文本聚类方法

聚类方法概述

1. 聚类：也称**聚类分析**，指将样本分到不同的组中，使得同一组中的样本差异尽可能的小，而不同组的样本差异尽可能的大。
   - **簇**：聚类得到的不同的组称为簇。
2. 聚类与分类的区别：
   - 聚类的样本**不具有类别标号**，而分类的样本具有**类别标号**；
   - 聚类是**无监督学习**，而分类是**有监督学习**，因此分类里有训练和测试，而聚类中没有训练；
   - 聚类提供了一种新的处理模式：先对数据集划分组，再给有限的组标号，从而避免了分类任务中收集和标记数据集的昂贵代价。
3. 应用领域：（1）经济领域；（2）生物学领域；（3）数据挖掘领域。
4. 数据类型：（1）二元变量；（2）**非对称**二元变量；（3）分类变量；（4）序数变量；（5）比例标度变量；（6）混合类型变量。
5. 兼类划分：
   - **软聚类**：一个对象可以属于多个聚类集合，但是对应的概率不同；
   - **硬聚类**：每个对象只能属于一个聚类集合。
6. 主要聚类方法：（1）**划分聚类方法**；（2）**层次聚类方法**；（3）**密度聚类方法。**
7. 其他聚类方法：（1）网格聚类方法；（2）基于模型的聚类方法；（3）基于条件约束的聚类方法等。

划分聚类方法

1. 划分聚类：给定n个对象的数据集，划分聚类将构造对数据的k个划分，每个划分代表一个簇。
   - k个划分/簇满足：（1）**每个簇至少包含一个对象**；（2）**每个对象属于且仅属于一个簇。**
   - 主要划分聚类算法：k-means算法，PAM算法。
2. **类内差异**和**类间差异**：

> 类内差异：**衡量聚类的紧凑性**，类内差异可以用特定的距离函数来定义：

$$w(C)=\sum_{i=1}^{k}w(C_i)=\sum_{i=1}^{k}\sum_{x\in C_i}d(x,\overline{x}_i)^2$$

其中$\{C_i\}$为划分集合，$\overline{x}_i$为划分C_i的聚类中心。$d(x,\overline{x}_i)$为C_i内某点x到聚类中心的距离。

类间差异：**衡量不同聚类之间的距离**，类间差异定义为聚类中心间的距离：

$$b(C)=\sum_{1\le j<i\le k}d(\overline{x}_j,\overline{x}_i)^2$$

聚类的总体质量可被定义为两者的简单组合，如$w(C)/b(C)$。

k-means算法

3. k-means算法：
   - 算法流程：
     1. 从n个数据对象任意选择k个对象作为**初始聚类中心**，依据最小距离划分出k个簇；
     2. 根据每个聚类对象的均值（中心对象），计算每个对象与这些中心对象的距离，并根据最小距离**重新对相应对象划分**；
     3. 重新计算每个（有变化）**聚类的均值**（中心对象）；
     4. 计算标准测度函数，当满足一定条件时则终止（如函数收敛）；否则返回步骤2。
   - 优点：（1）简单、快速；（2）可伸缩、高效；（3）对密集的结果簇效果好。
   - 缺点：
     - 需要对簇的平均值做出定义（某些应用中可能不适于定义）；
     - 需要事先给定k，且算法对初值敏感（不同的初值可能导致不同的结果）；
     - 不适于发现*非凸面形状的簇或大小差别大的簇*，且对"噪声"和孤立点数据敏感。
   - 变体：k-中心点算法：**不选用簇平均值作为参照点，而选用簇中位置最中心的对象即中心点，作为参照点。**

PAM算法

4. PAM算法：是一种k-中心点算法。
   - 基本概念：（1）中心点：即代表对象；（2）非代表对象：即除代表对象之外的点。
   - **反复地用非代表对象来代替中心点，以找出更好的中心点，从而改进聚类质量。**

层次聚类方法

5. 层次聚类方法：对给定的数据集进行**层次的分解**，直到某种条件满足为止。
   - **凝聚的层次聚类**：自底向上的策略，代表算法为AGNES算法；
   - **分裂的层次聚类**：自顶向下的策略，代表算法为DIANA算法；

