An Efficient Spectral Clustering Algorithm Based on Granular-Ball

汇报人 **xxx**

CONTENTS

研究背景

研究背景

谱聚类算法是图聚类算法的一种代表性算法，已被广泛应用于数据挖掘、模式识别、图像分析等领域。然而，随着数据规模的不断增大，谱聚类算法的缺陷也逐渐凸显：

- 首先，谱聚类算法的时间复杂度高，主要体现在三个方面:相似矩阵的构建(O(n^2*d))、拉普拉斯矩阵的特征值分解(O(n^3))以及最终的K-means聚类(O(nkt))。其中，n表示数据集的规模，d表示数据维度，t表示K-means的迭代次数。随着数据规模n的增大，谱聚类算法的计算复杂度急剧上升，使得其难以应用于大规模数据集。
- 其次，谱聚类算法的空间复杂度也存在问题，主要体现在需要存储O(n^2)的相似矩阵和拉普拉斯矩阵。对于大规模数据集，这种存储需求会导致内存消耗过大，从而限制了谱聚类算法的应用。

为了解决上述问题，现有的研究主要集中在两个方向:基于特征映射的方法和基于样本减少的方法。前者通过构建特征映射来降低相似矩阵的计算复杂度，后者通过选择代表性样本来减小数据规模。但这些方法往往会影响聚类效果，且难以兼顾时间效率和空间效率。

因此，如何在保持良好聚类效果的同时，大幅提高谱聚类算法的时间和空间效率，成为了亟待解决的问题。本文提出了一种基于颗粒球的改进谱聚类算法(GBSC)，旨在解决上述问题，为大规模数据的聚类分析提供一种高效可行的解决方案。

GBSC

The Process of Generating Granular-balls

- 颗粒球GBj的中心c和半径r

  中心c是颗粒球内所有数据点的平均值。
  半径r是颗粒球内数据点到中心的最大距离。

$$c_j = \frac{1}{n_j}\sum_{i=1}^{n_j} p_i.$$

$$r_j = max(||p_i - c_j||),$$

- 分布度量DM

  通过计算颗粒球内数据点数n和半径之和s的比值来度量的。
  DM越小，表示颗粒球内部分布越密集。

$$s_j = \sum_{i=1}^{n_j} ||p_i - c_j||$$

$$DM_j = \frac{s_j}{n_j}.$$

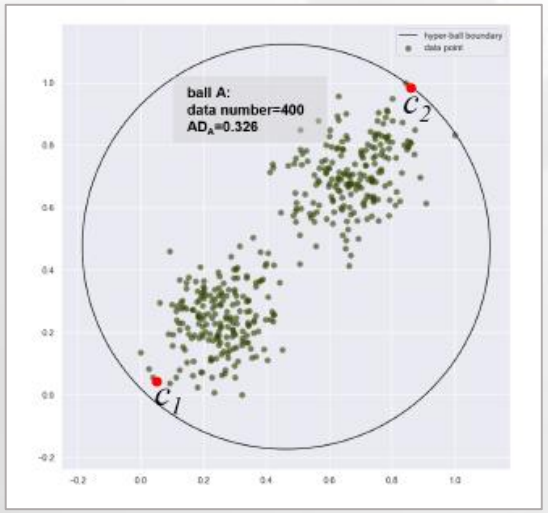
- 加权DM值

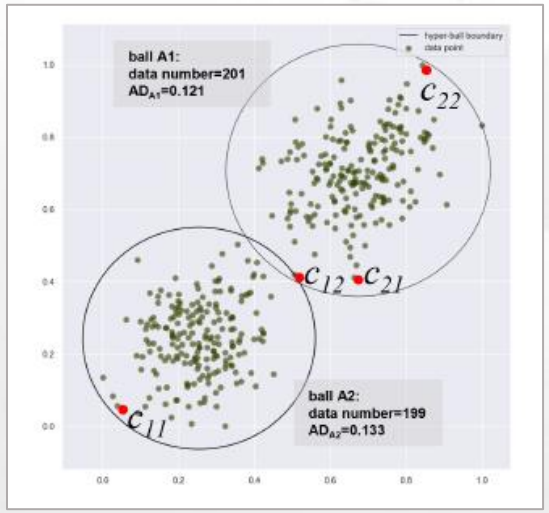
  在颗粒球的分裂过程中作为分裂标准。
  综合考虑了子颗粒球的数据点数和DM值，可以更好地适应噪声数据。

