règle de préférence  $r$  est dite fréquente par rapport à un seuil  $\sigma$  si  $\text{supp}(r) \geq \sigma$

**Definition 19.** (Règle de préférence intéressante) Une règle de préférence est dite intéressante pour un seuil de support  $\sigma$  et un seuil de confiance  $\delta$  si elle est fréquente par rapport au seuil  $\sigma$  et si elle a une confiance  $\text{conf}(r) \geq \delta$ .

**Definition 20.** (Règle de préférence minimale) Une règle de préférence contextuelle  $r = C \rightarrow X \succ Y$  est **minimale** par rapport à une base de préférences utilisateurs  $\mathcal{P}$  si et seulement s'il n'existe aucune règle  $r' = C' \rightarrow X' \succ Y'$  avec  $r \neq r'$ , tel que  $C' \subseteq C$ ,  $X' \subseteq X$  et  $Y' \subseteq Y$  avec  $\text{supp}_{\mathcal{P}}(r) = \text{supp}_{\mathcal{P}}(r')$   $\text{conf}_{\mathcal{P}}(r) = \text{conf}_{\mathcal{P}}(r')$ .

**Definition 21.** (Modèle de préférences) Un **modèle de préférence**  $\mathcal{M}_{\mathcal{P}}$  sur une base de préférence  $P$  est un ensemble trié et ordonné de règles de préférences contextuelles intéressantes minimales.

Dans le cadre d'un utilisateur, le modèle de préférence de la base de préférence de cet utilisateur est appelé **profil de préférences** de cet utilisateur. Ce profil de préférence est construit de façon à ce qu'il soit **précis** et **concis** par rapport à la base de préférences utilisateur. Le critère de concision est évalué par la cardinalité du profil de préférence (recherche de profils à faible cardinalité). Le critère de précision quand à lui est évalué par une **fonction de coût** que nous allons définir dans la suite de ce document.

On peut étendre les notations *agree*, *contradict* et *cover* à un ensemble de règles de préférences de la manière suivante :

- $\text{agree}(\Pi, \mathcal{P}) = \cup_{\pi \in \Pi} \text{agree}(\pi, \mathcal{P})$  ;
- $\text{contradict}(\Pi, \mathcal{P}) = \cup_{\pi \in \Pi} \text{contradict}(\pi, \mathcal{P})$  ;
- $\text{cover}(\Pi, \mathcal{P}) = \cup_{\pi \in \Pi} \text{cover}(\pi, \mathcal{P})$

**Definition 22.** (fonction coût) Etant donné une base de préférences  $\mathcal{P}$  et un ensemble de règles de préférences contextuelles  $\Pi$ , le coût de l'ensemble  $\Pi$  par rapport à  $\mathcal{P}$  noté  $\text{Cost}(\Pi, \mathcal{P})$  est défini par :

$$\text{cost}(\Pi, \mathcal{P}) = \frac{|\mathcal{P} \setminus \text{cover}(\Pi, \mathcal{P})| + |\text{contradict}(\Pi, \mathcal{P})|}{|\mathcal{P}|}$$

La précision d'un profil de préférence est caractérisée par la minimisation de la fonction coût définie ci dessus.

## 5 Utilisation des motifs séquentiels dans le cadre des préférences contextuelles (approche Sprex)

Les travaux antérieurs se sont penchés sur des méthodes d'extraction de connaissance afin de les exploiter pour extraire les profils de préférence. C'est dans cette logique que le travail effectué au niveau de (Giacometti et al.) a permis de proposer un mode d'extraction de profil utilisateur inspiré de l'extraction des motifs séquentiels. Ils ont nommé cette approche **Sprex** (Sequence-pattern based preference rule extraction) Il a été ainsi adopté de nouvelles représentation des préférences utilisateurs inspirées de la représentation des motifs séquentiels. Nous allons dans les sous sections suivantes décrire cette nouvelle représentation et les différentes étapes de Sprex qui permettent de construire les profils de préférence (à l'aide de son module nommé **Sprex-Build**) et prédire les préférences (à l'aide de son module nommé **Sprex-Predict**).

### 5.1 Représentation séquentielle des préférences

Soit  $I$  l'ensemble de tous les items. Un item de préférence est une paire  $L : i$  où  $L \in \{C, P, N\}$  est un label (avec  $C \equiv$  Contexte,  $P \equiv$  Préféré,  $N \equiv$  Non-préféré) et  $i$  un item de  $\mathcal{I}$ . Soient deux transactions  $T_1$  et  $T_2$  telles que  $T_1 \succ T_2$ . Dans leur cas, on définit par :

- **Itemset contextuel** l'ensemble d'items de préférence  $C = \{C : c_1, C : c_2, \dots, C : c_k\}$  tel que  $\{c_1, c_2, \dots, c_k\} = T_1 \cap T_2$
- **Itemset préféré** l'ensemble d'items de préférence  $P = \{P : x_1, P : x_2, \dots, P : x_k\}$  tel que  $\{x_1, x_2, \dots, x_k\} = T_1 \setminus T_2$
- **Itemset non préféré** l'ensemble d'items de préférence  $N = \{N : y_1, N : y_2, \dots, N : y_k\}$  tel que  $\{y_1, y_2, \dots, y_k\} = T_2 \setminus T_1$

Ces types d'itemsets sont appelés **itemsets de préférence**. La **base de séquences de préférence**  $\mathcal{P}$  correspondant à une **base de préférence** est l'ensemble des séquences de préférence qui correspondent à une préférence unique de la base de préférence. Dans ce cas on calcule le support *supp* d'une séquence de préférence  $s$  en comptant le nombre de

séquences de préférence  $s'$  de la base de séquence de préférence qui contiennent  $s$ .

$$supp(s) = \{s' \in \mathcal{P} / s \subset s'\}$$

On appelle **séquence de préférence** la séquence  $\langle CPN \rangle$  qui est une liste ordonnée des itemsets  $C, P, N$  représentant la transaction de préférence  $\langle T_1, T_2 \rangle$  avec  $T_1 = C \cup P$  et  $T_2 = C \cup N$ .

Une règle de préférence contextuelle  $C \rightarrow X \succ Y$  où  $C = \{c_1, c_2, \dots, c_k\}$ ,  $X = \{x_1, x_2, \dots, x_3\}$ ,  $Y = \{y_1, y_2, \dots, y_3\}$ , peut être écrite sous forme de séquence de préférence  $\langle C_C X_P Y_N \rangle$ .

Soit la paire de transactions  $\langle T_1, T_2 \rangle$  dont la séquence de préférence correspondante est  $\langle I_C I_P I_N \rangle$ . Ainsi :

$$C \rightarrow X \succ Y \text{ supporte } \langle T_1, T_2 \rangle \Leftrightarrow \langle C_C X_P Y_N \rangle \subseteq \langle I_C I_P I_N \rangle$$

$$C \rightarrow X \succ Y \text{ contredit } \langle T_1, T_2 \rangle \Leftrightarrow \langle C_C X_P Y_N \rangle \subseteq \langle I_C I_N I_P \rangle$$

Ces correspondances nous permettent de déduire que le support d'une règle de préférence  $C \rightarrow X \succ Y$  est égal au support de la séquence de préférence  $\langle C_C X_P Y_N \rangle$  qui lui correspond. De même le nombre de paires de préférence qui contredisent cette règle correspond au nombre de séquences de préférences qui supportent  $\langle C_C Y_P X_N \rangle$ .

## 5.2 Description de la construction du profil de préférence par Sprex-Build

Sprex-Build est la composante de Sprex dont le rôle est de permettre la construction d'un profil utilisateur (modèle de préférence)  $\mathcal{M}_{\mathcal{P}}$  à partir d'une base de préférences utilisateur  $\mathcal{P}$  en prenant en compte le seuil minimal de support  $\sigma$ , le seuil minimal de confiance  $\delta$  et en utilisant une fonction de modélisation  $\pi$  (choisie par l'utilisateur). Sprex build génère d'abord une base de séquence de préférences  $\mathcal{P}_S$  à partir de la base des préférences de l'utilisateur  $\mathcal{P}$  suivants les correspondances entre séquences de préférence et préférence qu'on a vu précédemment. Suite à cela, l'ensemble des séquences fréquentes de préférences  $\mathcal{F}$  est extrait de  $\mathcal{P}_S$  en sélectionnant toutes les séquences fréquentes dont le support  $supp$  est au dessus du seuil  $\sigma$  en utilisant n'importe quel algorithme d'extraction de motifs séquentiels. (dans le cas de Sprex c'est l'approche **Patterngrowth** qui est utilisée). Sprex-build calcule ensuite pour chaque séquence fréquente de préférence sa confiance  $conf_{\mathcal{P}_S}$  et l'ajoute au modèle  $\mathcal{M}$  à condition que  $conf_{\mathcal{P}_S} \geq \delta$ . Ensuite une fonction de

modélisation est utilisée pour construire le modèle de préférence final.

Voici c dessous ces étapes résumées sous forme d'algorithme.

