

数据挖掘与应用

分类

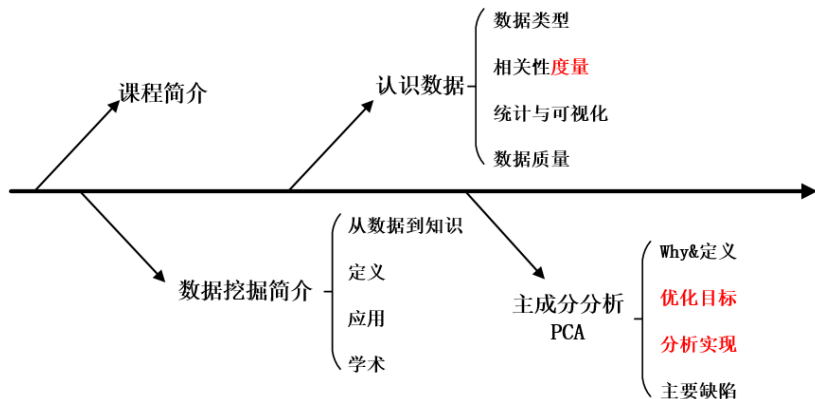
授课教师：周晟

浙江大学 软件学院

2024.9.24



上节课回顾

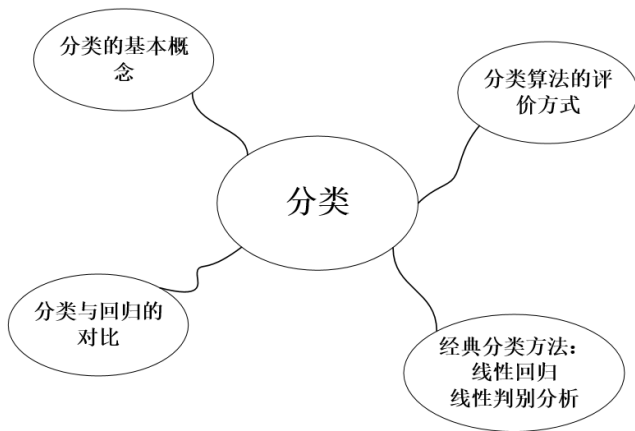


课程内容

- ① 认识分类
- ② 分类模型的评估
 - 二分类模型的评估
 - 多分类模型的评估
 - 分类模型的选择
- ③ 线性判别分析
 - 从 PCA 到 LDA
- ④ 从分类到回归
 - 线性判别分析 LDA
 - 多分类 LDA
 - 线性回归
- ⑤ 决策树
 - 决策树原理
 - 决策树的选择



本节课程结构



- 1 认识分类
- 2 分类模型的评估
 - 二分类模型的评估
 - 多分类模型的评估
 - 分类模型的选择
- 3 线性判别分析
 - 从 PCA 到 LDA
- 4 从分类到回归
 - 线性回归
- 5 决策树
 - 决策树原理
 - 决策树的选择



认识分类

分类的定义

Classification is the grouping of related facts into classes. It may also refer to a process which brings together like things and separates unlike things.

– Wikipedia

分类是自然界最常见的数据挖掘任务之一：

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



动画 音乐 舞蹈 知识
番剧 国创 游戏 科技



分类的特点

分类——离散（定性）分析

- 西瓜好坏
- 垃圾邮件判别
- 界门纲目科属种

分类——有监督（有标签）

- 西瓜好坏
- 垃圾邮件判别
- 界门纲目科属种

回归——连续（定量）分析

- 西瓜定价
- 房价预测
- 风险评估

聚类——无监督（无标签）

- 消费者群体聚类
- 产品定位
- 离群点检测



- 1 认识分类
- 2 分类模型的评估
 - 二分类模型的评估
 - 多分类模型的评估
 - 分类模型的选择
- 3 线性判别分析
 - 从 PCA 到 LDA
- 4 从分类到回归
 - 线性回归
- 5 决策树
 - 决策树原理
 - 决策树的选择



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**类别不平衡**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

