基于 K-Means 的居民地理位置保护方案

数据安全与隐私保护大作业

一、需求分析

1. 思路提出

本次选题思路来自课件第 5 章——隐私保护技术。课件中提到，个人用户最基本的隐私保护需求包括：身份隐私、属性隐私、社会关系隐私和**位置隐私**。我对其中的**用户位置隐私保护**非常感兴趣，因为这是最近非常热门的一个话题：许多手机 app 都要求用户打开地理位置服务、许多社会福利部门也要求居民提供家庭住址。这一方面有助于服务方分析数据、改进服务；另一方面却让用户有一种被“把控”、担心隐私泄露的感觉。经过一系列思考，我打算利用课件中老师介绍的 K-Means 聚类算法，来保障用户的地理位置隐私。

2. 背景设定

A 镇居民居住分布如图 1。以镇中心为原点建立直角坐标，数据集中每行的两个数据分别作为 X 值、Y 值；为方便做图，在数据集中直接加入了正负号表示方向（正负号只代表方向，绝对值代表距离）。

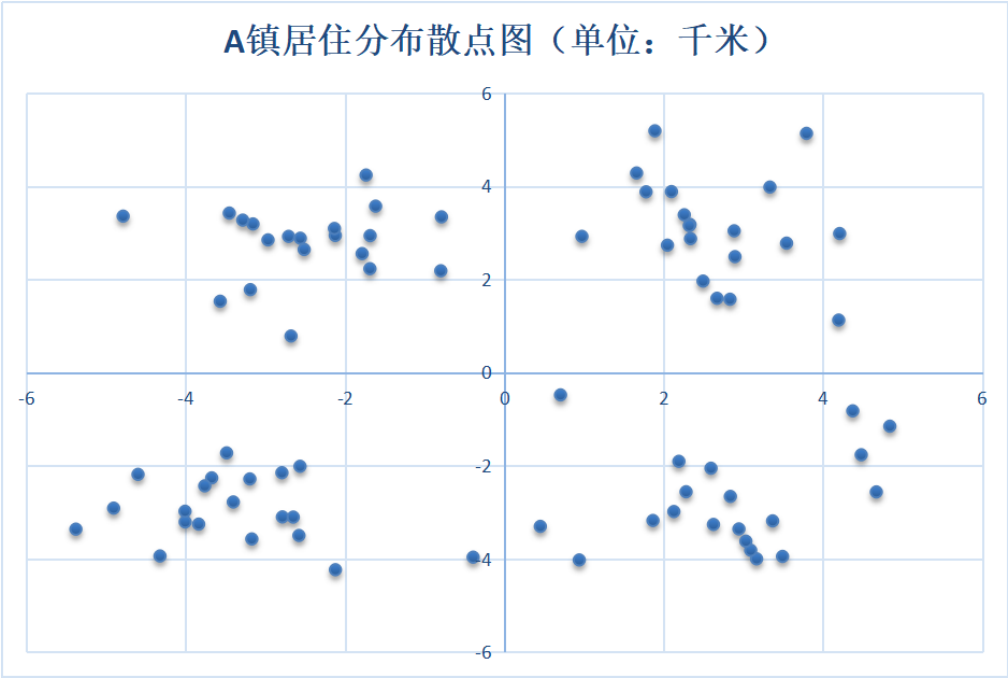


图 1：A 镇居民居住分布散点图

现 A 镇政府拟为居民新增一些公交站点，派遣交通公司调查。但交通公司一方面不知道公交站点安排在哪个位置合适；另一方面，居民也不愿意把家庭住址交给外来公司。

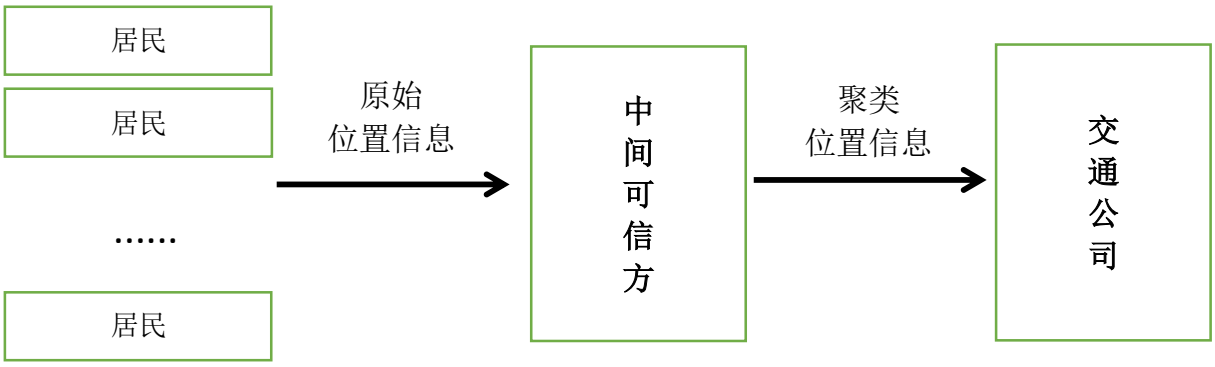
3. 需求分析

需求分析	解决方法
居民担心位置泄露	添加中央可信方收集、管理真实位置数据，交通公司无法拿到居民真实地址
交通公司存在主动/被动 泄露数据的可能性	中央可信方只交给其 K-Means 方法 处理后的数据，而非真实数据
交通公司无法判断 公交站的安置点	K-Means 方法除了能保护隐私，还能 确定“簇心”，即可能的安置点

经上述简要分析，可知算法需要达到的基本目标为：

- **隐藏用户真实位置信息**
- **输出簇心相关信息**

二、算法设计



1. 输入过程：

输入 80 组二维数据（每行两个数据分别对应位置的 X 坐标、Y 坐标），这是用户的真实地理位置数据。

2. 中间过程：

中间可信方是一个受信任的服务器，它负责接收居民的真实地理坐标，然后进行 K-Means 聚类运算，并且不断迭代到可接受的误差率内。

