Projet de Fouille de Données

Algorithme CLOSE

Projet réalisé par

- MEDDAH Amine

Sommaire

1 Description du projet

Description du projet

L'objectif du projet est d'implémenter l'algorithme CLOSE dans le contexte du module de Fouille de données.

On doit réaliser une application interfacé qui permet d'extraire un ensemble de règles à partir d'un jeu de données formaté.

L'algorithme CLOSE repose sur le principe de construction de « générateurs » à partir d'un ensemble d'items initiaux en fonction d'un support minimal. Ensuit il va créer les fermés de chaque générateur en se basant sur le nombre d'occurrence de chaque élément du générateur dans l'ensemble d'items initiaux. Il ne garde que les fermés qui ont un support supérieure au support minimal définit au par avant.

On réitère ces opérations pour faire ressortir tout les fermés jusqu'a ce qu'il n'ait plus de fermeture à générer.

On finit par données les règles exactes (confiance = 1) et les règles approximatives (support >= support minimal).

2 Méthodes de développement

Méthodes de développement

1. Description de l'architecture

Notre application est divisée en plusieurs packages selon le traitement.

On distingue quatre packages qui contiennent les classes utilisées en l'occurrence :

- handler : qui est constitué de la classe « CloseHandler » qui permet de mettre en place les différents concepts de l'algorithme CLOSE.

- model : qui est constitué des classes responsables du stockage des données et des règles. (« Element, Item, Rule, surEnsemble »)

- utils : qui contient la classe « Parser » qui est responsable du parsing du fichier de données pour le rendre interprétable par l'algorithme CLOSE.

- view : qui contient la vue de notre application.

2. Description des fonctions implémentées

Item : est une classe qui représente item et qui contient un nom qui lui correspond.

Element : est une classe qui représente une instance du résultat final.
Exepmle : a == > be, s = 0.3, c = 0.7, l = 0.4
Ou "a" l'item généré, "be" la fermeture, "s" le support, "c" la confiance et "l" le lift.

Rule : est la classe qui représente la règle d'association exacte ou approximative en utilisant une structure qui contient l'item de départ (gauche) et l'item d'arrivé (droite), le support et la confiance de la règle.
Ses règles sont construites à partir des d'une liste d'Element.

surEnsemble : est une classe qui représente les sur-ensembles.

Parser: cette classe contient une fonction principale qui permet de prendre en paramètre un fichier de données et de retourner une map qui représente chaque OID avec ses itèms.

CloseView : est la classe principale de notre projet car c'est de celle-ci que tout est lancé et c'est dans cette dernière qu'on récupère le résultat.

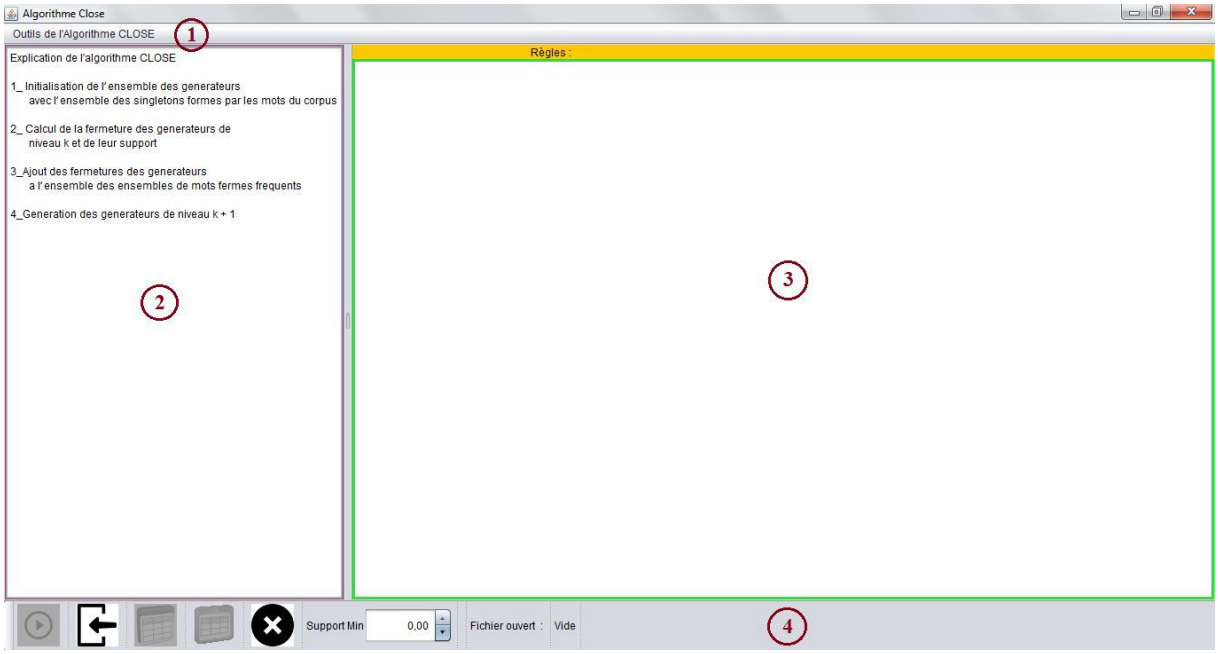
A partir d'une map d'item, elle construit les différentes règles en passant par plusieurs méthodes.

