

Les règles d'association

En collaboration avec

Marie Plasse

Sommaire

Introduction

Algorithmes

Améliorations - Extensions

Une application

Utilisation conjointe de la classification

Indices de pertinence

Introduction

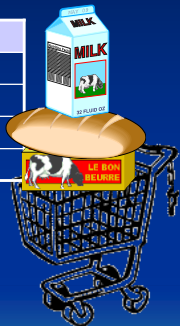
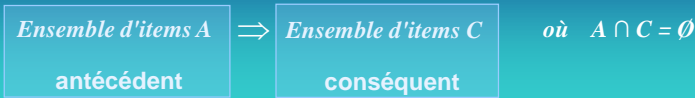
Qu'est ce qu'une règle d'association ?

- Origine Marketing : but = analyser les ventes des supermarchés
- Données en entrée : données de transaction , ticket de caisse

Caddie/ trans	Liste des items (articles) achetés
1	{pain, beurre, lait}
2	{pain,viande}
...	
n	{jus de fruit, poisson, fraises, pain}

- Données en sortie : règles d'association
"lorsqu'un client achète du pain et du beurre,
il achète 9 fois sur 10 du lait en même temps"

$\{ \text{pain, beurre} \} \Rightarrow \{ \text{lait} \}$



Introduction

Notions de base : le support

- Le support indique la fiabilité : % ou nbre de transactions présentant A et C
 $sup(A \Rightarrow C) = P(A \cap C) = P(C / A) \cdot P(A)$
- Exemple :



- Plus le support est élevé, plus la règle est général
- Extraction des règles : choix d'un seuil minimum pour le support : 10%

Les articles d'une règle doivent être présents dans au moins 10 caddies

Donc  = OK car support = 30

Introduction

Notions de base : la confiance

- La confiance indique la précision : % ou nombre de transactions présentant C sachant qu'ils ont A

$$conf(A \Rightarrow C) = P(C / A) = \frac{P(A \cap C)}{P(A)} = \frac{sup(A \Rightarrow C)}{sup(A)}$$

- Exemple :

Sur les 100

On compte ceux qui contiennent

Parmi ceux-ci on compte ceux qui contiennent

Confiance = 90%

- Choix d'un seuil minimum pour la confiance : 50 %

Pour la règle \Rightarrow Confiance = 90 (ou 90% ou "9 sur 10")

Critères d'extraction des règles

- Très grand nombre possible de règles:
 - À 1 , 2, ... , k items ; k = taille max d'une transaction
- Pas pertinent, règles pas intéressantes
- Solution: fixer des seuils pour le support et la confiance = extraction de règles sous contraintes
- Approche retenue dans les algorithmes

Algorithmes

Présentation de Apriori

- 1^{er} algorithme proposé en 1993 (**Agrawal et Srikant**)
- Algorithme **APriori** en 1994 = algorithme fondateur
- Principe de base très simple: 2 étapes
 - Recherche des **sous-ensembles fréquents** (support)
 - Un ensemble fréquent est un ensemble d'items supporté par un % de transactions >ou= minsup
 - Recherche des **règles d'association** (confiance)
- Propriété :
 - Calcule le support d'un minimum d'ensembles en générant les ensembles candidats a priori :
« **tout sous-ensemble d'un ensemble fréquent est également fréquent** »

Si

{A, B, C}

est fréquent, alors

{A, B}

{A, C}

{B, C}

sont fréquents.

Algorithmes

Déroulement pas à pas de Apriori

- Étape 1 : Recherche des sous-ensembles fréquents**

Support min = 40%

3 TRANSACTIONS

N°Transaction	Items
1	A, B, C
2	A, B, D
3	B, D, E

TAILLE 1

Itemset	Support
A	2
B	3
C	1
D	2
E	1

TAILLE 2

Itemset	Support
A, B	2
A, D	1
B, D	2

TAILLE 3

Itemset	Support
A, B, D	1

1/3 ~ 33% < 40 %
- Étape 2 : Recherche des associations**

Confiance min = 70%

Support de la règle

Support antécédent

=

Règles	Support Règle	Support Antécédent	Confiance
A ⇒ B	2	2	100%
B ⇒ A	2	3	2/3 ~ 66 %
B ⇒ D	2	3	2/3 ~ 66 %
D ⇒ B	2	2	100%

