Fouille de Données et Apprentissage

C2 - Règles d'association

Lina Soualmia

Université de Rouen
LITIS - Équipe TIBS-CISMeF
lina.soualmia@chu-rouen.fr

25 janvier 2016

Crédits

- P.Poncelet (Montpellier)
- S.Ullman (Stanford)
- R.Rakotomalala (Lyon)
- N.Pasquier (Nice)
- parmi d'autres ...

Plan

- Rappels C1
- Préparation des données
- Extraction de motifs fréquents
- Extraction de règles d'association

Définition

- La fouille de données (data-mining) est l'étape centrale du processus d'extraction de connaissances des bases de données (ECBD ou KDD pour Knowledge Discovery in Databases).
- Processus non trivial d'extraction de connaissances d'une base de données pour obtenir de nouvelles données, valides, potentiellement utiles, compréhensibles.
- Exploration et analyse, par des moyens automatiques ou semi-automatiques de grandes quantités de données en vue d'extraire des motifs intéressants.

Objectifs

- Un processus non trivial d'extraction de modèles valides, nouveaux, potentiellement utiles et compréhensibles à partir de grands volumes de données
- Objectifs :
  - Compréhension des données et des phénomènes sous-jacents (liens, récurrences, ...etc.)
  - Extrapolation d'informations pour la prédiction d'événements
  - Construction de modèles (calculs) pour la prédiction de valeurs (données)

Définition de l'EC-BD

C'est un processus d'extraction de connaissances :

- Nouvelles :
  - c'est- à dire pas déjà connues
  - On extraira des connaissances connues mais ce n'est pas l'objectif ...
- Potentiellement utiles : réutilisables dans un processus de raisonnement
- et ayant un degré de plausibilité : on cherche à contrôler la plausibilité des connaissances extraites
- dans de grands volumes de données :
  - Nécessitent des processus automatiques
  - Permettent une certaine validité statistique des connaissances extraites

Exemple

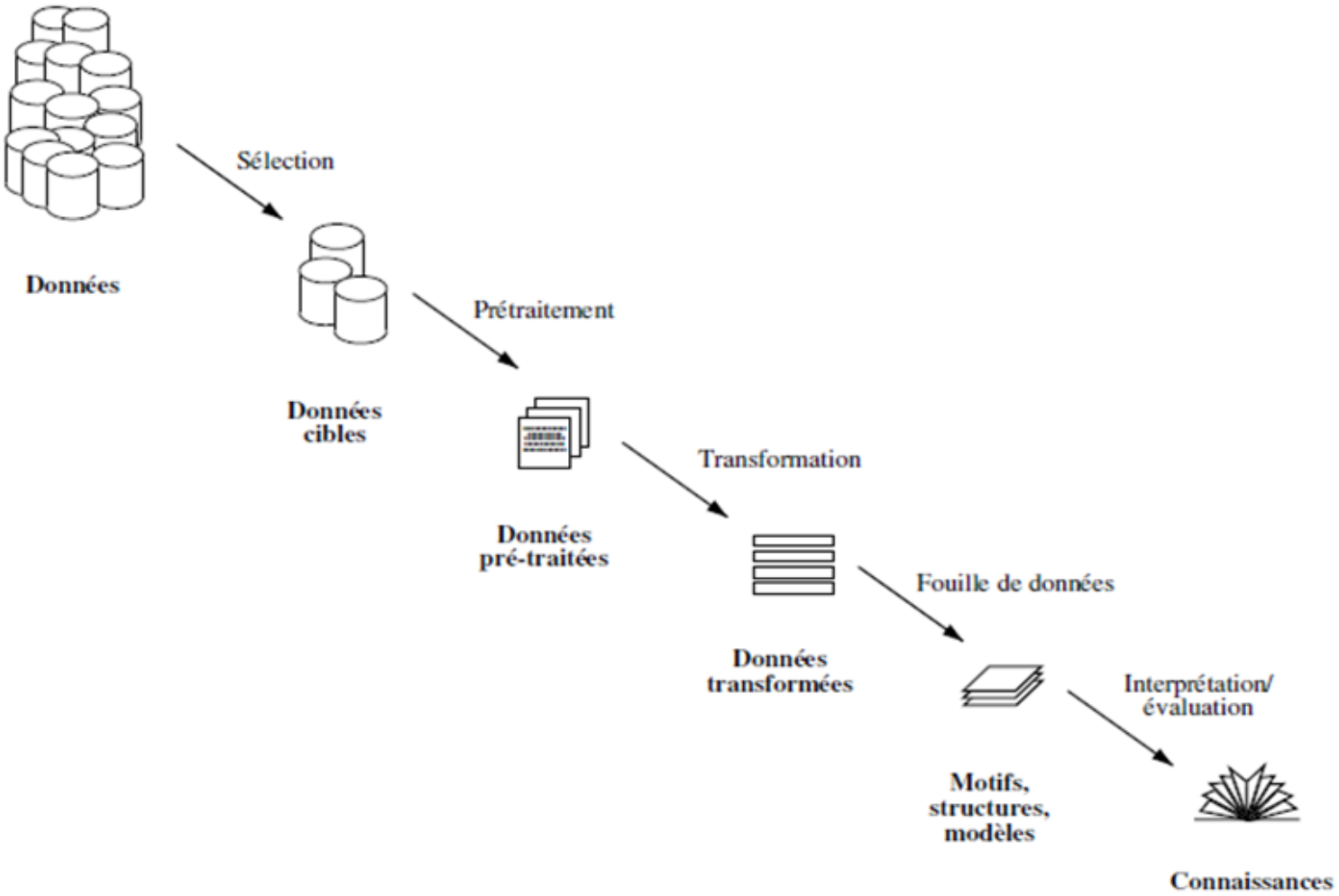
- Supposons qu'on extrait la règle suivante d'une base de données :
  - Blanc $\Longrightarrow$ né aux Etats Unis avec une probabilité de 99,07%
- Cette règle est totalement dépendante des données étudiées
- Elle donne des connaissances sur la base de données considérée
- C'est le même problème que celui de l'échantillonnage en statistique

Restons simples

- Données binaires
- Recherche de motifs fréquents : « lait, oeufs et farine sont souvent achetés ensemble »
- Recherche de règles d'association : « dans 75 % des cas, l'achat de couches va de pair avec l'achat de bière »
- Pas de notion du temps
- Ni disjonctions, ni négations

Principe

- Entrée : des données
- Sortie : des connaissances
- Les données ont des structures hétérogènes
  - On les trouve dans des bases de données relationnelles
  - Bases de données objet
  - Sous forme de liste de fichiers
  - Sur des pages Web
  - Dans des textes


Sélection
Données
Prétraitement
Données cibles
Transformation
Données pré-traitées
Fouille de données
Données transformées
Interprétation/ évaluation
Motifs, structures, modèles
Connaissances

Étapes

- Intégration des sources de données et sélection des données intéressantes
- Transformation et formatage :
  - Données brutes $\rightarrow$ données préparées et formatées
  - Transformation : enlever les données bruitées, traiter les données manquantes
  - Formatage : sous forme traitable par les algorithmes de fouille
- Fouille de données :
  - données préparées et formatées $\rightarrow$ éléments d'information extraits
  - C'est l'étape algorithmique du processus d'ECBD
  - mise en œuvre de méthodes de fouilles
- Interprétation
  - Éléments d'information extraits $\rightarrow$ unités de connaissances