3. 输出过程：

中间可信方在 K-Means 运算结束后，仅输出簇心的坐标到交通公司。这样既为交通公司提供了选址策略，又保护了用户真实位置信息不被泄露。

三、实现源码

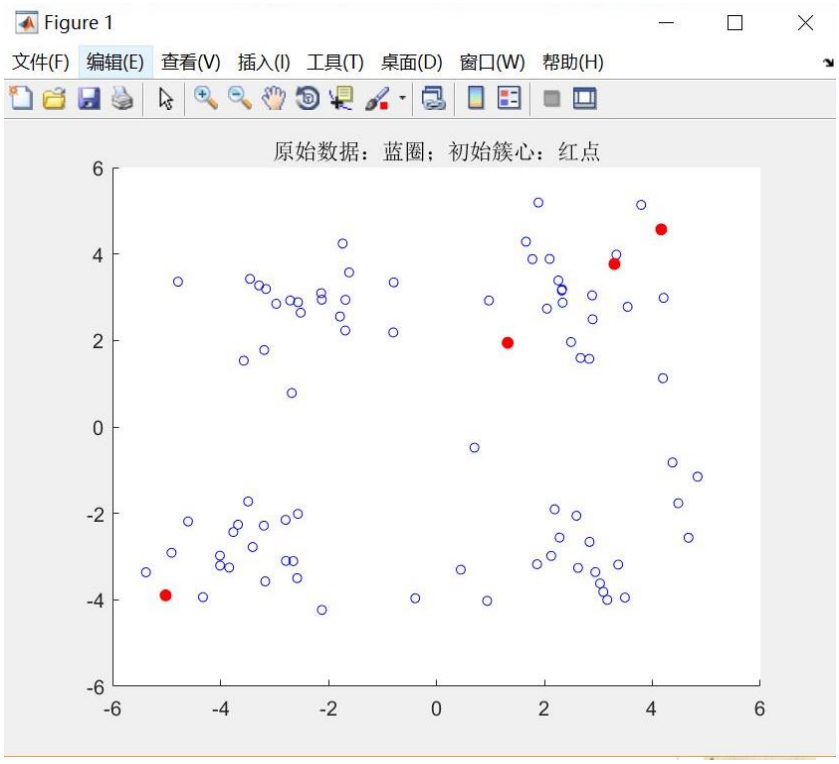
（实现环境：Windows 10 + Matlab 2016）

```
% 簇心数目 k
K = 4;

% 准备数据，假设是 2 维的,80 条数据，从 data.txt 中读取
%data = zeros(100, 2);
load 'data.txt'; % 直接存储到 data 变量中

x = data(:,1);
y = data(:,2);

% 绘制数据，2 维散点图
% x,y: 要绘制的数据点  20:散点大小相同，均为 20  'blue':散点颜色为蓝色
s = scatter(x, y, 20, 'blue');
title('原始数据：蓝圈；初始簇心：红点');

% 初始化簇心
sample_num = size(data, 1);            % 样本数量
sample_dimension = size(data, 2); % 每个样本特征维度


% 簇心赋初值：计算所有数据的均值，并将一些小随机向量加到均值上
clusters = zeros(K, sample_dimension);
minVal = min(data); % 各维度计算最小值
maxVal = max(data); % 各维度计算最大值

for i=1:K
    clusters(i, :) = minVal + (maxVal - minVal) * rand();
end


hold on; % 在上次绘图（散点图）基础上，准备下次绘图
% 绘制初始簇心
scatter(clusters(:,1), clusters(:,2), 'red', 'filled'); % 实心圆点，表示簇心初始位置

c = zeros(sample_num, 1); % 每个样本所属簇的编号

PRECISION = 0.001;
```