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

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

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

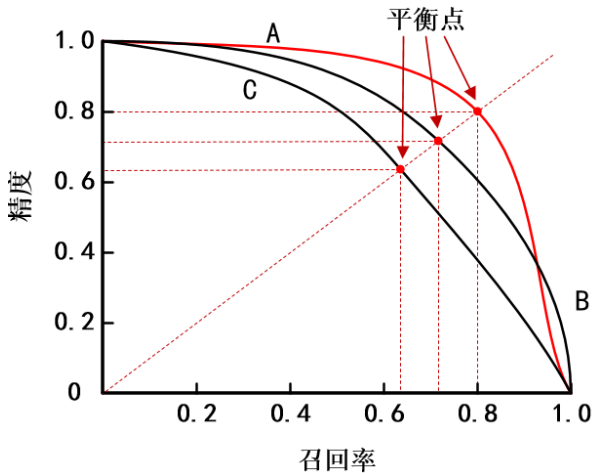
$$\text{recall} = \frac{TP}{TP + FN} = \frac{365}{365 + 5} = 0.986$$

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



精度与召回率



精度和召回率

高召回低精度的后果：六月飞雪窦娥冤



低精度 & 高召回的案例

有没有高精度低召回的例子？



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

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

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

以刷单检测为例

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

$$\textcircled{2} \beta = 1: F_{\beta} - \text{score} = \frac{(1+\beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}} = 0.962$$

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

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：

$$precision_i = \frac{TP_i}{TP_i + FP_i}$$

$$recall_i = \frac{TP_i}{TP_i + FN_i}$$

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

$$Precision_{\text{micro}} = \frac{\sum_{i=1}^K TP_i}{\sum_{i=1}^K TP_i + \sum_{i=1}^K FP_i}$$

$$Recall_{\text{micro}} = \frac{\sum_{i=1}^K TP_i}{\sum_{i=1}^K TP_i + \sum_{i=1}^K FN_i}$$



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:

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

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

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	合计
实际的类	红灯	39	0	1	40
	绿灯	1	33	5	39
	黄灯	3	4	10	17
	合计	43	37	16	96

Micro-F1:

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

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

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

Macro-F1:

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

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

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

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

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

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

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

多标签分类

多标签分类

在多标签分类问题中，每个实例可能会对应多个类别标签

Macro-F1 和 Micro-F1 也是作为评估模型多标签分类能力的主要指标，除此之外还有一些在多标签分类场景特有的评估指标如下：

- ① 汉明分数 (Hamming Score)：衡量类别标签被错分的次数。

$$\text{Hamming Score} = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \mathbb{I}(\hat{y}_{i,j} \neq y_{i,j})$$

- ② 标签排序分数 (Label Ranking Score)：衡量相关标签的预测概率值比不相关标签的预测概率值小的次数。

$$\text{LRS} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{1}{||y_i||_0(N - ||y_i||_0)} |\{(k, l) : \hat{y}_{ik} \leq \hat{y}_{il}, y_{ik} = 1, y_{il} = 0\}|$$

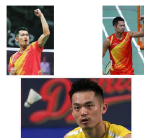
分类的基本步骤



周杰伦



陈赫



林丹



训练集



验证集

测试集



这是周杰伦 / 与其他两人不同

为什么要构造测试集和多个验证集？

机器学习中的过拟合

过拟合现象

当分类模型在训练数据集上表现很好，但是在测试数据集上表现不佳时，我们往往认为出现了**过拟合**现象。

出现过拟合的原因：

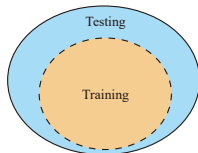
- ① 训练数据集过小，或者模式过于简单
- ② 训练数据集噪声过大
- ③ 模型参数量过大
- ④ 模型训练时间过长



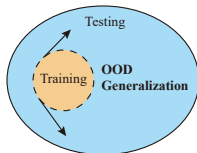
机器学习中的过拟合

如何减轻模型的过拟合？

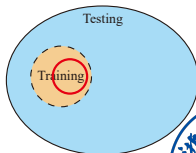
- ① 扩大训练数据集的数量和质量
- ② 参数/特征正则化
- ③ 特征选择
- ④ 早停策略 Early Stop
- ⑤ 数据增强



① “大”模型



②分布外泛化



③可信性评估



分类模型选择

给定上述分类度量指标，如何选择最优的分类模型？

单个分类模型的公平评估：

- ① 交叉验证 (Cross-Validation)
- ② 自助法 (Bootstrapping)

