

# *k*-means 中聚类簇数 *k* 的选择

李溢江

2021 年 6 月 30 日

## 摘要

*k*-means 是一个被广泛应用的聚类算法，它需要人为指定聚类簇数 *k* 作为算法输入，然而当我们对数据进行聚类时，聚类簇数 *k* 通常是未知的。前人提出一种可以自动寻找 *k* 的聚类算法：G-means [1]，该算法基于一个统计假设检验，即同一簇内的样本服从高斯分布，G-means 在样本子集上运行 *k*-means，缓慢增长 *k*，直到所有样本子集都服从高斯分布。G-means 只需要一个参数，即假设检验中的显著性水平  $\alpha$ 。本文的主要工作是对 G-means 进行了细微的改进，即在统计假设检验中改用主成分分析对数据进行降维，经过实验验证，在高维多簇的数据集上，对聚类簇数 *k* 的预测性能较原来有明显提升。

关键词：聚类，G-means 算法，*k*-means 算法，高斯分布，主成分分析

## 1 引言

聚类将数据集中的样本划分为若干个不相交的子集，每个子集称为一个簇，目标是使得同一簇的样本相似，而不同簇的样本相异。聚类算法在数据挖掘、数据压缩等工程应用领域发挥了重要的作用 [2]，然而，许多聚类算法都需要人为指定聚类簇数 *k*，有时我们并不知道 *k* 的确切取值，所以只能凭借先验知识来做出一个粗糙的决定。当面对高维数据时，即使数据有很好的可分性，我们也很难确定 *k* 的取值 [3]。

*k*-means 是一种基于划分的聚类算法 [4]，其通常假设划分得到的每个簇中样本服从单峰分布，如高斯分布。使用这些方法，应该为每个服从单峰分布的样本子集只分配一个簇中心，如果为一个服从高斯分布的样本子集分配了多个簇中心，那么这将会使得本该聚类为一簇的样本被划分为多簇，同样地，若为服从不同分布的样本子集只分配了一个簇中心，那么这将会导致本属于不同簇的样本被合并到了同一簇中，如图 1 所示。

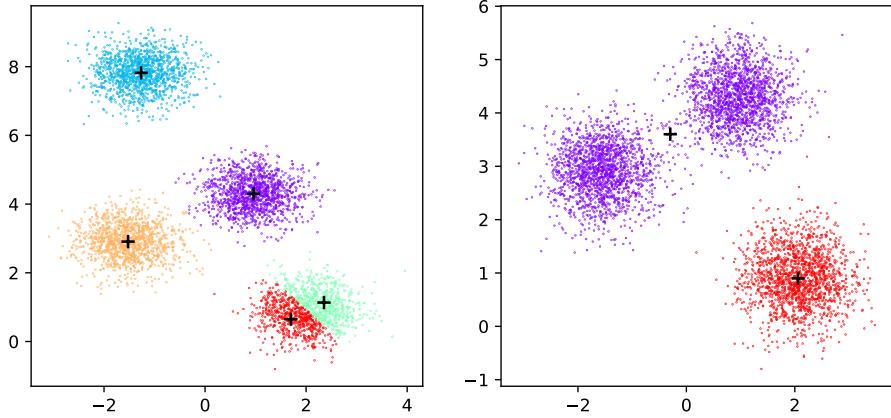


图 1: 当指定不恰当的聚类簇数  $k$  时  $k$ -means 的聚类结果

## 2 预备知识

### 2.1 $k$ -means

对于给定的  $n$  维样本集合  $X = \{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ ,  $k$ -means 算法的任务是将其划分为  $k$  簇  $C = \{C_1, C_2, \dots, C_k\}$ , 使得簇划分的平方误差最小化:

$$J(\lambda, \mu) = \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - \mu_{\lambda^{(i)}}\|^2$$

$$\min J(\lambda, \mu)$$

其中  $\lambda^{(i)}$  表示样本  $x^{(i)}$  所属簇的索引  $(1, 2, \dots, k)$ ,  $\mu_{\lambda^{(i)}}$  表示簇  $C_{\lambda^{(i)}}$  的中心,  $J(\lambda, \mu)$  刻画了簇内样本围绕簇中心的紧密程度, 最小化  $J$  使得簇内样本的相似度增加。

