




人工智能：模型与算法

# 统计机器学习：监督学习

吴飞

浙江大学计算机学院

# 提纲

- 1、机器学习基本概念
  - 2、线性回归
  - 3、提升算法 (Ada Boost)
- 

# 机器学习：从数据中学习知识

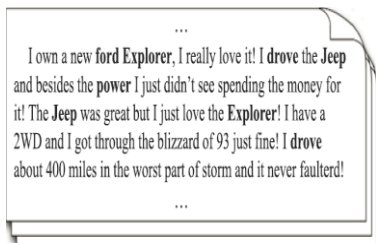


图像数据

$$\swarrow f \left\{ \begin{array}{cccc} 81 & 116 & \dots & 133 \\ 104 & 130 & \dots & 159 \\ \vdots & \vdots & \ddots & \vdots \\ 155 & 189 & \dots & 218 \\ 197 & 221 & \dots & 216 \end{array} \right\}$$

- Person
- Dog
- ...

类别分类



文本数据

$$\swarrow f \{ \text{car, money, drive, ...} \}$$

- 喜悦
- 愤怒
- ...

情感分类

1. 原始数据中提取特征
2. 学习映射函数 $f$
3. 通过映射函数 $f$ 将原始数据映射到语义空间，即寻找数据和任务目标之间的关系

# 机器学习的分类

**监督学习**(supervised learning)

数据有标签、一般为回归或分类等任务

**无监督学习**(un-supervised learning)

数据无标签、一般为聚类或若干降维任务

**强化学习**(reinforcement learning)

序列数据决策学习，一般为与从环境交互中学习

半监督学习  
(semi-supervised learning)

# 机器学习：分类问题

人员	数学好	身体好	会编程	嗓门大
程序员A	Yes	No	Yes	Yes
作家A	No	No	Yes	No
程序员B	Yes	Yes	No	No
...	...	...	...	...
医生A	Yes	Yes	Yes	Yes
程序员C	Yes	Yes	Yes	Yes
程序员D	Yes	Yes	Yes	No

标签数据

从数据  
中学习

映射函数

模式

$f$

(数学好 = Yes, 会编程 = Yes, 身体好 = ?, 嗓门大 = ?)

→ 程序员

类别

# 监督学习的重要元素

标注数据

■ 标识了类别信息的数据

学习模型

■ 如何学习得到映射模型

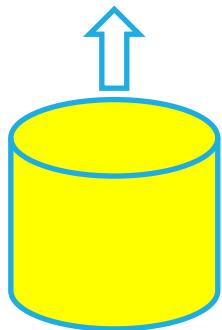
损失函数

■ 如何对学习结果进行度量

# 监督学习：损失函数

训练映射函数 $f$

使得 $f(x_i)$ 预测结果尽量等于 $y_i$



**训练数据集**

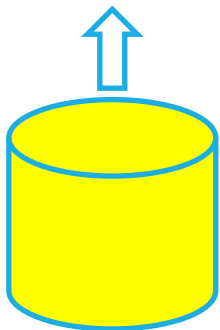
$(x_i, y_i), i = 1, \dots, n$

- 训练集中一共有 $n$ 个标注数据，第 $i$ 个标注数据记为 $(x_i, y_i)$ ，其中第 $i$ 个样本数据为 $x_i$ ， $y_i$ 是 $x_i$ 的标注信息。
- 从训练数据中学习得到的映射函数记为 $f$ ， $f$ 对 $x_i$ 的预测结果记为 $f(x_i)$ 。损失函数就是用来计算 $x_i$ 真实值 $y_i$ 与预测值 $f(x_i)$ 之间差值的函数。
- 很显然，在训练过程中希望映射函数在训练数据集上得到“损失”之和最小，即 $\min \sum_{i=1}^n \text{Loss}(f(x_i), y_i)$ 。

