Les algorithmes de génération des règles d'association

Hana Romdhane

Hajer Trabelsi

2014

Plan

Introduction(1)

- Nous présentons une approche assez récente de fouille de donnée qui est fondé sur la découverte de règles d'association à partir d'un ensemble de données qu'on appellera transaction (Agrawal et al. 1993).
- Ce thème est considéré aujourd'hui comme faisant parti des approches d'apprentissage symbolique non supervisé, utilisé dans le domaine de fouille de données (data mining) et d'extraction de connaissances.
- Un exemple d'application assez courant est l'analyse des logs web sur un serveur web afin de découvrir de comportements utilisateur (web usage mining) dans le but d'adapter ou de personnaliser le site ou de découvrir des comportements types sur certains sites (E-commerce par exemple).

Introduction(2)

- Un exemple classique de l'utilité de cette approche est *le panier du ménagère* qui décrit un ensemble d'achats effectué au supermarché ; les règles d'association permet de découvrir de régularités dans l'ensemble de transactions comme par exemple : *Si fromage alors vin rouge*, etc.

- Ces règles permettent par exemple au gérant de proposer des bons de réductions significatifs sur les achats futurs des clients ! !

Description du domaine(1)

- Un domaine d'application donné doit être décrit par une liste limitée d'atomes qu'on appelle *items*. Par exemple, pour l'application du *panier de ménagère* la liste des items correspond à l'ensemble d'articles disponibles dans le supermarché [*vin; fromage; chocolat; etc*].

- *Un ensemble d'items* est une succession d'items exprimée dans un ordre donné et prédéfini.

- *Une transaction* est un ensemble d'items **I** $\{i_1, i_2, i_3, \text{etc}\}$. Un ensemble de transactions **T** $\{t_1, t_2, t_3, t_4, \text{etc}\}$ correspond à un ensemble d'apprentissage qu'on va utiliser dans la suite pour déterminer les règles d'associations.

Par exemple, deux transactions possibles qui décrivent les achats dans un supermarché Sont :

t_1 = *{Vin Fromage Viande}* et t_2 = *{Vin Fromage Chocolat}*

Description du domaine(2)

- Remarquer bien qu'un ordre doit être défini sur l'ensemble d'items, autrement dit, dans toutes les transactions qui contiennent Vin et Fromage, Vin doit figurer avant Fromage.
- **Le volume** de la transaction est le nombre d'items contenu dans la transaction.
- Une notion importante pour un ensemble d'items est **son support** qui fait référence au nombre de transactions observées qui le contiennent.

Description du domaine(3)

- Exemple :

TID	Items
1	*{Vin, Fromage, Chocolat}*
2	*{Vin, Fromage, Viande}*
3	*{Fromage, Chocolat, Viande}*
4	*{Vin, Fromage, Chocolat}*
5	*{Vin, Coca, Chips}*

Panier *de la ménagère*

Le support *{Vin, Fromage, Chocolat} égale à 2*

Règle d’association

- Une règle d’association est une application sous la forme **X ➔ Y** ou X et Y sont des ensembles d’items disjoints.
- La force d’une règle d’association peut être mesurée en utilisant son support et sa confiance

$$Support, s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}$$

$$Confiance, c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

Règle d'association(2)

Exemple:

- Considérons la règle {vin , fromage} ➔ {chocolat}
- **Le support** du l'ensemble {vin , fromage, chocolat} étant égal à 2 et le nombre total du transaction est égal à 5, le support de la règle est égal 2/5 = 0.4
- **La confiance** est obtenue en divisant le support de l'ensemble {vin , fromage, chocolat} par le support de l'ensemble {vin , fromage} et comme il y'a 3 transactions contenant {vin , fromage} la confiance de cette règle

  est 2/3=0.67

Règle d’association(3)

- Le support est important parce qu’une règle qui à un support faible peut être observé seulement par hasard.
- **La confiance mesure la pertinence de l’inférence fait par une règle.**

Règle d'association: Recherche de règle d'association(1)

- Le problème de la recherche de règle d'association peut se formuler comme suit :
- Etant donnée un ensemble de transaction T, trouvé toute les règles d'association ayant un support ≥ minsup et une confiance ≥ minconf où minsup et minconf sont des seuils pour le support et la confiance .
- Il n'est pas envisageable de chercher toute les règles d'association pour ensuite sélectionné celle qui ont un support et une confiance suffisante, les coûts de calcule serait prohibitifs .
- Un premier pas permettant d'améliorer les performances d'un algorithme

  de recherche de règle consiste à découpler les exigences sur le support et la confiance.

