

数据挖掘与应用

分类

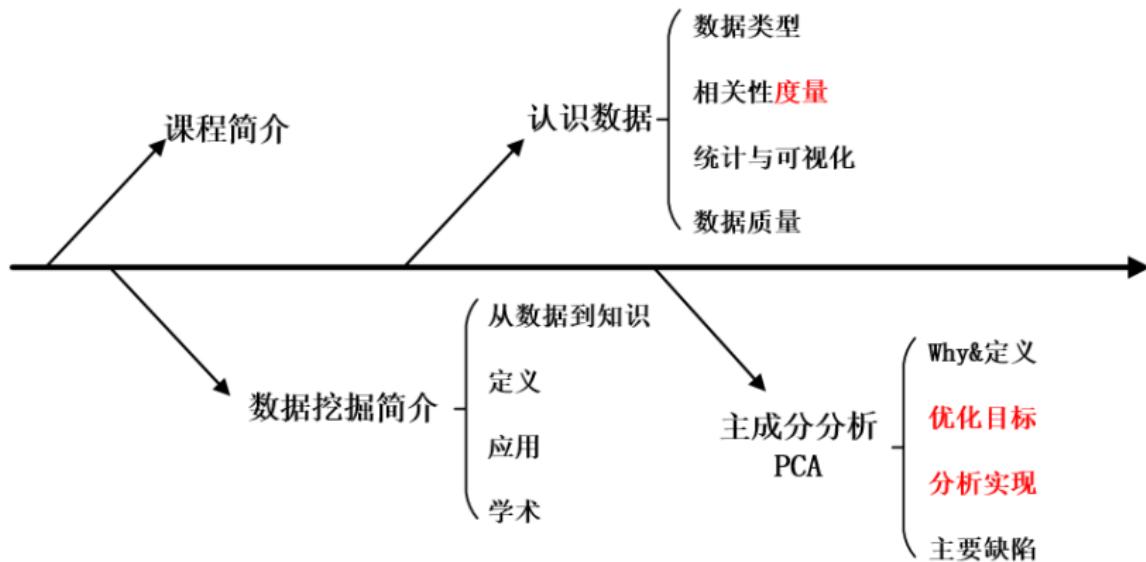
授课教师：周晟

浙江大学 软件学院

2022.09



上节课回顾



课程内容

① 认识分类

- 分类的定义
- 分类的基本步骤

② 分类模型的评估

- 二分类模型的评估
- 多分类模型的评估
- ROC 曲线

③ 决策树

- 决策树原理
- 决策树的选择

④ 线性判别分析

- 从 PCA 到 LDA
- 线性判别分析 LDA
- 多分类 LDA

⑤ 从分类到回归

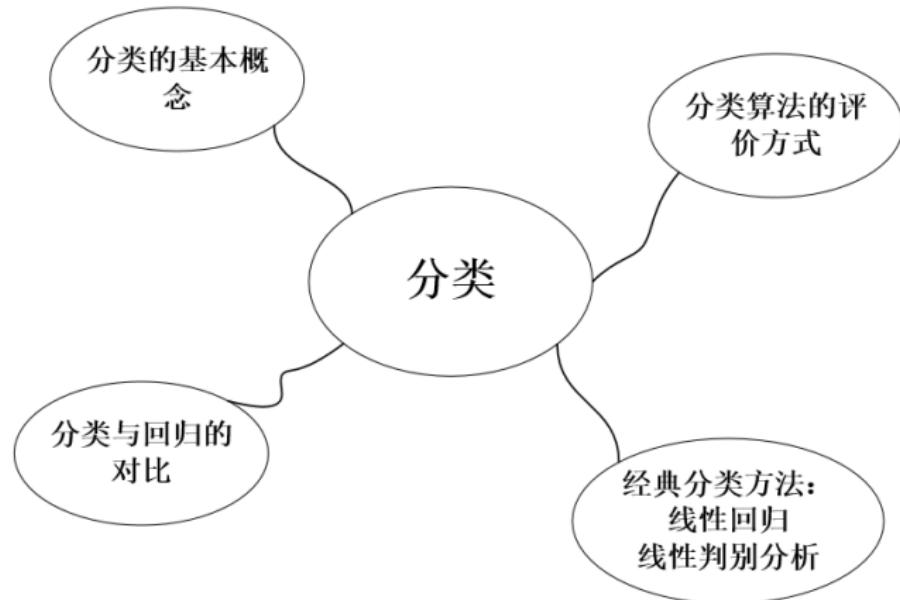
- 线性回归

⑥ 集成学习

- 集成学习概述
- Bagging, Stacking
- Boosting



本节课程结构



① 认识分类

- 分类的定义
- 分类的基本步骤

② 分类模型的评估

- 二分类模型的评估
- 多分类模型的评估
- ROC 曲线

③ 决策树

- 决策树原理
- 决策树的选择

④ 线性判别分析

- 从 PCA 到 LDA
- 线性判别分析 LDA
- 多分类 LDA

⑤ 从分类到回归

- 线性回归

⑥ 集成学习

- 集成学习概述
- Bagging, Stacking
- Boosting



认识分类

分类是自然界最常见的数据挖掘任务之一：

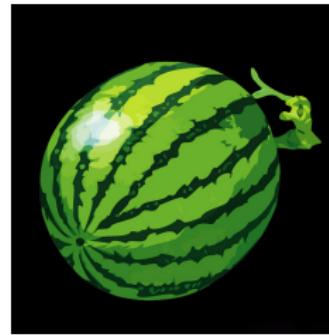
- ① 问卷调查
- ② 性别统计
- ③ 电影标签
- ④ 商品类目
- ⑤ 定罪定责
- ⑥ ...



分类 VS 聚类

分类——有监督

- 西瓜好坏
- 核酸检测
- 垃圾邮件判别
- 界门纲目科属种



聚类——无监督

- 消费者群体聚类
- 产品定位
- 离群点检测



分类的基本步骤



周杰伦



陈赫



林丹



验证集



训练集



测试集



这是周杰伦 / 与其他两人不同

1 认识分类

- 分类的定义
- 分类的基本步骤

2 分类模型的评估

- 二分类模型的评估
- 多分类模型的评估
- ROC 曲线

3 决策树

- 决策树原理
- 决策树的选择

4 线性判别分析

- 从 PCA 到 LDA
- 线性判别分析 LDA
- 多分类 LDA

5 从分类到回归

- 线性回归

6 集成学习

- 集成学习概述
- Bagging, Stacking
- Boosting



二分类模型的性能评估

分类结果的基本术语：

- ① 真正例/真阳性 (True Positive, TP): 正确分类的正样本个数。
- ② 真负例/真阴性 (True Negative, TN): 正确分类的负样本个数。
- ③ 假正例/假阳性 (False Positive, FP): 错误分类的负样本个数。
- ④ 假负例/假阴性 (False Negative, FN): 错误分类的正样本个数。

		预测的类		
		是	否	合计
实际的类	是	TP	FN	P
	否	FP	TN	N
	合计	P'	N'	P+N

二分类的混淆矩阵 (confusion matrix)



准确率与错误率

分类器的准确率 (accuracy) 是指被正确分类的样本所占的比例:

$$\text{accuracy} = \frac{TP + TN}{P + N}$$

分类器的错误率 (error rate) 是指被错误分类的样本所占的比例:

$$\text{error rate} = \frac{FP + FN}{P + N}$$

		预测的类		
		猫	狗	合计
实际的类	猫	50	10	60
	狗	6	54	60
	合计	56	64	120

$$\text{accuracy} = \frac{TP + TN}{P + N} = \frac{50 + 54}{60 + 60} = 0.867$$

$$\text{error rate} = \frac{FP + FN}{P + N} = \frac{10 + 6}{60 + 60} = 0.133$$

以猫狗分类为例



灵敏性和特效性

灵敏性 (sensitivity) 和特效性 (specificity) 是处理**类别不均衡**的分类问题的常用指标。

$$\text{sensitivity} = \frac{TP}{P}$$

$$\text{specificity} = \frac{TN}{N}$$

		预测的类		
		阳性	阴性	合计
实际的类	阳性	19	1	20
	阴性	2	9998	10000
	合计	21	9999	10020

$$\text{sensitivity} = \frac{TP}{P} = \frac{19}{20} = 0.95$$

$$\text{specificity} = \frac{TN}{N} = \frac{9998}{10000} = 0.9998$$

以核酸检测为例（数据为虚构）



精度和召回率

精度 (precision) 是指预测为正样本的样本中实际为正样本的比例

$$\text{precision} = \frac{TP}{TP + FP}$$

召回率 (recall) 是指实际为正样本的样本中被正确预测为正样本的比例

$$\text{recall} = \frac{TP}{TP + FN} = \frac{TP}{P}$$

		预测的类		
		刷单	正常	合计
实际的类	刷单	365	5	370
	正常	23	687	710
	合计	388	692	1080

$$\text{precision} = \frac{TP}{TP + FP} = \frac{365}{365 + 23} = 0.941$$

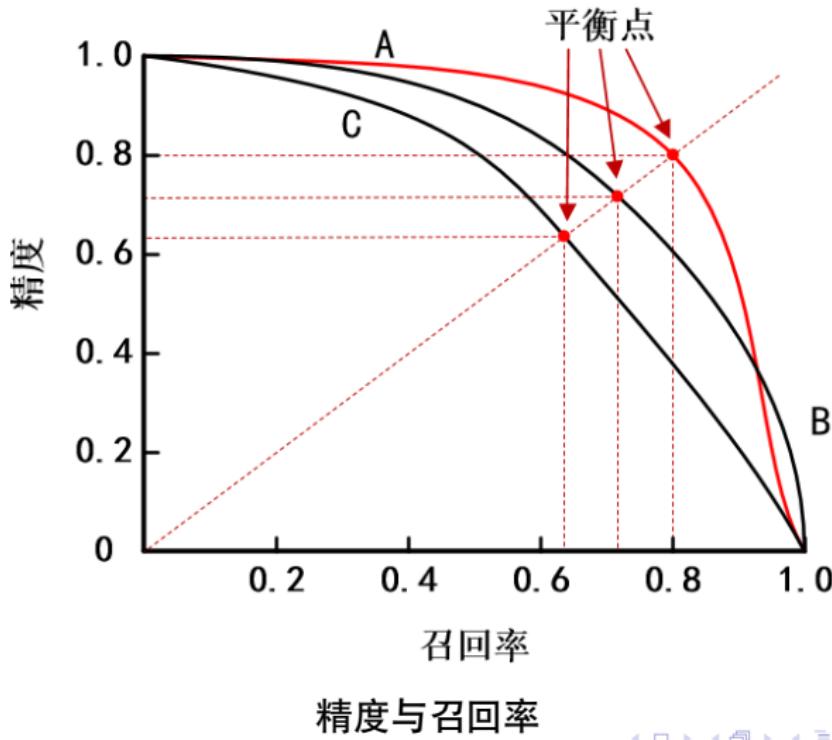
$$\text{recall} = \frac{TP}{TP + FN} = \frac{365}{365 + 5} = 0.986$$

以刷单检测为例



精度和召回率

精度和召回率是相互制衡的一组指标，提升一个往往会降低另一个



精度和召回率

判案低精度的后果：六月飞雪窦娥冤



低精度 & 高召回的案例



精度与召回率的融合：F 度量

F 度量 (F-score) 是一种融合了精度和召回率的统一度量。

$$F-score = \frac{2 \times precision \times recall}{precision + recall}$$

对 precision 和 recall 有偏好的可以使用 f_β 度量：

$$F_\beta-score = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

当有些情况下，我们认为精确率更重要些，那就调整 β 的值小于 1，如果我们认为召回率更重要些，那就调整 β 的值大于 1。

精度与召回率的融合：F 度量

		预测的类		
		刷单	正常	合计
实际的类	刷单	365	5	370
	正常	23	687	710
	合计	388	692	1080

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{365}{365 + 23} = 0.941$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{365}{365 + 5} = 0.986$$

以刷单检测为例

- ① $\beta = 0.5$: $F_\beta - score = \frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall} = 0.949$
 - ② $\beta = 1$: $F_\beta - score = \frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall} = 0.962$
 - ③ $\beta = 2$: $F_\beta - score = \frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall} = 0.976$

β 设置多少合适？



多分类 F 度量

传统的 F 度量仅用于二分类情况，多分类则通常使用 Micro-F1 或 Macro-F1 度量。

第 i 类的精度和召回率可以表示为：

$$\text{precision}_i = \frac{TP_i}{TP_i + FP_i}$$

$$\text{recall}_i = \frac{TP_i}{TP_i + FN_i}$$

Micro-F1 先计算出所有类别的总的 Precision 和 Recall：

$$\text{Precision}_{\text{micro}} = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FP_i}$$

$$\text{Recall}_{\text{micro}} = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FN_i}$$



多分类 F 度量

然后利用 F1 计算公式计算出来的 F1 值即为 Micro-F1：

$$F1_{\text{micro}} = 2 \cdot \frac{\text{Precision}_{\text{micro}} \cdot \text{Recall}_{\text{micro}}}{\text{Precision}_{\text{micro}} + \text{Recall}_{\text{micro}}}$$

Macro-F1 则先对各类别的 Precision 和 Recall 求平均：

$$\text{Precision}_{\text{macro}} = \frac{\sum_{i=1}^n \text{Precision}_i}{n}$$

$$\text{Recall}_{\text{macro}} = \frac{\sum_{i=1}^n \text{Recall}_i}{n}$$

然后再利用 F1 计算公式计算出来的 F1 值即为 Macro-F1。



多分类 F 度量

		预测的类		
		红灯	绿灯	黄灯
实际的类	红灯	39	0	1
	绿灯	1	33	5
	黄灯	3	4	10
	合计	43	37	16
		合计		

Micro-F1:

$$\text{Precision}_{\text{micro}} = \frac{\text{TP}_{\text{红}} + \text{TP}_{\text{绿}} + \text{TP}_{\text{黄}}}{\text{TP}_{\text{红}} + \text{TP}_{\text{绿}} + \text{TP}_{\text{黄}} + \text{FP}_{\text{红}} + \text{FP}_{\text{绿}} + \text{FP}_{\text{黄}}} \\ = \frac{39 + 33 + 10}{39 + 33 + 10 + 4 + 4 + 6} = 0.854$$

$$\text{Recall}_{\text{micro}} = \frac{\text{TP}_{\text{红}} + \text{TP}_{\text{绿}} + \text{TP}_{\text{黄}}}{\text{TP}_{\text{红}} + \text{TP}_{\text{绿}} + \text{TP}_{\text{黄}} + \text{FN}_{\text{红}} + \text{FN}_{\text{绿}} + \text{FN}_{\text{黄}}} \\ = \frac{39 + 33 + 10}{39 + 33 + 10 + 1 + 6 + 7} = 0.854$$

$$\text{F1}_{\text{micro}} = 2 \cdot \frac{\text{Precision}_{\text{micro}} \cdot \text{Recall}_{\text{micro}}}{\text{Precision}_{\text{micro}} + \text{Recall}_{\text{micro}}} \\ = 2 \cdot \frac{0.854 * 0.854}{0.854 + 0.854} = 0.854$$

Macro-F1:

$$\text{precision}_{\text{红}} = \frac{\text{TP}_{\text{红}}}{\text{TP}_{\text{红}} + \text{FP}_{\text{红}}} = \frac{39}{39 + 4} = 0.907$$

$$\text{recall}_{\text{红}} = \frac{\text{TP}_{\text{红}}}{\text{TP}_{\text{红}} + \text{FN}_{\text{红}}} = \frac{39}{39 + 1} = 0.975$$

$$\text{precision}_{\text{绿}} = 0.892 \quad \text{recall}_{\text{绿}} = 0.846$$

$$\text{precision}_{\text{黄}} = 0.625 \quad \text{recall}_{\text{黄}} = 0.588$$

$$\text{Precision}_{\text{macro}} = \frac{\text{precision}_{\text{红}} + \text{precision}_{\text{绿}} + \text{precision}_{\text{黄}}}{3} \\ = \frac{0.907 + 0.892 + 0.625}{3} = 0.808$$

$$\text{Recall}_{\text{macro}} = \frac{\text{recall}_{\text{红}} + \text{recall}_{\text{绿}} + \text{recall}_{\text{黄}}}{3} \\ = \frac{0.975 + 0.846 + 0.588}{3} = 0.803$$

$$\text{F1}_{\text{macro}} = 2 \cdot \frac{\text{Precision}_{\text{macro}} \cdot \text{Recall}_{\text{macro}}}{\text{Precision}_{\text{macro}} + \text{Recall}_{\text{macro}}} \\ = 2 \cdot \frac{0.808 * 0.803}{0.808 + 0.803} = 0.805$$

ROC 曲线

接收机工作特征曲线 (Receiver Operating Characteristic curve, ROC)



ROC 曲线最早用于英国雷达分辨鸟或德国飞机的概率。



ROC 曲线

飞机与鸟的误判

当时的雷达技术还没有那么先进，存在很多噪声（比如一只大鸟飞过）有的雷达兵比较谨慎，凡是有信号过来，他都会倾向于解析成是敌军轰炸机；而有的雷达兵又比较克制，会倾向于解析成是飞鸟。



急需一套评估指标来帮助他汇总每一个雷达兵的预测信息，以及来评估这台雷达的可靠性。



ROC 曲线

ROC 曲线的定义

ROC 曲线是在二维空间中对分类模型的效率进行度量。它将伪阳性率 (False Positive Rate, FPR) 定义为 X 轴，真阳性率 (True Positive Rate, TPR) 定义为 Y 轴，将不同阈值对应的分类模型计算 (FPR, TPR) 绘制得到。

TPR：在所有实际为阳性的样本中，被正确地判断为阳性的概率，即灵敏性 (sensitivity)。

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{P} = \text{sensitivity}$$

FPR：在所有实际为阴性的样本中，被错误地判断为阳性的概率，即 1-特异性 (specificity)。

$$FPR = \frac{FP}{FP + TN} = 1 - \frac{TN}{N} = 1 - \text{specificity}$$

AUC 指标

AUC 指标

AUC 指标 (Area under the Curve of ROC) 是定义在 ROC 曲线下的一种评价分类模型质量的指标。AUC 为 ROC 曲线下方的面积，取值范围在 [0,1]。值越大，说明模型的性能越好。

AUC 通常适用于只输出分类概率而无法输出分类标签的情况：

- ① Link Prediction
- ② Anomaly Detection
- ③ ...



