

Master ISI 2010-2011



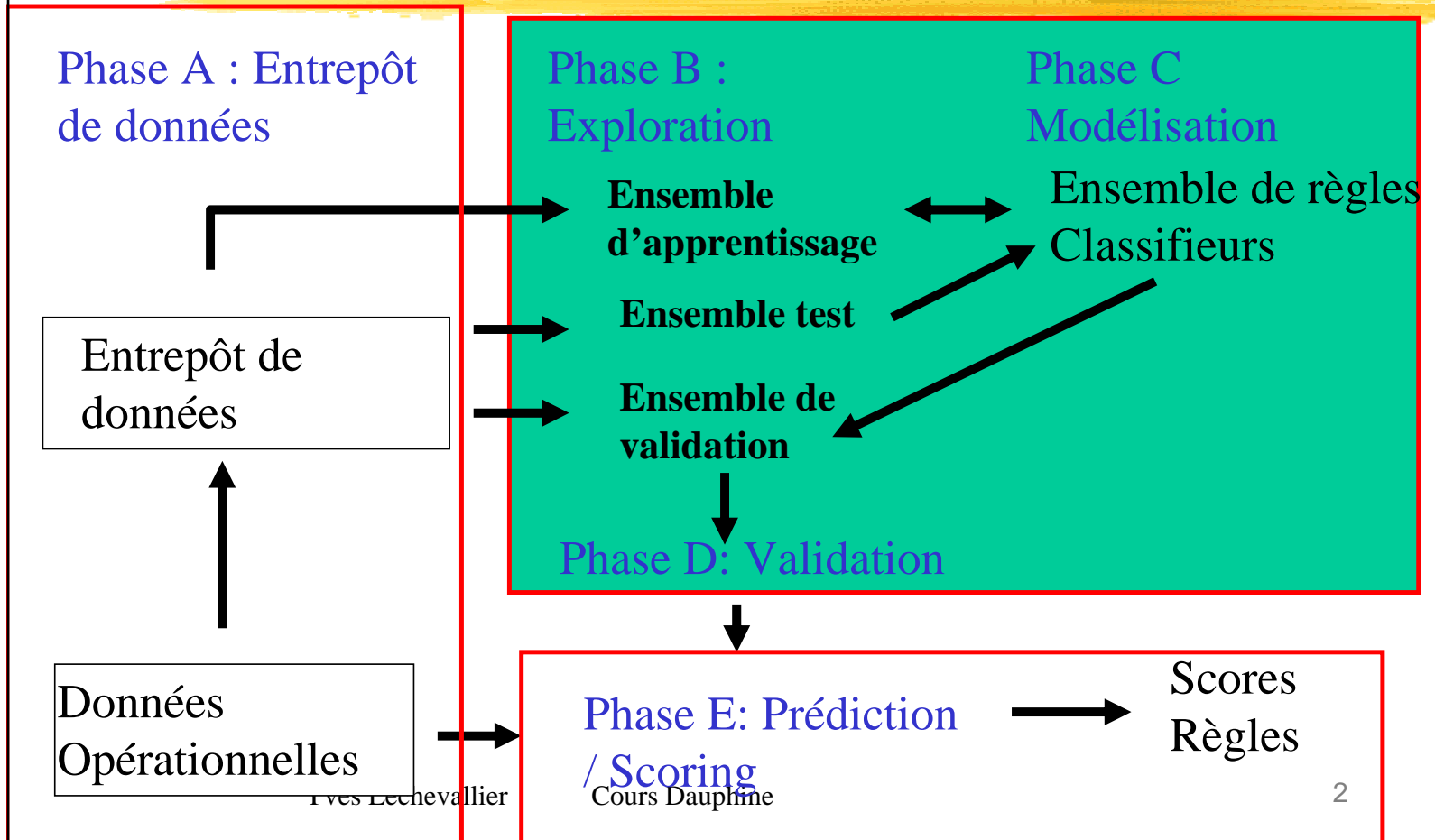
Data Mining Recherche des sous-ensembles fréquents

