lec11 Clustering Problem & Clustering Evaluation

Clustering Problem

给定一个数据集，将其对象划分为若干个集合（簇）$C_1, \ldots, C_k$，使得每个簇中的对象彼此“相似”

相似性的具体定义依赖于如何定义“相似”的概念。这可以根据不同的应用场景和需求进行定义，例如：

- 欧氏距离（用于衡量数值数据的相似性）
- 余弦相似度（用于衡量向量之间的相似性）
- Jaccard 相似系数（用于衡量集合之间的相似性）
- 其他距离或相似度度量方法

聚类数据的用途

- 数据摘要（Data summarization）：聚类可以帮助将大规模数据简化为几个代表性簇，从而更容易理解和分析数据。例如，将客户数据聚类，可以得到不同类型的客户群体
- 主题检测（Topic detection）：在文本分析中，聚类可以用于检测文档中的主题。例如，将新闻文章聚类，可以发现不同的新闻主题
- 可视化（Visualization）：聚类可以帮助将高维数据映射到低维空间，从而进行可视化展示。例如，将数据点聚类后，可以在二维或三维空间中直观地展示数据结构
- 异常检测（Outlier detection）：通过聚类，可以发现数据中的异常点（离群点）。这些异常点通常不属于任何一个簇，可以进一步分析以发现潜在的问题或异常情况
- 社区检测（Community detection）：在社交网络分析中，聚类可以用于检测社交网络中的社区结构。例如，识别社交网络中的不同用户群体，以便更好地了解用户之间的关系

聚类是无监督学习（Unsupervised Learning）

聚类是无监督学习的一种方法，用于在没有标签的数据中发现数据的内在结构

- 监督学习（Supervised Learning）
  - 特点：提供训练实例的标签
  - 应用：模型使用已知标签的数据进行训练，以便对新数据进行预测。例如，分类和回归任务
- 无监督学习（Unsupervised Learning）
  - 特点：不提供训练实例的标签
  - 应用：模型在没有标签的数据上进行训练，发现数据的内在结构。例如，聚类和降维任务
- 半监督学习（Semi-supervised Learning）
  - 特点：提供部分带标签和部分未带标签的训练实例
  - 应用：结合少量标注数据和大量未标注数据进行训练，以提高模型的性能。例如，半监督分类
- 无标签数据的学习
  - 问题：如果没有任何标签，我们能从训练数据中学到什么？
  - 答案：我们仍然可以学习特征的相似性和分布，这可以用于创建丰富的特征空间，为监督学习提供帮助（如果需要）

一些总体观点

一个数据集可以以多种方式聚类，不同的聚类方法可以揭示数据的不同方面；但是并没有单一的正确或错误的聚类，聚类结果没有绝对的对错，只有对同一数据的不同视角

测量聚类算法质量的方法：

- 外在方法（Extrinsic methods）
  - 外在方法通过比较聚类算法产生的簇与某个参考（黄金标准或真实）簇集来评估聚类质量。
  - 这种方法依赖于有一个已知的、可信的簇划分作为参考。
- 内在方法（Intrinsic methods）
  - 内在方法仅使用对象在簇中的分配来评估聚类质量，而不依赖于外部参考。
  - 这些方法通常基于簇的紧密度和分离度等内部特性来评估。

几种常见的聚类算法

- 代表性（基于中心点）聚类算法（Representative-based）
  选择 k 个代表（中心点），将数据集中每个元素分配给一个代表，并迭代更新划分
  - k-means：通过最小化簇内方差来更新中心点
  - k-medoids：类似于 k-means，但选择实际数据点作为中心点，并使用绝对差而不是平方差进行计算
- 层次聚类（Hierarchical）
  创建一个簇的层次结构（树状图，dendrogram）
  - 凝聚层次聚类（自底向上）：从每个数据点开始，不断合并最相似的簇，直到所有数据点被聚为一个簇
  - 分裂层次聚类（自顶向下）：从一个包含所有数据点的簇开始，不断分裂最不相似的簇，直到每个数据点成为一个单独的簇
- 基于图的聚类（Graph-based clustering）
  将数据集表示为图，节点表示数据点，边表示数据点之间的相似性
  - 社区检测（Community detection）：例如，模块度优化（Modularity optimization），用于发现图中的社区结构
  - 图割算法（Graph-cut algorithms）：例如，谱聚类（Spectral Clustering），通过最小化图的割值来进行聚类
- 其他类型
  除了上述方法，还有许多其他类型的聚类算法，例如基于密度的聚类（如DBSCAN）、基于网格的聚类、模糊聚类等

