Projets à rendre

L'évaluation se fera sous la forme de deux projets :

- Le projet 1 est à rendre pour le 15/12/2023
- Le projet 2 est à rendre pour le 05/12/2023. Une présentation orale aura lieu le 06/12 durant la séance de cours

Projet 1 : Le calcul du PageRank

C'est l'exemple par lequel Google a mis en place son infrastructure.

On a un réseau de N pages qui se citent ; `i→j` traduit le fait que la page i cite la page j.

Principe de l'algorithme du PageRank :

- Initialement toutes les pages sont affectées d'un même poids $w_i^0 = \dfrac{1}{N}$.
- Pour chaque sommet i, on calcule le nombre n_i de liens `i→j` .
- Ensuite, dans un premier mouvement, on affecte chaque sommet i d'un nouveau poids :

$$w_i^1 = cw_i^0 + (1-c)\sum_{j\to i}\frac{w_j^0}{n_j}$$

où c est un coefficient qui est souvent pris égal à 0.15 pour des raisons heuristiques.

  - On itère le processus une dizaine de fois (là encore, le choix du nombre d'itération est essentiellement heuristique) :

$$w_i^{k+1} = cw_i^0 + (1-c)\sum_{j\to i}\frac{w_j^k}{n_j}$$

Le vecteur des poids des sommets tend vers une valeur limite qu'on appelle le page rank de chaque sommet.

Comment traduire cela en algorithme map-reduce ?

Voici un pseudo-code extrait de **Jimmy Lin and Chris Dyer** - Data-Intensive Text Processing with MapReduce (https://lintool.github.io/MapReduceAlgorithms/MapReduce-book-final.pdf).

ALGORITHM

```
Class Mapper
    method Map(nid n, node N )
        p <- N.PageRank/|N.AdjacencyList|
        Emit(nid n, N )   # Pass along graph structure
        for all nodeid m in N.AdjacencyList do
            Emit(nid m, p) # Pass PageRank mass to neighbors

Class Reducer
    method Reduce(nid m, [p1 , p2 ,...])
        M <- empty
        for all p in counts [p1 , p2 ,...] do
            if IsNode(p) then
                M <- p  # Recover graph structure
            else
                s <- s + p # Sum incoming PageRank contributions
        M.PageRank <- s
        Emit(nid m, node M )
```

Ce code n'est pas entièrement complet puisqu'ici $c = 0$.

Exercice

On part d'un fichier (https://math.univ-angers.fr/~badreau/data/soc-Epinions1.txt) décrivant un réseau de page web, sous la forme :

```shell
123 2546
258 25548
```

Chaque ligne correspond à un lien web, le premier champ est le numéro de la page source et le deuxième celui de la page cible (la page citée).

Ecrire avec `mrjob` un programme qui prend en entrée un tel fichier et rend les valeurs des pagerank des différentes pages, avec éventuellement un classement par ordre décroissant.

On utilisera la valeur du coefficient $c = 0.15$ ainsi que 10 itérations dans un premier temps. On pourra aussi tester avec d'autres valeurs et voir la différence.

Ce projet peut se faire en binôme. Il est attendu un script python clair et commenté (un seul script par groupe) utilisant le module `mrjob`, éventuellement associé à un court fichier texte d'explication du code (si vous le jugez nécessaire). Le script doit pouvoir être exécuté par la commande unix

```shell
python nom1_nom2.py soc-Epinions1.txt
```

et doit renvoyer le calcul du PageRank des liens web du fichier (avec éventuellement un classement par ordre décroissant).

Projet 2 : Accidents de la circulation en France

Les accidents de la circulation en France sont recensés chaque année. Pour une année `xxxx`, on a 4 fichiers appelés `caracteristiques_xxxx.csv`, `lieux_xxxx.csv`, `usagers_xxxx.csv` et `vehicules_xxxx.csv`.

Les fichiers correspondant aux années 2011 à 2018 sont disponibles dans cette archive (https://math.univ-angers.fr/~badreau/data/accidents.zip). Un document complet disponible ici (https://math.univ-angers.fr/~badreau/data/description-des-bases-de-donnees-onisr-annees-2005-a-2018.pdf) décrit ces fichiers de données.

- Le fichier `caracteristiques_xxxx.csv` contient un enregistrement par accident pour l'année `xxxx` :

Num_Acc	an	mois	jour	hrmn	lum	...
201800000001	18	1	24	1505	1	...
201800000002	18	2	12	1015	1	...
201800000003	18	3	4	1135	1	...

`lum` décrit par exemple les conditions d'éclairage dans lesquelles l'accident s'est produit :

1. Plein jour,
2. Crépuscule ou aube,
3. Nuit sans éclairage public,
4. Nuit avec éclairage public non allumé,
5. Nuit avec éclairage public allumé.

- Le fichier `vehicules_xxxx.csv` contient un enregistrement par véhicule mis en cause dans un accident pour l'année `xxxx` :

Num_Acc	senc	catv	occutc	obs	obsm	choc	manv	num_veh
201400000001	0	33	0	0	2	1	1	A01
201400000001	0	7	0	0	0	6	15	B01

`obs` = obstacle fixe heurté : 0 : rien, 1 : véhicule , 2 : arbre, etc.

`obsm` = obstacle mobile : 0 : rien, 1 : piéton, 2 : véhicule, etc.

- Le fichier `usagers_xxxx.csv` contient un enregistrement par usager mis en cause dans un accident pour l'année `xxxx` :

Num_Acc	place	catu	grav	sexe	trajet	secu	locp	actp	etatp	an_nais	num_veh
201400000001	1	1	3	1	5	21	0	0	0	1971	A01
201400000001	1	1	1	1	5	11	0	0	0	1992	B02

`catu` désigne la catégorie d'usager : 1 : conducteur, 2: passager, 3 piéton

`grav` désigne la gravité de l'accident : 1 : indemne, 2 : tué, 3: blessé hospitalisé, 4 : blessé léger.

- Le fichier `usagers_xxxx.csv` contient les descriptifs des lieux d'accidents ayant eu lieu l'année `xxxx`

Exercice

En tant que futur(e) data-scientist(e)s, on vous demande d'examiner ces données et de faire ressortir quelques pistes de réflexion en vue de réduire le nombre d'accidents. Les informations retenues devront être utiles et précises (focalisez-vous par exemple sur un type de véhicule ou d'intersection, certaines conditions extérieures, le type de blessés, ...). L'objectif est de préparer un exposé d'une dizaine de minutes durant lequel vous présenterez le résultat de vos recherches, d'éventuelles propositions d'amélioration, et conclurez par une explication plus théorique du fonctionnement de votre programme.

Toute prise d'initiative sera récompensée (graphique original, information pertinente, utilisation poussée de spark voire du Machine Learning via spark, etc.)

Ce travail se fera de manière individuelle. Il est attendu un script python clair et commenté utilisant obligatoirement le framework Spark (par l'intermédiaire du module `pyspark`), éventuellement associé à un court fichier texte d'explication du code (si vous le jugez nécessaire).

La note reposera principalement sur la présentation orale de votre programme et sur la mise en oeuvre des concepts étudiés ensemble (même si la quantité de données à votre disposition reste largement raisonnable, il est attendu que votre programme puisse permettre le passage à une échelle plus importante). Le respect du temps imparti ainsi que la clarté de l'exposé seront également évalués.

Index (https://math.univ-angers.fr/~badreau/)

Last updated 2023-10-11 13:12:14 +0200