UNIVERSITE GASTON BERGER

Master2 Informatique

Memoire de fin de cycle

Présenté par:

PEKPASSI Digonaou

29 Juillet 2015

Table des matières

1 Etat de l'art

1.1 Notions de base

1.1.1 Pré requis

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 peut être :

- *réflexive si $\forall x \in E,\ x\mathcal{R}x$*
- *irréflexive 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 pout out $x, y \in E$ 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 de préférence*** *Une relation de préférence sur un ensemble d'éléments E est une relation binaire.*
Dans ce cas, étant donné une relation de préférences notée $\succ_{\mathcal{R}}$:

1. *$x \succ_{mathcalR} y$ signifie que x est strictement préféré à y,*
2. *$x\tilde{\mathcal{R}}y$ signifie que x et y sont également préférés, et enfin,*
3. *$x||_{\mathcal{R}}y$ signifie qu'il n'y a ni indifférence, ni préférence entre x et y et on dit que x et y ne sont pas comparables.*

Definition 5. ***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 6. ***Préordre***
Un préordre sur un ensemble E est une relation binaire réflexive et transitive.

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

Definition 7. *(Relation d'ordre strict) Une relation d'ordre strict sur un ensemble E est une relation binaire irréflexive et transitive. L'ordre strict est dit faible, s'il est partiel et négativement transitif.*

1.1.2 Les différents types de représentation de préférences qualitatives

Dans ce qui suit, nous considérons l'ensemble E comme étant l'ensemble des tuples $t = (u_1, u_2, ..., u_d)$ avec $u_i \in dom(A_i)$ de $R(A_1, A_2, ..., A_d)$. Nous définissons suivant cela une relation de préférence ($\succ_p$) sur un schéma R par un ensemble de tuples (t_i, t_j) de r. Par conséquent $t_i \succ_p t_j$ signifiera que t_i est préféré à t_j sous $\succ_p$.

Formulation de Jan Chomicki

CHOMICKI a dans l'article [2] établit la notion de **relation de préférence** $\succ$ comme un sous ensemble de $Dom(A) \times Dom(A)$. Ceci veut intuitivement dire que $\succ$ est une relation binaire entre des transactions d'une même instance r du schémas relationnel $\mathcal{R}$. Ainsi on a t_1 est préféré à t_2 suivant $\succ$ si $t_1 \succ t_2$. De même il a définit la notion de formule de préférence $C(t_1, t_1)$ comme étant une une formule de premier ordre définissant une relation de préférence $\succ_C$ de telle sorte que $t_1 \succ_C t_2 \equiv C(t_1, t_2)$. Cette formule est définie ci dessous.

Definition 8. *Une formule de préférence suivant le formalisme de CHOMICKY est une formule représentée sous la forme normale disjonctive DNF sans les quantificateurs. Ainsi on a :*

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

où f_{ij} sont des formules atomiques sous la forme $x\delta y$ ou $x\delta c$ avec x et y des variables d'un attribut de R correspondants respectivement à t_1 et t_2. c quand à lui est une constante. δ prend les valeurs $=$,$\neq$ dans le cas où les éléments x, y, c sont de type quelconque (Exemple : $x = y$,$x = a$,$y \neq a$), et $\leqslant, \geqslant, <, >$ dans le cas où on a à faire à des valeurs numériques

(Exemple : $x < y$,$x < a$,$y \geqslant a$).

Example 1

- 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(Plat,TypePlat, TypeVin). On a qu'ils vérifient la formule de préférence C si et seulement si

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

  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(TypeFilm,Annee, 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 veux 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 meême anné, il préfère ceux avec l'acteur Sylvester Stalone à ceux avec l'acteur Pierce Brosnan.

Formulation de Nic Wilson

Le formalisme de Wilson se fonde sur la logique des préférences conditionnelles. Une règle de préférence conditionnelle a sa forme générale comme suit : "si [conjonction de conditions élémentaires] alors [décision]"

La partie condition indique dans quelle condition la règle est appliquée et la partie décision montre le choix à opérer sur les valeurs si la partie condition est satisfaite. Plus précisément, le formalisme proposé par Wilson est défini de la façon suivante :

Definition 9. *Soit $\mathcal{A}$ un ensemble d'attributs, $U \subseteq \mathcal{A}$, a_{i1} et $a_{i2} \in dom(A_i)$, $A_i \notin U$. Une règle de préférence contextuelle est sous la forme $u : a_{i1} \succ a_{i2}[W]$ tel que $u \in dom(U)$ et*

W un sous-ensemble de $S = \mathcal{A} - (U \cup A_i)$.

Cette formulation peut être réécrite suivant le formalisme de CHOMICKY (Définition 8) de la façon suivante (dans cette situation on a $p = u : a_{i1} \succ a_{i2}[W]$) :
$t_i \succ_p t_j$ ssi $t_i[S - W] = t_j[S - W] \wedge t_i[U] = t_j[U] = u \wedge t_i[A_i] = a_i1 \wedge t_j[A_i] = ai$

De façon intuitive cette règle de préférence contextuelle indique que tout tuple t_i contenant u et a_{i1} est préféré à tout tuple t_j contenant u et a_{i2} indépendamment des valeurs des attributs W dans $dom(W)$, sachant que les valeurs dans $dom(S - W)$ sont les mêmes pour les deux tuples.