Processus

- Itératif : se fait en plusieurs passes
- Interactif : l'analyste est dans la boucle et il oriente le processus selon ses connaissances du domaine des données et selon son intuition
- L'analyste est un expert du domaine des données, s'il connait les principes de l'informatique et de l'ECBD c'est un plus
- C'est l'analyste qui effectue l'interprétation
- Les éléments d'information extraits des processus de fouille doivent être sous une forme compréhensible par l'analyste (intelligibilité)
- Il est utile d'avoir des connaissances du domaine représentées et manipulables informatiquement pour assister l'analyste. Ces connaissances du domaine sont souvent sous la forme d'une ou plusieurs ontologies.
- Si une connaissance extraite n'est pas nouvelle et qu'elle est représentée en machine, inutile de demander une confirmation de l'analyste

Objectifs : décrire ou prédire

- Caractérisation :
  - Identification de critères d'appartenance à une classe
- Discrimination :
  - Identification de critères discriminants entre l'appartenance à deux classes
- Extraction de règles d'association
  - Recherche de relations « attribut-valeur » qui occurrent ensemble fréquemment
- Classification
  - Construction d'un modèle ou d'une fonction à partir du jeu de données, pour permettre la prédiction de l'appartenance du nouvel objet à une classe
- Clustering
  - Regroupement d'objets de sorte à minimiser la distance entre objets similaires et à maximiser la distance entre objets différents

Fonctionnalités

Caractérisation–Discrimination

- Requêtes SQL
- Requêtes OLAP (Online Analytical Processing)
- Description analytique
- Mesures statistiques

Analyse d'association (corrélation et causalité)

- Découvrir des règles d'association de la forme X $\rightarrow$Y où X et Y sont des conjonctions de termes attributs-valeurs ou prédicats
- Les mesures de support et confiance indiquent la portée et la précision de la règle
- Exemples :
  - Achat=pain et Achat=café $\rightarrow$ Achat=beurre (support = 5%, confiance = 70%)
  - Age>20 et Age<29 et Revenu>1000 $\rightarrow$ Achète_PC="oui" (support = 2%, confiance = 60%)

Clustering

- Trouver des groupes ou classes d'objets tels que la similarité intra-classe est élevée et la similarité inter-classes est faible
- Pas de variable identifiant la classe
- Classification (apprentissage) non-supervisé : classes inconnues à l'avance
- Exemples :
  - segmentation des clients
  - cluster d'étoiles caractérisées par leur luminosité et leur température

Classement et prédiction (classification supervisée)

- Apprendre un modèle qui associe un objet à une classe prédéfinie
- Apprendre une fonction permettant de prédire la valeur d'une variable numérique
- Exemples :
  - Classer des patients selon leur risque de développer une malade en fonction de leurs symptômes
  - Évaluer le risque d'un incident de remboursement en fonction des caractéristiques des demandeurs de crédit
- Forme du résultat :
  - Arbres de décision, règles de classification, réseaux de neurones, classifieurs Bayésiens ...etc.

Fonctionnalités

Analyse de déviations

- Déviation : objet qui n'est pas conforme au comportement général
- Donnée bruitée ou exception : information utile dans le cas de détection de fraudes ou d'évènements rares

Recherche de corrélations

- Analyse statistique
- Recherche de motifs séquentiels
- Analyse de régression

Méthodes de fouille de données

- Il existe plusieurs moyens de différencier les méthodes de fouille de données.
- Un moyen classique est de les distinguer selon leur objectif
- Les méthodes descriptives
  - Caractérisation, discrimination
  - Extraction de règles d'association
  - Analyse de cas extrêmes
- Les méthodes prédictives
  - Classification
  - Clustering

Méthodes de fouille de données

- On peut les distinguer en fonction du type de données qu'elles traitent en entrée
- Méthodes symboliques
  - Extraction de motifs fréquents
  - Extraction de règles d'associations
  - Classification par treillis
  - Classification par arbres de décision
- Méthodes numériques issues de la reconnaissance des formes
  - Méthodes statistiques
  - Analyse de données (classification automatique, classification par composantes principales)
  - Modèles de Markov caché d'ordre 1 et 2 (très bons résultats en fouille de données en génomique)
  - Les méthodes neuronales, algorithmes génétiques
- Fouille de textes :
  - Fouille de base de données bibliographiques d'un domaine pour le décrire

Limites

Deux points critiques au niveau des données :

- A cause du volume de données étudiées, il faut faire un passage à l'échelle pour passer :
  - Des algorithmes d'apprentissage standards qui ne sont applicables en pratique que sur des volumes de données relativement faibles
  - Aux algorithmes qui prennent en compte de très grands volumes de données
- Qualité des données :
  - Problèmes dus à l'absence de données (il en manque) et aux bruits (données erronées)
  - Nécessite d'avoir des méthodes pour améliorer cette qualité notamment par filtrage

Préparation des données

- Données réelles imparfaites/endommagées
  - Incomplètes
  - Bruitées
  - Incohérentes
- Nécessité de préparer les données
  - Nettoyage
  - Intégration et transformation
  - Réduction
  - Discrétisation

Types de données

- Numériques linéaires :
  - Ex : poids, taille, longitude, vitesse, etc.
- Binaires : une valeur parmi deux possibles
  - 0 : la variable est absente, 1 : la variable est présente
- Nominales : valeur prise dans une liste finie
  - Ex : couleur peut être « vert, bleu, rouge, jaune, noir »
- Ordinales : l'ordre des valeurs est plus important
  - Ex : résultat d'un concours
- Par ratios : variables numériques sur une échelle exponentielle/logarithmique
  - Valeurs non-linéaires

Préparation des données

- Nettoyage
  - Compléter les valeurs manquantes, lisser les données bruitées, supprimer les déviations et corriger les incohérences
- Intégration
  - Intégrer des sources de données multiples
- Transformation
  - Normaliser
- Réduction
  - Réduire le volume des données (agréger, supprimer une dimension, etc.)
- Discrétisation
  - Pour les attributs numériques, permet de réduire le volume

Nettoyage-Valeurs manquantes

- Les valeurs manquantes peuvent être codées par diverses valeurs :
  - <Vide>, « 0 », « . », « NA », « ? » , « NULL »
  - Il est nécessaire d'uniformiser le code
- Valeurs manquantes sont interdites
  - Ignorer le tuple
  - Compléter la valeur à la main
  - Utiliser une constate globale
  - Utiliser la valeur moyenne
  - Utiliser la valeur moyenne pour les exemples d'une même classe
  - Utiliser la valeur la plus probable

Nettoyage - Données bruitées

On peut :

- Trier et partitionner (discrétiser)
- Classifier (exceptions)
- Appliquer un modèle de prédiction (ex : une fonction de régression)

Partitionnement et lissage

- Les valeurs triées sont réparties en largeur (distance)
  - La suite triée est partitionnée en N intervalles de même amplitude
  - Amplitude de chaque intervalle $W = \frac{(max - min)}{N}$
  - Solution la plus simple mais les exceptions peuvent dominer
- Les valeurs triées sont réparties en profondeur (fréquence)
  - La suite triée est partitionnée en N intervalles contenant le même nombre de valeurs

Partitionnement

Exemple :

- Données triées : 4 8 9 15 21 21 24 25 26 28 29 34
- Partition en profondeur :
  - Part 1 : 4 8 9 15
  - Part 2 : 21 21 24 25
  - Part 3 : 26 28 29 34