Algorithmes


Avantages et limites de Apriori

- ☺ très simple
- ⊗ quelques limites
 - Nombre de lectures important :
 - 1^{ère} lecture: **Calcul du support** des items → ensembles fréquents de longueur 1
 - **Génération des candidats** de longueur 2 à partir des ensembles fréquents de longueur 1
 - 2^{ème} lecture: **Calcul du support** des candidats de longueur 2 → ensembles fréquents de longueur 2
 - ...
 - Arrêt lorsque aucun nouvel ensemble fréquent n'est trouvé
 - Ensemble fréquent de taille maximale k
→ k ou k+1 lectures des données
 - Problèmes de mémoire
- ☺ Nombre minimum de candidats
→ Nombre minimum de calcul du support
- ☺ Nombreuses optimisations proposées

Algorithmes


Optimiser la recherche de sous-ensembles fréquents

- **Enjeux :**
 - Calcul efficace du support
 - Diminuer le nombre d'ensembles dont on calcule le support
 - Mode de calcul
 - Mémoire utilisée durant l'algorithme
 - Nombre de lectures de l'ensemble des données
- **Notions complémentaires :**
lecture horizontale et lecture verticale des données



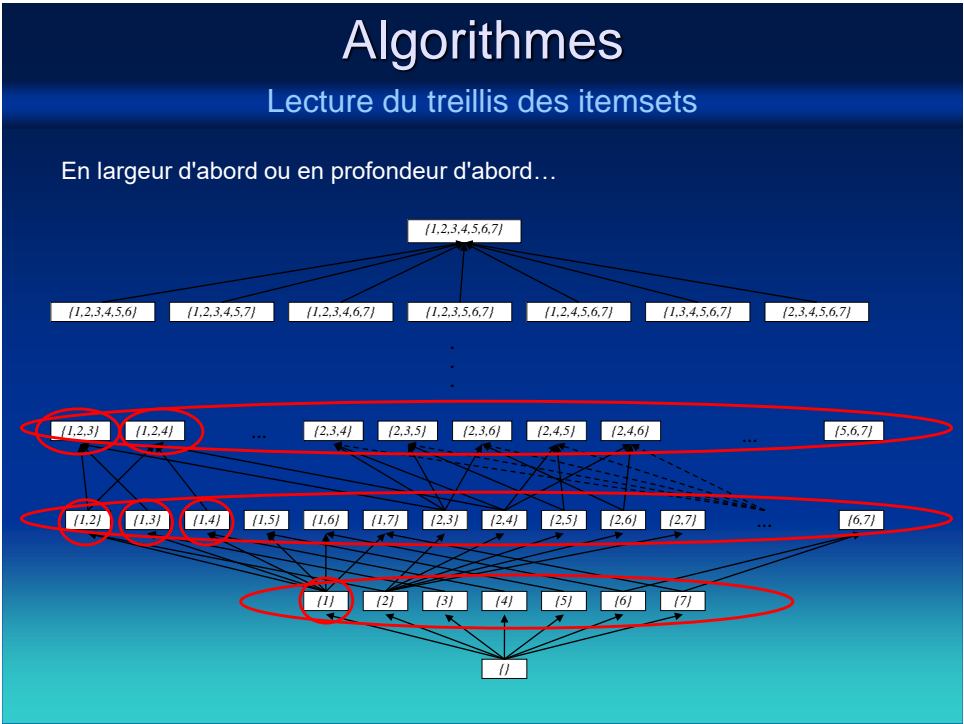
	It1	It2	It3	It4
T 1	1	1	1	0
T 2	1	0	0	1
T 3	0	1	1	1

T1 : It1, It2, It3
T2 : It1, It4
T3 : It2, It3, It4



	It1	It2	It3	It4
T 1	1	1	1	0
T 2	1	0	0	1
T 3	0	1	1	1

It1 : T1, T2
It2 : T1, T3
It3 : T1, T3
It4 : T2, T3



Algorithmes

Sampling (Toivonen, 1996)

- Caractéristiques :**
 - Représentation horizontale des données
 - Parcours en largeur d'abord du treillis
- Principe :**
 - on prend un échantillon aléatoire de transactions,
 - on détermine les items fréquents dans cet échantillon
 - avec $minsup' < minsup$
 - on calcule le support exact des items en utilisant le reste des données
- Bilan :**
 - ⊕ 2 lectures au plus de l'ensemble des données
 - ⊕ Nombre de candidats plus élevé que APriori
 - ⊗ Taille de l'échantillon ?
 - ⊗ Choix de la valeur $minsup'$?