Par cette formulation, l'exemple 1 peut se réécrire de la façon suivante :

Example 2

Dans le cas d'un schéma relationnel $\mathcal{R} = \{Langage, Title, Director, Actor\}$ on a l'exemple suivant
$Langage = English \succ Langage = French[Title, Director, Actor]$

Cette règle indique la préférence des films Anglais aux films Français sans tenir compte du titre du film, du directeur du film ou bien de l'acteur, pour tout couple de film ayant la même année de production.

Formulation de Werner Kießling

Dans les travaux de [Kie02], l'auteur propose un langage formel pour modéliser les préférences qui forment une relation d'ordre partiel strict. Pour cela, l'auteur propose un certain nombre de constructeurs de base qui sont des patrons de préférence, qui dès qu'ils sont instanciés donnent des préférences de base. Ces constructeurs sont caractérisés par plusieurs arguments dont le premier indique les noms des attributs concernés par le constructeur et les autres arguments indiquant les caractéristiques de l'ordre partiel stricte $< P$ qui sera appliqué sur les attributs énoncés dans le premier argument. Dans ce travail, nous présenterons deux types de constructeurs de base à savoir les **constructeurs de base non numériques** et les **constructeurs de base numériques**.

Les constructeurs de base non numériques

Leurs définition ne prend pas en compte d'opérations numériques.
Exemple :

Préférence POS : POS(A, POS-set)
P est une préférence POS si :

$$x < Pyssix \notin POS - set \wedge y \in POS - set$$

Ceci veut dire que parmis deux valeurs de A, la valeur préférée est celle qui appartient à $POS-set \subset dom(A)$. Autrement si aucune des valeurs n'est dans $POS-set$, alors une des deux est accepable. (Ex :scénario d'un choix de voiture : POS(transmission, automatique))

Préférence POS/NEG : POS/NEG(A,POS-set ;NEG-set)
P est appelé préférence POS/NEG si :

$$x < Pyiff(x \in NEG-set \wedge y \notin NEG-set) \vee (x \notin NEG-set \wedge x \notin POS-set \wedge y \in POS-set)$$

Ceci veut dire que parmi deux valeurs, la préférée est celle qui appartient à $POS-set \subset A$, autrement si aucune des deux valeurs n'est dans $POS-set$, et qu'une d'elle n'appartient pas à $NEG-set \subset A$, elle est la préférée, autrement si les deux appartiennent à $NEG-set$ alors une d'elles est choisie arbitrairement. (Ex : scénario de choix de voiture : POS/NEG(couleur,jaune ;gris)

Les constructeurs de base non numériques
Leur définition prend en compte des opérations numériques.
Exemple :

Préférence AROUND : AROUND(A, z)
Etant donné $z \in dom(A), pourtoutv \in dom(A)$, on défini $distance(v, z) := abs(v - z)$. P est appelé préférence AROUND, si :

$$x < Pyssidistance(x, z) > distance(y, z)$$

Ceci veut dire que parmi deux valeurs de A, si l'une d'elle est z, elle est la valeur préférée. Autrement, si aucune des valeurs n'est z, la plus préférée est celle qui iest la plus proche de Z e terme de distance. (Ex : scénario de choix de voiture : AROUND(prix, 40000))

Préférences LOWEST, HIGHEST : LOWEST(A),HIGHEST(A)

P est appelé préférence LOWEST, si :

$$x < Pyssix > y$$

P est appelé préférence HIGHEST, si :

$$x < Pyssix < y$$

Ceci veut dire qu'une valeur désirée doit être la plus élevé (basse) possible. (Ex : scénario choix de voiture : HIGHEST(puissance).

1.2 Les différents modes de combinaison de préférences

Après avoir présenté précédemment les différentes formulations des préférences dans la littérature, nous allons à présent étudier les différentes méthodes de combinaisons de ces préférences afin de former des modèles de préférence.

1.2.1 Composition de préférences prioritaires

Elle est définie comme suit :

Definition 10. *Soient P_x et P_y deux relations de préférence définies sur le même schéma relationnel R. La relation de préférence prioritaire $\succ_{P_x} \& \succ_{P_y}$ est définie sur R par :*

$$\forall t_i, t_j \text{ d'une instance de } R, t_i \succ_{P_x} \& \succ_{P_y} t_j \text{ ssi } (t_i \succ_{P_x} t_j) \vee (t_i \sim_{P_x} t_j \wedge t_i \succ_{P_y} t_j).$$

Ceci veut dire qu'on utilise la préférence P_x si elle est applicable et autrement on utilise P_y.

Example 3

Soient deux relations de préférence $\succ_{P_x}$ et $\succ_{P_y}$ tel que :

- $t_i \succ_{P_x} t_j$, ssi $t_i[genre] =' Drama' \wedge t_j[genre] =' Comedy'$.
- $t_i \succ_{P_y} t_j$, ssi $t_i[Language] =' French' \wedge t_j[Language =' English']$.

