



# 数据集及其拆分

(Data Set and Its Split)

刘远超

哈尔滨工业大学

计算机科学与技术学院

# Iris(鸢尾花)数据集



山鸢尾(*Iris setosa*)



变色鸢尾(*Iris versicolor*)



维吉尼亚鸢尾(*iris virginica*)

分类特征：花萼（sepal）和花瓣（petal）的宽度和长度

# Iris(鸢尾花)数据集(续)

**Samples**  
(instances, observations)

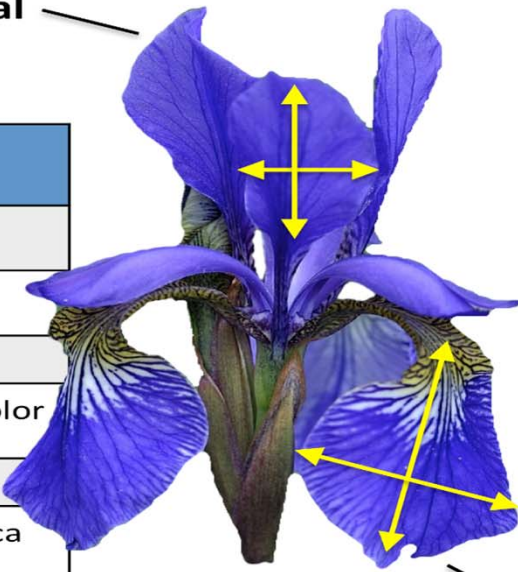
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

**Features**  
(attributes, measurements, dimensions)

**Petal**

**Sepal**

**Class labels**  
(targets)



- 每个样本包含4个特征（单位：cm），1个类别标签（类别编码）。
- 共有150个样本，3类，每类50个样本。

# 数据集(dataset)的数学表示

**Samples**  
(instances, observations)

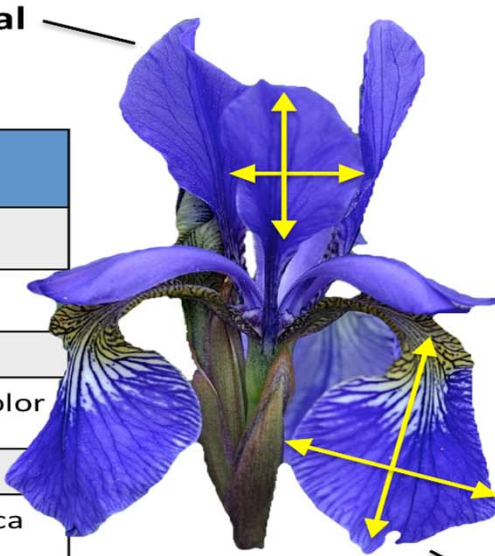
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

**Features**  
(attributes, measurements, dimensions)

**Petal**

**Sepal**

**Class labels**  
(targets)



数据集在数学上通常表示为 $\{(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots, (x_m, y_m)\}$ 的形式,

- 其中 $x_i$ 为样本特征。由于样本一般有多个特征, 因而 $x_i = \{x_i^1, x_i^2, \dots, x_i^n\}^T$ 。
- 而 $y_i$ 表示样本  $i$  的类别标签。

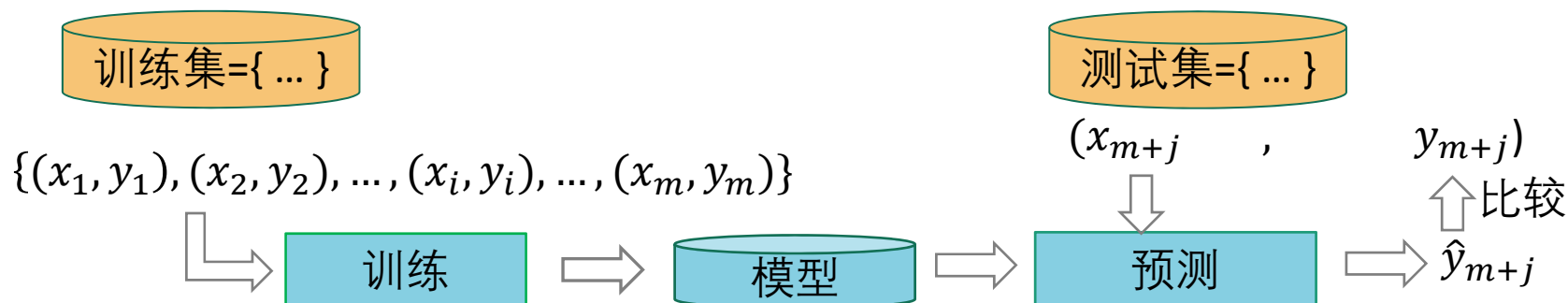


# 类别标签的ground truth与gold standard



- **ground truth**: 可翻译为地面实况等。在机器学习领域一般用于表示真实值、**标准答案**等，表示通过直接观察收集到的真实结果。
- **gold standard**: 可翻译为金标准。医学上一般指诊断疾病公认的最可靠的方法。
- 在机器学习领域，更倾向于使用“**ground truth**”。而如果用 **gold standard**这个词，则表示其可以很好地代表**ground truth**。