多个分类模型选择：

- ① 接收机工作特征曲线 (ROC Curve)
- ② 显著性检验



k-折交叉验证

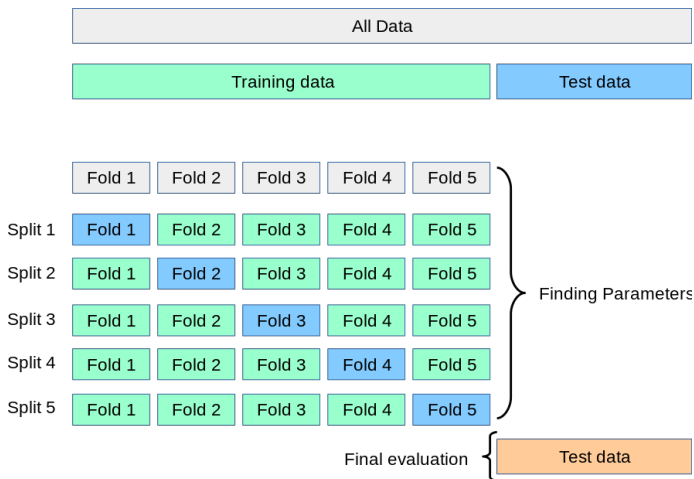
k-折交叉验证 (k-fold cross-validation)

将数据随机地划分为 k 个互不相交的子集或“折” D_1, D_2, \dots, D_k ，每个折的数据大致相等。在这些数据上将进行 K 次训练和验证，在第 i 次迭代中，分区 D_i 用作验证集，其余的用作训练集。最终的分类准确率是 k 次迭代正确分类的样本数据量除以样本量总和。

在 K 折交叉验证中，每个数据只有一次被用作验证， $K-1$ 次用于训练。
最常用的是 5-折和 10-折交叉验证。



5-折交叉验证



5-折交叉验证¹

¹https://scikit-learn.org/stable/modules/cross_validation.html

自助法 bootstrapping



Pull yourself up by your bootstraps(自力更生)



.632 自助法

自助法 (bootstrap)

不同于交叉验证中数据只有一次用作检验，自助法 (bootstrap) [2] 从给定的训练样本中**有放回**地均匀采样。其中最常用的是**.632 自助法**。对于一个包含 n 个样本的数据，.632 自助法将数据有放回地抽样 n 次，产生 n 个训练集。按照这种抽样方法，有 63.2% 的样本将出现在训练集中，38.8% 的样本将出现在验证集中。

数学原理：数据集中每个样本被选入训练集的概率为 $\frac{1}{N}$ ，选为验证集的概率为 $1 - \frac{1}{N}$ ，选择 N 次之后，一个样本被选入验证集的概率为 $(1 - \frac{1}{N})^N$ ，当 N 较大时，该概率近似为 $e^{-1} = 0.368$ 。



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)。

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

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-特效性 (specificity)。

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

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 ...



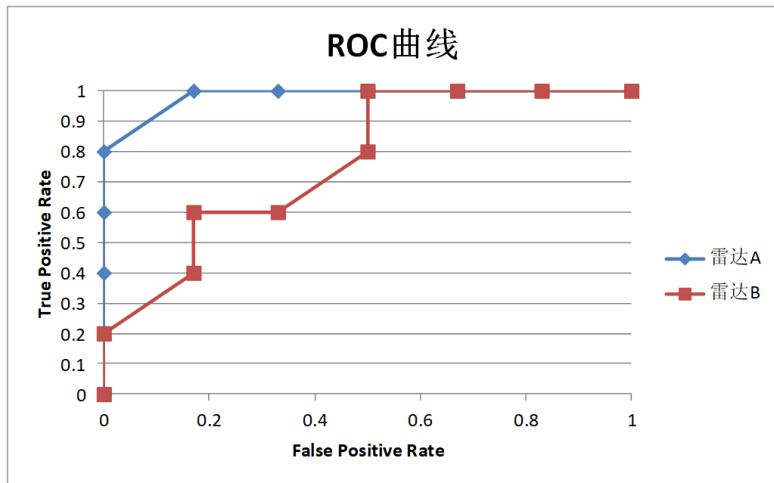
ROC 分析——以雷达检测为例

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A							
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B							
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



统计显著性检验

机器学习算法的对比一般都是简单的走以下几个步骤：

- ① 入门版：直接对比误分率 (misclassification rate)
- ② 初级版：对比 precision 和 recall 以及 F1-score
- ③ 高级版：比较 ROC 曲线下的面积