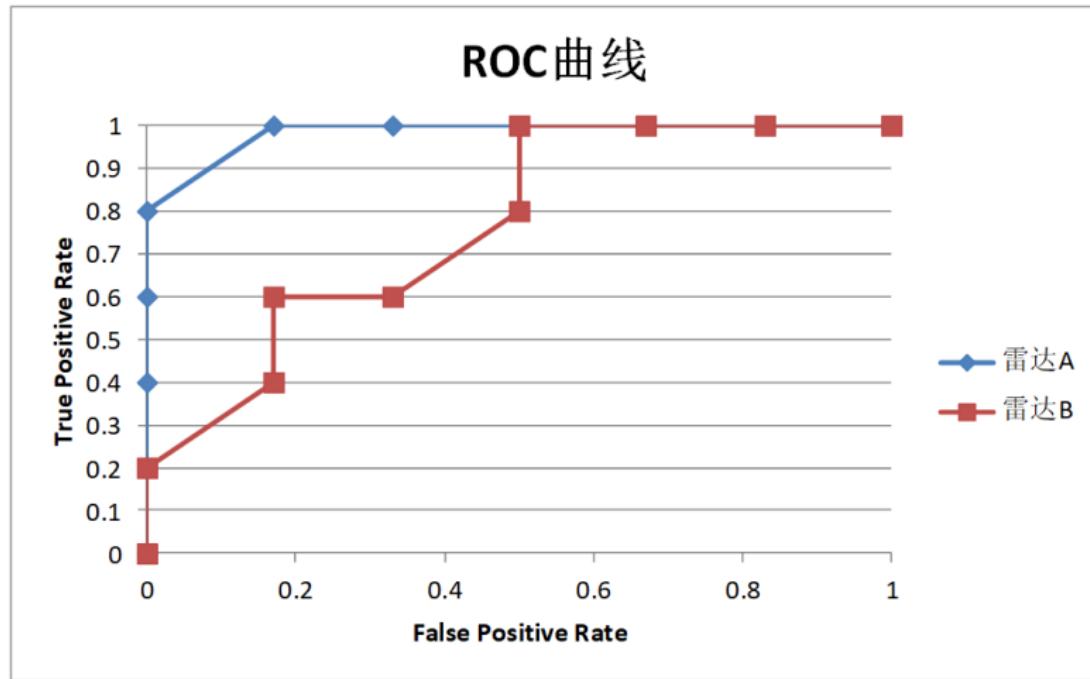
ROC 分析——以雷达检测为例

		雷达辨别概率	
不明物体	实际类别	分辨为飞机概率	
		雷达A	雷达B
1	飞机	0.9	0.75
2	鸟	0.5	0.55
3	飞机	0.75	0.65
4	鸟	0.4	0.45
5	鸟	0.1	0.3
6	飞机	0.98	0.8
7	飞机	0.6	0.5
8	鸟	0.6	0.75
9	鸟	0.3	0.4
10	鸟	0.55	0.6
11	飞机	0.7	0.55

雷达兵	判断阈值	评价指标					
		TP	FP	FN	TN	FPR	TPR
1	0.1	5	6	0	0	1	1
2	0.3	5	5	0	1	0.83	1
3	0.4	5	4	0	2	0.67	1
4	0.5	5	3	0	3	0.5	1
5	0.55	5	2	0	4	0.33	1
6	0.6	5	1	0	5	0.17	1
7	0.7	4	0	1	6	0	0.8
8	0.75	3	0	2	6	0	0.6
9	0.9	2	0	3	6	0	0.4
10	0.95	1	0	4	6	0	0.2
11	1	0	0	5	6	0	0

雷达兵	判断阈值	评价指标					
		TP	FP	FN	TN	FPR	TPR
1	0.3	5	6	0	0	1	1
2	0.4	5	5	0	1	0.83	1
3	0.45	5	4	0	2	0.67	1
4	0.5	5	3	0	3	0.5	1
5	0.55	4	3	1	3	0.5	0.8
6	0.6	3	2	2	4	0.33	0.6
7	0.65	3	1	2	5	0.17	0.6
8	0.75	2	1	3	5	0.17	0.4
9	0.8	1	0	4	6	0	0.2
10	0.9	0	0	5	6	0	0

ROC 分析——以雷达检测为例



① 认识分类

- 分类的定义
- 分类的基本步骤

② 分类模型的评估

- 二分类模型的评估
- 多分类模型的评估
- ROC 曲线

③ 决策树

- 决策树原理
- 决策树的选择

④ 线性判别分析

- 从 PCA 到 LDA
- 线性判别分析 LDA
- 多分类 LDA

⑤ 从分类到回归

- 线性回归

⑥ 集成学习

- 集成学习概述
- Bagging, Stacking
- Boosting



规则学习

规则学习

规则学习 (Rule Learning) 是从训练数据中学习出一组能用于对未见样本进行分类的规则。其通用形式为：

$$\oplus \leftarrow f_1 \wedge f_2 \wedge \cdots \wedge f_L$$

规则学习的主要优势是：

- ① 可解释性强
- ② 可以在学习的过程中自然地引入人为先验和领域知识。



决策树案例

筛选 -

屏幕尺寸

13.0 英寸以下 13.0-13.9 英寸 14.0-14.9 英寸

处理器

Intel i3 AMD 龙 intel i5

内存容量

4GB 6GB 8GB
24GB 12GB 16GB
20GB 32GB 36GB
64GB 40GB 128GB

显卡型号

MX230 RTX3070Ti RTX 3070Ti
RTX 3080Ti INTEL IRIS XE RTX 2050
RTX A3000 Radeon 625 RX 6700S
RX 6800S MX550 MX570
RTX A1000 ARC A370M Apple M2 集成...
RX 6650M TS50 ARC 730M
RX 6850M XT ARC 370M RX 6600S
MX250 RTX A4000 MX330
MX450 GTX 1050 P620

重置 确定 (98 万+件商品)

决策树案例

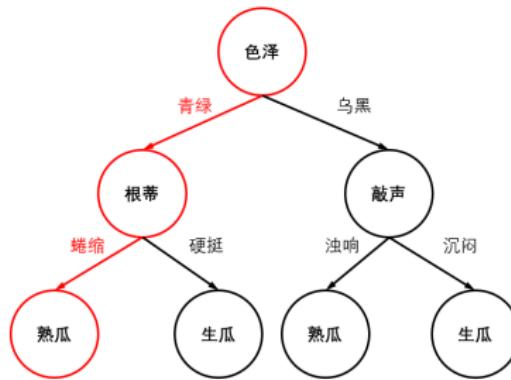


编号	色泽	根蒂	敲声	熟瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否
5



决策树

瓜是否成熟，可通过不同提问者对要猜的事物提问，回答者只能回答是或否。每次是或否的选择都将目前候选的事物根据某种规则分为两半，进而构建一棵以提问次数为深度的树从根节点到叶子节点的一条路径。



决策树的定义

决策树

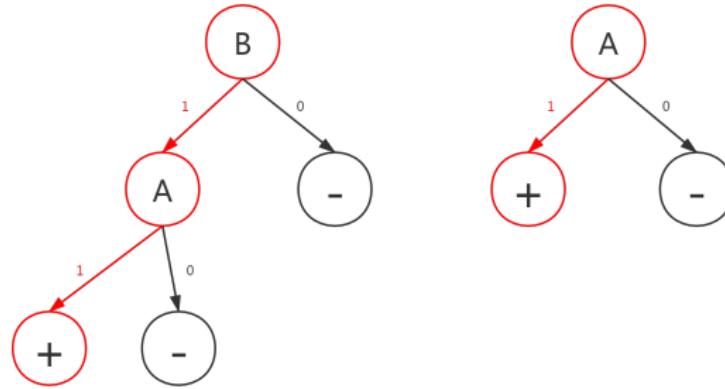
决策树是一种模拟人类决策过程的常用分类方法，通过递归选择属性维度将样本分配到对应的分支中，直到每个分支能尽可能把所有数据分开。

- ① 决策树的输入为样本的特征向量，输出为离散的类别属性。（输出连续值的模型称为回归树）
- ② 一般而言决策树的**非叶子节点**表示一次对样本特征的**测试**，根据测试结果划分子树；**叶子节点**代表样本所属的**类别（标签）**。
- ③ 决策树善于处理样本数量较大的分类问题，但无法应对过多的特征维度。

决策树的学习

样本数量	属性 A	属性 B	标签
50	A=0	B=0	-
50	A=0	B=1	-
0	A=1	B=0	-
100	A=1	B=1	+

显然，决策树有两棵，如果先在 B 属性上分，这棵树深度为 2：



决策树的学习阶段-递归伪代码

Algorithm

```
Build_DcisionTree(Examples,Attributes):
    if 所有样本标签都为y:
        return 叶节点 with 标签y
    else:
        if 属性为空:
            return 叶节点 with 占大多数的标签
        else:
            选择一个 Attribute A 作为根节点 root
            for A 可能的所有值 a:
                Let Examples(a) 代表所有属性 A==a 的样本
                给根节点增加一个 branch (检查 A==a)
                if Examples(a) 为空:
                    创建一个叶节点 with 占大多数的标签
                else:
                    Build_DcisionTree(Examples(a),Attributes)
```



决策树的递归中断

有三种情形会导致递归返回的三种情形：

- ① 当前结点包含的样本全属于同一类别，无需划分；
- ② 当前属性集为空，或是所有样本在所有属性上取值相同，无法划分；
- ③ 当前结点包含的样本集合为空，不能划分。