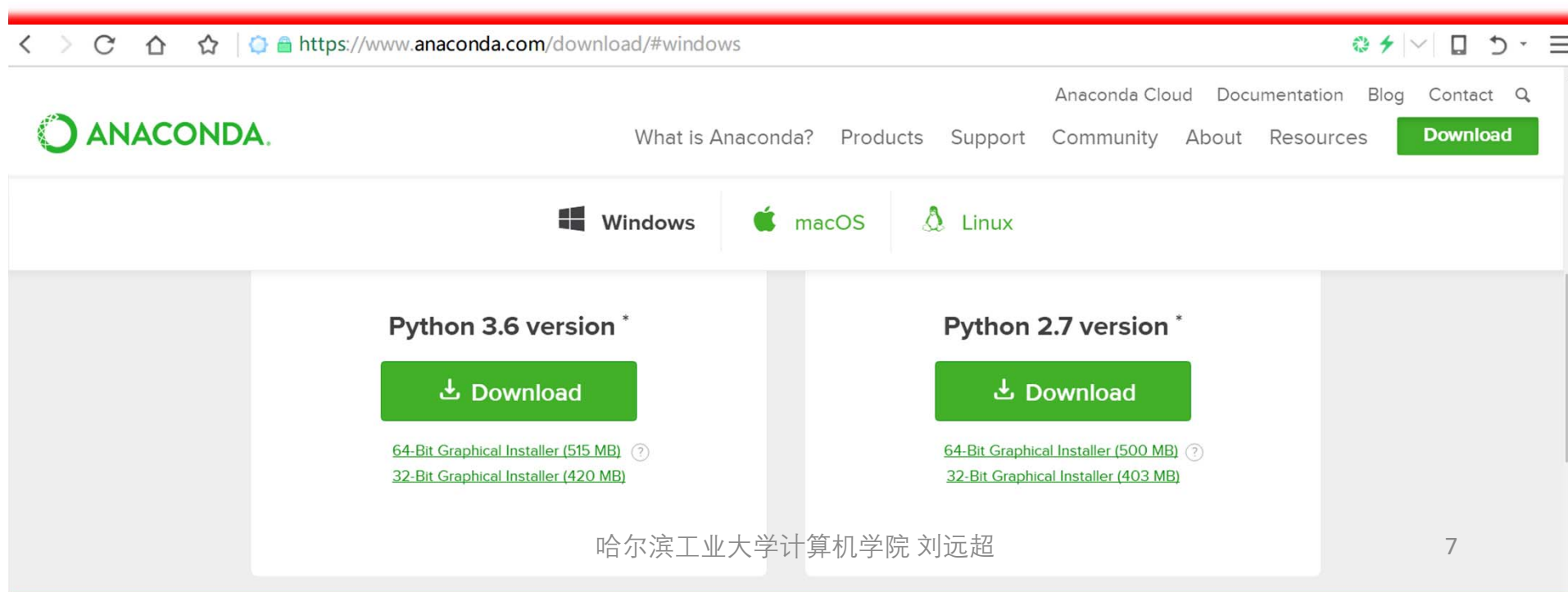
# 数据集与有监督学习



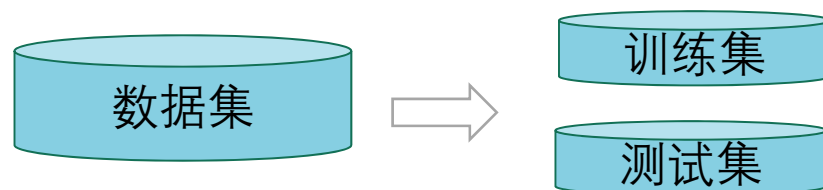
- 有监督学习中数据通常分成训练集、测试集两部分。
  - **训练集(training set)**用来训练模型，即被用来学习得到系统的参数取值。
  - **测试集(testing set)**用于最终报告模型的评价结果，因此在训练阶段测试集中的样本应该是unseen的。
- 有时对训练集做进一步划分为训练集和验证集(validation set)。验证集与测试集类似，也是用于评估模型的性能。区别是验证集主要用于模型选择和调整超参数，因而一般不用于报告最终结果。

# 训练集、测试集的拆分

- 可以使用sklearn (即scikit-learn) 进行训练集、测试集的拆分。
- 如何安装sklearn: anaconda是一个开源的Python发行版本，其包含了很多科学包及其依赖项，也包含sklearn。
  - 网址: <https://www.anaconda.com/download/#windows>



# 训练集测试集拆分—留出法



- 留出法（**Hold-Out Method**）数据拆分步骤：
  1. 将数据随机分为两组，一组做为训练集，一组做为测试集
  2. 利用训练集训练分类器，然后利用测试集评估模型，记录最后的分类准确率为此分类器的性能指标
- 留出法的优点是处理简单。而**不足之处**是在测试集上的预测性能的高低与数据集拆分情况有很大的关系，所以基于这种数据集拆分基础上的性能评价结果不够稳定。



# K折交叉验证



5折交叉验证示意图  Train set  Test set

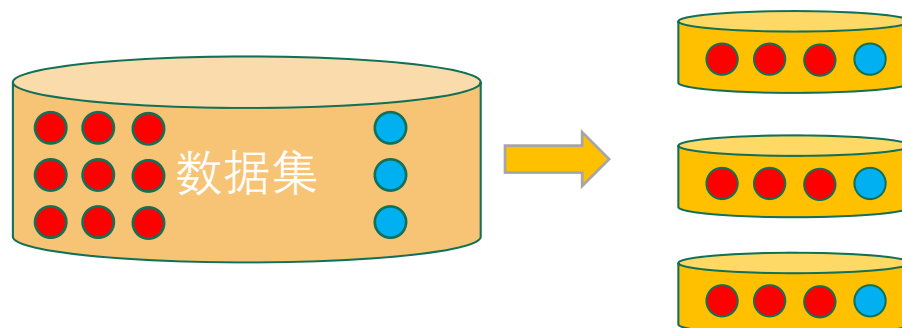
## ●过程:

1. 数据集被分成K份（K通常取5或者10）
2. 不重复地每次取其中一份做测试集，用其他K-1份做训练集训练，这样会得到K个评价模型
3. 将上述步骤2中的K次评价的性能均值作为最后评价结果

## ● K折交叉验证的上述做法有助于提高评估结果的稳定性

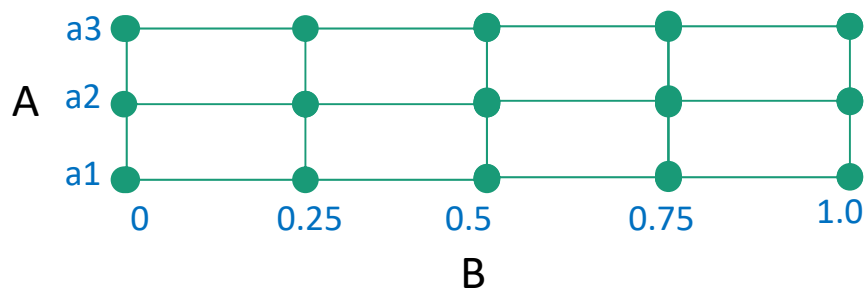
# 分层抽样策略 (Stratified k-fold)

