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：202000300125 | 姓名：贾星宇 | | 班级：2020级5班 |
| 实验题目：**新事物扩散模拟实验** | | | |
| 实验学时：4 | | 实验日期：2022年5月1日 | |
| 实验目的：  理解新事物的扩散过程，进一步深化“博弈”的思想，理解优秀的新事物为何会以更快的速度扩散到一个群体中。 | | | |
| 硬件环境：  Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz | | | |
| 软件环境：  CLion 2021.2.3 | | | |
| 实验步骤与内容：  【题目要求】  输入：给定任意网络结构，初始节点集合和门槛值  输出：模拟扩散过程，一步步看哪些节点“接受了”新事物，直到不再有扩散的可能。  一种新事物会如何在网络中得到传播，或者说传播的效果将会如何，在鼓励创新的年代，是一个尤其值得关心的问题。网络级联，即人们通过与有关系的人之间的互动决定是否采纳一项新事物（例如某一款新手机），是学界已经形成的一类基本模型，其形象化的表现就是新事物在网络中从一些点开始，逐步向周边扩散，直至完全覆盖所有节点，或者到某个程度后再也不会扩散了。  【假设分析】  假设原来的旧节点给人们带来的收益为a=2，新节点带来的收益为b=3。  构建无向网状结构，起初只有少部分人使用b  对于某节点来说，如果它与邻接节点同用a/b，则其收益为2/3，如果不相同则收益为0。  根据博弈论的思路，此人周围的节点至少有2/5（0.4）个节点用b时，此人才会用b  利用此思路来编写对应代码，无向网状结构用邻接矩阵来存储。  【代码】  #include <iostream> using namespace std;  *//得到相邻的a节点数目* int getNeiborA(int \*\*graph,char \*things,int length,int theNode){  int result = 0;  for (int i = 0; i < length; ++i) {  if (graph[theNode][i]==1&&things[i]=='a')result++;  }  return result; }   *//得到相邻的b节点数目* int getNeiborB(int \*\*graph,char \*things,int length,int theNode){  int result = 0;  for (int i = 0; i < length; ++i) {  if (graph[theNode][i]==1&&things[i]=='b')result++;  }  return result; }   *//更新节点,如果进入且更新了返回true* bool updateOneNode(int \*\*graph,char \*things,int length,int theNode){  double a = getNeiborA(graph,things,length,theNode);  double b = getNeiborB(graph,things,length,theNode);  if ((double )(b/(a+b))>=0.4){  things[theNode]='b';  return true;  }  return false; }   int main() {   *//利用邻接矩阵存储有向图  //5  //0 1 1 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 0 1 0 1 0 b b a a a* int length;  cin >> length;  int \*\* graphMartix = new int\*[length];   for (int i = 0; i < length; ++i) {  graphMartix[i] = new int[length];  }   *//创建图* for (int i = 0; i < length; ++i) {  for (int j = 0; j < length; ++j) {  cin >> graphMartix[i][j];  }  }   *//创建节点--事物对应关系* char \* things = new char[length];  for (int i = 0; i < length; ++i) {  cin >> things[i];  }   *//一次次更新节点* while (true){  int flag = 0;   for (int i = 0; i < length; ++i) {  if (things[i]=='a'){  if (updateOneNode(graphMartix,things,length,i)) flag=1;  }  }   for (int i = 0; i < length; ++i) {  cout << things[i]<<", ";  }  cout << endl;   if (flag==0)break;  }   return 0; } | | | |
| 结论分析与体会：  测试数据：  *//5  //0 1 1 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 0 1 0 1 0 b b a a a*  示例无向网状结构：    得到结果：    可以看到，第一次更新后每个节点都被更新成了b，说明在一开始黑色节点的邻接红色节点数大于0.4。  第二次检查每个节点都没有被更新后退出了循环。  【总结】  1.一种新的行为、做法、意见、约定或技术是如何在朋友的影响下，从一个人到另一个人在社会网络中传播。  2.在一个基本的社会网络中，每个节点有两种可能的选择，标记为A和B。如果节点v和w由一个边连接，那么对它们来说存在一个行为相匹配的动机。  相应的回报定义如下  如果v和w都选择A，它们分别得到回报a > 0  如果它们都选择B，分别得到回报b > 0  如果它们选择不同的选项，那么都得到回报为0  3. 如果有多个邻居，一些邻居选择A，另一些邻居选择B，那么v应该选择哪个选项使他的回报达到最大呢？  回报值对应为a，b。假设v的邻居中，比例为p的邻居选择A，比例为(1-p)的邻居选择B。如果有d个邻居，则pd个采用A，(1-p)\*d个采用B  如果v选择A，得到回报为pda，选择B，得到的回报为(1-p)\*db  如果pda ≥ (1-p)\*db  则选择A更好，即p ≥ b / (a + b) | | | |