---

**Algorithme 1 : Sprex-Build**

---

**Entrées :** l'ensemble des préférences de l'utilisateur  $\mathcal{P}$ , un seuil minimal de support  $\delta$ , un seuil minimal de confiance  $\sigma$  et une fonction de modélisation  $\pi$

**Sorties :** Modèle de règles de prférences  $\mathcal{M}_{\mathcal{P}}$

**début**

```

 $\mathcal{P}_S = \emptyset;$ 
pour chaque  $\langle T, U \rangle \in \mathcal{P}$  faire
     $C_c = \lambda_C(T \cap U), P_p = \lambda_P(T \setminus T \cap U), N_N = \lambda_N(U \setminus T \cap U);$ 
     $\mathcal{P}_S = \mathcal{P}_S \cup \langle C_C P_P N_N \rangle;$ 
 $\mathcal{F} = \text{FrequentSequenceMining}(\mathcal{P}_S, \sigma);$ 
 $\mathcal{M} = \emptyset;$ 
pour chaque  $s \in \mathcal{F}$  faire
    si  $\text{conf}_{\mathcal{P}_S} \geq \delta$  alors
         $\mathcal{M} = \mathcal{M} \cup s;$ 
 $\mathcal{M}_{\mathcal{P}} = \pi(\text{sort}(\mathcal{M}));$ 
Retourner  $\mathcal{M}_{\mathcal{P}};$ 
```

---

Nous avons décrit antérieurement ce que c'est qu'un profil utilisateur et qu'il se doit de vérifier les critères de précision et de concision. Pour vérifier ces critères, Sprex le construit de la manière suivante :

— **A compléter**

### 5.3 Description de l'approche utilisée pour la prédiction des prafarences utiliateurs (Sprex-Predict)

Sprex-Predict est le module de Sprex permettant d'effectuer des prédictions entre deux préférences. Nous le décrivons dans la suite du document.

Etant donné un modèle de préférences  $\mathcal{M}$ , une **fonction de préférences**  $\rho$  retourne un score  $c$  entre 0 et 1 qui prédit entre les transactions  $T$  et  $U$ , celle que l'utilisateur préfère en s'inspirant du modèle  $\mathcal{M}$ .

— Si  $c > 0.5$ , il est prédit que l'utilisateur va préférer  $T$  à  $U$  ;

- Si  $c < 0.5$ , il est prédit que l'utilisateur va préférer  $U$  à  $T$  ;
- Autrement Sprex-Predict retourne qu'il y'a une indécision au niveau du choix du plus préféré.

La fonction de préférence utilise des règles du modèle afin de déterminer quelle est la transaction la plus préférée. Une approche simple est d'utiliser la **meilleure règle**  $r_{best}$  du modèle qui permet de comparer  $T$  à  $U$ . Si  $T$  est préféré à  $U$  alors  $r_{best}$  retourne une valeur  $conf_{\mathcal{P}}(R_{best})$  ( $>0.5$ ), si  $U$  est préféré à  $T$ ,  $r_{best}$  retournera  $1 - conf_{\mathcal{P}}(r_b)$  ( $<0.5$ ), autrement dans le cas de l'indécision ( $T \sim U$ ),  $r_{best}$  va retourner la valeur 0.5.

Malheureusement cette fonction mène souvent à l'indécision suivant une étude de (de Amo et al. 2012) d'où Sprex-Predict utilise une fonction de préférence basée sur le **vote par valeur**. Il se décrit de la sorte : c'est sous forme d'une élection où les transactions d'un ensemble  $\mathcal{E}$  votent pour les candidats  $U$  ou  $T$  suivant  $r_{best}$  de telle façon que chacun accorde à chaque candidat la valeur retournée par  $r_{best}$ . Toutes ces valeurs sont sommées pour chaque candidat et celui ayant le plus haut score est le gagnant.

$$\rho_{vote} = \begin{cases} 1, si \sum_{V \in \mathcal{E}} \rho_{best}(\mathcal{M}, \langle T, V \rangle) > \sum_{V \in \mathcal{E}} \rho_{best}(\mathcal{M}, \langle U, V \rangle) \\ 0, si \sum_{V \in \mathcal{E}} \rho_{best}(\mathcal{M}, \langle T, V \rangle) < \sum_{V \in \mathcal{E}} \rho_{best}(\mathcal{M}, \langle U, V \rangle) \\ 0.5, si \sum_{V \in \mathcal{E}} \rho_{best}(\mathcal{M}, \langle T, V \rangle) = \sum_{V \in \mathcal{E}} \rho_{best}(\mathcal{M}, \langle U, V \rangle) \end{cases}$$

## 6 Extraction des préférences contextuelles dans notre approche

Le rôle de Sprex a consisté à exploiter l'efficacité des méthodes d'extraction des motifs séquentiels afin de l'appliquer à l'extraction des profils de préférence utilisateur. De même contrairement aux précédents travaux qui permettaient de comparer des items en fonction d'un contexte (représentation des règles limitées à la forme  $C \rightarrow i_1 \succ i_2$  avec  $i_1, i_2$  des items et  $C$  un itemset), Sprex a permit d'améliorer l'approche en permettant d'étendre la comparaison à des itemsets(ensemble d'items) en fonction d'un contexte (représentation des règles sous la forme  $C \rightarrow X \succ Y$  avec  $X, Y$  et  $C$  tous des itemsets). Cette extension des possibilités de comparaison, a permit l'enrichissement de l'expressivité des règles de préférences extraites et ainsi du profil utilisateur.

Suite à cela, différentes approches ont été étudiées afin d'enrichir encore plus cette expressivité des règles. C'est dans cette logique que s'est déroulé notre travail. Il a consisté

à rechercher une formulation améliorée des règles de préférences afin de contribuer à l'amélioration de cette expressivité. Au cours des travaux, l'approche que nous avons adopté s'est inspiré d'un travail de l'article (Chomicky et al..... à compléter) qui exploitait l'utilisation de prédicats logiques afin de représenter les préférences.

## 6.1 Formules de préférence suivant les travaux de ([?])

Une partie du travail de [?] a porté sur l'étude de la représentation des préférences utilisateur en s'inspirant des formules de prédicat du premier ordre. Dans leur travail le terme **relations de préférences** est utilisé similairement au terme **règle de préférence** de l'approche Sprex. Nous allons nous contenter de garder le terme règle de préférence. Nous allons décrire par la suite la formulation qu'ils ont proposé.

**Definition 23.** (*formule de préférence*) Une **formule de préférence**  $C(t_1, t_2)$  est une **formule de premier ordre** (formule de prédicats) qui est utilisée pour définir une règle de préférence (relation de préférence)  $\succ_C$  de telle sorte que  $t_1 \succ_C t_2 \equiv C(t_1, t_2)$ , c'est à dire la transaction  $t_1$  est préférée à la transaction  $t_2$  suivant la préférence  $\succ_C$  si et seulement si elles vérifient la formule de prédicat  $C(t_1, t_2)$ .

L'utilisation des formules de préférence nécessite de préciser le type d'attribut auquel les items (des transactions) appartiennent.

Ces formules de préférence sont écrites sous une **forme normale disjonctive**(DNF) excepté le fait que les quantificateurs ne sont pas utilisés contrairement aux formules des prédicats (dans la logique des prédicats). Ainsi elles se présentent comme suit :

$$C(t_1, t_2) = \bigvee_{i=1..k} ( \bigwedge_{j=1..l} f_{ij} )$$

où  $f_{ij} \forall i = 1..k, j = 1..l$  sont des **formules atomiques** (caractérisées par des contraintes de comparaison  $=, \neq, <, >, \leq, \geq$ ) sous la forme  $x\delta y$  ou  $x\delta c$  avec  $x$  et  $y$  des variables (d'un type d'attribut donné) qui correspondent respectivement à la première transaction  $t_1$  et à la deuxième transaction  $t_2$ .  $c$  quand à lui est une constante.

$\delta$  prend les valeurs

- $=, \neq$  dans le cas où les éléments  $x, y, c$  sont de type d'attribut quelconque (Exemple :  $x = y, x = a, y \neq a$ ),



- $\leq, \geq, <, >$  dans le cas où on a à faire à des types d'attributs numériques (Exemple :  $x < y, x < a, y \geq a$ ).

### Example 12

- Soit deux transactions  $t_1 = \{x_1, x_2, x_3\}$  et  $t_2 = \{y_1, y_2, y_3\}$  d'une instance de la relation  $R(\text{Plat}, \text{TypePlat}, \text{TypeVin})$ . On a qu'ils vérifient la formule de préférence  $C$  si et seulement si

$$\begin{aligned} C(t_1, t_2) \equiv & (x_1 = y_1 \wedge x_2 = \text{"fish"} \wedge y_2 = \text{"fish"} \wedge x_3 = \text{"white"} \wedge y_3 = \text{"red"}) \\ & \vee (x_1 = y_1 \wedge x_2 = \text{"meat"} \wedge y_2 = \text{"meat"} \wedge x_3 = \text{"red"} \wedge y_3 = \text{"white"}) \end{aligned}$$

est vérifié. Cette condition veut dire que dans le cas où c'est un plat à base de poisson, le client préfère le vin blanc au vin rouge et dans le cas où c'est un plat à base de viande, le client préfère le vin rouge au vin blanc.

- En prenant en compte les valeurs numériques, soit deux transactions  $t_1 = \{x_1, x_2, x_3\}$  et  $t_2 = \{y_1, y_2, y_3\}$  d'une instance de la relation  $R(\text{TypeFilm}, \text{Annee}, \text{Acteur})$ . On a  $t_1 \succ_C t_2$  ssi :

$$\begin{aligned} C(t_1, t_2) \equiv & (x_1 = y_1 \wedge x_2 > 2010 \wedge y_2 < 2000) \\ & \vee (x_1 = y_1 \wedge x_2 = y_2 \wedge x_3 = \text{"Sylvester Stalone"} \wedge y_3 = \text{"Pierce Brosnan"}) \end{aligned}$$

Cette règle veut tout simplement dire que le client préfère pour des films de même type, des films au delà de 2010 aux films produits avant 2000, et si les films sont de même type et de même année, il préfère ceux avec l'acteur Sylvester Stalone à ceux avec l'acteur Pierce Brosnan.