	x1	x2	x3	x4
T1	1	1	1	0
T2	1	0	0	1
T3	0	1	1	1

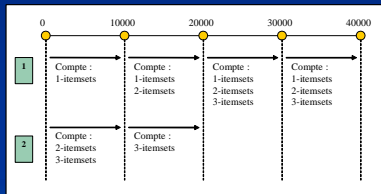
Algorithmes

DIC Dynamic Itemset Counting (Brin & Motwani, 1997)

- Caractéristiques :**
 - Représentation horizontale des données
 - Parcours en largeur d'abord du treillis
- Principe :** compte les itemsets fréquents dynamiquement. Algo complexe
 - on commence à calculer le support d'un ensemble candidat dès que cela est possible
- Bilan :**
 - ☺ Nombre de lectures inférieur ou égal à celui d'APriori
 - ☺ Nombre minimum de calculs du support
 - ☹ Tests coûteux à mettre en place à chaque étape
 - ☹ Nombre de lectures de l'ensemble des données supérieur à ceux de Sampling et Partition

	R1	R2	R3	R4
T1	1	1	1	0
T2	1	0	0	1
T3	0	1	1	1



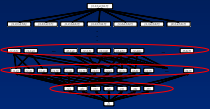


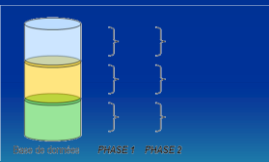
Algorithmes

Partition (Savarese, Omiecinski & Navathe, 1995)

- Caractéristiques :**
 - Représentation verticale des données
 - Parcours en largeur d'abord du treillis
- Principe :**
 - Partitionne la base
 - Calcul des ensembles fréquents locaux, ils forment les ensembles candidats globaux
 - Calcul du support exact des ensembles candidats globaux => ensembles fréquents globaux
- Bilan :**
 - ☺ Calcul rapide du support pour un candidat donné
 - ☺ Nombre de lectures égal à 2
 - ☹ Nombre de candidats plus élevé que APriori (si nb partitions > 1)
 - ☹ Choix du nombre de partitions ?

	R1	R2	R3	R4
T1	1	1	1	0
T2	1	0	0	1
T3	0	1	1	1





Algorithmes

Eclat (Zaki, 2000)

- Caractéristiques :**
 - Représentation verticale des données
 - Parcours en profondeur d'abord du treillis
- Principe :**
 - Division du treillis des itemsets en parties disjointes pouvant être résolues indépendamment en mémoire
- Bilan :**
 - ☺ Calcul du support rapide pour un ensemble donné
 - ☺ Faible besoin mémoire
 - ☹ Pas d'élagage des candidats possible → nombre de candidats > à APriori

	I1	I2	I3	I4
T1	1	1	1	0
T2	1	0	0	1
T3	0	1	1	1

Algorithmes

FP-Growth (Han & Pei, 2003)

- Caractéristiques :**
 - Représentation verticale des données
 - Parcours en profondeur d'abord du treillis
- Principe :**
 - Résume les données dans une structure appelée FP-Tree
- Bilan :**
 - ☺ Calcul rapide du support pour un candidat donné
 - ☺ Besoin global en mémoire faible
 - ☹ Structure de données compliquée

	I1	I2	I3	I4
T1	1	1	1	0
T2	1	0	0	1
T3	0	1	1	1

Algorithmes						
Synthèse pour la recherche d'ensembles fréquents						
	Nombre de candidats générés	Simplicité de la structure de données	Nombre de lectures de l'ensemble de données	Mémoire	Calcul support	Remarques
REPRESENTATION HORIZONTALE ET PARCOURS EN LARGEUR						
APriori	+	-	k ou $k+1$	-	-	Nombreuses optimisations Nombreuses lectures de la base
Sampling	-	-	≤ 2	-	-	Nombre de candidats élevés Taille échantillon ? Choix de <i>minsup</i> ?
DIC	+	-	$\leq k$ ou $k+1$	-	-	Test coûteux
REPRESENTATION VERTICALE ET PARCOURS EN LARGEUR						
Partition	-	+	2	-	+	Choix nombre et taille des partitions ?
REPRESENTATION VERTICALE ET PARCOURS EN PROFONDEUR						
Eclat	-	+	1	+	+	
FP-Growth	-	-	1	+	+	

