

数据挖掘与应用

分类

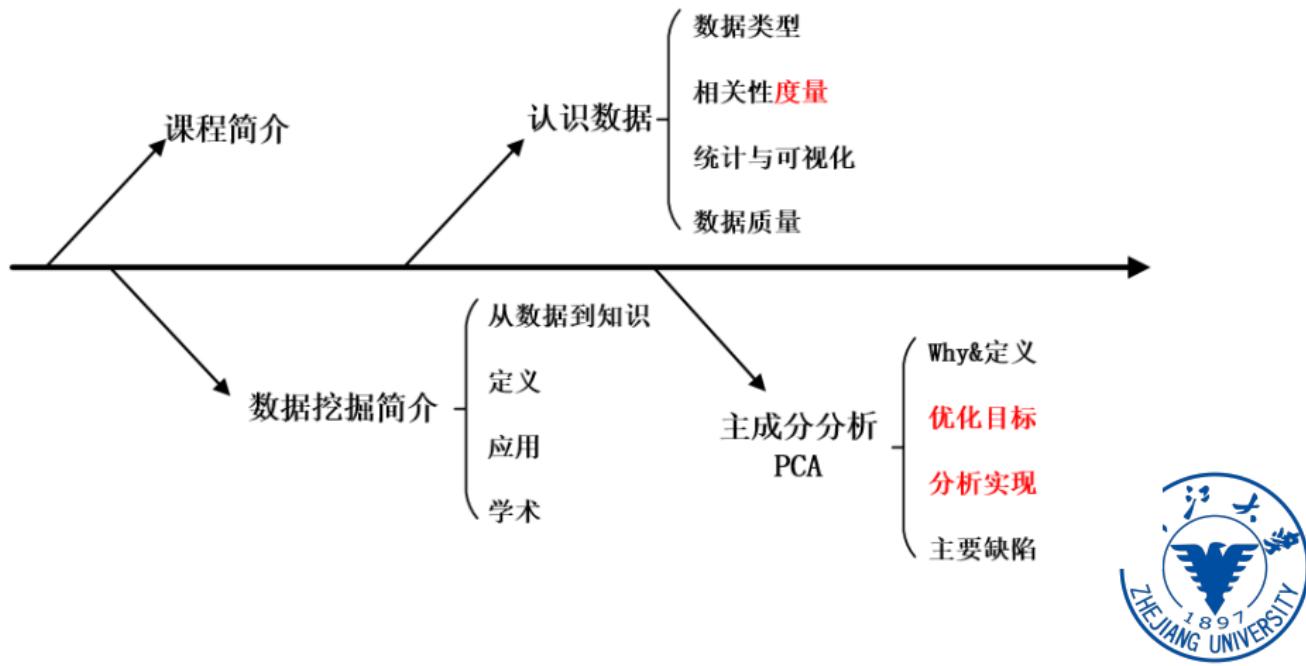
授课教师：周晟

浙江大学 软件学院

2021.09



上节课回顾

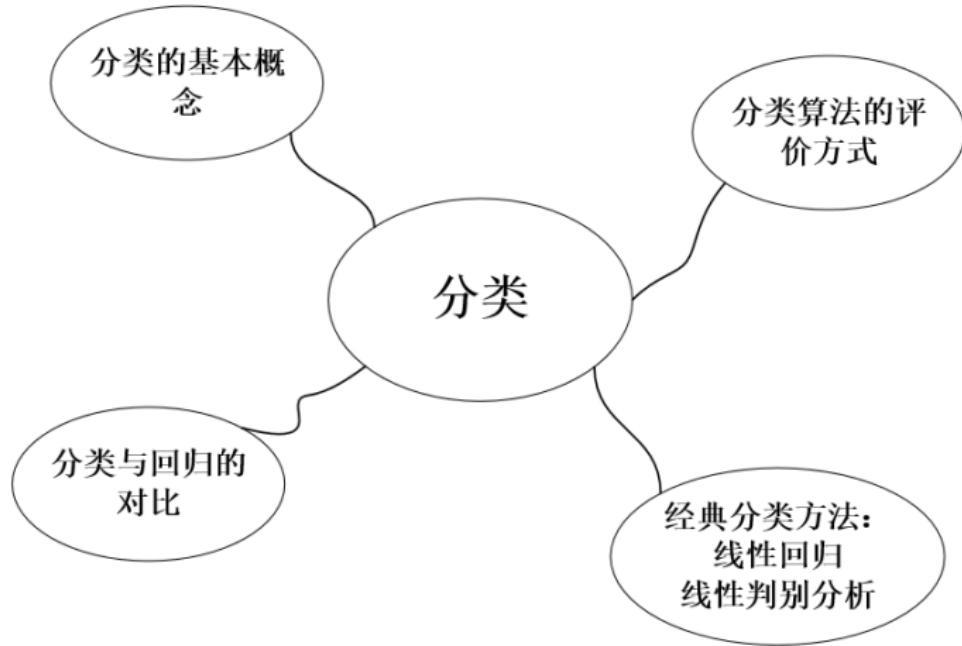


课程内容

- ① 认识分类
- ② 分类模型的性能评估
- ③ 分类模型选择
- ④ 决策树
- ⑤ 回归任务
- ⑥ 线性判别分析
- ⑦ 初探支持向量机



本节课程结构



① 认识分类

② 分类模型的性能评估

③ 分类模型选择

④ 决策树

⑤ 回归任务

⑥ 线性判别分析

⑦ 初探支持向量机



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 ...