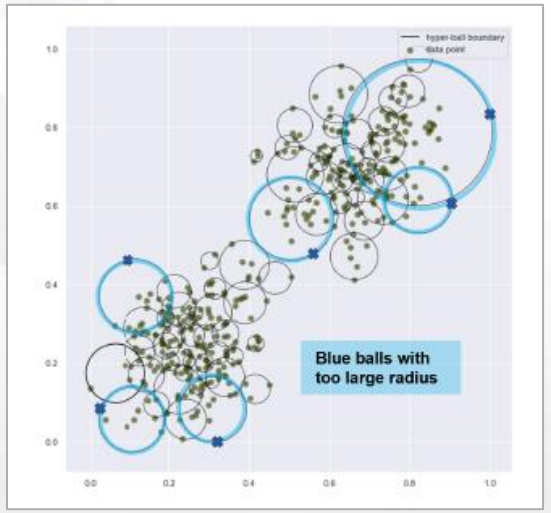
$$DM_{weight} = \frac{n_{A1}}{n_A} DM_{A1} + \frac{n_{A2}}{n_A} DM_{A2}.$$

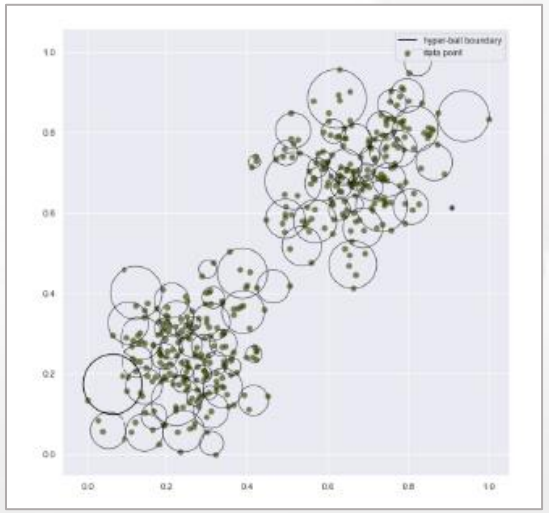
The Process of Generating Granular-balls

自适应的颗粒球生成方法：









Construction of Similarity Matrix Based on Granular Ball

在传统谱聚类算法中，给定n个样本点pi(i=1，2，...，n)，计算每对样本点之间的高斯核函数距离，形成n×n的相似性矩阵W，但是这种全连接的相似性矩阵构建方式在处理大规模数据时效率较低。