特殊情况：模型 1 在一个数据集 A 上表现突出，但模型 2 在 10 个数据集上都有不错的表现，但略逊于前者在 A 上的表现。这两个模型谁比较好？

两两对比效率低下，如果算法 A 比 B 好，B 比 C 好，C 比 D 好，那么是否 A 比 D 好？这种推论仅当其中每个过程都不可逆或者非常明显才可得。

统计显著性检验

- ① 显著性检验是用于检测科学实验中**实验组与对照组之间是否有差异以及差异是否显著**的办法 (学术论文中常用)
- ② 把要检验的假设称之为**原假设**，记为**H0**；把与 H0 相对应（相反）的假设称之为**备择假设**，记为**H1**。
- ③ 要证明模型 A 显著好于模型 B，做如下原假设：随机初始化一万次，模型 A 的表现**显著**优于模型 B。我们可以通过假设检验来接受或拒绝原假设。
- ④ **反证法**：先假定原假设正确，然后根据样本信息，基于“小概率事件不易发生”这一原理，观察由此假设而导致的结果是否合理，从而判断是否接受原假设。

统计显著性检验

假设检验的步骤为：

- ① 提出二择一的假设 H_0 （往往与试验目的相反）与 H_1 （往往是欲得到的结论）
- ② 给定显著水平（小概率）
- ③ 在 H_0 成立下，收集数据，寻找检验统计量
- ④ 将样本值和 H_0 代入检验统计量进行计算，将计算结果与临界值比较，若大于临界值，小概率事件发生，根据小概率原理，在一次试验中小概率事件是不会发生的。现在，居然发生了。错在哪里？
- ⑤ 原来是假设 H_0 错了，因为一切都是在 H_0 成立下推证的，于是拒绝 H_0 。否则，不拒绝 H_0 。

统计显著性检验

以大海捞针问题为例。我们现在要证明海里不只一根针，方法是：

H0：海里只有一根针

H1：海里不止一根针

显著水平 = 0.01（给定的小概率的定义，我们认为在大海里捞到针是不可能的）

进行试验：到大海里捞针

- ① 如果一百次潜水（试验）以内竟然就捞上了一根针，不可能的事发生了，于是拒绝 H0，认为海里不只一根针
- ② 多次试验均未捞到针，则接受原假设，认为海底只有一根针

统计假设检验的魅力就在于用统计学的方法证明一些难以证明的事

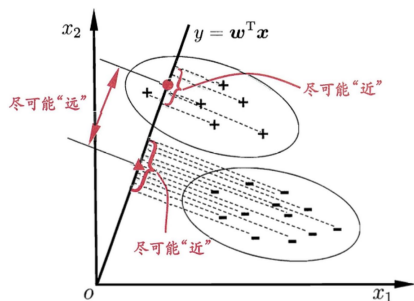
- 1 认识分类
- 2 分类模型的评估
 - 二分类模型的评估
 - 多分类模型的评估
 - 分类模型的选择
- 3 线性判别分析
 - 从 PCA 到 LDA
- 4 从分类到回归
 - 线性回归
- 5 决策树
 - 决策树原理
 - 决策树的选择



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。



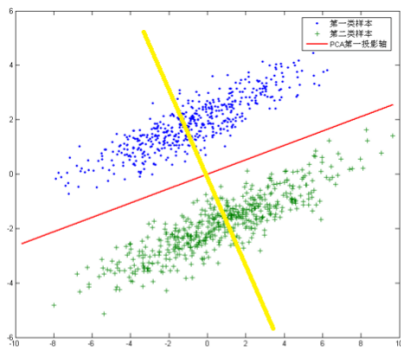
以二分类为例



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

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

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

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

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

注意到上式的分子和分母都是关于 w 的二次项，令 $w^T S_w w = 1$ ，则上式等价于：

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

利用拉格朗日乘子法，上式等价于：

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

其中 λ 为拉格朗日乘子。



线性判别分析 LDA——二分类

拉格朗日乘子法：

一种寻找多元函数在一组约束下的极值的方法

$$\begin{aligned} \min_x f(x) \quad s.t. \quad g(x) \\ \Rightarrow L(x, \lambda) = f(x) + \lambda g(x) \\ \Rightarrow \begin{cases} \nabla_x L(x, \lambda) = 0 \\ \nabla_\lambda L(x, \lambda) = 0 \end{cases} \\ \Rightarrow \begin{cases} \nabla f(x) + \lambda \nabla g(x) = 0 \\ g(x) = 0 \end{cases} \end{aligned}$$

