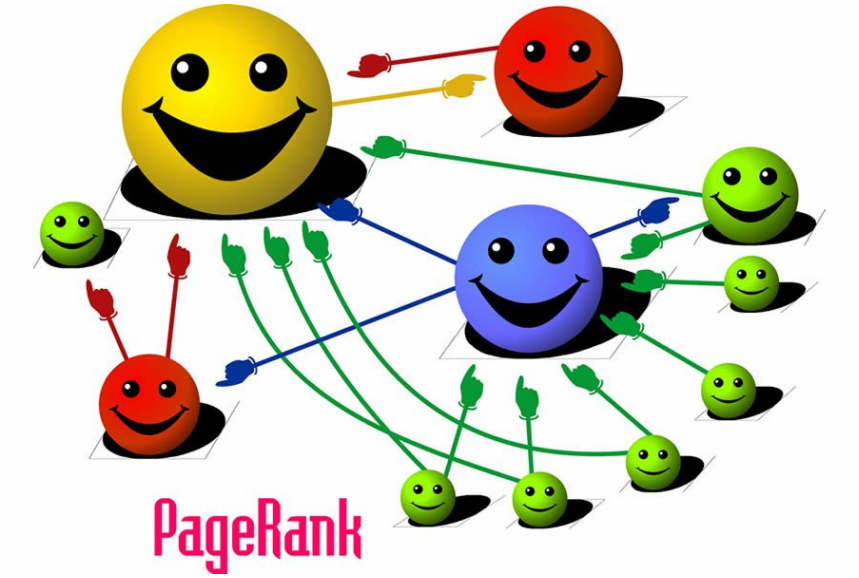
CAIM Lab
Session 5: Pagerank



Ana Mestre Borges

1. Problemes principals

Per la implementació d'aquesta pràctica els problemes principals han estat comprendre del tot com funcionava l'algorisme de Page Rank, que la base en sí és senzilla però comprendre el perquè de cada càlcul ha portat més temps i sobretot, pensar la forma més òptima d'aplicar-lo en el nostre cas. Es podria haver fet un algorisme quadràtic en nombre de rutes molt més senzill però, hagués estat increïblement més ineficient en temps. Per tant, en quant a pensar en la optimització del codi, m'he trobat amb bastants problemes amb aclarir-me amb els índexs de cada classe, llista, diccionari... però finalment ho he pogut treure.

Una de les altres dificultats era que la suma del page rank donés igual a 1, no aconseguia arribar a aquest número i al principi tenia localitzat quin era el problema, els aeroports amb out weight igual a zero, finalment, després de rumiar-ho una estona, he arribat a la conclusió de que s'havien de distribuir el pageranks d'aquests nodes per tal de no "perdre" el seu valor.

2. Decisions que s'han pres

Per observar com es comportava el programa s'han hagut de prendre decisions sobre el valor del damping factor i la stopping condition del bucle que va calculant el pagerank.

- Damping factor:
    - Per valors petits (per exemple 0.2), els valors en general del pagerank són més semblants entre ells, tarda molt poc (0.91 segons) i es realitza el càlcul en poques iteracions (38 iteracions).
    - Per valors grans (per exemple 0.95), els valors comencen a ser més distants entre ells, el més gran és 0.00689 i el menor 1.46-05, tarda molt més que en l'experiment anterior (12.32 segons) i es realitza el càlcul en moltes més iteracions (512 iteracions).
    - Per valors entre el rang 0.8 i 0.9 (per exemple 0.85), s'aconsegueixen uns valors que també estan bastant distants entre ells però que no s'allunyen massa dels resultats obtinguts amb 0.95, però, en canvi el temps ha estat 3 cops més ràpid (3.771 segons) i ha fet 3 cops menys iteracions, per tant és bastant clar que per obtenir uns valors suficientment bons però sense sacrificar temps d'execució la millor idea és quedar-nos entre l'interval de 0.8 i 0.9.

- Stopping condition:

    En un principi havia plantejat el problema amb un nombre concret d'iteracions bastant gran per assegurar-me de que el codi funcionava correctament i feia el que s'esperava d'ell. No obstant, aquesta condició estava imposada per mi, òbviament no era la més adequada i es podia observar que tardava bastant temps en treure un resultat, aproximadament 5 cops més que amb el resultat que tinc ara. Per tant, el següent pas era optimitzar-lo, per a fer-ho vaig decidir que el moment d'aturar-se seria quan el nou valor del pagerank i el que s'havia calculat en la iteració anterior, eren prou semblants. Semblants, no tenen perquè ser iguals, després de jugar amb diversos números, vaig decidir quedar-me amb que s'aturés quan la diferència fos menor que $1x10^{-15}$. A més a més, he observat que si poso aquest número més petit aconsegueixo pràcticament els mateixos resultats però és bastant més lent.