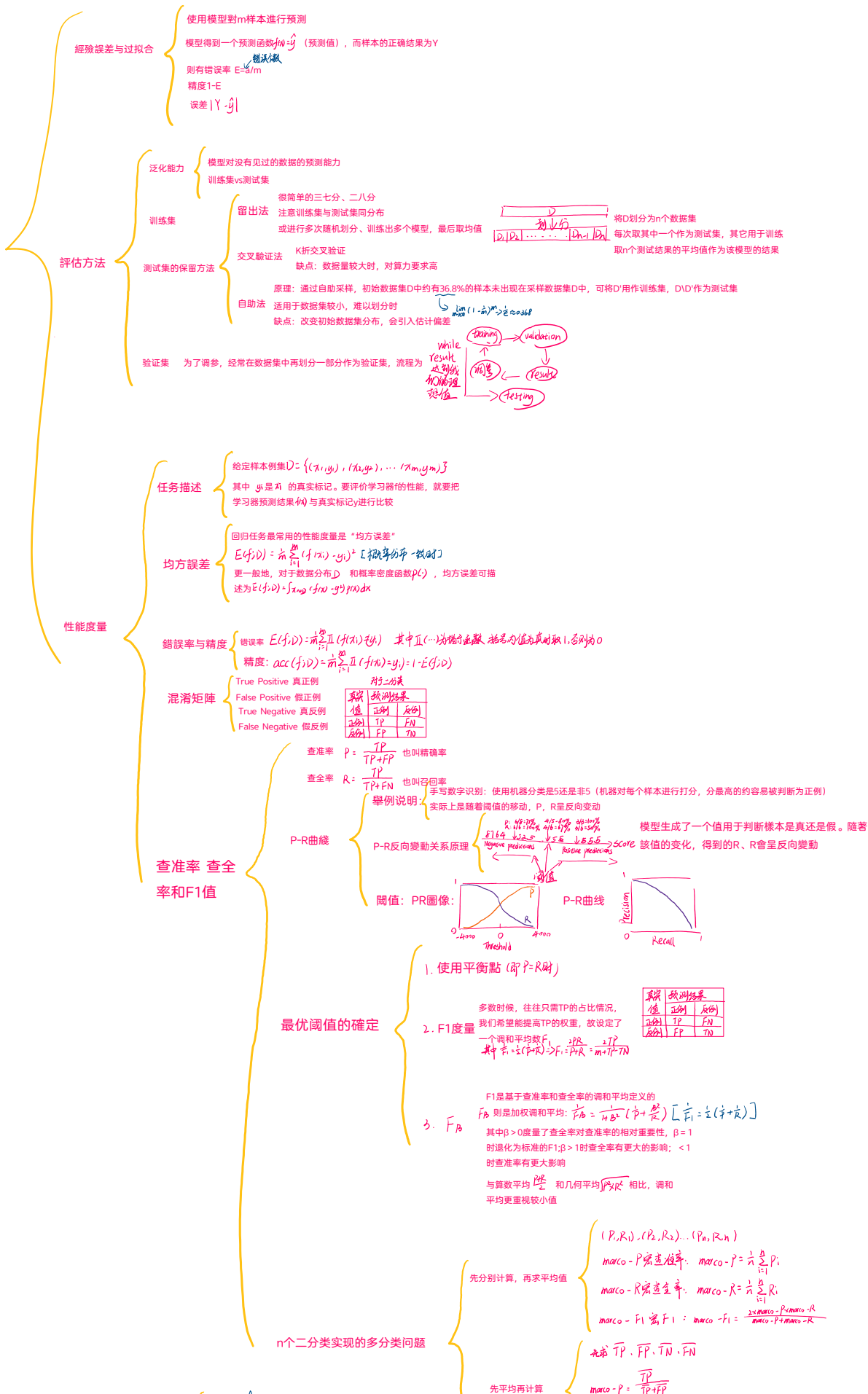


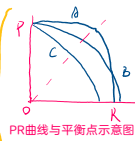
# 模型评估与选择

## 一種訓練集一種算法



一种训练多种算法

P-R曲线



n个二分类实现的多分类问题

先平均再计算

先求 TP, FP, TN, FN

Macro-P = (TP+FP)/n  
Macro-R = (TP+FN)/n  
Macro-F1 = (2\*Macro-P\*Macro-R)/(Macro-P+Macro-R)