Yves Lechevallier


INRIA-Rocquencourt

E_mail : Yves.Lechevallier@inria.fr

Processus Data Mining




La recherche d'un modèle (1)



- Analyse logique
 - Règles d'association, ensembles flous
 - Arbre de décision, d'induction
- Analyse décisionnelle ou supervisée
 - Analyse discriminante linéaire ou non
 - Régression, régression logistique
 - Réseaux de neurones
 - Scoring

La recherche d'un modèle (2)



- Analyse de typologie, classification
 - Classification hiérarchique
 - Méthodes de partitionnement
 - Méthodes neuronales
- Techniques de projection
 - Méthodes factorielles

Les associations



- La recherche d'associations vise à construire un modèle sur des règles conditionnelles (de type $A \rightarrow B$) à partir d'un fichier de données
- La recherche d'associations vise à retrouver la liaison qui existe entre deux ou n descripteurs d'un tableau de données

Recherche de règles d'associations



Cette méthode a été introduite en 1993 par R. Agrawal, T. Imielinski et A. Swami du centre de recherche d'IBM.

Le but est d'extraire des règles du type :

« Si un client achète une poupée et du pain alors il achète aussi en même temps un paquet de bonbons »

Les données

TICKET 1
Farine
Sucre
Lait

TICKET 3
Farine
Oeuf
Sucre
Chocolat

TICKET 2
Oeuf
Sucre
Chocolat

TICKET 4
Oeuf
Chocolat
Thé

Un ensemble T dont les m éléments sont appelés **transactions**

Principe de construction

Contenu d'un ticket de caisse

TICKET 1
Farine
Sucre
Lait

Création des associations

Farine -> Sucre	Sucre->Farine
Sucre -> Lait	Lait -> Sucre
Lait -> Farine	Farine -> Lait

Un ensemble I dont les n éléments sont appelés **items**

$I = \{\text{Farine, Sucre, Lait, Œuf, Chocolat, Thé}\}$

Structure des règles produites(1)



On obtient un ensemble de règles de la forme:

$\{\text{Lait}, \text{Œuf}\} \rightarrow \{\text{Chocolat}\}$

support = 10%, confiance = 25%

Cela signifie que 10% des transactions contiennent à la fois les items *Lait*, *Œuf* et *Chocolat* et que 25% des transactions contenant *Lait*, *Œuf* contiennent aussi l'item *Chocolat*

Structure des règles produites(2)

- Une **règle d'association** a la forme :

$X \rightarrow Y$, où $X \subseteq I$ et $Y \subseteq I$ avec $X \cap Y = \emptyset$

la *prémisse* est X et la *conclusion* est Y

Le *support* est

$$\text{sup}(X \rightarrow Y) = \frac{\text{card}\{t \in T / X \cup Y \subseteq t\}}{\text{card}(T)}$$

La *confiance* est

$$\text{conf}(X \rightarrow Y) = \frac{\text{card}\{t \in T / X \cup Y \subseteq t\}}{\text{card}\{t \in T / X \subseteq t\}}$$

Recherche des règles *intéressantes* (1)

- **Critère d'extraction des règles**

A partir d'un ensemble T de transactions, trouver toutes les règles avec un support $sup > s_0$ et une confiance $conf > c_0$ où s_0 et c_0 sont des seuils fixés a priori par l'utilisateur.

Si *Chocolat* est contenu dans beaucoup de tickets cela devient moins intéressant.

Recherche des règles *intéressantes* (2)

- **Critère d'évaluation des règles**

Le *lift* :

$$\mathit{lift}(X \rightarrow Y) = \mathit{conf}(X \rightarrow Y) \frac{\mathit{card}\{t \in T\}}{\mathit{card}\{t \in T / Y \subseteq t\}}$$

En terme de probabilité on a :

$$\mathit{lift}(X \rightarrow Y) = \frac{\Pr(X \cap Y)}{\Pr(X) \cdot \Pr(Y)}$$

Algorithme de base **APriori**

C'est l'algorithme proposé par Agrawal et Srikant en 1994. La complexité est d'ordre $O(m)$.