分类和聚类

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



训练集



验证集

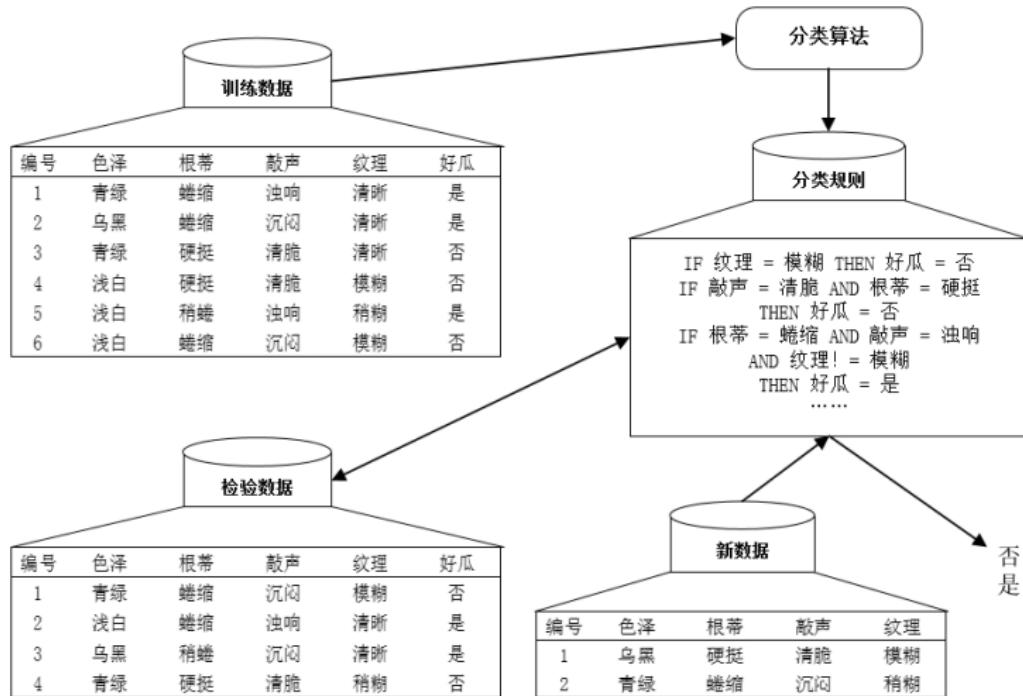
测试集



这是周杰伦 / 与其他两人不同



分类的基本步骤



① 认识分类

② 分类模型的性能评估

③ 分类模型选择

④ 决策树

⑤ 回归任务

⑥ 线性判别分析

⑦ 初探支持向量机



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。

混淆矩阵 (confusion matrix) 是分析分类器识别不同类样本的有效工具。

		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



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

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

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

图: 以猫狗分类为例



灵敏性和特效性

许多真实的分类问题中，用户感兴趣的类别样本存在**类别不均衡**问题。
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

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

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

图：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

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

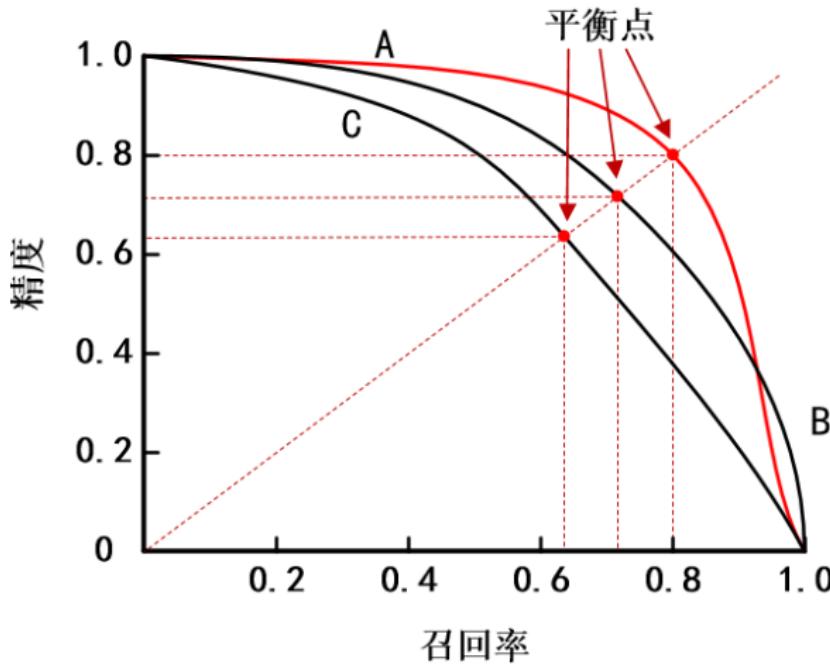
$$\text{recall} = \frac{TP}{TP + FN} = \frac{365}{365 + 5} = 0.986$$