AGNES算法

6. **AGNES**算法：将每个对象作为一个簇，再令这些簇根据某些准则被一步步合并。
   - 算法流程：
     1. 将**每个对象当成一个初始簇**；
     2. 计算任意两个簇的距离，并找到最近的两个簇；
     3. 合并两个簇，生成新的簇的集合；
     4. 重复第2,3步，直到满足终止条件。
   - 簇的距离：

    - 最小距离：$d_{\min}(C_i, C_j) = \min\limits_{p \in C_i, q \in C_j} |p - q|$（单链接方法）；
    - 最大距离：$d_{\max}(C_i, C_j) = \max\limits_{p \in C_i, q \in C_j} |p - q|$；
    - 均值距离：$d_{\text{mean}}(C_i, C_j) = |\overline{p} - \overline{q}|$；
    - 平均距离：$d_{\text{avg}}(C_i, C_j) = \dfrac{1}{n_i \cdot n_j} \sum\limits_{p \in C_i} \sum\limits_{q \in C_j} |p - q|$。
  - 复杂度：$\mathcal{O}(n^2)$。

DIANA算法

7. **DIANA**算法：将所有样本放入一个簇，然后选择一个簇，根据某些准则进行分裂。
   - 度量指标：
     - 簇的直径：$d_{\max}(C_i) = \max\limits_{p,q \in C_i} |p - q|$；
     - 平均相异度：$d_{\text{avg}}(p, C_i) = \dfrac{1}{n_i} \sum\limits_{q \in C_i} |p - q|$。
   - 算法流程：
     1. 将**所有对象作为一个初始簇**；
     2. 找出所有簇中有**最大直径**的簇C。在C中找出**平均相异度最大**的一个点p，将其放入**splinter group**。其余点放入**old party**中；
     3. 在old party中选择新的点q，计算q到splinter group中的点的平均距离D_1，再计算q到old party中的点的平均距离D_2，保持$D_2 - D_1$的值。选取$D_1 - D_2$**最大的点**q'，若$D_1 - D_2 > 0$，则将q'分配到splinter group中；
     4. 重复步骤3，直到没有新的old party的点被分配到splinter group。splinter group和old party为被选中的簇分裂成的两个簇；
     5. 重复步骤2,3,4，直到满足终止条件。

Birch算法

8. **Birch**算法：是层次聚类算法之一，引入了聚类特征和聚类特征树。
   - **聚类特征树**：树中的每个结点都可以用其**聚类特征**（CF）表示，形式为(N, LS, SS)，其中N为簇内样本个数，LS为N个点的线性和，SS为样本的平方和：

$$LS = \sum_{i=1}^{N} \vec{P}_i, \quad SS = \sum_{i=1}^{N} |\vec{P}_i|^2$$

   - CF的**合并性**：若簇A的特征为(N_A, LS_A, SS_A)，簇B的特征为(N_B, LS_B, SS_B)，则合并以后的簇AB的特征为$(N_A + N_B, LS_A + LS_B, SS_A, SS_B)$。
   - CF树的参数：（1）每个非叶子结点最多有B个簇分支；（2）每个叶子结点最多有L个簇分支；（3）每个簇分支的直径不超过阈值T。
   - 算法流程：
     1. 从根节点开始，自上而下选择最近的子结点；
     2. 到达叶节点后，**检查最近的簇元组**L_i**能够吸收新样本**。若是，则更新CF值；
     3. 若否，则**检查在当前叶节点**是否可以添加新的簇元组。若是，则添加；
     4. 若否，则**分裂最远的一对簇元组**，作为种子，按最近距离重新分配其他簇元组。
     5. 更新每个非叶节点的CF值。若产生分裂节点，则在**父节点**中插入分裂的簇元素，并检查分裂是否合法。若不合法，继续向上插入，直到根节点。
   - 复杂度：$\mathcal{O}(n)$。

