Extraction de connaissances
Exemple fil rouge
Aide à la Décision et Intelligence Artificielle
Licence 3

Bruno Crémilleux - Abdelkader Ouali

Département Informatique
UFR des sciences
Université de Caen Normandie

2020

Contexte

Problématiques : contraintes et résolution de contraintes, planification, découverte de connaissances à partir d'exemples

Fil conducteur : maisons respectant des motifs spécifiques (p.ex. le fait que les maisons qui ont une salle de bains en (2,2) ont souvent une cuisine en (2,3))

1. **data-mining :** liste des maisons ayant des plans similaires (i.e. transactions en data-mining) : on apprend/découvre des règles [1])
2. **satisfaction de contraintes :** énumérer toutes les solutions possibles respectant un ensemble de contraintes
3. **planification :** simulation d'une prise en charge de construction

1. on dispose pour ce TP fil rouge d'une vérité terrain (les règles correspondent aux contraintes utilisées pour générer les maisons) ce qui est un cas particulier

L'exemple "fil rouge"

Variables

- pour chaque position (i,j) dans la grille, $pièce_{i,j}$: catégorielles
- dalle coulée : binaire
- sol humide : binaire
- murs élevés : binaire
- toiture terminée : binaire

Attention : chaque variable $pièce_{i,j}$ est multivaluée :
par exemple : valeur de $pièce_{i,j} \in \{$salle de bains, cuisine, séjour, chambre1, chambre2, toilettes$\}$

L'exemple "fil rouge"

Propositionnalisation : Nécessaire pour passer à des variables booléennes pour la fouille

- **variable binaire :** item [2] présent si la variable a la valeur "vrai", pas d'item sinon, cela évite de générer des règles "triviales" comme $\neg toiture \rightarrow \neg Murs$

- **variable multivaluée (un item par valeur) :**
  $pièce_{2,2}$ ="salle de bain", $pièce_{2,2}$ ="toilettes", $pièce_{2,2}$ ="chambre2", etc.
  (pour une transaction, un seul des 3 items est présent)

En résumé :

- Variable avec 2 valeurs : création d'1 seul item (i.e. vrai)
- Sinon (i.e. variable a n valeurs avec $n > 2$) : création de n items

2. item : une variable booléenne

L'exemple "fil rouge"

Contraintes (ou règles) :

Deux pièces d'eau (salles de bains, cuisines, toilettes) doivent nécessairement être envisagées côte à côte

- $(pièce_{1,1} = \text{salles de bains}) \rightarrow (pièce_{2,2} = (cuisines))$
- $(pièce_{1,1} = \text{toilettes}) \rightarrow \neg(pièce_{1,3} = \text{cuisines})$
- $(pièce_{1,1} \in \{\text{salles de bains, cuisines, toilettes}\}) \rightarrow \neg(pièce_{1,2} \in \{\text{chambre1, chambre2, séjour}\})$

Une toiture n'est présente que si les murs sont présents

- $toiture \rightarrow Murs$

Motif

Chaque pièce envisagée occupe au plus une position :

- une seule occurrence d'une pièce dans chaque ligne de la base de données
- une seule occurrence d'une pièce dans le motif

Remarque : compte-tenu de notre choix de propositionnalisation, on ne retrouvera pas des règles avec des négations

Problématique : apprendre les règles caractérisant des maisons construites ou en construction

Démarche :

- soit une base de transactions (i.e., les données)
- utiliser un algorithme de fouille de données (p.ex. extraction de règles d'association) pour apprendre les règles[1] caractérisant les maisons.
- comme très souvent dans la vraie vie, les données sont incertaines (ici, un peu bruitées...)

  ➥ recherche de régularités avec des exceptions (i.e. non nécessairement de confiance 100%)

1 : pour ce TP fil rouge, on dispose d'une vérité terrain (les règles correspondent aux contraintes utilisées pour générer les solutions) ce qui est un cas particulier.