Clustering Evaluation

聚类质量评估方法有：

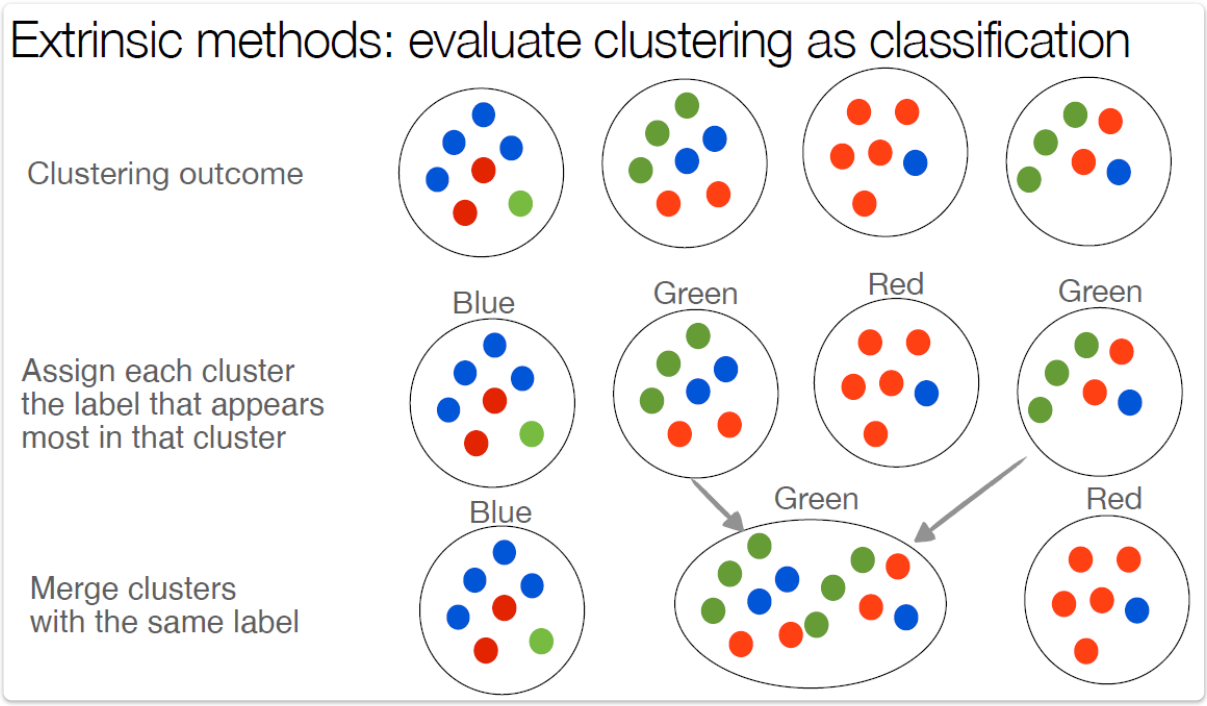
1. 外在方法（监督方法）：
   - 特点：使用真实标签（ground truth labels）进行评估
   - 方法：将聚类结果与已知的真实标签进行比较，根据比较结果给聚类结果打分
   - 应用：当数据集有已知的标签时，可以使用这种方法。例如 Precision、Recall、Accuracy 与 F-score等
2. 内在方法（无监督方法）：
   - 特点：不使用任何真实标签，仅依赖于聚类结果本身
   - 方法：仅使用对象在簇中的分配来评估聚类质量，检查簇之间的分离度和簇内的紧凑度

- 应用：当数据集没有已知的标签时，可以使用这种方法。例如，使用轮廓系数（Silhouette Coefficient）、戴维斯-鲍丁指数（Davies-Bouldin Index）、轮廓分数（Silhouette Score）等指标

外在方法

将聚类视为分类进行评估

1. 对于每个簇，分配在该簇中出现次数最多的标签作为该簇的标签
2. 将具有相同标签的簇合并在一起
3. 计算每个标签类型的Precision、Recall和F-score
4. 计算宏平均（macro-average）：
   - Precision
   - Recall
   - F-score



B-CUBED Measure

在不标记任何簇的情况下评估聚类

$$\text{precision}(x) = \frac{\text{No. of items in } C(x) \text{ with } A(x)}{\text{No. of items with } A(x)}$$
$$\text{recall}(x) = \frac{\text{No. of items in } C(x) \text{ with } A(x)}{\text{Total no. of items with } A(x)}$$

其中

- $A(x)$ 是 x 的标签
- $C(x)$ 是 x 所在簇的 ID

计算平均值：

$$\text{Precision} = \frac{1}{N}\sum_{p\in\text{DataSet}}\text{Precision}(p)$$

$$\text{Recall} = \frac{1}{N}\sum_{p\in\text{DataSet}}\text{Recall}(p)$$

$$\text{F-Score} = \frac{1}{N}\sum_{p\in\text{DataSet}}F(p)$$

其中

- $F(p)$ 是实例 p 的 F1-score
- N 是是所有簇中实例的总数

内在方法

轮廓系数（Silhouette Coefficient）

1. 设 $C_1,\dots,C_k$ 为聚类
2. 计算 $a(x)$ 和 $b(x)$：
   对于对象 x（假设 $x\in C_i$），定义 $d(x,y)$ 为 x 与 y 之间的距离
   - $a(x)$：x 与 x 所在簇内其他点的平均距离

     $$a(x) = \frac{1}{|C_i|-1}\sum_{y\in C, y\neq x}d(x,y)$$

     $a(x)$ 衡量 x 与其所在簇的异质性，值越小越好，表示 x 与其簇内点更相似
   - $b(x)$：x 与下一个最近簇内所有点的平均距离

     $$b(x) = \min_{\substack{j=1,\dots,k\\ j\neq i}}\frac{1}{|C_j|}\sum_{y\in C_j}d(x,y)$$

     $b(x)$ 衡量 x 与其最近邻簇的匹配度，值越大越好，表示 x 与其他簇的点不相似
3. 轮廓系数 $s(x)$：

   $$s(x) = \frac{b(x)-a(x)}{\max\big(a(x),b(x)\big)}$$

   如果 $|C_i|=1$，则 $s(x)=0$
   - $s(x)$ 接近 1 表示 x 被恰当地聚类
   - $s(x)$ 接近 -1 表示 x 更适合聚类到其他临近簇
   - $s(x)$ 接近 0 表示 x 位于两个自然簇的边界
4. 数据集的轮廓系数是所有对象的轮廓系数的平均值