$k$ -means 算法是一个迭代的过程, 在每次迭代中主要进行两步, 而在这两步中采用贪心策略来最小化平方误差  $J$ :

- 簇分配: 将样本划分到距离最近的中心所属的簇, 通过更新  $\lambda$  以优化  $J$
- 移动簇中心: 重新计算每个簇的中心, 通过更新  $\mu$  以优化  $J$

可以看到,  $k$ -means 算法需要人为指定聚类簇数  $k$  作为算法输入, 并且是固定的, 而在真实的应用场景中, 对于未知领域的数据集, 特别是面对高维数据时, 由人们来确定聚类簇数  $k$  是十分困难的, 即使所面对的数据集是十分理想的, 即同类样本相似度高, 异类样本相似度低。当我们指定的聚类簇数偏小时, 会将相似度低的样本划分到同一簇; 聚类簇数偏大时, 会使相似度高的样本划分到不同簇中。

## 2.2 主成分分析

主成分分析 (Principal Component Analysis, 简称 PCA) 是一种常用的降维方法，寻找一个超平面，使得样本点在这个超平面的投影尽可能地分开，这些投影点即为样本点降维后的结果。降维后低维空间中的维数人为指定，当然保留的维数越多，对于原样本的信息保留程度越好。降维必然导致的部分信息被丢失，但是这也是有必要的：因为当数据中存在噪声时，通过降维舍弃部分信息在一定程度上会起到去噪声的效果。

使用主成分分析降维后，原始空间中样本之间的距离在低维空间中得以保持，即样本所服从的分布不会改变，同时会在一定程度上对数据进行去噪处理。这刚好符合 G-means 中统计假设检验的需求：将高维数据映射到一维，然后通过检验映射到一维的样本是否服从高斯分布来反映样本在原始空间中是否服从（多元）高斯分布。这也是本文提出对 G-means 改进的重要支撑理由。

## 3 G-means

G-means (Gaussian-means) 算法是由 Hamerly 与 Elkan 在 2004 提出的 [1]，它赋予  $k$ -means 自动寻找聚类簇数  $k$  的能力。在  $k$ -means 的簇分配的过程中， $k$ -means 算法隐式地假设每个簇中的样本围绕中心呈球形分布，同时通常假设划分得到的每个簇中的样本服从高斯分布，G-means 正是基于此假设前提，也是其名称的来源。

G-means 开始时选择一个很小的聚类簇数  $k$ （例如  $k = 1$ ，或者基于我们的先验知识来选择一个稍大的  $k$ ，这样会有助于算法更快收敛），并以此作为聚类簇数运行一次  $k$ -means，聚类得到  $k$  个样本子集，在这里没有将其称为簇，是因为算法远远还未结束，这并不是最终的聚类结果。然后 G-means 对这每个样本子集进行假设检验，若一个样本子集中的样本服从高斯分布，那么 G-means 就不会对该样本子集进行划分，只会为其分配一个簇中心；若不服从，那么在该样本子集上运行聚类簇数为 2 的  $k$ -means，使之划分为两个子集，聚类簇数  $k + 1$ ，重复上述过程，直到所有样本子集都服从高斯分布。

若 G-means 算法最终得到的簇数为  $k$ ，那么共运行了  $O(k)$  次  $k$ -means。其中参数  $\alpha$  是假设检验中的显著性水平，需要预先给定。

### 3.1 检验簇中数据点是否服从高斯分布

这一节将详细介绍在 G-means 算法中，如何检验一个样本子集是否服从高斯分布，同时也会给出使用主成分分析的改进方案。对样本子集的检验结果给出下面两种假设：

- $H_0$ : 样本子集服从高斯分布
- $H_1$ : 样本子集不服从高斯分布

当检验结果接受假设  $H_0$  时，那么该样本子集不会被划分，因为一个簇中心足以代表这个样本子集建模。若检验结果接受假设  $H_1$ ，那么该样本子集将被划分为两个，即生成两个簇中心。

