Lecture12 集成学习&聚类

1. 集成学习 Ensemble

多数投票 Majority Voting

- 如果分类的标签 y_i 是离散的（可分类的）$y_i \in Y$（例如 $Y = \{-1, +1\}, Y = \{0, 1\}$）
- 假设我们用抛硬币的方法分类，分错的概率是 0.5
- 假设我们拥有 m 个独立的分类器，比随机的效果要好一些，也就是说 $error = 0.5 - \epsilon$
- 如果我们通过 m 个独立的分类器（这些分类器比随机分类好一点）的分类结果进行多数投票呢?

效果肯定是 m 越多，分类效果越好

Condorcet's 陪审团理念

假设

- 每一个个体有 p 的概率投出正确的票
- 每个投票人是独立的

在二分类中，如果 $p > 0.5$，那么增加陪审团的成员能增加投票的决定是正确的的概率，如果 $p < 0.5$，那么增加陪审团的成员会降低投票的决定是正确的的概率

2. 集成学习方法

- 一种集合方法结合了许多单一分类器通过多数投票投出的结果
- 这样的单一的分类器，也叫做**弱学习者 weak learner**，要求比随机（例如在二分类中随机的概率 $p = 0.5$）要稍微好一些

问题引入

陪审团理念要求投票人（也就是分类器）尽可能是**独立**的，但是因为我们训练的数据集只有一个，也就是**所有的模型都只能在同一数据集训练**，训练的结果有很大的**相关性**，在这种情况下，每个训练出的分类器不是完全独立的，我们需要采取一些措施使得训练的分类器尽可能独立

Bagging

训练

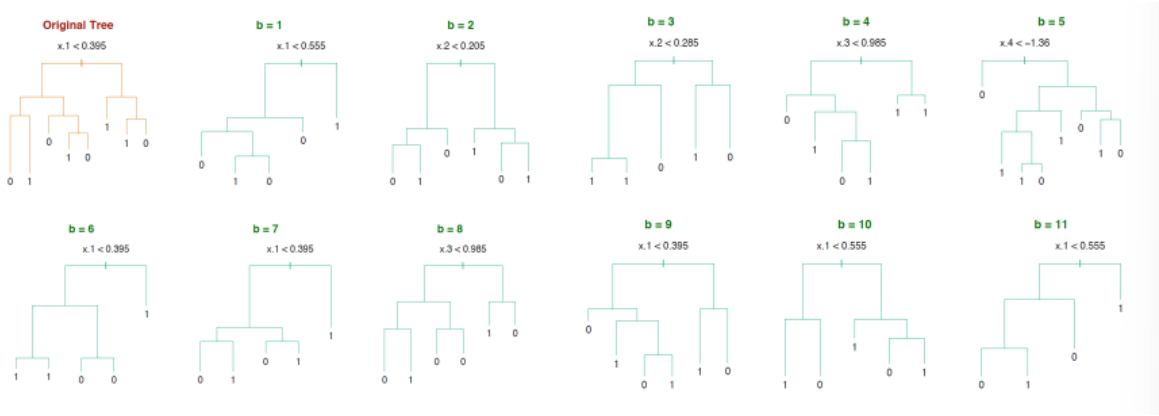
- 假设原始数据集的大小为 L，且我们需要训练 B 个 weak learners
- 对于每个 weak learner，在原始数据集中进行随机有放回采样，构成它的数据集 B_b
- 使用数据集 B_b 对特定的 weak learner b 进行训练

分类

- 采用多数投票
- $f_{avg} := \frac{1}{B} \sum_{b=1}^{B} f_b(x)$

随机森林

- 使用 Bagging 的方法构建一系列决策树分类器
- 对于随机森林来说，不像决策树在所有的特征中做节点分裂，随机森林只选取特征的一个子集做分裂条件
- 一般而言，假设特征向量的维度为 d，随机森林的选择的子集大小为 $\sqrt{d}$