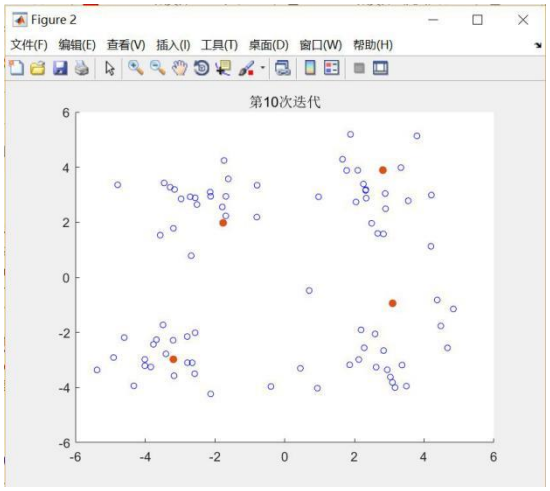
```
iter = 100; % 假定最多迭代 100 次
basic_eta = 0.1;   % 学习率
for i=1:iter
    pre_acc_err = 0;   % 上一次迭代中，累计误差
    acc_err = 0;   % 累计误差
    for j=1:sample_num
        x_j = data(j, :);        % 取得第 j 个样本数据

        % 所有簇心和 x 计算距离，找到最近的一个（比较簇心到 x 的模长）
        gg = repmat(x_j, K, 1);
        gg = gg - clusters;
        tt = arrayfun(@(n) norm(gg(n,:)), (1:K)');
        [minVal, minIdx] = min(tt);

        % 更新簇心：把最近的簇心(winner)向数据 x 拉动。 eta 为学习率.
        eta = basic_eta/i;
        delta = eta*(x_j-clusters(minIdx,:));
        clusters(minIdx,:) = clusters(minIdx,:) + delta;
        acc_err = acc_err + norm(delta);
    end

    if(rem(i,10) ~= 0)
        continue
    end
    figure;
    f = scatter(x, y, 20, 'blue');
    hold on;
    scatter(clusters(:,1), clusters(:,2), 'filled'); % 实心圆点，表示簇心初始位置
    title(['第', num2str(i), '次迭代']);
    if (abs(acc_err-pre_acc_err) < PRECISION)
        disp(['收敛于第', num2str(i), '次迭代']);
        break;
    end
     disp((clusters(:,1)));
     disp((clusters(:,2)));

    disp(['累计误差：', num2str(abs(acc_err-pre_acc_err))]);
    pre_acc_err = acc_err;
end


disp('done');
```