基于安德森-达令检验（Anderson-Darling Test，简称 AD 检验）[5] 来验证簇中样本是否服从高斯分布。AD 统计量用于度量数据服从特定分布的程度，数据与分布越拟合，那么该统计量就越小。

对于一个  $d$  维样本子集  $X$ ，其当前的簇中心  $c$ ，那么对该样本子集的检验过程如下：

1. 设置显著性水平  $\alpha$
2. 在样本子集  $X$  上运行聚类簇数为 2 的 k-means，得到两个簇中心  $c_1, c_2$
3. 令  $v = c_1 - c_2$ ，表示连接簇中心  $c_1, c_2$  之间的一个向量，然后将  $X$  中所有的样本投影到向量  $v$  上： $x'_i = x_i \cdot v / |v|$ ，得到 1 维样本子集  $X'$ ，然后对  $X'$  使用归一化，使其均值为 0，方差为 1。
4. 对  $X'$  使用 AD 检验，若在显著性水平  $\alpha$  下，接受假设  $H_0$ ，那么样本子集  $X$  将不会被划分，保留原始簇中心  $c$ ，丢弃  $c_1, c_2$ ；若接受假设  $H_1$ ，那么  $X$  将被划分为分别以  $c_1, c_2$  为新中心的两个样本子集。

在第 3 步中，将样本子集中的样本投影到连接簇中心  $c_1, c_2$  之间的向量  $v$  上，然后使用 AD 检验，G-means 的作者解释这样做的理由： $v$  是由 k-means 得到的对分离样本最重要的一个方向。

上述的处理过程可简单描述为对数据降维，然后检验其是否服从特定的分布，所以要求在对数据进行降维后，尽可能地仍然服从原分布。本文中提出的改进方法就是使用主成分分析对  $X$  降维，这使得原始空间中样本之间的距离在低维空间中继续保持，投影到低维空间后丢失的信息最少，样本在最大程度上保留原来的分布特征，同时还达到了一定的去噪效果。将改进后的 G-means 记为 G-means(PCA)，经过实验验证，这个微小的改进使得 G-means(PCA) 在面对高维多簇数据集时，聚类簇数  $k$  的预测性能有了明显的提升。

### 3.2 G-means 的一个运行实例

在图 2 中展示了 G-means 算法在生成数据集上的运行过程，该数据集包含 5000 个样本，特征维数为 2，且真实的簇数为 4，每个簇都服从方差为 0.5 高斯分布，我们选取显著性水平  $\alpha = 1\%$ ，其对应的临界值为 1.029，也就是当假设检验得到的 AD 统计量大于 1.029，假设  $H_1$  将被接受，否则假设  $H_0$  将被接受。设置 G-means 的初始聚类簇数为 1，AD 统计量为 155.2，假设  $H_1$  将被接受，运行聚类簇数为 2 的 k-means，样本被划分到两个子集，如图 2(2)；再对这两个样本子集分别进行 AD 检验，上方的样本子集的 AD 统计量为 0.5，下方的样本子集的 AD 统计量为 246.23，所以只对下方的样本子集进行划分，在下方样本

子集中运行聚类簇数为 2 的  $k$ -means，得到聚类结果如图 2(3)；重复上述过程，直到所有的数据子集上的 AD 检验都接受假设  $H_0$ ，如图 2(4)，最后 G-means 输出聚类簇数  $k = 4$ 。

