山东大学 软件 学院

众智科学与网络化产业 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201800301249 | 姓名： 王帅 | | 班级： 计软18.4 |
| 实验题目：Pagerank算法实验 | | | |
| 实验学时：4 | | 实验日期： 2020.4.1 | |
| 实验目的：  熟悉Pagerank算法、六度空间理论 | | | |
| 硬件环境：  Pc机 | | | |
| 软件环境：  Dev C++ | | | |
| 实验步骤与内容：  输入：随机有向图，n\*n的矩阵  输出：   1. 实现pagerank算法，算出每个节点的pagerank值 2. 将图改为无向图后，计算符合“六度空间”理论的结点占结点总数的百分比   代码如下：  #include <iostream>  #include <iomanip>  #include <queue>  using namespace std;  int a[100][100]; //最多可容纳100个节点  queue <int>q; //队列用来检查是否符合六度空间理论  void pagerank(int n)  {  int numin[n]; //存放每个节点的入链数  int numout[n]; //存放每个节点的出数  int i,j;  for(i=0;i<n;i++) //初始化，全部放0  {  numin[i]=0;  numout[i]=0;  }    for(int i=0;i<n;i++) //统计每个节点的入链和出链  {  for(int j=0;j<n;j++)  {  if(a[i][j]==1)  {  numin[j]=numin[j]+1;  numout[i]=numout[i]+1;  }  }  }    double rank[n]; //存放每个节点当前的pagerank值  for(int i=0;i<n;i++) //初始化，全为1/n  {    rank[i]=1/double(n);  }    cout<<"各节点的pagerank值："<<endl;  for(int i=0;i<n;i++)  {  cout.width(10);  cout<<i+1;  }  cout<<endl;    for(int i=0;i<n;i++)  {  cout.width(10);  cout<<setprecision(3)<<std::fixed<<1/double(n);  }  cout<<endl;    double tempin,tempout; //每次迭代每个节点的平均入链，出链值  double temp[n]; //临时存放本次的PageRank值    for(int k=0;k<3;k++) //统计三次迭代的PageRank值  {  for(int i=0;i<n;i++)  {  temp[i]=rank[i];  }    for(int i=0;i<n;i++)  {  if(numout[i]!=0)  {  tempout = rank[i]/numout[i];  for(j=0;j<n;j++) //找i点到达的点，并加上i给他们的pagerank值  {  if(a[i][j]==1)  {  temp[j] = temp[j]+tempout;  }  }  }  temp[i]=temp[i]-rank[i];    }    for(int i=0;i<n;i++) //temp再赋给rank，完成rank的更新  {  rank[i]=temp[i];  cout.width(10);  cout<<setprecision(3)<<std::fixed<<rank[i];  }    cout<<endl;  }  }  bool check(int n,int k) //检查第n个节点  {  int m; //存当前检验点  int step = 0;  bool visited[n]; //存节点有无入队过  for(int i=0;i<n;i++) visited[i] = false;    for(int i=0;i<n;i++)  {  if(a[k][i]==1)  {  q.push(i); //k能到的点入队  visited[i] = true;  }  }    int last = q.back(); //last存队尾元素，方便统计step  step++;    while(!q.empty())  {  m = q.front();  q.pop();  visited[m]=true;    if(m = last)  {  step++;  if(a[m][k]==1 && step<=6) return true;  for(int i=0;i<n;i++)  {  if(a[m][i]==1 && visited[i]==false) q.push(i);  }  last = q.back();  }  else  {  if(a[m][k]==1 && step<=6) return true;  for(int i=0;i<n;i++)  {  if(a[m][i]==1 && visited[i]==false) q.push(i);  }  }  }  return false;  }  double SixDegrees(int n)  {  int fitSD = 0; //符合六度空间的节点数  for(int i=0;i<n;i++) //检查每一个节点是否符合  {  bool result = check(n,i);  if(result) fitSD++;  }  double SDpersent = double(fitSD)/n;  return SDpersent;  }  int main()  {  cout.setf(std::ios::left);  cout<<"请输入节点个数：";  int n;  cin>>n;  cout<<"请逐行输入有向图的关系矩阵："<<endl;  for(int i=0;i<n;i++)  {  for(int j=0;j<n;j++)  {  cin>>a[i][j];  }  }    pagerank(n);    //将a转换为无向图的矩阵  for(int i=0;i<n;i++)  {  for(int j=0;j<n;j++)  {  if(a[i][j]!=a[j][i])  {  a[i][j]=1;  a[j][i]=1;  }  }  }    double SDpersent = SixDegrees(n);  cout<<"将图改为无向图后，符合六度空间理论的结点占结点总数的比例为"<<SDpersent;  return 0;  }  运行结果：    输入关系矩阵后，输出每个节点的pagerank值和目标比例。 | | | |
| 结论分析与体会：  本次实验让我上课所学到的知识与熟悉的软件编程结合起来，即加强了我的编程能力，又加深了我对Pagerank和六度空间这两个概念的理解，更让我感受到了众智科学与网络化产业这门课程的实用性。 | | | |