Algorithmes

Génération des règles d'association

- Objectif
 - Générer, à partir des ensembles fréquents, les règles d'association valides (= ayant une confiance supérieure ou égale à minconf.)
- Algorithme proposé en 1994 par Agrawal et Srikant en même temps que APriori
 - Soit EF un ensemble fréquent
 - C un ensemble inclus dans EF
 - C' un ensemble inclus dans C

Résultat : Pour qu'une règle de la forme $EF - c \rightarrow c$ soit valide, il faut que toutes les règles de la forme $EF - c' \rightarrow c'$ soient valides

- Principe :
 - Génération de toutes les règles ayant une conséquence de longueur 1 à partir d'un ensemble fréquent donné
 - Génération des conséquences de longueur 2 (même principe que la génération des candidats d'APriori)
 - Appels récursifs
 - Passage à l'ensemble fréquent suivant

Algorithmes

Génération des règles d'association : Exemple

- on considère l'ensemble fréquent {1, 2, 4}

Ens. Fréq.	Sup
{1, 2, 4}	3
{1, 2}	4
{1, 4}	3
{2, 4}	5
{1}	5
{2}	6
{4}	5

- Conséquents de taille 1 :

$$\text{conf}(\{1,2\} \rightarrow 4) = \frac{\text{sup}(\{1,2,4\})}{\text{sup}(\{1,2\})} = \frac{3}{4}$$

$$\text{conf}(\{1,4\} \rightarrow 2) = \frac{\text{sup}(\{1,2,4\})}{\text{sup}(\{1,4\})} = 1$$

$$\text{conf}(\{2,4\} \rightarrow 1) = \frac{\text{sup}(\{1,2,4\})}{\text{sup}(\{2,4\})} = \frac{3}{5}$$

- Conséquents de taille 2 :

$$\text{conf}(1 \rightarrow \{2,4\}) = \frac{\text{sup}(\{1,2,4\})}{\text{sup}(\{1\})} = \frac{3}{5}$$

Confiance min = 3/4

Améliorations

Quelques extensions

- Introduction de taxinomies
 - Chaussures taille 40, chaussures taille 42 sont des Chaussures
 - Pantalon, jupe sont des vêtements
- Introduction de séquences
 - {Télévision, magnétoscope} => {VHS vierges}
 - {DVD "Un nouvel espoir", DVD "L'empire contre attaque"} => {DVD "Le retour du jedi"}
- Introduction d'autres types de données que les données binaires
 - Variables catégorielles

Données de transactions

transactions	items
A	1, 2
B	1, 3
C	2

Variables booléennes

	1	2	3
A	1	1	0
B	1	0	1
C	0	1	0

Variable catégorielle couleur

Couleur
Vert
Jaune
Rouge
Bleu

- 4 modalités
- 4 items

Si couleur =Jaune :

Vert	Jaune	Rouge	Bleu
0	1	0	0

Application

Problématique

- + de 80000 véhicules décrits par plus de 3000 attributs binaires :
- Données "clairsemées" :

Véhicules	A1	A2	A2	A2	A3	...	AP
	1	0	0	1	0		0
	0	0	1	1	0		0
	0	1	0	0	1		0
	1	0	0	0	1		0
	0	1	0	0	0		1
	0	1	0	0	0		0
	0	0	1	0	0		0

- Trouver des corrélations

PSA PEUGEOT CITROËN

Application

Événements rares → faibles valeurs du support

- Exemple : 20 caddies contenant diverses combinaisons d'articles

- Support de 30% (6 caddies minimum)
 - Articles les + fréquents

- Baisse du support à 15% (3 caddies minimum)
 - + de règles
 - Articles plus rares

11

Application

Recherche de règles d'association

- Utilisation de Apriori optimisé et de Eclat :

Nombre minimum de véhicules vérifiant une règle (support minimum)	Confiance minimum	Nombre de règles	Complexité maximum (nombre d'attributs contenus dans la règle)
500	50 %	16	3
400	50 %	29	3
300	50 %	194	5
200	50 %	102 981	10
100	50 %	1 623 555	13

