|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 程序编号 | 1-1 | 文件名称 | dataprework.m | 程序说明 | 数据处理 |
| clc,clear,close;  % 数组中存在非数值数据如字符串时候用元胞数组读取  [~,~,finor] = xlsread('已结束项目任务数据.xlsx');  [~,~,vip] = xlsread('会员信息数据.xlsx');  [~,~,newor] = xlsread('新项目任务数据.xlsx');  %% 处理已经完成任务单数据  finor(1,:) = [];  N = size(finor,1);  % 处理元胞数组中的字符串  for i =1:N  string = char(finor(i,1)); % 当前元胞单元格的字符串  string(1) = [];  finor(i,1) = {str2double(string)}; % 将任务号码转化为数字形式  end  finor = cell2mat(finor);  % 经纬度异常值处理  for i = 1:N  if finor(i,2) >= 90 && (finor(i,3) <=90 && finor(i,3) >=0)  fprintf('已完成订单第 %d 条数据出现异常值\n',i);  tmp = finor(i,2);  finor(i,2) =finor(i,3);  finor(i,2) = tmp;  elseif finor(i,2) >180 || finor(i,3) > 180  fprintf('已完成订单 %d 条数据出现异常值\n',i);  finor(i,:)=[];  end  end  %% 处理会员信息数据  vip(1 , :) = [];  N = size(vip,1);  for i =1:N  string = char(vip(i,1)); % 当前元胞单元格的字符串  string(1) = [];  vip(i,1) = {str2double(string)}; % 将任务号码转化为数字形式    tmp = char(vip(i,2));  index = find(tmp == ' ');  string1 = tmp(1:index-1);  string2 = tmp(index+1:end);  vip(i,2) = { str2double(string1)};  vip(i,6) = { str2double(string2)};  end  vip = cell2mat(vip);  tmp1= vip(:,1);tmp2=vip(:,2);tmp3=vip(:,6);  vip(:,1)=[];vip(:,1)=[];vip(:,4)=[];  vip = [tmp1,tmp2,tmp3,vip];  % 经纬度异常值处理  for i = 1:N  if vip(i,2) >= 90 && (vip(i,3) <=90 && vip(i,3) >=0)  fprintf('会员第 %d 条数据出现异常值\n',i);  tmp = vip(i,2);  vip(i,2) =vip(i,3);  vip(i,2) = tmp;  elseif vip(i,2) >180 || vip(i,3) > 180  fprintf('会员第 %d 条数据出现异常值\n',i);  vip(i,:)=[];  end  end  %% 处理新项目任务数据  newor(1,:) = [];  N = size(newor,1);  % 处理元胞数组中的字符串  for i =1:N  string = char(newor(i,1)); % 当前元胞单元格的字符串  string(1) = [];  newor(i,1) = {str2double(string)}; % 将任务号码转化为数字形式  end  newor = cell2mat(newor);  % 经纬度异常值处理  for i = 1:N  if newor(i,2) >= 90 && (newor(i,3) <=90 && newor(i,4) >=0)  fprintf('新订单第 %d 条数据出现异常值\n',i);  tmp = newor(i,2);  newor(i,2) =newor(i,3);  newor(i,2) = tmp;  elseif newor(i,2) >180 || newor(i,3) > 180  fprintf('新订单第 %d 条数据出现异常值\n',i);  newor(i,:)=[];  end  end  N = size(finor,1);  M = size(vip,1);  viptoor=zeros(M,N);  R=6370;  for i = 1:M  for j=1:N  % 点 a 的经度为 u\_a , 纬度 v\_a  ua = vip(i,3); va = vip(i,2);  ua = deg2rad(ua);va = deg2rad(va);    %点 b 的经度为 u\_b , 纬度 v\_b  ub = finor(j,3); vb = finor(j,2);  ub = deg2rad(ub); vb = deg2rad(vb);    % 计算城市i和j的距离  viptoor(i,j) = R \* acos( cos(ua - ub) \* cos(va) \* cos(vb) + sin(va) \* sin(vb) );  end  end  vipd = zeros(M,M);  % 转化经纬度求距离矩阵  for a = 1 : M  for b = 1 : a  if(a == b)  vipd(a,b) = 1000000;  continue;  end  % 点 a 的经度为 u\_a , 纬度 v\_a  ua = vip(a,3); va = vip(a,2);  ua = deg2rad(ua);va = deg2rad(va);  %点 b 的经度为 u\_b , 纬度 v\_b  ub = vip(b,3); vb = vip(b,2);  ub = deg2rad(ub); vb = deg2rad(vb);  % 计算城市i和j的距离  vipd(a , b) = R \* acos( cos(ua - ub) \* cos(va) \* cos(vb) + sin(va) \* sin(vb) );  end  end  vipd = vipd + vipd';  N = size(finor,1);  d = zeros(N); % 初始化两个城市的距离矩阵全为0  % 转化经纬度求距离矩阵  for a = 1 : N  for b = 1 : a  if(a == b)  d(a,b) = 1000000;  continue;  end    % 点 a 的经度为 u\_a , 纬度 v\_a  ua = finor(a,3); va = finor(a,2);  ua = deg2rad(ua);va = deg2rad(va);    %点 b 的经度为 u\_b , 纬度 v\_b  ub = finor(b,3); vb = finor(b,2);  ub = deg2rad(ub); vb = deg2rad(vb);    % 计算城市i和j的距离  d(a , b) = R \* acos( cos(ua - ub) \* cos(va) \* cos(vb) + sin(va) \* sin(vb) );  end  end  d=d+d';  M = size(vip,1);  ornearvip=zeros(1,N);  ornearor = zeros(1,N);  for i =1:N  for j=1:N  if i==j  continue;  end  if d(i,j) <= 10  ornearor(i)=ornearor(i)+1;  end  end  end  for i =1:N  for j =1:M  if viptoor(j,i) < 10  ornearvip(i) = ornearvip(i)+1;  end  end  end  Li = min(viptoor);  save('data.mat','finor','vip','newor','viptoor','vipd','d','ornearvip','ornearor','Li'); | | | | | |
| 程序编号 | 1-2 | 文件名称 | ShowInMap.m | 程序说明 | 价格分布可视化 |
| clc,clear,close;  load data.mat;  x = finor(:,3);y=finor(:,2);z=finor(:,4);  x\_=min(x(:,:)):(max(x)-min(x))/500:max(x(:,:));  y\_=min(y(:,:)):(max(y)-min(y))/500:max(y(:,:));  [xi,yi]=meshgrid(x\_,y\_);  zi=griddata(x , y , z , xi , yi , 'cubic'); %这里有三种方法 linear cubic 可以自行选择  figure(1)  scatter3(x,y,z,'b^');  hold on;  meshc(xi , yi , zi);  hold on  colorbar;  hold on;  shading interp  hold on  title('任务定价可视化');  xlabel('经度');ylabel('纬度');zlabel('订单价格'); | | | | | |
| 程序编号 | 1-3 | 文件名称 | main1.m | 程序说明 | k-means聚类 |
| clc,clear,close;  load data.mat;  y= finor(:,2);x=finor(:,3);  figure(1), plot(x,y,'o','MarkerSize',6,'MarkerFaceColor',[255,255,255]/255);  title('聚类前二维点');xlabel('X轴');ylabel('Y轴');  %随机获取150个点  % X = [randn(50,2)+ones(50,2);randn(50,2)-ones(50,2);randn(50,2)+[ones(50,1),-ones(50,1)]];  X = [x,y];  opts = statset('Display','final');    %调用Kmeans函数  %X N\*P的数据矩阵  %Idx N\*1的向量,存储的是每个点的聚类标号  %Ctrs K\*P的矩阵,存储的是K个聚类质心位置  %SumD 1\*K的和向量,存储的是类间所有点与该类质心点距离之和  %D N\*K的矩阵，存储的是每个点与所有质心的距离;    [Idx,Ctrs,SumD,D] = kmeans(X,7,'Replicates',10,'Options',opts);    %画出聚类为1的点。X(Idx==1,1),为第一类的样本的第一个坐标；X(Idx==1,2)为第二类的样本的第二个坐标  figure(2);  plot(X(Idx==1,1),X(Idx==1,2),'r.','MarkerSize',14)  hold on  plot(X(Idx==2,1),X(Idx==2,2),'b.','MarkerSize',14)  hold on  plot(X(Idx==3,1),X(Idx==3,2),'g.','MarkerSize',14)  hold on  plot(X(Idx==4,1),X(Idx==4,2),'k.','MarkerSize',14)  hold on  plot(X(Idx==5,1),X(Idx==5,2),'m.','MarkerSize',14)  hold on  plot(X(Idx==6,1),X(Idx==6,2),'c.','MarkerSize',14)  hold on  plot(X(Idx==7,1),X(Idx==7,2),'y.','MarkerSize',14)  hold on  %绘出聚类中心点,kx表示是x  plot(Ctrs(:,1),Ctrs(:,2) , 'kx' , 'MarkerSize' , 15 , 'MarkerEdgeColor' ,[180,100,100]/255,'linewidth',5);  legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster 6','Cluster 7','Centroids','Location','NW')    Ctrs  SumD | | | | | |