La relation de préférence prioritaire $\succ_{P_x} \& \succ_{P_y}$ est alors définie comme suit :

$$t_i \succ_{P_x} \& \succ_{P_y} t_j \text{ ssi } (t_i[genre] =' Drama' \wedge t_j[genre] =' Comedy') \vee (t_i[genre] \neq' Drama' \wedge t_i[Language] =$$

1.2.2 Réseaux de préférences conditionnelles (CP-nets)

Un CP-nets est une approche de représentation des préférences fondée sur un représentation graphique. Ceci se fait à l'aide de deux notions à savoir :
- la notion de préférence conditionnelle ;
- la notion de ceteris paribus.

Soit $V = \{X_1, X_2, ..., X_i\}$ 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 $Pa(X_i)$.

Example 4

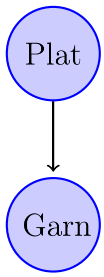
Cas des préférences au niveau des plats de riz. L'utilisateur préfère accompagné le riz au poisson plutôt qu'à la viande, et dans le cas du spagetti, il préfère la viande au poisson.

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.



	Poisson≻ Viande
Riz	Poisson≻ Viande
Spaghetti	Viande≻ Poisson

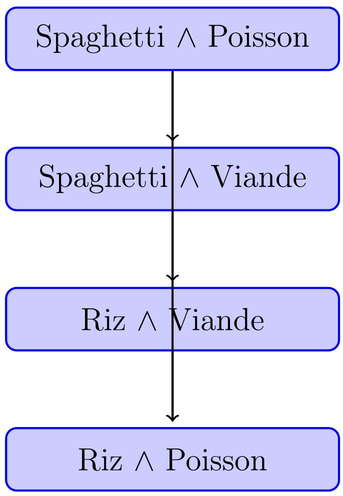
Ainsi nous définissons un CP-nets de la manière suivante :

Definition 11. *Un CP-net sur les attributs* $V = \{X_1, X_1, ..., X_n\}$ *est un graphe orienté sur les attributs* $X_1, X_1, ..., X_n$ *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 6

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 ∧ Poisson) est l'élément le moins préféré tandis que l'élément le plus en bas (Riz ∧ Poisson) est l'élément le plus préféré.

Comparaison de deux transaction suivant un CP-net

Definition 12. *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* $Y = V_{(}U \cup \{X_i\})$*. 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 $u \in Asst(U)$. 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 13. *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.*

---

2 Résumé article Preference Learning : An Introduction de Johannes Furnkranz 1 and Eyke Hullermeier

Introduction

Le raisonnement à base de préférence est un domaine de recherche en Intelligence Artificielle. Les préférences sont des contraintes faibles qui permetttent une meilleure flexibilité en terme d'étude et deprédiction des choix utilisateurs. Ainsi on peut dans le cas où les conditions de recherche fixé par un utilisateur ne donnent pas des résultats consistants, se baser sur ces préférences afin de proposer d'autres résultats suivant ses préférences qui porteront aussi autant d'intérêt.

Il est à noter que l'quisition de préférences se base sur plusieurs principes à savoir :

- Les languages de modélisation et de formalisme
- Les méthodes d'apprentissage automatique, de découverte

Ainsi l'étude des préférences est un domaine qui fait l'objet de recherche dans les disciplines tel que l'apprentissage artificiel, la fouille de données, les systèmes de recommandation.

L'apprentissage de préférence, grosso modo parlant consiste à extraires des modèles de préférences de données empiriques.

Deux approches sont à distinguer en terme de modélisation de préférences à savoir

- Les fonctions utilités

– Les relations de préférence

Comparativement aux méthodes basées sur les fonctions utilité, l'apprentissage des relations de préférence se distingue fortement des méthodes classiques comme la classification ou la régression du fait qu'elle permet d'obtenir des structures complexes comme les classements et les relations d'ordre partielles plutôt que des valeurs.

Ce document parcours de façon générale les recherches en cours dans le domaine de l'apprentissage de préférences et alors permettra d'avoir une meilleure vue de ce domaine.

2.1 Tâches d'apprentiissage de prférences

La tâche d'apprentissage de classement est celle qui a le plus d'attention dans l'apprentissage de préférence.Dans cette 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 supervisée. Ainsi un objet caractérisant une donnée est appellé 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 :
$x = (x_1, x_2, ..., x_m) \in \mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times ... \times \mathcal{X}_m$

Nous distinguons trois types de classement à savoir :

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

2.1.1 Le classement de labels

Le classement de labels est un type de classement prenant en compte un espace d'instances $\mathcal{X}$ et un ensemble fini de labels $\mathcal{Y} == \{y_1, y_2, ..., y_k\}$. Il consiste à apprendre un "classeur de labels" qui est une fonction définie sur $\mathcal{X} \Rightarrow S_y$ qui prend en entrée une instance et fournie en sortie une permutation de l'ensemble des labels suivant un ordre total. Ainsi le classement de label peut être vu comme une généralisation de la classification conventionnelle où un classement total

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

est associé à une instance x au lieu d'un seul label de classe das le cas de la classification. π_x est une permutation de $\{1, 2, ..., k\}$ telle que $\pi_x(i)$ est la position du label $y_(i)$ dans le classement associé à x.

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

Comme exemple de situations où on a affaire à un classement de label, on peut parler du classement de la pertinence des types de rubriques par journal comme le sport, les technologies, la santé etc..