Votre mission (dans un premier temps) :

- coder un algorithme (plus ou moins) naïf d'extraction de règles d'association
- **des "plus" :** implémentation de critères d'élagage dans l'extraction des motifs [3] et la génération des règles,...
- **des "extra" :** étude expérimentale des temps d'exécution de votre/vos algorithmes en fonction :
  - des caractéristiques des bases de données fouillées
  - des valeurs des paramètres utilisées dans l'extraction de motifs
  - et la génération de règles
  - des éventuels critères d'élagage mise en œuvre

3. les motifs sont les éléments qui composent les prémisses et conclusions des règles

Remarques :

- **Mauvaise nouvelle :** en 2020, en data mining, le défi n'est plus d'extraire des règles...

Remarques :

- **Mauvaise nouvelle :** en 2020, en data mining, le défi n'est plus d'extraire des règles...

- **Bonne nouvelle :** vous verrez en master quelques défis actuels :

  - extraire des motifs/règles assurant une **diversité** de la connaissance

  - produire des ensembles de motifs/règles dont la qualité d'un motif/règle **dépend** des autres motifs/règles de l'ensemble

  - extraire des motifs/règles selon des **préférences utilisateurs** apprises au cours du processus de fouille

  - Etc.

TP 1 : un exemple de BD jouet

	A	B	C	D
t1	×	×		×
t2		×	×	×
t3	×			×
t4	×			

Implémentation de la classe **BooleanDatabase**

Règles d'association dans le contexte de la science des données

Règles d'association : un exemple

Ensemble de données

Id.	Items
1	chaussures, chemise, veste
2	chaussures, veste
3	chaussures, jeans
4	chemise, sweat-shirt

Itemsets fréquents

Itemsets valides	Fréquence
{chaussures}	75%
{chemise}	50%
{veste}	50%
{chaussures, veste}	50%

fréquence min. : 50 %

Règles	Fréquence	Confiance
Si chaussures alors veste	50%	66.6%
Si veste alors chaussures	50%	100%

But : extraction de **toutes** les règles ayant une fréquence (i.e. support) et une confiance supérieures aux seuils.

Règles d'association : définition

- $\mathcal{I}$: un ensemble de littéraux : items (e.g., A, B)
- **itemset** (noté X) ou **motif** : un sous-ensemble de $\mathcal{I}$
- Langage de motifs : $\mathcal{L}^2$
- Règle d'association : $X \rightarrow Y$
- Deux mesures usuelles pour quantifier l'intérêt d'une règle :
  - **fréquence :** $\mathcal{F}(X \rightarrow Y) = \mathcal{F}(XY)$
    ➥ "poids", "représentativité" d'une règle
  - **confiance :** $conf(X \rightarrow Y) = \frac{\mathcal{F}(XY)}{\mathcal{F}(X)}$
    ➥ "force" du lien entre prémisse et conclusion : probabilité conditionnelle de Y sachant X
- La fréquence et la confiance ont l'avantage de véhiculer une certaine interprétation des règles.
- Il existe de nombreuses mesures d'intérêt !

Règles d'association : processus de découvert

Deux étapes :

- **Étape 1 : trouver toutes les combinaison des items ayant une fréquence supérieure à la fréquence minimale :**
  ➥ un motif fréquent prend la majorité du temps de calcul : le nombre de motifs candidats (i.e. potentiellement fréquents) doit être minimisé (élagage de l'espace de recherche)

- **Étape 2 : utiliser les motifs fréquents pour générer les règles**

Un exemple (1/2)

TID	Items				
1	A	B	C	D	E
2	A			D	
3	A	B	C	D	
4		B	C		
5	A	B	C		
6					E

- donner l'ensemble des itemsets fréquents et leurs fréquences (avec minfr $= 2$)
- quelle est la taille de l'espace de recherche ? comment est-il structuré ? å tracer le treillis

Un exemple (2/2)

TID	Items				
1	A	B	C	D	E
2	A			D	
3	A	B	C	D	
4		B	C		
5	A	B	C		
6					E

- **avec minfr = 3 :**
  ∅ (6),
  A (4), B (4), C (5),
  AB (3) , AC (4), BC (4),
  ABC (3)
- **avec minfr = 2 :**
  ∅ (6),
  A (4), B (4), C (5), D (2), E (2),
  AB (3) , AC (4), AD (2), BC (4), BD (2),
  CD (2),
  ABC (3), ABD (2), ACD (2), BCD (2),
  ABCD (2)

Algorithme 1 pour extraire les fréquents

Principe : [4]

- parcours en largeur (approche par niveau) pour énumérer tous les candidats et, pour chaque candidat, tester si il est fréquent
- plus précisément : générer tous les candidats de longueur 1, ne conserver que les fréquents, puis générer tous les candidats de longueur 2 à partir des fréquents de longueur 1, ne conserver que les fréquents, etc.