# 监督学习：损失函数

训练映射函数  $f$

使得  $f(x_i)$  预测结果尽量等于  $y_i$



**训练数据集**

$(x_i, y_i), i = 1, \dots, n$

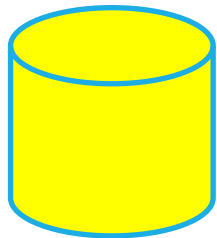
损失函数名称	损失函数定义
0-1损失函数	$Loss(y_i, f(x_i)) = \begin{cases} 1, & f(x_i) \neq y_i \\ 0, & f(x_i) = y_i \end{cases}$
平方损失函数	$Loss(y_i, f(x_i)) = (y_i - f(x_i))^2$
绝对损失函数	$Loss(y_i, f(x_i)) =  y_i - f(x_i) $
对数损失函数/ 对数似然损失 函数	$Loss(y_i, P(y_i x_i)) = -\log P(y_i x_i)$

典型的损失函数



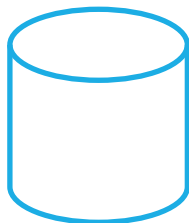
# 监督学习：训练数据与测试数据

从训练数据集学习  
得到映射函数 $f$



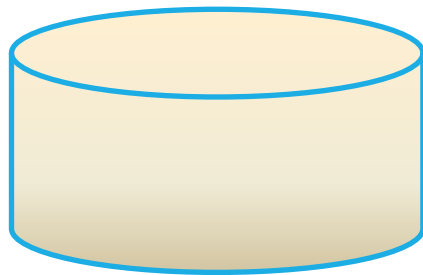
**训练数据集**  
 $(x_i, y_i), i = 1, \dots, n$

在测试数据集  
测试映射函数 $f$



**测试数据集**  
 $(x_i', y_i'), i = 1, \dots, m$

未知数据集  
上测试映射函数 $f$



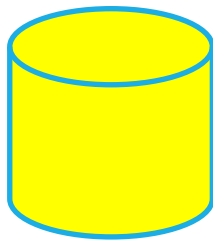
# 监督学习：经验风险与期望风险

从训练数据集学  
习得到映射函数 $f$

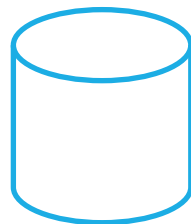
在测试数据集  
测试映射函数 $f$

经验风险(empirical risk)

- 训练集中数据产生的损失。经验风险越小说明学习模型对训练数据拟合程度越好。



训练数据集  
 $(x_i, y_i), i = 1, \dots, n$



测试数据集  
 $(x_i', y_i'), i = 1, \dots, m$

期望风险(expected risk):

- 当测试集中存在无穷多数据时产生的损失。期望风险越小，学习所得模型越好。

# 监督学习：经验风险与期望风险

映射函数训练目标：经验风险最小化  
(empirical risk minimization, ERM)

$$\min_{f \in \Phi} \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i))$$

选取一个使得训练集所有数据  
损失平均值最小的映射函数。  
这样的考虑是否够？



**训练数据集**  
 $(x_i, y_i), i = 1, \dots, n$

映射函数训练目标：期望风险最小化  
(expected risk minimization)

$$\min_{f \in \Phi} \int_{x \times y} \text{Loss}(y, f(x)) P(x, y) dx dy$$



**测试数据集数据无穷多**  
 $(x_i', y_i'), i = 1, \dots, \infty$

- 期望风险是模型关于联合分布期望损失，经验风险是模型关于训练样本集平均损失。
- 根据大数定律，当样本容量趋于无穷时，经验风险趋于期望风险。所以在实践中很自然用经验风险来估计期望风险。
- 由于现实中训练样本数目有限，用经验风险估计期望风险并不理想，要对经验风险进行一定的约束。

