THU-70250043-0, Pattern Recognition (Spring 2021) **Homework: 13**

Clustering Methods

Lecturer: Changshui Zhang zcs@mail.tsinghua.edu.cn

Hong Zhao vzhao@tsinghua.edu.cn

Student: Jingxuan Yang yangjx20@mails.tsinghua.edu.cn

K-means

1. Consider a set D of $n = 2k+1$ samples, k of which coincide at $x = -2$, k at $x = 0$ and one at $x = a > 0$. As you have learned in class, we are always trying to minimize the distortion measure in K-means, given by

$$J_e = \sum_{D_i \neq \varnothing} \sum_{x \in D_i} \|x - m_i\|^2 \tag{1}$$

where m_i is the mean value of the samples of the nonempty subset D_i.

1.1 Show that the two-cluster partitioning that minimizes J_e groups the k samples at $x = 0$ with the one at $x = a$ if $a^2 < 2(k+1)$.

解: 首先证明 k 个位于 $x = -2$ 的样本属于同一类.

假设 $x_i = -2 \in D_1$, $x_j = -2 \in D_2$, 我们可以断言将 x_i 从 D_1 取出放入 D_2 会使得 J_e 下降, 因为如若不然则将 x_j 从 D_2 取出放入 D_1 会使得 J_e 下降. 因此, 所有 k 个位于 $x = -2$ 的样本属于同一类. 同理可知 k 个位于 $x = 0$ 的样本也属于同一类. 所以, 在进行两类聚类时, 全部可能的结果有三种情况:

(a) $x = -2, x = 0 \in D_1$, $x = a \in D_2$

(b) $x = -2, x = a \in D_1$, $x = 0 \in D_2$

(c) $x = -2 \in D_1$, $x = 0, x = a \in D_2$

对三种情况分别计算 J_e 可得

$$\begin{aligned} J_e(a) &= k\|-2-(-1)\|^2 + k\|0-(-1)\|^2 = 2k \\ J_e(b) &= k\left\|-2-\frac{-2k+a}{k+1}\right\|^2 + \left\|a-\frac{-2k+a}{k+1}\right\|^2 = \frac{k}{k+1}(a+2)^2 \\ J_e(c) &= k\left\|0-\frac{a}{k+1}\right\|^2 + \left\|a-\frac{a}{k+1}\right\|^2 = \frac{k}{k+1}a^2 \end{aligned} \tag{2}$$

由 $a > 0$ 可知 $J_e(b) > J_e(c)$, 若 $a^2 < 2(k+1)$, 则

$$J_e(c) = \frac{k}{k+1}a^2 < 2k = J_e(a) \tag{3}$$

因此, 当 $a^2 < 2(k+1)$ 时, 两类聚类的情况为 (c), 即 k 个 $x = -2$ 的点属于一个类, k 个 $x = 0$ 的点以及 $x = a$ 属于另一个类.

1.2 What is the optimal grouping if $a^2 > 2(k+1)$?

解: 由 1.1 可知当 $a^2 > 2(k+1)$ 时, 有

$$J_e(b) > J_e(c) > J_e(a) \tag{4}$$

因此两类聚类的情况为 (a), 即 k 个 $x = -2$ 的点与 k 个 $x = 0$ 的点属于一个类, $x = a$ 属于另一个类.

Hierarchical Clustering

2. Consider a hierarchical clustering procedure in which clusters are merged to produce the smallest increase in the sum-of-squared error at each step. If the i-th cluster contains n_i samples with the sample mean $\boldsymbol{m}_i$, show that the smallest increase results from merging the pair of clusters for which

$$\frac{n_i n_j}{n_i + n_j} \left\| \boldsymbol{m}_i - \boldsymbol{m}_j \right\|^2 \tag{5}$$

is minimum.