首先可以确定B和A优于C, BA之间有交叉, 无法确定

比较ABC 3个模型的好坏

对比AB

- 方法1: 比较AB的面积大小, 在一定程度上表征了模型的优劣, 但是这个值不容易估算
- 方法2: F1
- 方法3: Fβ

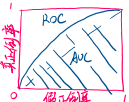
ROC:

True Positive Rate: TPR = TP/(TP+FN)  
False Positive Rate: FPR = FP/(FP+TN)

横着比

真实值 \ 预测值	正例	反例
正例	TP	FN
反例	FP	TN

示意图



ROC与AUC

AUC: Area Under ROC Curve 即ROC曲线下的面积

AUC = 1/2 \* sum\_{i=1}^m (x\_{i+1} - x\_i) \* (y\_{i+1} + y\_i)

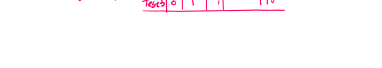
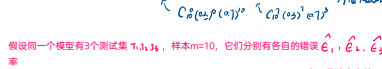
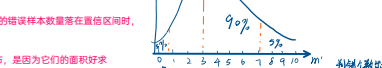
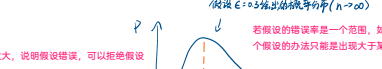
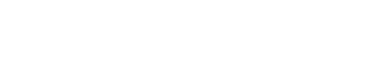
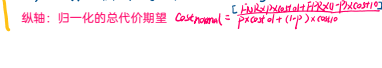
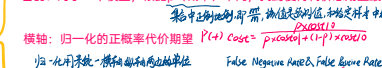
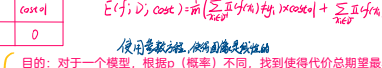
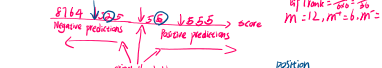
AUC

排序损失

形式化地看, AUC考虑的是样本预测的排序质量, 因此它与排序误差有紧密联系。给定m个正例和m个反例, 令D+表示正例集合, D-表示反例集合

则排序损失为 rank = 1/(m^2) \* sum\_{x+ in D+} sum\_{x- in D-} (I(f(x+) < f(x-)) + I(f(x-) < f(x+)))

以手写数字为例 [机器对样本进行排序, 越靠左值越小, 越靠右值越大] 总共有4个数字



多种训练集一种算法

代价敏感错误率与代价曲线

偏差与方差

二分类代价矩阵

真实类别 \ 预测类别	第1类
第0类	0 cost=1
第1类	cost=0 0

代价敏感错误率为

E(f; D; Cost) = sum\_{x in D} (sum\_{y in Y} I(f(x) != y) \* cost(x, y))

使用函数编程, 使用向量表示法

目的: 对于一个模型, 根据p(概率)不同, 找到使得代价期望最小的模型的阈值

阈值: 归一化的正概率代价期望

P(+|Cost) = p \* cost(1) + (1-p) \* cost(0)

注: 归一化阈值 = 模型输出结果的期望

纵轴: 归一化的总代价期望

CostNormal = (p \* cost(1) + (1-p) \* cost(0)) / (p \* cost(1) + (1-p) \* cost(0))

横轴: 归一化的正概率代价期望

P(+|Cost) = p \* cost(1) + (1-p) \* cost(0)

纵轴: 归一化的总代价期望

CostNormal = (p \* cost(1) + (1-p) \* cost(0)) / (p \* cost(1) + (1-p) \* cost(0))

横轴: 归一化的正概率代价期望

P(+|Cost) = p \* cost(1) + (1-p) \* cost(0)

纵轴: 归一化的总代价期望

CostNormal = (p \* cost(1) + (1-p) \* cost(0)) / (p \* cost(1) + (1-p) \* cost(0))

横轴: 归一化的正概率代价期望

P(+|Cost) = p \* cost(1) + (1-p) \* cost(0)

纵轴: 归一化的总代价期望

CostNormal = (p \* cost(1) + (1-p) \* cost(0)) / (p \* cost(1) + (1-p) \* cost(0))