- 将数据集划分成k份，特点在于，划分的k份中，每一份内各个类别数据的比例和原始数据集中各个类别的比例相同。

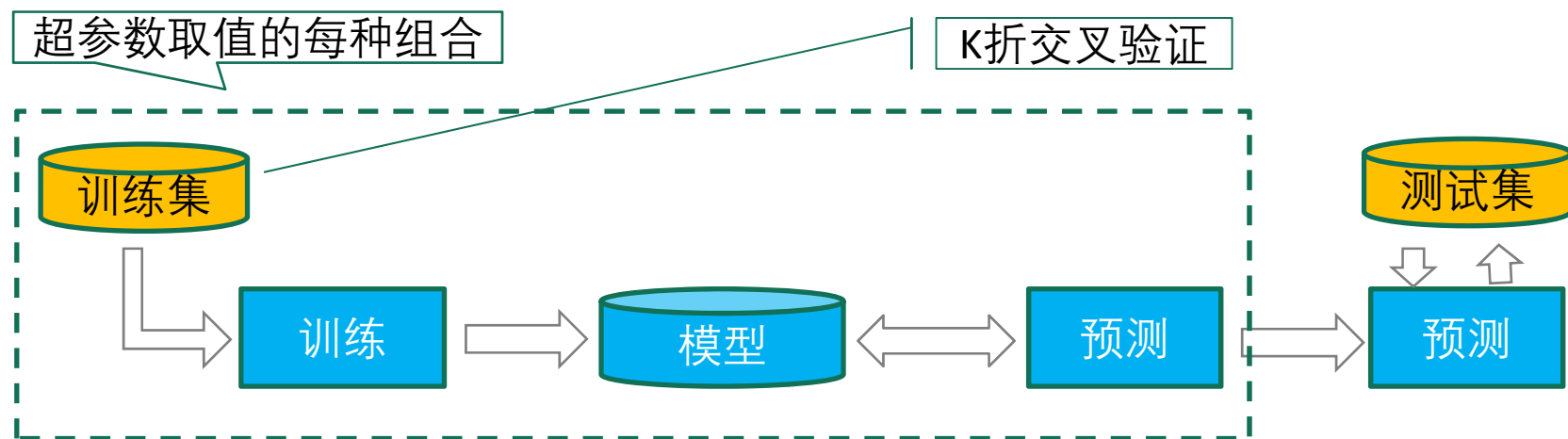


# 用网格搜索来调超参数 (一)

- **什么是超参数?** 指在学习过程之前需要设置其值的一些变量, 而不是通过训练得到的参数数据。如深度学习中的学习速率等就是超参数。
- **什么是网格搜索?**
  - 假设模型中有2个超参数: A和B。A的可能取值为{a1, a2, a3}, B的可能取值为连续的, 如在区间[0-1]。由于B值为连续, 通常进行离散化, 如变为{0, 0.25, 0.5, 0.75, 1.0}
  - 如果使用网格搜索, 就是尝试各种可能的(A, B)对值, 找到能使用的模型取得最高性能的(A, B)值对。



## 用网格搜索来调超参数 (二)



### 网格搜索与K折交叉验证结合调整超参数的具体步骤：

1. 确定评价指标；
2. 对于超参数取值的每种组合，在训练集上使用交叉验证的方法求得其K次评价的性能均值；
3. 最后，比较哪种超参数取值组合的性能最好，从而得到最优超参数的取值组合。

# Thanks!







# 分类及其性能度量

(Classification Problem and Its Performance Evaluation)

刘远超

哈尔滨工业大学

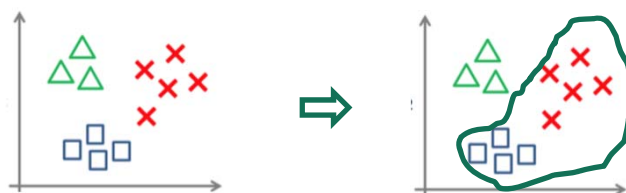
计算机科学与技术学院

# 分类问题

- 分类问题是有监督学习的一个核心问题。分类解决的是要预测样本属于哪个或者哪些预定义的类别。此时输出变量通常取有限个离散值。



- 分类的机器学习的两大阶段：1) 从训练数据中学习得到一个分类决策函数或分类模型，称为分类器 (classifier)；2) 利用学习得到的分类器对新的输入样本进行类别预测。
- 两类分类问题与多类分类问题。多类分类问题也可以转化为两类分类问题解决，如采用一对其余(One-vs-Rest)的方法：将其中一个类标记为正类，然后将剩余的其它类都标记成负类。



# 分类性能度量—准确率

- 假设只有两类样本，即正例(positive)和负例(negative)。通常以关注的类为正类，其他类为负类。

实际类别	预测类别			
		正	负	总计
	正	TP	FN	P(实际为正)
	负	FP	TN	N(实际为负)

表中AB模式：第二个符号表示预测的类别，第一个表示预测结果对了(T rue)还是错了(F alse)

- 分类准确率 (**accuracy**)：分类器正确分类的样本数与总样本数之比：

$$accuracy = \frac{TP+TN}{P+N}$$

思考：假设共有100个短信，其实际情况为，其中有1个是垃圾短信，99个是非垃圾短信。某分类模型将这100个短信都分为非垃圾短信，则准确率 (**accuracy**) 为？