图：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



图：准确率与召回率



精度和召回率

判案低精度的后果：六月飞雪窦娥冤



图：低精度 & 高召回的案例



精度与召回率的融合：F 度量

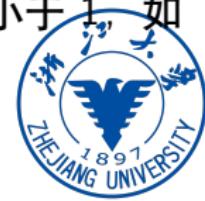
F 度量 (F-score) 是一种融合了精度和召回率的统一度量。

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

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

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

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

图：以刷单检测为例

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

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

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



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），所以每当有信号出现在雷达屏幕上，雷达兵就需要对其进行破译。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神经大条，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 的分析对象为二元分类模型，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。

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）。

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



ROC 曲线

ROC 空间里的单点，是给定分类模型且给定阈值后，通过计算坐标值 (FPR, TPR) 得出的。

然而，同一个二元分类模型的阈值可能设定为高或低，不同阈值的设定可能会得出不同的 FPR 和 TPR。

将同一模型每个阈值的 (FPR, TPR) 坐标都画在 ROC 空间里，就成为了特定模型的 ROC 曲线。

定量分析指标：Area under the Curve of ROC, AUC

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。



ROC 分析——以雷达检测为例

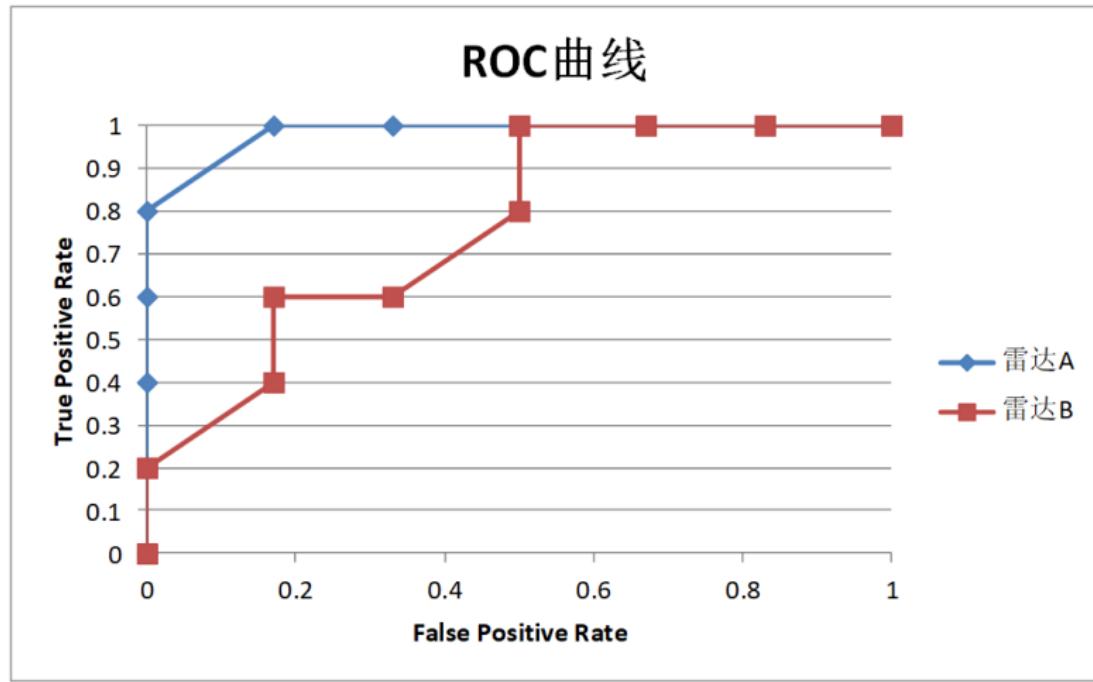
		雷达辨别概率	
不明物体	实际类别	分辨为飞机概率	
		雷达A	雷达B
1	飞机	0.9	0.75
2	鸟	0.5	0.55
3	飞机	0.75	0.65
4	鸟	0.4	0.45
5	鸟	0.1	0.3
6	飞机	0.98	0.8
7	飞机	0.6	0.5
8	鸟	0.6	0.75
9	鸟	0.3	0.4
10	鸟	0.55	0.6
11	飞机	0.7	0.55

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
- ② 分类模型的性能评估
- ③ **分类模型选择**
- ④ 决策树
- ⑤ 回归任务
- ⑥ 线性判别分析
- ⑦ 初探支持向量机



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？单数据集的模型性能校准：

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

多数据集的模型性能评估

- ① 显著性检验
- ② 接收机工作特征曲线



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。



.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) 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 $e^{-1} = 0.368$ 。



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 初学者直接对比误分率 (misclassification rate)
- ② 上过课的同学会对比 precision 和 recall 以及 F1-score
- ③ 要求高一点的同学会比较一下 ROC 曲线下的面积 (这个咱们待会再讲)

但是这样仍然无法科学地进行对比，比如模型 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？而且，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。
- ② 把要检验的假设称之为**原假设**, 记为 **H_0** ; 把与 H_0 相对应(相反)的假设称之为**备择假设**, 记为 **H_1** 。
- ③ 例如: 要证明模型 A 显著好于模型 B, 做如下假设: 随机初始化一万次, 模型 A 的表现**显著大于**模型 B。这是一个原假设, 我们可以通过假设检验来接受或拒绝原假设。
- ④ 假设检验采用的逻辑推理方法是反证法。先假定原假设正确, 然后根据样本信息, 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。而判断结果合理与否, 是基于“小概率事件**不易发生**”这一原理的。