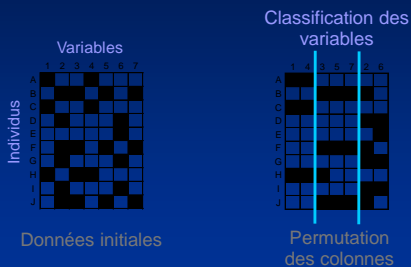
- Objectifs :
 - Réduire le nombre de règles
 - Simplifier les règles

Support minimum	Confiance minimum	Nombre de règles	Complexité maximum
100	50 %	600636	12

- Utilisation d'une classification de variables préalable

La classification de variables

Objectif

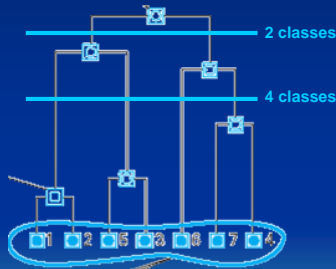


- Découvrir une structure de l'espace des variables
- Obtenir des groupes de variables corrélées
- Utile pour :
 - Rechercher les multicolinéarités entre variables
 - Réduire le nombre de variables s'il est très important
 - Transformer les variables en dimensions indépendantes

La classification de variables

Méthodes hiérarchiques : algorithme agglomératif

- Hiérarchie de partitions emboîtées : arbre hiérarchique
- Coupure horizontale de l'arbre : partition
- Algorithme :
 - 1 Calcul des distances
 - 2 Agrégation des deux éléments les plus proches
 - 3 Répétition des étapes 1 et 2 jusqu'à obtention d'une seule classe
- Choix :
 - d'une mesure de ressemblance
 - d'une stratégie d'agrégation



La classification de variables

Méthodes hiérarchiques : algorithme agglomératif

- Choix d'une mesure de similarité entre variables selon leur type :
 - Numériques : coefficient de corrélation linéaire
 - Ordinales (transformation en rangs)
 - Fréquences : distance du χ^2 , coefficient d'affinité, distance de Hellinger
 - Binaires : Φ^2 de Pearson, ou indices type Jaccard...

– Indice de Russel Rao : $S_{RR}(j, j') = \frac{p_{11}}{p}$

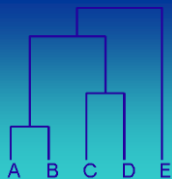
– Indice de Jaccard : $S_J(j, j') = \frac{p_{11}}{p_{11} + p_{10} + p_{01}}$

Variable j	Variable j'		
	1	0	
1	p_{11}	p_{10}	$p_{1.}$
0	p_{01}	p_{00}	$p_{0.}$
	$p_{.1}$	$p_{.0}$	$\Sigma = p$

- À transformer en indices de dissimilarité :

$$D_{jj'} = \max_{j, j'} (S_{jj'}) - S_{jj'}$$

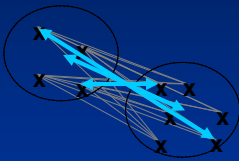
- Nominales (codage disjonctif)



La classification de variables

Méthodes hiérarchiques : algorithme agglomératif

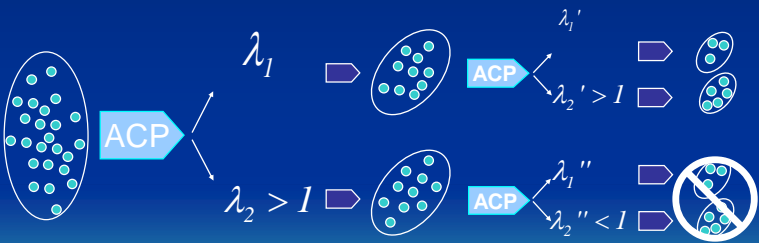
- Calcul des distances entre deux classes :
 - Saut minimum
 - Saut maximum
 - Saut moyen
- Critère de Ward (distances euclidiennes)
 - L'inertie est une mesure de la dispersion du nuage de points
 - Inertie totale = inertie intra classe + inertie inter classe
la + faible possible la + forte possible
 - Inertie interclasse = Somme des carrés des distances entre les centres de gravité de chaque groupe et le centre de gravité global
 - Un regroupement entraîne une baisse de l'inertie interclasse
 - Agrégation des éléments qui engendre la plus faible perte d'inertie interclasse