Lissage

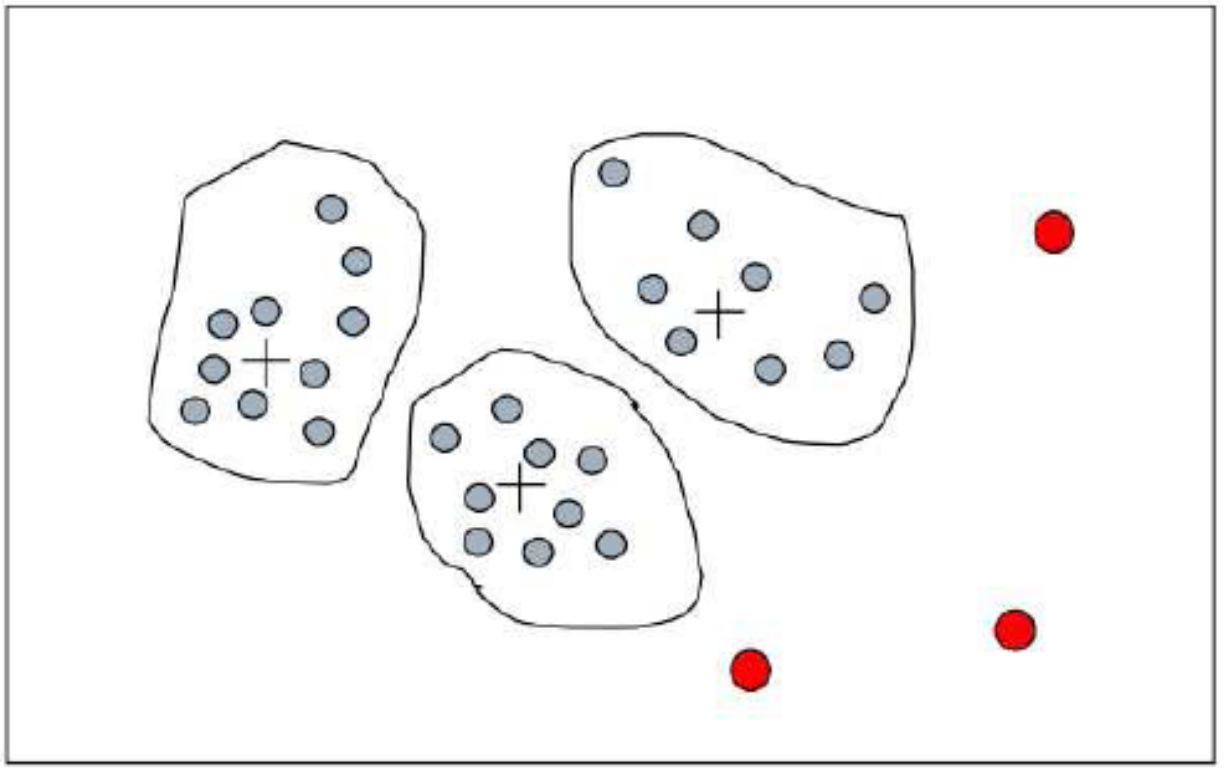
Exemple :

- Lissage par les moyennes : chaque valeur de la partition est remplacée par la moyenne
  - Part 1 : 9 9 9 9
  - Part 2 : 23 23 23 23
  - Part 3 : 29 29 29 29
- Lissage par les extrêmes : chaque valeur de la partition est remplacé par la valeur extrême la plus proche
  - Part 1 : 4 4 4 15
  - Part 2 : 21 21 25 25
  - Part 3 : 26 26 26 34

Nettoyage - Supprimer les déviations

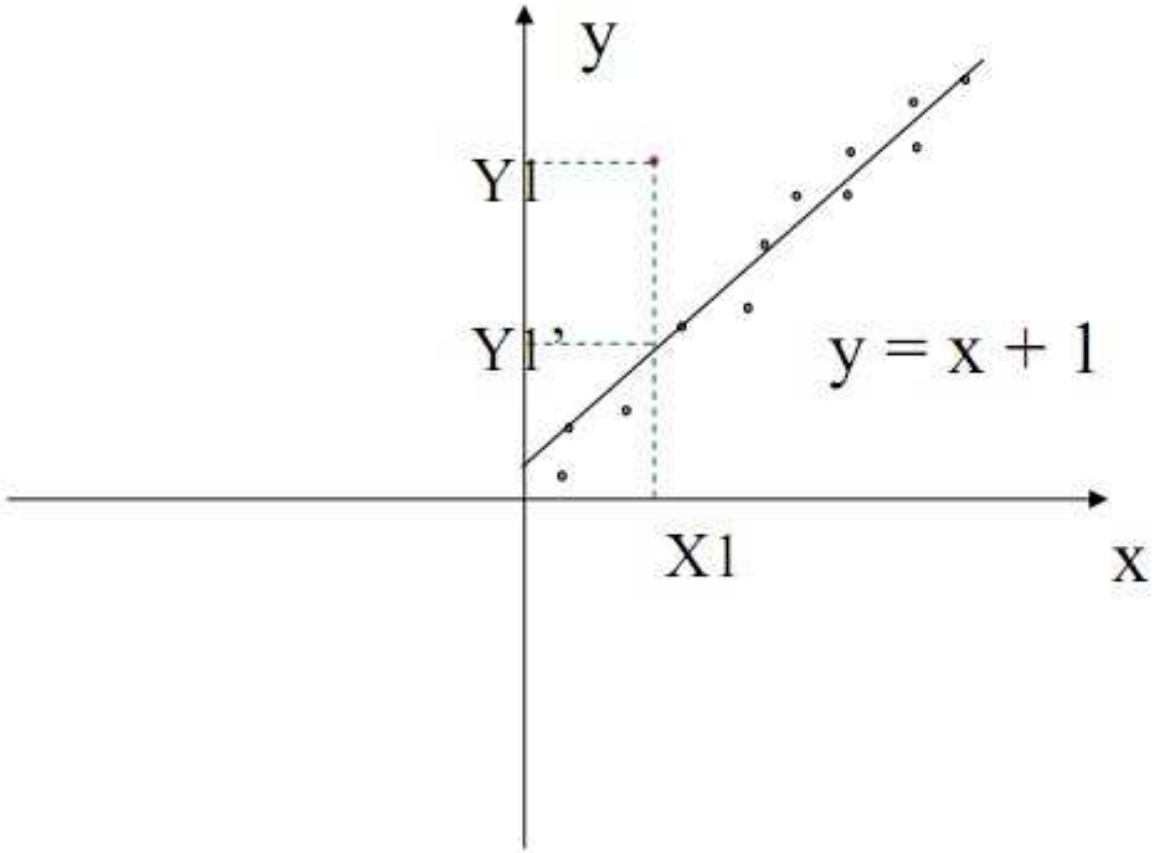
Classifier

- Les valeurs similaires sont organisées en classes
- Les valeurs hors-classes sont considérées comme des déviations



Lissage par régression

- Les données sont lissées de manière à approcher une fonction
  - Régression linéaire
  - Régression linéaire multiple



Intégration

- Combinaison de données issues de différentes sources
- Intégration de schémas
  - Identifier les entités similaires
  - Ex : ID, Code, Matricule
- Détection et résolution de conflits
  - Résoudre les problèmes d'attributs symbolisant les mêmes entités avec des représentations différentes, des unités différentes
  - Ex : âge et date de naissance

Redondances

- Détection de données redondantes par analyse de corrélation
- Par ex. redondance entre attributs : Prix HT et TTC
- Mesure de la corrélation entre les attributs A et B

Transformation

Les transformations appliquées :