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 找出小概率发生的临界值（一般是查分布表）。
- ⑤ 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？
- ⑥ 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：

H_0 : 海里只有一根针

H_1 : 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 H_0 ，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 分类模型的性能评估

③ 分类模型选择

④ 决策树

⑤ 回归任务

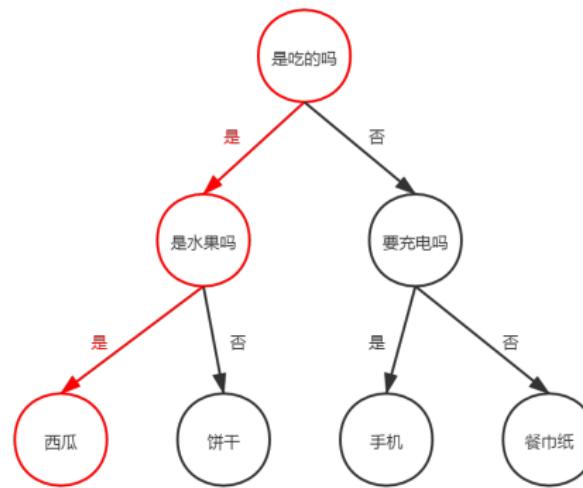
⑥ 线性判别分析

⑦ 初探支持向量机



决策树

回想一个经典游戏，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，每次游戏就相当于一棵以提问次数为深度的二叉树从根节点到叶子节点的一条路径。



决策树

- ① 类似地，决策树是一种十分常用的分类方法，适用于分类由特征向量表示的数据。
- ② 决策树的输入为样本的特征向量，输出为离散的类别属性。（输出也可以是连续的值，该模型称为回归树）
- ③ 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）**。
- ④ 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。



决策树的学习阶段-递归伪代码

Algorithm

```
Build_DecimalTree(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_DecimalTree(Examples(a),Attributes-{A})
```



决策树的递归中断

决策树的生成是一个递归过程。在决策树基本算法中，有三种情形会导致递归返回：

- 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。



决策树的学习阶段

对于同一批数据，我们可以建多少不同的树？最坏情况和最好情况呢？

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”

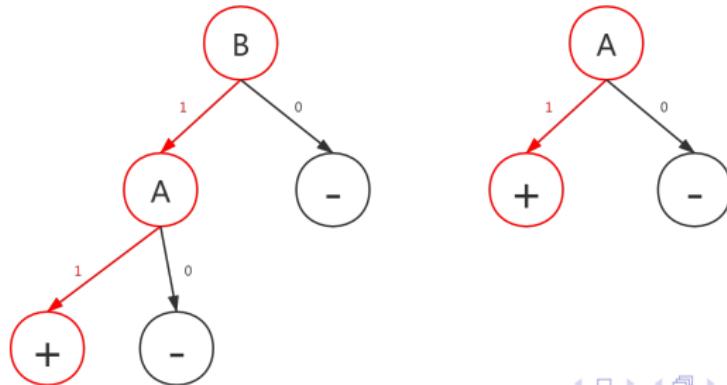


决策树的学习阶段

考虑如下例子：

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

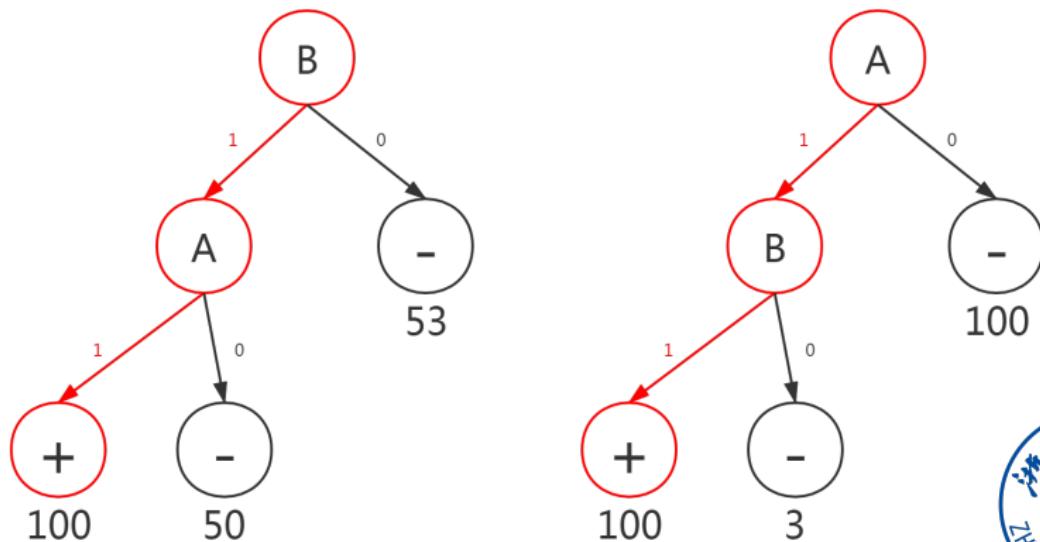
做一点点修改：

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	+



决策树的学习阶段

显然，决策树仍然有两棵，它们深度一致，树的结构也一致，如何比较它们呢？



熵 (Entropy)