Soit :

- Un ensemble d'instances d'apprentissage $\{x_l | l = 1, 2, ..., n\} \subseteq \mathcal{X}$ (chaque instance peut ou pas être représentée par un vecteur).
- Un ensemble de labels $\mathcal{Y} = \{y_i | i = 1, 2, ..., k\}$
- Pour toute instance d'entraînement x_l, un enesmble de préférences de la forme $y_i \succ_{x_l} y_j$

Trouvons :

Une fonction de classement qui lie toute instance x à un classement $\succ_x$ de $\mathcal{Y}$ (i.e une permutation $\pi_x \in \mathcal{S}_k$)

Mesures de performance

- Erreur de classement (ex : basée sur les mesures de corrélation entre les rangs) comparant les classements prédits aux classements cibles.
- Erreur de position comparant le rang prédit au label cible.

2.1.2 Classement d'instance

Ce classement se fonde sur le principe d'une 'classification ordinale où une instance x appartient à une des classes de l'ensemble de classes ordonnées $\mathcal{Y} = \{y_1, y_2, ..., y_k\}$ tel que $y_1 < h_2 < ... < y_k$.

Le jeux de données d'apprentissage consistent en un ensemble $\mathcal{T}$ d'instances labellisées. Comme exemple on peut considérer la répartition d'articles suivant les catégories "rejeté", "faible rejet", "faible acceptation", "acceptation". Suivant les données d'apprentissage, le but est alors d'apprendre des fonctions de classement assignant un score à chaque instance fournie en entrée. Ce score permet de classer les différentes instances. Ainsi des problèmes concrêts étudiés prennent en compte des classements basés sur des ensembles de deux classes (k=2) appelés **problèmes de classement bipartites** et des classements basés sur des ensembles de k classes appelés de **problèmes de classement k-partite ou multipartite**. Dans le cas des classements bipartites, en prennant deux instances x_1 et x_2 , avec le score donné par la fonction, on a que :

$$f(x_1) > f(x_2) \Rightarrow x_1 \text{ a le label supérieur et } x_2 \text{ le label inférieur}$$

Du coté du classement multipartite de même on a en prennant deux instances x_1 et x_2 , avec le score donné par la fonction, on a que :

$$f(x_1) > f(x_2) \Rightarrow x_1 \text{ a un label supérieur à celui de } x_2$$

Mais dans ce cas ci il est à prendre en compte l'écart entre les niveau des labels lors de l'évaluation de la précision de la méthode d'apprentissage.

Soit :

- Un ensemble d'instances d'apprentissage $\{x_l | l = 1, 2, ..., n\} \subseteq \mathcal{X}$ (chaque instance peut ou pas être représentée sous forme d'un vecteur).
- Un ensemble de labels $\mathcal{Y} = \{y_i | i = 1, 2, ..., k\}$ munis d'un ordre $y_1 < y_2 < ... < y_k$
- Pour toute instance d'entraînement x_l, un label associé y_l

Trouvons :
Une fonction de classement qui permet de classé un nouvel ensemble d'instances $\{x_j\}_{j=1}^{t}$ suivant leur degré de préférence

---

2.1.3 Classement d'objets

L'approche de Cohenref : article de Cohen :Learning to Order Things

L'approche de Cohen est caracérisée par la définition des **fonctions de préférence** PREF qui sont des fonctions définies sur $X \times X \rightarrow [0, 1]$ indiquant pour tout couple $(u, v) \in X \times X$, avec quel taux de recommandation l'un doit être préféré à l'autre. Ainsi si $PREF(u, v)$ tend vers 1, alors il y'a une forte recommandation que u soit préféré à v et si $PREF(u, v)$ tend vers 0, alors il y'a une forte recommandation que v soit préféré à u. Une valeur proche de 1/2 est interprétée comme une abstenion de faire une recommandation.

Cette fonction de préférence est obtenue par la combinaison pondérée de N **fonctions de préférence primitive** $R_1, ..., R_N$. Elle est écrite alors de la manière suivante :

$$PREF(u, v) = \sum_{i=1}^{N} w_i R_i(u, v)$$

Chaque fonction de préférence primitive $R_i(u, v)$ est déjà disponible et prend les valeurs 1 si u est préféré à v, 0 si v est préféré à u et 1/2 autrement. Afin de déterminer les poids w_i alloué à chaque fonction de préférence primitive R_i, plusieurs tours sont effectués en fournissant à chaque tour t un ensemble d'instance X^t à classer. Après le classement de ces instances par la fonction de préférence de base, un feedback F^t lui est fourni en contenant des paires (u, v) indiquant quel élément u devrait être préféré à quel élément v dans le classement. Avecce feedback on peut définir la **perte** de la fonction de préférence R_i définie par :

$$Loss(R, F) = \frac{\sum_{(u,v)\in F}(1 - R(u, v))}{|F|} = 1 - \frac{1}{|F|} \sum_{(u,v)\in F} R(u, v)$$

Initialement $w_i^1 = 1/N$ mais par la suite, les valeurs de w_i^t par les itérations suivantes sont obtenus par la formule suivante :

$$w_i^(t + 1) = \frac{w_i^t.\beta^{Loss(R_i^t,F^t)}}{Z^t}$$

où $\beta \in [0, 1]$ est un paramètre et Z_t est une **constante de normalisation**.