La classification de variables

Méthodes hiérarchiques : algorithme divisif

- Recherche d'une partition en classes unidimensionnelles
- Regroupement autour de la composante principale avec laquelle la variable présente le plus fort coefficient de corrélation



La classification de variables

Méthodes de partitionnement

- Partition directe des variables en un nombre de classes k connu
- Méthode de Qannari et Vigneau
- Recherche K classes G_1, G_2, \dots, G_K et K variables latentes c_1, c_2, \dots, c_K associées
- en rendant maximum un critère exprimant la colinéarité entre une variable d'une classe et la variable latente correspondante.
- Cet algorithme itératif s'apparente un peu à l'algorithme des centres mobiles (Forgy, 1965) :

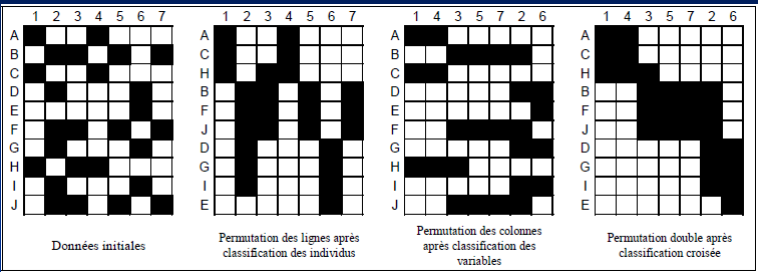
Algorithme général :

Initialisation : Choix d'une partition initiale

- 1 Calcul des centres des classes
- 2 Affectation des points au centre le plus proche
- 3 Étapes 1 et 2 jusqu'à convergence

La classification croisée

Objectif



- Découvrir une structure de l'espace des variables et des individus
- Obtenir des groupes de variables corrélées et des groupes homogènes d'individus : blocs ou biclusters

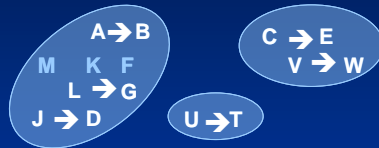
Algorithme itératif (K-means):

Optimisation alternée de la partition des individus en bloquant celle des variables puis de la partition des variables en fixant celle des individus

Application

Avec classification préalable des variables

- Objectif : regrouper les variables en un nombre restreint de classes homogènes



- Identification d'une classe atypique de variables
 - Règles complexes en grand nombre
- Règles obtenues en dehors de cette classe :

	Nombre de règles	Complexité maximum	Réduction du nombre de règles
Sans classification : Rappel premier résultat	1 623 555	13	.
Sans classification : regroupement manuel	600636	12	60%
Avec classification préalable	218	4	99%

Indices de pertinence

Sélection des "meilleures" règles

- Pour faire valider les règles par un expert du terrain, il faut sélectionner les "meilleures" règles
- On peut les classer par ordre décroissant de leur intérêt statistique
- Il existe plusieurs indices pour évaluer la pertinence des règles
- comment en choisir un ?

Insuffisance de l'approche support confiance

Le choix de la valeur de *minsup* est difficile puisque le support est le premier critère de sélection des règles. Un seuil élevé pour le support risque de disqualifier certaines règles très intéressantes, ayant un support faible mais une confiance très élevée. Par exemple, l'item "*caviar*" étant rare, il ne sera pas retenu alors qu'il peut cacher une règle du type "*caviar*→*vodka*" valable dans 85% des cas. D'un point de vue marketing, il est dommage de passer à côté d'une telle règle car le caviar est un produit de luxe. Sachant que l'achat de l'un entraîne de manière quasi sûre l'achat de l'autre, il pourrait être pertinent, par exemple, de faire une promotion sur la vodka lors de l'achat de caviar.

A l'inverse, fixer un seuil très bas pour le support peut entraîner l'extraction de règles sans intérêt du type "*caviar*→*lait*". En effet, le lait est un produit très commun qui figure dans un nombre élevé de transactions. Il n'est donc pas surprenant de le trouver dans le caddie de tous les clients qui

Insuffisance de l'approche support confiance

Dans la règle "*caviar*→*lait*", le support mesure la fréquence globale des exemples de la règle ; il est donc faible étant donné la faible fréquence d'achat du produit caviar. La confiance est proche de 100% puisqu'elle s'intéresse à la répartition des achats de *caviar* entre les achats de *lait* et les "non achats" de *lait*. La répartition des "non achats" de caviar n'est pas du tout prise en compte alors qu'elle permettrait de discriminer ce type de règles.

