Clustering of Data from Four Regions

Illusionna

18:05, Wednesday 2nd August, 2023 --→ 12:10, Friday 4th August, 2023

概述

聚类形式丰富，可以分为有监督的聚类和无监督的聚类，譬如 K-means、GMM、SOM 等. 由于本人电脑性能不足，无法跑程序，学校服务器远程用不来，所以虚拟了一组数据 testData.xlsx 示例，以简单说明最优聚类数目的判断. 其次，由于聚类初始点的随机性，所以很多时候再次执行程序得到的最终聚类标签极大可能不尽相同，甚至所属类别都不同，这是正常的现象.

择优选取聚类数，仅仅我个人而言，之所以采用 TOPSIS 优劣解距离法思想，是因为考虑到 DBI 指数越小簇内性能越好而 DI 指数越大簇外（间）结果越好，而 TOPSIS 应该可以解决这样的问题. 最后强调一下，这仅是一个示例，如果你赞同这种思想，就往下看.

I. 准备数据执行程序

查看 ./**Cluster**/**Programs**/**README.md** 准备好数据（100M 有点大所以我没有打包）.

II. 一处解释

三种聚类方法没有采取训练集和测试集的划分. 如果，我将 testData.xlsx 按照三七开，设置聚类数（假如 n_clusters = 4），去拿训练集训练得到的模型预测测试集，那么，我会得到预测标签结果，就像：

测试集：[2, 1, 0, 0, 3]

这表明，测试集第一行数据（第一个样本）属于第 2 类，第二个样本属于第 1 类，第三个样本属于第 0 类，以此类推. 起源数据的标签只有 0、1 两种（人

为设置的标签，可以看作有监督的），而我们测试集预测的数据却反映 0、1、2、3 四类，显而易见，牛头不对马嘴.

而且，即便设置聚类数 n_clusters = 2，也存在这样一个现象.

$$\text{第一次执行测试集：}[0,\ 1,\ 0,\ 0,\ 1]$$

$$\text{第二次执行测试集：}[1,\ 0,\ 1,\ 0,\ 1]$$

$$\text{第二次执行测试集：}[1,\ 1,\ 0,\ 0,\ 0]$$

在起源数据中，假设我们监督的测试集第一个样本标签是 0，但多次执行程序，未必见得预测的第一个样本就一定隶属第 0 类，这个例子中，第一个样本在三次执行情况下分别隶属第 0、1、1 类.

假设我们人类认为的标签 0 代表 “迦南”，1 代表 “安可”，那就是说，测试集第一个样本被标记为迦南，但现在预测结果反映第一个样本第一次被机器认为是迦南，但第二次第三次被认为是安可.

迦南被认为机器判为迦南 → 安可 → 安可，若我们采用交叉熵评判：

$$\text{accuracy} = \frac{1}{3}$$

但，很可能我下次执行得到：

$$\text{第一次执行测试集：}[0,\ 1,\ 0,\ 0,\ 1]$$

$$\text{第二次执行测试集：}[0,\ 0,\ 1,\ 0,\ 1]$$

$$\text{第二次执行测试集：}[1,\ 1,\ 0,\ 0,\ 0]$$

准确率：

$$\text{accuracy} = \frac{2}{3}$$

更重要的是，这只是测试集一个样本，而测试集有起源数据 ×30% 的量，这会使得多次执行程序得到的交叉熵判断混乱，第一次可能是迦南判为迦南，安可判为安可，到了第二次就变成迦南判为安可，安可判为安可，第一次的 accuracy 和第二次的 accuracy 可能天差地别，交叉熵不稳定.

正是鉴于上面这两种现象，所以没有将起源数据划分训练集测试集，如果，使用者有划分需求，则需要自行补充相关 Python 函数.

III. 聚类结果解读

自动生成相应聚类算法 Results 子文件夹.

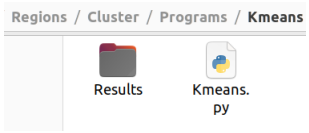

Figure 3.1 ./Cluster/Programs/Kmeans

testData.xlsx 聚类数据放置在 Results 文件夹下.

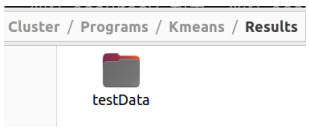

Figure 3.2 ./Cluster/Programs/Kmeans/Results

左边文件夹存放聚类指数 DBI 和 DI 结果，后续可用于 TOPSIS，右边文件夹存放聚类结果，看需要使用.

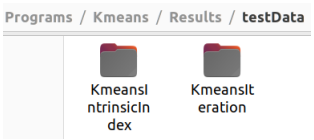

Figure 3.3 Iteration Results

IV. 综合评价

这里提供之前 Matlab 函数 ./**Cluster**/**Programs**/**TOPSIS.m**，可能需要使用者自行修改读取.txt 文件数据，应该包括“路径”和“读取跳跃的步长”.

```
1 M = importdata("./SOM_Intrinsic_Exponential/
      Intrinsic_Exponential.txt").data;
2 N = length(M);
3 GMM_DBI = M(1:2:N-1);
4 GMM_DI = M(2:2:N);
```

最后，待使用者跑完数据，仿照下面形式择优选取较佳的聚类数目.

2023 年 6 月 22 日

依次进行 Kmeans、GMM 和 SOM 三种聚类，其中 Kmeans 进行两次实验，得到内部指标选代结果图像如下.