# 监督学习: “过学习(over-fitting)” 与 “欠学习(under-fitting)”

## 经验风险最小化

$$\min_{f \in \Phi} \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i))$$

## 期望风险最小化

$$\min_{f \in \Phi} \int_{x \times y} \text{Loss}(y, f(x)) P(x, y) dx dy$$

经验风险小（训练集上表现好）	期望风险小（测试集上表现好）	泛化能力强
经验风险小（训练集上表现好）	期望风险大（测试集上表现不好）	过学习（模型过于复杂）
经验风险大（训练集上表现不好）	期望风险大（测试集上表现不好）	欠学习
经验风险大（训练集上表现不好）	期望风险小（测试集上表现好）	“神仙算法”或“黄粱美梦”

# 监督学习: 结构风险最小

经验风险最小化: 仅反映了局部数据

$$\min_{f \in \Phi} \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i))$$

期望风险最小化: 无法得到全量数据

$$\min_{f \in \Phi} \int_{x \times y} \text{Loss}(y, f(x)) P(x, y) dx dy$$

结构风险最小化(structural risk minimization):

为了防止过拟合, 在经验风险上加上表示模型复杂度的正则化项(regularizer)或惩罚项(penalty term) :

$$\min_{f \in \Phi} \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i)) + \lambda J(f)$$

经验风险                      模型复杂度

在最小化经验风险与降低模型复杂度之间寻找平衡

## 监督学习两种方法：判别模型与生成模型

监督学习方法又可以分为生成方法(generative approach)和判别方法(discriminative approach)。所学到的模型分别称为生成模型(generative model)和判别模型(discriminative model)。

- 判别方法直接学习判别函数 $f(X)$  或者条件概率分布 $P(Y|X)$  作为预测的模型，即判别模型。
- 判别模型关心在给定输入数据下，预测该数据的输出是什么。
- 典型判别模型包括回归模型、神经网络、支持向量机和Ada boosting等。

$$f(\text{人脸}) \longrightarrow \text{人脸}$$

$$P(\text{人脸} | \text{人脸}) = 0.99$$

## 监督学习两种方法：判别模型与生成模型

- 生成模型从数据中学习联合概率分布 $P(X, Y)$ （通过似然概率 $P(X|Y)$ 和类概率 $P(Y)$ 的乘积来求取）

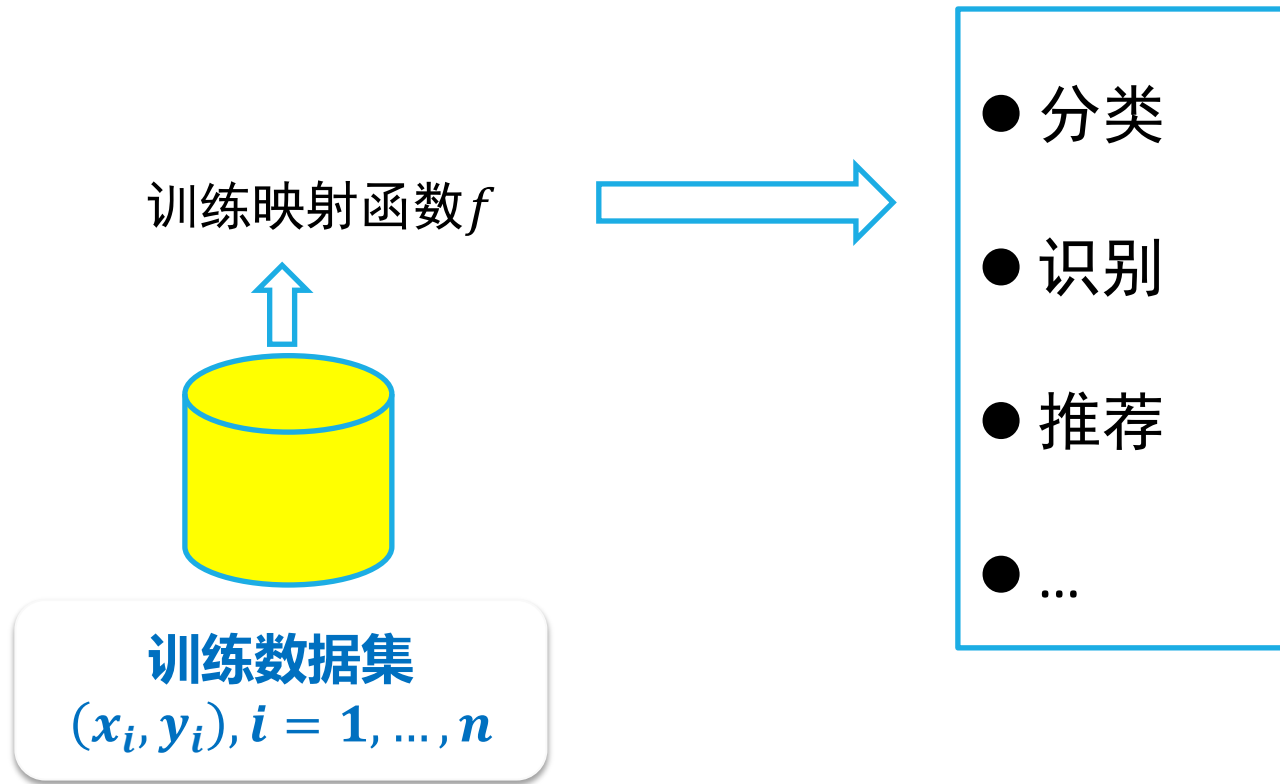
$$P(Y|X) = \frac{P(X, Y)}{P(X)} \text{ 或者 } P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)}$$