## 6.2 Formules de préférence suivant notre approche

Dans notre travail, nous allons nous inspirer de la représentation en formules de préférences décrite précédemment pour l'appliquer au cas des règles de préférence contextuelles. En effet nous voulons définir les formules de préférence de telle sorte qu'elles puissent représenter nos règles de préférence contextuelles. Dans la section précédente les formules de préférences ont été définies comme suit :

$$C(t_1, t_2) = \bigvee_{i=1..k} \left( \bigwedge_{j=1..l} f_{ij} \right)$$

Dans notre contexte, les formules de préférence vont seulement utiliser les connecteurs logiques  $\wedge$ . C'est à dire elles seront sous la forme :

$$P(t_1, t_2) = \bigwedge_{k \in E} P_{A_k}(t_1, t_2) \quad (1)$$

avec  $E \subset \{1, 2, \dots, n\}$  où  $n$  est le nombre des différents types d'attributs  $A_k$  qui définissent les items des transactions  $t_1, t_2$  et  $P_{A_k}(t_1, t_2), k \in E$  est une conjonction de formules atomiques définissant les items pour chaque attribut  $A_k, k \in E$  comme

### Exemple 13

Pour deux transactions  $t_1 = \{x_1, x_2\}$  et  $t_2 = \{y_1, y_2\}$ , on a que  $t_1 \succ_P t_2$  ssi  $P(t_1, t_2) = \{x_1 = y_1\} \wedge \{x_2 > 4\} \wedge \{y_2 < 3\}$  est vrai.

Après avoir défini les formules de préférence élémentaires, nous allons définir les formules de préférence.

**Definition 24.** Une *formule de préférence élémentaire* est une conjonction de formules atomiques (définies dans [?]) construite à partir de la comparaison de deux items  $x, y$  de même type d'attribut.

### Exemple 14

Si on a deux items  $x$  et  $y$  de type année de valeurs 2000 et 2002, alors par comparaison, on obtient les formules de préférence élémentaires  $x < y, x \neq y, x < 2003 \wedge y < 2003, x > 1999 \wedge y > 1999$  etc..

Nous fournissons ci dessous les types de formules de préférences élémentaires que nous allons prendre en compte dans notre travail. Soit  $\mathcal{F}$  l'ensemble des formules de préférence élémentaires issues de la comparaisons entre deux items de  $t_1$  et de  $t_2$  ayant un même type d'attribut  $A_k$  donné.

Les formules de préférences élémentaires qui peuvent être déduites sont :

- Cas où les items de type  $A_k$  ont des valeurs numériques. Soit un item  $x$  de  $t_1$  de type  $A_k$  et  $y$  un item de  $t_2$  de type  $A_k$ .
- Si  $x$  et  $y$  sont tous deux égaux à une valeur  $a$  alors les formules de préférences élémentaires extraites sont  $x = y$  et  $x = y \wedge x = a \wedge y = a$ .

- Si  $x$  est différent de  $y$  alors la formule de préférence élémentaire  $x \neq y$  est extraite.
- Si  $x < y$  alors la formule de préférence élémentaire  $x < y$  est extraite.
- Si  $x > y$  alors la formule de préférence élémentaire  $x > y$  est extraite.
- Soit  $a$  appartenant à l'ensemble des valeurs possibles des items de type  $A_k$ ,
  - $x < a$  et  $y < a$  alors la formule de préférence élémentaire  $x < a \wedge y < a$  est extraite.
  - $x > a$  et  $y > a$  alors la formule de préférence élémentaire  $x > a \wedge y > a$  est extraite.
  - $x < a$  et  $y \geq a$  alors la formule de préférence élémentaire  $x < a \wedge y \geq a$  est extraite
  - $x > a$  et  $y \leq a$  alors la formule de préférence élémentaire  $x > a \wedge y \leq a$  est extraite.
  - $y > a$  et  $x \leq a$  alors la formule de préférence élémentaire  $y > a \wedge x \leq a$  est extraite.
  - $y < a$  et  $x \geq a$  alors la formule de préférence élémentaire  $y < a \wedge x \geq a$  est extraite.
  - $x = a$  et  $y \neq a$  alors la formule de préférence élémentaire  $x = a \wedge y \neq a$  est extraite.
  - $y = a$  et  $x \neq a$  alors la formule de préférence élémentaire  $y = a \wedge x \neq a$  est extraite.
- Cas où pour un type d'attribut donné, les items prennent des valeurs symboliques.
  - Si  $x$  et  $y$  sont tous deux égaux à une valeur  $a$  alors les formules de préférence élémentaires  $x = y$  et  $x = y \wedge x = a \wedge y = a$  sont extraites ;
  - Si  $x$  est différent de  $y$  alors la formule de préférence élémentaire  $x \neq y$  est extraite.

### 6.2.1 Séquences de formules de préférence élémentaires

Avant d'aller plus en détail, nous faisons un rappel sur ce que sont les relations binaires symétriques et les relations binaires asymétriques.

**Definition 25.** (*relation binaire symétrique*) Une relation binaire  $\mathcal{R}$  est symétrique si :

$$\forall (x, y) \in E^2, x\mathcal{R}y \Rightarrow y\mathcal{R}x$$

**Definition 26.** (*relation binaire asymétrique*) Une relation binaire  $\mathcal{R}$  est dite asymétrique si :

$$\forall (x, y) \in E^2, x\mathcal{R}y \Rightarrow \neg y\mathcal{R}x$$

**Definition 27.** (*Formules de préférence élémentaires symétriques*) Les formules de préférence élémentaires symétriques  $C(t_1, t_2)$  sont caractérisées par la propriété de symétrie c'est à dire  $C(t_1, t_2) \Rightarrow C(t_2, t_1)$ .

**Example 15**

la formule  $F(x, y) \equiv (x = y)$  est symétrique car  $F(x, y) \equiv (x = y) \Rightarrow (y = x) \equiv F(y, x)$ , d'où  $F(x, y) \Rightarrow F(y, x)$ .

**Definition 28.** (*Formules de préférence élémentaires asymétriques*) Les formules de préférence élémentaires asymétriques  $P(t_1, t_2)$  (ou resp  $N(t_1, t_2)$ ) sont caractérisées par la propriété d'asymétrie c'est à dire  $P(t_1, t_2) \Rightarrow \neg P(t_2, t_1)$  (ou resp  $N(t_1, t_2) \Rightarrow \neg N(t_2, t_1)$  ).

**Example 16**

la formule  $F(x, y) \equiv (x > y)$  est asymétrique car  $F(x, y) \equiv (x > y) \Rightarrow \neg(y > x \vee y = x) \Rightarrow \neg(y > x) \wedge \neg(x = y) \Rightarrow \neg(y > x) \equiv \neg F(y, x)$ , d'où  $F(x, y) \Rightarrow \neg F(y, x)$ .

Nous nous rappelons que l'approche Sprex a décrit les séquences de préférences comme étant un itemset composé de trois parties à savoir une partie *contextuelle* qui rassemble les items communs à  $t_1$  et  $t_2$  une la partie *préférée* distinguant les items qui sont dans  $t_1$  et pas dans  $t_2$  et une partie *non préférée* distinguant les items qui sont dans  $t_2$  et pas dans  $t_1$ .

Dans notre cas où i s'agit de séquences de formules de préférences élémentaires, nous ne parlons pas d'items mais de formules de préférences élémentaires au niveau de la composition des séquences de préférences élémentaires.

Ainsi nous définissons différamment la composition de ces séquences de formules de préférence élémentaires. Elles sont décomposées en deux parties : une partie rassemblant les formules de préférences élémentaires symétriques, appelée contexte, et une partie rassemblant les formules de préférence élémentaires asymétriques, appelées préférence.

Cette approche s'explique par le fait que la partie contexte regroupe les formules vérifiées autant par la paire de transaction  $\langle t_1, t_2 \rangle$  que  $\langle t_2, t_1 \rangle$  (ce sont les formules de préférence élémentaires symétriques). La partie préférence permet de spécifier les formules de préférence élémentaires qui permettent de différencier la paire  $\langle t_1, t_2 \rangle$  de la paire  $\langle t_2, t_1 \rangle$  (ce sont les formules de préférence élémentaires asymétriques).

Ainsi nous définissons une séquence de formules de préférences élémentaires comme suit :

**Definition 29.** (*Séquence de formules de préférence élémentaire*) Pour une paire de transaction  $\langle t_1, t_2 \rangle$ , la séquence de formules de préférence élémentaires est la suite de préférences élémentaires  $\langle C, PN \rangle$  tel que

- $C \equiv \{F(t_1, t_2)/F(t_1, t_2) \in \mathcal{F}(t_1, t_2) \text{ et symétrique}\}$
- $PN \equiv \{F(t_1, t_2)/F(t_1, t_2) \in \mathcal{F}(t_1, t_2) \text{ et asymétrique}\}$

avec  $\mathcal{F}(t_1, t_2)$  qui est l'ensemble des formules de préférence élémentaires extraites de la comparaison entre les items de  $t_1$  et de  $t_2$ .

Voici un diagramme répertoriant les formules de préférence élémentaires réparties suivant qu'elles soient symétriques ou asymétriques :

$$\begin{array}{l} \text{Formules de préférence élé-} \\ \text{mentaires symétriques} \\ \text{(Contexte)} \end{array} \quad \left\{ \begin{array}{l} C : \{x = y\} \quad \equiv \quad x = y \\ C : \{x \neq y\} \quad \equiv \quad x \neq y \\ C : \{x, y = a\} \quad \equiv \quad x = y \wedge x = a \wedge y = a \\ C : \{x, y < a\} \quad \equiv \quad x < a \wedge y < a \\ C : \{x, y > a\} \quad \equiv \quad x > a \wedge y > a \end{array} \right.$$

$$\begin{array}{l} \text{Formules de préférence} \\ \text{élémentaires} \\ \text{asymétriques} \\ \text{(Préférences)} \end{array} \quad \left\{ \begin{array}{l} P : \{x < y\} \quad \equiv \quad x < y \\ P : \{x > y\} \quad \equiv \quad x > y \\ P : \{x = a\} \quad \equiv \quad x = a \wedge x \neq y \\ P : \{x < a\} \quad \equiv \quad x < a \wedge y \geq a \\ P : \{x > a\} \quad \equiv \quad x > a \wedge y \leq a \\ N : \{y > a\} \quad \equiv \quad y > a \wedge x \leq a \\ N : \{y < a\} \quad \equiv \quad y < a \wedge x \geq a \\ N : \{y = a\} \quad \equiv \quad y = a \wedge x \neq y \end{array} \right.$$

Dans ce diagramme, nous avons à droite des équivalences ( $\equiv$ ), les formules de préférences élémentaires et à gauche nous avons leur notation ( $C : \{\dots\}, P : \{\dots\}, N : \{\dots\}$ ) qui sera adoptée lors de leur représentation au niveau des séquences de formules de préférence.