# 分类性能度量—精确率和召回率

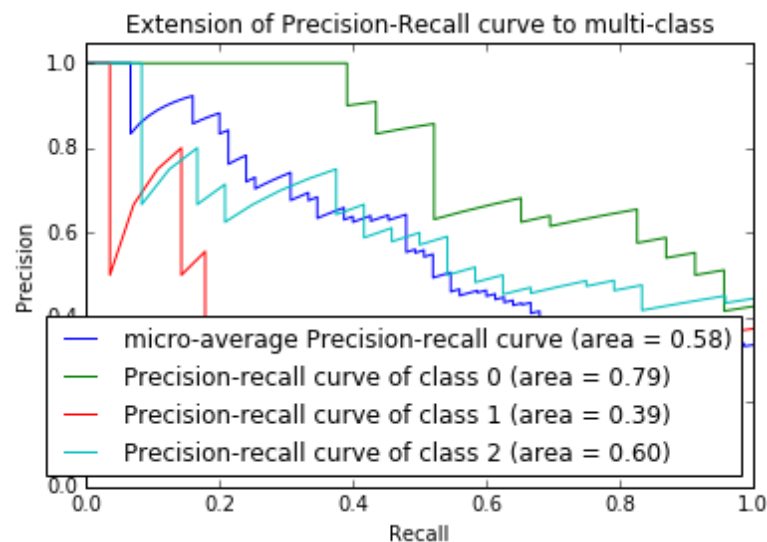
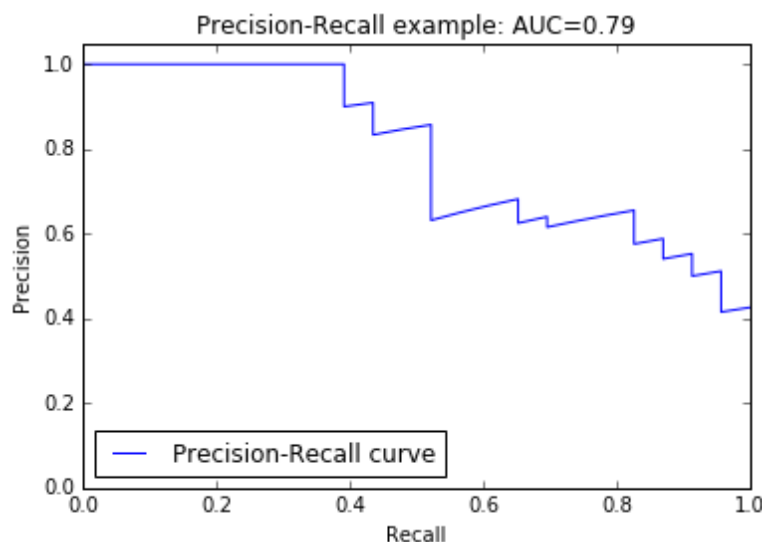
实际类别	预测类别			
		正例	负例	总计
	正例	TP	FN	P(实际为正例)
	负例	FP	TN	N(实际为负例)

- **精确率(precision)和召回率(recall)**: 是二类分类问题常用的评价指标。

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

- **精确率**反映了模型判定的正例中真正正例的比重。在垃圾短信分类器中，是指预测出的垃圾短信中真正垃圾短信的比例。
- **召回率**反映了总正例中被模型正确判定正例的比重。医学领域也叫做灵敏度（sensitivity）。在垃圾短信分类器中，指所有真的垃圾短信被分类器正确找出来的比例。

# 分类性能度量—P-R曲线



## ● Area (Area Under Curve, 或者简称AUC)

■ Area的定义（p-r曲线下的面积）如下：

$$Area = \int_0^1 p(r)dr$$

■ Area有助于弥补P、R的单点值局限性，可以反映全局性能。



## 如何绘制P-R曲线

- 要得到P-R曲线，需要一系列Precision和Recall的值。这些系列值是通过阈值来形成的。对于每个测试样本，分类器一般都会给了“Score”值，表示该样本多大概率上属于正例。
- 步骤：
  1. 从高到低将“Score”值排序并依此作为阈值threshold;
  2. 对于每个阈值，“Score”值大于或等于这个threshold的测试样本被认为正例，其它为负例。从而形成一组预测数据。

实际类别	预测类别			
	正例	负例	总计	
	正例	TP	FN	P(实际为正例)
	负例	FP	TN	N(实际为负例)

$$(\text{precision} = \frac{TP}{TP+FP}, \quad \text{recall} = \frac{TP}{P})$$

样本#	实际类别	预测分值
1	P	0.9
2	N	0.8
3	P	0.75
4	N	0.7
5	P	0.65

19

# 分类性能度量--F值

- F值( $F_\beta$ -score)是精确率和召回率的调和平均:

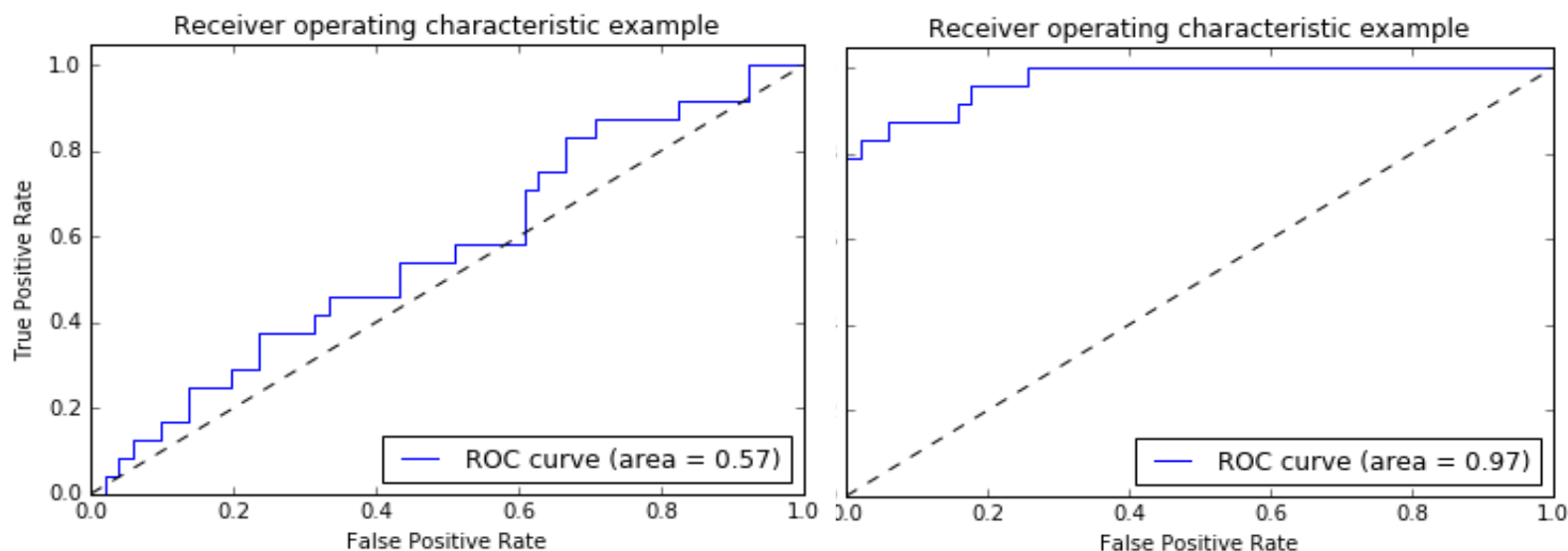
$$F_\beta\text{-score} = \frac{(1+\beta^2)*precision*recall}{(\beta^2*precision+recall)}$$