Boosting

- 是一种非常强大的机器学习方法之一
- 核心思想：在**带权重**的样本训练集中训练 weak learners

决策树桩 Decision Stumps

决策树桩是一种一般只有两层的二叉树样子的决策树，它是一个非常弱的分类器

- 决策树桩通常是通过暴力计算的的：将训练集中的实数从最小到最大离散，枚举所有可能的分类器，然后选择训练误差最小的那个

在 Boost 集成学习中，weak learners 通常可以是一堆决策树桩分类器

介绍

假设有 M 个 weak learners，每一个 weak learner $G_m, m \in \{1, \dots, M\}$ 有一个投票权重 α_m

在这些 weak learners 中，分类效果好的分类器的权重应该比效果差的要高

投票分类结果

$$G(x) = sign[\sum_{m=1}^{M} \alpha_m G_m(x)]$$

对于每一个 weak learner，它的错误率

$$err_m = \frac{\sum_{i=1}^{n} w_i 1\{y_i \neq G_m(x_i)\}}{\sum_{i=1}^{n} w_i}$$

- w_i ：训练集样本 i 的权重

- 越容易预测错的样本，权重越大
- 越容易预测对的样本，权重越小
- 这样，如果分类器能预测对权重大的样本，它的错误率越少

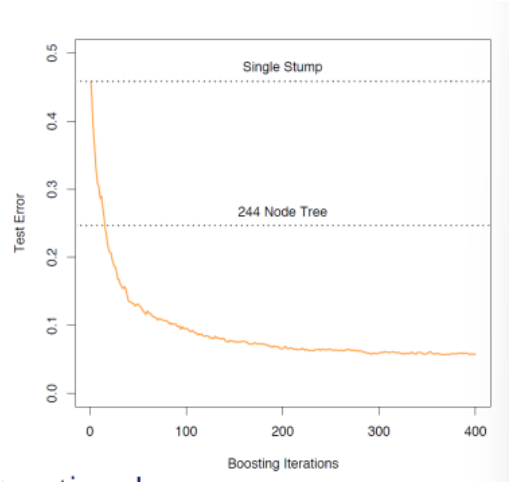
对于每个 weak learner，它的投票权重

$$\alpha_m = \frac{1}{2} log(\frac{1 - err_m}{err_m})$$

AdaBoost 算法

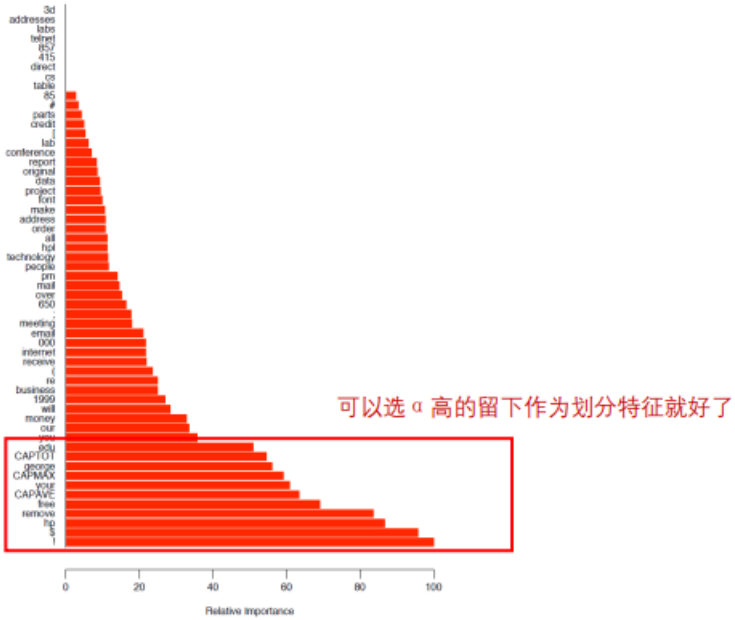
1. 初始化所有的样本权重 $w_i = \frac{1}{n}$， n 为样本个数
2. for $m = 1$ to M（M 为 weak learners 的个数）
3. 使用带权的样本来训练一个 weak learner $G_m(x)$
4. 计算 weak learner 的错误率 $err := \frac{\sum_{m=1}^{M} 1\{y_i \neq G_m(x_i)\}}{M}$
5. 计算 weak learner 的投票权重 $\alpha_m = \frac{1}{2} log(\frac{1-err_m}{err_m})$
6. 通过分类效果，更新每个样本的权重 w_i，满足 $w_i \leftarrow w_i \times e^{-\alpha_m y_i G_m(x_i)}$
7. 输出 $G(x) = sign[\sum_{m=1}^{M} \alpha_m G_m(x)]$

AdaBoost 效果



- 假设是一个二分类问题
- 随机的错误率为 50%
- 决策树桩的错误率为 45.8%
- 如果建立一个决策树，它的错误率为 24.7%
- 如果建立一群决策树桩作为 weak learners 进行集成学习，错误率在 400 次迭代后仅为 5.8%

不仅如此，使用 AdaBoost 对一群决策树桩的 weak learners 进行集成学习，这些弱分类器的投票参数 α_m 还可以作为一种**特征选择**的参数，表示按照某个节点 split 的决策树桩的效果更好



