Bases de données NoSQL

Spark MLLib

Ons Jelassi, ons.jelassi@telecom-paristech.fr

Telecom Evolution, Paris, France

Sommaire

Sommaire

MLLib

- Sous-projet Spark fournissant des primitives de Machine Learning
- Initialisé par AMPLab, UC Berkeley
- Fourni avec Spark depuis la version 0.8

Algorithmes

- **Classification** : régression logistique, SVM linéaire, naive Bayes
- **Régression** : GLM (Generalized Linear Regression)
- **Filtrage collaboratif** : ALS
- **Clustering** : k-means
- **Décomposition** : SVD, PCA

Pourquoi MLLib

- Construit sur Apache Spark
- Composant standard de Spark fournissant des primitives de machine learning
- Avantages
  - Passage à l'échelle
  - Performance
  - API
  - Intégration avec Spark et ses autres composants (Spark Streaming, GraphX, Spark SQL)

Passage MLLib ML

- Algorithmes pouvant être entrainés par train() ou run()
- Nécessité de mapping des données RDD[Vector] ou RDD[LabeledPoint]
- Différences par rapport à SkLearn (qui prend des Array Numpy ou DataFrames Pandas)
- Nouvelle API ML, incorporant les pipelines et la cross-validation
- Algorithmes prenant en entrée des DataFrames

Sommaire

Recommandation

- Méthodes adaptatives : basées sur l'historique de navigation, le flux de clics
- Méthodes basées sur le contenu : historique des comportements des clients, informations complémentaires sur le profil
  - accès nécessaire à des descriptions d'articles
  - difficulté à traiter les nouveaux utilisateurs, à extrapoler d'un domaine à un autre
- Filtrage collaboratif : connaissance uniquement des interactions ou avis clients*produits
  - données représentées sous forme matricielle, 1 ligne par utilisateur et 1 colonne par produit
  - aucune connaissance nécessaire des articles
- Méthodes hybrides

Filtrage collaboratif

- Popularisé par le concours Netflix
- Matrice très creuse clients*films de notes entre 1 et 5
- Objectif : prévoir la note d'un client pour un produit qu'il n'a pas acheté (intéressé par les valeurs élevées)
- Utilisé par FNAC, Amazon, etc.

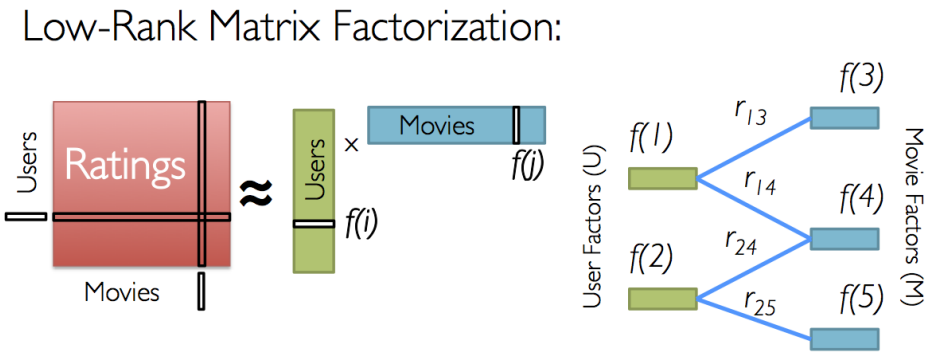
Méthodes de voisinage/Memory-based methods

- Fondées sur les indices de similarité entre clients ou produits (corrélation linéaire, distance cosinus, etc.)
- Hypothèse que les clients qui ont des préférences similaires vont apprécier les produits de façon similaire
  - Trouver un sous-ensemble S_i de clients qui notent de manière similaire au client i et calculer la combinaison linéaire de leurs notes pour le produit j
- Hypothèse que les clients préfèrent des produits similaires à ceux qu'ils ont déjà notés
  - Trouver un sous-ensemble S_j de produits notés de façon similaire par le client i et retrouver la note manquante par combinaison linéaire des notes

Modèles à facteurs latents

- Basés sur une décomposition de faible rang de la matrice très creuse avec une éventuelle contrainte de régularisation
- La note du client i sur le produit j est approchée par le produit scalaire de 2 facteurs :
  - la représentation réduite d'un utilisateur
  - la représentation réduite d'un produit
- Les articles et les utilisateurs sont décrits par des vecteurs de même dimension, donnée par le nombre de facteurs latents
- Chaque utilisateur est décrit par les contributions de ces facteurs latents à la note qu'il donnerait à un article

Alternating Least Squares



Iterate:

$$f[i] = \arg\min_{w \in \mathbb{R}^d} \sum_{j \in \text{Nbrs}(i)} \left(r_{ij} - w^T f[j]\right)^2 + \lambda ||w||_2^2$$

FIGURE: Low Rank Matrix Factorization

Objectif du TP

Utiliser la méthode ALS de MLLib pour construire un moteur de recommandation de films sur la base MovieLens