- ❖ Recherche des sous-ensembles de I présentant un support $sup(X)$ supérieur à s_0 .
- ❖ Construction des règles dont la confiance est supérieure à c_0 .

$$sup(X) = card\{t \in T / X \subseteq t\} / card(T)$$

Recherche des sous-ensembles fréquents

$m = \text{card}(T)$. C_1 est l'ensemble des **items**

Un **motif** est un sous-ensemble d'items.

Le principe de l'algorithme est de rechercher l'ensemble L_1 de tous les items apparaissant dans au moins $s_0 \times m$ transactions. Puis, parmi C_2 qui est le produit cartésien de L_1 avec lui-même, on construit l'ensemble L_2 de tous les couples d'items apparaissant dans au moins $s_0 \times m$ transactions.

L'algorithme s'arrête quand L_k est vide.

Construction de C_1

Item1	Nombre
Farine	2
Sucre	3
Lait	1
Œuf	3
Chocolat	3
Thé	1

On décide de retenir un taux de support de 30%

Construction de C_2

Item1	Item2	Nombre
Farine	Sucre	2
Farine	Œuf	1
Farine	Chocolat	1
Sucre	Œuf	2
Sucre	Chocolat	2
Œuf	Chocolat	3

L_2 contient 4 couples Farine-Sucre, Sucre-Œuf, Sucre-Chocolat et Oeuf-Chocolat.

Construction de C_3

Item1	Item2	Item3	Nombre
Farine	Sucre	Oeuf	1
Farine	Sucre	Chocolat	1
Sucre	Oeuf	Chocolat	2

Insert into C_3 Select p.item1, p.item2, q.item2

From L_2 p, L_2 , q

where p.item1=q.item1 **and** p.item2 < q.item2

Ensemble des sous-ensembles fréquents

L'ensemble L des sous-ensembles fréquents est l'union des ensembles L_1, \dots, L_K .

$$L_1 = \{\text{Farine, Sucre, Œuf, Chocolat}\}$$

$$L_2 = \{(\text{Farine, Sucre}), (\text{Sucre, Œuf}), (\text{Sucre, Chocolat}), (\text{Œuf, Chocolat})\}$$

$$L_3 = \{(\text{Sucre, Œuf, Chocolat})\}$$

Construction des règles

Pour chaque ensemble fréquent on construit des règles vérifiant la contrainte de seuil de confiance.

Un algorithme simple pour produire des règles à partir d'un sous-ensemble fréquent f est de considérer tous les sous-ensembles possibles g de f et de produire la règle $g \rightarrow (f-g)$ si la condition sur la confiance est vérifiée.

Cependant si une règle $(f-g) \rightarrow g$ vérifie la contrainte de confiance, alors, pour chaque partie h de g , la règle $(f-h) \rightarrow h$ vérifie aussi la condition de confiance.

Alors on commence par les règles ayant un seul conséquent, puis sur les règles retenues on génère les règles ayant deux conséquents.

Tableau des règles

Sous-ensemble	Règle	Support	Confiance
{ Farine, Sucre }	Farine->Sucre	2/4	2/2
	Sucre->Farine	2/4	2/3
{ Sucre, Œuf }	Sucre->Œuf	2/4	2/3
	Œuf->Sucre	2/4	2/3
{ Sucre, Chocolat }	Sucre->Chocolat	2/4	2/3
	Chocolat->Sucre	2/4	2/3
{ Œuf, Chocolat }	Œuf->Chocolat	3/4	3/3
	Chocolat->Œuf	3/4	3/3
{ Sucre, Œuf, Chocolat }	Sucre->{Œuf, Chocolat}	2/4	2/3
	{Œuf, Chocolat}->Sucre	2/4	2/3
	Œuf->{Sucre, Chocolat}	2/4	2/3
	{Sucre, Chocolat}-> Œuf	2/4	2/2
	Chocolat->{Sucre, Œuf}	2/4	2/3
	{Sucre, Œuf}->Chocolat	2/4	2/3

Tableau des règles *intéressantes*

Liste des règles ayant une confiance égale à 1.