| 程序编号 | 2-1 | 文件名称 | Setprice.m | 程序说明 | 价格设置 |
| clc,close,clear;  load data.mat;  center = [83,68.5;  379,67;  608,66;  772,66.5];  n = size(finor,1);  m = size(finor,2);  new = zeros(n,m);  new(:,1:3) = finor(:,1:3);  alpha = 0.0294; belta = 0.058; theta = 0.8053;  for i =1:n  id = new(i,1);    dis = [d(id , center(1,1)) ,1,center(1,2); % 分别存储，距离、城市号、中心点  d(id,center(2)) ,2,center(2,2);  d(id,center(3)) ,3,center(3,2);  d(id,center(4)),4,center(4,2)];  dd = sort(dis,1);belong = dd(1,2);tbelong = dd(2,2); mi=dd(1,1);tmi =dd(2,1); % 按照距离排序，获取距离最近的城市编号以及距离  if mi <= 4  new(i,4) = center(belong,2);  continue;  end    if mi > 24  new(i,4) = 75;  continue;  end  if mi<16 && (ceil(mi/4) == ceil(tmi/4))  price = max(center(belong,2),center(tbelong,2));  price = price + 2\*ceil(mi/4);  if price >85  price = 85;  end  if price < 65  price = 65;  end  new(i,4) = price;  continue;  end  if mi > 4 && mi<=24  price = floor(mi) + center(belong,2);  if price > 85  price = 85;  end  if price < 65  price =65;  end  new(i,4) = price;  end  price = new(i,4);  price = price - alpha\* ornearor(id) - belta \* ornearvip(id) + theta \* Li(id);  if price > 85  price = 85;  end  if price < 65  price =65;  end  new(i,4) = price;  end  save('newpri.mat','new'); | | | | | |
| 程序编号 | 2-2 | 文件名称 | main2\_2.m | 程序说明 | 对任务聚类 |
| load data.mat;  clc,clear,close;  id= find(finor(:,5) ==0);  id = id'  uncom = finor(id(:),:);  y= uncom(:,2);x=uncom(:,3);  subplot(1,2,1), plot(x,y,'o','MarkerSize',6,'MarkerFaceColor',[255,255,255]/255),grid on;  title('聚类前二维点');xlabel('经度');ylabel('维度');  X = [x,y];  opts = statset('Display','final');    %调用Kmeans函数  %X N\*P的数据矩阵  %Idx N\*1的向量,存储的是每个点的聚类标号  %Ctrs K\*P的矩阵,存储的是K个聚类质心位置  %SumD 1\*K的和向量,存储的是类间所有点与该类质心点距离之和  %D N\*K的矩阵，存储的是每个点与所有质心的距离;    [Idx,Ctrs,SumD,D] = kmeans(X,3,'Replicates',10,'Options',opts);    %画出聚类为1的点。X(Idx==1,1),为第一类的样本的第一个坐标；X(Idx==1,2)为第二类的样本的第二个坐标  subplot(1,2,2)  plot(X(Idx==1,1),X(Idx==1,2),'r.','MarkerSize',14)  hold on  plot(X(Idx==2,1),X(Idx==2,2),'b.','MarkerSize',14)  hold on  plot(X(Idx==3,1),X(Idx==3,2),'g.','MarkerSize',14)  hold on  plot(X(Idx==4,1),X(Idx==4,2),'m.','MarkerSize',14)  hold on  %绘出聚类中心点,kx表示是x  plot(Ctrs(:,1),Ctrs(:,2) , 'kx' , 'MarkerSize' , 15 , 'MarkerEdgeColor' ,'k','linewidth',14);  grid on;  title('聚类完成图')  xlabel('经度');ylabel('纬度');  legend('聚类 1','聚类 2','聚类 3','质心','Location','NW')    Ctrs  SumD | | | | | |
| 程序编号 | 2-3 | 文件名称 | Main2.m | 程序说明 | 蒙特卡洛模拟原价格 |
