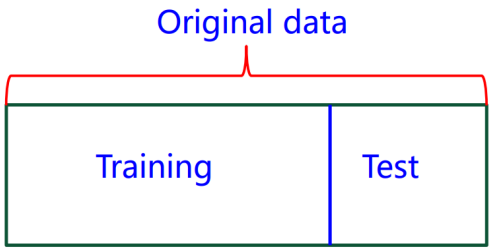
12Model Selection模型选择

Model Evaluation模型评估

1.Hold-out Method 留出法

直接将数据集D划分成两个互斥的集合，其中一个为训练集S，另一个作为测试集T，这称为**“留出法”**。

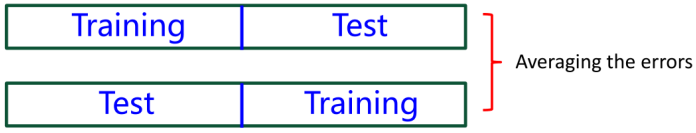


- *分层采样(stratified sampling)*： 在对数据集进行划分的时候，保留类别比例的采样方式称为**“分层采样”**。若对数据集D（包含500个正例，500个反例）则分层采样的到的训练集S（70%）应为350个正例，350个反例，测试集(30%)应为150个正例，150个反例。
- 限制：可能会有很大的方差，评估结果很大程度取决于那些数据在训练集，那些数据在测试集
- 很少的被标记的样本被训练，而是拿来测试

2.Cross-Validation 交叉验证（k折交叉验证）

假设我们把数据分成两个大小相等的子集，选择一个子集来训练，另一个来测试，然后交换，

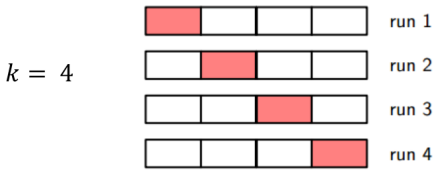
总误差是两次训练的误差的平均



更加普遍的是：k折交叉验证

将数据分成大小相等的k分，每次选择一份来测试，其他用来训练，重复这个过程

总误差是所有误差的平均



3.Boosting 自助法

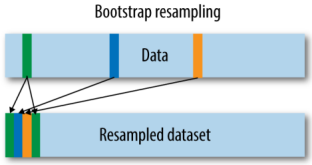
在数据中采样抽取作为训练集，每次抽取替换训练集

如果原始数据有n条数据，那么一个boostrap样本的大小大概是63.2%，一个数据被选成样本的可能性是0.632

未包含的部分作为测试集

boostrap可以从原始数据中得到很多不同的训练集，这将有利于集成学习

由于boostrap会改变训练集的数据分布，这就会有额外的误差。如果我们有足够的数据，那么holdout和交叉验证会得到广泛的应用



混淆矩阵confusion matrix

TP FP TN FN

	Predicted Class		
Actual Class		Class = YES	Class = No
	Class = Yes	TP True Positive	FN False Negative
	Class = No	FP False Positive	TN True Negative

Accuracy=(TP+TN)/(TP+FP+TN+FN)

准确性的限制：

1. 更高的准确率不意味着在目标任务上有更好的表现
2. 包含隐形的假设：样本之间的类的分布相对平衡

Precision （查准率/准确率） $\frac{TP}{TP+FP}$

Recall （查全率/召回率） $\frac{TP}{TP+FN}$

Recall ↑ → precision ↓ , Precision ↑ → recall ↓

P-R曲线

F_1-measure

现在假设你有两个分类器——分类器A和分类器B。

•一个有更好的回忆分数，另一个有更好的准确性。我们想谈谈它们的相对表现。

•我们希望将模型的性能总结为单个指标。

$F_1 = \frac{2*presion*recall}{precision*recall}$

$\frac{1}{F_1} = \frac{1}{2}(\frac{1}{presion} + \frac{1}{recall})$

$$\mathrm{F1} = \frac{2 \times \mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}} = \frac{2 \times \mathrm{TP}}{\text{样例总数} + \mathrm{TP} - \mathrm{TN}} = \frac{2 \times \mathrm{TP}}{2 \times \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

- More general:

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

- $\beta > 1$, recall is more important
- $\beta < 1$, precision is more important

我们可能有多个混淆矩阵

1. 重复训练很多次，
2. 在多个训练集上训练
3. 多任务，多分类，多标签

We use the following standard evaluation metrics [37] to measure the performance of all the methods.