Ces méthodes sont :

- generateFirstFermeture() : cette méthode en premier lieu de générer les candidats, ensuit de calculer leurs fermeture en éliminant ceux qui ont un support inférieur au support minimal et les mettre dans une liste d'Element. (FF1)
- genereteElementFermeture() : cette méthode permet de calculer les fermeture jusqu'à ce qu'il n'y ait plus de générateurs. (FFk)
- generateNewGenerators() : cette méthode va combinant entre les différents candidats déjà existant pour donner de nouveaux candidats.
- generateSurEnsembles() : cette méthode permet de générer les sous ensembles d'un élément en vérifiant s'il est contenu dans les fermetures existantes.
- handelExactRules() : cette méthode permet d'extraire les règles de confiance égale à 1.
- handelApproxRules() : cette méthode permet d'extraire les règles de confiance inférieur à 1 en utilisant les sous ensembles.
- runClose() : cette méthode constitue la succession des étapes de l'algorithme CLOSE.

Guide d’utilisation

3.1. Présentation de l’interface de l’application



L’interface de notre application se compose d’ :

1. Une barre de menu avec la quelle on peut : lancer l’algorithme CLOSE, importer un fichier, afficher les règles approximatives et exactes dans des tableaux et effacer le contenu des tableaux et du TextArea qui affiche le résultat d’exécution de l’algorithme.

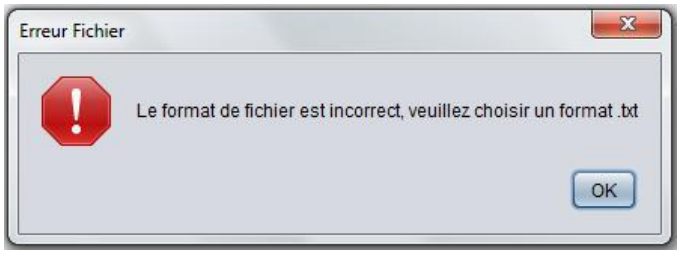
On remarquera que chaque action peut être réalisée avec l'accélérateur qui lui correspond.

2. Une zone qui résumer les étapes de l'algorithme CLOSE.
3. Une zone qui permet d'afficher le résultat d'exécution de l'algorithme CLOSE.
4. Une barre d'outil qui reprend se que contient la barre de menu ainsi qu'un Spinner pour donner le support minimal qui doit être compris entre 0 et 1 et l'affichage du fichier qui est ouvert sur l'application (Vide si il n'ya pas de fichier importé).

