第9章：无监督学习

蒋良孝 | 中国地质大学（武汉）

CUG-Miner 机器学习与数据挖掘团队

ljiang@cug.edu.cn

http://grzy.cug.edu.cn/jlx/

本章内容

一、无监督学习基础知识

二、K均值聚类算法

三、K均值聚类算法的变种

四、K均值聚类算法的理解

一、无监督学习基础知识

- 提及无监督学习，我们首先会想到聚类。聚类跟分类的区别在于训练样本的类标记是未知的。
- 所谓聚类就是将对物理或抽象对象的集合分组成为由类似的对象组成的多个簇的过程。
- 聚类生成的组称为簇，簇是数据对象的集合。簇内部的任意两个对象之间具有较高的相似度，而属于不同簇的两个对象之间具有较高的相异度。
- 相似度和相异度可以根据描述对象的属性值来计算，对象间的距离是最常采用的相异度度量指标。相似度与相异度通常成反比函数关系。

一、无监督学习基础知识

- 聚类既能作为一个单独过程，用于找寻数据内在的分布结构，也可作为分类等其他学习任务的前驱过程。
- 比如，在一些商业应用中需对新用户的类型进行判别，但定义用户类型对商家来说却可能不太容易，此时往往可先对用户数据进行聚类，然后再根据聚类结果将每个簇定义为一个类，最后再基于这些类训练分类模型，用于判别新用户的类型。

一、无监督学习基础知识

- 基于不同的学习策略，人们设计出多种不同类型的聚类算法，很难对这些聚类算法提出一个简洁的分类。大体上，主要的聚类算法可以分为如下五类：

✓ 1) 基于划分的方法：

✓ 2) 基于层次的方法：

✓ 3) 基于密度的方法：

✓ 4) 基于网格的方法：

✓ 5) 基于模型的方法：

一、无监督学习基础知识

- 下面就基于划分的方法做一个简单的介绍。简单说，基于划分的方法就是采用目标函数最小化的策略，通过迭代把数据对象划分成K个组，每个组为一个簇。
- 基于划分的方法需要满足如下两个条件：

✓ 1）每个分组至少包含一个对象；

✓ 2）每个对象属于且仅属于某一个分组。

- 基于划分的方法主要包括K均值（***k*-means**）聚类算法及其变种：K众数(***k*-modes**)、K原型(***k*-prototypes**)、K中心点(***k*-medoids**)、以及K分布(***k*-distributions**)。

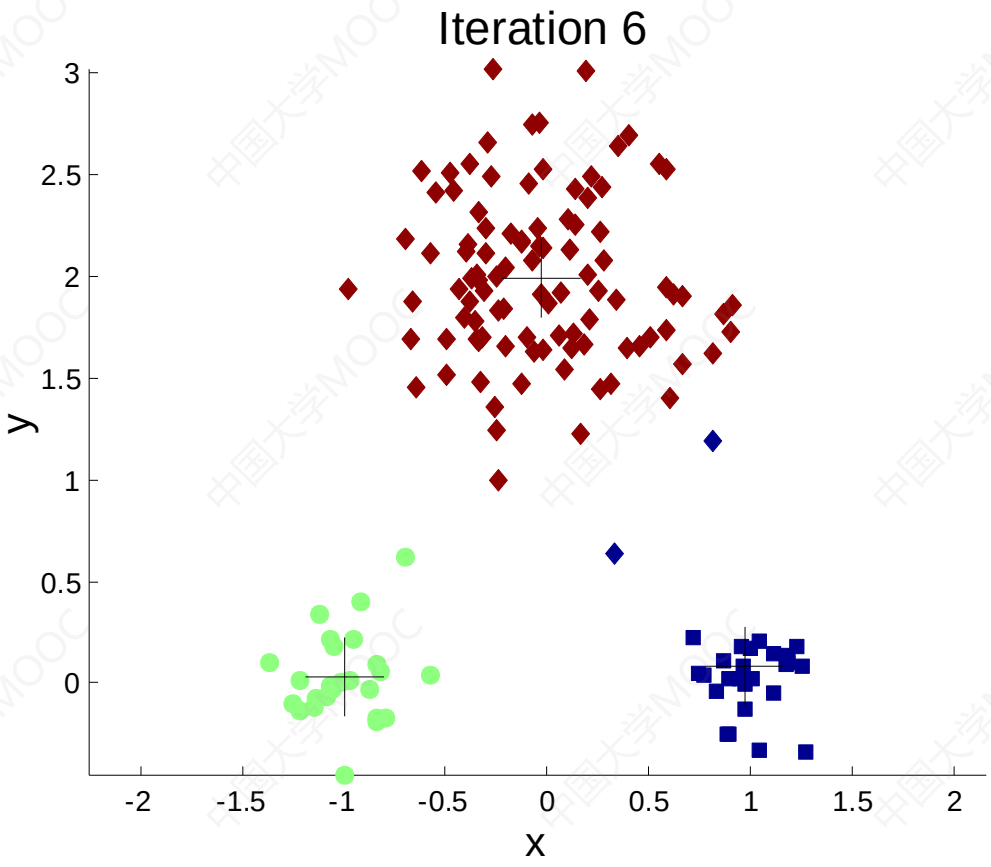
二、K均值聚类算法

K均值(*k*-means) [MacQueen, 1967]