基于颗粒球的相似性矩阵构建方法：

- 首先定义颗粒球之间的距离：两颗粒球中心距离减去两个半径之和，如果有重叠则距离设为0。

$$dis(GB_{j1}, GB_{j2}) = dis(c_{j1}, c_{j2}) - (r_{j1} + r_{j2}).$$

- 不包括属于离群点集的颗粒球参与相似性矩阵的构建。

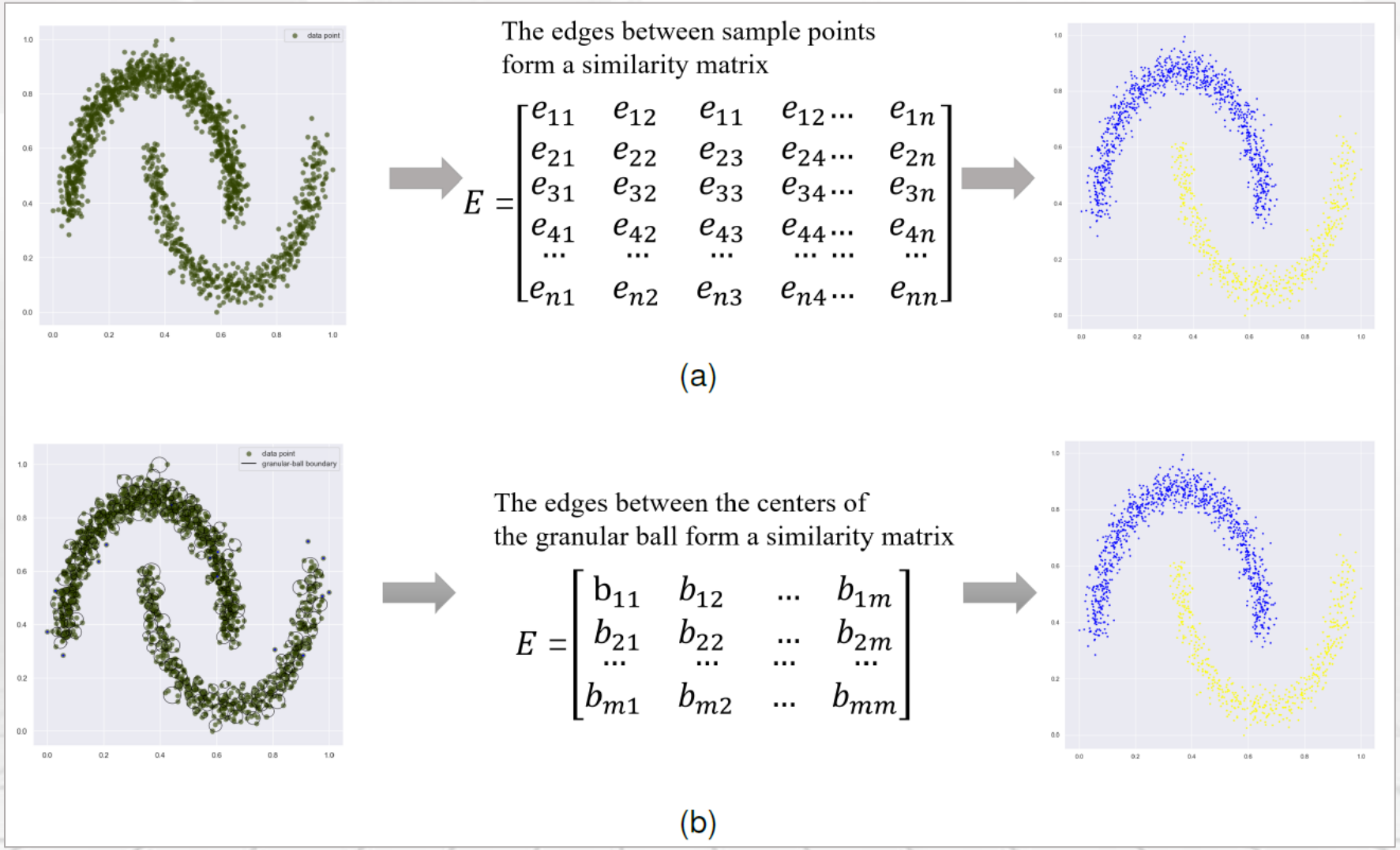
$$outlier = \{GB_j | n_j \leq 2\}.$$

- 最后计算颗粒球之间的高斯核函数相似度，形成m×m的相似性矩阵W，其中m<<n。

$$b_{ij} = e(b_i, b_j) = \sum_{i=1,j=1}^{m} \exp \frac{-\|b_i - b_j\|^2}{2\sigma^2}$$

$$W = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ b_{k1} & b_{k2} & b_{k3} & b_{mm} \end{bmatrix}$$

Construction of Similarity Matrix Based on Granular Ball



Improved Spectral Clustering Algorithm Based on Granular Ball

基于颗粒球的改进谱聚类算法GBSC：

- 生成颗粒球

Algorithm 1 Generation of granular-balls.