我们先来了解熵的概念。对于刚才的二分类任务而言，熵的计算公式如下：

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

其中 P_+ 代表正样本在总体 S 中出现的概率

P_- 代表负样本在总体 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，随机分类的概率为 0.5， $\text{Entropy}=1$

可以看出，熵可以代表样本的**纯净程度**。熵的一般形式如下：

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

其中 P_i 代表样本是第 i 类的概率



熵 (Entropy)

- 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 熵的计算使我们知道样本中标签分布是否均匀，熵越小，样本标签越纯净；熵越大，样本标签分布越平均。
- ③ 如此一来，**选择所有分支熵最小的属性**自然成为决策树建树的策略。
- ④ 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

- ① 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
属性 A 的信息增益是由该属性产生分支而减少的熵
信息增益的计算公式如下:

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

- where S_a is the subset of S for which attribute A has value a
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

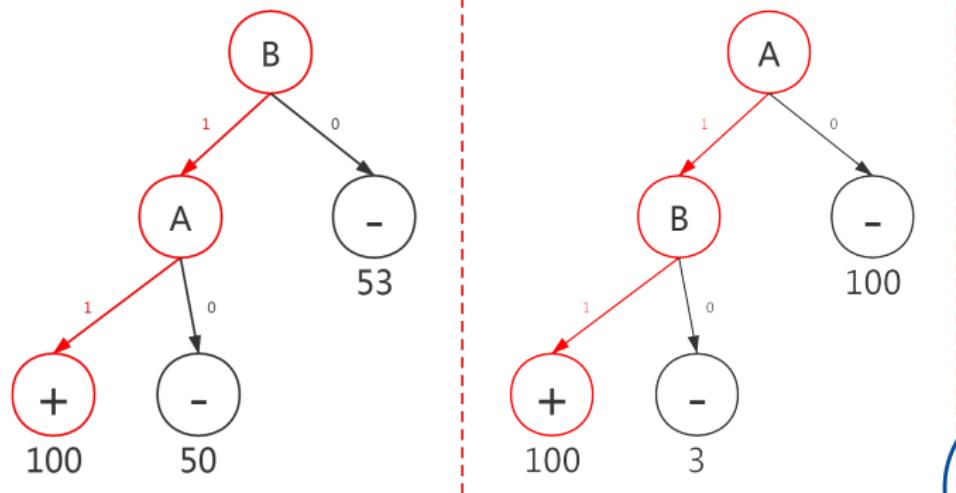
$$\begin{aligned}
 \text{Gain}(S, A) &= \text{Entropy}(S) - \sum_{a \in \{0,1\}} \frac{|S_a|}{|S|} \text{Entropy}(S_a) \\
 &= - \sum_{i=1}^{\{+,-\}} P_i \log P_i - \sum_{a \in \{0,1\}} \frac{|S_a|}{|S|} \text{Entropy}(S_a) \\
 &= 0.693 - \frac{103}{203} * \text{Entropy}(\{100+, 3-\}) - \frac{100}{203}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

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



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
- ② ID3 后面的 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



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：

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



① 认识分类

② 分类模型的性能评估

③ 分类模型选择

④ 决策树

⑤ 回归任务

⑥ 线性判别分析

⑦ 初探支持向量机



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



互动环节

在特征空间中，好的分类器应该是怎么样的？



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：

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

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

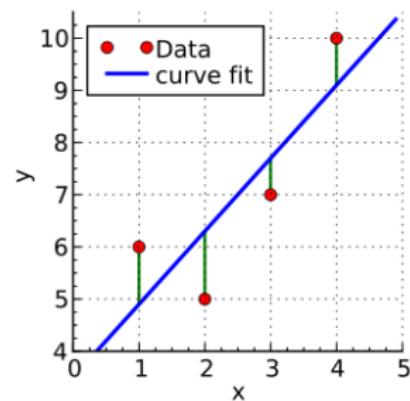


最小二乘法求解单元线性回归

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)

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

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：

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



对数几率回归



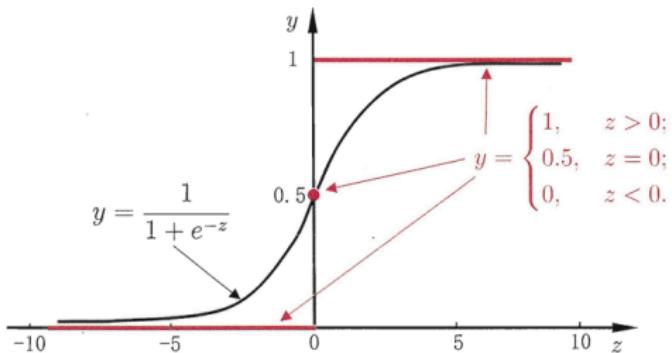
编号	色泽	根蒂	敲声	熟瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

吃过的瓜



对数几率回归

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 $y \in \{0, 1\}$ 。



对数几率函数

单位阶跃函数：

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

对数几率函数：

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



对数几率函数

使用对数几率函数作为回归的函数，

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

y 表示样本为正例的概率， $1 - y$ 表示样本为负例的概率

$$\begin{aligned} y &= p(y = 1|x) \\ 1 - y &= p(y = 0|x) \end{aligned}$$

$\frac{y}{1-y}$ 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)