- 输入：簇的数目K和包含n个对象的数据集D
- 输出：K个簇的集合。
- 方法：
  1. 从D中任意选择K个对象作为初始簇的质心；
  2. 计算每个对象与各簇质心的距离，并将对象划分到距离其最近的簇；
  3. 更新每个新簇的质心；
  4. 重复执行第2-3步，直到簇中的对象不再变化。

二、K均值聚类算法

演示

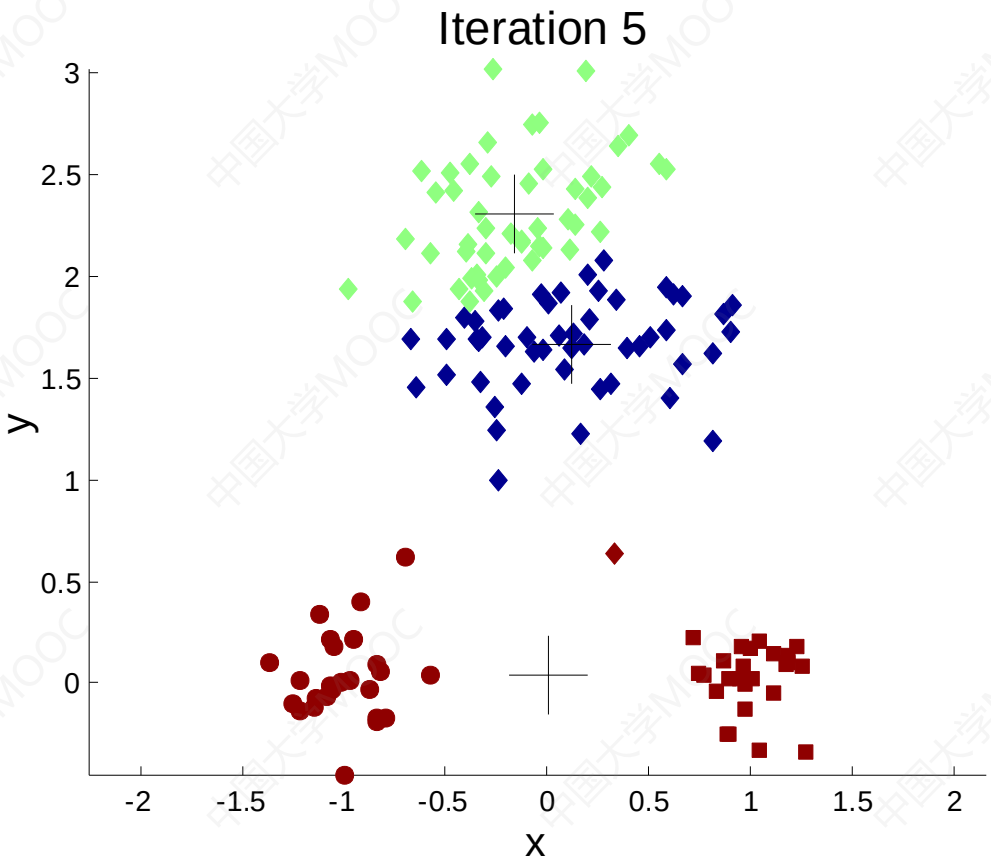


二、K均值聚类算法

对K均值聚类算法的几点说明：

- 只适合于数值属性数据，当它碰到名词性属性数据的时候，均值可能无定义。
- 簇的质心就是簇中所有对象在每一维属性上的均值组合而成的虚拟点，并非实际存在的数据点。
- 对噪声和离群点（孤立点）数据是敏感的，因为它们的存在会对均值的计算产生极大的影响。
- 对象到质心的距离通常使用欧式距离来计算。
- 要求用户事先给出要生成簇的数目，即K值要已知。
- 算法收敛的速度和结果容易受初始质心的影响。

二、K均值聚类算法



三、K均值聚类算法的变种

K众数算法(*k*-modes) [Huang, 1997]

- K均值算法不能聚类名词性属性数据，要聚类名词性属性数据需解决两个问题：1）簇质心的计算问题；2）对象到质心距离的计算问题。
- ✓ 对于第1个问题，可以用众数(mode)去替换均值(mean)，名词性属性的众数就是具有最高频率的属性值。因此，簇的质心就是簇中所有对象在每一维属性上的众数组合而成的虚拟点。
- ✓ 对于第2个问题，可以用适合于名词性属性的距离函数，比如用OM距离去替换欧式距离。

三、K均值聚类算法的变种

K原型算法(*k*-prototypes)[Huang, 1997]

