Règles d'association

Sommaire

Exemple : Analyse du panier de la ménagère

- Découverte d'**associations** et de **corrélations** entre les articles achetés par les clients en analysant les achats effectués (panier)

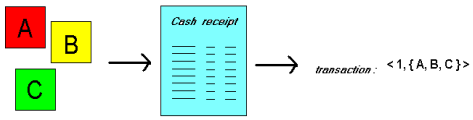
Exemple : Analyse du panier de la ménagère

Etant donnée :

- Une base de données de transactions de clients, où chaque transaction est représentée par un ensemble d'articles -**set of items-** (ex., produits)

Trouver :

- Groupes d'articles (itemset) achetés fréquemment (ensemble)



Exemple : Analyse du panier de la ménagère

Extraction d'informations sur le comportement de clients

- SI achat de riz + vin blanc ALORS achat de poisson (avec une grande probabilité)

Intérêt de l'information : peut suggérer ...

- Disposition des produits dans le magasin
- Quels produits mettre en promotion, gestion de stock, …

Approche applicable dans d'autres domaines

- Cartes de crédit, e-commerce, …
- Services des compagnies de télécommunication
- Services bancaires
- Traitements médicaux, …

Règles d'associations

Recherche de règles d'association :

- Découvrir des patterns, corrélations, associations fréquentes, à partir d'ensembles d'items contenus dans des base de données.

Compréhensibles : Facile à comprendre

Utiles : Aide à la décision

Efficaces : Algorithmes de recherche

Applications :

- Analyse des achats de clients, Marketing, Accés Web, Design de catalogue, Génomique, etc.

Règles d'associations

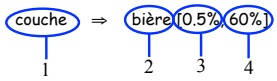
Formats de représentation des règles d'association :

- couches ⇒ bière [0.5%, 60%]
- achète:couches ⇒ achète:bière [0.5%, 60%]
- "SI achète couches ALORS achète bière dans 60% de cas. Les couches et la bière sont tous deux achetés dans 0.5% des transactions de la base de données."

Autres représentations (utilisée dans l'ouvrage de Han) :

- achète(x, "couches") ⇒ achète(x, "bière") [0.5%, 60%]

Règles d'associations



"**SI** achète couche, **ALORS** achète bière, dans 60% de cas, dans 0.5% de la base"

Condition, partie gauche de la règle
Conséquence, partie droite de la règle
Support, fréquence ("partie gauche **et** droite sont présentes ensemble dans la base")
Confiance ("si partie gauche de la règle est vérifiée, probabilité que la partie droite de la règle soit vérifiée")

Règles d'associations

- **Support :** % d'instances de la base vérifiant la règle.

support(A ⇒ B [s, c]) = p(A et B) = support ({A,B})

- **Confiance :** % d'instances de la base vérifiant l'implication

confiance(A ⇒ B [s, c]) = p(B|A) = p(A et B) / p(A) = support({A,B}) / support({A})

Exemple

TID	Items
1	Pain, Lait
2	Bière, Couches, Pain, Oeufs
3	Bière, Coca, Couches, Lait
4	Bière, Pain, Couches, Lait
5	Coca, Pain, Couches, Lait

$\{Couches, \text{Lait}\} \Rightarrow_{s,\alpha} Bière$

Règle : $X \Rightarrow_{s,\alpha} y$

Support : $s = \frac{\sigma(X \cup y)}{|T|} (s = P(X, y))$

Confiance : $\alpha = \frac{\sigma(X \cup y)}{\sigma(X)|} (\alpha = P(y|X))$

$s = \frac{\sigma(Couches, \text{Lait}, \text{Bière})}{\text{Nombre total d'instances}} = \frac{2}{5} = 0.4$

$\alpha = \frac{\sigma(Couches, \text{Lait}, \text{Bière})}{\sigma(Couches, \text{Lait})|} = 0.66$

Règles d'associations

Support minimum σ :

- Elevé ⇒ peu d'itemsets fréquents
- ⇒ peu de règles valides qui ont été souvent vérifiées
- Réduit ⇒ plusieurs règles valides qui ont été rarement vérifiées

Confiance minimum γ :

- Elevée ⇒ peu de règles, mais toutes "pratiquement" correctes
- Réduite ⇒ plusieurs règles, plusieurs d'entre elles sont "incertaines"

Valeurs utilisées : σ = 2 - 10 %, γ = 70 - 90 %

Recherche de règles d'association

Données d 'entrée : liste d 'achats

Achat = liste d 'articles (longueur variable)

	Produit A	Produit B	Produit C	Produit D	Produit E
Achat 1	★			★	
Achat 2	★	★	★		
Achat 3	★				★
Achat 4	★			★	★
Achat 5		★		★	

Recherche de règles d'association

13

Tableau de co-occurrence : combien de fois deux produits ont été achetés ensemble ?