- **Micro-$\mathbf{F_1}$** is a conventional metric used to evaluate classification decisions [37], [19]. Let TP_t, FP_t, FN_t denote the true-positives, false-positives and false-negatives for the class-label $t \in T$. The micro-averaged F_1 is

微查准率 $$P = \frac{\Sigma_{t \in T} TP_t}{\Sigma_{t \in T} TP_t + FP_t}$$

微查全率 $$R = \frac{\Sigma_{t \in T} TP_t}{\Sigma_{t \in T} TP_t + FN_t}$$

微F1 $$Micro\text{-}F_1 = \frac{2PR}{P+R}$$

- **Macro-$\mathbf{F_1}$** is also conventional metric used to evaluate classification decisions; unlike Micro-F_1 which gives equal weight to all instances in the averaging process, Macro-F_1 gives equal weight to each class-label.

宏查准率 $$P_t = \frac{TP_t}{TP_t + FP_t}$$

宏查全率 $$R_t = \frac{TP_t}{TP_t + FN_t}$$

宏F1 $$Macro\text{-}F_1 = \frac{1}{|T|}\sum_{t \in T} \frac{2P_tR_t}{P_t + R_t}$$

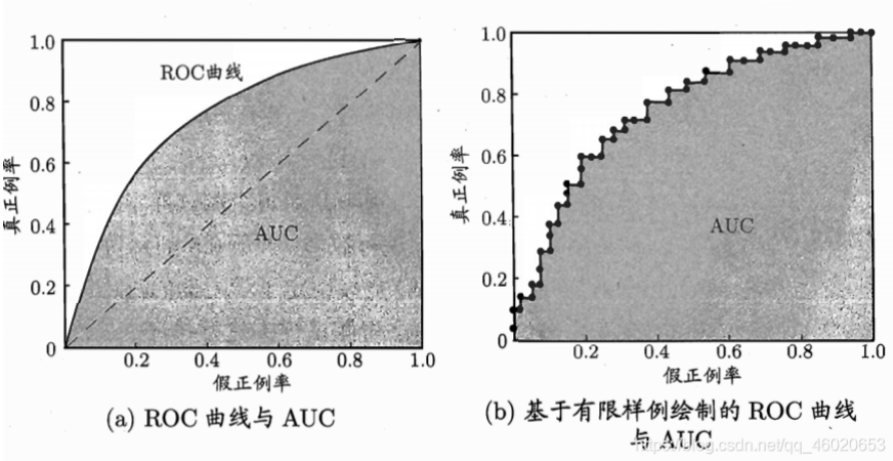
True Positive Rate(TPR=$\frac{TP}{TP+FN}=\frac{Tp}{p}$)

Fales Positive Rate(FPR=$\frac{FP}{TN+FP}=\frac{FP}{N}$)

TPR：TPR越大意味着TP越大，也就意味着对于测试样本中的所有正例来说，其中大部分都被学习器预测正确。
FPR：FPR越小意味着FP越小、TN越大，也就意味着FPR越小，则对于测试样例中的所有反例来说，其中大部分被学习器预测正确。
由上面可以看出，一个好的模型是TPR大PFR偏小的。