Sous-ensemble	Règle	Confiance	lift
{Farine, Sucre}	Farine->Sucre	2/2	4/3
{Œuf, Chocolat}	Œuf->Chocolat	3/3	4/3
	Chocolat->Œuf	3/3	4/3
{Sucre, Œuf, Chocolat}	{Sucre, Chocolat}-> Œuf	2/2	4/3

$$lift(X \rightarrow Y) = conf(X \rightarrow Y) \frac{card\{t \in T\}}{card\{t \in T / Y \subseteq t\}}$$

Extensions de la méthode



- **Introduction de taxinomies**

Donner une structure hiérarchique sur l'item,
par exemple Sucre, Sucre_Marque,
Sucre_Marque_Poids.

- **Amélioration des performances** des algorithmes en utilisant le langage SQL.

Remarque

Le nombre des combinaisons des items croît très rapidement

Nombre	Combinaisons
1	100
2	4950
3	161 700
4	3 921 225
5	75 287 520
6	1 192 052 400
7	16 007 560 800
8	186 087 894 300

Le nombre d'items est égal à n .

Le nombre de combinaisons de k items est égal à $n!/(n-k)!k!$

Réduction du nombre de sous-ensembles fréquents

Construire l'ensemble des sous-ensembles fréquents fermés F à partir de L .

g est un sous-ensemble fréquent **fermé** si :

$$\forall h \in L \quad g \not\subset h \quad \text{ou} \quad \forall h \notin L \quad h \not\subset g$$

C'est-à-dire qu'il n'existe pas de sous-ensembles fréquents contenant un sous-ensemble fréquent fermé.

$$F = \{ (\text{Farine, Sucre}), (\text{Sucre, Œuf, Chocolat}) \}$$

Treillis de Galois

- Triplet : (O,A,R) tel que :
 - O ensemble des objets (transactions)
 - A ensemble des attributs (items)
 - R relation binaire entre O et A
- Tableau d'incidence ou correspondance entre O et A
 - $H = \{(o,a) / oRa\}$

Exemple

	Farine	Sucre	Lait	Œuf	Chocolat	Thé
Ticket 1	X	X	X			
Ticket 2		X		X	X	
Ticket 3	X	X		X	X	
Ticket 4				X	X	X

$H = \{(T1, Farine), (T1, Sucre), (T1, Lait), (T2, Sucre), \dots, (T4, Thé)\}$

Correspondance de Galois

$$f : \wp(O) \rightarrow \wp(A)$$

$$\forall G \subseteq O \quad f(G) = \{a \in A / oRa, \forall o \in G\} \quad \text{intension}$$

$$f(G) = \{a \in A / (o, a) \in H, \forall o \in G\}$$

$$g : \wp(A) \rightarrow \wp(O)$$

$$\forall B \subseteq A \quad f(B) = \{o \in O / oRa, \forall a \in B\} \quad \text{extension}$$

$$f(B) = \{o \in O / (o, a) \in H, \forall a \in B\}$$

(f, g) est une correspondance de Galois

f et g sont deux fonctions monotones et décroissantes

Concept

(G, B) est un concept si et seulement si G est l'extension de B et B est l'intension de G .

$G = g(B)$ et $B = f(G)$ L ensemble des concepts

Relation d'ordre sur L

$G_1, G_2 \in O, B_1, B_2 \in A, (G_1, B_1) \leq (G_2, B_2) \Leftrightarrow G_1 \subseteq G_2$ et $B_1 \supseteq B_2$

Treillis de Galois

$T = (L, \leq)$ Ensemble des concepts muni d'une relation d'ordre

Concept/sous-ensemble fréquent fermé

- Un **concept** est un couple $(B, g(B))$ où B est un sous-ensemble fermé et $g(B)$ est l'ensemble des objets (transactions) contenant tous les items de B .
- Un **sous-ensemble fréquent fermé** est un concept ayant un support supérieur à s_0 .

Recherche de séquences fréquentes



- On est capable de conserver la trace du passage d'un même client à différents instants
- En plus de la recherche de règles d'associations, il est possible de rechercher des séquences d'achats fréquentes.
- Chaque élément d'une séquence fréquente peut être composée de plusieurs items.