Règle d'association: Recherche de règle d'association(2)

- La définition du support montre que le support d'une règle **X ➔ Y ne dépend que de X ∪ Y**

  Exemple:

  {Vin, Fromage} ➔ *{Chocolat}*

  {Vin, Chocolat} ➔ *{Fromage}*

  {Vin} ➔ *{Chocolat , Fromage}*

  ➔ les règles suivantes ont le même support car elles sont toute construites à partir du même ensemble *{Vin, Fromage, Chocolat}* .

Règle d'association: Recherche de règle d'association(3)

- Une stratégie adoptée par la plupart des algorithmes de recherche de règle d'association consiste à décomposer le problème en deux étapes:
- **Génération des ensembles d'items fréquents**
- **Génération des règles**

Règle d'association: Génération des ensembles d'items fréquents

➔ *L'objectif est de trouver tous les ensembles d'items qui satisfont le seuil minsup.*

Algorithmes d'extraction des items fréquents

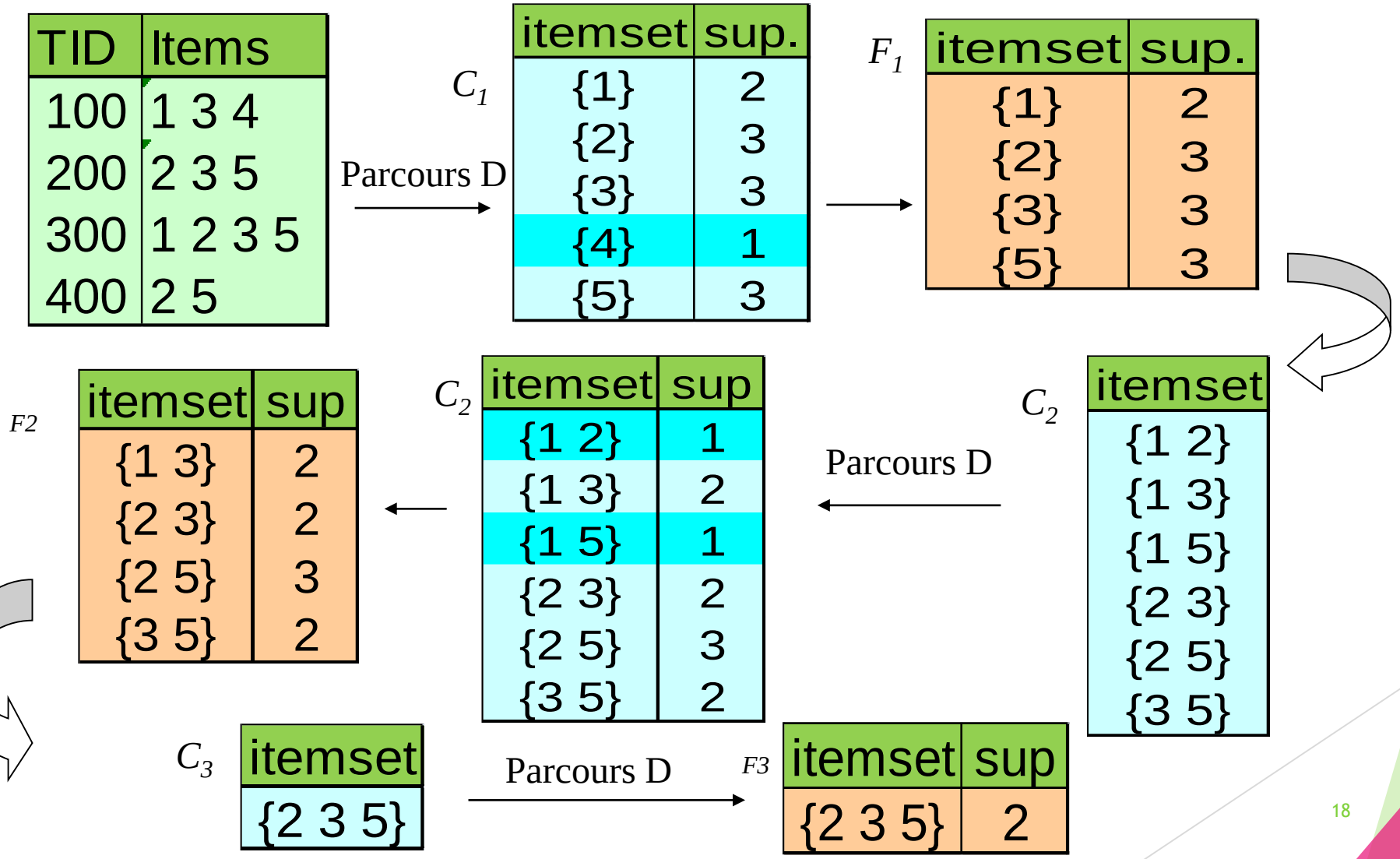
- **APRIORI** (Agrawal & Srikant, 1994)
- **Close** (Pasquier et al, 1998)
- **OCD** (Mannila & al, 1994) qui réalisent un nombre de balayages du contexte égal à la taille des plus longs itemsets fréquents
- **Partition** (Savasere, 1995) qui autorise la parallélisation du processus d'extraction
- **DIC** - Dynamic Itemset Counting (Brin, 1997) qui réduit le nombre de balayages du contexte en considérant les itemsets de plusieurs tailles différentes lors de chaque itération