解: 合并第 i 类与第 j 类的均方误差增量为

$$\begin{aligned}
\Delta E &= \sum_{\boldsymbol{x}\in D_i, D_j} \left\| \boldsymbol{x} - \frac{n_i\boldsymbol{m}_i + n_j\boldsymbol{m}_j}{n_i+n_j} \right\|^2 - \sum_{\boldsymbol{x}\in D_i} \|\boldsymbol{x} - \boldsymbol{m}_i\|^2 - \sum_{\boldsymbol{x}\in D_j} \|\boldsymbol{x} - \boldsymbol{m}_j\|^2 \\
&= \sum_{\boldsymbol{x}\in D_i, D_j} \left[\boldsymbol{x}^\top\boldsymbol{x} - 2\boldsymbol{x}^\top \frac{n_i\boldsymbol{m}_i + n_j\boldsymbol{m}_j}{n_i+n_j} + \frac{(n_i\boldsymbol{m}_i + n_j\boldsymbol{m}_j)^\top(n_i\boldsymbol{m}_i + n_j\boldsymbol{m}_j)}{(n_i+n_j)^2} \right] \\
&\quad - \sum_{\boldsymbol{x}\in D_i} (\boldsymbol{x}^\top\boldsymbol{x} - 2\boldsymbol{x}^\top\boldsymbol{m}_i + \boldsymbol{m}_i^\top\boldsymbol{m}_i) - \sum_{\boldsymbol{x}\in D_j} (\boldsymbol{x}^\top\boldsymbol{x} - 2\boldsymbol{x}^\top\boldsymbol{m}_j + \boldsymbol{m}_j^\top\boldsymbol{m}_j) \\
&= -\frac{(n_i\boldsymbol{m}_i + n_j\boldsymbol{m}_j)^\top(n_i\boldsymbol{m}_i + n_j\boldsymbol{m}_j)}{n_i+n_j} + n_i\boldsymbol{m}_i^\top\boldsymbol{m}_i + n_j\boldsymbol{m}_j^\top\boldsymbol{m}_j \\
&= -\frac{(n_i\boldsymbol{m}_i + n_j\boldsymbol{m}_j)^\top(n_i\boldsymbol{m}_i + n_j\boldsymbol{m}_j)}{n_i+n_j} + \frac{(n_i\boldsymbol{m}_i^\top\boldsymbol{m}_i + n_j\boldsymbol{m}_j^\top\boldsymbol{m}_j)(n_i+n_j)}{n_i+n_j} \\
&= \frac{-2n_in_j\boldsymbol{m}_i^\top\boldsymbol{m}_j + n_in_j\boldsymbol{m}_i^\top\boldsymbol{m}_i + n_in_j\boldsymbol{m}_j^\top\boldsymbol{m}_j}{n_i+n_j} \\
&= \frac{n_in_j}{n_i+n_j}\|\boldsymbol{m}_i - \boldsymbol{m}_j\|^2
\end{aligned} \tag{6}$$

因此, 当

$$\frac{n_i n_j}{n_i + n_j} \|\boldsymbol{m}_i - \boldsymbol{m}_j\|^2 \tag{7}$$

最小时, 合并第 i 类与第 j 类的均方误差增量最小.

Spectral Clustering

3. Given a set of m data points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$, the input to a spectral clustering algorithm typically consists of a matrix A, of pairwise similarities between data points. A is often called the affinity matrix. The choice of how to measure similarity between points is one which often left to the practitioner. A very simple affinity matrix can be constructed as follows:

$$A(i,j) = A(j,i) = \begin{cases} 1, & \text{if } d(\boldsymbol{x}_i, \boldsymbol{x}_j) < \Theta \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The general idea of spectral clustering is to construct a mapping of the data points to an eigenspace of A with the hope that points are well separated in this eigenspace so that something simple like $k - means$ applied to these new points will perform well.