其中 $L(x, \lambda)$ 为拉格朗日函数， λ 为拉格朗日乘子。



线性判别分析 LDA——二分类

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

S_w 为对称矩阵:

$$\begin{aligned} S_w^T &= \left(\sum_{i=0,1} [(x - \mu_i)(x - \mu_i)^T] \right)^T \\ &= \sum_{i=0,1} [(x - \mu_i)(x - \mu_i)^T]^T \\ &= \sum_{i=0,1} [(x - \mu_i)(x - \mu_i)^T] = S_w \end{aligned}$$

因此, S_w 为对称矩阵, 同理可证明 S_b 为对称矩阵。



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 (对称正定矩阵, 特征值全为正)
- 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

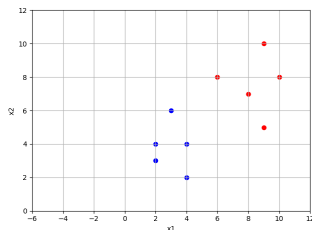
一个简单的例子：现有一个包含两个类的二维数据集 D ，需要将其投影到一条直线 w 上。

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:

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

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：

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

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

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

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

$$\Rightarrow \left| \begin{bmatrix} 9.2213 - \lambda & 6.489 \\ 4.2339 & 2.9794 - \lambda \end{bmatrix} \right|$$

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

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

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



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

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

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

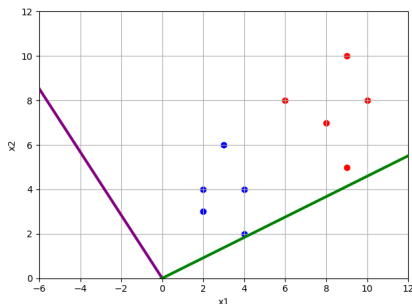


线性判别分析 LDA——例子

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

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

$\lambda = J(w)$ ，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：

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

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。



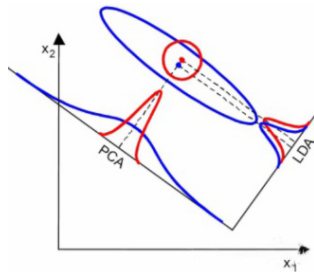
线性判别分析 LDA——优点与缺点

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：

- 协方差思想基于高斯分布假设，LDA 不适合对非高斯分布样本
- 类间散度矩阵的秩最多是 $K-1$ （全局均值减少了一个自由度），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
- 2 分类模型的评估
 - 二分类模型的评估
 - 多分类模型的评估
 - 分类模型的选择
- 3 线性判别分析
 - 从 PCA 到 LDA
- 4 从分类到回归
 - 线性判别分析 LDA
 - 多分类 LDA
 - 线性回归
- 5 决策树
 - 决策树原理
 - 决策树的选择



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

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

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：

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

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

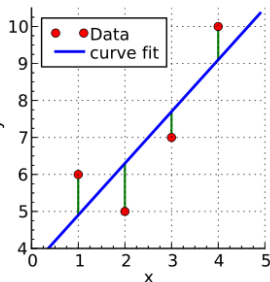


最小二乘法求解单元线性回归

令上述导数为 0，可得到线性回归的最优参数：

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

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



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（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：

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} & 1 \\ x_{21} & x_{22} & \dots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{Nd} & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1^T & 1 \\ \mathbf{x}_2^T & 1 \\ \vdots & \vdots \\ \mathbf{x}_N^T & 1 \end{pmatrix}$$



多元线性回归

最优参数向量 \mathbf{w}^* 的目标是最小化

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

对上式子求导：

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

令导数为 0，可得 \mathbf{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



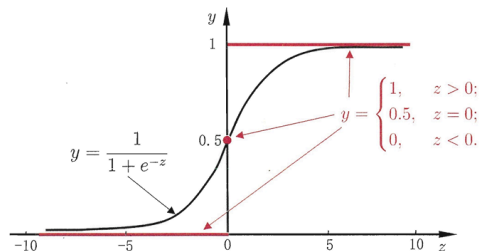
编号	色泽	根蒂	敲声	熟瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

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

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

回归函数可以通过简单的数学变换得到：

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

$\frac{y}{1-y}$ 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 $\beta = (\mathbf{w}, b)$ 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：