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 任意阶可导，可以用数值优化算法**快速**求最优解；
- ② 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）
- ③ 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；
- ④ 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



对数几率回归

根据对数几率函数，分类模型对样本的标签估计为：

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

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

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：

$$\ell(\beta) = \sum_{i=1}^m \ln p(y_i \mid \mathbf{x}_i; \beta)$$



① 认识分类

② 分类模型的性能评估

③ 分类模型选择

④ 决策树

⑤ 回归任务

⑥ 线性判别分析

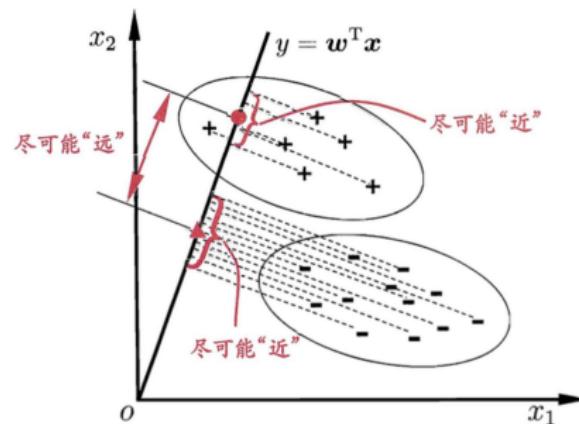
⑦ 初探支持向量机



线性判别分析 LDA

线性判别分析（Linear Discriminant Analysis, LDA）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。



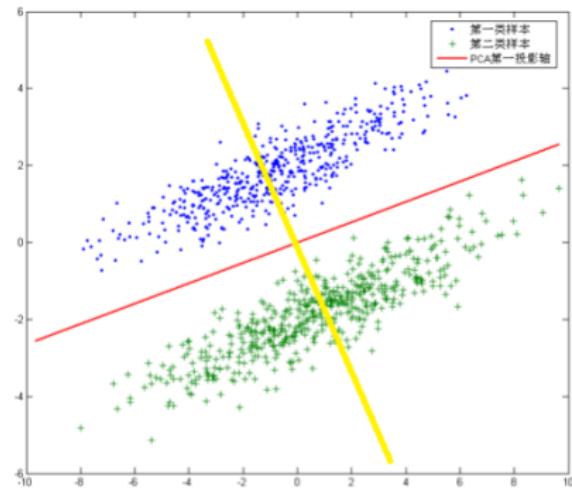
图：以二分类为例



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。



图：PCA 与 LDA



线性判别分析 LDA

我们先以二分类为例，分析理解 LDA 的优化目标和推导。

符号：

- x : 数据样本，用列向量表示
- x_i^j : 第 i 类中的第 j 个样本 ($i = 0, 1$)
- N_i : 第 i 类样本的数目
- N : 样本的总数目 $N = N_0 + N_1$
- μ_i : 第 i 类样本的均值向量 $\mu_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_i^j$
- μ : 所有样本的均值向量 $\mu = \frac{1}{N} \sum_{i=1}^N x_i$
- w : 特征向量
- D_i : 第 i 类样本集合
- S_w : 类内散度矩阵
- S_b : 类间散度矩阵 \sum_i : 投影前第 i 类的协方差矩阵

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 为特征向量，就是我们想要的投影方向。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

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 $w^T \sum_1 w$

我们想要让同类样本的投影点尽可能接近, 可以使同类样本点的协方差矩阵尽可能小, 即 $w^T \sum_0 w + w^T \sum_1 w$ 尽可能小; 而想要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。同时考虑两者, 则可得到想要最大化的目标:

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

$\|\cdot\|$ 表示欧几里得范数, $\|a - b\|_2^2 = (a - b)^T (a - b)$



线性判别分析 LDA——二分类

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

我们继续定义类间散度矩阵 (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

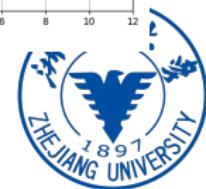
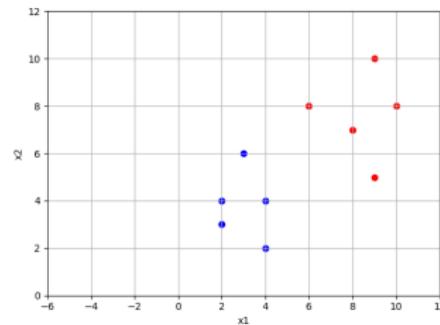
一个简单的例子：现有一个包含两个类的二维数据集 D ，需要将其投影到一条直线 w 上。

