TP N° 2 : Spark MLLib

- Moteur de Recommandation -

L'objectif de ce TP est la recommandation de films. La base de données utilisée est MovieLens. La base de données MovieLens 1m est une ensemble de 1 000 000 de points représentant les notes données par un ensemble d'utilisateurs à un ensemble de films. Elle contient aussi des méta-données sur les films et les profils utilisateurs.

Les systèmes de recommandation se basent sur :

- des approches basées sur le contenu (Content-based approach)
- le filtrage collaboratif (collaborative filtering)

Filtrage collaboratif

Le filtrage collaboratif est souvent utilisé dans les systèmes de recommandation. Ces techniques ont pour but de déterminer les entrées manquantes dans une matrice d'association Utilisateur - produit, en l'occurrence une matrice de notation utilisateur - film. MLlib implémente le filtarge collaboratif basé sur les modèles (model-based collaborative filtering), dans lequel les utilisateurs et les produits sont décrits par un petit ensemble de facteurs latents qui sont utilisés pour prédire les entrées manquantes. En particulier, nous utiliserons l'algorithme des moindres carrés alternés ALS (alternating least squares) pour apprendre ces facteurs latents.

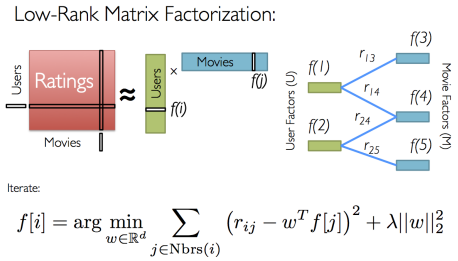

FIGURE 1 – Low Rank Matrix Factorization

Base de données

1) Télécharger et décompresser la base de données MovieLens suivante :
   http://files.grouplens.org/datasets/movielens/ml-1m.zip
   Nous utiliserons les 2 fichiers : "ratings.dat" et "movies.dat". Les notes sont contenues dans le fichier "ratings.dat" et sont sous format :
   UserID : :MovieID : :Rating : :Timestamp
   Les informations concernant les films sont dans le fichier "movies.dat" et sont sous format :
   MovieID : :Title : :Genres

Préparation des données

2) A partir du fichier ratings.dat, créer une RDD de notes (ratings) qui contient les paires (id_film, Rating). Ne laisser que le dernier chiffre du timestamp comme clé aléatoire. Utiliser la classe *Rating* définie dans le package *org.apache.spark.mllib.recommendation*

3) A partir du fichier movies.dat, Construire une correspondance (id_film,titre)

4) Déterminer le nombre de films, le nombre d'utilisateurs et le nombre de ratings.

Recommandation avec ALS

Nous utiliserons la fonction ALS de MLlib pour entrainer un modèle de factorisation de matrice (MatrixFactorizationModel), qui prend en entrée la RDD[Rating].
ALS ayant des paramètres tels que le rang (rank) ou les constantes de régularisation, plusieurs valeurs de ces paramètres doivent être entrainées sur un ensemble de validation.

5) Nous commençons, donc, par partager les données en trois ensembles disjoints : apprentissage, validation et test. Vous pouvez utiliser le dernier chiffre du timestamp pour faire ce partage.

6) Utiliser ALS.train pour entrainer un certain nombre de modèles (en variant les variables rank, lambda (constante de régularisation), et le nombre d'itérations). Calculer le RMSE pour chaque modèle sur l'ensemble de validation. Le modèle avec le plus petit RMSE sera le modèle retenu. Son RMSE sur l'ensemble de test sera la métrique finale.

7) Donner les paramètres rank et lambda de votre meilleur modèle ainsi que son RMSE sur l'ensemble de test.