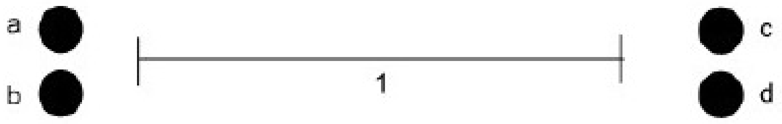
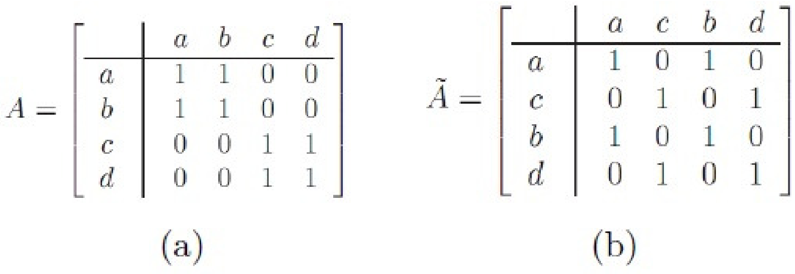


图 1: A simple data set.

As an example, consider forming the affinity matrix for the dataset in Figure 1 using Equation 8 with $\Theta = 1$. Then we get the affinity matrix in Figure 2.



$$A = \left[\begin{array}{c|cccc} & a & b & c & d \\ \hline a & 1 & 1 & 0 & 0 \\ b & 1 & 1 & 0 & 0 \\ c & 0 & 0 & 1 & 1 \\ d & 0 & 0 & 1 & 1 \end{array}\right]$$

(a)

$$\tilde{A} = \left[\begin{array}{c|cccc} & a & c & b & d \\ \hline a & 1 & 0 & 1 & 0 \\ c & 0 & 1 & 0 & 1 \\ b & 1 & 0 & 1 & 0 \\ d & 0 & 1 & 0 & 1 \end{array}\right]$$

(b)

图 2: Affinity matrices of Figure 1 with $\Theta = 1$.

Now for this particular example, the clusters $\{a, b\}$ and $\{c, d\}$ show up as nonzero blocks in the affinity matrix.

This is artificial since we could have constructed the matrix A using an ordering of $\{a, b, c, d\}$. For example, another possible affinity matrix for A could have been as in Figure 2(b).

The key insight here is that the eigenvectors of matrices A and $\tilde{A}$ have the same entries (just permuted). The eigenvectors with nonzero eigenvalues of A are $\boldsymbol{e}_1 = (1, 1, 0, 0)^\top$ and $\boldsymbol{e}_2 = (0, 0, 1, 1)^\top$. And the nonzero eigenvectors of $\tilde{A}$ are: $\boldsymbol{e}_1 = (1, 0, 1, 0)^\top$ and $\boldsymbol{e}_2 = (0, 1, 0, 1)^\top$. Spectral clustering embeds the original data points into a new space by using the coordinates of these eigenvectors. Specifically, it maps the point $\boldsymbol{x}_i$ to the point $\boldsymbol{e}_1(i), \boldsymbol{e}_2(i), \ldots, \boldsymbol{e}_k(i)$ where $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k$ are the top k eigenvectors of A corresponding to the biggest eigenvalues. We refer to this mapping as the spectral embedding.

Hint: This is different from what you learned in class. With the Graph Laplacian L, we are trying to find its eigenvectors corresponding to the smallest eigenvalues. What you should notice here is that the matrix A is not Graph Laplacian, this leads to different algorithms. You can analyze the underlying meaning of this approach as what we did in the class.

In this problem, we will analyze the operation of one of the variants of spectral clustering methods on a simple data set shown in Figure 3:

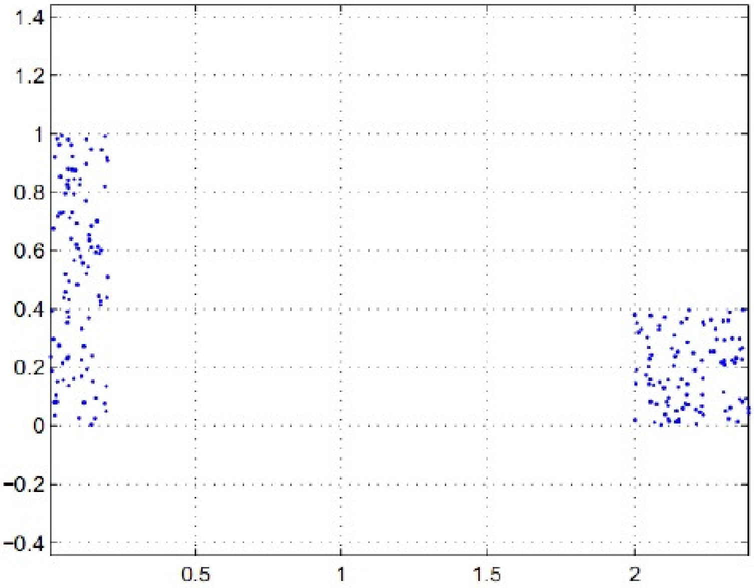


图 3: Dataset with rectangles

3.1 For the data set in the above figure, assume that the first cluster has m_1 points and the second one has m_2 points. If we use Equation 8 to compute the corresponding affinity matrix A, what Θ value would you choose and why?

解: 记图 3 中左侧数据为第 1 堆 (cluster), 右侧数据为第 2 堆. 记第 1 堆数据之间的最大距离为 d_1, 第 1 堆数

据与第 2 堆数据之间的最短距离为 d_2. 为使每堆数据之内的相似性度量为 1 而两堆数据之间的相似性度量为 0, 则需选择

$$\Theta \in (d_1, d_2] \tag{9}$$

根据图中具体数据可知, 选择 $\Theta = 1.5$ 可以满足上述要求.

3.2 The second step is to compute first k dominant eigenvectors of the affinity matrix, where k is the number of clusters we want to have. For the data set in Figure 3 and the affinity matrix defined by Equation 8, is there a value of Θ for which you can analytically compute the first two eigenvalues and eigenvectors? If not, explain why not. If yes, compute and write these eigenvalues and eigenvectors down. What are the other eigenvalues? Explain briefly.

解: 存在, 取 3.1 中 $\Theta = 1.5$ 即可解析计算相似 (affinity) 矩阵 A 的前两个特征值与特征向量.

首先通过调整顺序可写出 A 为

$$A = \begin{bmatrix} \mathbf{1}_{m_1\times m_1} & \mathbf{0}_{m_1\times m_2} \\ \mathbf{0}_{m_2\times m_1} & \mathbf{1}_{m_2\times m_2} \end{bmatrix} \tag{10}$$

易知其前两个特征值为 m_1 与 m_2, 对应的特征向量分别为

$$\boldsymbol{e}_1 = \begin{bmatrix} \mathbf{1}_{m_1\times 1} \\ \mathbf{0}_{m_2\times 1} \end{bmatrix}, \quad \boldsymbol{e}_2 = \begin{bmatrix} \mathbf{0}_{m_1\times 1} \\ \mathbf{1}_{m_2\times 1} \end{bmatrix} \tag{11}$$

由于矩阵 A 的秩 $\operatorname{rank}(A) = 2$, 则其他特征值均为 0.

3.3 We can now compute the spectral embedding of the data points using the k top eigenvectors. For the data set in Figure 3, write down your best guess for the coordinates of the $k = 2$ cluster centers using the Θ that you picked in the first part.

解: 根据 $\boldsymbol{e}_1$ 与 $\boldsymbol{e}_2$ 可知, 两堆的聚类中心分别为

$$\boldsymbol{c}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{c}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{12}$$

Programming: K-means, hierarchical clustering and spectral clustering

Test the clustering algorithms K-means, hierarchical clustering and spectral clustering with different parameters on MNIST dataset or subsets of it when the scale is too large for the algorithm involved.

