INF8111

Fouille de données

Été 2019

Examen final - Partie II

Enseignant: Daniel Aloise

Cet examen comporte 5 questions

Pondération: 50% de la note de l'examen final

Directives:
Un aide-mémoire recto verso de format 8 1/2 X 11 est permis à l'examen.
Calculatrice permise.
Vous remettez seulement le cahier d'examen.
Durée de l'examen: 2 heures

1 Question 1 (2 points)

Un analyste applique un algorithme de détection de données aberrantes (*outliers*) sur un ensemble de données. Étant prudent, avant d'annoncer ses résultats, il décide d'appliquer à nouveau l'algorithme de détection, cette fois-ci uniquement sur l'ensemble des données aberrantes identifiées précédemment. Finalement, l'analyste conclut que les données aberrantes sont celles identifiées par l'algorithme à la fin de cette démarche.

(a) Discutez les résultats obtenus par l'analyste dans le cas où il utilise pour chaque application de son algorithme un modèle de détection de données aberrantes par clustering. En plus, discutez sur les résultats obtenus lorsqu'il applique à la place un modèle basé sur des distances.

Les modèles par clustering ne marcheraient pas puisque la distribution de données change complètement entre la première et la deuxième application de l'algorithme. Fait intéressant, les approches basées sur des distances continueraient à bien détecter les mêmes données aberrantes, du moins si les hyperparamètres d'origine ont été conservés.

(b) À votre avis, est-ce que l'approche de l'analyste est raisonnable pour identifier les données aberrantes de l'ensemble de données au complet?

Si une donnée aberrante dépend de l'ensemble complet de données, il est donc probablement déraisonnable de s'attendre à ce qu'un algorithme de détection de données aberrantes identifie un ensemble de telles données en tant que tel en l'absence d'une partie considérable de l'ensemble de données d'origine.

2 Question 2 (2 points)

L'algorithme de k-moyennes (*k-means*) vu en classe cherche à regrouper un ensemble de points donnés en k clusters.

L'algorithme peut-être écrit comme:

Sélectionnez k moyennes initiales $\mu_1, \ldots, \mu_k$;
while *convergence pas atteinte* **do**
  Pour chaque point $x \in X$, attribuez x à sa moyenne la plus proche, c.a.d., attribuez x au cluster C_{k^*}, où $k^* = argmin_{j=1,\ldots,k}\{\|x - \mu_j\|^2\}$;
  Pour $j = 1, \ldots, k$, calculez μ_j comme le centroïde des points appartenant à C_j;
end

Algorithm 1: k-means

Concevez les fonctions `map` et `reduce` concernant **une seule itération** de l'algorithme de k moyennes.

Important:

- Assumez que toutes les machines de la grappe distribuée possèdent l'ensemble des moyennes de l'itération courante.

- chaque fonction `map` doit traiter un point à la fois.
- L'objectif des fonctions `map` et `reduce` est de calculer les moyennes pour l'itération suivante de l'algorithme k-means.

La solution présenté ci-dessous n'est pas dans un langage valide. Elle présente plutôt la logique des fonctions map et reduce demandées.

```
map(x):
  emit (argmin_k \{\|x - \mu_k \}, x)

reduce(k, iterator points):
      sum=0;
      count=0
      while(points.hasNext()):
          sum += *points;
          count++;
          points = points.next();
      end while
      emit(k,(sum/count))
```

3 Question 3 (2 points)

Considérez un filtre de Bloom pour $|S| = 1$ milliard de clés et qui utilise $k = 2$ fonctions hash.

(a) Combien de bits doivent être disponibles (i.e. $|B|$) afin de garantir une probabilité de faux positifs inférieure ou égale à 1%?