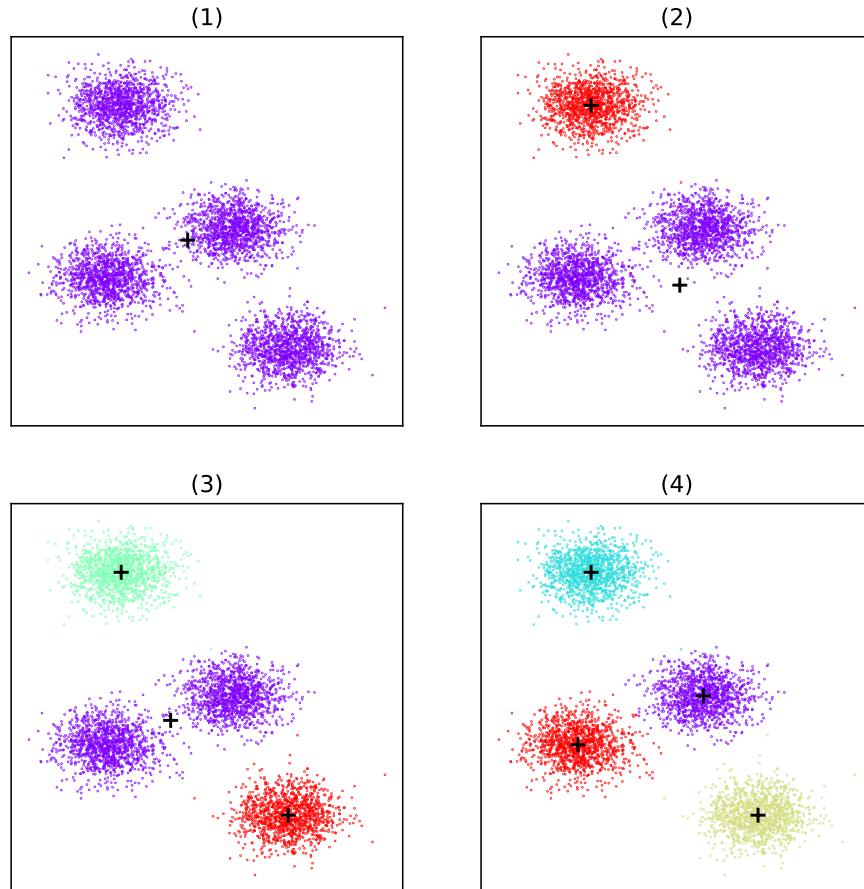


图 2: G-means 在真实簇数为 4 的数据集上的运行过程

## 4 实验

在这一个节中，将 G-means 与改进后的 G-means(PCA) 进行对比，通过实验证它们对聚类簇数的预测性能。实验所用到的数据集是生成的，数据集中样本数为 5000，且真实簇中的样本是服从方差为 0.5 高斯分布，样本特征的维数  $d = \{8, 16, 32, 64, 128\}$ ，真实

簇的数量  $k = \{4, 16, 32, 64\}$ , 同时考虑到在实际场景中, 数据的每一维特征的取值范围是不尽相同的, 所以对生成数据的每一维特征进行范围为  $[0.5, 2]$  的随机放缩。G-means 与 G-means(PCA) 的输入参数均设置为: 初始簇数  $k = 1$ , 显著性水平  $\alpha = 1\%$ 。对于每一种数据集类型, 都会随机生成 50 个数据集进行实验, 并记录聚类簇数预测结果的平均值、标准差、最小值、最大值, 实验结果记录在表 1 中。

表 1: G-means 与 G-means(PCA) 的对比实验结果

$d$	$k$	G-means				G-means(PCA)			
		avg	std	max	min	avg	std	max	min
8	4	4.0	0.0	4	4	4.0	0.0	4	4
	16	16.0	0.0	16	16	16.0	0.0	16	16
	64	75.0	0.1	75	76	73.6	1.3	71	75
	128	163.0	0.3	161	163	155.0	0.0	155	155
16	4	4.0	0.0	4	4	4.0	0.0	4	4
	16	16.0	0.0	16	16	16.0	0.0	16	16
	64	71.1	1.0	71	78	69.7	1.7	68	72
	128	165.1	0.7	165	170	143.9	0.7	140	144
32	4	6.9	0.4	4	7	4.0	0.0	4	4
	16	19.3	2.1	19	34	16.0	0.0	16	16
	64	113.3	1.8	113	126	65.0	1.0	64	66
	128	202.6	3.9	202	230	134.5	0.5	134	135
64	4	219.9	0.4	217	220	4.0	0.0	4	4
	16	237.6	2.7	219	238	16.0	0.0	16	16
	64	376.5	3.8	376	403	65.5	1.5	64	67
	128	386.9	6.3	386	431	132.4	0.7	129	133
128	4	-				4.0	0.0	4	4
	16	-				16.0	0.0	16	16
	64	-				64.0	0.1	64	65
	128	-				129.0	0.2	128	130

