

模型评估与选择

杨昆

计算机学院

杭州电子科技大学

经验误差与过拟合

◆ 误差率 error rate

- 通常把分类错误的样本数 a 占样本总数 m 的比例 $e=a/m$
- 准确率(1-误差率): $1-a/m$

◆ 更一般地, 把学习器的预测结果与样本的真实值之间的差异称为误差(error)

- 学习器在训练集上的误差称为"训练误差training error"或"经验误差empirical error", 新样本上的误差称为"泛化误差generalization error"

◆ 希望得到泛化误差小的学习器

- 事先不知道新样本, 实际上是努力使经验误差最小化

经验误差与过拟合

◆ “过拟合overfitting”与“欠拟合underfitting”

- 努力从训练样本中学习所有潜在样本的“普遍规律”，很可能把训练样本的特点当成所有样本的特性，导致泛化性能下降。
- 与过拟合相对的是欠拟合（一般是学习器的学习能力低）

◆ 模型选择(model selection)问题

- 同一任务有多种学习算法可选，同一算法有不同的参数配置。选哪个算法？选什么参数？
- 理想方案：评选候选模型的泛化误差——实际上无法直接获得泛化误差
- 评估方法是什么？

评估方法

◆ 用实验测试来对学习器的泛化误差进行评估/估计

- 使用一个"测试集" (testing set) 来测试学习器对新样本的预测能力, 以测试集上的"测试误差" (testing error) 作为泛化误差的近似.
- 只有一个包含 m 个样例/样本的数据集 D , 怎么做?

◆ 常见做法

- Holdout method, random subsampling
- 交叉验证 Cross-validation
- 自助法 (解靴带) Bootstrap

留出法Holdout

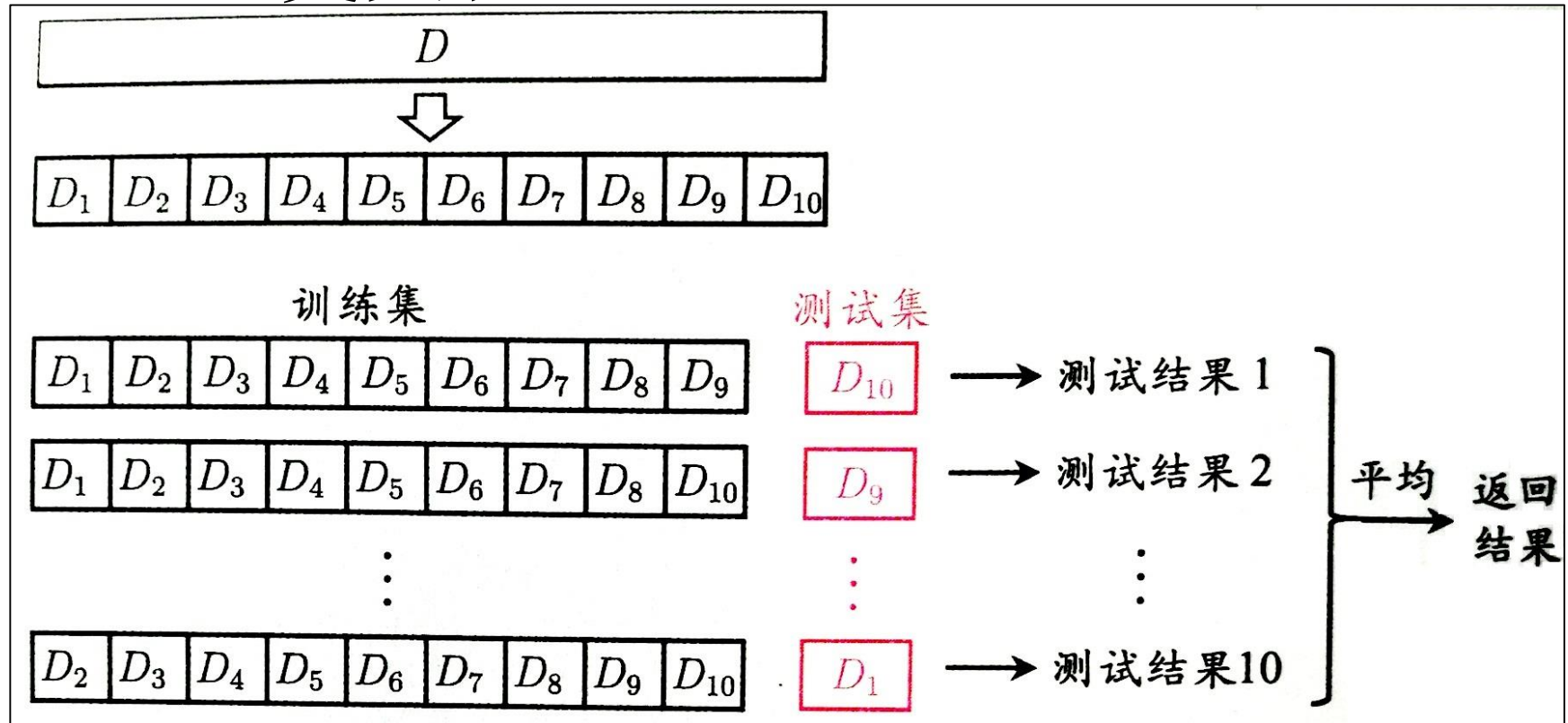
◆ Holdout method“留出法”/“保持方法”