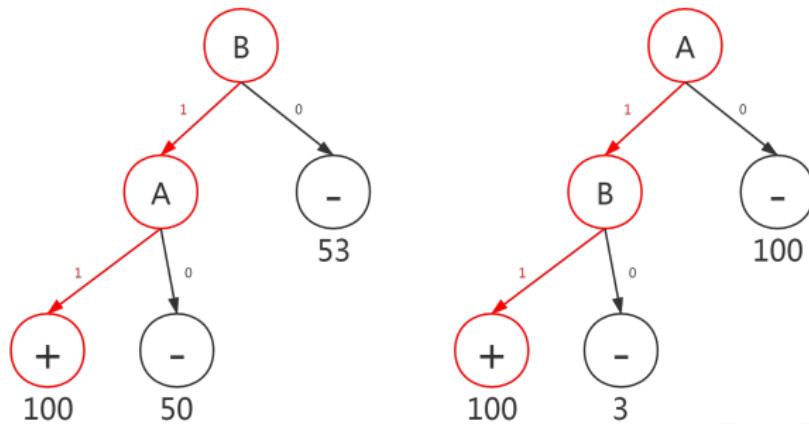
解决方案：

- ① 第一种情况，可视为决策树正常结束
- ② 第二种情况，将当前节点标记为叶结点，并将其类别设定为该结点所含样本最多的类别
- ③ 第三种情况，把当前节点标记为叶节点，但将其类别设定为其父结点所含样本最多的类别。

决策树的选择

情况逐渐复杂，如何挑选决策树：

样本数量	属性 A	属性 B	标签
50	A=0	B=0	-
50	A=0	B=1	-
3	A=1	B=0	-
100	A=1	B=1	+



决策树的选择

给定的一批数据，往往可以用多种决策树建模：

最好情况

最好情况下，只需要在某个特征上分一次就可以完成分类，条件是这个特征有“主键”的决定性作用。

最坏情况

最坏情况下，决策树的深度为数据集中相异的样本数量，即每个样本都对应了决策树的一个叶子节点。这显然产生了“过拟合问题”

更为一般的情况下如何选择？是否有可以度量的指标？



熵 (Entropy)

熵的定义

熵是一种描述系统稳定性和纯净程度的常见指标：

$$\text{Entropy } (S) = - \sum_{i=1}^c P_i \log P_i$$

其中 P_i 代表样本是第 i 类的概率。

在二分类（二叉树）的情况下表示为：

$$\text{Entropy } (S) = -P_+ \log P_+ - P_- \log P_-$$

其中 P_+, P_- 代表正样本/负样本在总体 S 中出现的概率

- ① 如果所有样本都属于一类， $\text{Entropy}=0$
- ② 如果两类样本数量相同， $\text{Entropy}=1$

决策树与熵

决策树的优化目标

决策树的优化目标是使用尽可能浅的树结构对所有数据进行分类。

- ① 决策树选择属性的依据是**让每一个分支中的样本标签尽可能纯净**，这样这个分支就更接近叶子节点，树的深度才会更小。
- ② 熵可以表征叶子结点中样本标签分布是否均匀，熵越小，样本标签越纯净；熵越大，样本标签分布越平均。**选择所有分支熵最小的属性**成为决策树建树的策略。
- ③ 这个想法最开始由 Quinlan 在 1975 年应用在 ID3 算法中。



信息增益（Information Gain）与 ID3

ID3 算法

ID3 算法（Iterative Dichotomiser 3 迭代二叉树 3 代）是建立在奥卡姆剃刀的基础上的一种决策树算法。其基本思想是选取使得信息增益最大的特征进行分裂。

属性 A 的信息增益是由该属性产生分支而减少的熵，计算公式如下：

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{a \in \text{values}(A)} \frac{|S_a|}{|S|} \text{Entropy}(S_a)$$

S 代表某个样本的集合，A 代表某个特征，a 代表 A 的一个取值。
如果 A 产生的分支熵较低，那么属性 A 的信息增益就更高。

信息增益 (Information Gain) 与 ID3

回到刚才的例子上，在第一次选择时

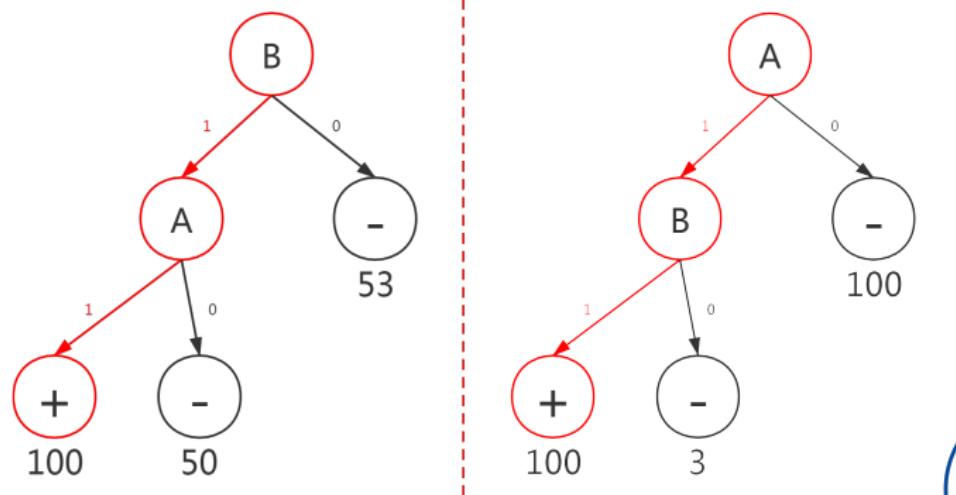
$$\begin{aligned} \text{Gain}(S, A) &= \text{Entropy}(S) - \sum_{a \in \{0,1\}} \frac{|S_a|}{|S|} \text{Entropy}(S_a) \\ &= - \sum_{i=1}^{\{+,-\}} P_i \log P_i - \sum_{a \in \{0,1\}} \frac{|S_a|}{|S|} \text{Entropy}(S_a) \\ &= 0.693 - \frac{103}{203} * \text{Entropy}(\{100+, 3-\}) - \frac{100}{203} * \text{Entropy}(\{100-\}) \\ &= 0.626 \end{aligned}$$

$$\begin{aligned} \text{Gain}(S, B) &= \text{Entropy}(S) - \sum_{b \in \{0,1\}} \frac{|S_b|}{|S|} \text{Entropy}(S_b) \\ &= 0.222 \end{aligned}$$



信息增益 (Information Gain) 与 ID3

通过比较信息增益，从 A 开始分更好，所以 ID3 选择从 A 开始分



信息增益率与 C4.5

- ① ID3 存在一个问题，那就是越细小的分割分类错误率越小，所以 ID3 会越分越细，训练集错误率达到 0，但是一旦有新来的样本立刻出现问题（过拟合）
- ② ID3 的改进版 C4.5 采用了信息增益率这样一个概念

$$\text{SplitInfo}(S, A) = - \sum_{j=1}^a \frac{|S_j|}{|S|} \times \log_2 \left(\frac{|S_j|}{|S|} \right)$$

$$\text{GainRatio}(S, A) = \frac{\text{Gain}(S, A)}{\text{SplitInfo}(S, A)}$$

- ① 显然，分割太细分母增加，信息增益率会降低，相当于对过拟合进行惩罚



分类回归树（CART）与连续特征值

- ① 当某一特征不是离散值的时候，决策树产生分支需要做额外的“阈值确定”，简单来说就是通过划分阈值将连续的数据划分为离散的区间，进而重新采用离散特征值的评估方法。
- ② 如何划分阈值才能使离散化的特征达到决策树的优化目标？（比如我们之前提到的信息增益、信息增益率、gini 指数等）
- ③ CART 将 n 个样本的连续特征值从大到小排序，产生 $n-1$ 个间隔，逐一比较这些间隔作为阈值划分所产生的 gini 指数，从而确定阈值。



基尼系数 (Gini Index)

- ① GINI 指数：总体内包含的类别越杂乱，GINI 指数就越大（跟熵的概念很相似）
- ② 对决策树的节点 t , Gini 指数计算公式如下：

$$\text{Gini}(t) = 1 - \sum_i [p(c_i | t)]^2$$

- ① 分类学习过程的本质是样本不确定性程度的减少（即熵减过程），故应选择最小 Gini 指数的特征分裂。
- ② 父节点对应的样本集合为 S , CART 选择特征 A 分裂为两个子节点，对应集合为 S_L 与 S_R ; 分裂后的 Gini 指数定义如下：

$$Gini(S, A) = \frac{|S_L|}{|S|} \text{Gini}(S_L) + \frac{|S_R|}{|S|} \text{Gini}(S_R)$$



互动环节

在特征空间中，好的分类器应该是怎么样的？



① 认识分类

- 分类的定义
 - 分类的基本步骤

② 分类模型的评估

- 二分类模型的评估
 - 多分类模型的评估
 - ROC 曲线

③ 决策树

- 决策树原理
 - 决策树的选择

④ 线性判别分析

- 从 PCA 到 LDA
 - 线性判别分析 LDA
 - 多分类 LDA

⑤ 从分类到回归

- ## • 线性回归

6 集成学习

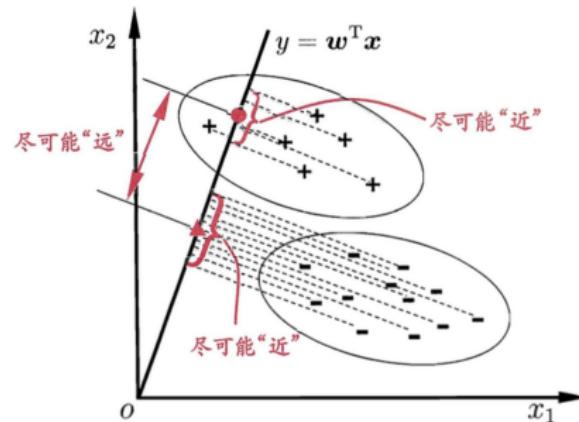
- 集成学习概述
 - Bagging, Stacking
 - Boosting



