

# Lab 8: PageRank

Ruofan Zhou

May 27, 2014

## 1 Random Surfer Model

### 1.1 NaiveRandomSurfer

The method **compute** is complemented as below:

```
public PageRank compute(Graph graph) {  
    int nbNodes = graph.size();  
    double[] probabilities = new double[nbNodes];  
  
    Random rand = new Random();  
    int nownode = rand.nextInt(nbNodes);  
    for (int loops = 0; loops < NB_ITERATIONS; ++loops) {  
        List<Integer> nei = graph.neighbors(nownode);  
        int mount = nei.size();  
        int next = rand.nextInt(mount);  
        next = (Integer) nei.get(next);  
        probabilities[next] += 1;  
        nownode = next;  
    }  
    for (int i = 0; i < nbNodes; ++i)  
        probabilities[i] = (double)probabilities[i]  
        / NB_ITERATIONS;  
    return new PageRank(graph, probabilities);  
}
```

The **component.graph** is not a connected graph, thus the NaiveRandomSurfer can't go to somewhere of the graph(i.e., some nodes' PageRank is 0).

The **absorbing.graph** has a node with no outdegree, thus cause an Exception when using NaiveRandomSurfer.

## 1.2 RandomSurfer

The method **compute** is complemented as below:

```
public PageRank compute(Graph graph) {  
    int nbNodes = graph.size();  
    double[] probabilities = new double[nbNodes];  
  
    Random rand = new Random();  
    int nownode = rand.nextInt(nbNodes);  
    for (int loops = 0; loops < NB_ITERATIONS; ++loops) {  
        List<Integer> nei = graph.neighbors(nownode);  
        int mount = nei.size();  
        double rant = rand.nextDouble();  
        if (rant - 0.15 < 1e-9 || mount < 1) {  
            nownode = rand.nextInt(nbNodes);  
            probabilities[nownode] += 1;  
        }  
        else {  
            int next = rand.nextInt(mount);  
            next = (Integer) nei.get(next);  
            probabilities[next] += 1;  
            nownode = next;  
        }  
    }  
    for (int i = 0; i < nbNodes; ++i)  
        probabilities[i] = (double)probabilities[i]  
            / NB_ITERATIONS;  
    return new PageRank(graph, probabilities);  
}
```

It now can deal with the component.graph and absorbing.graph.

## 2 Power Iteration Method

### 2.1 googleMatrix

```
public static double[][] googleMatrix(Graph graph) {  
    int n = graph.size();  
    double[][] probs = new double[n][n];  
    double delta = DAMPING_FACTOR / n;  
    double theta = 1 - DAMPING_FACTOR;  
  
    double[][] H = new double[n][n];
```

```

    for (int i = 0; i < n; ++i) {
        List<Integer> nei = graph.neighbors(i);
        if (nei.size() == 0) {
            for (int j = 0; j < n; ++j) {
                H[i][j] = 1 / (double)n;
            }
        } else {
            for (int j = 0; j < n; ++j) {
                if ((i != j) && graph.containsEdge(i, j)) {
                    H[i][j] = 1 / (double)nei.size();
                }
            }
        }
    } // ^H
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < n; ++j)
            probs[i][j] = theta * H[i][j] +
                (1 - theta) / (double) n;

    return probs;
}

```

## 2.2 20 First PageRank

- 1 United States (PR: 0.73570%)
- 2 United Kingdom (PR: 0.52027%)
- 3 France (PR: 0.48089%)
- 4 Europe (PR: 0.43404%)
- 5 England (PR: 0.39715%)
- 6 Germany (PR: 0.38127%)
- 7 Latin (PR: 0.37843%)
- 8 World War II (PR: 0.35895%)
- 9 India (PR: 0.34712%)
- 10 English language (PR: 0.33081%)
- 11 Australia (PR: 0.32051%)
- 12 London (PR: 0.30620%)
- 13 Japan (PR: 0.30054%)
- 14 Italy (PR: 0.28641%)
- 15 Canada (PR: 0.28597%)
- 16 Water (PR: 0.28466%)
- 17 China (PR: 0.28365%)
- 18 Spain (PR: 0.24892%)
- 19 Russia (PR: 0.24787%)
- 20 Animal (PR: 0.24766%)

## 2.3 Gaming The System

## 2.4 AddIncomingEdges

```
public static void addIncomingEdges(Graph graph, int node) {  
    PageRankAlgorithm algo = new PowerMethod();  
    PageRank pr = algo.compute(graph);  
    int max = -1;  
    double maxans = -1;  
    for (int loops = 0; loops < 300; ++loops) {  
        max = -1;  
        for (int i = 0; i < graph.size(); ++i) {  
            if (i == node) continue;  
            if (!graph.containsEdge(i, node)) {  
                double now = pr.get(i) /  
                    (double)graph.neighbors(i).size();  
                if (max == -1) {max = i; maxans = now;}  
            } else if (maxans - now < 1e-9) {  
                max = i;  
                maxans = now;  
            }  
        }  
    }  
    if (max < 0) return;  
    graph.addEdge(max, node);  
}
```

And ran 98.51% of the standard best pagerank.

## 2.5 AddEdges

I tried some algorithms but none of them work better than **addIncomingEdges**.