- 给定数据随机分成两个部分
 - 训练集 (e.g., 2/3-4/5) 用于模型构造
 - 测试集 (e.g., 1/3-1/4) 用于错误率估计
- 训练/测试集的划分尽可能保持数据分布的一致性

◆ 单次使用留出法得到的估计结果往往不够稳定可靠

- 随机抽样: a variation of holdout
 - 重复holdout k 次, accuracy = 所有正确率的平均值

交叉验证Cross-Validation



◆ 留一法: Leave-One-Out Cross Validation (LOOCV)

- ◆ $k = |D|$, 即 m 个样本划分成 m 分
- ◆ 留一法的结果往往被认为比较准确 (未必永远都好)
- ◆ 数据集比较大时, 难以忍受

Bootstrap自助法

◆ Bootstrap

- 对于小样本数据，效果很好
- 从给定样本中有放回的均匀抽样 *with replacement*
 - i.e., 每次一个样本被选中, 把它加入训练集并且等可能得被再次选中

◆ 多个自助法，最常用的是 .632 bootstrap

- 含 m 个样本的数据集有放回抽样 m 次，产生 m 个样本的训练集. 没有被抽到的样本组成测试集. 大约63.2%的样本被抽中，剩余的36.8% 形成测试集(因为 $(1 - 1/m)^m \approx e^{-1} = 0.368$)
- 重复抽样过程 k 次，总体准确率为：

$$Acc(M) = \frac{1}{k} \sum_{i=1}^k (0.632 \times Acc(M_i)_{test_set} + 0.368 \times Acc(M_i)_{train_set})$$

调参与最终模型

- ◆ 大多数学习算法都有些参数 (parameter) 需要设定, 参数不同模型的性能往往有显著差别

- 模型评估与选择时除了选算法, 还要设定参数——"参数调节"或"调参parameter tuning"

- ◆ 最终模型的通常做法

- 基于训练集(数据集D的一部分)选定学习算法和配置参数, 然后用整个D重新训练模型 (最终提交用户的模型)

- ◆ 调参的数据集 ?

- 把训练数据再划分成训练集和“验证集validation set”——基于验证集上的性能来进行算法模型的选择和调参。

模型(算法)的性能度量

◆ 衡量模型泛化能力的评价指标(标准) — 性能度量

- 性能度量(performance measure)反应任务需求, 使用不同度量往往导致不同的评判结果 — 好与坏是相对的

◆ 回归任务

- 对学习器 f 常用均方误差mean squared error

$$E(f; D) = \frac{1}{m} \sum_{i=1}^m (f(x_i) - y_i)^2$$

◆ 分类任务

- 错误率/准确率;
- 灵敏性Sensitivity, 特效性Specificity
- 查准率/查全率与F1, 等等。

分类结果的混淆矩阵

◆ 二分类: 设定某类为“**正类/阳性类**”, 对应类为“**负/阴性类**”

