大学 计算机科学与技术 学院

数据分析 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号： | 姓名： | | 班级： |
| 实验题目：gps | | | |
| 实验学时： | | 实验日期： 2018年1月20日 | |
| 实验目的：  1 通过对trajectory或point进行聚类，划分出样本为若干活动热区  聚类的数据点选取方法有两种方案，可选择任意一种：   1. 以单个trajectory为计算单位(以go\_track\_trackspoints.csv 中track\_id为关键字)，该方法需要计算trajectory的代表节点（计算方法自行决定） 2. 以trajectory中的每个point为计算单位   2 聚类方法采用密度聚类  3 把聚类结果以图形化方式展示出来  4 从空间和时间上分析运行效率  5 分析算法的优缺点，提出改进的方向 | | | |
| 软件环境：  Ubuntu16.04、python | | | |
| 实验步骤与内容：   1. **以trajectory中的每个point为计算单位**，聚类方法采用密度聚类，使用DBSCAN算法对数据进行聚类。   1）算法伪代码：  首先将数据集D中的所有对象标记for（数据集D中每个对象p） do  if （p已经归入某个簇或标记为噪声） then  continue;  else  检查对象p的Eps邻域 NEps(p) ；  if (NEps(p)包含的对象数小于MinPts) then  标记对象p为边界点或噪声点；  else  标记对象p为核心点，并建立新簇C, 并将p邻域内所有点加入C  for (NEps(p)中所有尚未被处理的对象q) do  检查其Eps邻域NEps(q)，若NEps(q)包含至少MinPts个对象，则将NEps(q)中未归入任何一个簇的对象加入C；  end for  end if  end if  end for  2）算法代码：  def dbscan(D,e,m):  # core T,k=0 cluster,c=[], unvisited p，T用来存储核心点  T=set()  k=0  C=[]  P=set(D)  for d in D:  if( len([i for i in D if distance(d,i)<=e])>=m):  T.add(d)  # start cluster  while len(T):  p\_old=P  o=list(T)[np.random.randint(0, len(T))]#随机选择一个核心点  P=P-set(o)  Q=[]#Q对象o的邻域集合  Q.append(o)  while len(Q):  q=Q[0]  nq=[i for i in D if distance(q,i)<=e]  if len(nq)>=m:  S=P&set(nq)  Q+=list(S)  P=P-S  Q.remove(q)  k+=1  Ck=list(p\_old-P)#ck为一个聚类  T=T-set(Ck)  C.append(Ck)  return C  3）20个数据对象的处理过程：  UI%MGUL)Q28}@WPJBG[E[(M   1. **数据的分类结果**   使用Haversine公式将经纬度转换为球面距离，使用球面距离作为距离衡量。算法输入为待聚类数据，输出为一个列表集合，每个列表代表一簇。共分成52簇：    **3.数据图形化显示**  （聚类半径，聚类点数）=（0.1,20）：    **4.时空复杂度分析**  （1）时间复杂度：O（n\*n），在低维空间数据中,有一些数据结构如KD树，使得可以有效的检索特定点给定距离内的所有点，时间复杂度可以降低到O(NlogN)。  （2）空间复杂度：O(n)，对于每个点它只需要维持少量数据，即簇的标号和每个点的标识(核心点或边界点或噪音点)。   1. **算法优缺点的分析，及改进意见**   优点：  不需要输入划分的聚类个数;聚类簇的形状没有偏差;可以在需要时，输入过滤噪声的参数.还可以处理任意形状和大小的簇。  缺点：  （1）当数据量增大时, 要求较大的内存支持, I/ 0 消耗也很大。  （2）当空间聚类的密度不均匀, 聚类间距离相差很大时, 聚类的质量较差。  改进意见：  DBSCAN 算法可以看出, 全局变量 Eps 值影响了聚类质量, 尤其是数据分布不均匀时. 因此, 考虑对数据进行划分, 每一个划分中的数据分布相对较均匀, 根据每个划分中数据的分布密集程度来选取 Eps 值. 这样一方面降低了全局变量 Eps 值的影响, 另一方面由于具有多个划分, 就可以考虑并行处理, 从而提高聚类效率, 也降低了 DBSCAN 算法对内存的较高要求。  **6.程序源代码**：  import pandas as pd  import numpy as np  import pylab as pl  import math  #read csv and return list  #读取数据函数  def readcsv():  marks=pd.read\_csv("/home/tsinghua/Desktop/go\_track\_trackspoints.csv")#使用pandas中的read\_csv方#法  #print(marks)  list=[]  #将读出的dataframe对象转换为元组列表  for index in marks.index:  list.append((marks["latitude"][index],marks["longitude"][index]))  return list  # acculate distance  #计算距离，使用Haversine公式  def distance(p1,p2):  lat1, lon1 = p1  lat2, lon2 = p2  lon1, lat1, lon2, lat2 = map(math.radians, [lon1, lat1, lon2, lat2])  dlon = lon2 - lon1  dlat = lat2 - lat1  a = math.sin(dlat / 2) \*\* 2 + math.cos(lat1) \* math.cos(lat2) \* math.sin(dlon / 2) \*\* 2  c = 2 \* math.asin(math.sqrt(a))  r = 6371 # Radius of earth in kilometers. Use 3956 for miles  return c \* r  #dbscan算法，小字母为对象，大写字母为集合  def dbscan(D,e,m):  # core T,k=0 cluster,c=[], unvisited p，T用来存储核心点  T=set()  k=0  C=[]  P=set(D)  for d in D:  if( len([i for i in D if distance(d,i)<=e])>=m):  T.add(d)  # start cluster  while len(T):  p\_old=P  o=list(T)[np.random.randint(0, len(T))]#随机选择一个核心点  P=P-set(o)  Q=[]#Q对象o的邻域集合  Q.append(o)  while len(Q):#开始从一个核心点向邻域扩张  q=Q[0]  nq=[i for i in D if distance(q,i)<=e] #q的邻域  if len(nq)>=m:#如果q是核心点，将q的邻域也并入进来  S=P&set(nq)  Q+=list(S)  P=P-S  Q.remove(q)  k+=1  Ck=list(p\_old-P)  T=T-set(Ck)  C.append(Ck)  return C  #图形化显示，使用pyplot  def draw(c):  colvalue=['r', 'y', 'g', 'b', 'c', 'k', 'm']  for i in range(len(c)):  coo\_x = []  coo\_y = []  for j in range(len(c[i])):  coo\_x.append(c[i][j][0])  coo\_y.append(c[i][j][1])  pl.scatter(coo\_x, coo\_y, marker='x', color=colvalue[i % len(colvalue)], label=i)  pl.legend(loc='upper right')  pl.show()  #m=distance((-10.93934139,-37.06274211),(-10.93921056,-37.06284305))  #print m #test  d=readcsv()  print(len(d))  C = dbscan(d, 0.1,20)  draw(C)   1. **参考资料：**   [1]python菜鸟教程  [2]pyplot：<https://matplotlib.org/users/pyplot_tutorial.html>  [3]Pandas:http://wiki.jikexueyuan.com/project/start-learning-python/312.html  [4]《机器学习》 周志华 | | | |