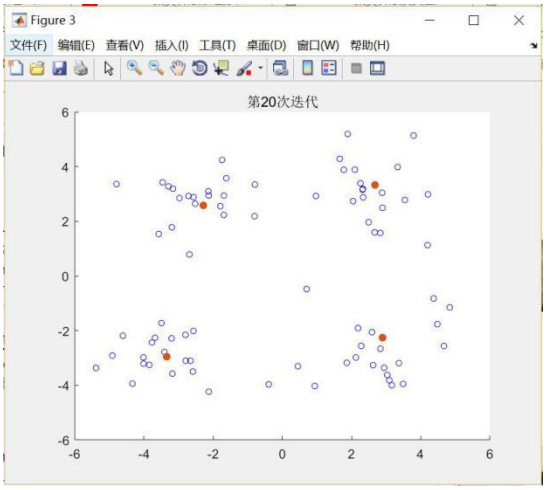
四、效果展示

1. 初始状况



2. 第 10-100 次迭代





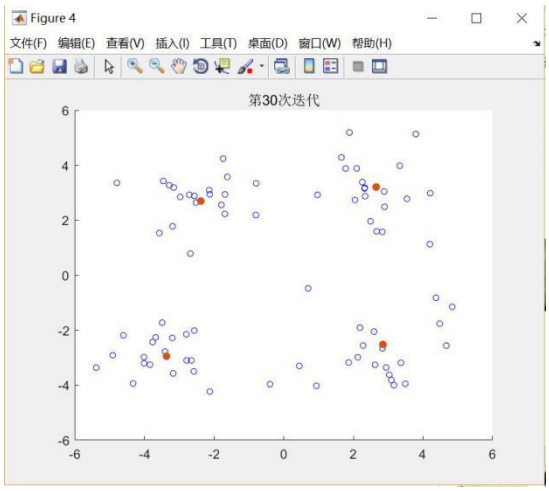


Figure 4
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
第30次迭代

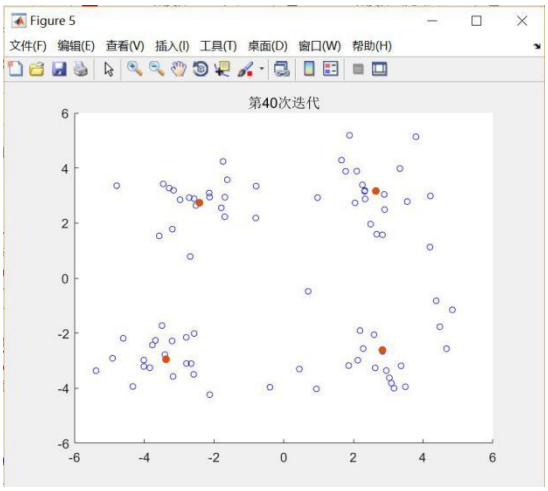


Figure 5
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
第40次迭代

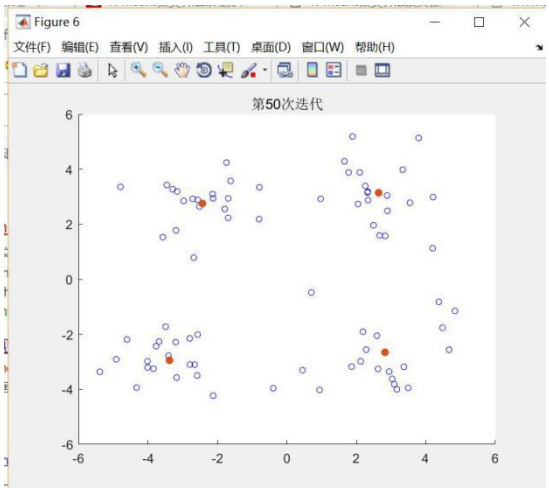


Figure 6