- 正样本/负样本

- 真正例/真阳性; 假正例/假阳性;

- 真负例/真阴性; 假负例/假阴性

混淆矩阵Confusion Matrix:

真实结果	预测结果	
	正类	负类
正类	真正例True Positives (TP)	假负例False Negatives (FN)
负类	假正例False Positives (FP)	真负例True Negatives (TN)

例子:

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	6954	46	7000
buy_computer = no	412	2588	3000
Total	7366	2634	10000

准确率, 错误率, 敏感性, 特效性

◆ 分类器准确度: 测试元组被正确识别的比例

- $\text{Accuracy} = (\text{TP} + \text{TN}) / \text{All}$

◆ 错误率

- $1 - \text{Accuracy}$ 或 $\text{Error rate} = (\text{FP} + \text{FN}) / \text{All}$

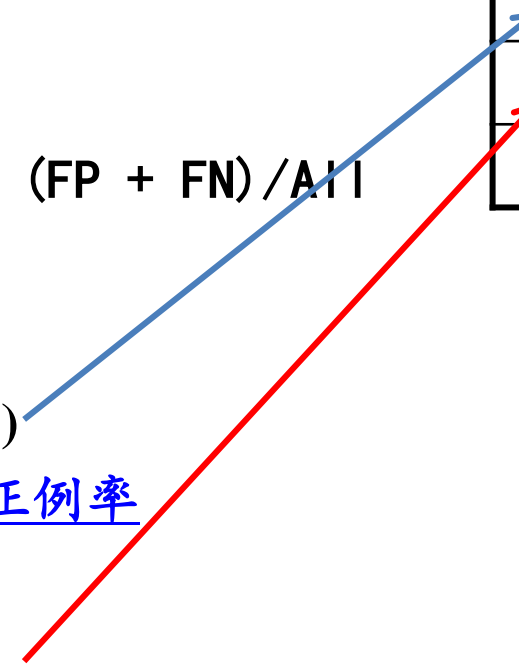
◆ 敏感性Sensitivity

- True Positive recognition rate
- $\text{Sensitivity} = \text{TP} / \text{P} = \text{TP} / (\text{TP} + \text{FN})$
- 就是召回率/查全率, 又称为真正例率

◆ 特效性Specificity

- True Negative recognition rate
- $\text{Specificity} = \text{TN} / \text{N} = \text{TN} / (\text{FP} + \text{TN})$
- $1 - \text{false positive rate}$ (FPR-假真正率)

A\P	正	负	
正	TP	FN	P
负	FP	TN	N
	P'	N'	All



查准率/查全率

表 2.1 分类结果混淆矩阵

真实情况	预测结果	
	正例	反例
正例	TP (真正例)	FN (假反例)
反例	FP (假正例)	TN (真反例)