To compare the effectiveness of different clustering methods, *Normalized mutual information* (NMI) are widely used as a measurement. NMI is defined as following:

$$\mathrm{NMI} = \frac{\sum_{s=1}^{K}\sum_{t=1}^{K} n_{s,t}\log\left(\frac{nn_{s,t}}{n_s n_t}\right)}{\sqrt{\left(\sum_{s=1}^{K} n_s \log\frac{n_s}{n}\right)\left(\sum_{t=1}^{K} n_t \log\frac{n_t}{n}\right)}}. \tag{13}$$

Where n is the number of data points, n_s and n_t denote the numbers of the data in class s and class t, $n_{s,t}$ denotes the number of data points in both class s and class t. For more details and other measurements, google "evaluation of clustering".

4.1 Give a brief analysis of time complexity of each algorithm mentioned above (of standard implementation). Estimate how many samples each algorithm can manage with a reasonable time cost.

(Optional) Can you verify your estimations with experiments? Can you speed it up further?

解: 记 k 为分类数, n 为点数, T 为算法循环次数, 则 K-means 算法的复杂度为 $O(knT)$. 分级聚类算法计算两类距离复杂度 $O(n^2)$, 总共计算 n 次, 所以分级聚类算法的复杂度为 $O(n^3)$. 谱聚类算法计算相似度矩阵的复杂度为 $O(n^2)$, 计算特征值与特征向量的复杂度为 $O(n^3)$, 所以谱聚类算法的复杂度为 $O(n^3)$.

以 CPU 主频为 3GHz 来估算, 若需要 1s 之内计算完成, 则最多计算样本数大约为:

(a) K-means: 3×10^9

(b) 分级聚类: $\sqrt[3]{3} \times 10^3$

(c) 谱聚类: $\sqrt[3]{3} \times 10^3$

4.2 Consider each data set, and use the true number of classes as the number of clusters.

(1) With K-means, will the initial partition affect the clustering results? How can you solve this problem? And do J_e and NMI match? Show your experiment results.

解: 选择样本数量 $n = 3000$, 分别使用 k-means++ [1], 随机初始化与聚类中心重合初始化三种初始化方法进行 K-means 聚类, 其结果如表 1 所示.

表 1: K-means 聚类结果

methods	$J_e \times 10^9$	NMI
k-means++	7.527918	0.495957
random	7.514934	0.503458
coincide	7.700415	0.473963

由表 1 可知, 三种初始化方法对聚类结果有影响, 但是影响并不是很大. 实际上, K-means 可能会受到初始化影

响而陷入局部极小, 对此我们一般可以采用多次随机初始化, 然后取效果最好的结果. 由表 1 也可看出当样本数量相同时, J_e 越小则 NMI 越大, 聚类效果越好. 但是当样本数量发生变化时, J_e 与 NMI 的变化情况则会有所不同.

为了测试 J_e 与 NMI 关于样本数量的变化关系, 选择样本数量 $n = [100, 200, \ldots, 3000]$, 使用 k-means++ 初始化方法得到 K-means 聚类结果 J_e 如图 4 所示, NMI 如图 5 所示.