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
第50次迭代

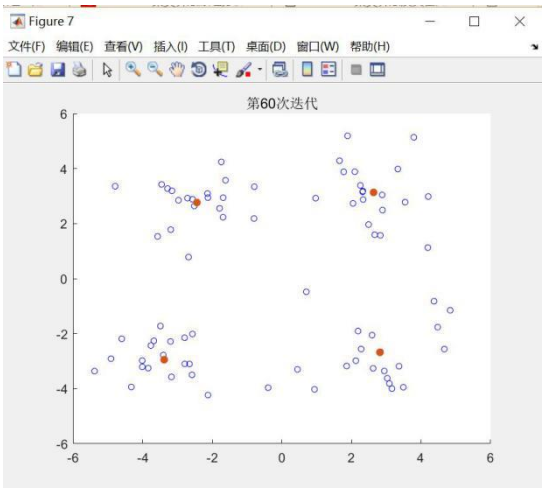


Figure 7
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
第60次迭代

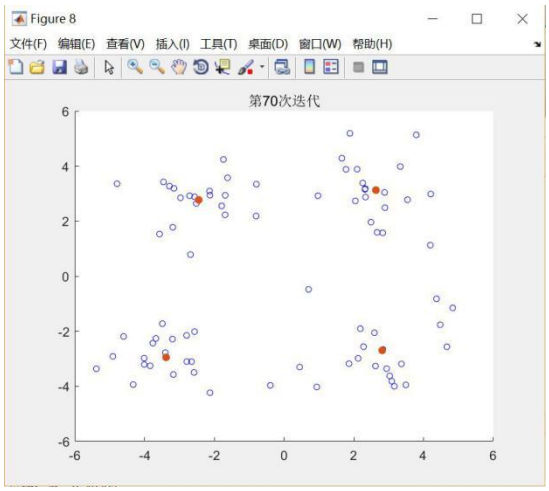


Figure 8
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
第70次迭代

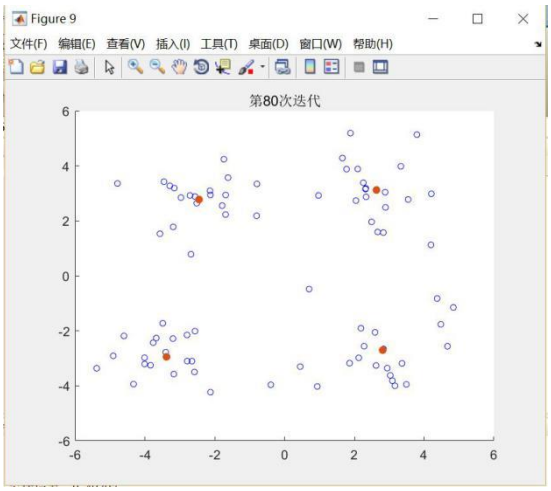
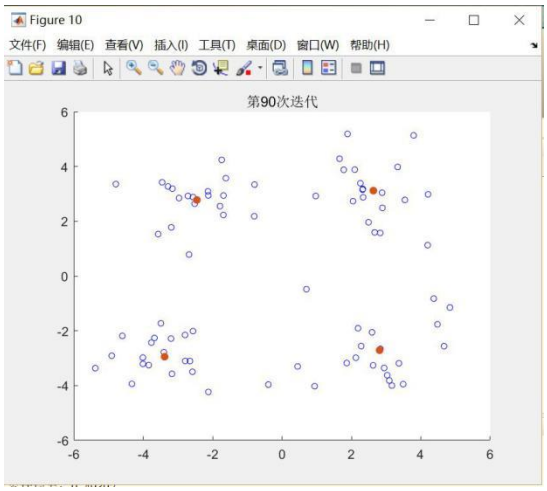
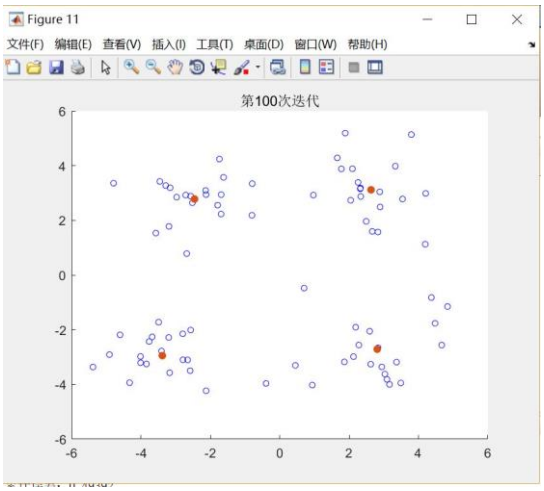