Suite à l'obtention de la fonction de préférence finale fournissant ainsi un score à chaque couple d'un l'ensemble $X \times X$ fourni en entrée, le but est de fournir un ordre total de ces instances. Le processus de classement effectué à partir de l'évaluation des scores des couples (u, v) fourni par $PREF$ est NP-complet,. D'où la necessité d'utiliser un algorithme optimisant ce classement. Le but est d'avoir l'ordre O maximisant

$$\sum_{(u,v)\in O} PREF(u, v)$$

Ainsi il a été proposé un algorithme glouton décrit ci dessous :

Algorithme 1 : Algorithme glouton de classement (Cohen)

$O \leftarrow \emptyset$;
forall the $x \in X_u$ **do**
  $score(x) \leftarrow \sum_{x' \in X_u} PREF(x, x')$;
while $X_u \neq \emptyset$ **do**
  $x_{top} \leftarrow argmax_x score(x)$;
  $O_u \leftarrow O_u \succ x_{top}$;
  $X_u \leftarrow X_u - x_{top}$;
  forall the $x \in X_u$ **do**
    $score(x) \leftarrow score(x) - PREF(x, x_{top})$;
return O_u;

Algorithme RankBoost

C'est un algorithme proposé par Freund et al ref : livre preference learning qui est un dérivé de Adaboost, Algorithme de boosting connu dans le domaine de l'apprentissage. Sa spécificité est qu'il se base sur des ordre au lieu de se baser sur des scores dans le cas de Adaboost. Les données d'entrée de Rankboost sont des **fonctions de retour** $\phi(x_a, x_b)$ qui impliquent que $x_b \succ x_a$ si $\phi(x_a, x_b) > 0$. Ces fonctions de retour sont issu du processus d'appréciation fournie en retour de l'information de classement fournie par l'algorithme. De même en entré, il y'a un ensemble de fonctions appellées **ranking features** $f(x_i)$ qui fournie une information partielle sur l'ordre dans l'ensemble X des instances. Avec ces entrées, Rankboost retourne le classement final $H(x_i)$ qui fonctionne comme une fonction de score. Initialement, une distributino est calculée par $D_1(x_a, x_b) = max(\phi(x_a, x_b), 0)/Z_1$ pour tout $(x_a, x_b) \in X \times X$ où Z_1 est un coefficient de normalisation qui assure que $\sum_{x_a, x_b} D_1(x_a, x_b) = 1$ Ainsi pour chaque tour $t = 1, ..., T$, l'algorithme répète un processus de sélection du poids α_t et de la fonction d'apprentissage de base $h_t(x)$ de manière à minimiser la valeur de Z_t. Il existe plusieurs méthodes pour cette minimisation que nous n'allons pas expliciter (conf ...). A l'itération suivante on met à jour la distribution :

$$D_{t+1}(x_a, x_b) = \frac{1}{Z_t} \exp^(\alpha_t(h_t(X_a) - h_t(x_b))$$

Les fonctions d'apprentissage de base h_t sont caractérisées par le fait que $h_t(x_b) > h_t(x_a) \Rightarrow$

$x_b \succ x_a$. Après les T itérations, nous obtenos une fonction de score :

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

Ainsi cette foncion permet de classer toutes les instances de X tel que :

$$\forall x_a, x_b \in X \times X, H(x_a) > H(x_b) \Rightarrow x_a \succ x_b$$

Classement à base de SVM

---

3 Définition et extraction des règles de préférence contextuelle

Soit $\mathcal{R}(\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_n)$ un schémas relationnel tel que chaque attribut $\mathcal{A}_i$ soit caractérisé par son domaine de valeurs $Dom(\mathcal{A}_i)$. Ainsi notons $Dom(\mathcal{A}) = \prod_1^n Dom(\mathcal{A}_i)$. On appelle une **transaction**, un élément de $Dom(\mathcal{A})$. Une base de transactions $\mathcal{B}$ est un ensemble de transactions, chacune associée à un identifiant unique. Avec $\mathbf{I} = \bigcup_1^n Dom(\mathcal{A}_i)$, tout $i \in \mathbf{I}$ est appelé **item**. Tout sous ensemble d'une transaction T est appelé **itemset**.

Example 7

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

3.1 Préférences utilisateur

Definition 14. *(Préférence utilisateur)*
*Une **préférence utilisateur** est une paire de transactions $\langle t_1, t_2 \rangle$ qui spécifie que l'utilisateur préfère t_1 à t_2. Elle est aussi écrite sous la forme $t_1 \succ t_2$.*

Une ***base de préférence*** *est un ensemble de préférences utilisateur.*

Definition 15. *(Règle de préférence contextuelle)*
Une ***règle de préférence contextuelle*** *est une règle de la forme* $C \to X \succ Y$ *signifiant que l'utilisateur préfère* X *à* Y *si le contexte* C *est observé.*
Il est à noté que, pour généraliser, C,X,Y *peuvent être des* ***itemsets*** *(Ensemble d'items) ou des* ***itemsets de préférence*** *(conditions que doivent vérifier les items. Pour ces derniers nous en reparlerons à la suite du travail).*

Example 8

La règle de préférence contextuelle $\{Action\} \to \{1990\} \succ \{2000\}$ (ici C, X, Y sont des itemsets) 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 $\{MemeGenre\} \to \{Avant\ l'annee\ 2000\} \succ \{Apres\ l'annee\ 2000\}$ (ici C, X, Y sont des itemsets de préférence) signifie que dans le cas où deux films sont de même genre, l'utilisateur préfère les films produits au delà de l'année 2000.