	Produit A	Produit B	Produit C	Produit D	Produit E
Produit A	4	1	1	2	2
Produit B	1	2	1	1	0
Produit C	1	1	1	0	0
Produit D	2	1	0	3	1
Produit E	2	0	0	1	2

Illustration / Exemple

14

- Règle d 'association :
  - Si A alors B (règle 1)
  - Si A alors D (règle 2)
  - Si D alors A (règle 3)
- Supports :
  - Support(1)=20% ; Support(2)=Support(3)=40%
- Confiances :
  - Confiance(2) = 50% ; Confiance(3) = 67%
- On préfère la règle 3 à la règle 2.

Description de la méthode

15

- Support et confiance ne sont pas toujours suffisants
- Ex : Soient les 3 articles A, B et C

article	A	B	C	A et B	A et C	B et C	A, B et C
support	45%	42,5%	40%	25%	20%	15%	5%

- Règles à 3 articles : même support 5%
- Confiance
  - Règle : Si A et B alors C = 0.20
  - Règle : Si A et C alors B = 0.25
  - Règle : Si B et C alors A = 0.33

Recherche de règles

16

- Soient une liste de n articles et de m achats.
- 1. Calculer le nombre d'occurrences de chaque article.
- 2. Calculer le tableau des co-occurrences pour les paires d 'articles.
- 3. Déterminer les règles de niveau 2 en utilisant les valeurs de support, confiance et amélioration.
- 4. Calculer le tableau des co-occurrences pour les triplets d 'articles.
- 5. Déterminer les règles de niveau 3 en utilisant les valeurs de support, confiance et amélioration
- ...

Complexité

17

- Soient :
  - n : nombre de transactions dans la BD
  - m : Nombre d'attributs (items) différents
- Complexité
  - Nombre de règles d'association : $O(\mathbf{m}.2^{\mathbf{m}-1})$
  - Complexité de calcul : $O(\mathbf{n}.\mathbf{m}.2^{\mathbf{m}})$

Réduction de la complexité

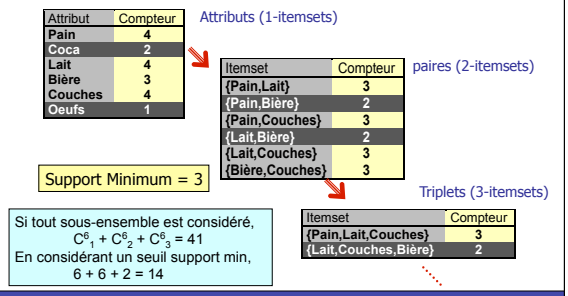
18

- n de l'ordre du million (parcours de la liste nécessaire)
- Taille des tableaux en fonction de m et du nombre d 'articles présents dans la règle

	2	3	4
n	n(n-1)/2	n(n-1)(n-2)/6	n(n-1)(n-2)(n-3)/24
100	4950	161 700	3 921 225
10000	5.10^7	$1.7\ 10^{11}$	$4.2\ 10^{14}$

- Conclusion de la règle restreinte à un sous-ensemble de l 'ensemble des articles vendus.
  - Exemple : articles nouvellement vendues.
- Création de groupes d 'articles (différents niveaux d'abstraction).
- Elagage par support minimum.

Illustration sur une BD commerciale



Attribut	Compteur
Pain	4
Coca	2
Lait	4
Bière	3
Couches	4
Oeufs	1

Itemset	Compteur
{Pain,Lait}	3
{Pain,Bière}	2
{Pain,Couches}	3
{Lait,Bière}	2
{Lait,Couches}	3
{Bière,Couches}	3

Itemset	Compteur
{Pain,Lait,Couches}	3
{Lait,Couches,Bière}	2

L'algorithme Apriori [Agrawal93]

- Deux étapes
  - Recherche des k-itemsets fréquents (support≥MINSUP)
    - (Pain, Fromage, Vin) = 3-itemset
    - Principe : Les sous-itemsets d'un k-itemset fréquent sont obligatoirement fréquents
  - Construction des règles à partir des k-itemsets trouvés
    - Une règle fréquente est retenue si et seulement si sa confiance c≥ MINCONF
    - Exemple : ABCD fréquent
    - AB → CD est retenue si sa confiance ≥ MINCONF

Recherche des k-itemsets fréquents (1)