Figure 9
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
第80次迭代

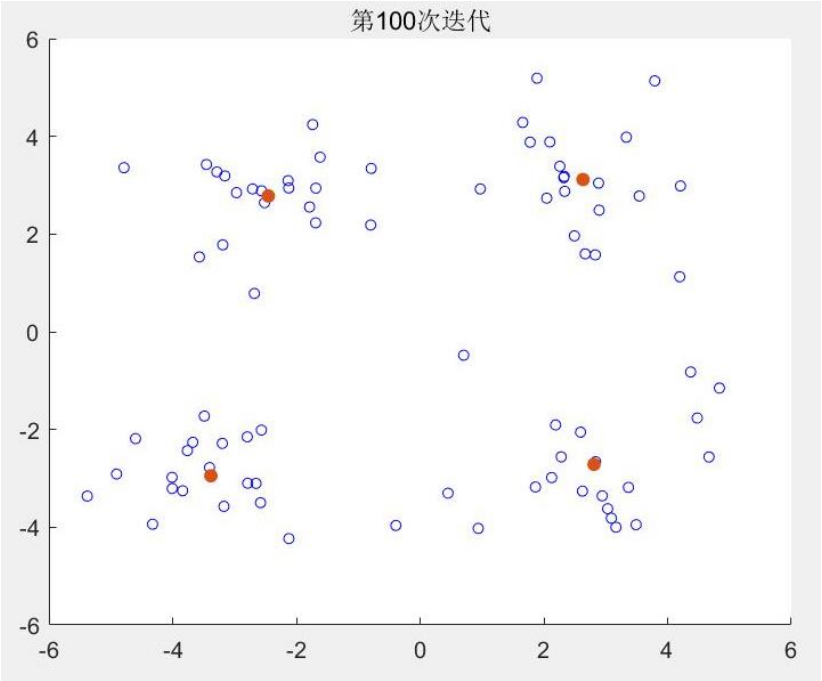




3. 误差分析

累计误差：1.3139
累计误差：0.49392
累计误差：0.31774
累计误差：0.23602
累计误差：0.1881
累计误差：0.15649
累计误差：0.13401
累计误差：0.1172
累计误差：0.10414
累计误差：0.093698

4. 最终结果（图示与坐标）



-2.4615　　2.7874

2.6265　　3.1087

-3.3824　　-2.9473

2.8029　　-2.7315

五、方案评价

1. 优点

(1) 算法适合

K-Means 算法早已被广泛应用在地理位置分析中，比如滴滴、Uber 的客流量集中点分析以及各类快递的收发站安排，许多工程都证明了 K-Means 非常适合大量的、离散的数据进行数据分析。

(2) 有效保护

K-Means 非常重要的特征就是为数据分配簇，其簇心值代表了平均值。在本作品中，提交给交通公司的只有簇心值，而不是真正的具体地址值，这有效的防止了交通公司内部盗取或泄露用户隐私。

2. 缺点

(1) 数据量小

数据分析精度常常和数据量有很大关系，但由于时间有限，我没能发现数据数量、质量更合适的数据集。因此可以说本作品的结果仅供模拟参考，而与真实情况有一定差距。

(2) K 值选取

观察本作品的数据集作出的散点图，可以直接确定 K 值为 4。但在现实生活中，大量数据的情况下可能无法直接判断 K 的取值。我上网查询了相关资料，发现工业界也没有什么更好的方法直接计算 K 值，一般都是靠工程经验判断 K 值，再反复迭代调整。

附录

（数据集：data.txt）

1.659 4.2851
-3.4537 3.4243
4.8381 -1.1515
-5.3797 -3.3621
0.9726 2.9241
-3.5679 1.5316
0.4506 -3.3022
-3.4871 -1.7244
2.6688 1.5948
-3.1565 3.1911
3.1655 -3.9998
-2.7868 -3.0994
4.2082 2.9849
-2.1233 2.9434
0.7042 -0.4795
-0.3924 -3.9637
2.8317 1.574
-0.7902 3.3431
2.9435 -3.3571
-3.1959 -2.2839
2.3364 2.8751
-1.7863 2.5542
2.1901 -1.906
-3.4034 -2.7783
1.7781 3.8808
-1.6883 2.2303
2.593 -2.0544
-4.0073 -3.2071
2.2577 3.3876
-2.679 0.7851
0.9395 -4.0236
-3.6744 -2.2611
2.0463 2.7353
-3.1895 1.7803
4.3726 -0.8222
-2.5793 -3.4976
1.889 5.1904
-0.7987 2.1856

2.8365	-2.6586
-3.8379	-3.2538
2.0967	3.886
-2.709	2.9239
3.367	-3.1848
-2.1215	-4.2326
2.3295	3.1798
-3.2848	3.2731
3.0914	-3.8152
-3.7621	-2.4322
3.5421	2.7788
-1.7368	4.241
2.1271	-2.9837
-4.3238	-3.9381
3.7921	5.1358
-4.7865	3.3585
2.6241	-3.2607
-4.0093	-2.9781
2.4935	1.9637
-2.5137	2.6422
1.8644	-3.1763
-3.1712	-3.5725
2.8942	2.4891
-2.5625	2.8844
3.4911	-3.9475
-2.5657	-2.0121
3.3329	3.9831
-1.6168	3.5732
2.2806	-2.5594
-2.6512	-3.1032
2.3214	3.155
-1.6857	2.9397
3.031	-3.6203
-4.5996	-2.1858
4.1962	1.1267
-2.1339	3.0937
4.6689	-2.5627
-2.7932	-2.1497
2.8841	3.0434
-2.9676	2.8487
4.4793	-1.7648
-4.9056	-2.9111