线性判别分析 LDA

线性判别分析 (Linear Discriminant Analysis, LDA) [1] 是一种经典的有监督数据降维/分类方法。

主要思想：将高维空间中的数据投影到较低维的空间中，使同类样本的投影点之间尽可能接近，不同类样本的投影点中心尽可能远离。即投影后类内方差最小，类间距离最大。



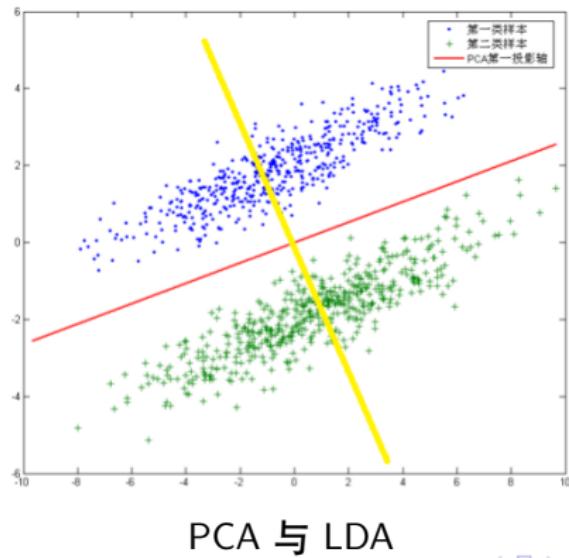
以二分类为例



为什么要用 LDA ?

PCA：无监督数据降维方法，约束目标为将数据投影到方差最大的若干个相互正交的方向上，从而具有更大的发散性

LDA：有监督数据降维方法，利用了标签信息，约束目标为最小化类内方差，最大化类间距离，从而具有更好的分类性能。



PCA 与 LDA

线性判别分析 LDA——二分类

优化目标：

所谓线性，就是将数据点投影到直线（可能为多条直线）上，即

$$z = w^T x$$

z 为投影后的样本点， w 为投影向量。将数据投影到直线 w 上，则两类中心的投影分别为 $w^T \mu_0$ 和 $w^T \mu_1$ ，协方差分别为 $w^T \sum_0 w$ 和 $w^T \sum_1 w$ 。

$$\begin{aligned} \sum_{x \in D_i} (w^T x - w^T \mu_i)^2 &= \sum_{x \in D_i} (w^T (x - \mu_i))^2 = \sum_{x \in D_i} w^T (x - \mu_i)(x - \mu_i)^T w \\ &= w^T \sum_{x \in D_i} [(x - \mu_i)(x - \mu_i)^T] w \\ &= w^T \sum_i w \end{aligned}$$



线性判别分析 LDA——二分类

LDA 的优化目标

LDA 的优化目标是寻找最优的投影平面，使得同类样本的投影点尽可能接近（协方差尽量小）即 $w^T \sum_0 w + w^T \sum_1 w$ 尽可能小；同时让不同类样本的投影点尽可能远，可以让类中心之间的距离尽可能大，即 $\|w^T \mu_0 - w^T \mu_1\|_2^2$ 尽可能大

上述目标可以表示为：

$$J = \frac{\|w^T \mu_0 - w^T \mu_1\|_2^2}{w^T (\sum_0 + \sum_1) w} = \frac{w^T (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T w}{w^T (\sum_0 + \sum_1) w}$$



线性判别分析 LDA——二分类

$$J = \frac{w^T(\mu_0 - \mu_1)(\mu_0 - \mu_1)^T w}{w^T(\Sigma_0 + \Sigma_1)w}$$

定义类间散度矩阵 (between-class scatter matrix) :

$$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T$$

以及类内散度矩阵 (within-class scatter matrix) :

$$S_w = \sum_0 + \sum_1 = \sum_{x \in D_0} (x - \mu_0)(x - \mu_0)^T + \sum_{x \in D_1} (x - \mu_1)(x - \mu_1)^T$$

由此，我们可以将上述优化目标重新写为：

$$J = \frac{w^T S_b w}{w^T S_w w}$$



线性判别分析 LDA——二分类

$$\max_w \quad J(w) = \frac{w^T S_b w}{w^T S_w w}$$

注意到上式的分子和分母都是关于 w 的二次项，因此上式的解与 w 的长度无关，只与其方向有关。由此，令 $w^T S_w w = 1$ ，则上式等价于：

$$\begin{aligned} & \min_w \quad -w^T S_b w \\ & s.t. \quad w^T S_w w = 1 \end{aligned}$$

利用拉格朗日乘子法，上式等价于：

$$\begin{aligned} L(w, \lambda) &= -w^T S_b w + \lambda(w^T S_w w - 1) \\ \Rightarrow \frac{dL}{dw} &= -2S_b w + 2\lambda S_w w = 0 \\ \Rightarrow S_b w &= \lambda S_w w \end{aligned}$$

其中 λ 为拉格朗日乘子。



线性判别分析 LDA——二分类

$$\begin{aligned}S_b w &= \lambda S_w w \\ \Rightarrow S_w^{-1} S_b w &= \lambda w \\ \Rightarrow |S_w^{-1} S_b - \lambda I| &= 0\end{aligned}$$

I 为单位矩阵，由此可以计算出特征值 λ ，从而进一步求解 w 。



线性判别分析 LDA——算法流程

输入：数据集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, 其中, 任意样本 x_i 为 m 维向量, $y_i \in \{C_1, C_2, \dots, C_K\}$, 降维到的维度为 d 。

输出：降维后的数据集 D' 。

- 1) 计算各个类的中心 μ_i 和所有样本的中心 μ
- 2) 计算类内散度矩阵 S_w
- 3) 计算类间散度矩阵 S_b
- 4) 计算矩阵 $S_w^{-1}S_b$
- 5) 计算矩阵 $S_w^{-1}S_b$ 的特征值和特征向量, 按从大到小的顺序选取 d 个特征值和对应的 d 个特征向量, 得到投影矩阵 w
- 6) 对数据集里的每一个样本 x_i , 投影得到新的样本 $z_i = w^T x_i$
- 7) 得到降维后的数据集 $D' = \{(z_1, y_1), (z_2, y_2), \dots, (z_N, y_N)\}$



线性判别分析 LDA——例子

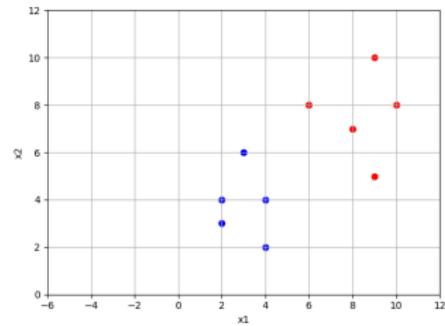
一个简单的例子：现有一个包含两个类的二维数据集 D ，需要将其投影到一条直线 w 上。

第 0 类样本：

$$D_0 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \left\{ \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 3 \\ 6 \end{bmatrix}, \begin{bmatrix} 4 \\ 4 \end{bmatrix} \right\}$$

第 1 类样本：

$$D_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \left\{ \begin{bmatrix} 9 \\ 10 \end{bmatrix}, \begin{bmatrix} 6 \\ 8 \end{bmatrix}, \begin{bmatrix} 9 \\ 5 \end{bmatrix}, \begin{bmatrix} 8 \\ 7 \end{bmatrix}, \begin{bmatrix} 10 \\ 8 \end{bmatrix} \right\}$$



线性判别分析 LDA——例子

各个类的中心 μ_i :

$$\mu_0 = \frac{1}{N_0} \sum_{x \in D_0} x = \frac{1}{5} \left[\begin{bmatrix} 4 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 \\ 4 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 3 \\ 6 \end{bmatrix} + \begin{bmatrix} 4 \\ 4 \end{bmatrix} \right] = \begin{bmatrix} 3 \\ 3.8 \end{bmatrix}$$

$$\mu_1 = \frac{1}{N_1} \sum_{x \in D_1} x = \frac{1}{5} \left[\begin{bmatrix} 9 \\ 10 \end{bmatrix} + \begin{bmatrix} 6 \\ 8 \end{bmatrix} + \begin{bmatrix} 9 \\ 5 \end{bmatrix} + \begin{bmatrix} 8 \\ 7 \end{bmatrix} + \begin{bmatrix} 10 \\ 8 \end{bmatrix} \right] = \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix}$$

第 0 类的协方差矩阵:

$$\begin{aligned} \Sigma_0 &= \sum_{x \in D_0} (x - \mu_0)(x - \mu_0)^T = \left[\begin{bmatrix} 4 \\ 2 \end{bmatrix} - \begin{bmatrix} 3 \\ 3.8 \end{bmatrix} \right]^2 + \left[\begin{bmatrix} 2 \\ 4 \end{bmatrix} - \begin{bmatrix} 3 \\ 3.8 \end{bmatrix} \right]^2 \\ &\quad + \left[\begin{bmatrix} 2 \\ 3 \end{bmatrix} - \begin{bmatrix} 3 \\ 3.8 \end{bmatrix} \right]^2 + \left[\begin{bmatrix} 3 \\ 6 \end{bmatrix} - \begin{bmatrix} 3 \\ 3.8 \end{bmatrix} \right]^2 + \left[\begin{bmatrix} 4 \\ 4 \end{bmatrix} - \begin{bmatrix} 3 \\ 3.8 \end{bmatrix} \right]^2 \\ &= \begin{bmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{bmatrix} \end{aligned}$$