Input Dataset D, number of clusters k
Output GB sets
For each granular-ball GB_j in D **do**
  calculate DM_A, DM_{weight},
    according to Eq.2, Eq.3, Eq.4,Eq.5;
  If $DM_{weight} \geq DM_A$ **Then**
    Split GB_j;
  End If
 If the number of GBs is not changing **Then**
    break;
 End For
 For each granular-ball GB_j in D **do**
  calculate $mean(r)$, $median(r)$,
  If $r_j \geq 2 \times \max(mean(r), median(r))$ **Then**
    Split GB_j;
  End If
 If the number of GBs is not changing **Then**
  break;
 End For
return GB sets;

- 基于颗粒球的谱聚类

Algorithm 2 GBSC Algorithm.

Input GB sets
Output Clustering result Y
Construct the granular-ball similarity matrix W.
Constructing non normalized Laplacian matrix L_w.
Constructing normalized Laplace matrix L_n.
Will L_n feature decomposition to calculate k minimum eigenvalues and corresponding eigenvectors.
Mapping the corresponding nodes to the k dimensional space for $k-means$ clustering to obtain the cluster partition $C_1, C_2, \ldots, C_k$.

模拟实验

模拟实验

- 实验环境

实验在Python环境下进行，硬件环境为16GB内存、3.0GHz i5-9500 CPU的Windows 10电脑。
为了防止读写速度影响实验结果，实验数据集已经提前读取完毕。

- 实验数据

实验使用了12个合成数据集和4个UCI真实数据集进行测试。
合成数据集的详细信息如表I所示，包括样本数、维度和类别数。
真实数据集的信息如表II所示，包括Abalone、Drybean、Waveform和Pendigits。

TABLE I: Information of the Synthetic Datasets

dataset	number of the sample	dimensionality	category
D1	944	2	2
D2	1043	2	2
D3	1039	2	4
D4	567	2	2
D5	1641	2	3
D6	876	2	2
D7	1741	2	6
D8	1016	2	4
D9	6699	2	5
D10	1427	2	4
D11	3603	2	3
D12	1020	2	3

TABLE II: Information of the Real Datasets

dataset	number of the sample	dimensionality	category
Abalone	4177	8	3
Drybean	13611	16	7
Waveform	5000	21	3
Pendigits	10992	16	10

模拟实验

- 评估指标

使用NMI(标准互信息)和ARI(调整兰德指数)两个指标来评估聚类效果。
每个算法在每个数据集上运行10次，取平均值作为最终结果。

NMI (Normalized Mutual Information)标准化互信息:
NMI是用来衡量两个聚类结果之间的相似度的指标。
NMI将互信息归一化到[0，1]区间，0表示两个聚类结果完全不同，1表示两个聚类结果完全相同。

$$NMI = \frac{2I(Y;C)}{H(Y)+H(C)}.$$

$$H(X) = -\sum_{i=0}^{|X|} P(i) log_2 P(i).$$

$$I(Y;C) = H(Y) - H(Y|C),$$

ARI (Adjusted Rand Index)调整兰德指数:
ARI度量了聚类结果与真实类别之间的相似度。
ARI的取值范围为[-1，1]，1表示两个聚类结果完全一致，0表示两个聚类结果完全随机，负值表示两个聚类结果负相关。

$$ARI = \frac{2(ad-bc)}{(a+b)(b+d)(a+c)(c+d)},$$

模拟实验

- 不同算法的NMI值以及ARI值比较

TABLE V: Comparison of NMI Values of Different Algorithms

dataset	USENC	USPEC	LSCH	DNCSC	FRWL	KMEANS	CSSP	KASP	SC	GBSC
Abalone	0.042	0.010	0.069	0.012	0.136	0.102	0.132	0.121	0.165	**0.166**
Drybean	0.508	0.161	0.467	0.160	0.424	0.387	0.508	0.496	0.711	**0.730**
Waveform	0.367	**0.371**	0.370	0.368	0.369	0.368	0.368	0.365	0.368	0.368
Pendigits	**0.858**	0.802	0.822	0.792	0.687	0.587	0.478	0.558	0.597	0.597

