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

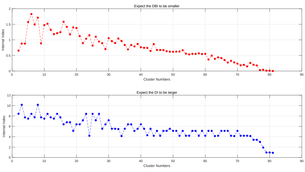

图 1: GMM 内部指标

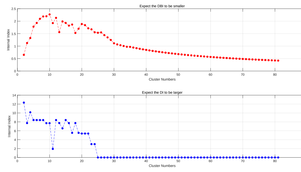

图 2: SOM 内部指标

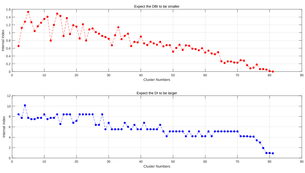

图 3: Kmeans1 内部指标

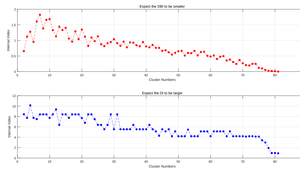

图 4: Kmeans2 内部指标

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：

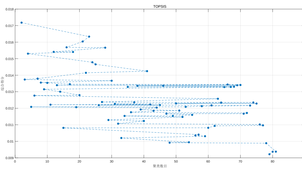

图 5: 第一次试验 Kmeans 的 TOPSIS

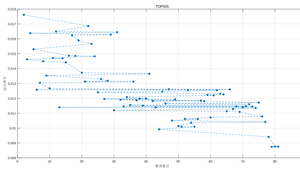

图 6: 第二次试验 Kmeans 的 TOPSIS

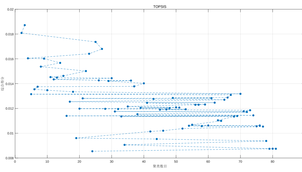

图 7: GMM 的 TOPSIS

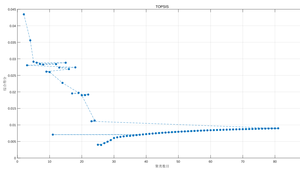

图 8: SOM 的 TOPSIS

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	$81^{double}(2,\ 5,\ 12^{double})$	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);
standardMartix = 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([(standardMartix - repmat(max(standardMartix),n,1)) .^ 2 ] .* repmat(weight, n, 1) ,2) .^ 0.5;
minIntercept = sum([(standardMartix - repmat(min(standardMartix),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")
```