3. 聚类的介绍 Clustering

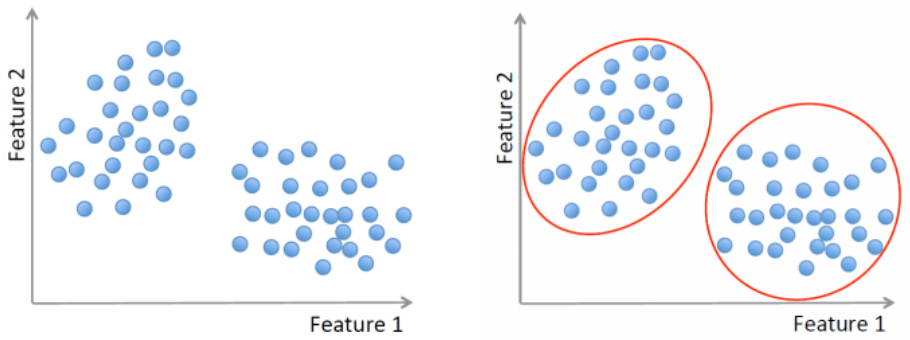
非监督学习

训练数据：样本 x，没有标签 y

$$x_1, \ldots, x_n, x_i \in X \subset \mathbb{R}^d$$

- x_i：样本
- X：样本数据集
- $\mathbb{R}^d$：d 维的实数集（特征）

聚类的图示



$$f : \mathbb{R}^d \to \{C_1, \ldots C_k\}$$

- f：聚类函数 $f(x_i)$
- C_i：聚类结果

聚类的示例

- 按人口统计数据对人口进行聚类
- 地理对象的聚类（矿藏、房屋等）
- 恒星的聚类
- 音频信号分离
- 图像分割

相似性的定义

- 在聚类中，对于**相似性 similarity** 的定义非常的关键
- 一般我们认为相似性和距离成反比

距离 distance

几种常见的衡量距离的方法

- 欧氏距离： $d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||^2$
- 曼哈顿距离： $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} |\mathbf{x}_i - \mathbf{y}_i|$
- Kernelized distance: $d(\mathbf{x}, \tilde{\mathbf{x}}) = \|\phi(\mathbf{x}) - \phi(\tilde{\mathbf{x}})\|$

问题

相似性定义是主观的，有时候很难去辨别

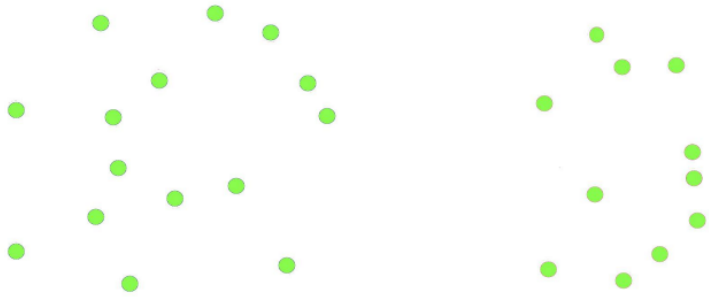
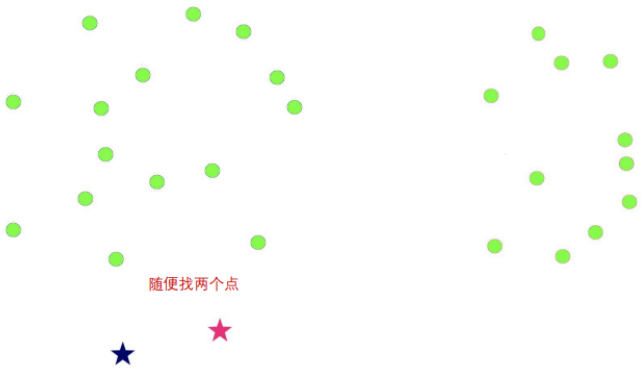
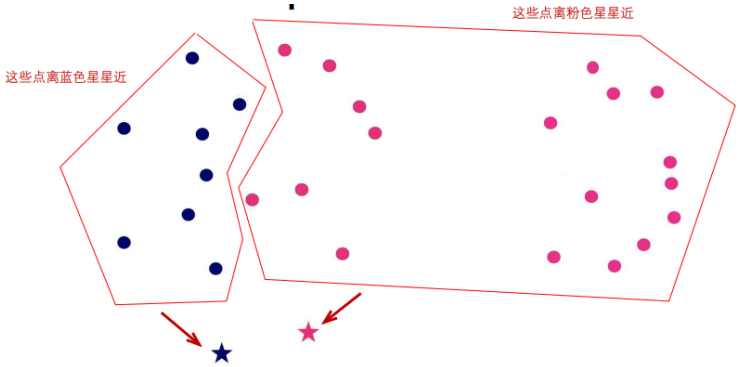
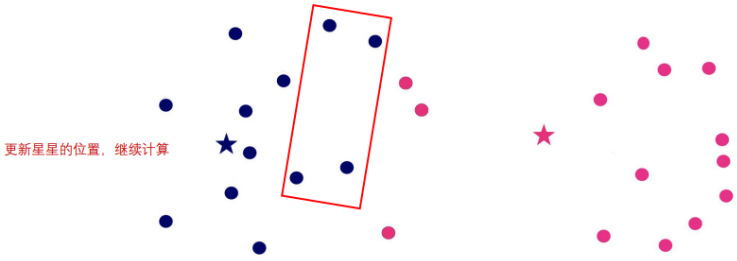
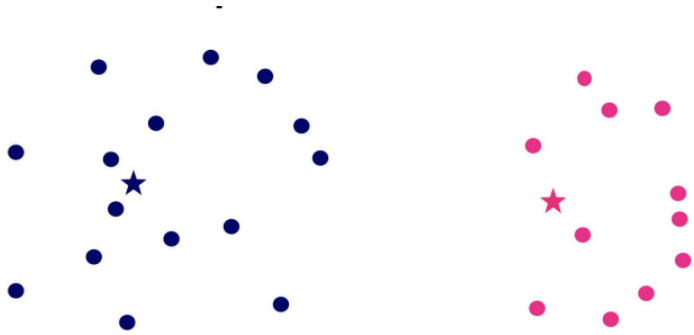
- 不同的相似性标准可能导致不同的聚类效果