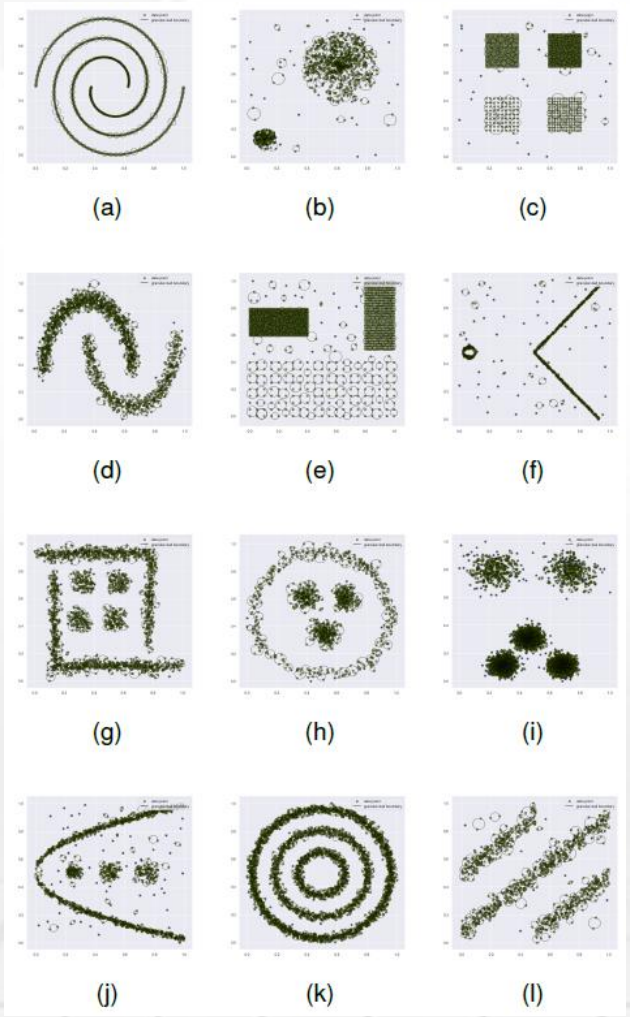
TABLE VI: Comparison of ARI Values of Different Algorithms

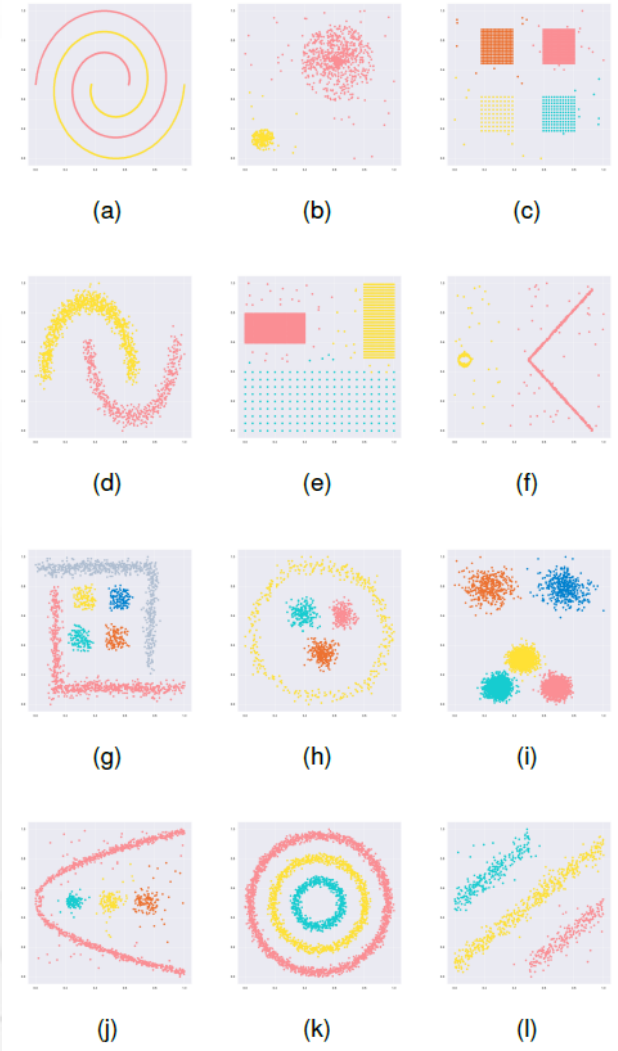
dataset	USENC	USPEC	LSCH	DNCSC	FRWL	KMEANS	CSSP	KASP	SC	GBSC
Abalone	0.011	0.012	0.074	0.005	0.125	0.073	0.136	0.080	0.124	**0.164**
Drybean	0.338	0.032	0.302	0.032	0.282	0.238	0.367	0.282	0.643	**0.669**
Waveform	0.250	0.256	**0.259**	0.254	0.253	0.250	0.250	0.257	0.250	0.251
Pendigits	**0.790**	0.717	0.753	0.650	0.602	0.436	0.366	0.402	0.424	0.426