- 典型方法为贝叶斯方法、隐马尔可夫链
- 授之于鱼、不如授之于“渔”
- 联合分布概率 $P(X, Y)$ 或似然概率 $P(X|Y)$ 求取很困难

似然概率：计算  
导致样本 $X$ 出现  
的模型参数值

$$P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)}$$

# 监督学习





# 提纲

- 1、机器学习基本概念
- 2、线性回归
- 3、提升算法 (Ada Boost)

## 线性回归 (linear regression)

- 在现实生活中，往往需要分析若干变量之间的关系，如碳排放量与气候变暖之间的关系、某一商品广告投入量与该商品销售量之间的关系等，这种分析不同变量之间存在关系的研究叫回归分析，刻画不同变量之间关系的模型被称为回归模型。如果这个模型是线性的，则称为线性回归模型。
- 一旦确定了回归模型，就可以进行预测等分析工作，如从碳排放量预测气候变化程度、从广告投入量预测商品销售量等。

# 线性回归 (linear regression)

$$y = 33.73(\text{英寸}) + 0.516x$$

$y$ : 子女平均身高

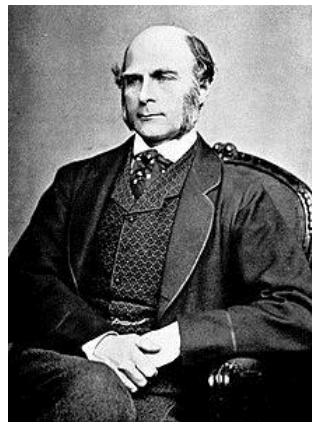
$x$ : 父母平均身高

- 父母平均身高每增加一个单位，其成年子女平均身高只增加0.516个单位，它反映了这种“衰退 (regression)”效应（“回归”到正常人平均身高）。
- 虽然 $x$ 和 $y$ 之间并不总是具有“衰退”（回归）关系，但是“线性回归”这一名称就保留下来了。

课件高尔顿公式

计算身高公式应该是  
 $y=33.73+0.516x$  (单位为英寸)  
这样的才正确吧

zbqdwq126com 19:11



英国著名生物学家兼  
统计学家高尔顿  
Sir Francis Galton  
(1822-1911)