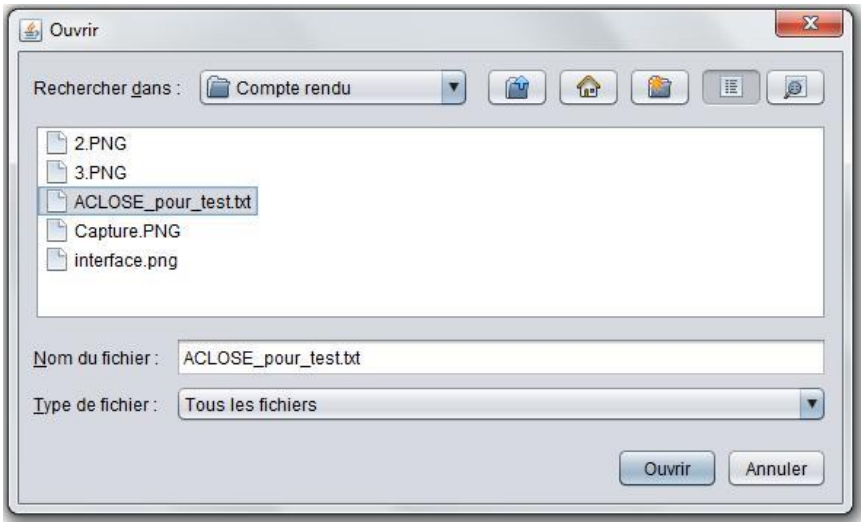
3.2. Présentation des fonctionnalités de l'application

On peut voir qu'au lancement de l'application, le bouton d'exécution de l'algorithme ainsi que les deux boutons d'affichage des tableaux de règles approximative et surs sont grisés car on n'a pas encore importé de fichier.

Lors de l'import d'un fichier, on vérifie que c'est bien un fichier .txt, sinon un message d'erreur serra affiché.



Après l'import d'un fichier, on pourra voir que sur la barre d'outils le fichier est bien ouvert.



Avant de lancer l'algorithme, on choisit un support minimal. On récupère ensuite le résultat d'exécution sur la partie central de l'interface.

Règles :

Règles d'association exactes

ab	->	ce	(s=0.33, c=1.0, l=3.0)
a	->	c	(s=0.5, c=1.0, l=2.0)
d	->	ac	(s=0.17, c=1.0, l=6.0)
e	->	b	(s=0.83, c=1.0, l=1.2)
ce	->	b	(s=0.67, c=1.0, l=1.5)
b	->	e	(s=0.83, c=1.0, l=1.2)
bc	->	e	(s=0.67, c=1.0, l=1.5)
ae	->	bc	(s=0.33, c=1.0, l=3.0)

Règles d'association approximatives

a	->	cd	(s=0.17, c=0.33, l=2.0)
a	->	bce	(s=0.33, c=0.67, l=2.0)
e	->	abc	(s=0.33, c=0.4, l=1.2)
e	->	bc	(s=0.67, c=0.8, l=1.2)
ce	->	ab	(s=0.33, c=0.5, l=1.5)
b	->	ace	(s=0.33, c=0.4, l=1.2)
b	->	ce	(s=0.67, c=0.8, l=1.2)
bc	->	ae	(s=0.33, c=0.5, l=1.5)
c	->	ad	(s=0.17, c=0.2, l=1.2)
c	->	abe	(s=0.33, c=0.4, l=1.2)
c	->	a	(s=0.5, c=0.6, l=1.2)
c	->	be	(s=0.67, c=0.8, l=1.2)

****

Pour plus de lisibilité, on pourra afficher chaque tableau de règles et tiré son contenu par support, confiance, lift, règle droite ou gauche.

Regles approximatives

Regle Gauche	Regle droite	Support	Confiance	Lift
a	cd	0.17	0.33	2.0
c	ad	0.17	0.2	1.2
a	bce	0.33	0.67	2.0
e	abc	0.33	0.4	1.2
ce	ab	0.33	0.5	1.5
b	ace	0.33	0.4	1.2
bc	ae	0.33	0.5	1.5
c	abe	0.33	0.4	1.2
c	a	0.5	0.6	1.2
e	bc	0.67	0.8	1.2
b	ce	0.67	0.8	1.2
c	be	0.67	0.8	1.2

Regles Surs

Regle Gauche	Regle droite	Support	Confiance	Lift
d	ac	0.17	1.0	6.0
ab	ce	0.33	1.0	3.0
ae	bc	0.33	1.0	3.0
a	c	0.5	1.0	2.0
ce	b	0.67	1.0	1.5
bc	e	0.67	1.0	1.5
e	b	0.83	1.0	1.2
b	e	0.83	1.0	1.2

Ici, on a trié les deux tableaux par rapport au support dans un ordre décroissant. On aurait pu le faire avec les autres paramètres en double cliquant sur la colonne voulue.

