LM206 - Initiation à `Scilab`

Projet final - sujet 1 (à rendre avant le 06/01/2014)

Un rapport et un seul fichier `.sci` (ou `.sce`) doivent être rendu. Tout document ou fichier rendu doit porter nom et prénom de l'étudiant. Tout exercice doit être séparé des autres et toute réponse commentée. La présentation et la lisibilité entreront pour une part importante dans l'appréciation des copies.

Partie 1 - Le jeu des anniversaires.

On se propose ici de réaliser un petit jeu interactif. L'idée est la suivante : `Scilab` génère une date d'anniversaire aléatoire et l'utilisateur doit deviner cette date. La date d'anniversaire donnée par `Scilab` est composée par un mois (un nombre entier entre 1 et 12) et un jour (attention : les nombres de jours varie selon le choix du mois, on considère que février a toujours 28 jours). Pour se faire aider, le joueur donne des propositions et `Scilab` lui répond selon les trois cas de figure suivants :

- le mois entré par l'utilisateur est différent de celui tiré par `Scilab`. Alors `Scilab` indique au joueur si le mois est supérieur ou inférieur à celui ciblé et invite le joueur à faire une nouvelle proposition de mois.
- le mois est correct, mais le jour entré par l'utilisateur est différent de celui tiré par `Scilab`. Alors `Scilab` indique au joueur si ce jour est supérieur ou inférieur à celui ciblé et invite le joueur à faire une nouvelle proposition de jour. Un nombre maximal de tentatives pour le choix du jour est fixé à 10, après ces 10 tentatives le joueur a perdu.
- la date (jour et mois) entrée par l'utilisateur est celle tirée par `Scilab`. Dans ce cas, `Scilab` félicite le joueur et lui précise en combien de propositions il a trouvé la solution (attention : on compte le nombre total de propositions relatives au mois et au jour).

Question. Ecrire un programme `Scilab` qui génère ce jeu en utilisant les outils de votre choix pour le dialogue entre `Scilab` et l'utilisateur.

Partie 2 - Google *PageRanking.*

Le moteur de recherche Google est le moteur de recherche sur le Web le plus utilisé au monde. L'efficacité de ses recherches est dû à l'utilisation d'un algorithme qui permet de classer les pages Web par ordre d'importance et de pertinence par rapport à la recherche effectuée. Cet algorithme, appelé *PageRanking*, a été introduit par le cofondateur de Google, Larry Page, en 1998.

Le principe du *PageRanking* se base sur l'attribution d'une valeur, dite *ranking*, à chaque page web qui en définit l'importance relative. Le Web est pensé comme un grand graphe où les pages Web sont liées les unes aux autres par des liens. Dans l'exemple de Figure 1, la flèche qui part du point D et pointe vers le point B veut dire que la page D contient un lien vers la page B. L'importance d'une page Web est proportionnelle au nombre de pages qui ont un lien vers la page même, divisé par leurs importances. Par exemple, la page C est plus importante que la page E même s'il existe un seul lien vers elle, car ce lien vient d'une page qui est beaucoup plus importante que les petites pages qui ont des liens vers la page E.

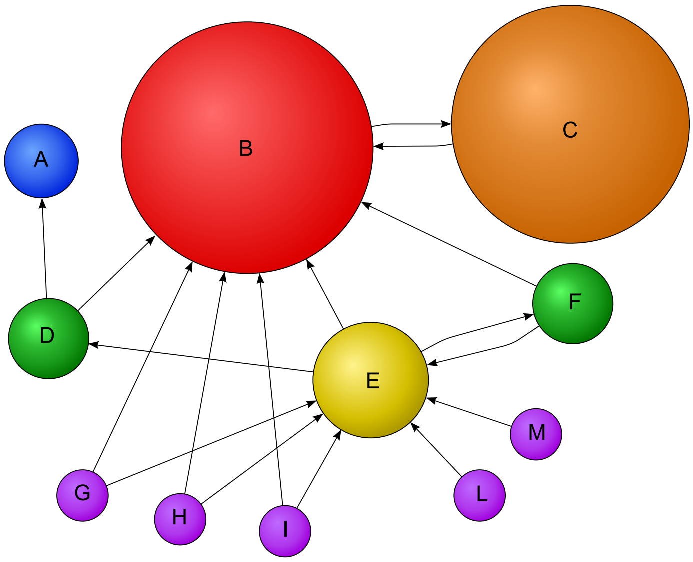

Figure 1: Exemple de graphe de *PageRanking*.

Sous un point de vue plus formel, le *ranking* de la page p est définie par

$$r(p) = \sum_{q \to p} \frac{r(q)}{N_q}$$

où $q \to p$ est l'ensemble de toutes les pages qui ont un lien vers la page p et N_q représente le nombre de liens présents dans la page q. Par exemple, le *ranking* de la page C est donné par

$$r(C) = \sum_{q \to C} \frac{r(q)}{N_q} = \frac{r(B)}{N_B} = r(B)$$

puisque la page C a une seule flèche qui pointe vers elle et qui provient de la page B, et la page B a un seul lien sortant ($N_B = 1$). Ou encore, le *ranking* de la page E sera donné par

$$r(E) = \sum_{q \to E} \frac{r(q)}{N_q} = \frac{r(F)}{2} + \frac{r(G)}{2} + \frac{r(H)}{2} + \frac{r(I)}{2} + r(L) + r(M).$$