Structure des données



- I un ensemble d'items
- C un ensemble de clients
- D un ensemble ordonné de dates
- T un ensemble de transactions

Chaque transaction est définie par

- Un ensemble d'items
- Identifiant du client
- La date de la transaction

Les données

TICKET 1
Farine Sucre Lait

TICKET 2
Oeuf Sucre Chocolat

TICKET 1
Farine Oeuf Sucre Chocolat
TICKET 2
Bière Pain
TICKET 3
Oeuf Chocolat Thé

TICKET 1
Viande Salade Sucre

TICKET 2
Chocolat Farine

Tableau des séquences

Client	Ticket	Items
1	1	Farine, Sucre, Lait
1	2	Œuf, Sucre, Chocolat
2	1	Farine, Œuf, Sucre, Chocolat
2	2	Bière, Pain
2	3	Œuf, Chocolat, Thé
3	1	Viande, Salade, Sucre
3	2	Chocolat, Farine

Séquences

- Une *séquence* est une liste ordonnée de sous-ensembles d'items.
- Une *relation d'ordre partielle* sur l'ensemble des séquences

$a=(a_1,\dots,a_p)$ et $b=(b_1,\dots,b_p)$ deux séquences

a est contenu dans b s'il existe des entiers

$i_1<\dots<i_p$ tel que $a_1 \subseteq b_{i_1}, \dots, a_p \subseteq b_{i_p}$

Séquence de client



- L'ensemble des transactions d'un client est une séquence particulière, appelée *séquence de client*.
- Le support d'une séquence est :
 $sup(s) = (\text{nombre de séquences de clients contenant } s) / (\text{nombre total de clients})$
- Séquence a de client supporte b si b est inclus dans a .

Critère d'extraction des séquences



A partir d'un ensemble T de transactions, trouver l'ensemble des séquences présentant un support supérieur à s , paramètre de la méthode.

Les séquences trouvées sont appelées *séquences fréquentes*.

Algorithme de base



- Rechercher les séquences de longueur 1 ayant un support supérieur à s . C'est l'ensemble des sous ensembles fréquents.
- A partir des séquences trouvées dans l'étape précédente, construire les séquences de longueur 2 avec un support supérieur à s .
- Par itération, construire des séquences de longueur k avec un support supérieur à s à partir de celles trouvées pour une longueur $k-1$.

Sous-ensembles fréquents



Farine(supp : 3)

Sucre(supp : 4)

Oeuf(supp : 3)

Chocolat(supp : 4)

Farine Sucre (supp: 2)

Farine Chocolat (supp: 2) Sucre Oeuf (supp: 2)

Sucre Chocolat (supp: 2)

Oeuf Chocolat (supp: 3)

Sucre Oeuf Chocolat (supp: 2)

Extension de la méthode



- Introduction de taxinomies
- Introduction de contraintes temporelles
 - Regroupement de transactions par fenêtre glissante
 - Ajout de contraintes temporelles
 - Comment vérifier ces contraintes temporelles

Bibliographie



- G Hébrail « La recherche de règles d'associations et de séquences fréquentes », Ecole Modulad 17-19 novembre 1999;
- Agrawal R, Imielinski T., Swami A. (1993) « Mining Associations between Sets of Items in Massive Databases » ACM SIGMOD'93
- Srikant R., Agrawal R. (1996) « Mining Generalized Association Rules » VLDB'95 Zurich.
- Ganter B., Wille R., (1999) *Formal concept analysis/ mathematical foundations*, Springer
- G. Hébrail, Y. Lechevallier. (2003) « Data mining et analyse des données » , in : *Analyse des données*, Hermes
- H. Cherfi, A. Napoli, Y. Toussaint (2006) « Deux méthodologies de classification de règles d'association pour la fouille de textes » , in: *Revue des Nouvelles Technologies de l'Information*.
- A. Marascu , F. Masegla . (2006) « Extraction de motifs séquentiels dans les flots de données d'usage du Web », in: *Extraction et Gestion des Connaissances (EGC), Lille*, p. 627-638.