Pour chaque candidat X :

- calculer $\mathcal{F}(X)$
- si $\mathcal{F}(X) \geq minfr$ alors retourner X
  sinon // $\mathcal{F}(X) < minfr$

4. Ce principe est utilisé par l'algorithme APRIORI

Algorithme 1 pour extraire les fréquents
Comment générer les candidats ?

Soit $\mathcal{MF}_k$ l'ensemble des motifs fréquents de longueur k et $n = |\mathcal{MF}_k|$:

- **méthode 1 :** pour chaque paire $\{X, Y\}$ de motifs de $\mathcal{MF}_k$, combiner X et Y pour donner lieu au motif $X \cup Y$.

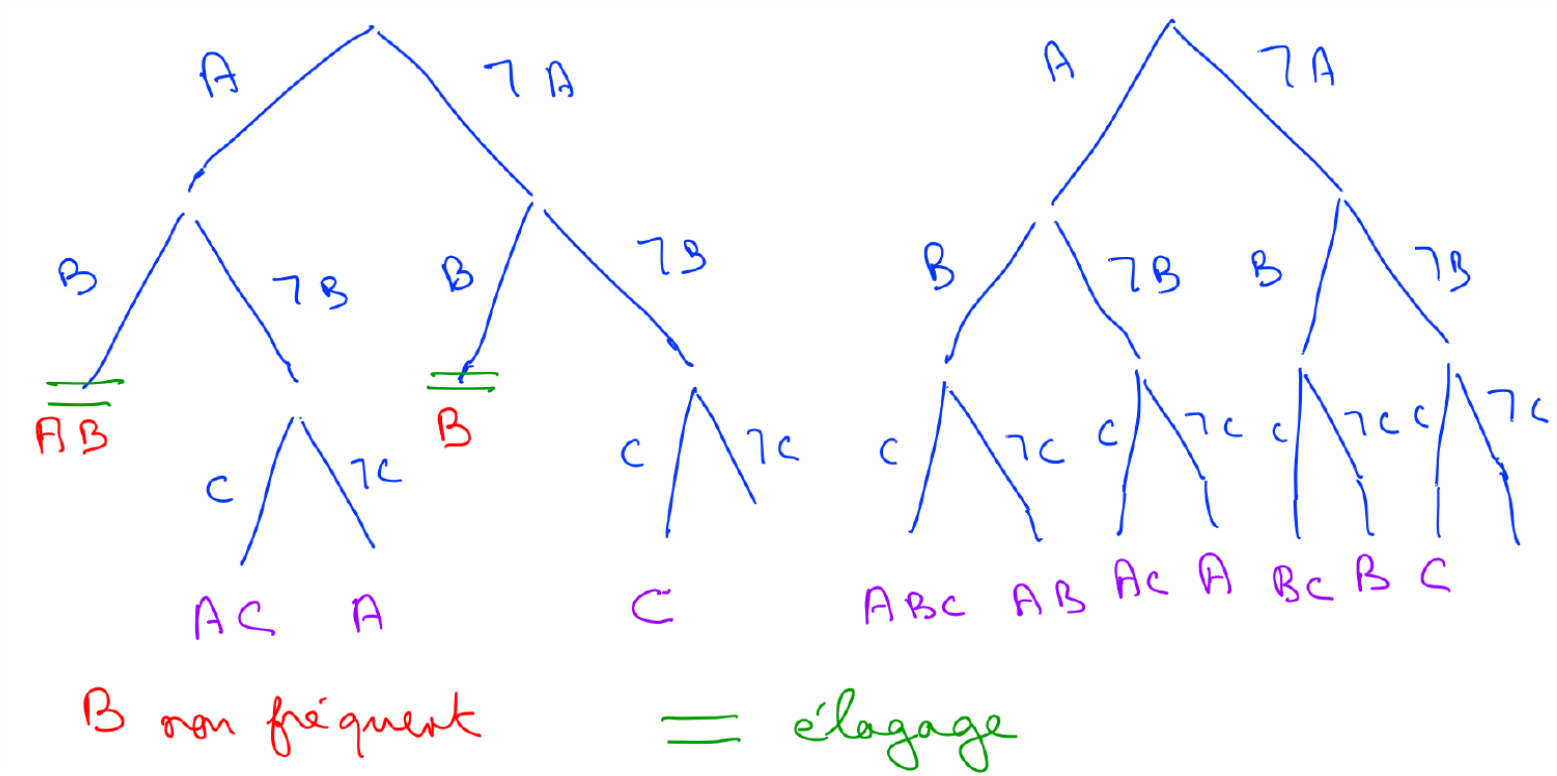
  ➥ Il y a $\frac{n*(n-1)}{2}$ paires car inutile de générer les paires symétriques