- $\beta$ 一般大于0。当 $\beta=1$ 时，退化为F1:

$$F_1\text{-score} = \frac{2*precision*recall}{(precision+recall)}$$

- 比较常用的是 $F_1$ ，即表示二者同等重要

# 分类性能度量--ROC



- 横轴：假正例率  $fp\ rate = \frac{FP}{N}$
- 纵轴：真正例率  $tp\ rate = \frac{TP}{P}$
- ROC (受试者工作特征曲线, receiver operating characteristic curve)描绘了分类器在 $tp\ rate$ (真正正例占总正例的比率, 反映**命中概率**, 纵轴)和 $fp\ rate$ (错误的正例占反例的比率, 反映误诊率、假阳性率、**虚惊概率**, 横轴)间的 trade-off。

# 分类性能度量—ROC曲线绘制

- 要得到一个曲线，需要一系列 *fp rate* 和 *tp rate* 的值。这些系列值是通过阈值来形成的。对于每个测试样本，分类器一般都会给了“Score”值，表示该样本多大程度上属于正例（或负例）。
- 步骤：
  1. 从高到低将“Score”值排序并依此作为阈值threshold;
  2. 对于每个阈值，“Score”值大于或等于这个threshold的测试样本被认为正例，其它为负例。从而形成一组预测数据。

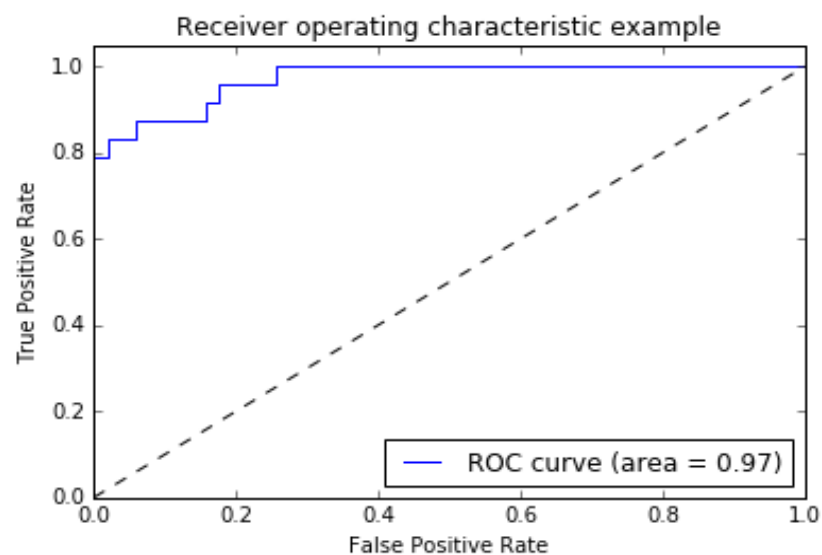
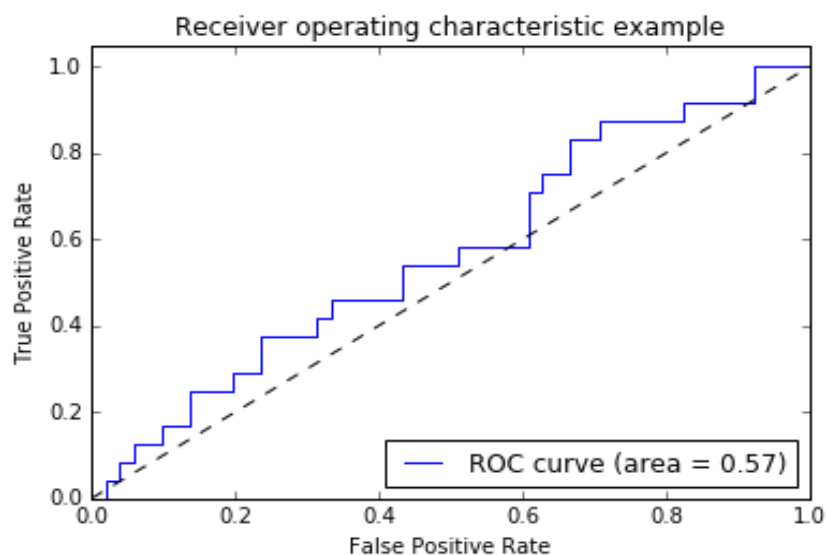
实际类别	预测类别			
		正例	负例	总计
	正例	TP	FN	P(实际为正例)
	负例	FP	TN	N(实际为负例)

$$(fp\ rate = \frac{FP}{N}, \quad tp\ rate = \frac{TP}{P})$$

样本#	实际类别	预测分值
1	P	0.9
2	N	0.8
3	P	0.75
4	N	0.7
5	P	0.65

Diagram illustrating the thresholding process for ROC curve generation. The table shows 5 samples with their actual classes (P for Positive, N for Negative) and predicted scores. Red horizontal lines indicate thresholds at 0.9, 0.8, and 0.7. To the right, a vertical axis shows the cumulative counts of Positive (P) and Negative (N) samples as the threshold decreases. At threshold 0.9, only sample 1 (P) is included. At threshold 0.8, samples 1 and 2 (N) are included. At threshold 0.7, samples 1, 2, and 3 (P) are included. Further decreases in threshold would include samples 4 (N) and 5 (P).

# 分类性能度量—ROC-AUC计算

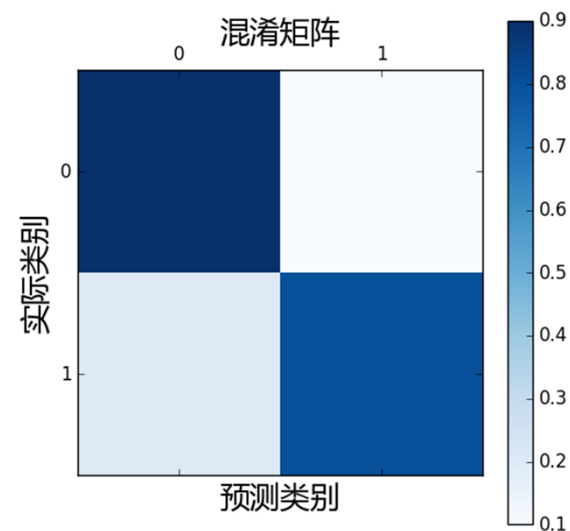


- **ROC- AUC (Area Under Curve)** 定义为ROC曲线下的面积
- AUC值提供了分类器的一个整体数值。通常AUC越大，分类器更好
- 取值范围为[0,1]