1- Algorithme APRIORI

- Principe de l'algorithme A Priori:
  - Génération d'ensembles d'items
  - Calcul des fréquences des ensembles d'items
  - On garde les ensembles d'items avec un support minimum: les ensembles d'items fréquents

Entrée : T : corpus, $minsup$: entier
Sortie : $\cup_k F_k$
Début
  $C_1 \leftarrow \{\text{singletons}\}$
  $k \leftarrow 1$
  tantque $C_k \neq \emptyset$ **faire**
    pour chaque $c \in C_k$ **faire**
      pour chaque $t \in T$ **faire**
        si $c \subset t$ **alors**
          $support(c) \leftarrow support(c) + 1$
      $F_k \leftarrow \{c \in C_k \mid support(c) \geqslant minsup\}$
      $k \leftarrow k + 1$
      $C_k \leftarrow \text{APRIORI-GEN}(F_{k-1})$
  Retourner $\cup_k F_k$
Fin

Exemple avec minsup=2



- Points faibles (algorithme apriori) !
  - Le calcul des supports est coûteux
  - La générations des règles est coûteuse
  - Le parcours des données initiales est récurrent

2- Algorithme Close

- repose sur l'extraction de générateurs d'ensemble de mots fermés fréquents
- La fermeture d'un ensemble de mots A est un ensemble de mots B tel que B apparait dans les mêmes textes que A.
- Pour la calculer on utilise deux fonctions :

  f : associe à un ensemble de mots les textes où il apparait

  g : associe à un ensemble de textes les mots qu'ils ont en commun

Exemple:

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

- f({D}) = {1,5}
- g({1,5}) = {B,D}
- fermeture({D}) = {B,D}
- {D} est un générateur de {B,D}

- Principe de l’algorithme Close:
  - Initialisation de l'ensemble des générateurs avec l'ensemble des singletons formés par les mots du corpus
  - Calcul de la fermeture des générateurs de niveau k et de leur support
  - Ajout des fermetures des générateurs à l'ensemble des ensembles de mots fermes fréquents
  - Génération des générateurs de niveau k + 1

Entrée : T : corpus, $minsup$: entier
Début
$G_1 \leftarrow$ ensemble de mots de cardinal 1
$k \leftarrow 1$
tantque $G_k \neq \emptyset$ **faire**
$C_k \leftarrow \text{FERMETURE}(G_k, T)$
pour chaque $c \in C_k$ **faire**
si $support(c) \geq minsup$ **alors**
$F_k \leftarrow F_k \cup \{c\}$
$ferm_k \leftarrow ferm_k \cup ferm(\{c\})$
$G_{k+1} \leftarrow \text{CLOSE-GEN}(F, k)$
Retourner $\cup_k ferm_k$
Fin

Exemple :

Exemple avec $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	$\emptyset$	0
{B}	$\emptyset$	0
{C}	$\emptyset$	0
{D}	$\emptyset$	0

Exemple :

Exemple avec $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	{A, B, D}	1
{B}	{A, B, D}	1
{C}	∅	0
{D}	{A, B, D}	1

Exemple :

Exemple avec $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	{A, B, D}	1
{B}	{B}	2
{C}	∅	0
{D}	{A, B, D}	1

Exemple :

Exemple avec $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	{A, B, D}	1
{B}	{B}	2
{C}	{C}	1
{D}	{A, B, D}	1

Exemple :

Exemple avec $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	{A, B, D}	1
{B}	{B}	3
{C}	{C}	2
{D}	{A, B, D}	1

Exemple :

Exemple avec $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	{A, B, D}	1
{B}	{B}	4
{C}	{C}	2
{D}	{B, D}	2