Jeu d’exemple

Au court de ce paragraphe, nous allons vous présentez deux exemples d’exécution.

Pour valider le bon fonctionnement de notre application, nous avons lancé le jeu d’exemple vu en cours.

Nous rappelons l’exemple : (support minimal = 2/6)

OID	Items
1	a | c | d
2	b | c | e
3	a | b | c | e
4	b | e
5	a | b | c | e
6	b | c | e

Règles :

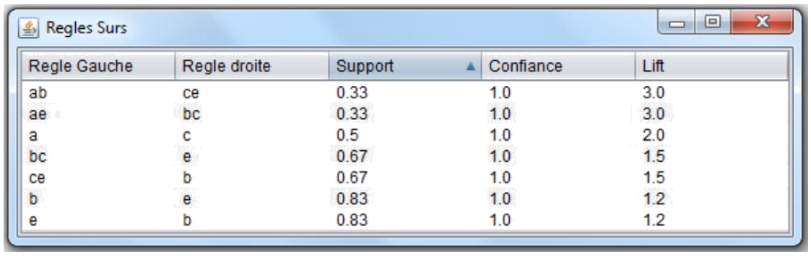
Règles d'association exactes

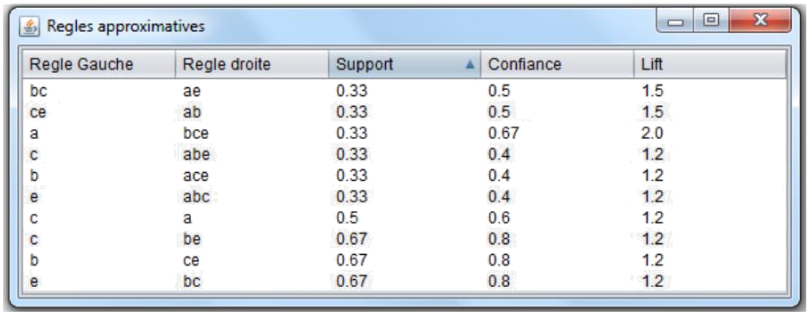
ab	->	ce	(s=0.33, c=1.0, l=3.0)
bc	->	e	(s=0.67, c=1.0, l=1.5)
ce	->	b	(s=0.67, c=1.0, l=1.5)
a	->	c	(s=0.5, c=1.0, l=2.0)
b	->	e	(s=0.83, c=1.0, l=1.2)
ae	->	bc	(s=0.33, c=1.0, l=3.0)
e	->	b	(s=0.83, c=1.0, l=1.2)

Règles d'association approximatives

bc	->	ae	(s=0.33, c=0.5, l=1.5)
ce	->	ab	(s=0.33, c=0.5, l=1.5)
a	->	bce	(s=0.33, c=0.67, l=2.0)
c	->	abe	(s=0.33, c=0.4, l=1.2)
c	->	be	(s=0.67, c=0.8, l=1.2)
c	->	a	(s=0.5, c=0.6, l=1.2)
b	->	ace	(s=0.33, c=0.4, l=1.2)
b	->	ce	(s=0.67, c=0.8, l=1.2)
e	->	abc	(s=0.33, c=0.4, l=1.2)
e	->	bc	(s=0.67, c=0.8, l=1.2)

****





On peut constater le bon fonctionnement de l'application due au faite qu'elle nous a donnée le résultat attendu.

Pour notre second d'exemple, nous avons pris un fichier plus conséquent qui contient plus de 2400 lignes et des milliers d'items.

Notre application a réussit à nous donner de bons résultats, notamment en temps d'exécution et en termes d'extraction de règles.

Tout au long du développement de l'application, nous avons sans cesse optimisé les actions réalisés par l'algorithme Close, que sa soit en utilisant des méthodes de hachages, ou en regroupant des actions qui étaient similaire.