Exemple
- I = {A, B, C, D, E, F}
- T = {AB, ABCD, ABD, ABDF, ACDE, BCDF}
- MINSUP = 1/2

Calcul de L1 (ensemble des 1-itemsets)
- C_1 = I = {A,B,C,D,E,F} // C1 : ensemble de 1-itemsets candidats
- s(A) = s(B) = 5/6, s(C) = 3/6, s(D) = 5/6, s(E) = 1/6, s(F) = 2/6
- L_1 = {A, B, C, D}

Calcul de L2 (ensemble des 2-itemsets)
- C_2 = L1xL1 = {AB,AC, AD, BC, BD, CD}
- s(AB) = 4/6, s(AC) = 2/6, s(AD) = 4/6, s(BC) = 2/6, s(BD) = 4/6, s(CD) = 3/6
- L_2 = {AB,AD, BD, CD}

Recherche des k-itemsets fréquents (2)

- Calcul de L_3 (ensemble des 3-itemsets)
  - C_3 = {ABD} (ABC ∉ C_3 car AC ∉ L_2)
  - s(ABD) = 3/6
  - L_3 = {ABD}
- Calcul de L_4 (ensemble des 4-itemsets)
  - C_4 = ϕ
  - L_4 = ϕ
- Calcul de L (ensembles des itemsets fréquents)
  - L = ∪L_i = {A, B, C, D, AB, AD, BD, CD, ABD}

L'algorithme Apriori

L1 = {1-itemsets fréquents};
for (k=2; Lk-1≠ ϕ; k++) **do**
  Ck = apriori_gen(Lk-1);
  forall instances t∈T **do**
    Ct = subset(Ck,t);
    forall candidats c ∈ Ct **do**
      c.count++;
  Lk = { c∈ Ck / c.count ≥ MINSUP }
L = ∪iLi;

La procédure Apriori_gen

{ Jointure L_{k-1} * L_{k-1} ; k-2 éléments communs}
insert into C_k;
select p.item$_1$, p.item$_2$, …, p.item$_{k-1}$, q.item$_{k-1}$
from L_{k-1p}, L_{k-1q}
where p.item$_1$=q.item$_1$, …, p.item$_{k-2}$=q.item$_{k-2}$
  , p.item$_{k-1}$< q.item$_{k-1}$
forall itemsets c ∈ C_k **do**
  forall (k-1)-itemsets s⊂c **do**
    if s∉L_{k-1} **then**
      delete c **from** C_k;

Apriori - Exemple

25

Base de données D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

Apriori - Exemple

26

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

Apriori - Exemple

27

C_3

itemset
{2 3 5}

Scan D

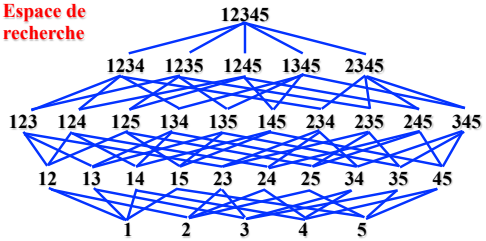
L_3

itemset	sup
{2 3 5}	2

Apriori - Exemple

28

Espace de recherche



Apriori - Exemple

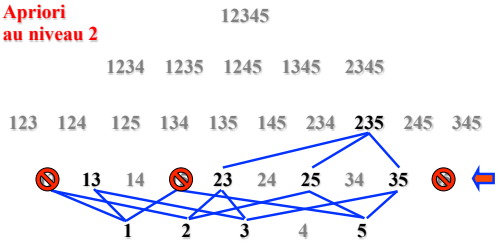
29

Apriori au Niveau 1

Apriori - Exemple

30

Apriori au niveau 2



Génération des règles à partir des itemsets

31

Pseudo-code :

- pour chaque itemset fréquent *l*
- générer tous les sous-itemsets non vides *s* de *l*
- pour chaque sous-itemset non vide *s* de *l*
- produire la règle "*s* ⇒ (*l*-*s*)" si *support(l)/support(s)* ≥ *min_conf*", où *min_conf* est la confiance minimale

- **Exemple : itemset fréquent l = {abc},**
- **Sous-itemsets s = {a, b, c, ab, ac, bc)**
  - a ⇒ bc, b ⇒ ac, c ⇒ ab
  - ab ⇒ c, ac ⇒ b, bc ⇒ a

Génération des règles à partir des itemsets

32

Règle 1 à mémoriser :

- La génération des itemsets fréquents est une opération coûteuse
- La génération des règles d'association à partir des itemsets fréquents est rapide

Règle 2 à mémoriser :

- Pour la génération des itemsets, le seuil support est utilisé.
- Pour la génération des règles d'association, le seuil confiance est utilisé.

Complexité en pratique ?