- L'extraction des règles est basée sur le support et la confiance – insuffisant pour mesurer la pertinence ou l'intérêt d'une règle (exemple Hébrail et exo)
 - Il existe plusieurs indices pour évaluer la pertinence des règles
 - Un des plus connus et utilisés : le **lift** (Brin et al., 1997)

Indices de pertinence

Mesurer l'intérêt des règles

$$lift(A \Rightarrow C) = \frac{P(A \cap C)}{P(A) \cdot P(C)}$$

Exemple :



$lift = 2 \Rightarrow$ les transactions contenant + + sont 2 fois plus nombreuses que si l'achat de + et l'achat de étaient indépendants.

	C	\overline{C}	Profils lignes
A	0,07	0,01	0,08
\overline{A}	0,33	0,59	0,92
Profils colonnes	0,4	0,6	1

Indices	Valeur
Confiance	0,88
Confiance centrée	0,48
Lift	2,19
Multiplicateur de cotes	10,5
Loevinger	0,79

Figure III- 6 : Valeurs des indices sur un cas d'école

considérons 100 consommateurs : 8 ont acheté du caviar, 40 ont acheté du lait et 7 ont acheté les

Autres indices de pertinence

Quelques indices

- Tableau de «contingence » :

A→C	C	\overline{C}	Profils lignes
A	$P(AC)$	$P(A\overline{C})$	$P(A)$
\overline{A}	$P(\overline{A}C)$	$P(\overline{A}\overline{C})$	$P(\overline{A})$
Profils colonnes	$P(C)$	$P(\overline{C})$	1

- Lift : $Lift = \frac{P(AC)}{P(A) \cdot P(C)} = \frac{\lambda_A}{P(C)}$ $\lambda_A = P(C/A)$ et $\lambda_C = P(A/C)$.

- Confiance centrée : $Confcen = \frac{P(AC) - P(A)P(C)}{P(A)} = \lambda_A - P(C)$

- Loevinger : $Loe = \frac{\frac{P(AC) - P(A)}{P(A)} - P(C)}{1 - P(C)} = \frac{\lambda_A - P(C)}{1 - \lambda_A}$

- Multiplicateur de cotes : $MC = \frac{P(AC) - P(C)P(A)}{P(A)P(C) - P(C)P(AC)} = \frac{\lambda_A(1 - P(C))}{P(C)(1 - \lambda_A)}$

- Jaccard : $Jac = \frac{P(AC)}{P(A) + P(C) - P(AC)} = \frac{1}{\frac{1}{\lambda_A} + \frac{1}{\lambda_C} - 1}$

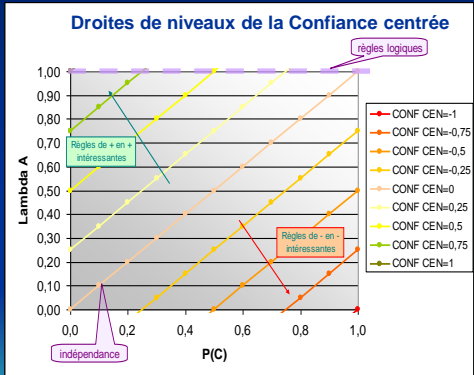
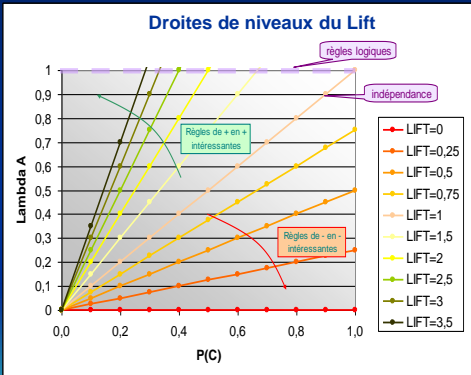
Comparaison graphique des indices

Le lift et la confiance centrée

Rappel : $\lambda_A = P(C/A)$
 $\lambda_C = P(A/C)$

$\lambda_A = Lift \cdot P(C)$

$\lambda_A = Confcen + P(C)$



- Lorsque λ_A est élevée, le lift et la confiance centrée favorisent les règles où le conséquent est peu fréquent mais ne sont pas assez sévères dans notre cas.