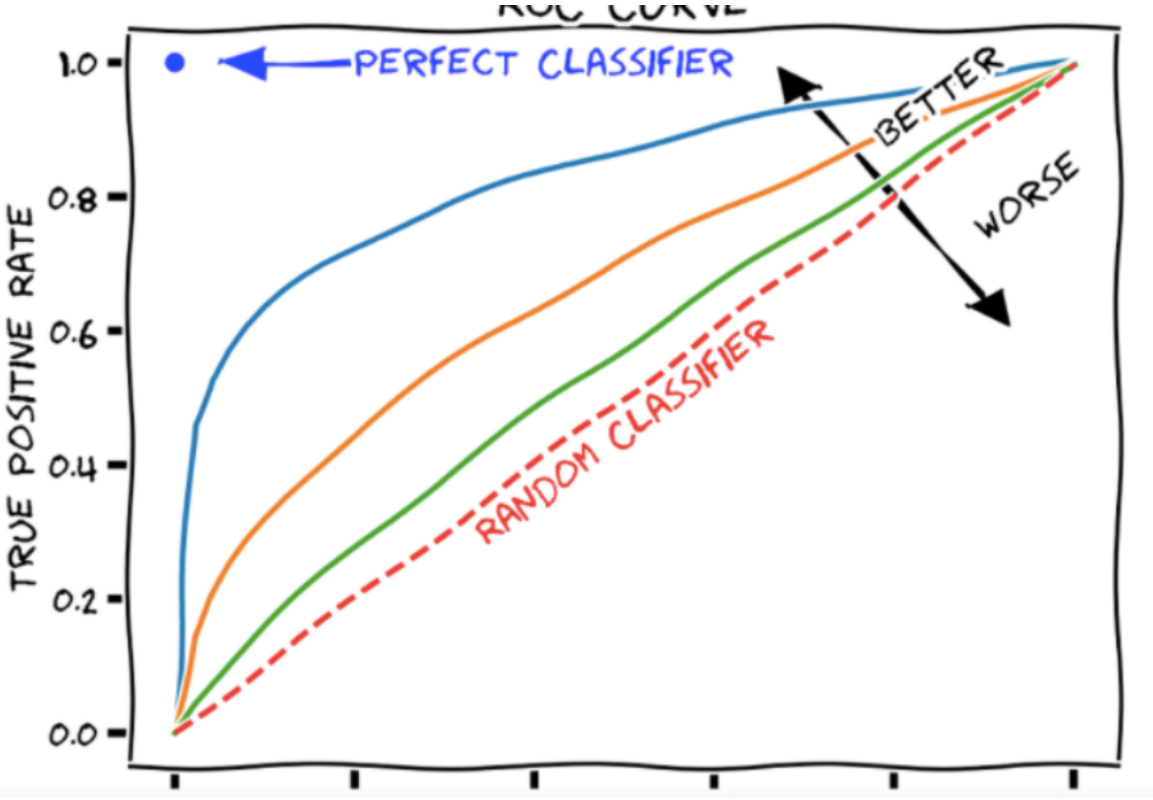
ROC曲线

我们根据学习器的预测结果对样例进行排序，按此顺序逐个把样本作为正例进行预测，每次计算出TPR和FPR，分别以它们为横、纵坐标作图，就得到了 “ROC曲线”.