- A partir d'un exemple réel (petite taille) …
- Expériences réalisées sur un serveur Alpha Citum 4/275 avec 512 MB de RAM & Red Hat Linux release 5.0 (kernel 2.0.30)

Apriori - Complexité

33

Phase coûteuse : Génération des candidats

- Ensemble des candidats de grande taille :
  - 10^4 1-itemset fréquents génèrent 10^7 candidats pour les 2-itemsets
  - Pour trouver un itemset de taille 100, e.x., {a1, a2, …, a100}, on doit générer $2^{100} \approx 10^{30}$ candidats.
- Multiple scans de la base de données :
  - Besoin de (n +1) scans, n est la longueur de l'itemset le plus long

Apriori - Complexité

34

En pratique :

- Pour l'algorithme Apriori basique, le nombre d'attributs est généralement plus critique que le nombre de transactions
- Par exemple :
  - 50 attributs chacun possédant 1-3 valeurs, 100.000 transactions (not very bad)
  - 50 attributs chacun possédant 10-100 valeurs, 100.000 transactions (quite bad)
  - 10.000 attributs chacun possédant 5-10 valeurs, 100 transactions (very bad...)
- Notons :
  - Un attribut peut avoir plusieurs valeurs différentes
  - Les algorithmes traitent chaque paire attribut-valeur comme un attribut (2 attributs avec 5 valeurs → "10 attributs"

Quelques pistes pour résoudre le problème …

Apriori – Réduction de la complexité

35

Suppression de transactions :

- Une transaction qui ne contient pas de k-itemsets fréquents est inutile à traiter dans les parcours (scan) suivants.

Partitionnement :

- Tout itemset qui est potentiellement fréquent dans une BD doit être potentiellement fréquent dans au moins une des partitions de la BD.

Echantillonage :

- Extraction à partir d'un sous-ensemble de données, décroitre le seuil support

Apriori - Avantages

36

- Résultats clairs : règles faciles à interpréter.
- Simplicité de la méthode
- Aucune hypothèse préalable (Apprentissage non supervisé)
- Introduction du temps : méthode facile à adapter aux séries temporelles. Ex : Un client ayant acheté le produit A est susceptible d 'acheter le produit B dans deux ans.

Apriori - Inconvénients

37

- Coût de la méthode : méthode coûteuse en temps
- Qualité des règles : production d'un nombre important de règles triviales ou inutiles.
- Articles rares : méthode non efficace pour les articles rares.
- Adapté aux règles binaires
- Apriori amélioré
  - Variantes de Apriori : DHP, DIC, etc.
  - Partition [Savasere et al. 1995]
  - Eclat et Clique [Zaki et al. 1997]
  - …

Typologie des règles

38

- Règles d'association binaires
  - Forme : *if* C *then* P. C,P : ensembles d'objets
- Règles d'association quantitatives
  - Forme : *if* C *then* P
    - C = terme1 & terme2 & … & termen
    - P = termen+1
    - termei = <attributj, *op*, valeur> ou <attributj, *op*, valeur_de, valeur_a>
  - Classes : valeurs de P
  - Exemple : if ((Age>30) & (situation=marié)) then prêt=prioritaire
- Règles de classification généralisée
  - Forme : *if* C *then* P, P=p1, p2, …, pm P: attribut but
- etc.

Règles d'association – Résumé

39

- Probablement la contribution la plus significative de la communauté KDD
- Méthodes de recherche de règles :
  - A-priori
  - Algorithmes génétiques
- Plusieurs travaux ont été publiés dans ce domaine

Règles d'association – Résumé

40

Plusieurs issues ont été explorées : intérêt d'une règle, optimisation des algorithmes, parallélisme et distribution, …

Directions de recherche :

- Règles d'associations pour d'autres types de données : données spatiales, multimedia, séries temporelles, …

Critères pour les règles

41

Mesure	Formule	Effet
Support S	$\frac{C \text{ et } P}{N}$	% transactions qui contiennent C et P
Confiance C	$\frac{C \text{ et } P}{C}$	Probabilité conditionnelle
Intérêt I	$\frac{C \text{ et } P}{C \times P}$	Privilégie les motifs rares (ayant un support faible)
Conviction V	$\frac{C \times \overline{P}}{C \text{ et } \overline{P}}$	Mesure la faiblesse de (C, not P) V >> :: P se passe avec C
Piatetsky-Shapiro's	$C \text{ et } P - C \times P$	Mesure la dépendance
Surprise R	$\frac{(C \text{ et } P - C \text{ et } \overline{P})}{P}$	Cherche des règles étonnantes Mesure l'infirmation(C, NOT P)