第八章 最近邻学习
基础知识
基本思想
常见问题
第九章 无监督学习
基础知识
K均值聚类算法（$k-means$）
K众数算法（$k-modes$）
K中心点（$k-medoids$）
K分布（$k-distributions$）
K均值聚类算法的理解

第八章 最近邻学习

基础知识

分类都包含两个阶段：训练阶段和工作阶段

积极学习（eager learning）

- 有显示的训练过程，在训练阶段就对训练样本进行学习处理构建起分类模型

消极学习（lazy learning）

- 没有显示训练过程，只是在训练阶段将训练样本保存起来，建模工作延迟到工作阶段才处理

最近邻学习：**不是在整个样本空间上一次性估计目标函数，而是针对每个待测样本作出局部目标函数逼近。**可以为不同的待测样本构建起不同的目标函数逼近，相比于那些积极的学习技术，最近邻学习往往具有较好的分类性能。

基本思想

给定待测样本，首先基于某种近邻索引方法找出训练集中与其最靠近的K个样本，然后基于这K个样本的后验概率来预测 待测样本的类标记。

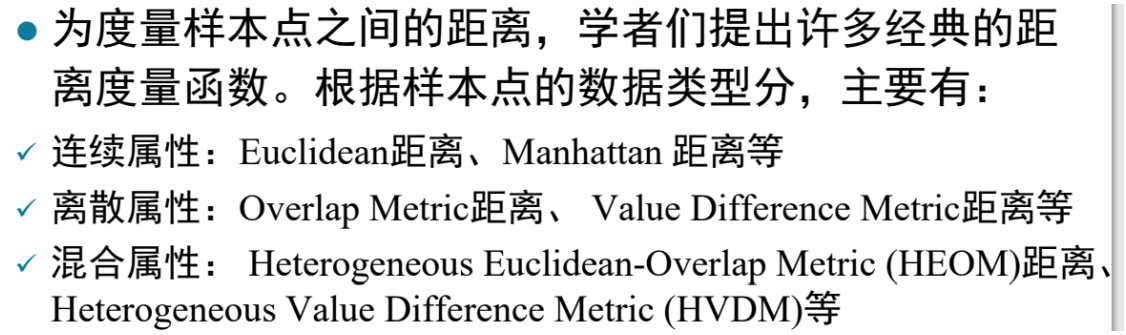
具体算法分为两个阶段：

- 训练阶段：将每个训练样本保存起来
- 工作阶段：
    - 给定一个待测样本
        - 基于某种近邻索引方法找出训练样本集中与其最靠近的K个样本
        - 基于这K个样本的后验概率来预测待测样本的类标记

常见问题

1.近邻索引问题

- 最近邻学习的所有计算几乎都花费在索引近邻问题上。使用最多的近邻索引方法就是通过计算待测样本与每一个训练样本之间的距离，然后基于距离 排序，选择距离最短的K个训练样本作为待测样本 的最近邻样本。
-



- > 除了上述基于距离排序的索引方法之外，目前还开 发了许多对存储的训练样本进行索引的方法，以便 更快速地确定最近邻样本。比如KD-Tree方法把训 练样本存储在树的叶子结点上，邻近的样本存储在 相同或相近的叶子结点上，然后通过测试待测样本 在内部结点上的划分属性把待测样本划分到相关的 叶子结点上。

2.维度灾害问题

- 由于存在很多不相关属性所导致的难题,解决方法为：
  - 1）属性加权
  - 2）属性选择

3.邻域大小问题

- 最近邻学习有一个很重要的参数，那就是邻域的大小，即最近邻样本的数目K，最近邻学习的预测结 果与K的大小密切相关。同样的数据，K值不同可能导致不同的预测结果。
- 解决方法：1）基于经验直接给定；2）基于数据自动学习

4.后验概率问题

- 给定待测样本的K个最近邻样本，估计其后验概率的常用方 法包括：投票法、加权投票法、局部概率模型法。

5.计算效率问题

- 最近邻学习推迟所有的计算处理，直到接收到一个 新的待测样本，所以分类每个新的待测样本就需要 大量的计算。
- 高效的近邻索引方法可以在一定程度上缓解计算效 率问题，比如KD-Tree近邻索引方法。

6.归纳偏置问题

- 在输入空间上相近的样 本点具有相似的目标函数输出。
- 有效的距离度量方法可以在一定程度上缓解归纳偏 置问题，比如属性加权的距离度量方法。

第九章 无监督学习

基础知识

聚类与分类的区别在于：**训练样本的类标记是未知的**

聚类：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。

聚类算法分为以下五类：

1)基于划分的方法

- 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。
- 满足两个条件
  - 每个分组至少包含一个对象；
  - 每个对象属于且仅属于某一个分组
- 主要包括：K均值聚类算法及其变种

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

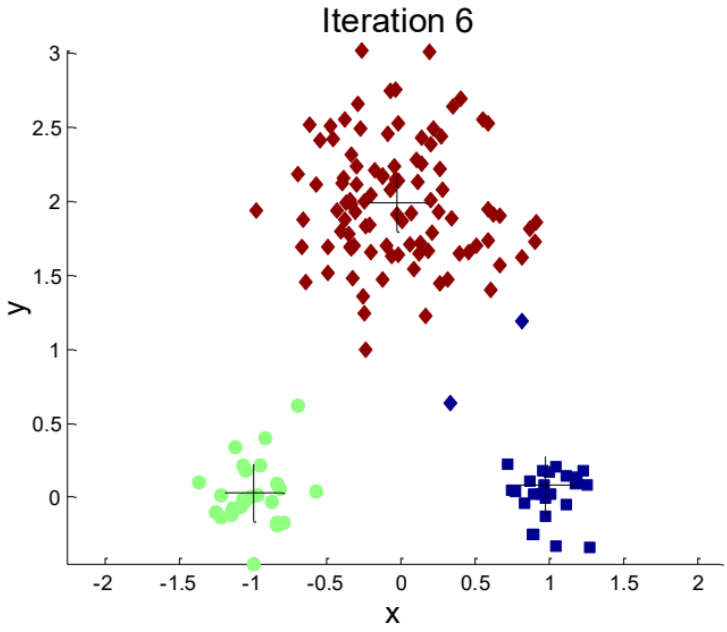
K均值聚类算法（$k-means$）

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，将对象划分到最近的簇中
3. 更新每个簇的质心
4. 重复2-3步骤直到簇中对象不改变

> 演示
>
> 
>

K均值算法的说明：

- 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

K众数算法（$k-modes$）

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。

- 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。、
- 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。

K中心点（$k-medoids$）

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。

K分布（$k-distributions$）

K分布算法的设计动机:避免计算

- 1）簇质心的计算问题； 2）对象到质心距离的计算问题。

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。

输入：簇的数目K；包含N个对象的数据D

输出：K个簇的集合

方法：

1. 将D随机划分成K个非空且不互相交的簇
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

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**

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。