On dit qu'une préférence $< T_1, T_2 >$ **supporte** une règle de préférence contextuelle $\mathcal{R} = C \to X \succ Y$ prenant en compte les items (resp les itemsets de préférence) si $C \in T_1 \cap T_2$ (resp T_1 et T_2 vérifient C) et si $X \in T_1 \backslash T_1 \cap T_2$ (resp seul T_1 vérifie X) et $Y \in T_2 \backslash T_1 \cap T_2$ (resp seul T_2 vérifie Y).
De même on dit qu'une préférence $< T_1, T_2 >$ **contredit** une règle de préférence contextuelle $\mathcal{R} = C \to X \succ Y$ prenant en compte les items (resp les itemsets de préférence) si $C \in T_1 \cap T_2$ (resp T_1 et T_2 vérifient C) et si $Y \in T_1 \backslash T_1 \cap T_2$ (resp seul T_1 vérifie Y) et $X \in T_2 \backslash T_1 \cap T_2$ (resp seul T_2 vérifie X).

Example 9

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

De même on dit que une transaction T_1 est **préférée** à une transaction T_2 suivant une règle contextuelle $\mathcal{R} = C \to X \succ Y$ 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 $Y \in T_1 \backslash T_1 \cap T_2$ (resp seul T_1 vérifie Y) et $X \in T_2 \backslash T_1 \cap T_2$ (resp seul T_2 vérifie X).

3.2 Aperçut du formalisme de CHOMICKY

CHOMICKI a dans l'article [2] établit la notion de **relation de préférence** $\succ$ comme un sous ensemble de $Dom(A) \times Dom(A)$. Ceci veut intuitivement dire que $\succ$ est une relation binaire entre des transactions d'une même instance r du schémas relationnel $\mathcal{R}$. Ainsi on a t_1 est préféré à t_2 suivant $\succ$ si $t_1 \succ t_2$. De même il a définit la notion de formule de préférence $C(t_1, t_1)$ comme étant une une formule de premier ordre définissant une relation de préférence $\succ_C$ de telle sorte que $t_1 \succ_C t_2 \equiv C(t_1, t_2)$

Cette formule est sous une forme normale disjonctive DNF mais sans considérer les quantificateurs. Ainsi on a :

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

où f_{ij} sont des formules atomiques sous la forme $x\delta y$ ou $x\delta c$ avec x et y des variables d'un attribut de R correspondants respectivement à t_1 et t_2. c quand à lui est une constante. δ prend les valeurs $=,\neq$ dans le cas où les éléments x, y, c sont de type quelconque (Exemple : $x = y$,$x = a$,$y \neq a$), et $\leqslant, \geqslant, <, >$ dans le cas où on a à faire à des valeurs numériques (Exemple : $x < y$,$x < a$,$y \geqslant a$).

Example 10

- 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(Plat,TypePlat, TypeVin). On a qu'ils vérifient la formule de préférence C si et seulement si

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

  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(TypeFilm,Annee, Acteur). On a $t_1 \succ_C t_2$ ssi :

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

Cette règle veux 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 meême anné, il préfère ceux avec l'acteur Sylvester Stalone à ceux avec l'acteur Pierce Brosnan.

3.3 Définition d'une règle de préférence suivant le formalisme de CHOMICKY

Dans notre travail, nous allons nous inspiré du formalisme de CHOMICKY décrit précédamment pour proposer un format de règle de préférence. Plus précisément les formules de préférences de CHOMICKY seront plus ou moins similaires à nos règles de préférence que nous définirons par la suite. On sait qu'une formule de préférence de Chomicky est sous la forme :

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

Dans notre cas, le **profil** utilisateur est un ensemble de règles de préférence qui permettra de prédire la préférence de l'utilisateur. Nous constatons que les formules de préférence de CHOMICKY sont des DNF (Formes Normales Disjonctives) c'est à dire une disjonction de conjonctions de formules élémentaires. Dans notre cas les règles ne prendrons en compte que des conjonctions de formules élémentaires. Ainsi on a :

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

avec $E \subset \{1, 2, .., n\}$ où n le nombre d'attributs de $\mathcal{R}$ et $P_{A_k}(t_1, t_2)$ est la conjonction de formules atomiques définies sur l'attribut l'attribut A_k.

Example 11

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.