- Le lissage qui supprime les données bruitées
- L'agrégation qui calcule des sommes, moyennes . . . etc.
- La généralisation qui remonte dans une hiérarchie de concepts (taxonomie, ontologie)
- La normalisation qui ramène les valeurs dans un intervalle donné
- La construction d'attributs

Réduction

- Permet d'obtenir une représentation réduite d'ensembles volumineux de données
- Stratégies appliquées
  - Agrégation
  - Réduction de dimensions
  - Compression
  - Discrétisation

Réduction de dimensions

- Suppression d'attributs : la présence d'attributs non pertinents détériore les performances des algorithmes
  - Par exemple, les algorithmes d'induction d'arbres
- Pour assurer de bonnes performances aux algorithmes d'extraction :
  - Supprimer les données non pertinentes
    - Identifiants, attributs à valeur unique
  - Supprimer les données redondantes
    - Données déductibles d'autres données
    - Âge et année de naissance

Réduction - Discrétisation

- Permet de réduire le nombre de valeurs d'un attribut continu en divisant le domaine de valeurs en intervalles.
- Utile pour la classification, extraction d'associations.
- Des techniques de discrétisation peuvent être appliquées récursivement pour fournir un partitionnement hiérarchique de l'attribut.

Exemple – jeu de Tennis

Attribut de classe : Play (yes/no)

- Yes : jeu possible ;
- No : jeu impossible.

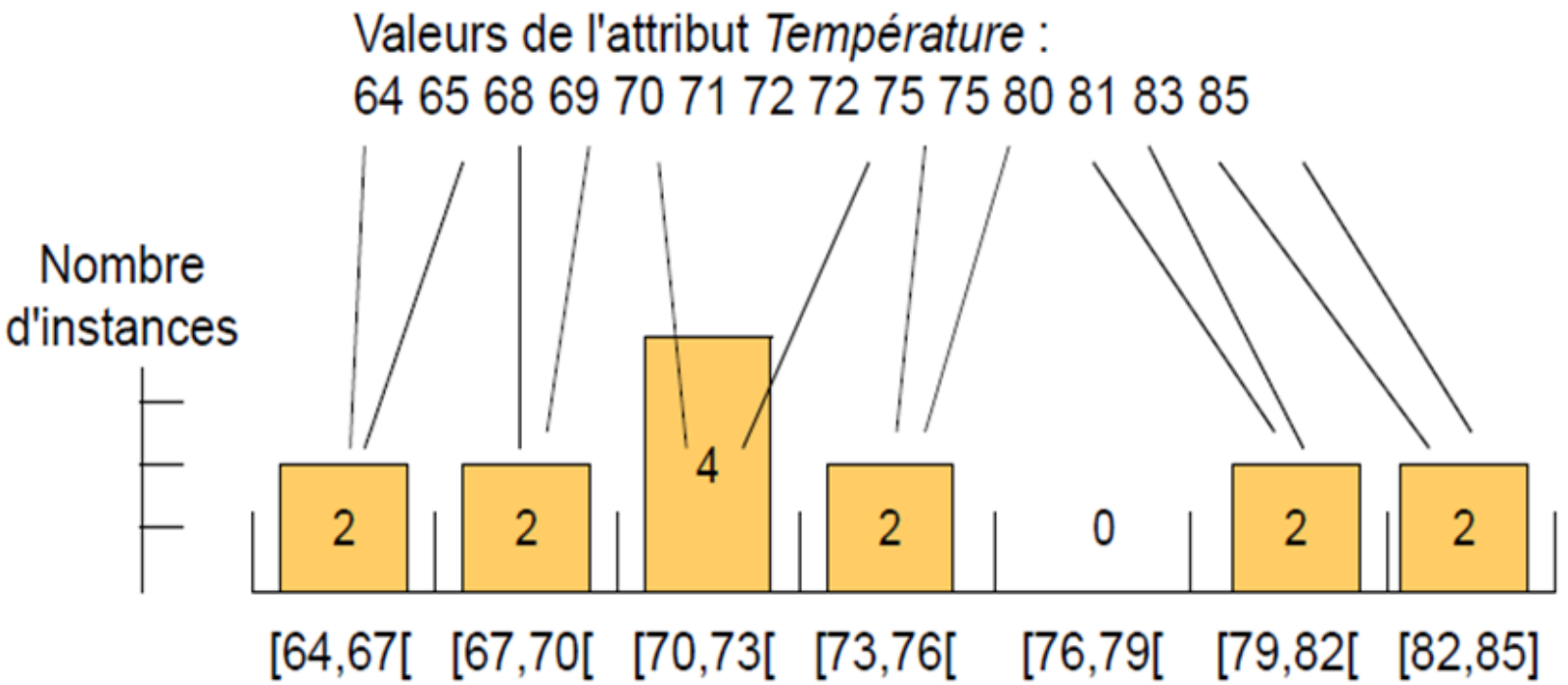
Autres attributs : météo

- Outlook : ciel (sunny, overcast, rainy)
- Température : degrés K
- Humidity : tx d'humidité
- Windy : présence de vent (Vrai/Faux)

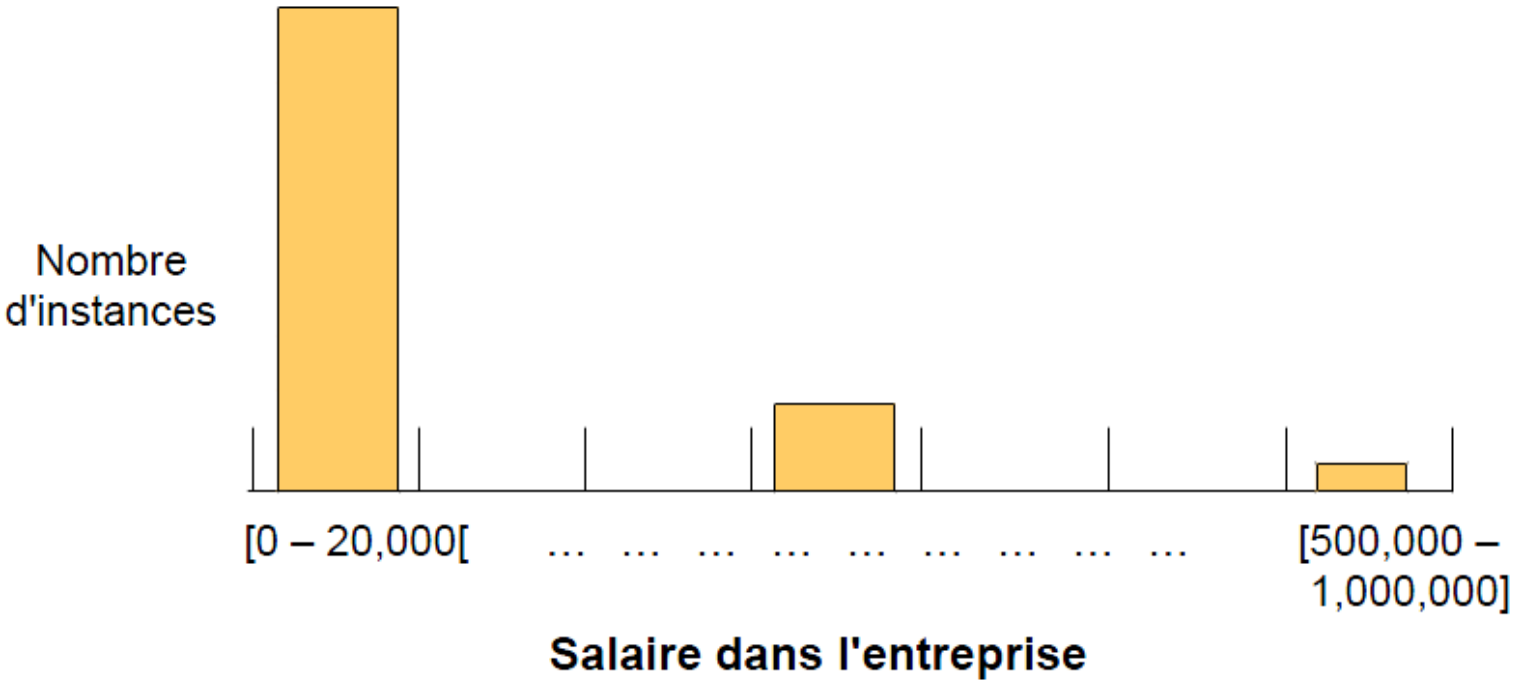
outlook	temperature	humidity	windy	play
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	86	false	yes
rainy	70	96	false	yes
rainy	68	80	false	yes
rainy	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rainy	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rainy	71	91	true	no