模拟实验

GBSC在12个合成数据集上的聚类结果:

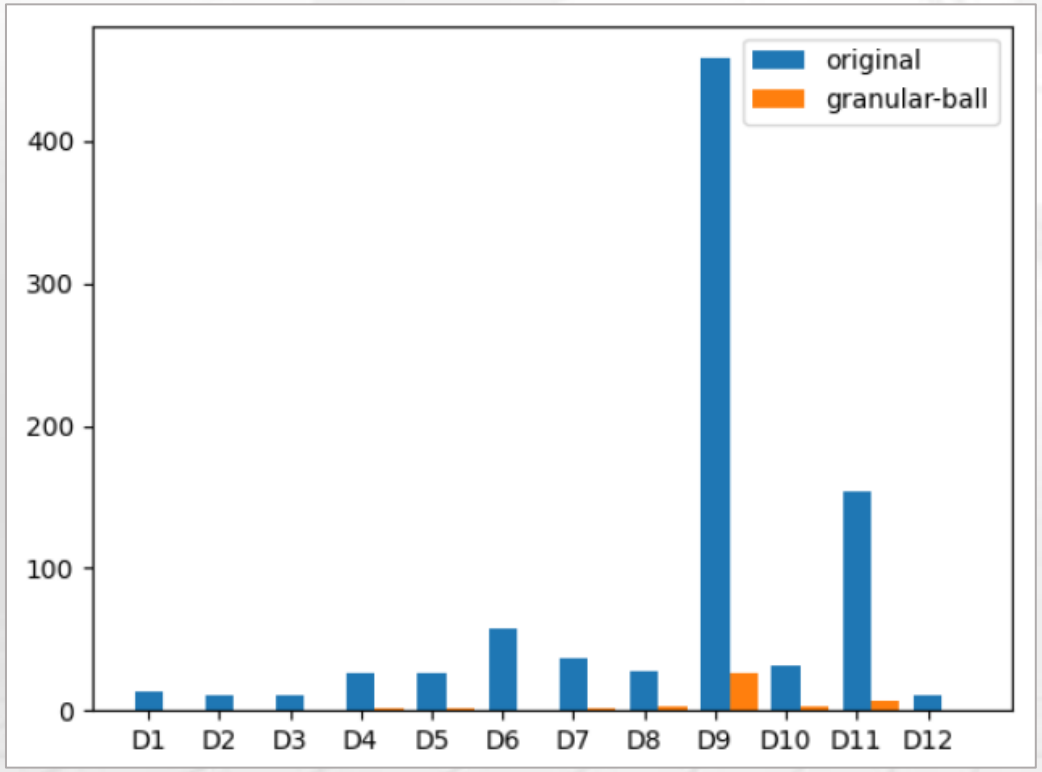
- GBSC算法能够很好地识别出各个数据集中的聚类结构，聚类效果较为理想。
- 对于不同的数据集，GBSC算法能够自适应地调整聚类结果，体现了其良好的鲁棒性。
- 这些结果表明GBSC算法在处理合成数据集时具有较强的聚类性能。