| clc,clear,close;  load data.mat;  com = 0;  tmp = sortrows(finor,5);  idx= find(tmp(:,5) ==1);  sort(idx); idx = idx(1);idx = idx-1;  tmp1 = tmp(1:idx,:);  tmp2 = tmp(idx+1:end,:);  figure(1);  plot(tmp1(:,3),tmp1(:,2),'kx','MarkerSize',7);  hold on;  plot(tmp2(:,3),tmp2(:,2),'o','MarkerSize',5,'MarkerFaceColor',[36,169,255]/255,'MarkerEdgeColor',[36,169,255]/255);  grid on ;  xlabel('经度');ylabel('纬度');  legend('未完成的任务','完成的任务');  for round =1:1000  [vipinc,id] = sortrows(vip,5);  time = vipinc(:,5);  time = unique(time);  totvipcnt = size(vip,1);  nowor = finor; % 动态维护当前还未被选择的订单  vipmaxor = 30; % 会员可以选取的最大订单数量  timeid = zeros(length(time),2);  N = length(time);  R = 57; % 会员可以选择的任务距离半径  totorcnt = size(finor,1);  tmp = zeros(totorcnt,1);  nowor=[nowor,tmp];  complete = zeros(1,totorcnt); % 统计最后每个订单的完成情况  oldprice = finor(:,4);  %% 得到同一时间所有会员的起始下标和终止下标  for i=1:N  tmpid = find(vipinc(:,5) == time(i));  tmpid = sort(tmpid);  timeid(i,1) = time(1);  timeid(i,2) = tmpid(1);  timeid(i,3) = tmpid(end);  end  %% 模拟会员选择过程  border = [113.75,114.5,22.5,22.8; % 深圳  113.230591,113.432465, 23.063307,23.248917; % 广州  113.023224,113.230591,22.928042,23.054462]; % 佛山  % 三个发达城区对任务完成与否的影响  pb = [0.7,0.8,0.9];  %% 按照时间段去依次枚举  for i =1:N %枚举不同的时间段    st = timeid(i,2); ed = timeid(i,3); % 当前时间段在时间有序vip序列中的起始位置  nowvip = vipinc(st:ed,:);  nowvip = sortrows(nowvip,-6); % 按照信誉度降序排序  nowvipcnt = size(nowvip,1); % 当前时间段的会员数量    %% 获得当前人选择订单成功的概率  p0 =zeros(1 , totvipcnt);  for u = 1 : nowvipcnt    id1 = nowvip(u,1) ;  div = 0;  this = nowvip(u,4);  for k = 1 : nowvipcnt % 求出所有当前订单10km内的当前会员数量    id2 = nowvip(k,1);  if id1==id2 || u == k  continue;  end  dist = vipd(id1,id2);  if dist<=5 % 两个会员之间的距离  div = div + nowvip(k,4);  end  end    if div == 0  p0(id1) = 1;  else  div = div + this;  p0(id1) = this / div;  end  end    %% 当前时间段的所有会员取得订单  for j = 1 : nowvipcnt % 依次枚举当前时间段的所有会员    vipid = nowvip(j,1); % 当前会员号    thisviporid = 0; % 初始化当前会员的订单容器  nowvipor = [0,0,0,0,0,0,0];    % 生成当前人选择的订单  for s = 1 : totorcnt  if nowor(s , 6) == 1  continue;  end    D = viptoor(vipid , nowor(s , 1));    if D <= 30 % average  id = nowor(s,1);  thisviporid = [thisviporid , id];  end    end  thisviporid(1) = []; % 去除当前会员容器头部  if isempty(thisviporid) % 当前会员周围无可接订单，直接跳过当前会员  continue;  end  thisvipor = [0,0,0,0,0,0];  len = length(thisviporid); %    for s = 1:len  id = thisviporid(s);  thisvipor = [thisvipor; nowor(id , :)];  end  thisvipor(1,:) = [];    for s = 1:len  id = thisvipor(s,1);  D = viptoor(vipinc(j,1),id);  D = 210 - D;  thisvipor(s,7) = thisvipor(s , 4) \* 1000 + D;  end    thisvipor = sortrows(thisvipor , -7); % 订单首先按照价格其次按照距离进行排序  view = 0;    last = vipinc(j,4);  last = min(vipmaxor , last); % 当前人能够抢到的单数    %% 贪心首先选择价格高的和距离近的遍历进行选择  leng = size(thisvipor,1);  for q = 1:leng % 遍历所有订单判断距离  id = thisvipor(q,1);  if nowor(id,6) == 1 % 当前的订单已经被人选择过  continue;  end    %% 计算抢到当前单的概率  p1 = p0(vipid);% 计算抢到订单的概率  t1 = rand(1);  %% 当前订单能够抢到，计算完成的概率做标记  if(t1 <= p1) % 当前订单被当前会员抢走更新  nowor(id,6) =1;  last = last - 1;  nowvipor = [nowvipor;thisvipor(q , :)];  if last <=0  break;  end  end  end    nowvipor(1,:) = [];  