线性判别分析 LDA——例子

第 1 类的协方差矩阵：

$$\begin{aligned}\sum_1 &= \sum_{x \in D_1} (x - \mu_1)(x - \mu_1)^T = \left[\begin{bmatrix} 9 \\ 10 \end{bmatrix} - \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix} \right]^2 + \left[\begin{bmatrix} 6 \\ 8 \end{bmatrix} - \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix} \right]^2 \\ &\quad + \left[\begin{bmatrix} 9 \\ 5 \end{bmatrix} - \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix} \right]^2 + \left[\begin{bmatrix} 8 \\ 7 \end{bmatrix} - \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix} \right]^2 + \left[\begin{bmatrix} 10 \\ 8 \end{bmatrix} - \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix} \right]^2 \\ &= \begin{bmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{bmatrix}\end{aligned}$$

类内散度矩阵 S_w ：

$$\begin{aligned}S_w &= \sum_0 + \sum_1 = \begin{bmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{bmatrix} + \begin{bmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{bmatrix} \\ &= \begin{bmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{bmatrix}\end{aligned}$$



线性判别分析 LDA——例子

类间散度矩阵 S_b :

$$\begin{aligned} S_b &= (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T \\ &= \left[\begin{bmatrix} 3 \\ 3.8 \end{bmatrix} - \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix} \right] \left[\begin{bmatrix} 3 \\ 3.8 \end{bmatrix} - \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix} \right]^T \\ &= \begin{bmatrix} -5.4 \\ -3.8 \end{bmatrix} \begin{bmatrix} -5.4 & -3.8 \end{bmatrix} \\ &= \begin{bmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{bmatrix} \end{aligned}$$



线性判别分析 LDA——例子

计算特征值：

$$S_w^{-1} S_b w = \lambda w$$

$$\Rightarrow |S_w^{-1} S_b - \lambda I| = 0$$

$$\Rightarrow \begin{vmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{vmatrix}^{-1} \begin{bmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

$$\Rightarrow \begin{vmatrix} 0.3045 & 0.0166 \\ 0.0166 & 0.1827 \end{vmatrix} \begin{bmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

$$\Rightarrow \begin{vmatrix} 9.2213 - \lambda & 6.489 \\ 4.2339 & 2.9794 - \lambda \end{vmatrix}$$

$$= (9.2213 - \lambda)(2.9794 - \lambda) - 6.489 * 4.2339 = 0$$

$$\Rightarrow \lambda^2 - 12.2007\lambda = 0 \quad \Rightarrow \lambda(\lambda - 12.2007) = 0$$

$$\Rightarrow \lambda_1 = 0, \lambda_2 = 12.2007$$



线性判别分析 LDA——例子

计算特征向量 w :

$$S_w^{-1} S_b w = \lambda w$$

$$\Rightarrow \begin{bmatrix} 9.2213 & 6.489 \\ 4.2339 & 2.9794 \end{bmatrix} \underbrace{\begin{bmatrix} w_{11} \\ w_{12} \end{bmatrix}}_{w_1} = \underbrace{0}_{\lambda_1} \underbrace{\begin{bmatrix} w_{11} \\ w_{12} \end{bmatrix}}_{w_1}$$

and

$$\begin{bmatrix} 9.2213 & 6.489 \\ 4.2339 & 2.9794 \end{bmatrix} \underbrace{\begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix}}_{w_2} = \underbrace{12.2007}_{\lambda_2} \underbrace{\begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix}}_{w_2}$$

$$\Rightarrow w_1 = \begin{bmatrix} -0.5755 \\ 0.8178 \end{bmatrix}, \quad w_2 = \begin{bmatrix} 0.9088 \\ 0.4173 \end{bmatrix}$$

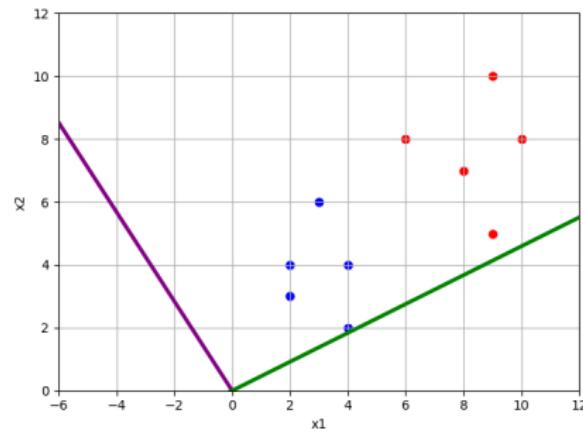


线性判别分析 LDA——例子

$$\lambda_1 = 0 \quad , \quad \lambda_2 = 12.2007$$

$$w_1 = \begin{bmatrix} -0.5755 \\ 0.8178 \end{bmatrix} \quad , \quad w_2 = \begin{bmatrix} 0.9088 \\ 0.4173 \end{bmatrix}$$

$\lambda = J(w)$, 而我们的目标是最大化 $J(w)$, 因此选择最大的 λ , 即 $\lambda_2 = 12.2007$ 。



线性判别分析 LDA——多分类

与二分类相似，对于第 i 类样本，类中心的投影和协方差分别为 $w^T \mu_i$ 和 $w^T \sum_i w$ 。

各个类投影后的协方差之和为:

$$w^T \sum_{i=1}^K \sum_i w = w^T \sum_{i=1}^K \sum_{x \in D_i} (x - \mu_i)(x - \mu_i)^T w$$

所有类别中心的距离之和为：

$$\sum_{\substack{i,j \\ i \neq j}} d_{ij} = w^T \sum_{\substack{i,j \\ i \neq j}} [(\mu_i - \mu_j)(\mu_i - \mu_j)^T] w$$



线性判别分析 LDA——多分类

优化目标为：

$$\max_w \quad J(w) = \frac{w^T \sum_{\substack{i,j \\ i \neq j}} [(\mu_i - \mu_j)(\mu_i - \mu_j)^T] w}{w^T \sum_{i=1}^K \sum_{x \in D_i} (x - \mu_i)(x - \mu_i)^T w}$$

相对应的，类间散度矩阵 S_b 和类内散度矩阵 S_w 分别为：

$$S_b = \sum_{\substack{i,j \\ i \neq j}} [(\mu_i - \mu_j)(\mu_i - \mu_j)^T]$$

$$S_w = \sum_{i=1}^K \sum_{x \in D_i} (x - \mu_i)(x - \mu_i)^T$$



线性判别分析 LDA——多分类

同样，我们可以得到优化目标：

$$\max_w J(w) = \frac{w^T S_b w}{w^T S_w w}$$

其余推导与二分类相同，可以转化为：

$$\begin{aligned} S_b w &= \lambda S_w w \\ \Rightarrow S_w^{-1} S_b w &= \lambda w \end{aligned}$$

即求特征值 λ 和特征向量 w 的过程。



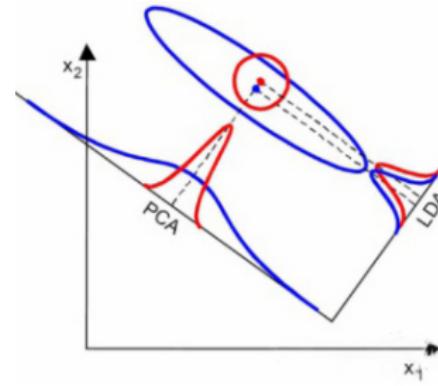
线性判别分析 LDA——优点与缺点

主要优点：

- 在降维过程中可以使用类别标签的先验知识

主要缺点：

- LDA 不适合对非高斯分布样本进行降维
- 降维最多降到 $K - 1$ 的维数
- 在 $J(w)$ 更依赖于协方差时，效果不佳



1 认识分类

- 分类的定义
- 分类的基本步骤

2 分类模型的评估

- 二分类模型的评估
- 多分类模型的评估
- ROC 曲线

3 决策树

- 决策树原理
- 决策树的选择

4 线性判别分析

- 从 PCA 到 LDA
- 线性判别分析 LDA
- 多分类 LDA

5 从分类到回归

● 线性回归

6 集成学习

- 集成学习概述
- Bagging, Stacking
- Boosting



线性回归



编号	色泽	根蒂	敲声	价格
1	青绿	蜷缩	浊响	2
2	乌黑	蜷缩	浊响	2
3	青绿	硬挺	清脆	1.5
4	乌黑	稍蜷	沉闷	1.5

吃过的瓜



线性模型

线性模型

给定数据集 $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$, 其中 $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 是第 i 个样本的 d 维特征向量, 线性模型 (Linear Model) 的目标是学习一个通过属性的线性组合来进行预测的函数, 即

$$f(x_i) = w_1 x_{i1} + w_2 x_{i2} + \dots + w_d x_{id} + b$$

向量形式为

$$f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + b$$

$$f_{\text{价格}}(x) = 0.2 * x_{\text{色泽}} + 0.5 * x_{\text{根蒂}} + 0.3 * x_{\text{敲声}} + 1$$



线性回归

线性回归

线性回归的目的是让线性模型尽可能接近标签，即

$$f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + b \quad \simeq \quad y_i$$

