M2 BDIA — MADM

Janvier 2018

TD/TP • **Détection de communautés**

Les expériences sont à réaliser avec R et la package igraph.

Exercice 1. Comparaison des algorithmes de détection de communauté

1. À partir des données de Game Of Thrones, utiliser les différents algorithmes de graphe pour détecter des communautés. Trier les algorithmes par famille, comparer les résultats obtenus.
2. Comment interpréter les résultats ?
3. Pour les algorithmes utilisant une mesure de la modularité, calculer le seuil de résolution et en déduire les communautés non exploitables.
4. Résumer quelques phrases les différences entre les algorithmes de graphe.
5. Appliquer un clustering hiérarchique ascendant, tester différentes mesures de distance. Comparer avec les algorithmes de graphe. Utiliser les packages `hclust` et `linkcomm`. Pour `linkcomm` en plus de la documentation R du package, lire le document `https://cran.r-project.org/web/packages/linkcomm/vignettes/linkcomm.pdf`.
6. Répondre à la même question pour k-means.
7. Dans toutes les expériences réalisées, comment faites vous pour évaluer la qualité du résultat ?

Exercice 2. Vers le traitement des grands graphes

1. Télécharger Apache Spark (`https://spark.apache.org/downloads.html`) et le décompresser dans votre répertoire. Fixer les variables pour le JDK 1.8. Lancer le shell Spark et charger un fichier CSV (cf documentation de Spark). Nous utiliserons la bibliothèque GraphX de Spark. GraphX étend l'abstraction RDD (*resilient data set*) de Spark en introduisant la notion de graphes de propriétés répartis résilients. Il s'agit un multigraphe dirigé avec des propriétés attachées à chaque sommet et lien. Pour prendre en charge les analyses sur ces graphes, GraphX propose un ensemble d'opérateurs fondamentaux (par exemple, `subgraph`, `joinVertices`, `mapReduceTriplets`, etc.) ainsi qu'une variante optimisée de l'API Pregel.

   Les algorithmes utilisés pour les analyses nécessitent généralement des mouvements de données en dehors de la topologie du graphe et s'expriment souvent plus naturellement en tant qu'opérations sur des tables. De plus, la manière dont les algorithmes utilisent les données dépend des objectifs et ainsi les mêmes données brutes (raw-data) peuvent nécessiter de nombreuses transformation, vues de type graphes ou tables tout au long du processus d'analyse.

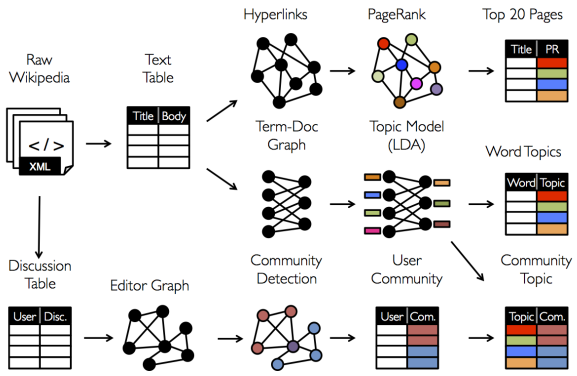

FIGURE 1 – Données brutes et différents modèles pour les analyses (d'après la documentation de GraphX)

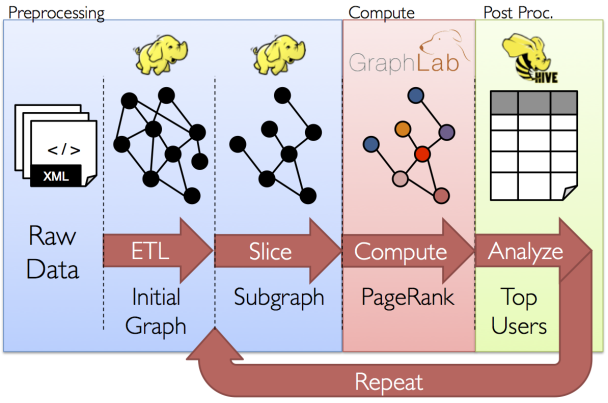

FIGURE 2 – Pipeline simplifié d'analyse de graphe (d'après la documentation de GraphX)

2. Charger les données Game Of Thrones, établir deux RDD pour les sommets et les arcs, les assembler dans une graphe puis calculer le degré des sommets et la répartition de degrés (histogramme par exemple). Compter le nombre de triangles.
3. On souhaite effectuer une analyse sur les données Wikipedia. Transformer les liens et articles en un graphe (cf LSA sur Wikipedia).
4. Dans graphX choisir deux algorithmes de détection de communautés et les appliquer sur le graphe Wikipedia. Conclure.