Discrétisation en largeur

- Discrétisation par intervalles égaux des valeurs



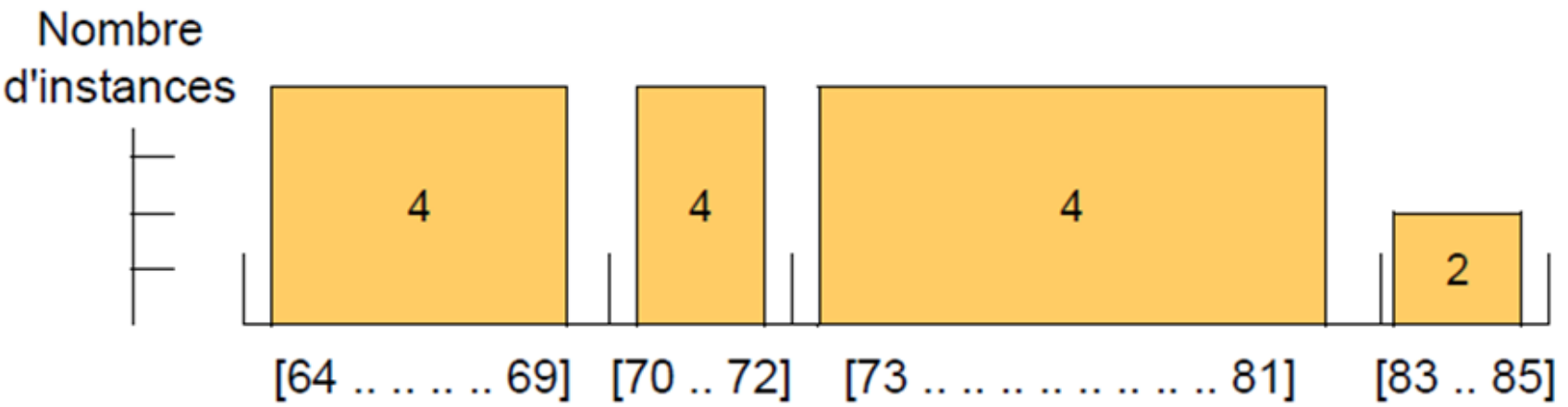
- Peut entraîner des inégalités importantes dans les effectifs



Discrétisation en profondeur

- Discrétisation par effectifs égaux
- Tailles égales excepté pour le dernier intervalle

Valeurs de l'attribut *Température* :
64 65 68 69 70 71 72 72 75 75 80 81 83 85



Discrétisation : autres méthodes

- Présence de seuils significatifs
  - Ex : Age $>$ 18 ans
- Discrétisation supervisée
  - Prend en compte la classification

temperature	64	65	68	69	70	71	72	73	74	81	83	85
class	yes	no	yes no	yes	yes no	no	no yes	yes yes	no	yes	yes	no

- Utilise l'entropie pour mesurer l'information et obtenir un critère de « pureté »

temperature	64	65	68	69	70	71	72	73	74	81	83	85
class	yes	no	yes no	yes	yes no	no	no yes	yes yes	no	yes	yes	no

- Comptage des occurrences de yes et no pour class
- Les intervalles maximisent les co-occurrences des valeurs

Jeu de données discrétisé

- Valeurs catégorielles uniquement
- Simplifie l'interprétation du résultat
- Améliore les performances (temps de calcul)

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Sites et outils :

- LA référence : KDNuggets
  - http ://www.kdnuggets.com/
- TheData Mine
  - http ://www.the-data-mine.com/
- Weka : en TPs
  - http ://www.cs.waikato.ac.nz/~ml/
- SPSS (SPSS Clementine)
  - http ://www.spss.com/SPSSBI/Clementine/
- IBM (Intelligent Data Miner)
  - http ://www.ibm.com/Stories/1997/04/data1.html

Extraction de motifs fréquents

Motivation

- L'extraction de *motifs fréquents* est une technique très utilisée en fouille de données et s'appuie sur des principes relativement simples
- Son objectif est de trouver des *motifs* qui apparaissent *fréquemment* dans une base de données
- L'exemple le plus connu est l'analyse du panier de la ménagère :
  - On dispose d'une base de données des tickets de caisse d'un supermarché
  - L'objectif est d'analyser les achats qui sont fréquemment associés $\Rightarrow$ les motifs fréquents dans les tickets de caisse
  - On a trouvé ainsi que les couches pour nourrissons sont souvent associés à la bière ...

Principe

- La version de base de l'extraction de motifs fréquents permet de faire la fouille dans une relation (table) d'une base de données relationnelle dont les valeurs sont des booléens qui indiquent la présence ou l'absence d'une propriété.
- Définition : une base de données formelle est la donnée d'un triplet (O, P, R) où :
  - O est un ensemble fini d'objets
  - P est un ensemble de propriétés
  - R est une relation sur $O \times P$ qui permet d'indiquer si un objet x a une propriété p (noté $x\ R\ p$).

Exemple

R	a	b	c	d	e
x_1	1	0	1	1	0
x_2	0	1	1	0	1
x_3	1	1	1	0	1
x_4	0	1	0	0	1
x_5	1	1	1	0	1
x_6	0	1	1	0	1

- $O = \{x_1, x_2, x_3, ..., x_6\}$
- $P = \{a, b, c, d, e\}$
- $x\ R\ p$ si et seulement si la ligne de x et de p se croisent sur un 1 : $x_1\ R\ a$

Définition d'un motif

- Un motif d'une base de donnée formelle (O, P, R) est un sous-ensemble de P.
- L'ensemble de tous les motifs d'une base est donc l'ensemble des parties de P noté 2^P
- On dira qu'un objet $x \in O$ possède un motif m si $\forall\, p \in m,\ x\, R\, p$

Exemples de motifs

Pour la base on a $2^P = 2^5 = 32$ motifs

Motifs fréquents

- Parmi cet ensemble de motifs, on va chercher ceux qui apparaissent fréquemment.
- Pour chercher les motifs fréquents on a besoin d'introduire les notions de *connexion de Galois* et de *support* d'un motif.

Connexion de Galois

- La connexion de Galois associée à une base de données formelle (O, P, R) est le couple de fonctions (f, g) définies par :
- $f : 2^P \longrightarrow 2^O$
  $m \longmapsto f(m) = \{x \in O \mid x \text{ possède } m\}$
- $g : 2^O \longrightarrow 2^P$
  $X \longmapsto g(X) = \{p \in P \mid \forall x \in X, x R p\}$
- g est dit dual de f et f dual de g.
- $f(m)$ est l'image du motif m.