Nous pouvons à présent définir les formules de préférence qui vont caractériser les règles de préférence contextuelles qu'on aura à extraire.

**Definition 30.** (*Formules de préférence*)

Une formule de préférence définissant une règle de préférence suivant est une formule qui s'écrit sous la forme :

$$\mathcal{P}(t_1, t_2) = C(t_1, t_2) \wedge PN(t_1, t_2)$$

et qui équivaut à une règle de préférence contextuelle  $C \rightarrow PN$  avec

$$\begin{aligned} C(t_1, t_2) &= \bigwedge_{k \in E} C_{A_k}(t_1, t_2) \text{ conjonction de formules de préférence élémentaires symétriques} \\ PN(t_1, t_2) &= \bigwedge_{k \in E} PN_{A_k}(t_1, t_2) \text{ conjonction de formules de préférence élémentaires asymétriques} \end{aligned}$$

Ici  $PN_{A_k}(t_1, t_2)$  est soit une formule de type  $P_{A_k}(t_1, t_2)$  ou  $N_{A_k}(t_1, t_2)$ .

$C(t_1, t_2)$  sera définie comme la **partie contexte** de la formule de préférence contex-

tuelle et  $PN(t_1, t_2)$  sera définie comme la **partie préférence** de la formule de préférence contextuelle.

Une formule de préférence  $\mathcal{P}(t_1, t_2) = C(t_1, t_2) \wedge PN(t_1, t_2)$  peut être écrite sous forme d'une séquence de formule de préférences  $\langle \lambda(C), \lambda(PN) \rangle$  avec  $\lambda(C)$  est l'ensemble des formules de préférence élémentaires constituant  $C(t_1, t_2)$  et  $\lambda(PN)$  est l'ensemble des formules de préférence élémentaires constituant  $PN(t_1, t_2)$ .

**Condition de non redondance :**

Il est à noter que les  $P_{A_k}$  ne doivent pas contenir de redondance i.e si on considère  $F_k$  l'ensemble des formules élémentaires constituant  $P_{A_k}$  ( $P_{A_k} = \bigwedge_{f \in F_k} f$ ), on a

$$F' \subsetneq F \Rightarrow \neg(\bigwedge_{f \in F'} f \Rightarrow P_{A_k}).$$

Le tableau ci dessous résume les différents cas évoqués en ce qui concerne les formules de préférence élémentaires :

	Valeurs Attr	Type Attr	Items de transaction de préférence
Attr monovalué	$x = a, y = a$	num/symb	$C : \{x = y\}$ $C : \{x, y = a\}$
		num	$C : \{x, y < c\}, \forall c   c < a$ $C : \{x, y > c\}, \forall c   c > a$
Attr monovalué	$x = a, y = b$ avec $a \neq b$	num/symb	$P : \{x = a\}$ $N : \{y = b\}$ $C : \{x \neq y\}$
		num	$P : \{x < y\}, \text{ si } a < b$ $P : \{x > y\}, \text{ si } a > b$ $P : \{x < c\}, \text{ si } a < b \text{ et } \forall c   a < c \leq b$ $P : \{x > c\}, \text{ si } a > b \text{ et } \forall c   a > c \geq b$ $N : \{y < c\}, \text{ si } a > b \text{ et } \forall c   b < c \leq a$ $N : \{y > c\}, \text{ si } a < b \text{ et } \forall c   b > c \geq a$ $C : \{x, y < c\}, \forall c   c < a \wedge c < b$ $C : \{x, y > c\}, \forall c   c > a \wedge c > b$
Attr multivalué	$x \in X, y \in Y$	Set(symb)	$C : \{x, y = c\}, \forall c \in X \cap Y$ $P : \{x = a\}, \forall a \in X \setminus Y$ $N : \{y = b\}, \forall b \in Y \setminus X$

### 6.3 Construction du profil utilisateur

Après avoir défini ce qu'est une séquence de formules de préférences, nous pouvons à présent utiliser cette formulation pour construire le profil utilisateur. En effet nous nous inspirons de Sprex afin de construire ce profil. Nous utilisons l'algorithme de Sprex-Build tout en remplaçant l'étape qui permet la conversion des paires de transactions en séquences de préférence, par l'étape qui permettra la conversion des paires de transactions en séquence de formules de préférence.

Voici l'algorithme Sprex-Build-Formula de construction du profil obtenu dans ce cas (qui est une modification de Sprex-Build) :



---

**Algorithme 2 : Sprex-Build-Formula**

---

**Entrées :** l'ensemble des préférences de l'utilisateur  $\mathcal{P}$ , un seuil minimal de support  $\delta$ , un seuil minimal de confiance  $\sigma$  et une fonction de modélisation  $\pi$

**Sorties :** Modèle de règles de préférences  $\mathcal{M}_{\mathcal{P}}$

**début**

```
 $\mathcal{P}_S = \emptyset;$ 
pour chaque  $\langle T, U \rangle \in \mathcal{P}$  faire
   $\mathcal{P}_S = \mathcal{P}_S \cup \text{SequenceFormula}(\langle T, U \rangle)$ 
 $\mathcal{F} = \text{FrequentSequenceMining}(\mathcal{P}_S, \sigma);$ 
 $\mathcal{M} = \emptyset;$ 
pour chaque  $s \in \mathcal{F}$  faire
  si  $\text{conf}_{\mathcal{P}_S} \geq \delta$  alors
     $\mathcal{M} = \mathcal{M} \cup s;$ 
 $\mathcal{M}_{\mathcal{P}} = \pi(\text{sort}(\mathcal{M}));$ 
Retourner  $\mathcal{M}_{\mathcal{P}};$ 
```

---

L'algorithme Sprex-Builds-Formula utilise la fonction `SequenceFormula()` qui permet la transformation d'une paire de transaction  $\langle T, U \rangle$  en séquences de formules de préférences. Cette fonction est définie comme suit :

---

**Algorithme 3 : SequenceFormula**

---

**Entrées :** Paire de transactions  $\langle T, U \rangle$

**Sorties :** Séquence de formules de préférence  $S$

**début**

```
 $S = \emptyset;$ 
pour chaque Type d'attribut  $A_k$  faire
  pour chaque paire d'items  $(x, y) \in T \times U$  de type  $A_k$  faire
     $S = S \cup \text{CompareFormula}(x, y, A_k)$ 
```

---

`CompareFormula` quand à elle est la fonction qui effectue la comparaison entre deux items respectivement de  $T$  et  $U$  afin de générer les formules de préférences élémentaires

correspondantes. Elle est définie comme suis :

---

**Algorithme 4 : CompareFormula**

---

**Entrées :**  $x$  ,  $y$  et l'attribut  $A_i$  auquel ils sont liés

**Sorties :** items issus de la comparaison entre  $x$  et  $y$

**début**

$\mathcal{E} = \emptyset$  ; // Ensemble des items générés

**si**  $x = y$  **alors**

        Ajouter  $P : \{x_{A_i} = y_{A_i}\}$  à  $\mathcal{E}$ ;

        Ajouter  $P : \{x_{A_i} = y_{A_i} = x\}$  à  $\mathcal{E}$ ;

**pour chaque**  $c \in DomActif(A_i)$  **et**  $A_i$  *numérique* **faire**

**si**  $x < c$  **alors**

                Ajouter  $C : \{x_{A_i}, y_{A_i} < c\}$  à  $\mathcal{E}$ ;

**sinon si**  $x > c$  **alors**

                Ajouter  $C : \{x_{A_i}, y_{A_i} > c\}$  à  $\mathcal{E}$ ;

**sinon si**  $x \neq y$  **alors**

        Ajouter  $C : \{x_{A_i} \neq y_{A_i}\}$  à  $\mathcal{E}$ ;

        Ajouter  $P : \{x_{A_i} = x\}$  à  $\mathcal{E}$ ;

        Ajouter  $N : \{y_{A_i} = y\}$  à  $\mathcal{E}$ ;

**si**  $A_i$  *numérique* **alors**

**si**  $x > y$  **alors**

                Ajouter  $P : \{x_{A_i} > y_{A_i}\}$  à  $\mathcal{E}$ ;

**si**  $x < y$  **alors**

                Ajouter  $P : \{x_{A_i} < y_{A_i}\}$  à  $\mathcal{E}$ ;

**pour chaque**  $c \in DomActif(A_i)$  **faire**

**si**  $x < c \wedge y \geq c$  **alors**

                    Ajouter  $P : \{x_{A_i} < c\}$  à  $\mathcal{E}$ ;

**sinon si**  $x \leq c \wedge y > c$  **alors**

                    Ajouter  $N : \{y_{A_i} > c\}$  à  $\mathcal{E}$ ;

**sinon si**  $x > c \wedge y \leq c$  **alors**

                    Ajouter  $P : \{x_{A_i} > c\}$  à  $\mathcal{E}$ ;

**sinon si**  $x \geq c \wedge y < c$  **alors**

                    Ajouter  $N : \{x_{A_i} < c\}$  à  $\mathcal{E}$ ;

**sinon si**  $x < c \wedge y < c$  **alors**

                    Ajouter  $C : \{x_{A_i}, y_{A_i} < c\}$  à  $\mathcal{E}$ ;

**sinon si**  $x > c \wedge y > c$  **alors**

                    Ajouter  $C : \{x_{A_i}, y_{A_i} > c\}$  à  $\mathcal{E}$ ;

    retourner  $\mathcal{E}$ ;

---

Cette fonction est équivalente à la description qu'on avait faite antérieurement sur les formules de préférence élémentaires qui peuvent être générées par la comparaison de deux items.

