Implement

Ya Zhu

2015/5/20

1 Structure

I use adjacency matrix to represent the graph, and save it by *scipy.sparse* in Python, which can just consume memories of non-zero elements in the matrix. Indeed, what we really need in the power iteration is A^T, so I directly parse the input file into sparse matrix $A' = A^T$.

2 Zero out-degree nodes

As described above, the matrix A' is like

$$a_{ij} = \begin{cases} \frac{1}{O_j} & \text{if } (j,i) \in E \\ \\ 0 & \text{otherwise.} \end{cases}$$

As for a node j without any out-links, I haven't assigned $a_{ij} = \frac{1}{N-1},\ \forall i \neq j$, where N is total number of nodes. This is because there could be too many zero out-degree nodes in a graph that the matrix would become dense. For instance, there are 350004 nodes in *stanford* but only 111550 nodes have out-links, it is inappropriate to save more than 200000 dense columns in the matrix. Therefore, I choose to deal with those nodes in the process of Power Iteration instead, maintaining the matrix stochastic.

3 Power Iteration

Note that

$$\begin{pmatrix} a_{11} & \frac{1}{N-1} & a_{13} & \frac{1}{N-1} \\ a_{21} & 0 & a_{23} & \frac{1}{N-1} \\ a_{31} & \frac{1}{N-1} & a_{33} & \frac{1}{N-1} \\ a_{41} & \frac{1}{N-1} & a_{43} & 0 \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix} = \begin{pmatrix} a_{11}P_1 + \frac{1}{N-1}P_2 + a_{13}P_3 + \frac{1}{N-1}P_4 \\ a_{21}P_1 + 0 * P_2 + a_{23}P_3 + \frac{1}{N-1}P_4 \\ a_{31}P_1 + \frac{1}{N-1}P_2 + a_{33}P_3 + \frac{1}{N-1}P_4 \\ a_{41}P_1 + \frac{1}{N-1}P_2 + a_{43}P_3 + 0 * P_4 \end{pmatrix}$$

$$= \begin{pmatrix} a_{11} & 0 & a_{13} & 0 \\ a_{21} & 0 & a_{23} & 0 \\ a_{31} & 0 & a_{33} & 0 \\ a_{41} & 0 & a_{43} & 0 \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix} + \frac{1}{N-1} \begin{pmatrix} P_2 + P_4 \\ P_2 + P_4 \\ P_2 + P_4 \\ P_2 + P_4 \end{pmatrix} - \frac{1}{N-1} \begin{pmatrix} 0 \\ P_2 \\ 0 \\ P_4 \end{pmatrix}$$

So the update equation in Power Iteration can be rewritten as

$$1 - d + dA^T P = 1 - d + d(A'P + \frac{1}{N-1}\sum_{j \in C} P_j - \frac{1}{N-1}P').$$

Where C is the list of zero out-degree nodes, and $P' = \Sigma P$, where Σ is a diagonal matrix,

$$\Sigma_{ii} = \begin{cases} 1 & \text{if } i \in C \\ 0 & \text{otherwise.} \end{cases}$$