Support d'un motif

- Soit $m \in 2^P$ un motif. Le support de m est la proportion d'objets dans O qui possèdent le motif :
  - $support : 2^P \longrightarrow [0;1]$
    $m \longmapsto support(m) = \frac{|f(m)|}{|O|}$
  - exemple $support(a) = 3/6$; $support(P) = ?$
- Le support est décroissant
- Si m est un sous-motif de m' $(m \subseteq m')$ alors $support(m) \geq support(m')$

Motif fréquent

- Le support mesure la fréquence d'un motif :
  - plus il est élevé, plus le motif est fréquent.
- On distinguera les motifs *fréquents* des motifs non fréquents à l'aide d'un seuil σ_s
- Soit $\sigma_s \in [0; 1]$. Un motif m est fréquent si $support(m) \geq \sigma_s$ sinon il est dit non fréquent.

Extraction des motifs fréquents

- Une approche naïve pour l'extraction des motifs fréquents consiste à
  - parcourir l'ensemble 2^P de tous les motifs,
  - à calculer leurs supports
  - et à ne garder que les plus fréquents.
- Malheureusement cette approche est trop consommatrice en temps : le nombre de motifs est $2^{|P|}$ et en pratique on veut manipuler des bases ayant un grand nombre de propriétés.
- L'approche que nous allons décrire permet d'extraire des motifs fréquents dans une base ayant plusieurs milliers de propriétés et plusieurs millions d'objets.

Extraction par niveaux

- L'extraction par niveaux s'appuie sur la décroissance du support et se fait selon le principe suivant :

1. On commence par chercher les motifs fréquents de longueur 1 ;
2. On combine ces motifs pour obtenir des motifs de longueur 2 et on ne garde que les fréquents parmi eux ;
3. On combine ces motifs pour obtenir des motifs de longueur 3 et on ne garde que les fréquents parmi eux ;
4. ...etc.

Extraction par niveaux

- Tout sous-motif d'un motif fréquent est fréquent.
  - si $m' \subseteq m$ et $support(m) \geq \sigma_S$ alors $support(m') \geq \sigma_S$
- Tout super-motif d'un motif non fréquent est non fréquent
  - si $m' \supseteq m$ et $support(m) < \sigma_S$ alors $support(m') < \sigma_S$

Propriétés

- Support pour les sous-ensembles
  - Si pour $A \subseteq B$ les motifs A, B alors $support(A) \geq support(B)$
  - car toutes les les transactions dans la base qui supportent B supportent aussi nécessairement A
  - A={Café, Sucre}, B={Café, Sucre, Lait}
- Les sous-ensembles d'ensembles fréquents sont fréquents
- Les sur-ensembles d'ensembles non fréquents sont non fréquents

`Apriori`

entrées : $(\mathcal{O}, \mathcal{P}, \mathcal{R})$: une base de données formelle
$\sigma_s \in [0; 1]$

sortie : l'ensemble des motifs fréquents de la base, relativement au seui

Début

$i \leftarrow 1$

$\mathtt{C}_1 \leftarrow$ ensemble des motifs de taille 1

tant que $\mathtt{C}_i \neq \emptyset$ **faire**

Calculer le support de chaque motif $m \in \mathtt{C}_i$

$\mathtt{F}_i \leftarrow \{m \in \mathtt{C}_i \mid \mathtt{support}(m) \geqslant \sigma_s\}$

$\mathtt{C}_{i+1} \leftarrow \mathtt{générer\text{-}candidats}(\mathtt{F}_i)$

$i \leftarrow i + 1$

fin-tant que

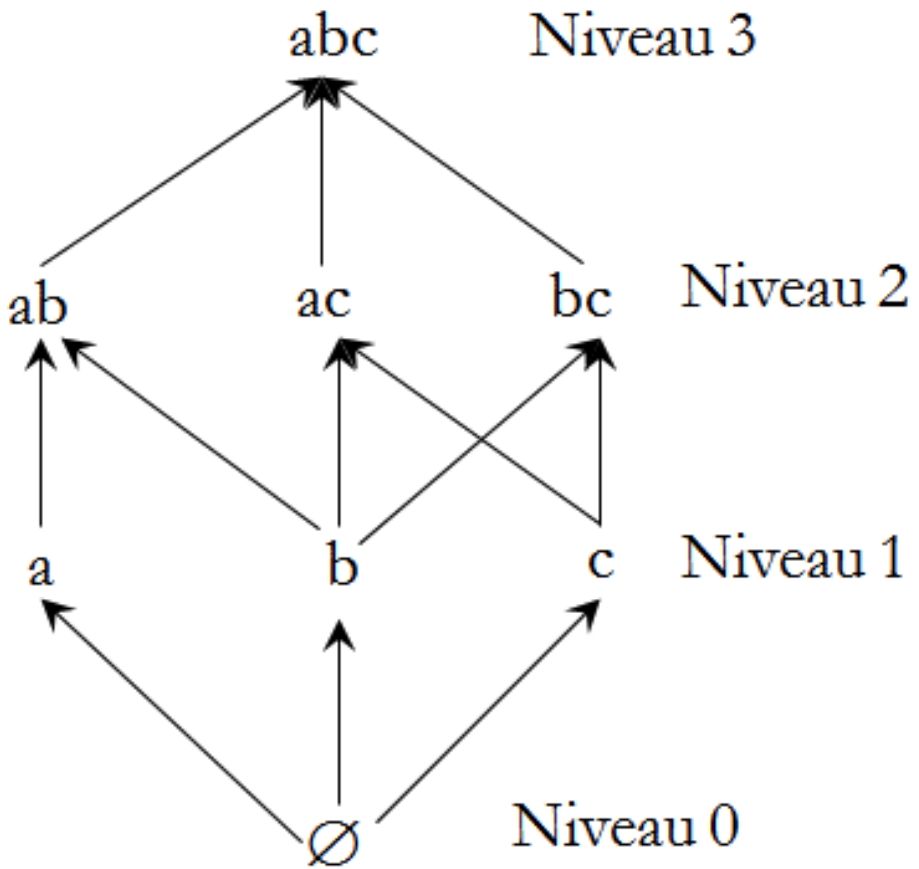
retourner $\bigcup_{i \geqslant 1} \mathtt{F}_i$

Fin

Déroulement de l'algorithme Apriori sur l'exemple avec $\sigma_S = 2/6$

R	a	b	c	d	e
x_1	1	0	1	1	0
x_2	0	1	1	0	1
x_3	1	1	1	0	1
x_4	0	1	0	0	1
x_5	1	1	1	0	1
x_6	0	1	1	0	1