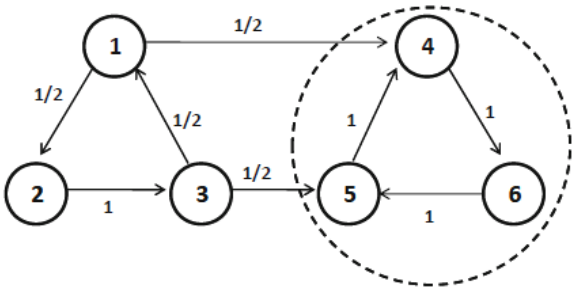
$$
\begin{aligned}
(1 - e^{-k|S|/|B|})^k &= \text{prob. de faux positifs} \\
(1 - e^{-2\times 1/|B|})^2 &= 10^{-2} \\
1 - e^{-2/|B|} &= 10^{-1} \\
e^{-2/|B|} &= 0.9 \\
\ln_e e^{-2/|B|} &= \ln_e 0.9 \\
\frac{-2}{n} &= -0.10 \\
n &\approx 19 \text{ milliards}
\end{aligned}
$$

(b) Dans le cas où $|B| = 2 \times |S|$, quel est le nombre idéal de fonctions hash qui minimise la probabilité de faux positifs?

$k = \frac{|B|}{|S|} ln(2) = 2ln(2) = 1.38 \approx 1$

4 Question 4 (2 points)

Considérez le graphe G ci-dessous. La matrice de transition M est trouvée directement à partir des valeurs sur les arêtes de G dans la figure.



(a) À l'aide de l'algorithme *PageRank*, calculez les rangs des sommets en utilisant $\beta = 0.9$ et $\beta = 0.8$. Initialisez avec $PR_0 = [1/6 \ldots 1/6]^T$ dans les deux cas.

Ne calculez que 4 itérations de *PageRank* pour chaque valeur de β.

Pour $\beta = 0.9$, les rangs finaux sont 0.0378, 0.0337, 0.0470, 0.2997, 0.2955, et 0.2864.

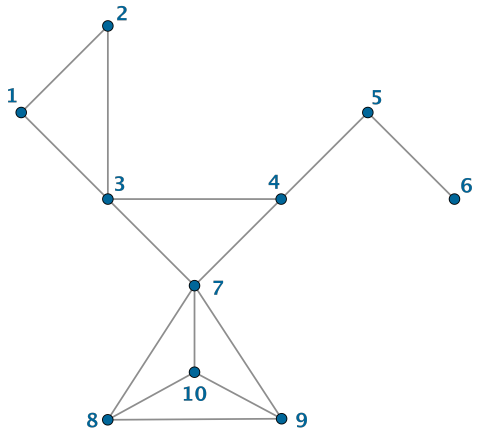
Pour $\beta = 0.8$, les probabilités sont 0.0657, 0.0596, 0.0810, 0.2737, 0.2676 et 0.2523.

(b) Que se passe-t-il aux rangs de différents sommets si on continue de diminuer la valeur de β? Les sommets 4, 5 et 6 représentent quelle structure problématique?

Les sommets 4,5 et 6 forment un *spider trap*. Du coup, au fur et à mesure qu'on diminue β nous augmentons les téléportations qui rendent possible aux internautes d'échapper de cette structure. En conséquence, les rangs des sommets du *spider-trap* diminuent.

5 Question 5 (2 points)

Obtenez pour le graphe ci-dessous:



(a) la valeur de son coefficient de regroupement.

$\eta_1 = \eta_2 = \eta_8 = \eta_9 = \eta_{10} = 1, \eta_3 = 2/6, \eta_4 = 1/3, \eta_5 = \eta_6 = 0, \eta_7 = 4/10$

$\frac{91}{15} \times \frac{1}{10} = \frac{91}{150} \approx 0.61$

(b) le sommet avec la plus grande valeur de la *centralité de dégré*.

sommet 7

(c) le sommet avec la plus grande valeur de la *centralité de proximité*.

sommet 7

(d) la partition obtenue pour $k = 2$ clusters en utilisant l'algorithme de Girvan-Newman.

L'algorithme de Girvan-Newman enlève les arêtes (3,7) et (4,7) dont la valeur de l'intermédiarité est la plus élevée.