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：

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

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

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

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

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

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

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

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



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

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

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

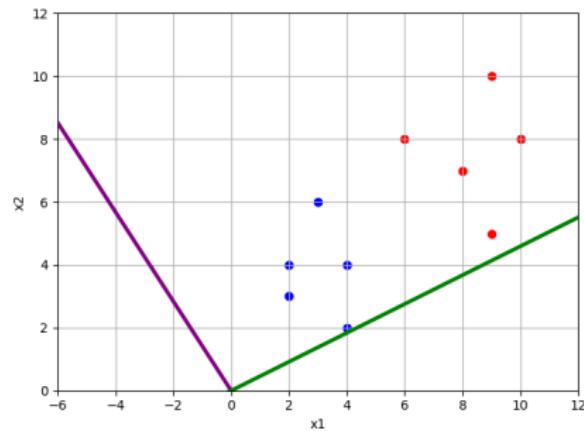


线性判别分析 LDA——例子

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

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

$\lambda = J(w)$, 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

符号：

- x : 数据样本，用列向量表示
- x_i^j : 第 i 类中的第 j 个样本
- K : 共有 K 类样本
- N_i : 第 i 类样本的数目
- N : 样本的总数目 $N = \sum_{i=1}^K N_i$
- μ_i : 第 i 类样本的均值向量 $\mu_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_i^j$
- μ : 所有样本的均值向量 $\mu = \frac{1}{N} \sum_{i=1}^N x_i$
- w : 特征向量 or 多个特征向量构成的矩阵
- D_i : 第 i 类样本集合
- S_w : 类内散度矩阵
- S_b : 类间散度矩阵 \sum_i : 投影前第 i 类的协方差矩阵

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。



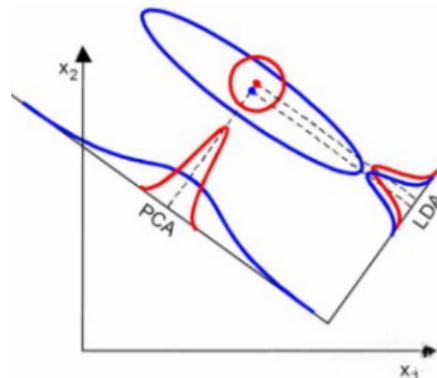
线性判别分析 LDA——优点与缺点

主要优点：

- 在降维过程中可以使用类别标签的先验知识
- 在 $J(w)$ 更依赖于均值时，性能相比 PCA 之类的算法较优

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

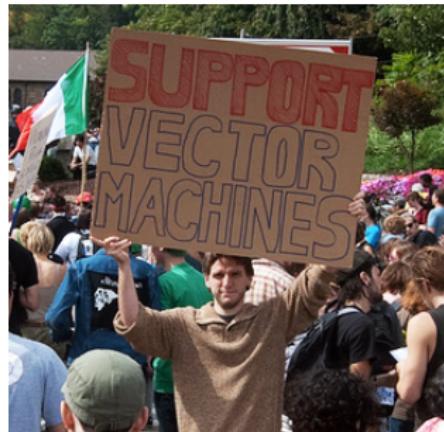


- 1 认识分类
- 2 分类模型的性能评估
- 3 分类模型选择
- 4 决策树
- 5 回归任务
- 6 线性判别分析
- 7 初探支持向量机



SVM

SVM 一直被认为是效果最好的现成可用的分类算法之一（其实有很多人都相信，“之一”是可以去掉的）。



一直以来学术界和工业界甚至只是学术界里做理论的和做应用的之间，都有一种“鸿沟”。而 SVM 则正好是一个特例——在两边都混得开。

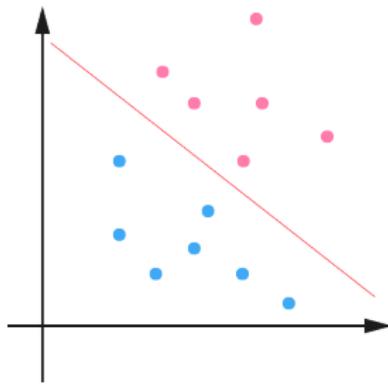


分类与超平面

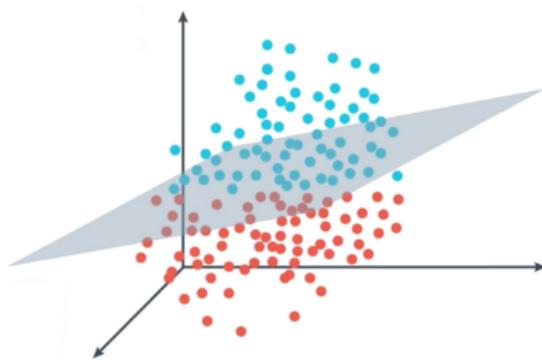
一个 n 维特征空间中的线性分类器就是要在特征空间中找到一个超平面，其方程可以表示为：

$$w^T x + b = 0$$

理想的超平面是在特征空间中将两类数据分隔开，即两类数据分别分布在超平面的两侧（虽然这种条件不一定可以满足）。



二维空间超平面



三维空间超平面

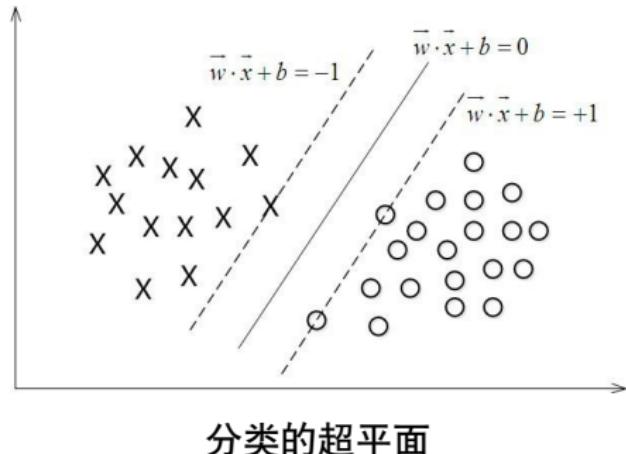


分类与超平面

为了计算方便，首先将类别信息数值化。

$$f(x) = w^T x + b \begin{cases} > 0 & y = 1 \\ = 0 & \text{超平面} \\ < 0 & y = -1 \end{cases}$$