- ne conserver comme candidats que les motifs $X \cup Y$ telle que $|X \cup Y| = k + 1$

- **méthode 2 (plus efficace) :** pour chaque paire $\{X, Y\}$ de motifs de $\mathcal{MF}_k$ dont les motifs X et Y partagent leurs $k - 1$ premiers items, combiner X et Y pour donner lieu au motif $X \cup Y$

Algorithme 2 pour extraire les fréquents

principe : parcours en **profondeur**, à chaque nœud, choix d'un item et séparer les transactions suivant qu'elles possèdent ou pas l'item.



critère d'**anti-monotonicité** de la fréquence par rapport à la spécialisation des items

TPs : organisation

- Classes pour maintenir des **bases de données** et des **motifs (itemsets)** afin de créer l'ensemble de règles
- Extraction de **motifs fréquents** en cherchant les motifs singletons fréquents puis leur combinaison avec la méthode 2 du diapo 16
- Extraction de **règles fréquentes** et **précises**
- **Propositionnalisation** sur des bases de données non-booléennes

Partie 1 du TP : précision (1/3)

Classe `BooleanDatabase` : représente une BD transactionnelle où toutes les variables sont booléennes

- une instance de `BooleanDatabase` contient une liste de variables et une liste de transactions (instances)
- une transaction est représentée par un Set<BooleanVariable>

Classe `Database` **:** représente une BD où les variables ne sont pas nécessairement booléennes

- les transactions sont représentées pas des Map<Variable, Object>
- **propositionalisation** pour passer d'une Database à une BooleanDatabase

Partie 1 du TP : précision (2/3)

Classe `Apriori` :

- possède un attribut de type `BooleanDatabase`
- une méthode `extract` prenant en argument un fréquence minimale
- retourne un ensemble de motifs (Set<Itemset>)

Tester sur de petits exemples : cf. exemples “Chaussures, chemise, veste,... ” et celui de la diapo 10.

Partie 1 du TP : précision

En cas de panique pour extraire les motifs fréquents, une solution simple (mais particulièrement naïve) :

- soit $\mathcal{I}$ l'ensemble de tous les items, générer comme candidats tous les sous-ensembles de $\mathcal{I}$ (c'est-à-dire $2^{\mathcal{I}}$).

  ➥ Attention : $2^{\mathcal{I}}$, croissance exponentielle du nombre de candidats en fonction du nombre d'items...

- pour chaque candidat, tester si il est fréquent

Remarque : cette méthode n'exploite pas l'anti-monotonicité de la fréquence par rapport à la spécialisation des items

Partie 2 du TP : précision

la première ligne donne le nom des variables et chaque ligne suivante une transaction
Effectuer une **propositionalisation des données** :

- une classe `Database` pour représenter une BD non nécessairement booléenne
- une méthode `propositionalize` transformant une BD non-booléenne en une BD booléenne en créant (cf. diapo 2) :
  - variable binaire : 1 seul item (i.e. vrai)
  - variable avec au moins 3 valeurs : une variable booléenne par couple (valeur,variable) pour les variables avec au moins 3 valeurs

  et en recodant chaque transaction suivant les variables booléennes.