◆ 查准率/精度 Precision-P

- 预测为正类样本实际属于“正类”的比例

◆ 查全率/召回率 Recall-R

- 也就是敏感性

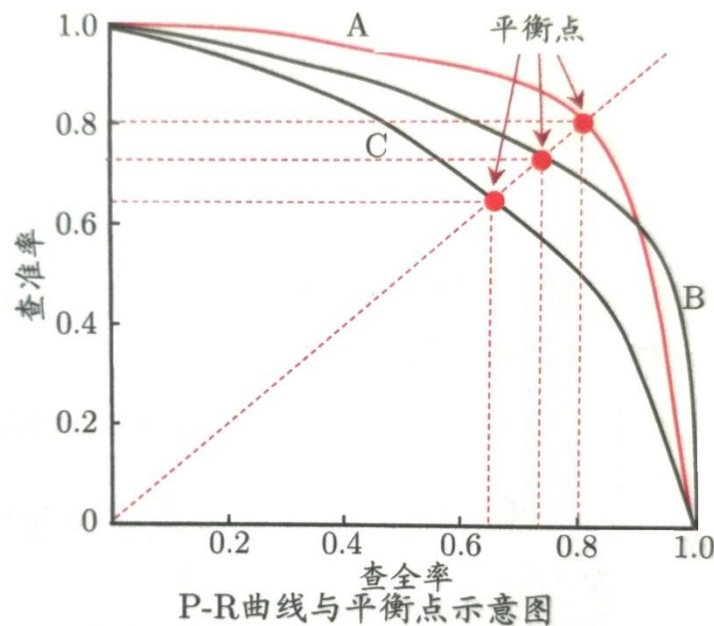
◆ 查准率与查全率是一对矛盾

◆ P-R曲线与平衡点

- 根据预测结果对样本进行排序, 最可能“正”的排前, 最不可能排后. 按此次序逐个把样本预测为正例, 计算每次的查准/查全率. 绘制查准率-查全率曲线
- “平衡点” (Break-Even Point-BEP) 就是一个度量-查准率=查全率时的取值
- 可以基于BEP比较两个交错的曲线

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

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



分类器评价指标: 例子

真实类\预测类	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	90	210	300	30.00 <i>(sensitivity)</i>
cancer = no	140	9560	9700	98.56 <i>(specificity)</i>
Total	230	9770	10000	96.40 <i>(accuracy)</i>

● $Precision = 90/230 = 39.13\%$

$$Recall = 90/300 = 30.00\%$$

F1和 F_β

◆ F measure(F1 or F-score)

- 精度和召回的调和平均值

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

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

◆ F_β : 查准率和查全率的加权量

- 对查准率和查全率的重视程度不同，
更一般的形式。
- $\beta > 1$ 时查全率有更大影响； $\beta < 1$ 时查准率
有更大影响

宏/微-查准率查全率

◆ 前提

- 多个数据集或多次训练/测试得到n个二分类混淆矩阵. 希望综合考察.

$$macro_P = \frac{1}{n} \sum_{i=1}^n P_i$$

$$macro_R = \frac{1}{n} \sum_{i=1}^n R_i$$

◆ 宏查准率/宏查全率

- 在每个混淆矩阵上计算查准率与查全率, 再计算平均值.

$$macro_F_1 = \frac{2 \times macro_P \times macro_R}{macro_P + macro_R}$$

◆ 微查准率/微查全率

- 先把混淆矩阵的对应元素求平均值, 基于这些均值再计算

$$micro_P = \frac{\overline{TP}}{\overline{TP} + \overline{FP}}$$

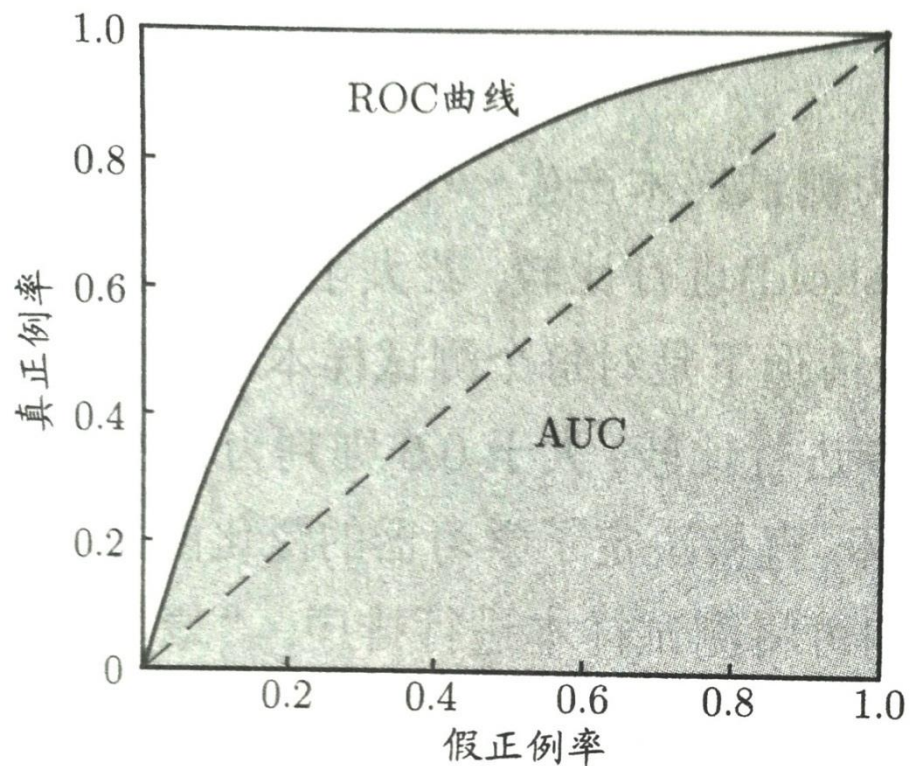
$$micro_R = \frac{\overline{TP}}{\overline{TP} + \overline{FN}}$$