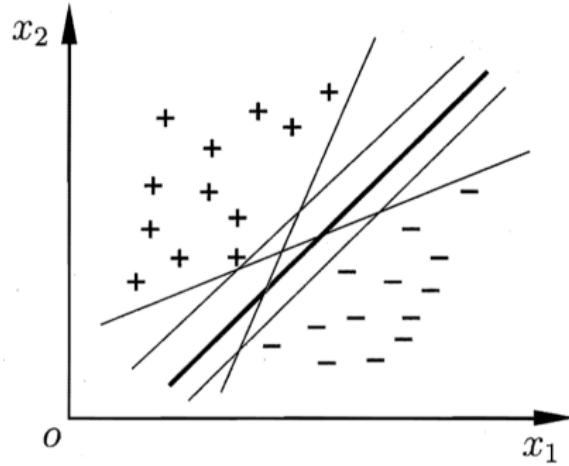
这是一种理想情况，真实的分类器往往难以找到这种超平面。



对于一个数据点 x 进行分类，实际上是通过把 x 带入到 $f(x) = w^T x + b$ 算出结果然后根据其正负号来进行类别划分的。



分类与超平面



用于分类的超平面

能够讲两个类分开的超平面有很多，每个超平面对应一个分类器。但是
不同分类器的鲁棒性却不同。



函数间隔与几何间隔

函数间隔

函数间隔 (functional margin) 定义为：

$$\hat{\gamma} = y (w^T x + b) = y f(x)$$

因为负类的标签 $y = -1$ ，因此函数间隔具有非负性。

几何间隔

几何间隔定义为点到超平面的距离。



函数间隔与几何间隔

定义样本 x 在超平面上的投影为 x_0 , w 是垂直于超平面的向量, 易得:

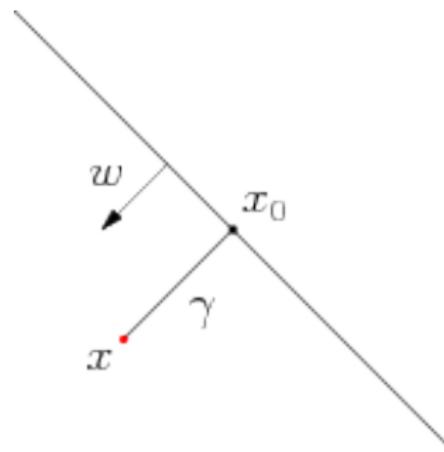
$$x = x_0 + \gamma \frac{w}{\|w\|}$$

由于 x_0 是超平面上的点, 满足 $f(x_0) = 0$, 易得:

$$\gamma = \frac{w^T x + b}{\|w\|} = \frac{f(x)}{\|w\|}$$

因此, 函数间隔与几何间隔满足关系:

$$\tilde{\gamma} = y\gamma = \frac{\hat{\gamma}}{\|w\|}$$



函数间隔与几何间隔

分类的最优间隔

对一个数据点进行分类，当它的间隔越大的时候，分类的置信度越大。对于一个包含 n 个点的数据集，我们可以很自然地定义它的间隔为所有这点的间隔值中最小的那个。为了使得分类的置信度高，我们希望所选择的超平面能够最大化这个间隔值。

函数间隔的缺陷

函数间隔可以在超平面不变的情况下被取得任意大，而集合间隔因为有对 $\|w\|$ 的缩放则没有这个问题，因此分类问题转化为最大化几何间隔：

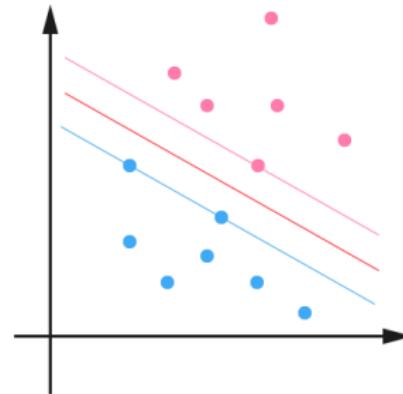
$$\begin{aligned} & \max \tilde{\gamma} \\ & y_i (w^T x_i + b) = \hat{\gamma}_i \geq \hat{\gamma}, \quad i = 1, \dots, n \end{aligned}$$

最优超平面

为了计算方便，令函数间隔为
1，目标函数转变为：

$$\max \frac{1}{\|w\|}$$

$$\text{s.t. } y_i (w^T x_i + b) \geq 1, i = 1, \dots, n$$



要点总结

① 认识分类

② 分类模型的性能评估

- 二分类模型
- 分类器性能评价指标
- 多分类

③ 分类模型选择

④ 决策树

⑤ 回归任务

- 线性回归

⑥ 线性判别分析

⑦ 初探支持向量机



参考文献

Blog

- ① SVM: <https://blog.pluskid.org/?p=632>