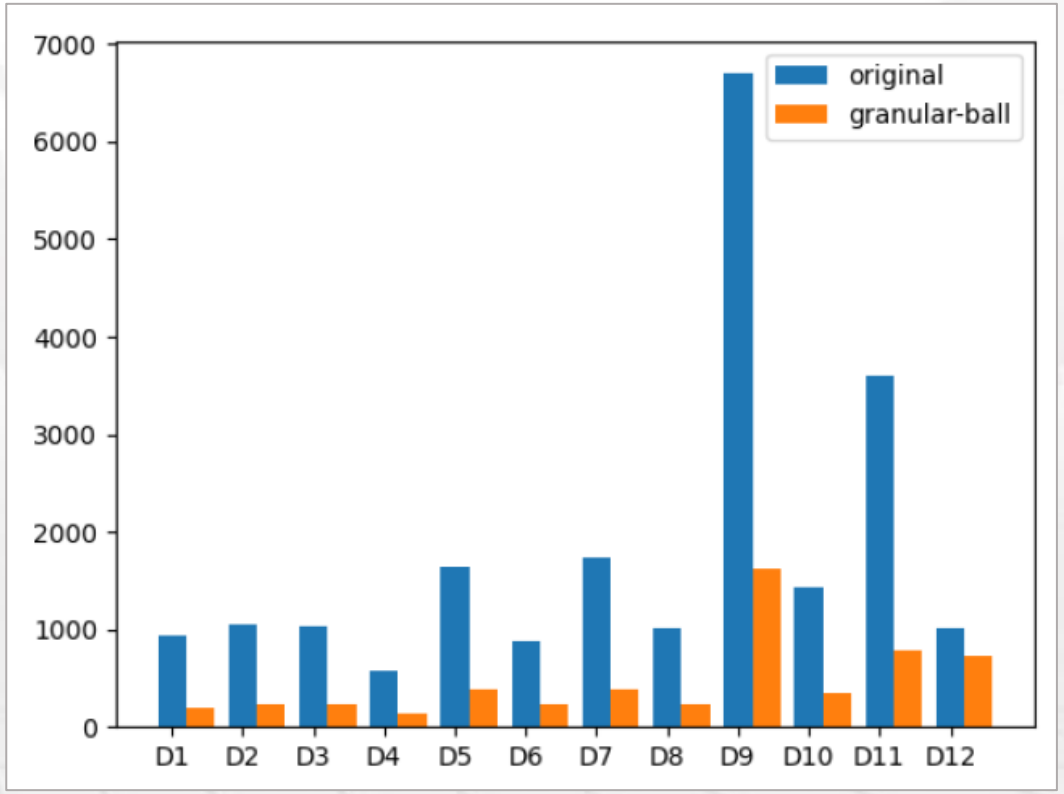


模拟实验

- 不同算法在时间消耗上的比较



- 不同算法在内存消耗上的比较



总结与讨论

总结与讨论

本文提出了一种基于granular-ball的改进型谱聚类算法(GBSC)。该算法通过使用granular-ball来表示数据，大大减小了相似性矩阵的规模，从而提高了谱聚类算法在大规模数据集上的时间和空间效率：

具体来说，文章首先介绍了granular-ball的生成过程，通过自适应的方式将数据划分为多个granular-ball，并利用加权的分布密度来判断是否需要进一步细分。这种方式能够有效抑制噪声点的影响，提高了算法的鲁棒性。
然后，文章提出了基于granular-ball的相似性矩阵构建方法。相比于传统谱聚类直接计算所有数据点之间的相似度，GBSC仅需计算granular-ball之间的相似度，大大减少了计算量。理论分析表明，GBSC的时间复杂度为O(m^3)，其中m为granular-ball的数量，远小于原始数据量n。

实验结果验证了GBSC在时间效率、空间效率和聚类性能方面的优势。在12个合成数据集和4个UCI数据集上的实验中，GBSC均取得了较好的聚类结果，且在大规模数据集上的性能尤为突出，最高可达104倍的加速比。
总的来说，GBSC算法通过引入granular-ball的概念，有效地解决了传统谱聚类在大规模数据集上的效率问题，在保持良好聚类性能的同时大幅提升了计算效率，为大规模数据聚类提供了一种有效的解决方案。未来可进一步探索如何将granular-ball思想与其他改进谱聚类算法相结合，进一步提升算法性能。

Thanks for your listening!