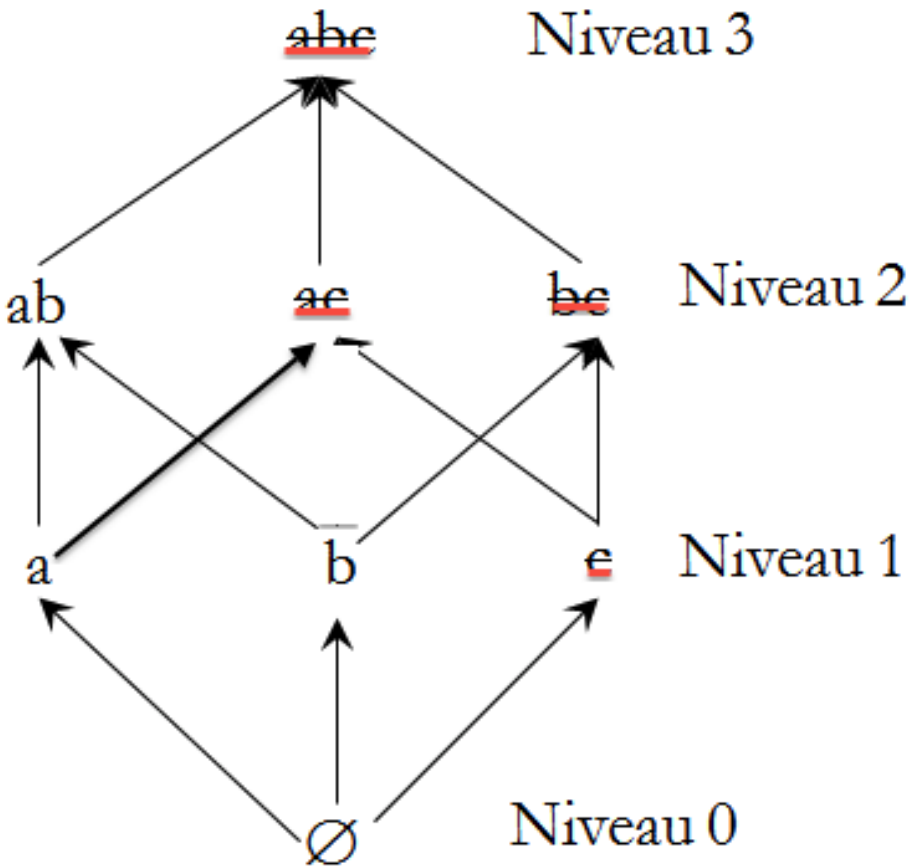
- On peut voir cet algorithme comme le parcours du treillis des parties de P ordonné pour l'inclusion Si par exemple P={a, b, c} le treillis des parties de P est :

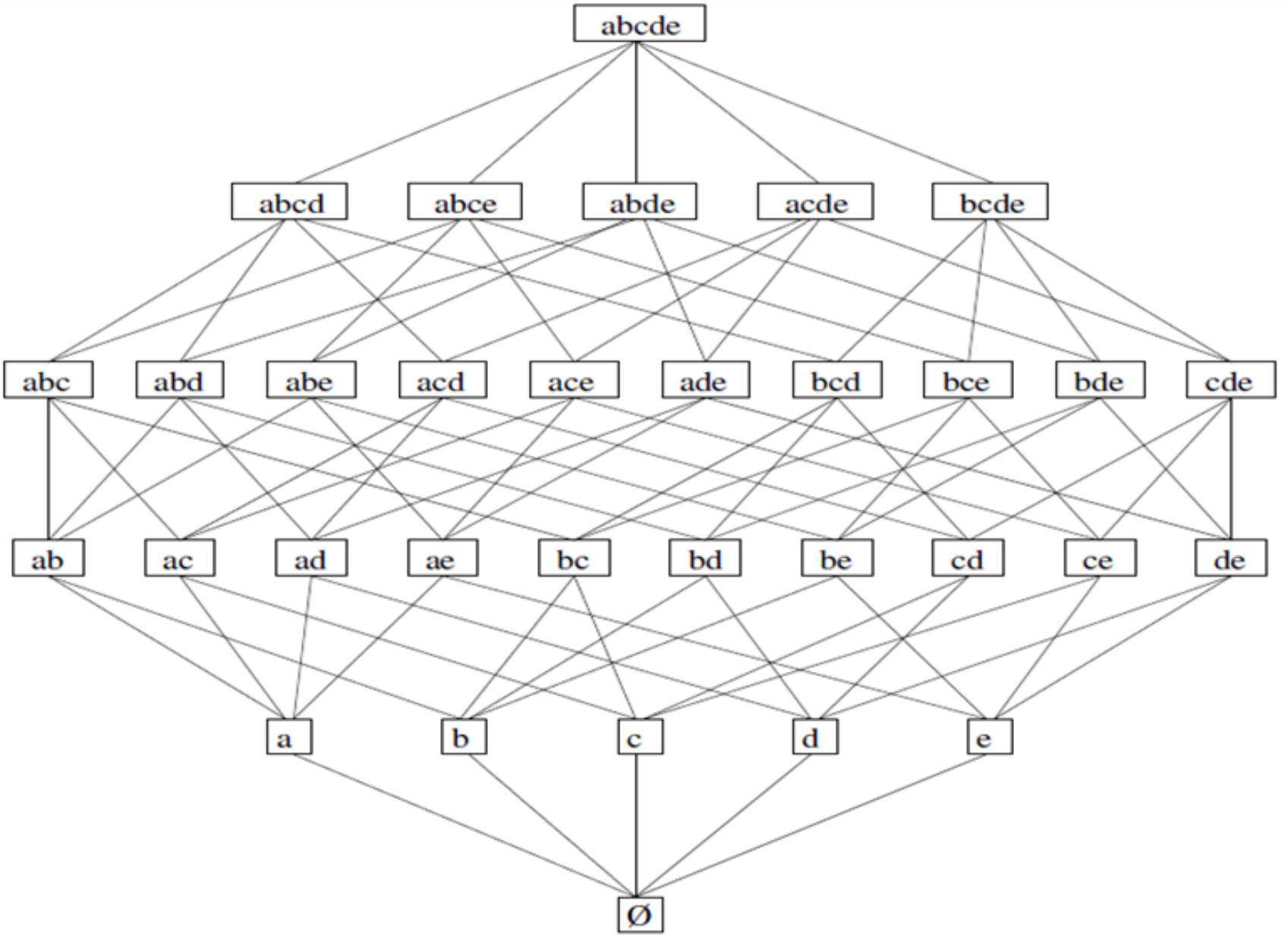


Treillis des parties

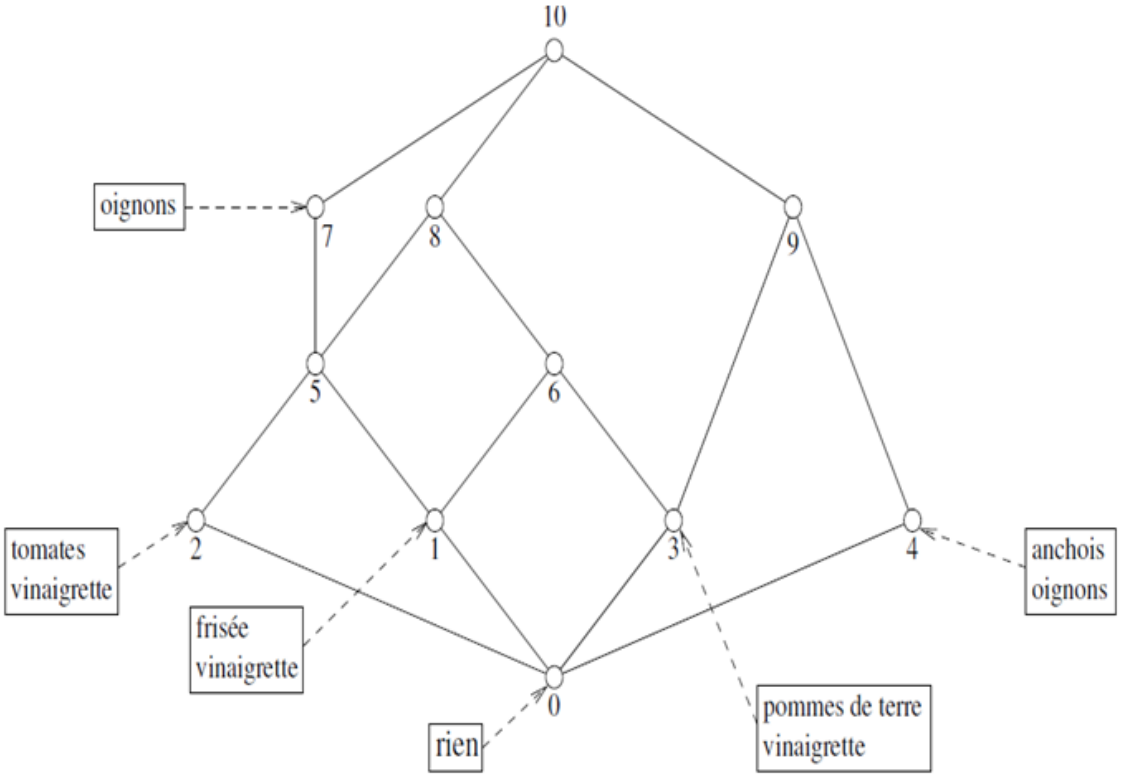
- Il est parcouru par niveau croissant à partir du niveau i=1.
- Quand un motif n'est pas fréquent, tous ses super-motifs sont non fréquents