# 分类性能可视化

实际类别	预测类别			
		正例	负例	总计
	正例	TP	FN	P(实际为正例)
	负例	FP	TN	N(实际为负例)



## ● 混淆矩阵 (Confusion matrix) 的可视化

- 如用热图 (heatmap) 直观地展现类别的混淆情况 (每个类有多少样本被错误地预测成另一个类)

# 分类报告

- **分类报告(Classification report)**显示每个类的分类性能。包括每个类标签的精确率、召回率、F1值等。

	precision	recall	f1-score	support
class 0	0.67	1.00	0.80	2
class 1	0.00	0.00	0.00	1
class 2	1.00	1.00	1.00	2
avg / total	0.67	0.80	0.72	5

# Thanks!





# 回归问题及其性能评价

(Regression Problem and Its Performance Evaluation)

刘远超

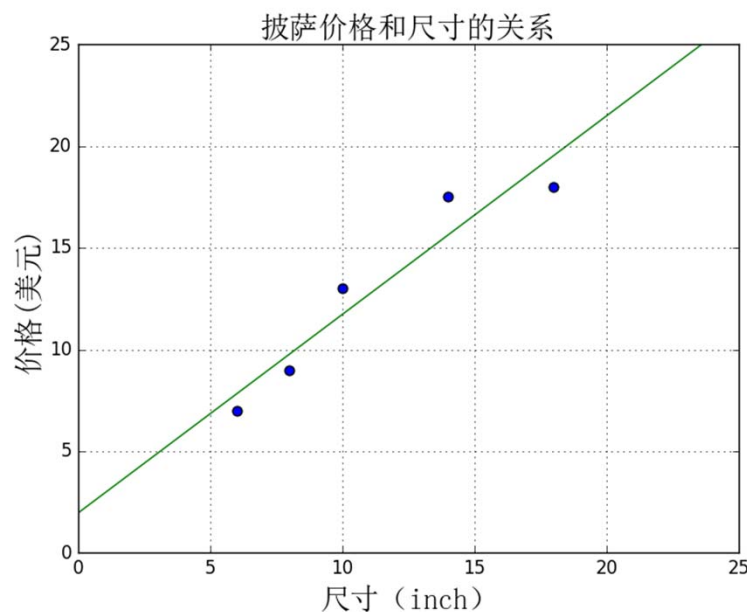
哈尔滨工业大学

计算机科学与技术学院

# 回归问题

## ● 什么是回归？

- 回归分析 (regression analysis) 是确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法。



- 和分类问题不同，回归通常输出为一个实数数值。而分类的输出通常为若干指定的类别标签。



# 回归性能度量方法(Regression metrics)

- 常用的评价回归问题的方法:

- 平均绝对误差MAE(mean\_absolute\_error)
- 均方误差MSE (mean\_squared\_error)及均方根差RMSE
- Log loss, 或称交叉熵loss(cross-entropy loss)
- R方值, 确定系数( r2\_score) (后文介绍)

# 平均绝对误差MAE

- **MAE (Mean absolute error)**是绝对误差损失 (absolute error loss) 的期望值。
- 如果  $\hat{y}_i$  是第*i*个样本的预测值， $y_i$  是相应的真实值，那么在 $n_{\text{samples}}$ 个测试样本上的平均绝对误差 (MAE) 的定义如下：

$$MAE(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i|$$

## 均方差MSE

- **MSE(Mean squared error)**, 该指标对应于平方误差损失 (squared error loss) 的期望值。
- 如果 $\hat{y}_i$ 是第*i*个样本的预测值,  $y_i$  是相应的真实值, 那么在 $n_{samples}$ 上的均方差的定义如下:

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} |y_i - \hat{y}_i|^2$$

- **均方根差RMSE**: Root Mean Squared Error, RMSE,是MSE的平方根

# 均方差MSE的应用举例

- 假设模型预测某日下雨的概率为P:

预测概率	Ground truth (1表示下雨, 0表示没下雨)	MSE	模型性能
1.0	1	$(1.0-1.0)^2=0$	完美
1.0	0	$(1.0-0)^2=1$	糟糕
0.7	1	$(0.7-1.0)^2=0.09$	误差较小
0.3	1	$(0.3-1.0)^2=0.49$	误差较大

# logistic回归损失(二类)

- 简称Log loss，或交叉熵损失(cross-entropy loss)

- 常用于评价逻辑回归LR和神经网络

- 对于二类分类问题：

1. 假设某样本的真实标签为 $y$  (取值为0或1)，概率估计为 $p = pr(y = 1)$ ,
2. 每个样本的log loss是对分类器给定真实标签的负log似然估计 (negative log-likelihood):

$$L_{\log}(y, p) = -\log(pr(y|p)) = -(y \log(p) + (1 - y) \log(1 - p))$$

## logistic回归损失(二类)

- 公式:  $L_{\log}(y, p) = -\log(\text{pr}(y|p)) = -(y \log(p) + (1 - y) \log(1 - p))$

- 假设  $y_{\text{true}} = [0, 0, 1, 1]$   
 $y_{\text{pred}} = [[.9, .1], [.8, .2], [.3, .7], [.01, .99]]$

对于第一个样本,  $y=0, p=0.1$

则  $L_{\log}(y, p) = -(y \log(p) + (1 - y) \log(1 - p)) = -(1 * \log 0.9)$

对于第二个样本,  $y=0, p=0.2,$

则  $L_{\log}(y, p) = -(y \log(p) + (1 - y) \log(1 - p)) = -(1 * \log 0.8)$

对于第三个样本,  $y=1, p=0.7,$

则  $L_{\log}(y, p) = -(y \log(p) + (1 - y) \log(1 - p)) = -(1 * \log 0.7)$

对于第四个样本,  $y=1, p=0.99,$

则  $L_{\log}(y, p) = -(y \log(p) + (1 - y) \log(1 - p)) = -(1 * \log 0.99)$

以对数e为底数, 则上述四项的均值为0.1738 (Sklearn中默认为以e为底)