3.3.1 Cas concret d'une règle de préférence contextuelle

Le but maintenant est de passer d'une règle de préférence sous le formalisme de CHOMICKY à une règle de préférence contextuelle. Comme précédemment définie, une règle de préférence contextuelle R est sous la forme $C \rightarrow X \succ Y$ de telle sorte que si on a $t_1 \succ_R t_2$, C est verifié par t_1 et t_2, X est vérifié seulement par t_1 et Y est vérifié seulement par t_2. Dans notre cas, une règle de préférence contextuelle R' suivant le formalisme de CHOMICKY est sous la forme $C \rightarrow PN$ (peut être écrit $C \wedge PN$) où si on a $t_1 \succ_{R'} t_2$, C prend en compte les conditions communes à t_1 et t_2 et est alors symétrique (ie $C(t_1, t_2) \Rightarrow C(t_2, t_1)$) et PN prend en compte les conditions qui permettent de distinguer t_1 de t_2 donc asymétrique (ie $C(t_1, t_2) \Rightarrow \neg C(t_2, t_1)$). On a alors la définition suivante :

Definition 16. *(Règle de préférence contextuelle suivant le formalisme de CHOMICKY) Une règle de préférence contextuelle suivant le formalisme de CHOMICKY s'ecrit sous la forme :*

$$P(t_1, t_2) = \bigwedge_{k \in E} P_{A_k}(t_1, t_2) = C(t_1, t_2) \wedge \ PN(t_1, t_2)$$

avec

$$\begin{array}{rcl} C(t_1, t_2) & = & \bigwedge\limits_{k \in E} C_{A_k}(t_1, t_2) \ \textit{Conditions vérifiées communément par } t_1 \textit{ et } t_2 \textit{ : symétrique} \\ PN(t_1, t_2) & = & \bigwedge\limits_{k \in E} PN_{A_k}(t_1, t_2) \ \textit{Conditions permettant de distinguer } t_1 \textit{ de } t_2 \textit{ : asymétrique} \end{array}$$

Il faut remarquer que $P_{A_k} = C_{A_k} \wedge PN_{A_k}$

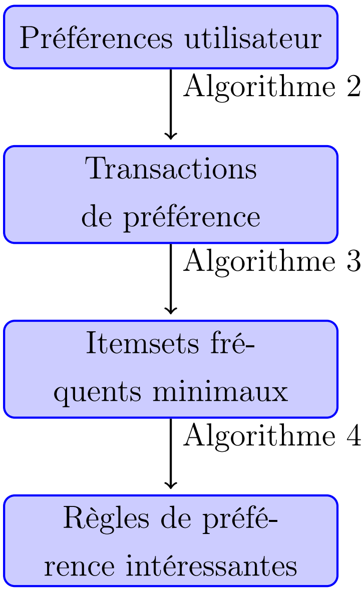
Condition de non redondance :

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} ($P_{A_k} = \bigwedge\limits_{f \in F_k} f$), on a

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

3.4 Processus d'extraction des règles de préférence

Suite à la définition précédente des règles de préférence contextuelles basées sur le formalisme de CHOMICKY, nous allons à présent décrire le processus qui nous permettra d'extraire des règles de préférence d'une base de préférence utilisateur.



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.

Definition 17. *Le* ***support*** *d'un itemset est le nombre de transactions de préférences contenant l'itemset*

$$sup(Itemset) = \frac{| < T_i, T_j > \in \mathcal{P} | Itemset \subset < T_i, T_j > |}{|\mathcal{P}|}$$

La ***confiance*** *d'une règle 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.*

$$conf(Rule) = \frac{| < T_i, T_j > \in \mathcal{P} | T_i \succ_{Rule} T_j|}{| < T_i, T_j > \in \mathcal{P} | T_i \succ_{Rule} T_j| + | < T_i, T_j > \in \mathcal{P} | T_j \succ_{Rule} T_i|}$$

3.4.1 Etapes de transformation des préférences utilisateurs en transaction de préférence

Cette étape consiste à transformer le couple $< t_1, t_2 >$ 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 :

$$\text{Items symétriques (Contexte)} \begin{cases} C : \{x = y\} & \equiv \quad x = y \\ C : \{x \neq y\} & \equiv \quad x \neq y \\ C : \{x, y = a\} & \equiv \quad x = y \wedge x = a \wedge y = a \\ C : \{x, y < a\} & \equiv \quad x < a \wedge y < a \\ C : \{x, y > a\} & \equiv \quad x > a \wedge y > a \end{cases}$$

$$\text{Items asymétriques (Préférences)} \begin{cases} P : \{x < y\} & \equiv \quad x < y \\ P : \{x > y\} & \equiv \quad x > y \\ P : \{x = a\} & \equiv \quad x = a \wedge x \neq y \\ P : \{x < a\} & \equiv \quad x < a \wedge y \geqslant a \\ P : \{x > a\} & \equiv \quad x > a \wedge y \leqslant a \\ N : \{y > a\} & \equiv \quad y > a \wedge x \leqslant a \\ N : \{y < a\} & \equiv \quad y < a \wedge x \geqslant a \\ N : \{y = a\} & \equiv \quad y = a \wedge x \neq y \end{cases}$$

Ces items sont obtenus suivant les conditions définies dans le tableau suivant :