noworcnt = size(nowvipor,1);  %% 遍历当前人选择的所有订单，计算每个订单的完成概率  for k = 1:noworcnt  % 根据价格计算一个概率  orid = nowvipor(k,1);  x = nowvipor(k,3); y =nowvipor(k,2);  p23=1;  for r = 1:3  if x >= border(r,1) && x<= border(r,2)  if y >=border(r,3) && y<=border(r,4)  p23 = pb(r);  end  end  end  nowprice = oldprice(orid);  if nowprice <= 70 % 根据价格生成一个完成概率  p21= 0.7;  elseif nowprice >70 && nowprice <=75  p21 = 0.8;  elseif nowprice >75 && nowprice <=80  p21= 0.9;  else  p21= 0.98;  end    % 根据距离计算一个概率  dis = viptoor(vipid,orid);  if dis >=25 && dis<30  p22 = 0.9;  elseif dis >=20 && dis<25  p22 = 0.92;  elseif dis >=15 && dis<20  p22 = 0.94;  elseif dis >=10 && dis<=15  p22 = 0.96;  else  p22 = 0.98;  end    p2= p21\*p22\*p23;  t2=rand(1); % 当前任务完成的概率    if(t2 <= p2)  complete(orid) = 1;  else  complete(orid) = 0;  end  end  end  end  ans = sum(complete)  tmp = finor;  tmp(:,5) = complete;  tmp = sortrows(tmp,5);  idx= find(tmp(:,5) ==1);  sort(idx); idx = idx(1);idx = idx-1;  tmp1 = tmp(1:idx,:);  tmp2 = tmp(idx+1:end,:);  figure(2);  plot(tmp1(:,3),tmp1(:,2),'kx','MarkerSize',7);  hold on;  plot(tmp2(:,3),tmp2(:,2),'o','MarkerSize',5,'MarkerFaceColor',[36,169,255]/255,'MarkerEdgeColor',[36,169,255]/255);  grid on ;  xlabel('经度');ylabel('纬度');  legend('未完成的任务','完成的任务');  end  com = com / 1000  rate = com / 835 | | | | | |
| 程序编号 | 2-4 | 文件名称 | Main2\_3.m | 程序说明 | 蒙特卡洛模拟新价格 |
| clc,clear,close;  load data.mat;  load newpri.mat;  com = 0;  % tmp = sortrows(finor,5);  % idx= find(tmp(:,5) ==1);  %  % sort(idx); idx = idx(1);idx = idx-1;  %  % tmp1 = tmp(1:idx,:);  % tmp2 = tmp(idx+1:end,:);  % figure(1);  %  % plot(tmp1(:,3),tmp1(:,2),'kx','MarkerSize',7);  % hold on;  % plot(tmp2(:,3),tmp2(:,2),'o','MarkerSize',5,'MarkerFaceColor',[36,169,255]/255,'MarkerEdgeColor',[36,169,255]/255);  % grid on ;  % xlabel('经度');ylabel('纬度');  % legend('未完成的任务','完成的任务');  for round =1:1000  [vipinc,id] = sortrows(vip,5);  time = vipinc(:,5);  time = unique(time);  finor = new;  timeid = zeros(length(time),2);  N = length(time);  totvipcnt = size(vip,1);  vipmaxor = 30; % 会员可以选取的最大订单数量  R = 57; % 会员可以选择的任务距离半径  totorcnt = size(finor,1);  nowor = finor;  tmp = zeros(totorcnt,1);  nowor=[nowor,tmp];  complete = zeros(1,totorcnt); % 统计最后每个订单的完成情况  oldprice = finor(:,4);  border = [113.759033,114.444305,22.502661,22.758453; % 深圳  113.023224,113.230591,22.928042,23.054462; % 广州  113.230591,113.432465,23.063307,23.248917]; % 佛山  % 三个发达城区对任务完成与否的影响  pb = [0.7,0.8,0.9];  %% 得到同一时间所有会员的起始下标和终止下标  for i=1:N  tmpid = find(vipinc(:,5) == time(i));  tmpid = sort(tmpid);  timeid(i,1) = time(1);  timeid(i,2) = tmpid(1);  timeid(i,3) = tmpid(end);  end  %% 模拟会员选择过程  %% 按照时间段去依次枚举  for i =1:N %枚举不同的时间段    st = timeid(i,2); ed = timeid(i,3); % 当前时间段在时间有序vip序列中的起始位置  nowvip = vipinc(st:ed,:);  