Partie 2 du TP : précision

Écrire une classe `BruteForceAssociationRuleMiner` possédant :

- ayant une fonction `extract` qui retourne un ensemble de règles d'association Set<AssociationRule> fréquentes par rapport au seuil minimale de fréquence, et précises par rapport au seuil minimale de confidence. Les motifs fréquents sont retrouvés avec la classe `Apriori`
  ➥ (cf. les diapos à partir de 26)

Tester sur les petits exemples vus en cours

Algorithme de génération de règles d'association (sans optimisation)

Générer les règles (1/3)

Entrée :
la collection de motifs fréquents et leurs fréquences ; minfr ; minconf

Principe :
soit Z un motif fréquent, décomposer Z en 2 parties X et Y pour générer une règle $X \rightarrow Y$ telle que $Z = X \cup Y$ et $X \cap Y = \emptyset$

Exemple :
à partir de ABC, on génère :
$A \rightarrow BC$, $B \rightarrow AC$, $C \rightarrow AB, AB \rightarrow C$, $AC \rightarrow B, BC \rightarrow A$

Générer les règles (2/3)

Une règle $X \rightarrow Y$ est-elle fréquente (i.e, $\mathcal{F}(X \rightarrow Y) \geq minfr$) et valide (i.e, $conf(X \rightarrow Y) \geq minconf$) ?

- **fréquence :** oui car $X \rightarrow Y$ est issue de $X \cup Y = Z$ et Z est fréquent
- **confiance :** tester si $conf(X \rightarrow Y) \geq minconf$

  $conf(X \rightarrow Y) = \frac{F(XY)}{F(X)}$ se calcule facilement car :

  - $\mathcal{F}(XY) = \mathcal{F}(Z)$ est connu,
  - $\mathcal{F}(X)$ est connu car $\mathcal{F}(X) \geq \mathcal{F}(XY)$

Générer les règles (3/3)

Pour décomposer un motif Z :

il faut générer l'**ensemble des parties de Z** pour obtenir toutes les prémisses, une conclusion étant les items de Z n'appartenant pas à une prémisse

- parcourir un arbre binaire en sélectionnant ou pas successivement chaque item de Z (cf. l'arbre généré par l'algorithme 2 pour extraire les fréquents) : chaque feuille correspond à une prémisse d'une règle

- générer les itemsets de longueur 1 de Z , puis ceux de longueur 2, puis ceux de longueur 3, etc. [5]

5. Une possibilité est d'utiliser le principe de génération en largeur des candidats de l'algorithme Apriori.

Motifs fermés

Motif fermé : h(X) : le motif fermé de X

- ensemble maximal d'items partagé par un ensemble de transactions
- formellement : h(X) est l'intersection des transactions supportant X
- autrement dit : les fermés sont les maximaux des classes d'équivalence (cf. diapo 15) sur l'exemple de la diapo 8

  ➥ $h(B) = BC,\ h(BD) = ABCD$

  il y a 8 motifs fermés $\{\emptyset, C, E, AC, ABC, BC, ABCD, ABCDE\}$

Méthode **très naïve** pour extraire les motifs fermés :

comparer 2 à 2 tous les motifs fréquents, chaque fois qu'un motif X est inclus dans un motif Y avec la même fréquence, X est éliminé (post-traitement des motifs fréquents)