密度聚类方法

9. 密度聚类方法：
    - 思想：**只要一个区域中的点的密度大于某个阈值，就把它加到与之相近的聚类中去。**
    - 优点：可以发现**任意形状**的簇，且对噪声不敏感。
    - 主要算法：DBSCAN、OPTICS、DENCLUE。

DBSCAN算法

10. 基本概念：
    - ϵ：若两个对象之间的距离不超过ϵ，则这两个对象将是同一类的；
        - ϵ-邻域：以给定对象为中心的半径为ϵ的区域。
    - $MinPts$：构成一个簇的最小的点的个数；
    - **核心对象**：若一个对象的ϵ-邻域内包含至少$MinPts$个对象，则称之为核心对象；
    - **直接密度可达**：给定对象集合D，$p, q \in D$。若p在q的ϵ-邻域内，且q是核心对象，则称**对象p从对象q出发是关于ϵ和$MinPts$直接密度可达的。**
    - **密度可达**：若p从q出发是关于ϵ和$MinPt$**（直接）密度可达的**，且q从r出发是关于ϵ和$MinPt$**（直接）密度可达的**，则称p是从r出发关于ϵ和$MinPts$**密度可达的。**
    - **密度相连**：若存在对象o，使得对象p, q均从o出发关于ϵ和$MinPts$密度可达，则对象p, q是**关于ϵ和$MinPts$密度相连的。**
    - **簇**：基于密度可达性的最大的密度相连对象的集合。
    - **噪声**：不包含在任何簇中的对象被认为是噪声。
11. **DBSCAN**算法：
    - 算法流程：
        1. 依次处理每个对象：若为核心对象，则**找出所有从该点密度可达的对象**；
        2. 依次处理每个核心对象：对所有核心对象的ϵ邻域所有直接密度可达点，**找到最大的密度相连对象集合**，中间涉及一些密度可达对象的合并。

OPTICS算法

12. 基本概念补充：
    - **核心距离**：**只有核心对象才能定义核心距离。**对象p的核心距离指使p称为核心对象的最小ϵ'。
    - **可达距离**：对象q到p的可达距离指p的**核心距离和p与q之间欧氏距离**的**较大值。**
13. **OPTICS**算法：不显示产生的结果类簇，而是为聚类分析生成一个**簇排序**，该排序代表了**各样本点基于密度的聚类结构。**
    - 算法流程：
        1. 创建一个**有序队列**（用来存储**核心对象及其直接可达对象**，并按**可达距离**升序排列）和一个**结果序列**（用来存储样本点的输出次序）；
        2. 选择一个**未处理（不在结果序列中）**且为**核心对象**的样本点，找到其所有**直接密度可达样本点。**如果该样本点不存在于结果序列中，则放入有序队列，并按**可达距离**升序排列；
        3. 若有序队列为空，则调至步骤2；否则，从有序队列中**取出第一个样本点**（即可达距离最小的样本点）**进行拓展。**若该取出的样本点不在结果序列中，则将其放入结果序列。拓展步骤如下：
            1. **判断是否为核心对象**，否则不处理；是则找到其**所有直接密度可达对象**；

2. **判断这些直接密度可达对象是否已经存在结果序列中**，是则不处理；否则进入下一步；
3. **若有序队列中已存在该直接密度可达点，且新的可达距离小于旧的可达距离**，则用新值取代，重新排序；
4. **若不存在该直接密度可达点**，则插入该点，并重新排序。

DENCLUE算法

14. **DENCLUE**算法：基于一组密度分布函数的聚类算法。
    - 主要思想：一个样本的影响可以用数学函数形式化建模，该函数称为**影响函数**，描述**数据点对其邻域的影响。**数据空间的整体密度则用**所有数据点的影响函数的和**来建模。簇可通过**识别密度吸引点**数学确定，其数据吸引点是**全局密度函数的局部最大值。**

其他聚类方法

15. 小波变换聚类：把小波变换应用到特征空间的一种多分辨率聚类算法。
    - 既是基于网格的，又是基于密度的。