On se rend compte que calculer le *ranking* de toute page du Web est bien trop compliqué. Il existe pourtant une théorie qui nous dit que le vecteur $\mathbf{r}$ est tel que

$$\mathbf{r} = A\mathbf{r} \tag{1}$$

où $A = (a_{ij})$ est appelée matrice de connexion et elle est définie par

$$a_{ij} = \begin{cases} \dfrac{1}{N_j}, & \text{s'il existe un lien entre } p_j \text{ et } p_i \\ 0, & \text{autrement.} \end{cases}$$

Cette matrice a la propriété suivante

$$\sum_i a_{ij} = 1, \quad \forall j, \tag{2}$$

car chaque élément a_{ij} peut être intérpreté comme la probabilité que quelqu'un arrive à la page p_j en cliquant "au hasard" en partant de la page p_i.

Si on regarde la rélation (1) on observe que le vecteur $\mathbf{r}$ est le vecteur propre associé à la valeur propre $\lambda = 1$. Puisque le nombre N de pages Web existantes est très grand il n'est pas envisageable d'utiliser une méthode directe pour le calcul du spectre de A. On veut donc utiliser un algorithme itératif comme la méthode des puissances, expliquée dans la suite de ce document.

La méthode des puissances

Il est possible d'appliquer la *méthode des puissances* pour calculer le spectre d'une matrice dont la valeur propre maximale λ_1 a une multiplicité unitaire, c.a.d. qu'elle est associée à un seul vecteur propre $\mathbf{v}_1$.

Chaque itération k de l'algorithme est définie par

$$\begin{aligned} \mathbf{q}^{(k)} &= \frac{A\mathbf{q}^{(k-1)}}{\|A\mathbf{q}^{(k-1)}\|_2}, \\ \nu^{(k)} &= \mathbf{q}^{(k)^T} A\mathbf{q}^{(k)}, \end{aligned} \tag{3}$$

où $\nu^{(k)}$, $\mathbf{q}^{(k)}$ répresentent respectivement les approximations de la valeur propre λ_1 et du vecteur propre à droite $\mathbf{v}_1$ associé[1,2]. Le critère d'arrêt de l'algorithme est donné par

$$\|A\mathbf{q}^{(k)} - \nu^{(k)}\mathbf{q}^{(k)}\|_2 \leq \epsilon.$$

Question 1. Écrire la fonction `Scilab` `matrice` qui définit la matrice de connexion A associé au graphe de la Figure 2. Quelle est la matrice A trouvée ?

Question 2. Écrire la fonction `sumA` qui prend en entrée une matrice A et qui vérifie que la matrice satisfait la propriété (2). La matrice associée au graphe de Figure 2 vérifie-t-elle cette propriété ?

[1] **Rappel** : on définie vecteur propre à droite associé à la valeur propre λ le vecteur $\mathbf{v}$ tel que $A\mathbf{v} = \lambda\mathbf{v}$.

[2] **Rappel** : on définit la norme 2 d'un vecteur $\mathbf{x}$ comme la racine carrée de la somme de ses éléments au carré : $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$

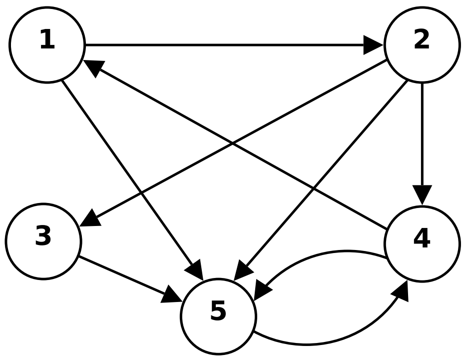

Figure 2: Graphe associé à 5 pages Web, chaque cercle répresente une page Web et chaque flèche répresente un lien.

Question 3. Écrire la fonction `puissance` qui calcule la première valeur propre et le vecteur propre associés d'une matrice. La fonction `puissance` prend en entrée la matrice A, le vecteur initial $\mathbf{q}^{(0)}$ et la tolérance ϵ :

```
[l,v] = puissance(A,q0,tol)
```

Question 4. Utiliser la fonction `puissance` pour calculer le *PageRanking* associé au graphe de la Figure 2. On prend comme valeurs d'entrée $\mathbf{q}^{(0)} = [0 \ldots 0]^T$ et $\epsilon = 10^{-5}$. Quelle est la valeur du vecteur $\mathbf{r}$?

Question 5. Comparer les valeurs de `l` et `v` trouvées à la question précédente à l'aide de la fonction `puissance`, avec le résultat trouvé avec la fonction `spec` de `Scilab`.

Question 6. Discuter dans votre rapport de l'influence du paramètre ϵ et de la valeur initiale $\mathbf{q}^{(0)}$ sur la précision de la méthode des puissances en comparant avec la fonction `spec`.

Question 7. La matrice B est donnée par :

$$B = \begin{bmatrix} 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 1/2 & 0 \\ 1/3 & 1/2 & 0 & 1 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

Dessiner (sur votre rapport) le graphe associé. En utilisant respectivement les fonctions `sumA` et `puissance`, vérifier que la propriétée (2) est satisfaite et déterminer le vecteur de *PageRanking* $\mathbf{r}$ associé.

Question Bonus. Dessiner le graphe associé à la matrice B en utilisant `Scilab` (les lignes de code et la figure trouvée doivent être inclus respectivement dans le fichier à envoyer et dans le rapport).