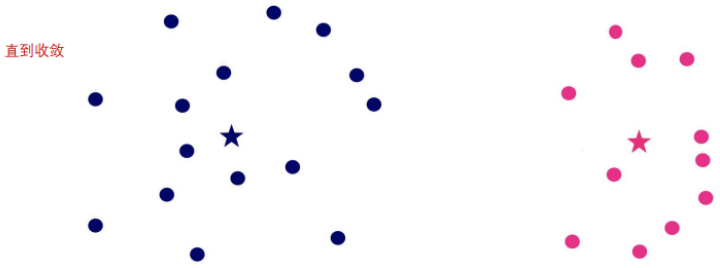
4. K-Means 聚类

示例

首先，选取聚类的类数 k

- 这里我们假设 $k = 2$

图示	解释
	决定聚类个数 $k=2$
	随机两个聚类点的位置（蓝色星星与粉色星星）
	找到离蓝色星星最近的一群点和离粉色星星最近的一群点分别划分为蓝色类群和粉色类群
	使用类群的点做平均，更新蓝色星星的位置
	继续计算 找到离蓝色星星最近的一群点和离粉色星星最近的一群点分别划分为蓝色类群和粉色类群

图示	解释
...	...
直到收敛 	重复计算，直到收敛（蓝色星星的位置和粉色星星的位置不再变动） 聚类完成

算法

1. 设置聚类的簇数 k
2. 初始化随机指定类群的簇节点 $\mu_1, \ldots, \mu_k$
3. 重复直到收敛（意味着所有簇节点中没有变化或达到最大迭代次数）
4. 　　将每个样本点 x_i 分配到最近的簇节点 u_i 的类群中
5. 　　更新每个簇的簇节点 $\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$

分析

- 目标：将每个样本 $(x_1, \ldots, x_n)$ 分配给 k 个类群中 $\{C_1, \ldots, C_k\}$
  - μ_j 意味着所有在 C_j 类群中的样本的均值
- 最小化：$J = \sum_{j=1}^{k} \sum_{x_i \in C_j} d(x_i, \mu_j)$
- K-Means 目标的精确优化是 NP-Hard 的
- K-means算法是一种收敛于局部最优的启发式算法

优缺点

优点

- 很容易实现

缺点

- k 是人为给定的
- 如果样本的特征维度很高，K-Means 的聚类效果很差
- 没有理论基础

与 K-Means 相关的问题

如何选取一个合适的 k

G-Means 算法

1. 将 k 初始化成一个较小的数字
2. 用 K-Means 算法进行聚类，然后存储这些簇节点的值
3. 将每个样本点分配到它最近的簇节点所在的类群
4. 判断每个样本点所在的类群是否满足高斯分布（Anderson-Darling 测试）

5. 对于每个类群，如果如果这些点看起来是正态分布的，则保持簇节点为中心，否则，将其替换为两个簇节点
6. 重复第 2 步，直到没有生成更多的簇节点

如何评价聚类模型的好坏

- 内部评价：使用相同的数据，高聚类内相似度（聚类内的文档相似）和低聚类间相似度
    - 例如，考虑了星团内部距离和星团之间距离的 davisesbouldin 指数，指数的值越低，不同簇之间的距离就越宽，每个簇内的点位于一起的距离就越紧密
- 外部评价：利用外部数据的评价
    - 例如：交互信息、熵、adjusted rand index 等

如何对非圆形形状分布的的样本进行聚类

还有其他处理其他形状的方法：光谱聚类、DBSCAN、BIRCH 等

如何选择一个比较好的初始化簇节点的位置

- K-Means 对初始化节点的位置比较敏感
- 多尝试几次
- 启发式的算法：初始化的簇节点距离相对远一些