$$micro_F_1 = \frac{2 \times micro_P \times micro_R}{micro_P + micro_R}$$

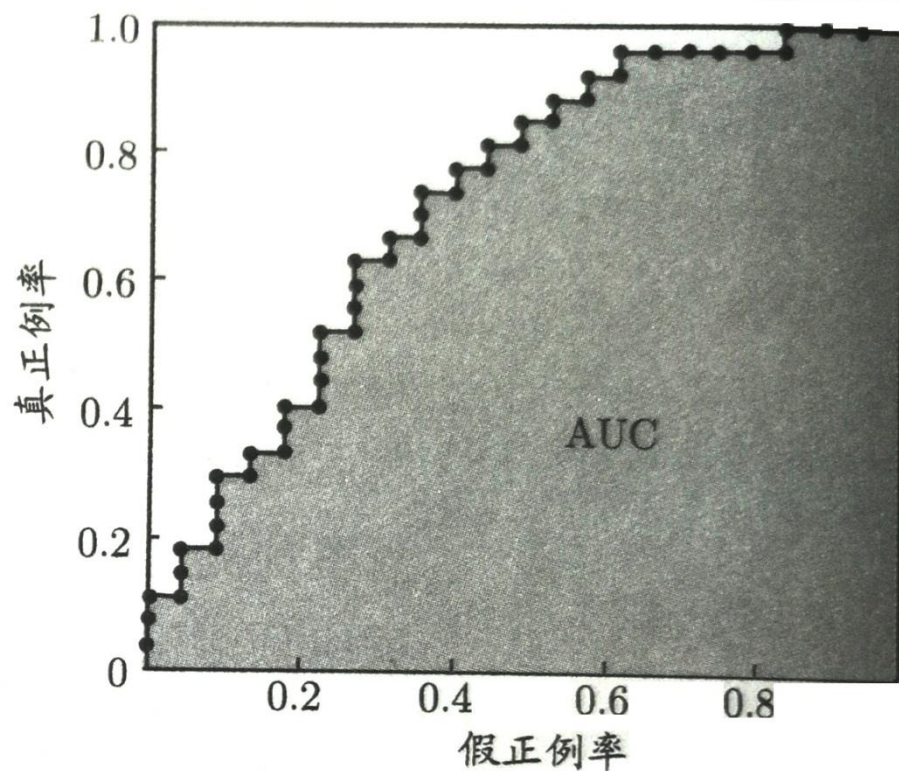
ROC Curves

- ◆ **ROC (Receiver Operating Characteristics) curves-接受器/受试者工作特征**
 - 源于雷达信号分析技术
 - true positive rate和false positive rate间的折衷
 - 测试样本递减排列：最可能属于正类的排在最顶端
- ◆ **ROC曲线下的面积就是模型正确率的度量-AUC(Area Under ROC Curve)**
 - 纵坐标true positive rate-真正例率 $TPR=TP/(TP+FN)$
 - 横坐标the false positive rate-假正例率 $FPR=FP/(TN+FP)=1-specificity$
 - A model with perfect accuracy will have an area of 1.0

