Règles d’association

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.

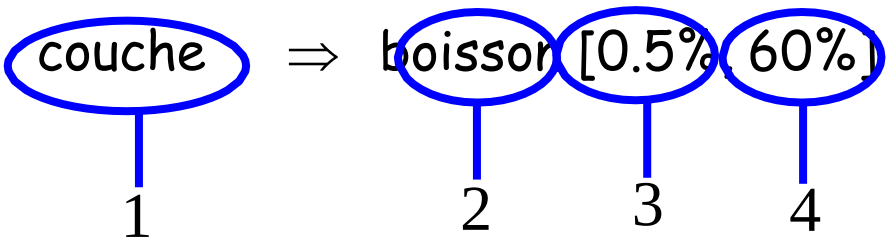
Règles d'associations

- **Formats de représentation des règles d'association :**
  - couches $\Rightarrow$ boisson [0.5%, 60%]
  - achète:boisson $\Rightarrow$ achète:boisson [0.5%, 60%]

  - "**SI** achète couches **ALORS** achète boisson dans 60% de cas. Les couches et la boisson 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") $\Rightarrow$ achète(x, "boisson") [0.5%, 60%]

Règles d'associations



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

1. **Condition**, partie gauche de la règle
2. **Conséquence**, partie droite de la règle
3. **Support**, fréquence ("partie gauche **et** droite sont présentes ensemble dans la base")
4. **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.

$$\textbf{support}(A \Rightarrow B\ [\ s, c\ ]) = p(A \cup B) = \textbf{support}(\{A,B\})$$

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

$$\textbf{confiance}(A \Rightarrow B\ [\ s, c\ ]) = p(B|A) = p(A \cup B) / p(A) = \textbf{support}(\{A,B\}) / \textbf{support}(\{A\})$$

Règles d'associations

- **Support minimum** σ :
  - **Elevé** $\Rightarrow$ **peu** d'itemsets fréquents

    $\Rightarrow$ **peu** de règles valides qui ont été **souvent** vérifiées
  - **Réduit** $\Rightarrow$ **plusieurs** règles valides qui ont été **rarement** vérifiées
- **Confiance minimum** γ :
  - **Elevée** $\Rightarrow$ **peu** de règles, mais toutes "**pratiquement**" **correctes**
  - **Réduite** $\Rightarrow$ **plusieurs** règles, plusieurs d'entre elles sont "**incertaines**"
- **Valeurs utilisées** : σ = 2 - 10 %, γ = 70 - 90 %

Recherche de règles

- 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

- 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

- 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.

Génération des règles à partir des itemsets

- **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

- **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.

Apriori – Réduction de la complexité

- **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

- 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

- 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

- Règles d'association binaires
  - Forme : *if C then* P. C,P : ensembles d'objets
- Règles d'association quantitatives
  - Forme : *if C then* P
    - C = terme$_1$ & terme$_2$ & … & terme$_n$
    - P = terme$_{n+1}$
    - terme$_i$ = <attribut$_j$, *op*, valeur> ou <attribut$_j$, *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.