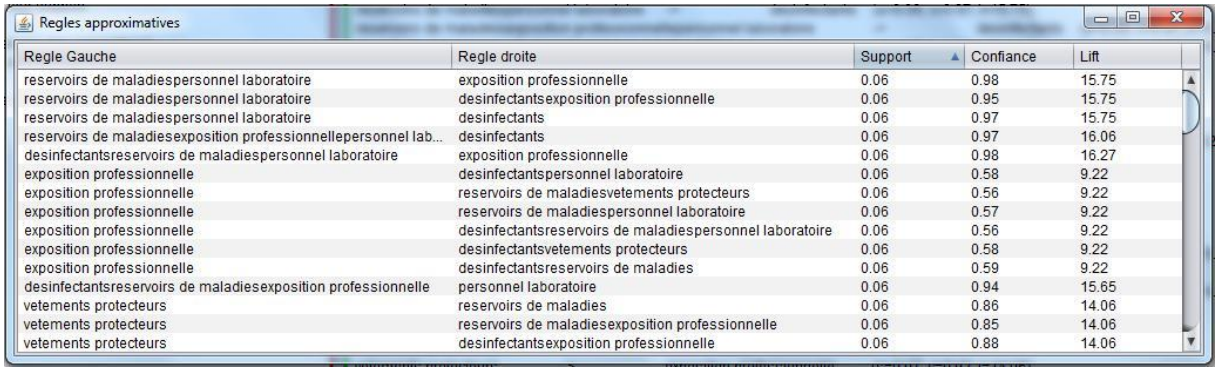
Nous savant que le domaine de fouille de données travail sur des données de masse, c'est pour cela que l'optimisation était pour nous un aspect qui devait être prioritaire.

Nous avons pu avoir des extractions de règles avec un support minimal qui se situe entre 0.03 et 0.07.

Règles extraites avec un support minimal de 0.07 :

Avec un temps d'exécution de 2s.

Règles extraites avec un support minimal de 0.06 :



Regle Gauche	Regle droite	Support	Confiance	Lift
reservoirs de maladiespersonnel laboratoire	exposition professionnelle	0.06	0.98	15.75
reservoirs de maladiespersonnel laboratoire	desinfectantsexposition professionnelle	0.06	0.95	15.75
reservoirs de maladiespersonnel laboratoire	desinfectants	0.06	0.97	15.75
reservoirs de maladiesexposition professionnellepersonnel lab...	desinfectants	0.06	0.97	16.06
desinfectantsreservoirs de maladiespersonnel laboratoire	exposition professionnelle	0.06	0.98	16.27
exposition professionnelle	desinfectantspersonnel laboratoire	0.06	0.58	9.22
exposition professionnelle	reservoirs de maladiesvetements protecteurs	0.06	0.56	9.22
exposition professionnelle	reservoirs de maladiespersonnel laboratoire	0.06	0.57	9.22
exposition professionnelle	desinfectantsreservoirs de maladiespersonnel laboratoire	0.06	0.56	9.22
exposition professionnelle	desinfectantsvetements protecteurs	0.06	0.58	9.22
exposition professionnelle	desinfectantsreservoirs de maladies	0.06	0.59	9.22
desinfectantsreservoirs de maladiesexposition professionnelle	personnel laboratoire	0.06	0.94	15.65
vetements protecteurs	reservoirs de maladies	0.06	0.86	14.06
vetements protecteurs	reservoirs de maladiesexposition professionnelle	0.06	0.85	14.06
vetements protecteurs	desinfectantsexposition professionnelle	0.06	0.88	14.06

On a pu extraire 80 règles en 3s.

Règles extraites avec un support minimal de 0.05 :
On a pu extraire 256 règles en 6s.

Règles extraites avec un support minimal de 0.04 :
On a pu extraire 260 règles en 8s.

Règles extraites avec un support minimal de 0.03 :
On a pu extraire 4091 règles en 38s.

Au dessous de 0.03 pour le support minimal, l'algorithme CLOSE génère beaucoup trop de candidats et de fermeture et les calculs deviennent très couteux en ressources (exponentiel).