	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	$C : \{x, y < c\},\ \forall c \mid c < a$ $C : \{x, y > c\},\ \forall c \mid c > a$
	$x = a,\ y = b$ avec $a \neq b$	num/symb	$P : \{x = a\}$ $N : \{y = b\}$ $C : \{x \neq y\}$
		num	$P : \{x < y\}$, si $a < b$ $P : \{x > y\}$, si $a > b$ $P : \{x < c\}$, si $a < b$ et $\forall c \mid a < c \leqslant b$ $P : \{x > c\}$, si $a > b$ et $\forall c \mid a > c \geqslant b$ $N : \{y < c\}$, si $a > b$ et $\forall c \mid b < c \leqslant a$ $N : \{y > c\}$, si $a < b$ et $\forall c \mid b > c \geqslant a$ $C : \{x, y < c\},\ \forall c \mid c < a \wedge c < b$ $C : \{x, y > c\},\ \forall c \mid c > a \wedge c > b$
Attr multivalué	$x \in X, y \in Y$	Set(symb)	$C : \{x, y = c\},\ \forall c \in X \cap Y$ $P : \{x = a\},\ \forall a \in X \backslash Y$ $N : \{y = b\},\ \forall b \in Y \backslash X$

Ainsi l'algorithme suivant effectue la transformation des tuples $< t_1, t_2 >$ de préférence vers les transactions de préférence.

Algorithme 2 : Comparaison

Entrées : x , y et l'attribut A_i auquel ils sont liés
Sorties : items issus de la comparaison entre x et y

```
début
  ℰ = ∅ ;                                    // Ensemble des items générés
  si x = y alors
    Ajouter P : {x_{A_i} = y_{A_i}} à ℰ;
    Ajouter P : {x_{A_i} = y_{A_i} = x} à ℰ;
    pour chaque c ∈ DomActif(A_i) et A_i numérique faire
      si x < c alors
        Ajouter C : {x_{A_i}, y_{A_i} < c} à ℰ;
      sinon si x > c alors
        Ajouter C : {x_{A_i}, y_{A_i} > c} à ℰ;
  sinon si x ≠ y alors
    Ajouter C : {x_{A_i} ≠ y_{A_i}} à ℰ;
    Ajouter P : {x_{A_i} = x} à ℰ;
    Ajouter N : {y_{A_i} = y} à ℰ;
    si A_i numérique alors
      si x > y alors
        Ajouter P : {x_{A_i} > y_{A_i}} à ℰ;
      si x < y alors
        Ajouter P : {x_{A_i} < y_{A_i}} à ℰ;
      pour chaque c ∈ DomActif(A_i) faire
        si x < c ∧ y ⩾ c alors
          Ajouter P : {x_{A_i} < c} à ℰ;
        sinon si x ⩽ c ∧ y > c alors
          Ajouter N : {y_{A_i} > c} à ℰ;
        sinon si x > c ∧ y ⩽ c alors
          Ajouter P : {x_{A_i} > c} à ℰ;
        sinon si x ⩾ c ∧ y < c alors
          Ajouter N : {x_{A_i} < c} à ℰ;
        sinon si x < c ∧ y < c alors
          Ajouter C : {x_{A_i}, y_{A_i} < c} à ℰ;
        sinon si x > c ∧ y > c alors
          Ajouter C : {x_{A_i}, y_{A_i} > c} à ℰ;
  retourner ℰ;
```

3.4.2 Etape de l'extraction des règles interressantes minimales des transactions de préférence

Cette étape peut utiliser toute méthode d'extraction d'itemsets intérresaants minimaux. Notre choix s'est porté sur une méthode d'extraction des itemsets fréquents minimaux décrite dans l'article [1], 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

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

3.5 Algorithme

Algorithme 3 : Algorithme d'extraction de règles interressantes

Entrées : Préférences utilisateur $\mathcal{P}$

Sorties : Règles de préférence interressantes $\mathcal{R}$

```
début
    P_S = ∅ ; // Ensemble des transactions obtenues des tuples de préférence
    pour chaque <T,U> ∈ P faire
        P_TU = ∅;                                   // Ensemble des items
        pour chaque A_i ∈ R faire
            x = T.A_i ; // Val correspondante à l'attribut A_i de T
            y = U.A_i;
            P_TU = P_TU ⋃ Comparaison(x, y, A_i);
        Ajouter P_TU à P_S;
    F =ExtraireItemsetFréquents(P_S);
    R =FiltrerRèglesInterressantes(F);
    Retourner R;
```

3.6 Exemple d'extraction des règles de préférence

Nous allons étudier deux situations :

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

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

Prenons un schémas relationnel $\mathcal{R}(\mathcal{A}_1, \mathcal{A}_2)$ avec A_1 et 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 :

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

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

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

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

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

3.7 Determination des itemsets interressants minimaux avec supmin=2 et confmin=0.5

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

$I_4 = \{P : \{x_2 = A\},\}$ sup=4.0 conf=0.8
$I_5 = \{N : \{y_2 = B\},\}$ sup=3.0 conf=0.75
$I_6 = \{P : \{x_1 = b\},\}$ sup=2.0 conf=0.666666666667
$I_7 = \{N : \{y_1 = a\},\}$ sup=2.0 conf=0.666666666667

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

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

Pas d'itemsets interressant minimaux de taille 3

3.7.1 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

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

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

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

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

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

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

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

3.8 Determination des itemsets interressants minimaux avec supmin=2 et confmin=0.5

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

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

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

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

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

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

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

Pas d'itemsets interressant minimaux de taille 4

Références

[1] François Rioult Arnaud Soulet. Extraire les motifs minimaux efficacement et en profondeur.

[2] JAN CHOMICKI. Preference formulas in relational queries.