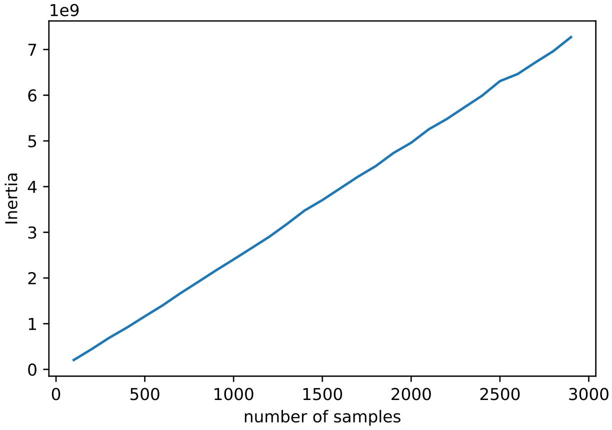


图 4: K-means 算法 k-means++ 初始化 J_e 与样本数量变化关系

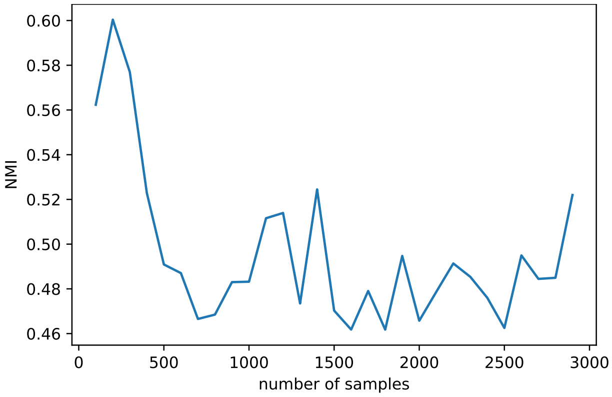


图 5: K-means 算法 k-means++ 初始化 NMI 与样本数量变化关系

使用随机初始化方法得到 K-means 聚类结果 J_e 如图 6 所示, NMI 如图 7 所示.

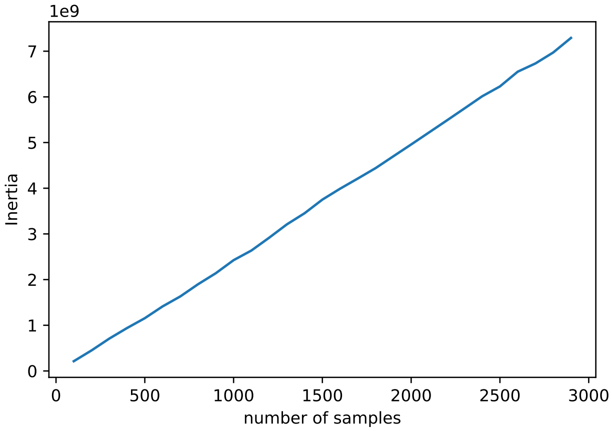


图 6: K-means 算法随机初始化 J_e 与样本数量变化关系

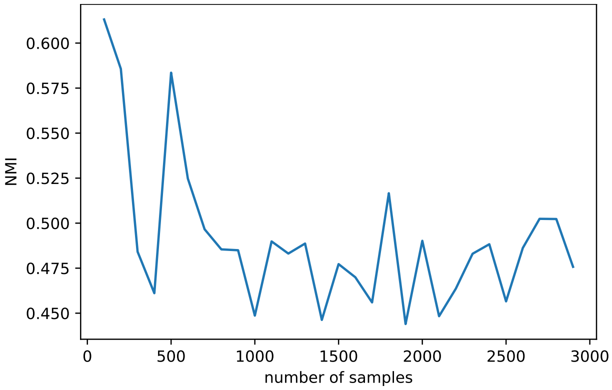


图 7: K-means 算法随机初始化 NMI 与样本数量变化关系

使用聚类中心重合初始化方法得到 K-means 聚类结果 J_e 如图 8 所示, NMI 如图 9 所示.

由此 6 幅对比图可知, 随着样本数量的增长, J_e 单调增加, 但是 NMI 呈现震荡变化, 因此若样本数量发生变化, 则 J_e 与 NMI 的结果不一定匹配 (match).

