Calcul distribué avec Spark
TP2 : Frequent Item Set

Olivier Schwander <olivier.schwander@sorbonne-universite.fr>

2021-2022

Exercice 1 *Introduction*

Étant donné un ensemble de *transactions* (par exemple la liste des achats d'un client), le problème *Frequent Item Set* cherche les sous-ensembles qui apparaissent fréquemment (selon un seuil fixé). L'objectif est de pouvoir *recommander* à de futurs clients des articles fréquemment achetés avec ceux déjà choisis.

Par exemple, pour un seuil fixé à 3 et l'ensemble de transactions suivant :

— {1, 2, 3, 4}
— {1, 2, 4}
— {1, 2}
— {2, 3, 4}
— {2, 3}
— {3, 4}
— {2, 4}

En comptant les sous-ensembles de taille 1, on obtient :

Sous-ensemble	Taille	
{1}	3	≥ 3
{2}	6	≥ 3
{3}	4	≥ 3
{4}	5	≥ 3

Tous ces sous-ensembles sont donc fréquents.

Pour les ensembles de taille 2, on obtient :

Sous-ensemble	Taille	
{1,2}	3	≥ 3
{1,3}	1	
{1,4}	2	
{2,3}	3	≥ 3
{2,4}	4	≥ 3
{3,4}	3	≥ 3

Les sous-ensembles {1,2}, {2,3}, {2,4}, {3,4} sont donc fréquents.

Question 1
Combien d'opération de comptage doit-on effectuer avec cette méthode naïve ?

Exercice 2 *Algorithme A Priori*

L'algorithme A Priori repose sur la remarque suivante : tous les sous-ensembles d'un sous-ensemble fréquent sont fréquents. On a donc une condition nécessaire pour qu'un sous-ensemble soit fréquent, ce qui permet d'élaguer l'ensemble des sous-ensembles à compter. L'algorithme est le suivant :

- **Initialisation** : les candidats sont tous les sous-ensembles de taille $k = 1$;
- **Étape 1** : élaguer les candidats qui ne sont pas fréquents ;
- **Étape 2** : construire les sous-ensembles de taille $k + 1$ à partir des candidats restants ;
- **Répéter** les deux étapes tant que tous les sous-ensembles n'ont pas été étudiés.

Question 1
Expliquer pour cette méthode n'est pas sujet au même problème que la méthode naïve ? Sous quelle hypothèse raisonnable est-ce le cas ?

Question 2
Expliquer pourquoi cette méthode n'est pas adaptée au cadre MapReduce.

Exercice 3 *Spark*

On travaillera sur les données du fichier http://www-connex.lip6.fr/~schwander/enseignement/m2stat_gd/T10I4D100K.dat.

Question 1
Écrire une transformation pour charger et interpréter les données d'entrée.

Question 2
L'idée clé de la version Map-Reduce est de construire la liste des sous-ensembles indépendemment pour chaque transaction.

Expliquer pour cette liste de sous-ensembles d'une transaction est faisable en pratique.

Question 3
Écrire une transformation pour réaliser cette liste de sous-ensembles.

Question 4
Écrire une transformation qui réalise la réduction finale.