- Par exemple, c n'est pas fréquent et par conséquent aucun de ses super-motifs n'est considéré
- On a ainsi élagué le parcours du treillis
- Un des objectifs de cet algorithme est de diminuer le nombre d'accès à la base de données




abcde
abcd
abce
abde
acde
bcde
abc
abd
abe
acd
ace
ade
bcd
bce
bde
cde
ab
ac
ad
ae
bc
bd
be
cd
ce
de
a
b
c
d
e
Ø

Treillis de salades



1. Frisée vinaigrette ; 2. Tomates vinaigrette ; 3. Pommes de terre vinaigrette ;

4. Anchois, oignons ; 5. Frisée, tomates, vinaigrette ;

6. Frisée, pommes de terre, vinaigrette ; 7. Frisée, tomates, oignons, vinaigrette ;

8. Frisée, tomates, pommes de terre, vinaigrette ; 9. Anchois, oignons, pommes de terre, vinaigrette ;

10. Frisée, tomates, anchois, oignons, pommes de terre, vinaigrette.

Remarque sur le seuil σ_S

- Le seuil σ_S est fixé par l'analyste qui peut suivre une approche itérative en fixant un seuil au départ et, en fonction du résultat, changera la valeur du seuil :
  - Si trop de motifs fréquents ont été trouvés, il augmentera le seuil
  - Dans le cas inverse il le diminuera
- On peut par ailleurs constater que le temps de calcul de l'algorithme Apriori décroit avec le seuil.
- Par conséquent, si l'analyste fixe une valeur de seuil trop grande, cela gaspillera moins de temps que s'il en fixe une trop petite.

Extraction des règles d'association

Définition

- Les règles d'association sont de la forme : $r = p_1 \rightarrow p_2$ où p_1 et p_2 sont deux motifs
- Une telle règle peut se lire intuitivement :
  - Si un objet x possède p_1 alors il est plausible que x possède p_2.
- On s'appuie également sur la notion de base de données formelle *(O, P, R)*
- Une règle d'association r est la donnée de deux motifs A et B. On la note $r = A \rightarrow B$
- A est appelé antécédent de la règle et B son conséquent

Support et confiance

- Le support d'une règle d'association $r = A \rightarrow B$ est :
  - $support(A \rightarrow B) = support(A \cup B)$
- La confiance d'association $r = A \rightarrow B$ est :
  - $confiance(A \rightarrow B) = \frac{support(A \cup B)}{support(A)}$
  - $confiance(r) \in [0; 1]$
  - Elle indique la proportion des objets qui possèdent à la fois A et B parmi les objets qui possèdent A
  - $confiance(A \rightarrow B) = 1$: tous les objets qui possèdent le motif A possèdent le motif B
  - $confiance(A \rightarrow B) = 0$: signifie que les objets possédant le motif A ne possède le motif B.

Remarque

- On peut rapprocher le support d'une règle d'une probabilité :
  - $support(A \rightarrow B)$ est la probabilité que x possède le motif A **et** que x possède le motif B .
  - Propriété : $support(A \rightarrow B) = support(B \rightarrow A)$
- On peut rapprocher la confiance des probabilités conditionnelles :
  - $confiance(A \rightarrow B)$ est la probabilité que x possède le motif B sachant que x possède le motif A

Règle valide

- Parmi les règles on s'intéresse aux règles dites valides :
- Une règle d'association r est valide si elle vérifie :
  - $support(r) \geq \sigma_S$
  - $confiance(r) \geq \sigma_C$
  - σ_S, $\sigma_C \in [0; 1]$ sont les valeurs de seuil prédéfinies

Règle exacte ; Règle approximative

- Une règle d'association est exacte si $confiance(r) = 1$
- Dans le cas contraire elle est dite approximative
- Si $r = A \rightarrow B$ est exacte alors $support(A \cup B) = support(A)$
- Ex : $support(a) = support(ac) = \frac{3}{6}$; $a \rightarrow c$ est une règle exacte
- Mais : ce n'est pas parce qu'une règle est exacte qu'elle est certaine :
  - cela signifie seulement qu'il n'existe aucun contre-exemple, dans la base de données fouillée, à cette règle

Remarque

- Si on a par exemple $ab \rightarrow ac$ on peut noter qu'elle a même support et même confiance que la règle $ab \rightarrow c$
- Répéter en partie droite d'une règle une propriété apparaissant en partie gauche est inutile.
- De façon générale quand on a une règle $A \rightarrow B$ alors $A \cap B = \phi$
- On peut noter les règles sous la forme $r = p_1 \rightarrow p_2 \backslash p_1 \text{avec } p_1 \subseteq p_2$
- $E \backslash F = \{x \mid x \in E \text{ et } x \notin F\}$
- Par exemple pour $ab \rightarrow c$ on aura $p_1 = ab$ et $p_2 = abc$

Support et confiance de r

- $r = p_1 \rightarrow p_2 \backslash p_1$
- $support(r) = support(p_2)$
  - $ab \rightarrow c$; $support(r) = support(abc)$
- $confiance(r) = support(p_2)/support(p_1)$
  - $ab \rightarrow c$; $confiance(r) = support(abc)/support(ab)$

Propriétés des règles d'association

- Pas de composition des règles :
  - Si X$\rightarrow$Z et Y$\rightarrow$Z sont vrais dans la base, X,Y $\rightarrow$Z n'est pas nécessairement vrai
- Pas de transitivité
  - Si X$\rightarrow$Y et Y$\rightarrow$Z sont vrais dans la base, nous ne pouvons pas en déduire que X $\rightarrow$Z

Schéma algorithmique de base

- La plupart des approches utilisent le même schéma algorithmique
- Pour construire les règles d'association, le support de tous les motifs fréquents dans la base doit être calculé
- L'algorithme procède en 2 phases :

1. Génération de tous les ensembles fréquents
2. Génération des règles d'association

Algorithme de génération

Le principe est le suivant :

- On s'intéresse aux règles valides r telles que $support(r) = support(p_2) \geq \sigma_S$

1. On considère les règles de la forme $p_1 \rightarrow p_2 \backslash p_1$ où la conclusion est de longueur 1
2. On élague les règles non valides
3. On combine les conclusions des règles valides
4. Puis on passe à des conclusions de longueur 2 et on itère (longueurs 3, 4, etc. . .)