可以发现, 当面对低维低簇的数据集时, 即  $d = 8, 16; k = 4, 16$  时, G-means 与 G-means(PCA) 都表现的很好, 对于聚类簇数的预测值与真实簇数完全一致。当面对低维高簇的数据集时, 即  $d = 8, 16; k = 64, 128$  时, G-means 与 G-means(PCA) 的预测结果相对真实簇数略有偏差, 但是 G-means(PCA) 的预测偏差要比 G-means 的小。当  $d = 32$  时, G-means 的预测簇数与真实簇数相差很大, 最高误差达 57%, 但是 G-means(PCA) 仍然表现出色。当面对高维数据集时, G-means 完全失去预测能力, 甚至表现为算法不收敛, 但是

G-means(PCA) 对聚类簇数的预测仍然准确，且预测结果稳定。

当预测的聚类簇数  $k$  相近时，G-mean(PCA) 的运行速度比 G-means 要慢，这是因为 PCA 中进行矩阵分解所花费的时间代价较高，而通过直接映射来降维的方法计算代价小；当面对高维多簇数据时，G-means 的对聚类簇数偏大，这导致 G-means 要更多次的运行  $k$ -means 算法，导致运行时间变长。

## 5 结论与未来工作

本文主要讨论了如何在聚类任务中确定聚类簇数  $k$ ，首先对前人提出的 G-means 算法进行了介绍，它基于簇中的样本服从高斯分布的假设，通过降维处理、假设检验来做出是否划分样本子集的决定。同时提供给算法的唯一参数是显著性水平  $\alpha$ ，它只是决定了假设检验结果的可信度，当  $\alpha$  越小，原假设就越难被推翻，假设检验越保守。G-means 在运行过程中会多次调用  $k$ -means 算法，时间复杂度为线性 [6]。G-means 在低维数据集上表现的很好，但是在面对高维数据时，使用直接投影的降维方法带来的信息损失很大，从而导致假设检验结果不准确，影响了数据子集的划分决策，算法性能大打折扣，甚至出现不收敛的情况。

基于 G-means，本文提出了使用主成分分析进行降维处理，使样本在降维后最大程度地保留原分布，以便在 AD 检验中可以真实反映样本在原始空间中的分布。实验结果证明，G-means(PCA) 在面对高维数据集时，仍然保持高预测性能。文中只是将其运用到了  $k$ -means，其实将这种方法运用到其他聚类算法上也具有很好的效果，因为它的主要贡献是确定  $k$ 。

由于 G-means(PCA) 在降维处理中引入了主成分分析，其中矩阵分解的时间代价高，在面对低维数据时，运行速度不如 G-means，所以未来工作的重点是如何缩短 G-means(PCA) 的运行时间。

## 6 参考文献

### 参考文献

- [1] Greg Hamerly and Charles Elkan. Learning the  $k$  in  $k$ -means. *Advances in neural information processing systems*, 16:281–288, 2004.
- [2] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.
- [3] Duc Truong Pham, Stefan S Dimov, and Chi D Nguyen. Selection of  $k$  in  $k$ -means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(1):103–119, 2005.

- [4] Jyoti Yadav and Monika Sharma. A review of k-mean algorithm. *Int. J. Eng. Trends Technol.*, 4(7):2972–2976, 2013.
- [5] Lloyd S Nelson. The anderson-darling test for normality. *Journal of Quality Technology*, 30(3):298, 1998.
- [6] Sariel Har-Peled and Bardia Sadri. How fast is the k-means method? *Algorithmica*, 41(3):185–202, 2005.
- [7] David Arthur and Sergei Vassilvitskii. How slow is the k-means method? In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 144–153, 2006.
- [8] Ahamed Shafeeq and KS Hareesha. Dynamic clustering of data with modified k-means algorithm. In *Proceedings of the 2012 conference on information and computer networks*, pages 221–225, 2012.
- [9] Horst Bischof, Aleš Leonardis, and Alexander Selb. Mdl principle for robust vector quantisation. *Pattern Analysis & Applications*, 2(1):59–72, 1999.
- [10] Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734, 2000.