ROC Curves



(a) ROC 曲线与 AUC



(b) 基于有限样例绘制的 ROC 曲线
与 AUC

ROC Curves

◆ 5个正样本， 5个负样本， $P=5$ ， $N=5$

纵轴true positive rate-真正例率 $TPR=TP/(TP+FN)$

横轴the false positive rate-假正例率 $FPR=FP/(TN+FP)=1-\text{specificity}$

元组编号	类	概率	TP	FP	TN	FN	TPR	FPR
1	P	0.90	1	0	5	4	0.2	0
2	P	0.80	2	0	5	3	0.4	0
3	N	0.70	2	1	4	3	0.4	0.2
4	P	0.60	3	1	4	2	0.6	0.2
5	P	0.55	4	1	4	1	0.8	0.2
6	N	0.54	4	2	3	1	0.8	0.4
7	N	0.53	4	3	2	1	0.8	0.6
8	N	0.51	4	4	1	1	0.8	0.8
9	P	0.50	5	4	1	0	1.0	0.8
10	N	0.40	5	5	0	0	1.0	1.0

图 8. 18 元组按递减得分排序，其中得分是概率分类器返回的值

代价敏感错误率

- ◆ 为了权衡不同类型错误的不同损失，为错误赋予“非均等代价” (unequal cost)

表 二分类代价矩阵

真实类别	预测类别	
	第 0 类	第 1 类
第 0 类	0	$cost_{01}$
第 1 类	$cost_{10}$	0

- ◆ 根据领域知识设定一个“代价矩阵” (cost matrix)

- 表示将第 i 类预测为第 j 类的代价

- ◆ 希望最小化“总体代价” total cost

- “代价敏感” (cost-sensitive) 错误率

$$E(f; D; cost) = \frac{1}{m} \left(\sum_{\mathbf{x}_i \in D^+} \mathbb{I}(f(\mathbf{x}_i) \neq y_i) \times cost_{01} + \sum_{\mathbf{x}_i \in D^-} \mathbb{I}(f(\mathbf{x}_i) \neq y_i) \times cost_{10} \right)$$

估计置信区间: 分类器 M_1 vs. M_2

- ◆ 假定有两个分类器 M_1 and M_2 , 那一个更好?
- ◆ 用10-fold cross-validation获得了 $\overline{err}(M_1)$ $\overline{err}(M_2)$
- ◆ 这些平均误差率仅仅是未来数据总体误差的一种估计
- ◆ 2个错误率之间差别如果是否是偶然的?
 - 使用统计显著性检验
 - 获得估计误差的**confidence limits**置信界

估计置信区间: Null Hypothesis

- 执行 10-fold cross-validation
- 假定样本服从 $k-1$ 个自由度的 **t distribution** ($k=10$)
degrees of freedom
- Use **t-test** (or **Student's t-test**)
- 零假设 **Null Hypothesis**: M_1 & M_2 相同（即没有区别）
- 如果可以拒绝 null hypothesis, 那么
 - 可以断定 M_1 & M_2 间的不同是统计上显著的
 - Chose model with lower error rate

估计置信区间: t-test

- 当只有一个测试集时: 成对比较 **pairwise comparison**
 - 对于10倍交叉验证中的 i^{th} round, 使用相同的样本分割 来计算 $err(M_1)_i$ and $err(M_2)_i$
 - 然后求平均over 10 $\overline{err}(M_1)$ and $\overline{err}(M_2)$
 - **t-test computes t-statistic with $k-1$ degrees of freedom:**

$$t = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{var(M_1 - M_2)/k}} \quad \text{其中}$$

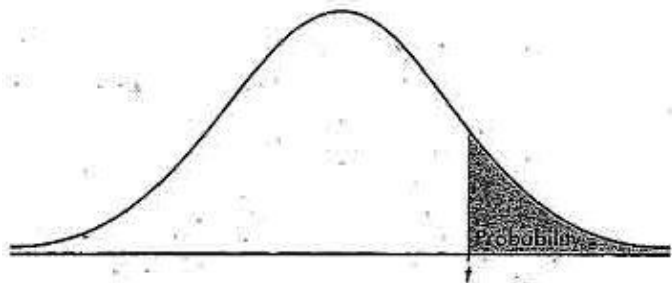
$$var(M_1 - M_2) = \frac{1}{k} \sum_{i=1}^k \left[err(M_1)_i - err(M_2)_i - (\overline{err}(M_1) - \overline{err}(M_2)) \right]^2$$

- 如果有两个测试集: use **non-paired t-test**

$$\text{where } var(M_1 - M_2) = \sqrt{\frac{var(M_1)}{k_1} + \frac{var(M_2)}{k_2}},$$

where k_1 & k_2 are # of cross-validation samples used for M_1 & M_2 , resp.