- 如果要聚类的数据既有数值属性又有名词性属性属性，那么我们只需把数据对象分解到每一维上，然后根据每一维的属性类型分别进行数值属性和名词性属性处理。
- ✓ 对于第1个问题，簇的质心就是簇中所有对象在每一维属性上的均值或者众数组合而成的虚拟点。
- ✓ 对于第2个问题，可以用适合于混合属性的距离函数，比如，Heterogeneous Euclidean-Overlap Metric (HEOM)距离，去替换Euclidean距离。

三、K均值聚类算法的变种

K中心点(*k*-medoids)[Kaufman & Rousseeuw, 1987]

- 在K均值算法中，簇的质心就是簇中所有对象在每一维属性上的均值组合而成的虚拟点。因此，当数据中存在噪声和离群点（孤立点）时，它们的存在就会对均值的计算产生极大的影响，进而使得计算得到的质心严重脱离了它本该所在的位置。
- ✓ 为了减轻K均值算法对孤立点的敏感性，K中心点算法被提出。K中心点算法不直接采用簇中对象的均值作为簇中心，而选用簇中离均值最近的实际对象作为簇中心。

三、K均值聚类算法的变种

K分布(*k*- distributions)[Cai, Wang & Jiang, 2007]

- 前面四种算法不仅需要计算簇的质心，还要计算对象到质心（中心点）的距离。
- 有没有哪种算法可以避开：1）簇质心的计算问题；2）对象到质心距离的计算问题。
- 这就是K分布算法的设计动机。K分布算法首先将所有对象随机划分成K个非空且互不相交的簇，然后计算每个对象在每个簇上的联合概率分布，并将其分配给具有最大联合概率分布的簇，一遍完成之后，再更新每个新簇包含的对象，此过程重复执行，直到簇的对象不再变化。

三、K均值聚类算法的变种

K分布(*k*- distributions)[Cai, Wang & Jiang, 2007]

- 输入：簇的数目K；包含n个对象的数据集D
- 输出：K个簇的集合。
- 方法：
  1. 将D随机划分成K个非空且互不相交的簇；
  2. 计算每个对象在每个簇上的联合概率分布，并将其分配给具有最大联合概率分布的簇；
  3. 更新每个新簇的对象；
  4. 重复执行第2-3步，直到簇的对象不再变化。

三、K均值聚类算法的变种

K分布(*k*- distributions)[Cai, Wang & Jiang, 2007]

- K分布算法需要反复计算每个对象$< a_1, a_2, \ldots, a_m >$在每个簇上的联合概率分布$P(a_1, a_2, \ldots, a_m)$。因为从一个给定的数据集中直接估计$P(a_1, a_2, \ldots, a_m)$是不现实的，所以为了简化计算，可以加入一些约束条件，比如假定在簇内所有属性是完全相互独立的。这样就可得到$P(a_1, a_2, \ldots, a_m) = \prod_{i=1}^{m} P(a_i)$。
- 当然这个假设在现实生活中不大可能会成立，但毕竟为解决联合概率分布的计算问题提供了一种可行解，至于更优解可借鉴贝叶斯学习的经验来解决。

四、K均值聚类算法的理解

其实，我们可以从分类的角度来理解K均值聚类算法：

首先，我们再来回顾一下K均值算法的步骤。

- 输入：簇的数目K；包含n个对象的数据集D
- 输出：K个簇的集合。
- 方法：
  1. 从D中任意选择K个对象作为初始簇的质心；
  2. 计算每个对象与各簇质心的距离，并将对象划分到距离其最近的簇；
  3. 更新每个新簇的质心；
  4. 重复执行第2-3步，直到簇的质心不再变化。

四、K均值聚类算法的理解

其实，我们可以从分类的角度来理解 K均值聚类算法：

- 算法第1步：从D中任意选择K个对象作为初始簇的质心。这相当于是在选择这K个对象作为训练样本，并给训练样本随机分配了类标记。
- 算法第2步：计算每个对象与各簇质心的距离，并将对象划分到距离其最近的簇。这相当于是在利用最近邻学习中的1近邻算法分类每一个对象。
- 算法第3步：更新每个新簇的质心。这相当于是在更新训练样本。
- 算法第4步：重复执行第2-3步，直到簇的质心不再变化。这相当于是在反复迭代利用1近邻算法分类每一个对象，直到分类结果不再发生变化，即算法收敛。

K均值聚类=随机初始标记+有限次迭代收敛的1近邻分类

四、K均值聚类算法的理解

再推广一下就是：

- 虽然聚类是一种无监督学习，给定的已知样本都没有类标记。但当聚类算法完成随机初始划分之后，每个样本点就相当于都有了类标记，只不过因为初始划分是随机选择的，这些类标记离真实的类标记可能还相差很远。
- 一旦样本点有了类标记，我们就可以利用监督学习技术来进行分类学习。因为这些类标记可能还存在错误，利用构建的分类器分类一遍样本是远远不够，还需要然后经过反复迭代分类多遍，不断更新这些类标记。

既然这样，我们是不是可以得出如下结论：

聚类=随机初始标记+有限次迭代收敛的分类？

如果结论成立，每一个聚类算法是不是都存在一个对应的分类算法？ K均值聚类对应于1近邻分类！K分布呢？其他呢？