单元线性回归最优参数：

$$\begin{aligned}(w^*, b^*) &= \arg \min_{(w,b)} \sum_{i=1}^N (f(x_i) - y_i)^2 \\ &= \arg \min_{(w,b)} \sum_{i=1}^N (y_i - wx_i - b)^2\end{aligned}$$



最小二乘法求解单元线性回归

线性回归的目标是寻找最优的 w, b 使得下述式子最小化：

$$E_{(w,b)} = \sum_{i=1}^N (y_i - wx_i - b)^2$$

将上式对两个参数分别求导可得：

$$\frac{\partial E_{(w,b)}}{\partial w} = 2 \left(w \sum_{i=1}^N x_i^2 - \sum_{i=1}^N (y_i - b) x_i \right)$$

$$\frac{\partial E_{(w,b)}}{\partial b} = 2 \left(Nb - \sum_{i=1}^N (y_i - wx_i) \right)$$

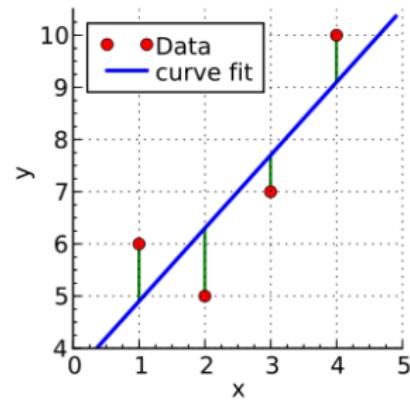


最小二乘法求解单元线性回归

令上述导数为 0, 可得到线性回归的最优参数:

$$w = \frac{\sum_{i=1}^m y_i (x_i - \bar{x})}{\sum_{i=1}^m x_i^2 - \frac{1}{m} (\sum_{i=1}^m x_i)^2}, \quad \bar{x} = \frac{1}{m} \sum_{i=1}^m x_i$$

$$b = \frac{1}{m} \sum_{i=1}^m (y_i - wx_i)$$



单元线性回归等价于在二维空间中寻找一条直线，使得这条直线上相同属性的点与真实标签的距离之和尽量小。



多元线性回归

多元线性回归

真实场景中的数据往往是高维的，高维数据的线性回归也称为多元线性回归 (multivariate linear regression)

$$f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + b, \text{ 使得 } f(\mathbf{x}_i) \simeq y_i$$

矩阵形式表达为：

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} & 1 \\ x_{21} & x_{22} & \dots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{md} & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1^T & 1 \\ \mathbf{x}_2^T & 1 \\ \vdots & \vdots \\ \mathbf{x}_m^T & 1 \end{pmatrix}$$



多元线性回归

最优参数向量 w^* 的目标是最小化

$$E_{\hat{w}} = (\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})^T(\mathbf{y} - \mathbf{X}\hat{\mathbf{w}})$$

对上式子求导：

$$\frac{\partial E_{\hat{w}}}{\partial \hat{w}} = 2\mathbf{X}^T(\mathbf{X}\hat{\mathbf{w}} - \mathbf{y})$$

令导数为 0，可得 w^* 的最优解：

$$\hat{\mathbf{w}}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$



用回归的方式做分类



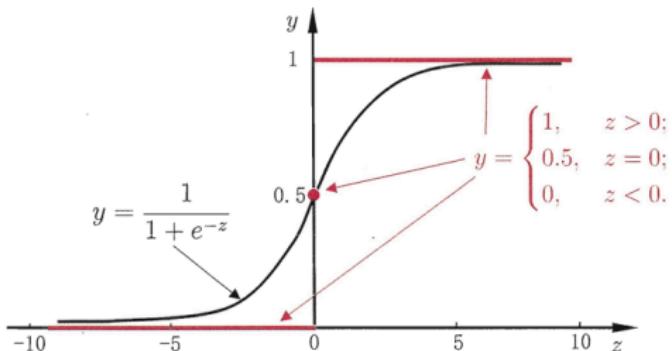
编号	色泽	根蒂	敲声	熟瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

吃过的瓜



使用回归的方式做分类的问题

线性回归多用于数值拟合，但是在做分类任务时，往往需要将分类任务的标签 y 与线性回归模型的预测联系起来。在二分类任务中，标签 $y \in \{0, 1\}$ 。



Sigmoid 函数

单位阶跃函数：

$$y = \begin{cases} 0, & z < 0 \\ 0.5, & z = 0 \\ 1, & z > 0 \end{cases}$$

Sigmoid 函数：

$$y = \frac{1}{1 + e^{-(w^T x + b)}}$$



对数几率函数

y 表示样本为正例的概率, $1 - y$ 表示样本为负例的概率

$$p(y = 1|x) = y = \frac{e^{\mathbf{w}^T \mathbf{x} + b}}{1 + e^{\mathbf{w}^T \mathbf{x} + b}}$$

$$p(y = 0|x) = 1 - y = \frac{1}{1 + e^{\mathbf{w}^T \mathbf{x} + b}}$$

回归函数可以通过简单的数学变换得到:

$$\mathbf{w}^T \mathbf{x} + b = \ln \frac{y}{1 - y}$$

$\frac{y}{1-y}$ 表示样本为正例的相对概率, 也称为“几率”(odds), 其对数形式则为“对数几率”(log odds, logit) 因此该函数也称为对数几率函数。

对数几率回归

最优的参数 $\beta = (w, b)$ 可以通过“极大似然法”(maximum likelihood method)以最大化似然函数的形式求得：

$$\begin{aligned}
\ell(\boldsymbol{\beta}) &= \sum_{i=1}^N \ln p(y_i \mid \mathbf{x}_i; \boldsymbol{\beta}) \\
&= \sum_{i=1}^N \ln[y_i p_1(\mathbf{x}_i; \boldsymbol{\beta}) + (1 - y_i) p_0(\mathbf{x}_i; \boldsymbol{\beta})] \\
&= \sum_{i=1}^N y_i \boldsymbol{\beta}^T \mathbf{x}_i - \ln \left(1 + e^{\boldsymbol{\beta}^T \mathbf{x}_i} \right) \\
&= \sum_{i=1}^N y_i [\mathbf{w}^T \mathbf{x} + b] - \ln \left(1 + e^{\mathbf{w}^T \mathbf{x} + b} \right)
\end{aligned}$$



对数几率函数

对数几率函数是数据挖掘和机器学习中最重要的函数之一。其具有如下的性质：

- ① 对输入数据没有任何限制，取值范围 $-\infty \rightarrow +\infty$ ，可用于多种特征（不需要额外缩放）
 - ② 可解释性强，从特征的权重可以看到不同的特征对最后结果的影响；
 - ③ 取值范围为 $(0, 1)$ ，可用于概率模型分类器

但仍然存在如下缺陷：

- ① 容易存在梯度消失问题
 - ② 没有做 0 中心化，反向传播时容易全正全负



① 认识分类

- 分类的定义
- 分类的基本步骤

② 分类模型的评估

- 二分类模型的评估
- 多分类模型的评估
- ROC 曲线

③ 决策树

- 决策树原理
- 决策树的选择

④ 线性判别分析

- 从 PCA 到 LDA
- 线性判别分析 LDA
- 多分类 LDA

⑤ 从分类到回归

- 线性回归

⑥ 集成学习

- 集成学习概述
- Bagging, Stacking
- Boosting



Ensemble Methods

当单一分类模型表现不佳的时候，有没有办法聚合多个模型的决策以达到更好的分类效果呢？

集成方法

为了更好地解决特定的机器学习问题，带有策略地生成和组合多个模型的过程。

- 提升单一模型的性能。
- 减少选择不佳模型的可能性。
- 这类方法被称为模型聚合，常见的方法有：
 - Bagging：将若干分类器预测的结果做平均
 - Random Forest
 - Boosting：将若干分类器预测的结果加权



Ensemble Methods

如何组合分类器的输出结果？

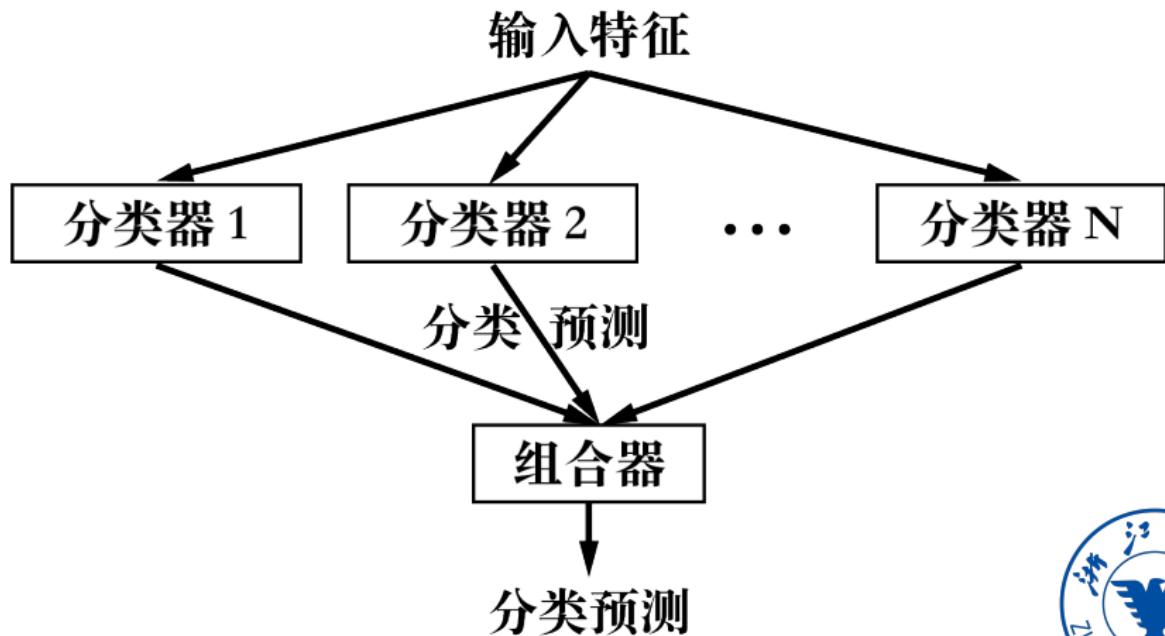
- 求平均
- 投票
 - 多数投票
 - 随机森林
 - 加权多数投票
 - AdaBoost
- 可学习组合器