Exemple :

Exemple avec $minsup = 2/5$

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{A}	{A, B, D}	1/5
{B}	{B}	4/5
{C}	{C}	2/5
{D}	{B, D}	2/5

- On ajoute {B}, {C} et {B, D} à l'ensemble de mots fréquents
- On conserve {B}, {C} et {D} pour calculer les générateurs de niveau supérieur

Exemple :

- À partir de {{B}, {C} et {D} }, on génère les ensembles {BC}; {BD}; {CD}

		textes				
		1	2	3	4	5
mots	A	X				
	B	X	X		X	X
	C			X	X	
	D	X				X

générateur	fermeture	support
{BC}	{BC}	1/5
{CD}	∅	0/5

- Pas de nouvel ensemble de mots fréquents

Règle d'association: Génération des règles

- *L'objectif est d'extraire toutes les règles de grande confiance à partir des ensembles d'items fréquents trouvés dans l'étape précédente. Ces règles sont appelées règles fortes.*

Algorithmes d'extraction des règles

- **GEN-REGLES** (Agrawal & Al, 1994)
- **OPUS** (Webb, G.I. (1995))
- **GEN_RULES, Eclat, GUHA,*Tertius*…**

Algorithme GEN-REGLES

GEN-RÈGLES(E, $minsup$, $minconf$)

```
▷ Entrée : E : ensemble d'ensembles de mots, minsup, minconf : entier
1  Début
2    pour chaque e ∈ E, card(e) ⩾ 2 faire
3      m ← 1
4      H ← {singletons sous-ensemble de e}
5      tantque m ⩽ card(e) faire
6        pour chaque h ∈ H faire
7          confiance(r) ← support(e)/support(e − h)
8          si confiance(r) ⩾ minconf alors
9            R ← R ∪ {(e − h) → h}
10         sinon H ← H \ {h}
11       H ← APRIORI-GEN(H)
12       m ← m + 1
13   Retourner R
14 Fin
```

Exemple avec minconf=1/2

l_k

Itemset	Support
{BCE}	4/6

Génération des règles $\longrightarrow$

1-itemset conséquence

Règle	Confiance
BC $\rightarrow$ E	4/4
BE $\rightarrow$ C	4/5
CE $\rightarrow$ B	4/4

Génération des règles $\longrightarrow$

2-itemset conséquence

Règle	Confiance
B $\rightarrow$ CE	4/5
C $\rightarrow$ BE	4/5
E $\rightarrow$ BC	4/5

l_k

Itemset	Support
{AC}	3/6

Génération des règles $\longrightarrow$

1-itemset conséquence

Règle	Confiance
A $\rightarrow$ C	3/3
C $\rightarrow$ A	3/5

l_k

Itemset	Support
{BE}	5/6

Génération des règles $\longrightarrow$

1-itemset conséquence

Règle	Confiance
B $\rightarrow$ E	5/5
E $\rightarrow$ B	5/5

Conclusion et perspective

- Cette approche est très importante dans plusieurs domaines tel que le domaine médical, commercial,…
- Plusieurs algorithmes sont également utilisé pour l’extraction d items fréquents la base de la génération des règles d'association et la réduction transitive de la base
- Les perspectives de travaux ultérieurs concernent l'étude des diverses techniques d'implémentation et structures de données afin d'améliorer les processus d'extraction de connaissances dans les bases de données selon leurs propriétés et les différents types de données.

Références

- [AS94] : R. Agrawal, R. Srikant. *Fast algorithms for mining association rules in large databases*. Proc. VLDB conf., pp 478–499, September 1994.
- [BMUT97] : S. Brin, R. Motwani, J. D. Ullman, S. Tsur. *Dynamic itemset counting and implication rules for market basket data*. Proc. SIGMOD conf., pp 255–264, May 1997.
- [MTV94] : H. Mannila, H. Toivonen, A. I. Verkamo. *Efficient algorithms for discovering association rules*. AAAI KDD workshop, pp 181–192, July 1994.
- [SON95] : A. Savasere, E. Omiecinski, S. Navathe. *An efficient algorithm for mining association rules in large databases*. Proc. VLDB conf., pp 432–444, September 1995.
- Data Mining. Algorithmes d'extraction et de reduction des regles d'association dans les bases de donnees (PhDThesis Pasquier 2000)
- Extraction de regles d'association - Thierry Lecroq (Univ. Rouen)
- GÉNÉRATION DES RÈGLES D'ASSOCIATION: TREILLIS DE CONCEPTS DENSES (ALAlN BOULANGER)

Questions ?