Les différentes étapes de la construction de profils de préférence étant décrites, on peut décrire la phase de prédiction des préférences utilisateur à partir de ce profil obtenu.

## **6.4 Prédiction des préférences utilisateur**

La prédiction des préférences de l'utilisateur peut être faite à l'aide du profil construit précédemment. Nous voyons que le profil qui est un ensemble de règles de préférences représentées sous forme de séquences de formules de préférence, nécessite que le couple de transaction à comparer soit d'abord converti en séquences de formules de préférences. Ensuite on peut utiliser l'approche Sprex-Predict pour déterminer la préférence entre ce couple de transactions.

Notre travail décrit, nous pouvons passer à la phase d'implémentation afin de permettre une évaluation de performance de l'approche adoptée.

## **6.5 Implémentation du travail**

L'implémentation du travail a consisté à mettre en place un programme capable de :

- Construire les profils utilisateurs basés sur les séquences de formules de préférence ;
- Evaluer la qualité de prédiction des profils utilisateurs.

Afin de le réaliser, nous avons exploité deux programmes qui avaient été développés dans le cadre des recherches antérieures.

- Le premier nommé Sprex a été développé afin d'implémenter l'algorithme de Sprex (Sprex-Build et Sprex-Predict) ;
- Le second est un programme surnommé Colico qui s'est basé sur un algorithme d'extraction de motifs séquentiels minimaux.

### 6.5.1 Le programme Sprex

Sprex est un programme qui a été développé pour le travail de l'aticle [?]. Les fonctionnalités principales sont :

- L'extraction de toutes les règles de préférence contextuelles ;
- L'extraction des règles de préférence contextuelles intéressantes minimales. Ceci est fait en précisant les valeurs seuils du support et de la confiance.
- La construction d'un model de préférences dépendant d'une fonction de modélisation. En effet c'est la fonction de modélisation qui permet de filtrer et trier les règles de préférence contextuelles minimales afin de construire le modèle de préférences ;
- La prédiction les préférences utilisateurs dépendant d'une fonction de préférence. En effet la fonction de préférence permet d'exploiter le modèle de préférence pour comparer un couple de transactions de préférences.
- Le test de la qualité de ces prédictions des préférences utilisateurs qui sont réalisées. Plusieurs indicateurs sont évalués pour déterminer la qualité de prédiction. Exemple : la précision, le Rappel etc..

Initialement dans notre travail, il a été prévu d'utiliser le programme Sprex afin d'extraire les règles de préférence intéressantes minimales, de construire ensuite le modèle de préférence à partr de ces règles et d'évaluer la qualité de la prédiction du modèle de préférence obtenu.

Mais nous avons été confrontés à un problème de saturation mémoire lors de la phase d'extraction des règles intéressantes minimales à partir des séquences de formules de préférence. Ceci est dû au fait que Sprex a été utilisé pour le traitement des séquences de préférences qui ont une taille plus réduite que les séquences de formules de préférence formalisées dans notre travail, alors que plus la taille des séquence est longue, plus l'extraction de motifs minimaux prend de l'espace lorsqu'on est confronté à un algorithme d'extraction de motifs minimaux qui effectue l'extraction par niveau.

Ainsi nous nous avons choisit autre programme nommé Colico pour réaliser la phase d'extraction de règles de préférences minimales intéressantes.

### 6.5.2 Le programme Colico

Colico est un programme qui a été développé par Arnault Soulet (en c++) afin de répondre au problème de saturation dû à l'extraction de motifs minimaux. Il se base sur les travaux de l'article [?] qui a proposé une méthode d'extraction de motifs minimaux avec l'avantage d'être polynomial-delay et polynomial-space.

**Principe de l'approche de Colico** Nous allons tout d'abord énoncer quelques fondements sur lequel se base l'algorithme de Colico.

**Definition 31.** (*Opérateur de couverture*) Soit un ensemble d'objets  $\mathcal{O}$ , un opérateur de couverture  $cov : 2^E \rightarrow 2^{\mathcal{O}}$  satisfait  $cov(X \cup Y) = cov(X) \cap cov(Y)$  pour tout  $X, Y \in 2^E$

**Definition 32.** (*Opérateur canonique*) Etant donné deux systèmes d'ensembles  $(\mathcal{F}, E)$  et  $(\mathcal{G}, E)$ , un opérateur canonique  $\phi : \mathcal{F} \cup \mathcal{G} \rightarrow \mathcal{F}$  est une fonction satisfaisant : (i)  $X \in Y \Rightarrow \phi(X) \in \phi(Y)$  et (ii)  $X \in \mathcal{F} \Rightarrow \phi(X) = X$  pour tout les ensembles  $X, Y \in \mathcal{G}$ .

**Definition 33.** (*Système minimisable d'ensembles*)

Un système minimisable d'ensembles est un tuple  $\langle (\mathcal{F}, \mathcal{E}, \mathcal{G}, cov, \phi) \rangle$  où :

- $(\mathcal{F}, E)$  est un système d'ensembles fini et fortement accessible. Un ensemble acceptable dans  $\mathcal{F}$  est appelé motif.
- $(\mathcal{G}, E)$  est un système d'ensembles fini et fortement accessible satisfaisant pour chaque couple d'ensembles acceptables  $X, Y \in \mathcal{F}$  tels que  $X \subseteq Y$  et chaque élément  $e \in E$ ,  $X \setminus \{e\} \in \mathcal{G}$ . Un ensemble acceptable de  $\mathcal{G}$  est appelé généralisation.
- $cov : 2^E \rightarrow 2^{\mathcal{O}}$  est un opérateur de couverture.
- $\phi : \mathcal{F} \cup \mathcal{G} \rightarrow \mathcal{F}$  est un opérateur canonique tel que pour chaque ensemble acceptable  $X \in \mathcal{G}$ , on ait  $cov(\phi(X)) = cov(X)$ .

**Definition 34.** (*Motif minimal*) Un motif  $X$  est minimal pour  $\langle (\mathcal{F}, E), \mathcal{G}, cov, \phi \rangle$  ssi  $X \in \mathcal{F}$  et pour chaque généralisation  $Y \in \mathcal{G}$  telle que  $Y \subset X$ ,  $cov(Y) \neq cov(X)$ .  $\mathcal{M}(\mathcal{S})$  dénote l'ensemble des motifs minimaux.

**Theorem 6.1.** (*Représentation condensée*) L'ensemble des motifs minimaux est une représentation condensée de  $\mathcal{F}$  adéquate pour le calcul de  $cov$  : pour tout motif  $W \in \mathcal{F}$ , il existe  $Y \in X$  tel que  $\phi(Y) \in \mathcal{M}(\mathcal{S})$  et  $cov(\phi) = cov(X)$ .

**Theorem 6.2.** (*Système indépendant*) Si un motif  $X$  est minimal pour  $S$ , alors tout motif  $Y \in \mathcal{F}$  satisfaisant  $Y \subseteq X$  est aussi minimal pour  $S$ .

**Lemma 6.3.** *Si  $X$  n'est pas minimal,  $\exists e \in X$  tel que  $X \setminus \{e\} \in \mathcal{G}$  et  $cov(X) = cov(X \setminus \{e\})$ .*

**Definition 35.** *(Objets critiques) Pour un motif  $X$ , les objets critiques d'un élément  $e \in X$ , notés  $\widehat{cov}(X, e)$  sont l'ensemble d'objets qui appartiennent à la couverture de  $X$  privé de  $e$  mais pas à la couverture de  $e$  :  $\widehat{cov}(X, e) = cov(X \setminus e) \setminus cov(e)$ .*

**Property 6.3.1.** *(Minimalité)  $X \in \mathcal{F}$  est minimal si  $\forall e \in X$  tel que  $X \setminus e \in \mathcal{G}$ ,  $\widehat{X}, e \neq \emptyset$*

**Property 6.3.2.** *L'égalité suivante est vraie pour tout motif  $X \in \mathcal{F}$  et deux éléments  $e, e' \in E$  :*

$$\widehat{cov}(Xe', e) = \widehat{cov}(X, e) \cap cov(e').$$

### Algorithme DEFME

---

#### Algorithme 5 : DEFME

---

**Entrées :**  $X$  est un motif,  $tail$  est l'ensemble des items restant à utiliser pour générer les candidats. Valeurs initiales :  $X = \emptyset$ ,  $tail = E$ .