## logistic回归损失(多类)

- 对于多类问题(multiclass problem), 可将样本的真实标签 (true label)编码成1-of-K (K为类别总数) 的二元指示矩阵Y:

- 转换举例: 假设K=3, 即三个类

$$Y_{\text{true}} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$
$$Y_{\text{true}_B} = \begin{bmatrix} [1, 0, 0] & [0, 1, 0] & [0, 0, 1] \end{bmatrix}$$

- 假设模型对测试样本的概率估计结果为P, 则在测试集(假设测试样本总数为N)上的交叉熵损失表示如下:

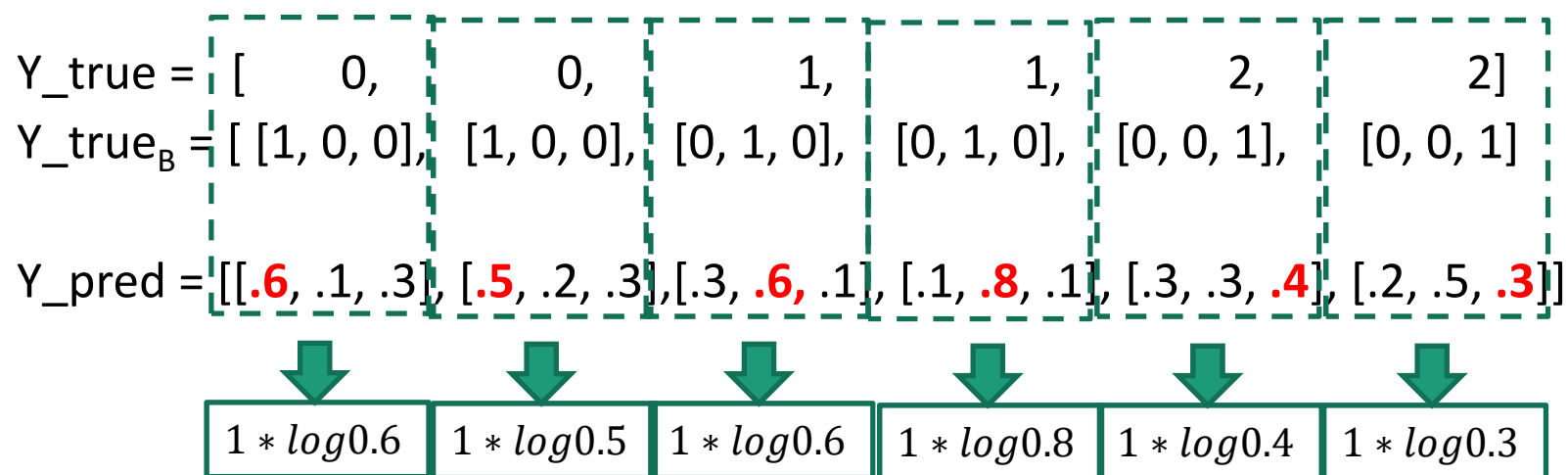
$$L_{\log}(Y, P) = -\frac{1}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{K-1} y_{i,k} \log p_{i,k}$$

其中 $y_{i,k}$ 表示第*i*个样本的第*k*个标签的真实值, 注意由于表示为“1-of-K”模式, 因此每个样本只有其中一个标签值为1, 其余均为0。 $p_{i,k}$ 表示模型对该样本的预测值。

## logistic回归损失(多类)

根据公式,  $L_{log}(Y, P) = -\frac{1}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{K-1} y_{i,k} \log p_{i,k}$

● 举例: 6个样本, 三个类



则  $L_{log}(Y, P) = -(\log 0.6 + \log 0.5 + \log 0.6 + \log 0.8 + \log 0.4 + \log 0.3)/6 = 0.6763$



# 讨论: 回归评价的ground truth如何获得?

- MAE, RMSE(MSE) 常用于评分预测评价

- 很多提供推荐服务的网站都有一个让用户给物品打分的功能。  
预测用户对物品评分的行为称为评分预测。



香橙力娇 Lv4 VIP

★★★★★ 口味: 4 环境: 4 服务: 4 人均: 0元

一如既往的好吃, 点的牛排和汉堡的程度都煎得刚刚好, 酱汁味道浓郁, 一份的量也好大, 吃的够够的。背景音乐超级喜欢, 听着很舒缓、很放松, 让人进餐的时候心情愉悦。另外, 美女经理的服务好好哦, 以后还会经常来的!



11-24 更新于17-11-24 09:11 四季酒店咖啡厅 签到点评

赞 (1) 回应 收藏 举报

# Thanks!





# 一致性的评价方法

(Agreement Evaluation)

刘远超

哈尔滨工业大学

计算机科学与技术学院

# 什么是一致性评价？

- **一致性评价**，是指对两个或多个相关的变量进行分析，从而衡量其相关性的密切程度。



来电狂响



钢铁飞龙之奥...



海王



森林奇缘



闯堂兔3囡囡...

假设两评委 (rater) 对5部电影的评分如下，则二者的一致如何？

rater1 = [0.5, 1.6, 2.5, 2.5, 2.4]

rater2 = [1.5, 2.6, 3.5, 3.5, 3.4]

# 一致性评价--皮尔森相关系数法

- **问题举例：**如何评价两个评委的一致性？

rater1 = [0.5, 1.6, 2.5, 2.5, 2.4]

rater2 = [1.5, 2.6, 3.5, 3.5, 3.4]

- 皮尔森相关系数(Pearson coefficient)的应用背景：

- 用来衡量两个用户之间兴趣的一致性
- 用来衡量预测值与真实值之间的相关性
- 既适用于离散的、也适用于连续变量的相关分析

- X和Y之间的皮尔森相关系数计算公式：

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

其中， $cov(X,Y)$ 表示X和Y之间的协方差 (Covariance)， $\sigma_X$ 是X的均方差， $\mu_X$ 是X的均值， $E$ 表示数学期望

- 取值区间为[-1, 1]。 **-1**: 完全的负相关， **+1**: 表示完全的正相关， **0**: 没有线性相关

# 一致性评价--Cohen's kappa 相关系数

● Cohen's kappa相关系数也可用于衡量两个评价者之间的一致性。其特点在于：

- 与pearson相关系数的区别：Cohen's kappa 相关系数**通常用于离散的**分类的一致性评价。

- 其通常被认为比两人之间的简单一致百分比**更强壮**，因为Cohen's kappa考虑到了二人之间的随机一致的可能性。

●如果评价者多于2人时，可以考虑使用[Fleiss' kappa](https://en.wikipedia.org/wiki/Fleiss%27_kappa).