横轴: 归一化的正概率代价期望

P(+|Cost) = p \* cost(1) + (1-p) \* cost(0)

纵轴: 归一化的总代价期望

CostNormal = (p \* cost(1) + (1-p) \* cost(0)) / (p \* cost(1) + (1-p) \* cost(0))

横轴: 归一化的正概率代价期望

P(+|Cost) = p \* cost(1) + (1-p) \* cost(0)

纵轴: 归一化的总代价期望

CostNormal = (p \* cost(1) + (1-p) \* cost(0)) / (p \* cost(1) + (1-p) \* cost(0))

横轴: 归一化的正概率代价期望

P(+|Cost) = p \* cost(1) + (1-p) \* cost(0)

纵轴: 归一化的总代价期望

CostNormal = (p \* cost(1) + (1-p) \* cost(0)) / (p \* cost(1) + (1-p) \* cost(0))

横轴: 归一化的正概率代价期望

P(+|Cost) = p \* cost(1) + (1-p) \* cost(0)

纵轴: 归一化的总代价期望

CostNormal = (p \* cost(1) + (1-p) \* cost(0)) / (p \* cost(1) + (1-p) \* cost(0))

横轴: 归一化的正概率代价期望

P(+|Cost) = p \* cost(1) + (1-p) \* cost(0)

纵轴: 归一化的总代价期望

CostNormal = (p \* cost(1) + (1-p) \* cost(0)) / (p \* cost(1) + (1-p) \* cost(0))

横轴: 归一化的正概率代价期望

P(+|Cost) = p \* cost(1) + (1-p) \* cost(0)

测试集上的性能在多大程度上保证真实的性能比较检验

问题

- 1. 测试集上的性能与真正的泛化性能未必相同
- 2. 测试集不同反映出来的性能不同
- 3. 机器学习算法本身具有随机性, 同一个测试集上多次运行, 可能会有不同的结果

一个测试集一种算法

二项分布

假设测试集有m个样本, 其中判断错误的有m'个, 可知错误率C = m'/m  
在所有的真实数据中, 假设其错误率为E, 则测试集中m=10, 则m'=?

常用的离散型随机变量

(1) 0-1随机分布

若随机变量X只有两个可能的取值0和1, 其概率分布为

P(X=x) = p^x \* (1-p)^(1-x), x=0, 1

则称X服从0-1分布

(2) 二项分布

设事件A在任意一次实验中出现的概率为p (0<p<1), X表示n重伯努利试验中发生的次数, 则X的所有可能取值为0, 1, 2, 3, ..., n, 且相应概率为

P(X=k) = C(n, k) \* p^k \* (1-p)^(n-k), k=0, 1, 2, ..., n

假设检验

从测试集中找出m'的个数, 若与图像出入太大, 说明假设错误, 可以拒绝假设

若与图像接近, 则不能拒绝假设

具体的, 若置信区间为90%, 则当测试集中测得的错误样本数落在置信区间时, 则无法拒绝假设

之所以要回归到二项分布、正态分布等标准分布, 是因为它们的面积好求

假设E=0.3输出错误率分布(n=100)

若假设的错误率是一个范围, 如E <= 0.5, 对于这个假设, 实际只设了上限, 拒绝这个假设的办法只能是出现大于某个数的错误

概率累加和

sum\_{i=0}^k P(X=i)

置信区间90%

拒绝假设输出错误率分布

C(n, k) \* p^k \* (1-p)^(n-k)

假设同一个模型有3个测试集T1, T2, T3, 样本m=10, 它们分别有各自的错误率E1, E2, E3, 对它们进行排列组合

率, 则它们每个的概率分布都是二项分布

各自错误样本数

	T0x1	0	0	0	10
T0x2	0	0	1	1	10
T0x3	0	1	1	1	10

由于是独立事件, 每个之间都是用乘法求的概率

得到的错误率E与方差sigma^2共有11^3个, 重复的需要合并, 得到新的概率分布

又回归到1个测试集1种算法的问题

多个测试集一种算法

t检验

高维型随机变量的函数分布

卡方分布

t分布

正态分布与处理为分布

交叉验证t检验

McNemar检验

Friedman与Nemenyi后续检验