nowvip = sortrows(nowvip,-6); % 按照信誉度降序排序  nowvipcnt = size(nowvip,1); % 当前时间段的会员数量    %% 获得当前人选择订单成功的概率  p0 =zeros(1 , totvipcnt);  for u = 1 : nowvipcnt    id1 = nowvip(u,1) ;  div = 0;  this = nowvip(u,4);  for k = 1 : nowvipcnt % 求出所有当前订单10km内的当前会员数量    id2 = nowvip(k,1);  if id1==id2 || u == k  continue;  end  dist = vipd(id1,id2);  if dist<=3 % 两个会员之间的距离  div = div + nowvip(k,4);  end  end    if div == 0  p0(id1) = 1;  else  div = div + this;  p0(id1) = this / div;  end  end    %% 当前时间段的所有会员取得订单  for j = 1 : nowvipcnt % 依次枚举当前时间段的所有会员    vipid = nowvip(j,1); % 当前会员号  thisviporid = 0; % 初始化当前会员的订单容器  nowvipor = [0,0,0,0,0,0,0];    % 生成当前人选择的订单  for s = 1 : totorcnt  if nowor(s , 6) == 1  continue;  end    D = viptoor(vipid , nowor(s , 1));    if D <= 30 % average  id = nowor(s,1);  thisviporid = [thisviporid , id];  end    end  thisviporid(1) = []; % 去除当前会员容器头部  if isempty(thisviporid) % 当前会员周围无可接订单，直接跳过当前会员  continue;  end  thisvipor = [0,0,0,0,0,0];  len = length(thisviporid); %    for s = 1:len  id = thisviporid(s);  thisvipor = [thisvipor; nowor(id , :)];  end  thisvipor(1,:) = [];    for s = 1:len % 当前会员订单处理后按照价格、距离进行排序  id = thisvipor(s,1);  D = viptoor(vipinc(j,1),id);  D = 210 - D;  thisvipor(s,7) = thisvipor(s , 4) \* 1000 + D;  end    thisvipor = sortrows(thisvipor , -7); % 订单首先按照价格其次按照距离进行排序  view = 0;    last = vipinc(j,4);  last = min(vipmaxor , last); % 当前人能够抢到的单数    % 贪心首先选择价格高的和距离近的遍历进行选择  leng = size(thisvipor,1);  for q = 1:leng % 遍历所有订单判断距离  id = thisvipor(q,1);  if nowor(id,6) == 1 % 当前的订单已经被人选择过  continue;  end  % 计算抢到当前单的概率  p1 = p0(vipid);% 计算抢到订单的概率  t1 = rand(1);  %% 当前订单能够抢到，计算完成的概率做标记  if(t1 <= p1) % 当前订单被当前会员抢走更新  nowor(id,6) =1;  last = last - 1;  nowvipor = [nowvipor;thisvipor(q , :)];  if last <=0  break;  end  end  end    nowvipor(1,:) = [];  noworcnt = size(nowvipor,1);  %% 遍历当前人选择的所有订单，计算每个订单的完成概率  for k = 1:noworcnt  % 根据价格计算一个概率  orid = nowvipor(k,1);  x = nowvipor(k,3); y =nowvipor(k,2);  p23=1;  for r = 1:3  if x >= border(r,1) && x<= border(r,2)  if y >=border(r,3) && y<=border(r,4)  p23 = pb(r);  end  end  end  nowprice = oldprice(orid);  if i <=11 && i>=1  nowprice = nowprice \* 1.1;  elseif i == 31  nowprice = nowprice \* 0.95;  end  if nowprice >85  nowprice = 85;  elseif nowprice <65  nowprice = 65;  end  if nowprice <= 70 % 根据价格生成一个完成概率  p21= 0.7;  elseif nowprice >70 && nowprice <=75  p21 = 0.8;  elseif nowprice >75 && nowprice <=80  p21= 0.9;  else  p21= 0.98;  end    % 根据距离计算一个概率  dis = viptoor(vipid,orid);  if dis >=25 && dis<30  p22 = 0.9;  elseif dis >=20 && dis<25  p22 = 0.92;  elseif dis >=15 && dis<20  p22 = 0.94;  elseif dis >=10 && dis<=15  p22 = 0.96;  else  p22 = 0.98;  end    p2= p21\*p22\*p23;  t2=rand(1); % 当前任务完成的概率    if(t2 <= p2)  complete(orid) = 1;  else  complete(orid) = 0;  end  