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



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函数之一。它具有如下性质：

- 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

但任然存在如下缺陷：

- 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
- 2 分类模型的评估
 - 二分类模型的评估
 - 多分类模型的评估
 - 分类模型的选择
- 3 线性判别分析
 - 从 PCA 到 LDA
- 4 从分类到回归
 - 线性回归
- 5 决策树
 - 决策树原理
 - 决策树的选择



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：

$$\oplus \leftarrow \mathbf{f}_1 \wedge \mathbf{f}_2 \wedge \cdots \wedge \mathbf{f}_L$$

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 Decision Tree

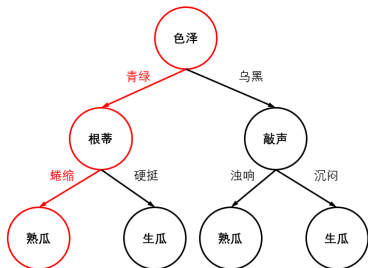


编号	色泽	根蒂	敲声	熟瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否
5



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

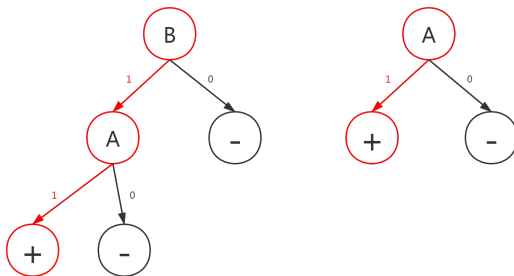
决策树是一种模拟人类决策过程的常用分类方法，通过递归选择属性维度将样本分配到对应的分支中，直到每个分支能尽可能把所有数据分开。

- 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_DecisionTree(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 $A==a$ 的测试)

if Examples(a) 为空:

创建一个叶节点 with 占大多数的标签

else:

Build_DecisionTree(Examples(a),Attributes-{A})



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。

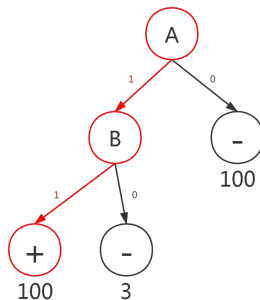
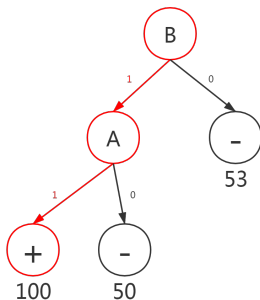
解决方案:

- 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，Entropy=0
- 2 如果两类样本数量相同，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，在第一次选择时

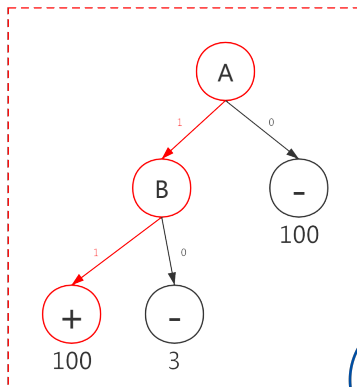
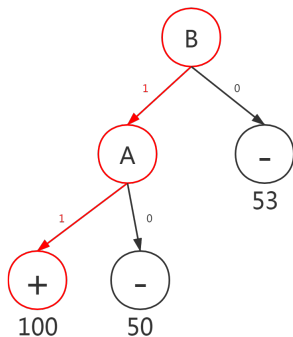
$$\begin{aligned}\text{Gain}(S, A) &= \text{Entropy}(S) - \sum_{a \in \{0,1\}} \frac{|S_a|}{|S|} \text{Entropy}(S_a) \\ &= - \sum_{i=1}^{\{+, -\}} P_i \log P_i - \sum_{a \in \{0,1\}} \frac{|S_a|}{|S|} \text{Entropy}(S_a) \\ &= 0.693 - \frac{103}{203} * \text{Entropy}(\{100+, 3-\}) - \frac{100}{203} *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

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



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：

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



要点总结

- ① 认识分类
 - 线性判别分析 LDA
 - 多分类 LDA
- ② 分类模型的评估
 - 二分类模型的评估
 - 多分类模型的评估
 - 分类模型的选择
- ③ 线性判别分析
 - 从 PCA 到 LDA
- ④ 从分类到回归
 - 线性回归
- ⑤ 决策树
 - 决策树原理
 - 决策树的选择



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.



KOHAVI, R., ET AL.

A study of cross-validation and bootstrap for accuracy estimation and model selection.

In *Ijcai* (1995), vol. 14, Montreal, Canada, pp. 1137–1145.