集成学习的关键

- 分类器之间能够互相纠正彼此的错误。
- 不同分类器有不同的输出结果，否则集成没有意义。



集成流程



Bootstrap

Bootstrap 起源

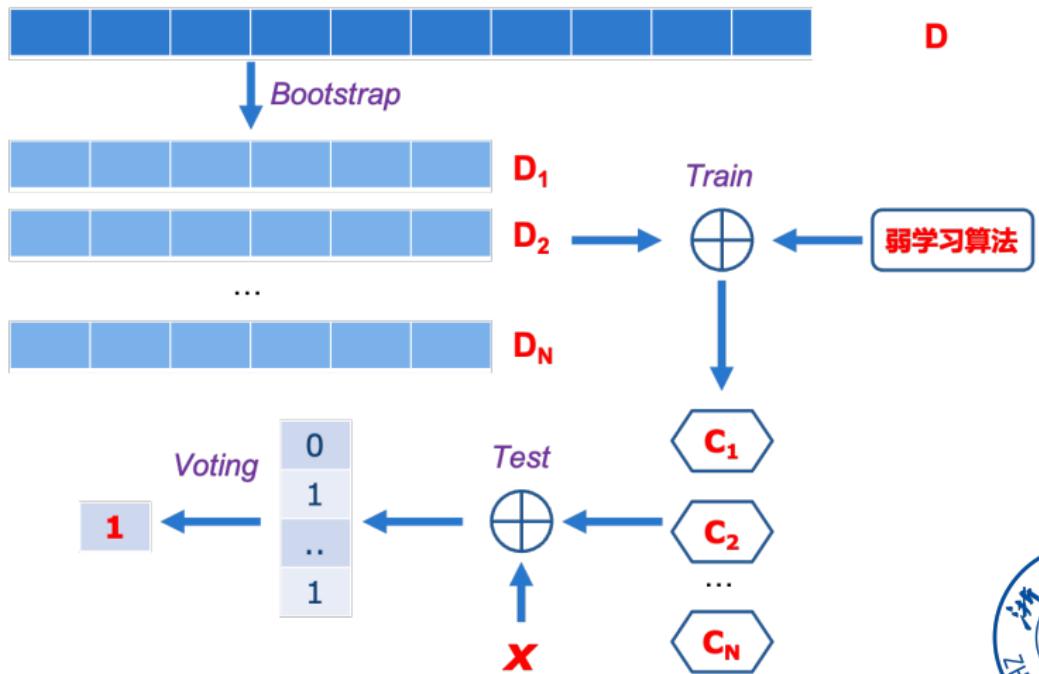
《吹牛大王历险记》“pull yourself up by your bootstraps”，常指自力更生。

集成学习中的 Bootstrap

在机器学习和集成学习中，Bootstrap 往往指对原始数据集进行有放回重采样，形成一系列子样本。

由于很多分类器的精度并不高，因此往往希望通过构造不同的分类器并进行投票来提升整体精度。在模型框架有限的情况下，使用 bootstrap 生成不同的训练数据，可以最低成本地构造不同的分类器。

Bagging: Bootstrap Aggregation



随机森林

随机森林是决策树的扩展，其基本流程如下：

- ① 有放回的抽取 N 次，每次抽取 1 个，最终形成了 N 个样本用来训练一个决策树。
- ② 当每个样本有 M 个属性时，在决策树的每个节点需要分裂时，随机从这 M 个属性中选取出 m ($m \ll M$) 个属性，然后从这 m 个属性中选择 1 个属性作为该节点的分裂属性。
- ③ 重复数据选择和分裂过程，一直到不能够再分裂为止。
- ④ 按照步骤 1, 2, 3 建立大量的决策树，构成随机森林了。



随机森林的优缺点

① 优点：

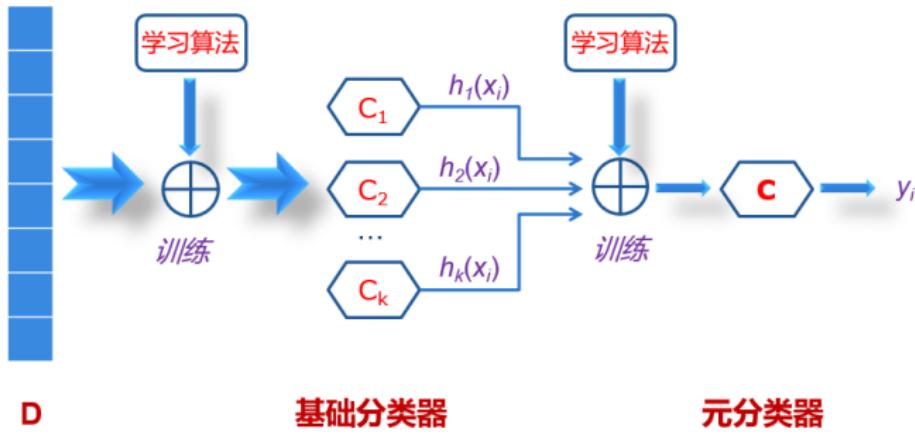
- 1、可以应对高维度（特征很多）的数据，并且不用降维，无需做特征选择。
- 2、可以通过投正确票的决策树的结构判断特征的重要程度。
- 3、不容易过拟合，学习速度快，可以并行训练整个随机森林。
- 4、如果有很大一部分的特征遗失，仍可以维持准确度

② 缺点：

- 1、随机森林无法控制模型内部的运行，只能在不同的参数和随机种子之间进行尝试。
- 2、可能有很多相似的决策树，多数人的暴政掩盖了真实的结果。



Stacking



Boost

Boosting 算法的工作机制：

- ① 从训练集用初始权重训练出一个弱学习器 1，根据弱学习的学习误差率表现来更新训练样本的权重，使得之前弱学习器 1 学习误差率高的训练样本点的权重变高
- ② 基于调整权重后的训练集来训练弱学习器 2.
- ③ 如此重复进行，直到弱学习器数达到事先指定的数目 T ，最终将这 T 个弱学习器通过集合策略进行整合，得到最终的强学习器。



AdaBoost

输入输出

- 输入：训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ ，其中 $x_i \in \mathcal{X} \subseteq R^n, y_i \in \mathcal{Y} = \{+1, -1\}$
- 输出：分类器 $G(x)$

1. 初始化训练数据的权值分布

$$D_1 = (w_{11}, w_{12}, \dots, w_{1N}), \quad w_{1i} = \frac{1}{N}, \quad i = 1, 2, \dots, N$$

w_{mi} 表示第 m 轮迭代中第 i 个样本的权值

2. 初始化 $m = 1, 2, \dots, M$ 基本分类器

$$G_m(x) : \mathcal{X} \rightarrow \{-1, +1\}$$



AdaBoost

2.2 计算 $G_m(x)$ 在训练数据集上的分类误差率

$$e_m = \sum_{i=1}^N P(G_m(x_i) \neq y_i)$$

$$= \sum_{i=1}^N w_{mi} I(G_m(x_i) \neq y_i)$$

2.3 计算 $G_m(x)$ 的系数

$$\alpha_m = \frac{1}{2} \log \frac{1 - e_m}{e_m}$$



AdaBoost

2.4 更新训练数据集的权值分布

$$D_{m+1} = (w_{m+1,1}, \dots, w_{m+1,i}, \dots, w_{m+1,N})$$

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} \exp(-\alpha_m y_i G_m(x_i)),$$

$$= \begin{cases} \frac{w_{mi}}{Z_m} \exp(-\alpha_m), & G_m(x_i) = y_i \\ \frac{w_{mi}}{Z_m} \exp(\alpha_m), & G_m(x_i) \neq y_i \end{cases} \quad i = 1, 2, \dots, N$$

其中, Z_m 是归一化因子, 为了使概率分布和为 1

$$Z_m = \sum_{i=1}^N w_{mi} \exp(-\alpha_m y_i G_m(x_i))$$

3. 构建基本分类器的线性组合

$$f(x) = \sum_{m=1}^M \alpha_m G_m(x)$$



要点总结

- ① 认识分类
 - 分类的定义
 - 分类的基本步骤
 - ② 分类模型的评估
 - 二分类模型的评估
 - 多分类模型的评估
 - ROC 曲线
 - ③ 决策树
 - 决策树原理
 - 决策树的选择
 - ④ 线性判别分析
 - 从 PCA 到 LDA
 - 线性判别分析 LDA
 - 多分类 LDA
 - ⑤ 从分类到回归
 - 线性回归
 - ⑥ 集成学习
 - 集成学习概述
 - Bagging, Stacking
 - Boosting



参考文献



BALAKRISHNAMA, S., AND GANAPATHIRAJU, A.
Linear discriminant analysis-a brief tutorial.
Institute for Signal and information Processing 18, 1998 (1998), 1–8.