end  end  end  com = com + sum(complete);  % tmp = finor;  % tmp(:,5) = complete;  % tmp = sortrows(tmp,5);  % idx= find(tmp(:,5) ==1);  %  % sort(idx); idx = idx(1);idx = idx-1;  %  % tmp1 = tmp(1:idx,:);  % tmp2 = tmp(idx+1:end,:);  % figure(2);  % plot(tmp1(:,3),tmp1(:,2),'kx','MarkerSize',7);  % hold on;  % plot(tmp2(:,3),tmp2(:,2),'o','MarkerSize',5,'MarkerFaceColor',[36,169,255]/255,'MarkerEdgeColor',[36,169,255]/255);  % grid on ;  % xlabel('经度');ylabel('纬度');  % legend('未完成的任务','完成的任务');  end  com = com / 1000  rate = com / 835 | | | | | |
| 程序编号 | 3 | 文件名称 | Main3.m | 程序说明 | 按照距离聚类并模拟 |
| clc,clear,close;  load data.mat;  load newpri.mat;  n = size(finor,1);  st = zeros(1,n);  finor = new;  result = zeros(140,6);  [vipinc,id] = sortrows(vip,5);  time = vipinc(:,5);  time = unique(time);  totvipcnt = size(vip,1);  timeid = zeros(length(time),2);  N = length(time); % 不同时间段的数量  border = [113.759033,114.444305,22.502661,22.758453; % 深圳  113.023224,113.230591,22.928042,23.054462; % 广州  113.230591,113.432465,23.063307,23.248917]; % 佛山  % 三个发达城区对任务完成与否的影响  pb = [0.7,0.8,0.9];  vipmaxor = 30;  maxorder = 0;  maxfun = 0;  com = 0;  for times = 1:100 % 枚举不同的聚类情况  st = zeros(n,1);  order = [0,0,0,0,0]; % 纬度、经度、价格、数量    %% 对订单按照距离进行聚类  for i = 1 : n  if st(i) == 1  continue;  end    price = finor(i,4);  lon = finor(i,3); lat = finor(i,2);  id = i;  cnt = 1;  for j =1 : n  if cnt > 5 % 订单聚类数量超过  break;  end  if i == j  continue;  end  if st(j) == 1  continue;  end    if d(i,j) <= 4  cnt = cnt +1;  price = price + finor(j,4);  st(i) = 1; st(j) = 1;  end  end  tmp = [id,lon,lat,price,cnt];  order = [order;tmp];  end  % 得到当前聚类完成你的订单  order(1,:) = [];  totcnt = size(order,1);  tmp = zeros(totcnt,1);  order= [order,tmp,tmp];  complete = zeros(1,totcnt); % 统计最后每个订单的完成情况  for i = 1:totcnt  order(i,7) = i;  end  len = size(order,1);  %% 画图可视化  %% 得到同一时间所有会员的起始下标和终止下标    for i = 1 : N  tmpid = find(vipinc(:,5) == time(i));  tmpid = sort(tmpid);  timeid(i , 1) = time(1);  timeid(i , 2) = tmpid(1);  timeid(i , 3) = tmpid(end);  end      %% 求解当前的聚类订单结果    for i =1:N %枚举不同的时间段  st = timeid(i,2); ed = timeid(i,3); % 当前时间段在时间有序vip序列中的起始位置  nowvip = vipinc(st:ed,:);  nowvip = sortrows(nowvip,-6); % 按照信誉度降序排序  nowvipcnt = size(nowvip,1); % 当前时间段的会员数量  %% 获得当前人选择订单成功的概率  p0 =zeros(1 , totvipcnt);  for u = 1 : nowvipcnt  id1 = nowvip(u,1) ;  div = 0;  this = nowvip(u,4);  for k = 1 : nowvipcnt % 求出所有当前订单10km内的当前会员数量  id2 = nowvip(k,1);  if id1==id2 || u == k  continue;  end  dist = vipd(id1,id2);  if dist<=3 % 两个会员之间的距离  div = div + nowvip(k,4);  end  end  if div == 0  p0(id1) = 1;  else  div = div + this;  p0(id1) = this / div;  end  end    %% 当前时间段的会员  for j = 1 : nowvipcnt % 依次枚举当前时间段的所有会员  vipid = nowvip(j,1); % 当前会员号  thisviporid = 0; % 初始化当前会员的订单容器  nowvipor = [0,0,0,0,0,0,0]; % 原id、经度、纬度、总价格、数量、是否选择、id  % 生成当前人选择的订单  viporcnt = 0;  for s = 1 : totcnt  if order(s , 6) == 1 % 当前聚类的任务已经被人选择  continue;  end  D = viptoor(vipid , order(s , 1)); % 当前的会员到当前缩聚点的距离  if D <= 30  id = order(s,7); % 当前聚为一类的代表编号  thisviporid = [thisviporid , id];  end  end    thisviporid(1) = []; % 去除当前会员容器头部  if isempty(thisviporid) % 当前会员周围无可接订单，直接跳过当前会员  continue;  end    thisvipor = [0,0,0,0,0,0,0]; % 原id、经度、纬度、总价格、数量、是否选择、id  len = length(thisviporid);  for s = 1:len  id = thisviporid(s);  thisvipor = [thisvipor; order(id , :)];  end    thisvipor(1,:) = [];    for s = 1:len % 当前会员订单处理后按照价格、距离进行排序  id = thisvipor(s,1);  D = viptoor(vipid,id);  D = 210 - D;  thisvipor(s,8) = thisvipor(s , 4) \* 10000 + D;  end  thisvipor = sortrows(thisvipor , -8); % 订单首先按照价格其次按照距离进行排序  thisvipor(:,8) = [];  view = 0;  last = vipinc(j,4);  last = min(vipmaxor , last); % 当前人能够抢到的单数    leng = size(thisvipor,1);    for q = 1:leng % 遍历所有订单判断距离  id = thisvipor(q,7);  if order(id,6) == 1 % 当前的订单已经被人选择过  continue;  end  % 计算抢到当前单的概率  p1 = p0(vipid);% 计算抢到订单的概率  t1 = rand(1);  % 当前订单能够抢到，计算完成的概率做标记  if(t1 <= p1) % 当前订单被当前会员抢走更新  order(id,6) =1;  if last < order(id,5)  continue;  end  last = last - order(id,5);  nowvipor = [nowvipor;thisvipor(q , :)];  if last <=0  break;  end  end  end    nowvipor(1,:) = [];  noworcnt = size(nowvipor,1);  %% 遍历当前人选择的所有订单，计算每个订单的完成概率  for k = 1:noworcnt  % 根据价格计算一个概率  orid = nowvipor(k,7);  thiscnt = nowvipor(k,5);  x = nowvipor(k,2); y =nowvipor(k,3);  p23=1;  for r = 1:3  if x >= border(r,1) && x<= border(r,2)  if y >=border(r,3) && y<=border(r,4)  p23 = pb(r);  end  end  end  nowprice = order(orid,4);  if i <=11 && i>=1  nowprice = nowprice \* 1.1;  elseif i == 31  nowprice = nowprice \* 0.95;  end  if thiscnt >1 && thiscnt <= 6  nowprice = nowprice \* (1-0.1\*(thiscnt-1));  end  nowprice = nowprice / thiscnt;  if nowprice >85  nowprice = 85;  elseif nowprice <65  nowprice = 65;  end  if nowprice <= 70 % 根据价格生成一个完成概率  p21= 0.7;  elseif nowprice >70 && nowprice <=75  p21 = 0.8;  elseif nowprice >75 && nowprice <=80  p21= 0.9;  else  p21= 0.98;  end  orid = nowvipor(k,1);  % 根据距离计算一个概率  dis = viptoor(vipid,orid);  if dis >=25 && dis<30  p22 = 0.9;  elseif dis >=20 && dis<25  p22 = 0.92;  elseif dis >=15 && dis<20  p22 = 0.94;  elseif dis >=10 && dis<=15  p22 = 0.96;  else  p22 = 0.98;  end  p2= p21\*p22\*p23;  t2=rand(1); % 当前任务完成的概率  orid = nowvipor(k,7);  if(t2 <= p2)  complete(orid) = 1;  else  complete(orid) = 0;  end  end  end  end    for i = 1:totcnt  if complete(i) == 1  com = com + order(i,5);  end  end  end  figure(1);  plot(finor(:,3),finor(:,2),'o','markersize',8,'markerfacecolor',[36,169,255]/255);  hold on ;  plot(order(:,2),order(:,3),'o','markersize',10,'markerfacecolor','r');  hold on;  com = com / 100 | | | | | |