Motifs fermés et règles d'association

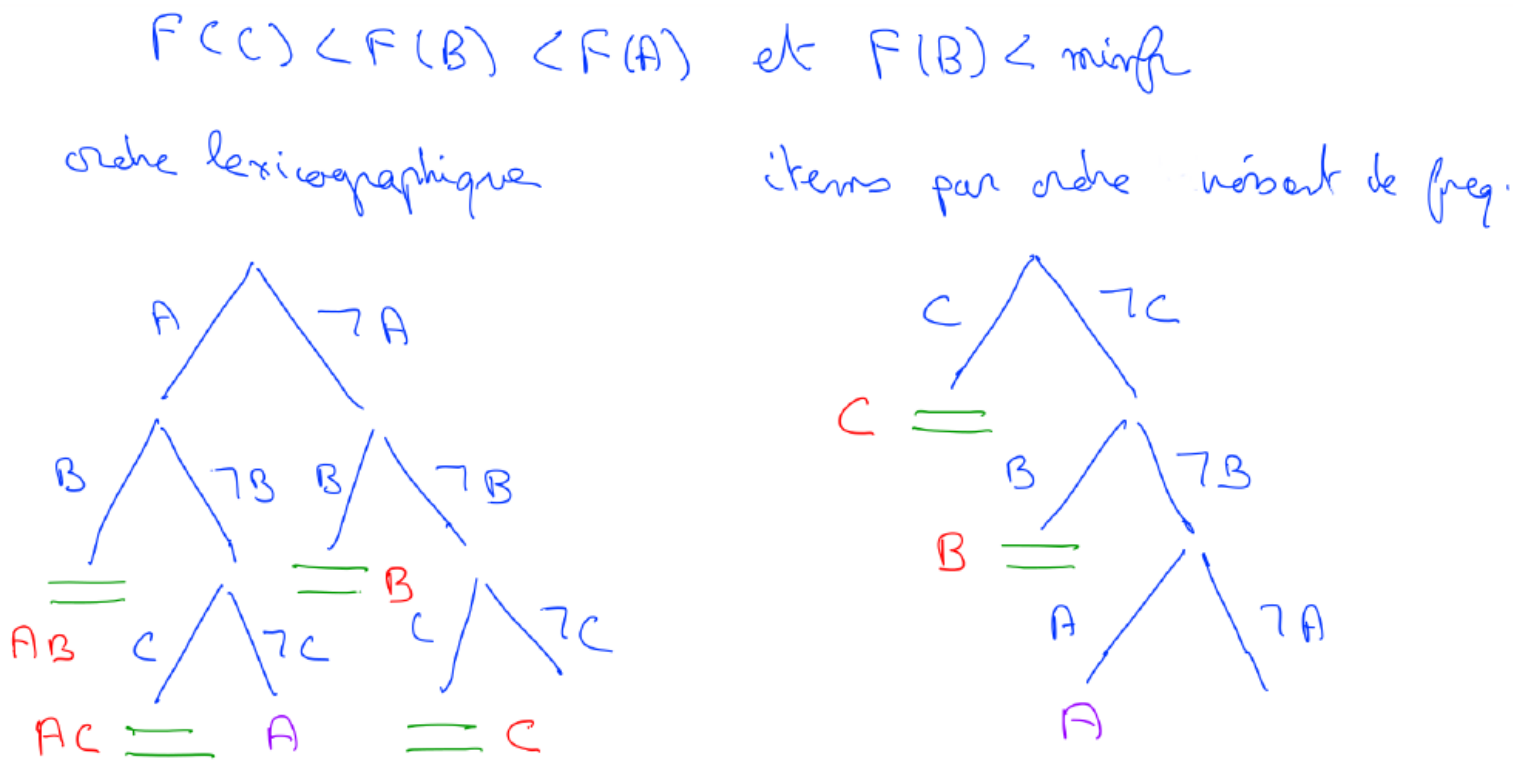
- inférer les règles d'association uniquement à partir des motifs **fermés**

  remarque : il est possible de reconstruire l'ensemble complet des règles d'association à partir des motifs fermés (mais pour cela, un motif fermé Z doit être découpé aussi selon des motifs X et Y tels que $X \cup Y \subsetneq (Z)$

- dans la pratique, on ne génère pas toutes les règles, mais utilisation de **couverture de règles**, par exemple en combinant les motifs minimaux et maximaux des classes d'équivalence (règles informatives et règles informatives approximatives)

Techniques d'élagage (++)

- **algorithme 1 :** tout motif inclus dans un motif fréquent est fréquent. Quel critère d'élagage sûr peut-on tirer de cette propriété ?
- **algorithme 2 :** heuristique : trier les motifs fréquents de longueur 1 par ordre décroissant de fréquence, énumérer les items suivant cet ordre



Fréquence : anti-montone par rapport à la spécialisation des items

- si X est infréquent (i.e., $\mathcal{F}(X) < minfr$), alors $\forall Y, Y \supset X,\ \mathcal{F}(Y) < minfr$ (Y est infréquent)

- si X est fréquent (i.e., $F(X) \geq minfr$), alors $\forall Y, Y \subset X,\ \mathcal{F}(Y) \geq minfr$ (Y est fréquent)