(a) ROC 曲线与 AUC　　(b) 基于有限样例绘制的 ROC 曲线与 AUC

不同的模型与不同的ROC曲线：很显然不同的算法模型对应不同的ROC曲线，超参数不同的模型也对应不同的ROC曲线。

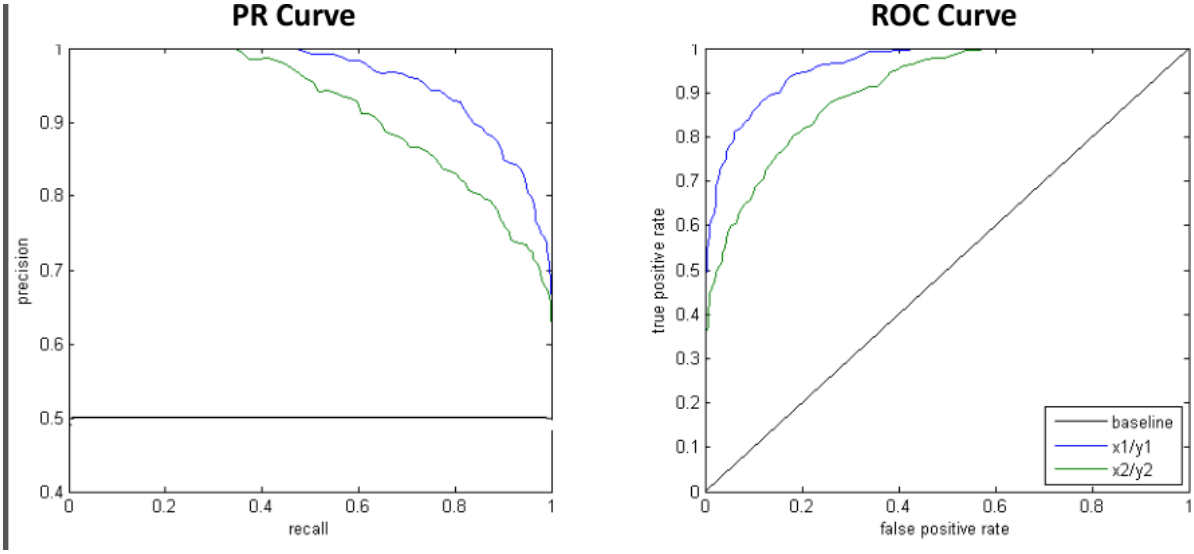


如上图所示，根据上面所说的一个结论，TPR越大，FPR越小则模型的性能就越好，图中的红色的虚线是盲猜时的POC曲线，也是一个基准，在红色上方的ROC曲线对应的模型时可取的，而红色下方的ROC曲线对应的模型时无效的。因为是盲猜，所以得到的模型的样本排序是随机的，也就意味着正反例的分布是按比例的随机分布的，所以在阈值改变的过程中，TPR和FPR是一直相等的。
ROC曲线上的点是一个模型上取不同阈值产生的不同的结果。
理想的决策阈值： 理想的决策阈值是TPR越接近1，FPR越接近0。即越接近上图中蓝色的点。

precision-Recall Curve(PR)

presion $\frac{TP}{TP+FP}$

recall $\frac{TP}{TP+FN}$



AUC（Area Under ROC Curve)

- *AUC*： AUC即ROC曲线下方的面积。
- AUC的值只是衡量各个模型的排名，其绝对值大小没有意义。
- AUC衡量的是在不管取什么阈值的情况下，模型的性能

Model Parameters Versus Hyperparameters 模型参数和超参数

什么是超参数?

比如算法中的**learning rate** α（学习率）、**iterations**(梯度下降法循环的数量)、L（隐藏层数目）、$n[l]$（隐藏层单元数目）、**choice of activation function**（激活函数的选择） 都需要你来设置，这些数字实际上控制了最后的参数W和b的值，所以它们被称作超参数。

超参数有什么用?

- 正则化超参数控制模型的容量。
- 适当控制模型容量可以防止过拟合。

另一种类型的超参数来自训练过程本身。

例如，随机梯度下降(SGD)优化需要学习率、批处理大小。

一些优化方法需要收敛阈值。

这些也需要设置为合理的值，以便训练过程找到一个好的模型。

验证集validation set

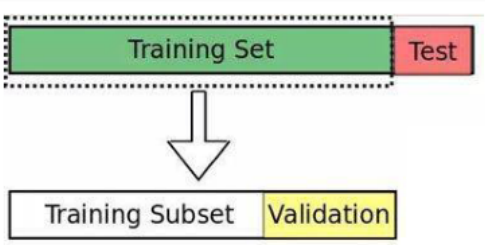
请注意，测试示例没有以任何方式用于对模型(包括其超参数)做出选择。

•因此，测试集中的任何示例都不能用于验证集。

具体来说，我们将训练数据分成两个不相交的子集。

•一个(训练集)用于训练模型参数。

•一个(验证集)用于估计训练期间或之后的泛化误差，允许超参数相应地更新。



超参数优化

每次对特定超参数设置的尝试都涉及到训练模型——一个内部优化过程。