估计置信区间: Table for t-distribution



- ◆ Symmetric
- ◆ Significance level, e. g., $sig = 0.05$ or 5% means M_1 & M_2 are *significantly different* for 95% of population
- ◆ Confidence limit, $z = sig/2$

TABLE B: t-DISTRIBUTION CRITICAL VALUES

df	Tail probability p											
	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	.688	.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	.687	.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850
21	.686	.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135	3.527	3.819
22	.686	.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119	3.505	3.792
23	.685	.858	1.060	1.319	1.714	2.069	2.177	2.500	2.807	3.104	3.485	3.768
24	.685	.857	1.059	1.318	1.711	2.064	2.172	2.492	2.797	3.091	3.467	3.745
25	.684	.856	1.058	1.316	1.708	2.060	2.167	2.485	2.787	3.078	3.450	3.725
26	.684	.856	1.058	1.315	1.706	2.056	2.162	2.479	2.779	3.067	3.435	3.707
27	.684	.855	1.057	1.314	1.703	2.052	2.158	2.473	2.771	3.057	3.421	3.690
28	.683	.855	1.056	1.313	1.701	2.048	2.154	2.467	2.763	3.047	3.408	3.674
29	.683	.854	1.055	1.311	1.699	2.045	2.150	2.462	2.756	3.038	3.396	3.659
30	.683	.854	1.055	1.310	1.697	2.042	2.147	2.457	2.750	3.030	3.385	3.646
40	.681	.851	1.050	1.303	1.684	2.021	2.123	2.423	2.704	2.971	3.307	3.551
50	.679	.849	1.047	1.299	1.676	2.009	2.109	2.403	2.678	2.937	3.261	3.496
60	.679	.848	1.045	1.296	1.671	2.000	2.099	2.390	2.660	2.915	3.232	3.460
80	.678	.846	1.043	1.292	1.664	1.990	2.088	2.374	2.639	2.887	3.195	3.416
100	.677	.845	1.042	1.290	1.660	1.984	2.081	2.364	2.626	2.871	3.174	3.390
1000	.675	.842	1.037	1.282	1.646	1.962	2.056	2.330	2.581	2.813	3.098	3.300
∞	.674	.841	1.036	1.282	1.645	1.960	2.054	2.326	2.576	2.807	3.091	3.291
	50%	60%	70%	80%	90%	95%	96%	98%	99%	99.5%	99.8%	99.9%

Confidence level C

估计置信区间: Statistical Significance

- M_1 & M_2 是否显著得不同?
 - Compute t . Select *significance level* (e.g. $sig = 5\%$)
 - Consult table for t-distribution: Find t value corresponding to $k-1$ *degrees of freedom* (here, 9)
 - t-分布对称: 通常显示分布的上百分点 % → 查找值 **confidence limit** $z=sig/2$ (here, 0.025)
 - If $t > z$ or $t < -z$, 那么 t 的值位于拒绝域:
 - **Reject null hypothesis** that mean error rates of M_1 & M_2 are same
 - Conclude: statistically significant difference between M_1 & M_2
 - **Otherwise**, conclude that any difference is **chance**

Friedman检验和Nemenyi后续检验

◆多数据集上同时比较多个算法

●周志华. 机器学习. 清华大学出版社

表 2.5 算法比较序值表

数据集	算法 A	算法 B	算法 C
D_1	1	2	3
D_2	1	2.5	2.5
D_3	1	2	3
D_4	1	2	3
平均序值	1	2.125	2.875