Reference: [https://en.wikipedia.org/wiki/Joseph\\_L.\\_Fleiss](https://en.wikipedia.org/wiki/Joseph_L._Fleiss).

# Cohen's kappa计算方法

- 假设有50个人申请奖学金。有两个评委A 和 B。每个评委对每个申请者说“Yes” or “No”。假设AB一致性情况如下矩阵所示。

		B	
		Yes	No
A	Yes	a	b
	No	c	d

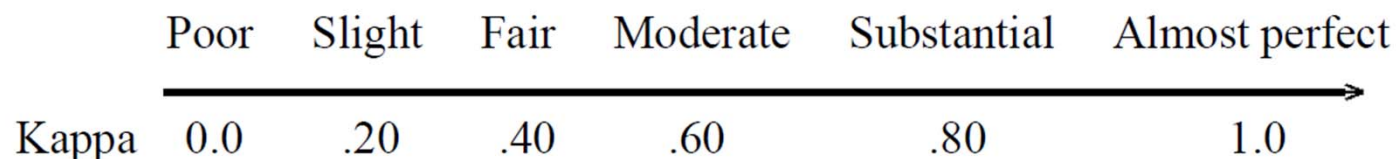
		B	
		Yes	No
A	Yes	20	5
	No	10	15

- 先计算AB一致性比例为  $p_o = \frac{(a+d)}{(a+b+c+d)} = \frac{20+15}{20+5+10+15} = 0.7$
- 再计算AB之间的随机一致性概率  $p_e$  (the probability of random agreement) , 注意到:
  1. A对25个申请者说yes, 因此 比例为25/50=50%
  2. B对30个申请者说yes, 因此比例为30/50=60%
  3. 因此, AB两人随机都说YES的概率  $p_{yes} = \frac{(a+b)}{(a+b+c+d)} \cdot \frac{(a+c)}{(a+b+c+d)} = 0.5 * 0.6 = 0.3$
  4. 同样,  $p_{no} = \frac{(c+d)}{(a+b+c+d)} \cdot \frac{(b+d)}{(a+b+c+d)} = 0.5 * 0.4 = 0.2$
  5. AB的整体随机一致性(Overall random agreement )概率  $p_e = p_{yes} + p_{no} = 0.3+0.2=0.5$
- 最后应用Cohen's kappa公式, 得到  $k = \frac{p_o - p_e}{1 - p_e} = \frac{0.7 - 0.5}{1 - 0.5} = 0.4$

## Cohen's kappa取值的一致性含义

- kappa score是一个介于-1到+1之间的数.

### Interpretation of Kappa



<u>Kappa</u>	<u>Agreement</u>
< 0	Less than chance agreement
0.01–0.20	Slight agreement
0.21– 0.40	Fair agreement
0.41–0.60	Moderate agreement
0.61–0.80	Substantial agreement
0.81–0.99	Almost perfect agreement



## Fleiss' kappa

$n_{ij}$	1	2	3	4	5
1	0	0	0	0	14
2	0	2	6	4	2
3	0	0	3	5	6
4	0	3	9	2	0
5	2	2	8	1	1
6	7	7	0	0	0
7	3	2	6	3	0
8	2	5	3	2	2
9	6	5	2	1	0
10	0	2	2	3	7
Total	20	28	39	21	32
$p_j$	0.143	0.200	0.279	0.150	0.229

以上是14个评价者对10个item进行5级评价的结果（ $N = 10$ ， $n = 14$ ， $k = 5$ ）。

则计算Fleiss Kappa相关系数的过程为：

**step 1.** 对每一列计算 $p_j$ ，即同列数据相加除以任务总数（ $14*10=140$ ）。 $p_j$ 可以理解为每个分类的随机一致概率。

$$\text{以第一列为例，则 } p_1 = \frac{0+0+0+0+2+7+3+2+6+0}{14*10} = 0.143$$

## Fleiss' kappa(续)

$n_{ij}$	1	2	3	4	5	$P_i$
1	0	0	0	0	14	1.000
2	0	2	6	4	2	0.253
3	0	0	3	5	6	0.308
4	0	3	9	2	0	0.440
5	2	2	8	1	1	0.330
6	7	7	0	0	0	0.462
7	3	2	6	3	0	0.242
8	2	5	3	2	2	0.176
9	6	5	2	1	0	0.286
10	0	2	2	3	7	0.286
Total	20	28	39	21	32	
$p_j$	0.143	0.200	0.279	0.150	0.229	

Step 2. 计算 $P_i = \frac{1}{n(n-1)} (\sum_{j=1}^k n_{ij}^2 - n)$ , 即对每一个标注任务进行实际一致性的计算,

以第2个item为例:  $P_2 = \frac{1}{14(14-1)} (0^2 + 2^2 + 6^2 + 4^2 + 2^2 - 14) = 0.253$

## Fleiss' kappa (续)

$n_{ij}$	1	2	3	4	5	$P_i$
1	0	0	0	0	14	1.000
2	0	2	6	4	2	0.253
3	0	0	3	5	6	0.308
4	0	3	9	2	0	0.440
5	2	2	8	1	1	0.330
6	7	7	0	0	0	0.462
7	3	2	6	3	0	0.242
8	2	5	3	2	2	0.176
9	6	5	2	1	0	0.286
10	0	2	2	3	7	0.286
Total	20	28	39	21	32	
$p_j$	0.143	0.200	0.279	0.150	0.229	

Step 3. 计算 $p_o$ 和 $p_e$ :

$$p_o = \frac{1}{N} \sum_{i=1}^N P_i = \frac{1}{10} (1.000 + 0.253 + \dots + 0.286 + 0.286) = \frac{1}{10} \cdot 3.780 = 0.378$$

$$p_e = \sum_{j=1}^k p_j^2 = 0.143^2 + 0.200^2 + 0.279^2 + 0.150^2 + 0.229^2 = 0.213$$

Step 4. 最后计算Fleiss Kappa系数

$$k = \frac{p_o - p_e}{1 - p_e} = \frac{0.378 - 0.213}{1 - 0.213} = 0.210$$

哈尔滨工业大学计算机学院 刘远超

# Thanks!