**Sorties :** Calcul des motifs minimaux de manière incrmentale polynomiale

**début**

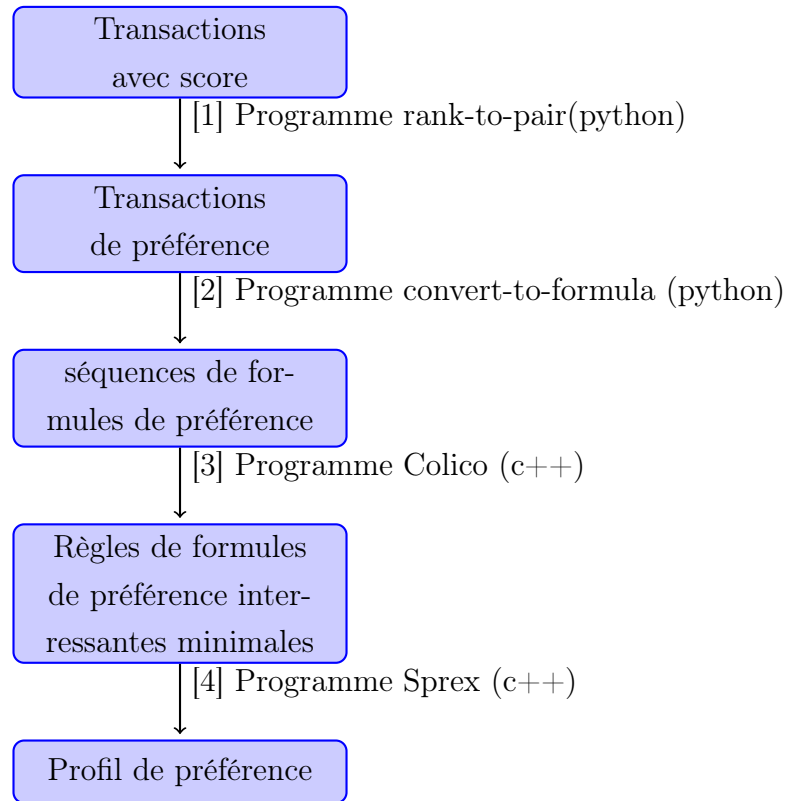
```

    si  $\forall e \in X, \widehat{cov}(X, e) \neq \emptyset$  alors
        print  $X$ ;
        pour chaque  $e \in tail$  faire
            si  $Xe \in \mathcal{F}$  alors
                 $tail := tail \setminus \{e\}$ ;
                 $Y := Xe$ ;
                 $cov(Y) := cov(X) \cap cov(e)$ ;
                 $\widehat{cov}(Y, e) := cov(X) \setminus cov(e)$ ;
                pour chaque  $e' \in X$  faire
                     $\widehat{cov}(Y, e') := \widehat{cov}(X, e') \cap cov(e)$ ;
                DEFME( $Y, tail$ );

```

---

L'extraction et la prédiction des préférences utilisateur a été réalisé avec des programmes écrits en c++ et python. Ainsi le côté implémentation des traitements s'est déroulé en plusieurs étapes comme le décrit le schémas ci dessous :



Nous disposions initialement d'une base de transactions utilisateur accompagné de leur scores représentant le degré de préférence de l'utilisateur. Nous devions alors convertir ces scores en préférences entre paires de transaction  $(U, T)_i$  indiquant que la première transaction est préférée à la seconde. Après cette étape, nous avons converti les transactions de préférence en séquences de formules de préférence en conformité avec l'approche que nous avons adopté et suivant les méthodes de conversions que nous avons décrits avec les algorithmms précédents. Ensuite nous avons extrait les règles de formules de préférence intéressantes minimales.

Nous avons bénéficié des ressources qui ont été utilisées dans les travaux antérieurs. Ainsi dans le cadre des travaux sur Sprex, un outil nommé Sprex avait été développé afin de pouvoir Extraire les profils utilisateurs (par l'approche basée sur les séquences de préférence) et évaluer la qualité des préférences prédites à partir de ces profils. Sprex est un programme qui a été développé en C++ et utilise des bibliothèques d'extraction de motifs séquentiel.

Le programme Sprex a été développé autant pour la construction de profils utilisateurs que pour à la prédiction des préférences.

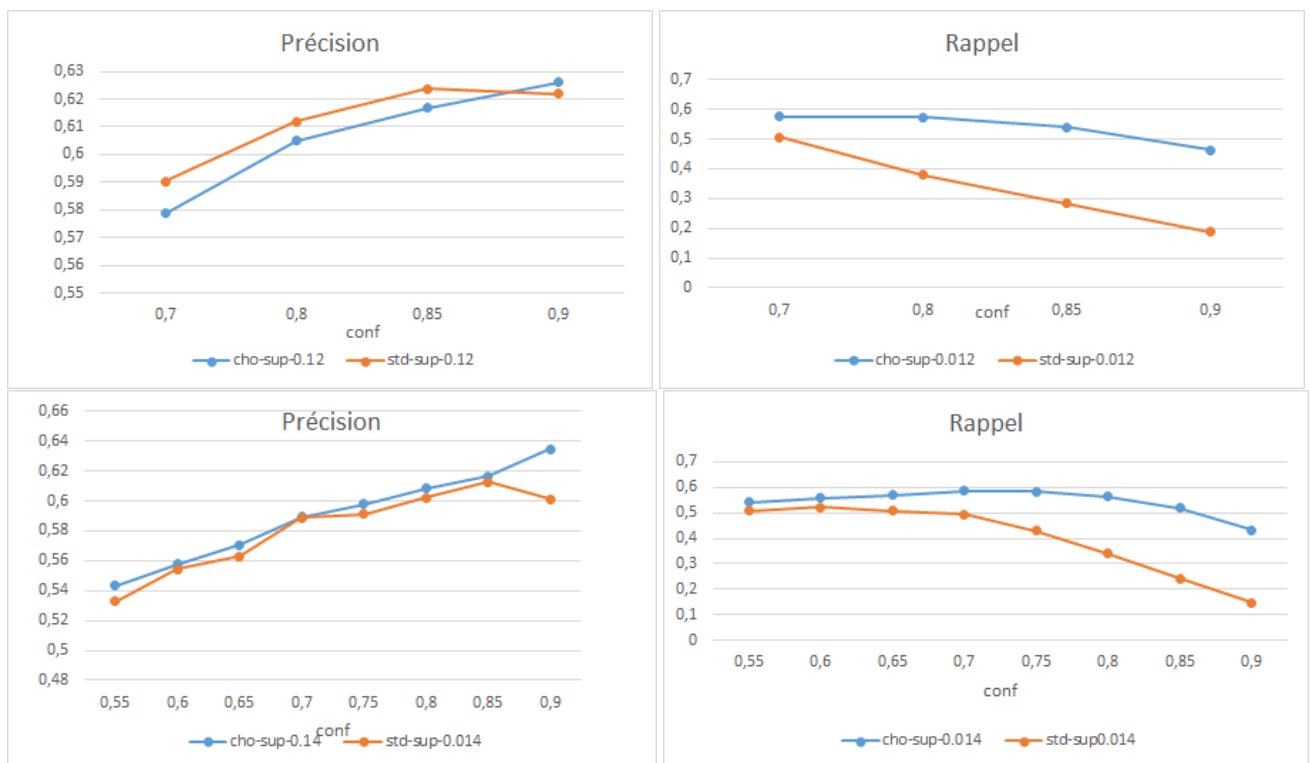
Malheureusement nous avons été confrontés à un problème de saturation mémoire lors de

l'extraction des règles de préférence à partir des séquences de formules de préférence. En effet ce problème s'est avéré être dû au fait que les séquences de formules de préférence peuvent contenir un grand nombre d'éléments(formules de préférence élémentaires) alors que lors des travaux antérieurs ([1]), les séquences de préférence quand à elles avaient un nombre d'éléments (items) beaucoup plus réduit et donc ils n'ont subit cette difficulté. Pour cela nous étions obligés d'utiliser un autre programme afin de gérer la partie consistant à extraire les règles de préférence intéressantes minimales. Pour résumer :

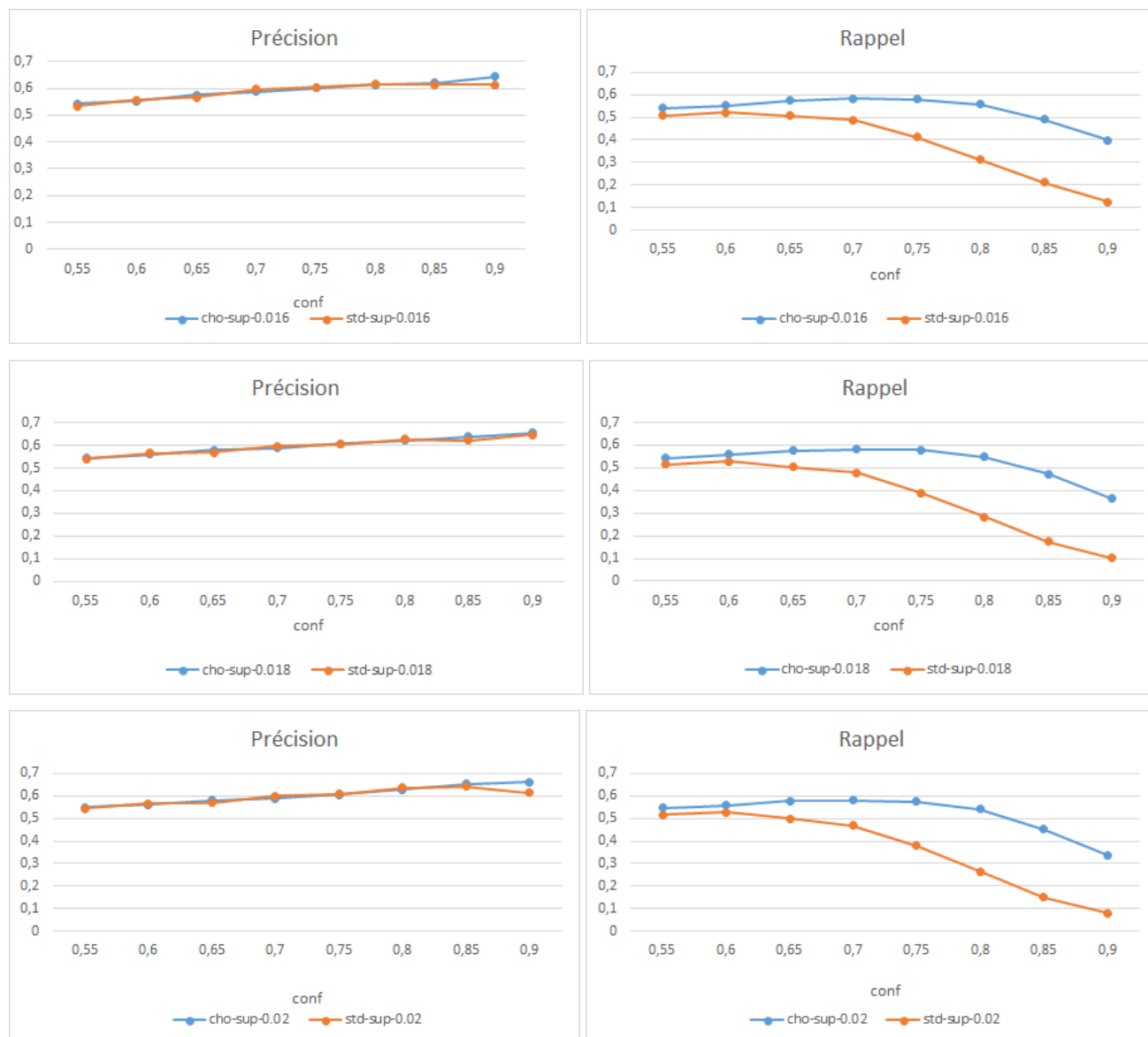
—  
—

Initialement nous cherchions à utiliser Sprex autan pour l'extraction de profil que l'évaluation de la qualité des préférences. Nous nous sommes confronté à un problème de saturation mémoire au niveau de Sprex dû au fait