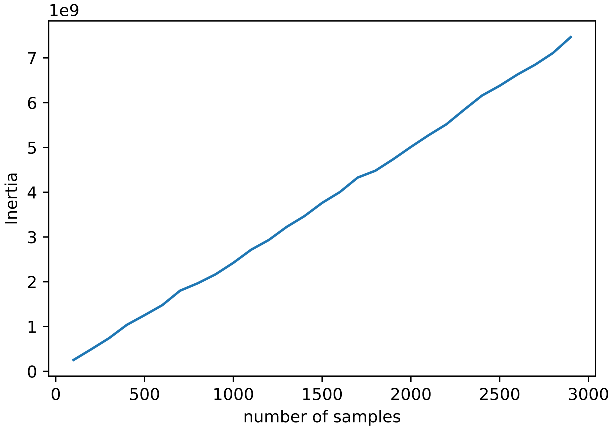


图 8: K-means 算法中心重合初始化 J_e 与样本数量变化关系

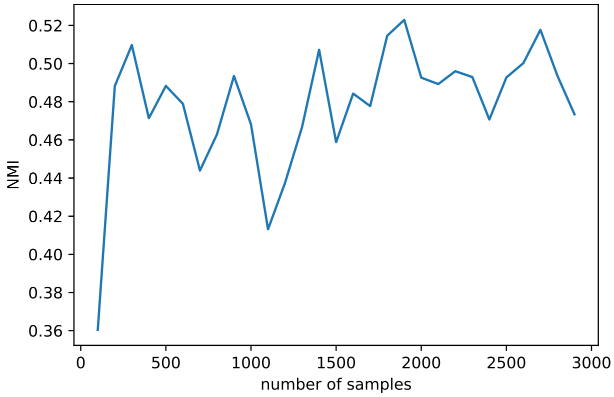


图 9: K-means 算法中心重合初始化 NMI 与样本数量变化关系

(2) When hierarchical clustering is adopted, the choice of linkage method depends on the problem. Give an analysis of linkage method's effects with experiments, and which is better in the sense of NMI?

As introduced in the class, some of the most common metrics of distance between two clusters $\{x_1, \dots, x_m\}$ and $\{y_1, \dots, y_p\}$ are:

- *Single linkage:* Distance between clusters is the *minimum* distance between any pair of points from the two clusters, i.e.,

$$\min_{i,j} \|x_i - y_j\| \tag{14}$$

- *Complete linkage:* Distance between clusters is the *maximum* distance between any pair of points from two clusters, i.e.,

$$\max_{i,j} \|x_i - y_j\| \tag{15}$$

- *Average linkage:* Distance between clusters is the *average* distance between all pairs of points from two clusters, i.e.,

$$\frac{1}{mp} \sum_{i=1}^{m} \sum_{j=1}^{p} \|x_i - y_j\| \tag{16}$$

解: 为了测试三种不同距离度量的 NMI 关于样本数量的变化关系, 选择样本数量 $n = [100, 200, \dots, 3000]$, 使用分级聚类方法得到 NMI 如图 10 所示.

由图 10 可知, 最近距离度量的 NMI 普遍低于另外两种度量的 NMI, 最远距离度量的 NMI 与平均距离度量的 NMI 相互交织, 但是大多时候最远距离度量的 NMI 略优于平均距离度量. 这说明 MNIST 数据的各个类别之间距离较近, 可能出现因为边缘距离近导致类别连接的情况. 同时 MNIST 的数据基本围绕在聚类中心各个方向的方差附近, 所以最远距离度量效果较好.

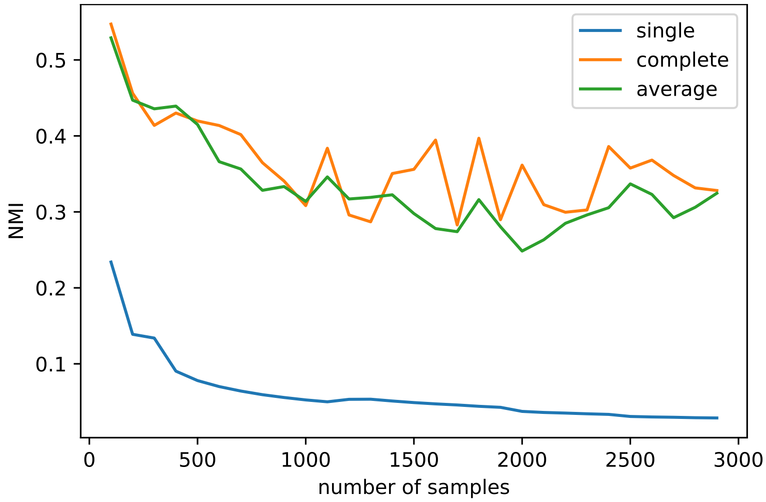


图 10: 分级聚类算法不同距离度量 NMI 与样本数量变化关系

(3) As for spectral clustering, give an experimental analysis of the choice of similarity graph and corresponding parameters. Which one is better?