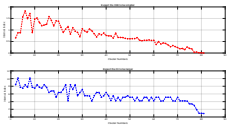

图 1: GMM 内部指标

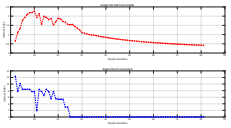

图 2: SOM 内部指标

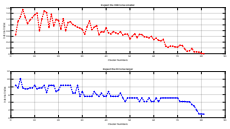

图 3: Kmeans1 内部指标

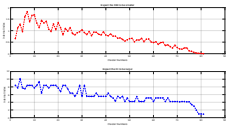

图 4: Kmeans2 内部指标

戴维森堡丁指数（Davies-Bouldin Index）是度量每个簇类最大相似度均值的指标，其核心思想是计算每个簇与其他簇的相似度，再求得所有相似度的平均值衡量整个聚类结果的优劣. 直观理解，如果簇间相似度越高，即 DBI 指数越大，则簇间距离越小，与我们聚类核心思想背驰，结果越差，反之亦然.

邓恩指数（Dunn Index）刻画的是任意两个簇之间（簇间）最短距离的最小值除以任意簇内距离（簇内）最远的两个点的距离最大值，DI 越大越好，如果簇间最近的距离最小值越大，DI 越大，如果任意一个簇内距离最远的两个点的距离的最大值越小，则 DI 越大.

DBI 越小意味着类内距离越小，同时类间距离越大，DI 越大意味着类间距离越大，同时类内距离越小. 期望 DBI 指数越低越好，DI 指数越高越好，因此，如果考虑两个指标的综合度量，需要一种综合评价方法，一般的，我们采用 TOPSIS（Technique for Order Preference by Similarity to Ideal Solution）逼近理想解排序法，简称优劣解距离法.

编写 Matlab 脚本 TOPSIS 对于极小型指标 DBI 正向化处理，采用函数：

$$\Im(\mathrm{DBI}) = \max\{\mathrm{DBI}\} - \mathrm{DBI}$$

优劣解距离法得到图像如下：

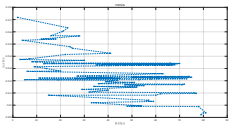

图 5: 第一次试验 Kmeans 的 TOPSIS

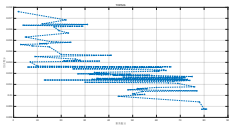

图 6: 第二次试验 Kmeans 的 TOPSIS

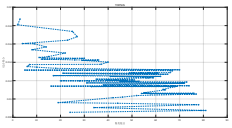

图 7: GMM 的 TOPSIS

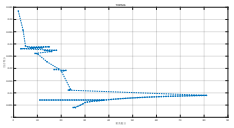

图 8: SOM 的 TOPSIS

依据 TOPSIS 图像得到如下表格结果.

表 1: 择优聚类数（从左向右聚类效果递减，如 DI 准则下 $4 > 10 > 12$）

	DBI	DI	TOPSIS
Kmeans	$81^{\mathrm{double}}(2, 5, 12^{\mathrm{double}})$	吻合性很高 4(10, 12)	2(23, 21, 16, 22, 12, 31, 4)
GMM	2(9, 29, 81)	3(8)	3(2, 25, 27, 23, 4)
SOM	2(13, 18, 81)	2(4, 12, 15)	2(4, 5, 6, 15, 12, 8, 7, 3)

最后整合，保留较优的聚类数目如下：

$$\begin{cases} \text{Kmeans} : 2,\ 4,\ 5,\ 12,\ 16 \\ \text{GMM} : 2,\ 3,\ 4,\ 8,\ 9 \\ \text{SOM} : 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 12,\ 13,\ 15,\ 18 \end{cases}$$

TOPSIS.m

```
clear
clc
%% ---------------------------------------------------------------------
%% Read txt file data.
M = importdata("./Kmeans_Intrinsic_Exponential/Experiment2/Intrinsic_Exponential.txt").data;
N = length(M);
GMM_DBI = M(1:2:N-1);
GMM_DI = M(2:2:N);
%% ---------------------------------------------------------------------
%% Forward processing, max -> x.
DBI = max(GMM_DBI) - GMM_DBI;
DI = GMM_DI;
%% ---------------------------------------------------------------------
%% Standardize.
matrix = [DBI, DI];
[n,m] = size(matrix);
standardMartrix = matrix ./ repmat(sum(matrix .* matrix) .^ 0.5, n, 1);
%% ---------------------------------------------------------------------
%% Weight.
judge = true
if judge == true
    weight = [0.25 0.75];
    if isempty(weight)
        error("Error.")
    else
        disp("Done.")
    end
elseif judge == false
    weight = ones(1,m) ./ m;
end
%% ---------------------------------------------------------------------
%% Compute score.
maxIntercept = sum([(standardMartrix - repmat(max(standardMartrix),n,1)) .^ 2] .* repmat(weight, n, 1) ,2) .^ 0.5;
minIntercept = sum([(standardMartrix - repmat(min(standardMartrix),n,1)) .^ 2] .* repmat(weight, n, 1) ,2) .^ 0.5;
unnormalizedScore = minIntercept ./ (maxIntercept + minIntercept);
disp("Ultimate score:")
standardScore = unnormalizedScore / sum(unnormalizedScore);
[sortScore,index] = sort(standardScore, "descend")
%% ---------------------------------------------------------------------
%% Plot.
plot(index-1,sortScore,"--.","MarkerSize",20)
grid on
xlabel("Cluster Numbers")
ylabel("Synthesis Score")
title("TOPSIS")
%% Save figure.
%% ---------------------------------------------------------------------
set(gcf,"Units","Inches");
pos = get(gcf,"Position");
set(gcf,"PaperPositionMode","Auto","PaperUnits","Inches","PaperSize",[pos(3), pos(4)])
filename = "KMEANS2OPSIS";
print(gcf,filename,"-dpdf","-r0")
```