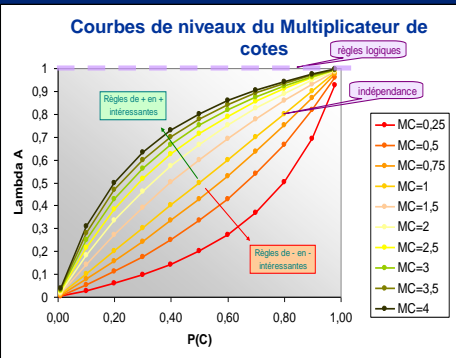
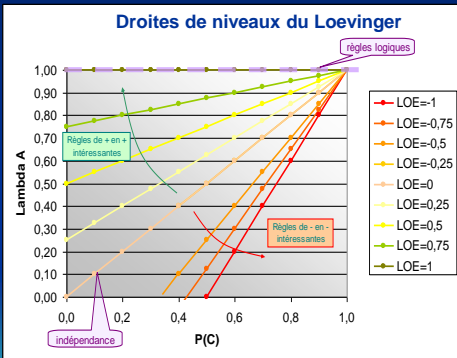
Comparaison graphique des indices

Le loevinger et le multiplicateur de cotes

Rappel : $\lambda_A = P(C/A)$
 $\lambda_C = P(A/C)$

$\lambda_A = Loe(1 - P(C)) + P(C)$

$\lambda_A = \frac{MC \cdot P(C)}{1 - P(C) + MC \cdot P(C)}$



- Lorsque λ_A est élevée, le Loevinger et le Multiplicateur de cotes prennent des valeurs élevées.

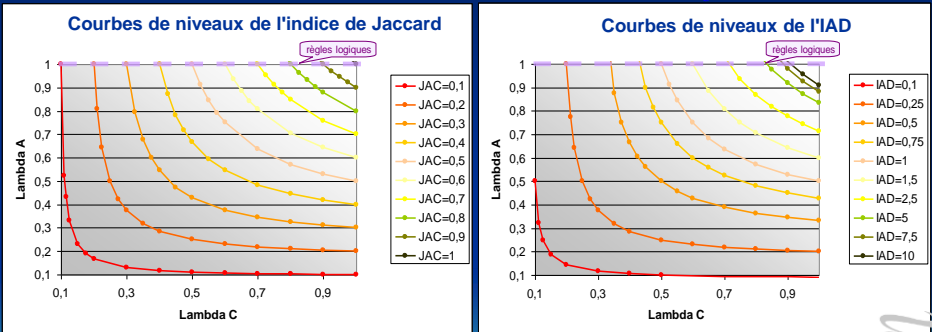
Comparaison graphique des indices

L'indice de jaccard et l'indice d'accords-désaccords

$$\lambda_A = \frac{I}{\frac{I}{IAC} - \frac{I}{\lambda_C} + I}$$

$$\lambda_A = \frac{I}{\frac{I}{IAD} - \frac{I}{\lambda_C} + 2}$$

Rappel : $\lambda_A = P(C/A)$
 $\lambda_C = P(A/C)$



- Prise en compte simultanée des deux probabilités conditionnelles
- Favorisent où $P(A)$, $P(C)$ et $P(A \cap C)$ sont proches.

Quelques références

- Agrawal R., Srikant R. (1994) *Fast Algorithms for Mining Association Rules*. In : Proceedings of the 20th Int'l Conference on Very Large Databases (VLDB), Santiago, Chile.
- Hébrail G., Lechevallier Y. (2003) Data mining et analyse des données. In : Govaert G. Analyse des données. Ed. Lavoisier, Paris, pp 323-355
- Vigneau E., Qannari E.M. (2003) *Clustering of variables around latent component - application to sensory analysis*. Communications in Statistics , Simulation and Computation, 32(4), pp 1131-1150
- Nakache J.P., Confais J. (2005) *Approche pragmatique de la classification*, Ed. Technip, Paris

Utilisation de R

- R permet de travailler avec les algorithmes classiques.
- Nous allons utiliser R et Rstudio sur un cas pratique