## 6.6 Evaluations expérimentales

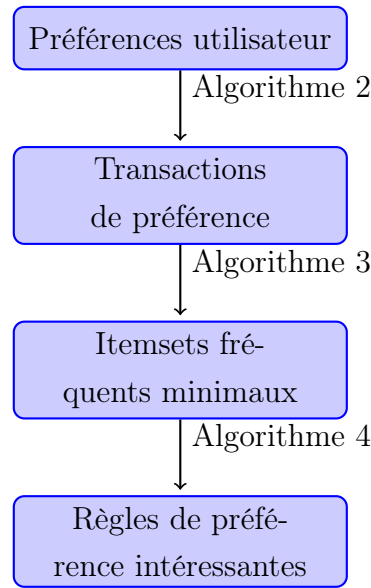






### 6.6.1 Interprétation

nous pouvons passer à la phase d'implémentation qui nous permettra alors d'effectuer une comparaison de cette approche avec l'approche de Sprex.



Le diagramme ci dessus énonce les différentes étapes du processus permettant l'extraction des règles de préférence intéressantes. Ces étapes seront décrites par la suite. Nous allons tout d'abord définir le support d'un itemset et la confiance d'une règle.

### 6.6.2 Etapes de transformation des préférences utilisateurs en transaction de préférence

Cette étape consiste à transformer le couple  $\langle t_1, t_2 \rangle$  de préférences utilisateur en une transaction de formules que l'on appellera **transaction de préférence**.

**Principe** Il se fonde sur le formalisme de Chomicky appliqué aux règles de préférence contextuelles comme précédemment énoncé. Ainsi la prise en compte de la symétrie et de l'asymétrie dans cette situation, nous amène à définir ci-dessous les items élémentaires qui seront utilisés dans les transactions de préférence :

### 6.6.3 Etape de l'extraction des règles interressantes minimales des transactions de préférence

Cette étape peut utiliser toute méthode d'extraction d'itemsets intérressaants minimaux. Notre choix s'est porté sur une méthode d'extraction des itemsets fréquents minimaux décrite dans l'article [?], du faite du peu de mémoire dont elle nécessite pour l'extraction.

## Principe

Elle consiste en un parcours en profondeur basé sur le principe du calcul des objets critiques. On a que

$$X \in \mathcal{F} \text{ est minimal ssi } \forall e \in X, \widehat{cov}(X, e) \neq \emptyset$$

avec  $\widehat{cov}(X, e) = cov(X \setminus e) \setminus cov(e)$  et  $cov$  étant la couverture d'un itemset (nombre de transactions contenant l'itemset). Donc à chaque ajout d'un item à un itemset  $X$  lors d'un parcours en profondeur, l'évaluation de la minimalité du nouvel itemset  $Y$  obtenu se fait en utilisant l'égalité suivante :

$$\widehat{cov}(Xe', e) = \widehat{cov}(X, e) \cap cov(e')$$

Ainsi il suffit d'obtenir la couverture du nouvel item ajouté pour savoir si le nouvel itemset formé est minimal.

A partir de cette procédure, étant donné que la minimalité que nous utilisons est une minimalité basée sur le support et sur la confiance, nous allons procéder comme suit :

- Générer la base de transactions de préférences inverses et ajouter celle ci à la base de préférence initiale.
- Extraire à l'aide du procédé précédent, les itemsets minimaux suivant leur support
- Filtrer les itemsets obtenus suivant que leur support dans la base initiale soit supérieur au support seuil.

Cette procédure s'explique par le fait que si un itemset  $X$  est minimal dans la base globale, c'est que

$$\forall e \in X, \text{supp}(X \setminus e) \neq \text{supp}(X) \text{ ou } \text{conf}(X \setminus e) \neq \text{conf}(X)$$

## 6.7 Algorithme

---

**Algorithme 6** : Algorithme d'extraction de règles intéressantes

---

**Entrées** : Préférences utilisateur  $\mathcal{P}$

**Sorties** : Règles de préférence intéressantes  $\mathcal{R}$

**début**

```
 $\mathcal{P}_S = \emptyset$  ; // Ensemble des transactions obtenues des tuples de
préférence
pour chaque  $\langle T, U \rangle \in \mathcal{P}$  faire
     $\mathcal{P}_{TU} = \emptyset$ ; // Ensemble des items
    pour chaque  $A_i \in R$  faire
         $x = T.A_i$  ; // Val correspondante à l'attribut  $A_i$  de  $T$ 
         $y = U.A_i$ ;
         $\mathcal{P}_{TU} = \mathcal{P}_{TU} \cup \text{Comparaison}(x, y, A_i)$ ;
    Ajouter  $\mathcal{P}_{TU}$  à  $\mathcal{P}_S$ ;
 $\mathcal{F} = \text{ExtraireItemsetFréquents}(\mathcal{P}_S)$ ;
 $\mathcal{R} = \text{FiltrerRèglesIntéressantes}(\mathcal{F})$ ;
Retourner  $\mathcal{R}$ ;
```

---

## 6.8 Exemple d'extraction des règles de préférence

Nous allons étudier deux situations :

- Cas où le schémas relationnel  $\mathcal{R}(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n)$  n'a que des attributs symboliques ;
- Cas où le schémas relationnel  $\mathcal{R}(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n)$  a des attributs symboliques et numériques.

### 6.8.1 Cas où le schémas relationnel n'a que des attributs symboliques

Prenons un schémas relationnel  $\mathcal{R}(\mathcal{A}_1, \mathcal{A}_2)$  avec  $\mathcal{A}_1$  et  $\mathcal{A}_2$  de type symboliques. Soit la base de préférence ci dessous :

	$t_1$		$t_2$	
$P_1$	a	A	b	B
$P_2$	b	A	a	B
$P_3$	a	A	a	B
$P_4$	a	A	a	C
$P_5$	b	B	a	A

### Passage des préférences vers les transactions de préférence

On a :

$$P_1 \Rightarrow T_1 = \{C : \{x_1 \neq y_1\}, P : \{x_1 = a\}, N : \{y_1 = b\}, C : \{x_2 \neq y_2\}, P : \{x_2 = A\}, N : \{y_2 = B\}, \}$$

$$P_2 \Rightarrow T_2 = \{C : \{x_1 \neq y_1\}, P : \{x_1 = b\}, N : \{y_1 = a\}, C : \{x_2 \neq y_2\}, P : \{x_2 = A\}, N : \{y_2 = B\}, \}$$

$$P_3 \Rightarrow T_3 = \{C : \{x_1 = y_1\}, C : \{x_1, y_1 = a\}, C : \{x_2 \neq y_2\}, P : \{x_2 = A\}, N : \{y_2 = B\}, \}$$

$$P_4 \Rightarrow T_4 = \{C : \{x_1 = y_1\}, C : \{x_1, y_1 = a\}, C : \{x_2 \neq y_2\}, P : \{x_2 = A\}, N : \{y_2 = C\}, \}$$

$$P_5 \Rightarrow T_5 = \{C : \{x_1 \neq y_1\}, P : \{x_1 = b\}, N : \{y_1 = a\}, C : \{x_2 \neq y_2\}, P : \{x_2 = B\}, N : \{y_2 = A\}, \}$$

## 6.9 Détermination des itemsets interressants minimaux avec sup-min=2 et confmin=0.5

Itemsets interressants minimaux de taille 1 ++++++

$$I_4 = \{P : \{x_2 = A\}, \} \text{ sup}=4.0 \text{ conf}=0.8$$

$$I_5 = \{N : \{y_2 = B\}, \} \text{ sup}=3.0 \text{ conf}=0.75$$

$$I_6 = \{P : \{x_1 = b\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667$$

$$I_7 = \{N : \{y_1 = a\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667$$

Itemsets interressants minimaux de taille 2 ++++++

$$\begin{aligned}
I_{14} &= \{C : \{x_1 \neq y_1\}, P : \{x_2 = A\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667 \\
I_{15} &= \{C : \{x_1 \neq y_1\}, N : \{y_2 = B\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667 \\
I_{29} &= \{P : \{x_2 = A\}, C : \{x_1 = y_1\}, \} \text{ sup}=2.0 \text{ conf}=1.0 \\
I_{30} &= \{P : \{x_2 = A\}, C : \{x_1, y_1 = a\}, \} \text{ sup}=2.0 \text{ conf}=1.0
\end{aligned}$$