解: 分别使用高斯核的全连接图和 k 近邻图进行相似矩阵构建, 谱聚类 NMI 结果如表 2 所示. 由表 2 可知最近邻的效果更好, 选择高斯核的全连接时, 很多点是孤立的因而没有构成一个完全图, 所以聚类效果很差. 而最近邻中 $k=5$ 的效果最好, 若 k 太大, 有可能导致不同类之间的样本连接起来.

表 2: 谱聚类全连通图与 k 近邻图实验结果

methods	NMI
rbf, $\gamma=1$	0.101404
rbf, $\gamma=2$	0.076044
rbf, $\gamma=4$	0.074711
knn, $k=5$	0.614761
knn, $k=50$	0.519622
knn, $k=100$	0.468890

4.3 In practice, we may not know the true number of clusters much. Can you give a strategy to identify the cluster number automatically for each algorithm? Show your results.

解: 可以使用轮廓系数 (Silhouette Coefficient) [2] 来进行聚类数量的选择, 其计算公式为

$$s(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \tag{17}$$

其中, $b(i)$ 是点 i 到任何其他堆中所有点的最小平均距离, $a(i)$ 是点 i 到其所在堆中其他点的平均距离.

绘出 K-means 的轮廓系数曲线如图 11 所示, 在轮廓系数曲线中 $k=2$ 时轮廓系数最大, 按照最大轮廓系数选择

结果为 $k=2$. 实际上, 我们可以结合一些先验知识, $k=11$ 的轮廓系数仅次于 $k=2$, 与实际的类别数 $k=10$ 更为接近.

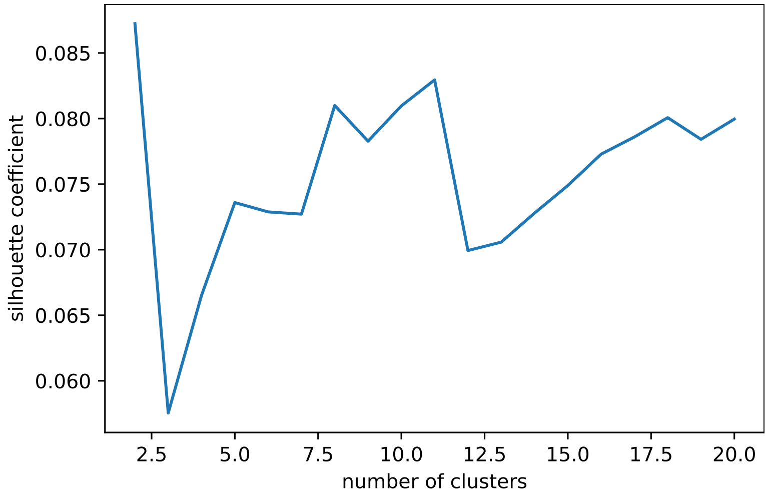


图 11: K-means 算法轮廓系数随聚类类别数变化曲线

4.4 According to the above analysis, which method do you prefer? Why?

解: 以上三种聚类方法各有优劣, 聚类方法的选择要根据样本的分布特性和数量综合考虑. 比如若样本点成团状分布或者样本数很大时, 则用 K-means 算法能取得较好效果, 且速度快; 而多级聚类和谱聚类在样本数很大时由于时间复杂度过大而无法使用. 当样本数量较少时, 可以选择基于最近邻图的谱聚类方法, 其聚类的效果较好, 而且不像分级聚类那样受距离度量选择的影响大.

参考文献

[1] sklearn.cluster.KMeans, https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans.

[2] Rousseeuw, Peter J. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis.” Journal of computational and applied mathematics, 20, 1987: 53-65.