caim2.pdf

fibsbook

Búsqueda y Análisis de Información Masiva

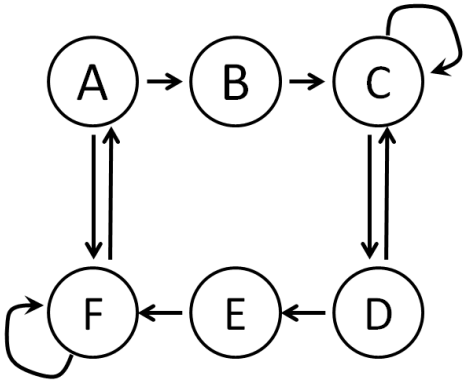
3º Grado en Ingeniería Informática

Facultad de Informática de Barcelona (FIB)

CAIM, segon examen parcial

21 de gener de 2012. Temps: 1 hora 50 minuts

Exercise 1 (2 points) Donat el graf



- Calculeu el pagerank de cada vèrtex usant damping factor 1; suggerim que comproveu la solució que trobeu per substitució en les equacions.
- Suposeu que ara prenem un valor $\lambda < 1$ com a damping factor i recalculeu els pageranks. Suposant que la vostra solució en l'apartat anterior és correcta, podem dir quins nodes milloraran el seu pagerank i quins l'empitjoraran? O això dependrà de λ?

Answer: By symmetry, $A = D$, $B = E$, $C = F$. Equations are $A = C/2$, $B = A/2$, $C = B + A/2 + C/2$, $A + B + C + D + E + F = 1$. The last is equivalent to $A + B + C = 1/2$. Replacing C from the last, then the second gives $A = (1/2 - A - B)/2 = 1/4 - A/2 - A/4$ which gives $7/4\, A = 1/4$ or $A = D = 1/7$, then $B = E = 1/14$, then $C = F = 2/7$.

When $\lambda < 1$, we are taking a convex combination of the computed pageranks and the uniform distribution. Those nodes with pagerank above $1/6$ (C,F) will decrease, and those with pagerank below $1/6$ (A,B,D,E) will increase.

Exercise 2 (2 points) Sigui $G = (V, E)$ un graf no dirigit, o sigui, si $(i, j) \in E$ llavors $(j, i) \in E$. Fem servir damping factor $\lambda = 1$. Anem a demostrar que per a tot $i \in V$, PageRank(i)=grau(i)/(2$|E|$).

1. Escriviu l'equació que expressa PageRank(i) en funció dels pageranks de tots els vèrtexos.
2. Preneu qualsevol constant c. Demostreu que si prenem PageRank(i) = $c \cdot$ grau(i) per a tot i, satisfarem les $|V|$ equacions anteriors.

3. Ara useu que els pageranks han de sumar 1 per deduir el valor que c ha de tenir.

Answer: 1) The equation is $P(i) = \sum_{(j,i)\in E} P(j)/degree(j)$.
2) Crucially $outdegree(j) = indegree(j) = degree(j)$, so the equation becomes $P(i) = \sum_{(j,i)\in E} P(j)/degree(j) = \sum_{(j,i)\in E} c \cdot degree(j)/degree(j) = c \cdot degree(i)$, as desired.
3) Since $\sum_i P(i) = \sum_i c \cdot degree(i) = c \cdot 2|E|$ must be 1, $c = 1/(2|E|)$.

Exercise 3 (0.1 points) Discutiu què és més fàcil de calcular (algorísmicament), donat accés complet a les dades: 1) el pagerank en el graf definit per la relació "follows" a Twitter. 2) el pagerank en el graf definit per la relació "amic" a Facebook.

Exercici 4 (2 punts) Considereu un graf bipartit complet amb n i m nodes. Calculeu les següents mesures sobre aquest graf en funció de n i m: diàmetre, coeficient de clustering (local), mitjana de totes les distàncies entre parells de nodes, i mitjana harmònica de totes les distàncies entre parells de nodes. A més a més, per a cada un dels nodes del graf, calcula la seva centralitat (de grau, "betweenness", i "closeness") sense normalitzar.
Answer: diameter = 2, for the shortest path computations use the fact that distances among nodes in each partition is 2 ($\binom{m}{2} + \binom{n}{2}$ such), and in different partitions is 1 (nm such). Clustering coefficient is 0 since there are no triangles in the graph. For the centrality measures, degree centrality is m for nodes in the partition with n nodes, and n for nodes in the partition with m nodes. For betweenness, the contribution of one pair of nodes i, j in one partition towards the centrality of another node k in the other partition is $1/n$ for the nodes k in the partition with n nodes. There are such $\binom{m}{2}$ such pairs i, j, so centrality of node k is $\frac{m(m+1)}{2n}$; for the other nodes (in partition with m nodes, this is (analogously) $\frac{n(n+1)}{2m}$). For closeness centrality, the avg. distance of a node i (in the partition with n nodes) to all other nodes is $\frac{(n-1)*2+m*1}{n+m-1}$ and closeness centrality is calculated as its inverse, that is, $\frac{n+m-1}{2n-2+m}$. For the other nodes, we just need to exchange n with m.

Exercise 5 (2 points) Ens donen un nombre gran de fitxers N escrits en llengües diferents. Volem triar un subconjunt S dels fitxers que cobreixi totes les llengües i determinar la freqüència de cada llengua. Això és, per a cada llengua present entre els fitxers hi ha d'haver exactament un fitxer a S en aquesta llengua, junt amb el nombre de fitxers en ella. Expliqueu com resoldre eficientment aquesta tasca en el model de programació Mapreduce.

Per ser precisos, suposem que a cada instància de map li arriba una llista de strings, on cada string és el nom d'un fitxer que ha de processar. Volem retornar, per cada llengua present al corpus, una tupla com ara `(‘‘English’’,(‘‘The_Road.txt’’,2489))`, que indica que hi ha 2489 fitxers en anglès i que `‘‘The_Road.txt’’` n'és un. Suposeu que hi ha una funció `language(string t)` que retorna el nom de la llengua en la què és escrit el text `t`, com ara `‘‘English’’`.
Answer:

Use the following map and reduce functions. The reducer can (and should) be used as combiner too.

```
map(list L)
for each filename in L
open(filename);
for each (line f in filename)
open(f);
output(language(f),(f,1));
```

```
reduce(key,list L of pairs(string,int))
f = L[0].first; // pick up, say, first file as representative
s = L[0].second+...+L[last].second;
output (key,(f,s));
```

Exercise 6 (2 points; aquest exercici serveix per avaluar la competència transversal "Aprenentatge Autònom") Expliqueu, en el procés SEO, alguna idea de com usar / optimitzar els següents elements d'una pàgina:

- Keywords
- Estructura de la web; clicks que els usuaris fan en el nostre website
- Enllaços que conté la nostra pàgina

(N'hi ha prou amb un parell o tres de frases per item).