Pas d'itemsets interessant minimaux de taille 3

### 6.9.1 Cas où le schémas relationnel a des attributs numériques et symboliques

Prenons un schémas relationnel  $\mathcal{R}(\mathcal{A}_1, \mathcal{A}_2)$  avec  $A_1$  et  $A_2$  de type symboliques. Soit la base de préférence ci dessous :

	$t_1$		$t_2$	
$P_1$	a	3	b	7
$P_2$	a	3	b	3
$P_3$	b	3	a	7
$P_4$	a	3	a	7
$P_5$	a	3	a	9
$P_6$	b	9	a	7
$P_7$	b	7	a	3

Transactions de preference obtenues

$$P_1 \Rightarrow T_1 = \{C : \{x_1 \neq y_1\}, P : \{x_1 = a\}, N : \{y_1 = b\}, C : \{x_2 \neq y_2\}, P : \{x_2 = 3\}, N : \{y_2 = 7\}, P : \{x_2 < y_2\}, N : \{y_2 > 3\}, P : \{x_2 < 7\}, C : \{x_2, y_2 < 9\}, \}$$

$$P_2 \Rightarrow T_2 = \{C : \{x_1 \neq y_1\}, P : \{x_1 = a\}, N : \{y_1 = b\}, C : \{x_2 = y_2\}, C : \{x_2, y_2 = 3\}, C : \{x_2, y_2 < 7\}, C : \{x_2, y_2 < 9\}, \}$$

$$P_3 \Rightarrow T_3 = \{C : \{x_1 \neq y_1\}, P : \{x_1 = b\}, N : \{y_1 = a\}, C : \{x_2 \neq y_2\}, P : \{x_2 = 3\}, N : \{y_2 = 7\}, P : \{x_2 < y_2\}, N : \{y_2 > 3\}, P : \{x_2 < 7\}, C : \{x_2, y_2 < 9\}, \}$$

$$P_4 \Rightarrow T_4 = \{C : \{x_1 = y_1\}, C : \{x_1, y_1 = a\}, C : \{x_2 \neq y_2\}, P : \{x_2 = 3\}, N : \{y_2 = 7\}, P : \{x_2 < y_2\}, N : \{y_2 > 3\}, P : \{x_2 < 7\}, C : \{x_2, y_2 < 9\}, \}$$

$$P_5 \Rightarrow T_5 = \{C : \{x_1 = y_1\}, C : \{x_1, y_1 = a\}, C : \{x_2 \neq y_2\}, P : \{x_2 = 3\}, N : \{y_2 = 9\}, P : \{x_2 < y_2\}, N : \{y_2 > 3\}, N : \{y_2 > 7\}, P : \{x_2 < 7\}, P : \{x_2 < 9\}, \}$$

$$P_6 \Rightarrow T_6 = \{C : \{x_1 \neq y_1\}, P : \{x_1 = b\}, N : \{y_1 = a\}, C : \{x_2 \neq y_2\}, P : \{x_2 = 9\}, N : \{y_2 = 7\}, P : \{x_2 > y_2\}, P : \{x_2 > 7\}, N : \{y_2 < 9\}, C : \{x_2, y_2 > 3\}, \}$$

$$P_7 \Rightarrow T_7 = \{C : \{x_1 \neq y_1\}, P : \{x_1 = b\}, N : \{y_1 = a\}, C : \{x_2 \neq y_2\}, P : \{x_2 = 7\}, N : \{y_2 = 3\}, P : \{x_2 > y_2\}, P : \{x_2 > 3\}, N : \{y_2 < 7\}, C : \{x_2, y_2 < 9\}, \}$$

## 6.10 Determination des itemsets interressants minimaux avec sup-min=2 et confmin=0.5

Itemsets interressants minimaux de taille 1 ++++++

$$\begin{aligned} I_4 &= \{P : \{x_2 = 3\}, \} \text{ sup}=4.0 \text{ conf}=0.8 \\ I_5 &= \{N : \{y_2 = 7\}, \} \text{ sup}=4.0 \text{ conf}=0.8 \\ I_6 &= \{P : \{x_2 < y_2\}, \} \text{ sup}=4.0 \text{ conf}=0.666666666667 \\ I_7 &= \{N : \{y_2 > 3\}, \} \text{ sup}=4.0 \text{ conf}=0.8 \\ I_8 &= \{P : \{x_2 < 7\}, \} \text{ sup}=4.0 \text{ conf}=0.8 \\ I_{13} &= \{P : \{x_1 = b\}, \} \text{ sup}=3.0 \text{ conf}=0.6 \\ I_{14} &= \{N : \{y_1 = a\}, \} \text{ sup}=3.0 \text{ conf}=0.6 \end{aligned}$$

Itemsets interressants minimaux de taille 2 ++++++

$$\begin{aligned} I_{32} &= \{C : \{x_1 \neq y_1\}, P : \{x_2 = 3\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667 \\ I_{33} &= \{C : \{x_1 \neq y_1\}, N : \{y_2 = 7\}, \} \text{ sup}=3.0 \text{ conf}=0.75 \\ I_{35} &= \{C : \{x_1 \neq y_1\}, N : \{y_2 > 3\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667 \\ I_{36} &= \{C : \{x_1 \neq y_1\}, P : \{x_2 < 7\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667 \\ I_{74} &= \{C : \{x_2 \neq y_2\}, P : \{x_1 = b\}, \} \text{ sup}=3.0 \text{ conf}=0.75 \\ I_{75} &= \{C : \{x_2 \neq y_2\}, N : \{y_1 = a\}, \} \text{ sup}=3.0 \text{ conf}=0.75 \\ I_{79} &= \{P : \{x_2 = 3\}, N : \{y_2 = 7\}, \} \text{ sup}=3.0 \text{ conf}=0.75 \\ I_{83} &= \{P : \{x_2 = 3\}, C : \{x_2, y_2 < 9\}, \} \text{ sup}=3.0 \text{ conf}=0.75 \\ I_{86} &= \{P : \{x_2 = 3\}, C : \{x_1 = y_1\}, \} \text{ sup}=2.0 \text{ conf}=1.0 \\ I_{87} &= \{P : \{x_2 = 3\}, C : \{x_1, y_1 = a\}, \} \text{ sup}=2.0 \text{ conf}=1.0 \\ I_{89} &= \{N : \{y_2 = 7\}, P : \{x_2 < y_2\}, \} \text{ sup}=3.0 \text{ conf}=0.75 \\ I_{90} &= \{N : \{y_2 = 7\}, N : \{y_2 > 3\}, \} \text{ sup}=3.0 \text{ conf}=0.75 \\ I_{91} &= \{N : \{y_2 = 7\}, P : \{x_2 < 7\}, \} \text{ sup}=3.0 \text{ conf}=0.75 \\ I_{92} &= \{N : \{y_2 = 7\}, C : \{x_2, y_2 < 9\}, \} \text{ sup}=3.0 \text{ conf}=0.75 \\ I_{93} &= \{N : \{y_2 = 7\}, P : \{x_1 = b\}, \} \text{ sup}=2.0 \text{ conf}=1.0 \\ I_{94} &= \{N : \{y_2 = 7\}, N : \{y_1 = a\}, \} \text{ sup}=2.0 \text{ conf}=1.0 \\ I_{100} &= \{P : \{x_2 < y_2\}, C : \{x_2, y_2 < 9\}, \} \text{ sup}=3.0 \text{ conf}=0.75 \\ I_{103} &= \{P : \{x_2 < y_2\}, C : \{x_1 = y_1\}, \} \text{ sup}=2.0 \text{ conf}=1.0 \\ I_{104} &= \{P : \{x_2 < y_2\}, C : \{x_1, y_1 = a\}, \} \text{ sup}=2.0 \text{ conf}=1.0 \\ I_{107} &= \{N : \{y_2 > 3\}, C : \{x_2, y_2 < 9\}, \} \text{ sup}=3.0 \text{ conf}=0.75 \end{aligned}$$



$I_{110} = \{N : \{y_2 > 3\}, C : \{x_1 = y_1\}, \} \text{ sup}=2.0 \text{ conf}=1.0$   
 $I_{111} = \{N : \{y_2 > 3\}, C : \{x_1, y_1 = a\}, \} \text{ sup}=2.0 \text{ conf}=1.0$   
 $I_{113} = \{P : \{x_2 < 7\}, C : \{x_2, y_2 < 9\}, \} \text{ sup}=3.0 \text{ conf}=0.75$   
 $I_{116} = \{P : \{x_2 < 7\}, C : \{x_1 = y_1\}, \} \text{ sup}=2.0 \text{ conf}=1.0$   
 $I_{117} = \{P : \{x_2 < 7\}, C : \{x_1, y_1 = a\}, \} \text{ sup}=2.0 \text{ conf}=1.0$   
 $I_{127} = \{P : \{x_1 = b\}, P : \{x_2 > y_2\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667$   
 $I_{130} = \{N : \{y_1 = a\}, P : \{x_2 > y_2\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667$

Itemsets interressants minimaux de taille 3 ++++++

$I_{151} = \{C : \{x_1 \neq y_1\}, N : \{y_2 = 7\}, P : \{x_2 < y_2\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667$   
 $I_{154} = \{C : \{x_2, y_2 < 9\}, C : \{x_1 \neq y_1\}, N : \{y_2 = 7\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667$   
 $I_{160} = \{C : \{x_2, y_2 < 9\}, C : \{x_1 \neq y_1\}, P : \{x_2 < y_2\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667$   
 $I_{197} = \{C : \{x_2, y_2 < 9\}, C : \{x_2 \neq y_2\}, P : \{x_1 = b\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667$   
 $I_{198} = \{C : \{x_2, y_2 < 9\}, C : \{x_2 \neq y_2\}, N : \{y_1 = a\}, \} \text{ sup}=2.0 \text{ conf}=0.666666666667$

Pas d'itemsets interressant minimaux de taille 4