# 线性回归 (linear regression)

该回归模型中两个参数

← 需要从标注数据  
中学习得到  
(监督学习)

$$y = 33.73(\text{英寸}) + 0.516x$$

$y$ : 子女平均身高

$x$ : 父母平均身高

- 给出任意一对父母平均身高，则可根据上述方程，计算得到其子女平均身高
- 从父母平均身高来预测其子女平均身高
- 如何求取上述线性方程（预测方程）的参数？

# 线性回归：参数学习

## 线性回归模型例子

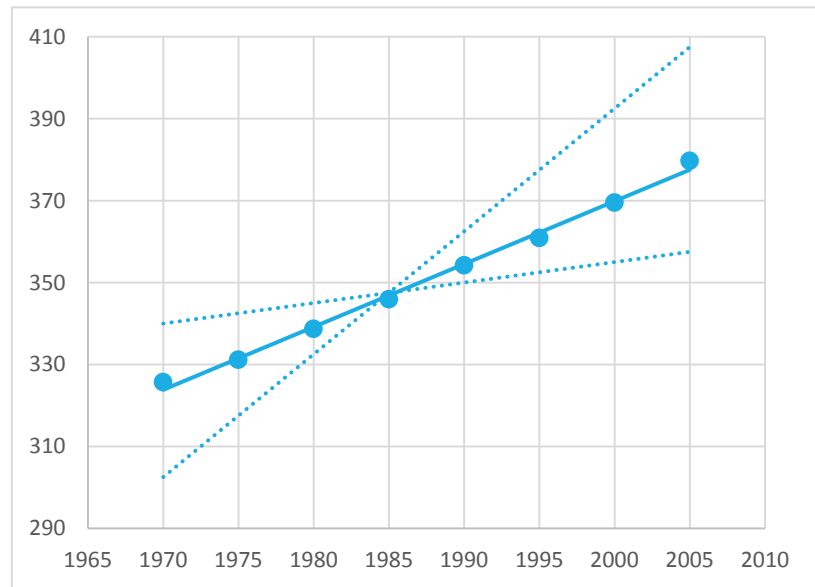
下表给出了莫纳罗亚山（夏威夷岛的活火山）从1970年到2005年每5年的二氧化碳浓度，单位是百万分比浓度（Parts Per Million, ppm）。

年份( $x$ )	1970	1975	1980	1985	1990	1995	2000	2005
CO2 ( $y$ )	325.68	331.15	338.69	345.90	354.19	360.88	369.48	379.67

问题：1) 给出1984年二氧化碳浓度值；2) 预测2010年二氧化碳浓度值

# 线性回归：参数学习

## 线性回归模型例子



莫纳罗亚山地区时间年份与二氧化碳浓度之间的一元线性回归模型（实线为最佳回归模型）

年份(x)	1970	1975	1980	1985	1990	1995	2000	2005
CO2 (y)	325.68	331.15	338.69	345.90	354.19	360.88	369.48	379.67



代入

回归模型： $y = ax + b$

求取：最佳回归模型是最小化残差平方和的均值，即要求8组 $(x, y)$ 数据得到的残差平均值 $\frac{1}{N} \sum (y - \hat{y})^2$ 最小。残差平均值最小只与参数 $a$ 和 $b$ 有关，最优解即是使得残差最小所对应的 $a$ 和 $b$ 的值。

# 线性回归：参数学习

回归模型参数求取： $y_i = ax_i + b$  ( $1 \leq i \leq n$ )

- 记在当前参数下第 $i$ 个训练样本 $x_i$ 的预测值为 $\hat{y}_i$
- $x_i$ 的标注值（实际值） $y_i$ 与预测值 $\hat{y}_i$ 之差记为 $(y_i - \hat{y}_i)^2$
- 训练集中 $n$ 个样本所产生误差总和为： $L(a, b) = \sum_{i=1}^n (y_i - a \times x_i - b)^2$

目标：寻找一组 $a$ 和 $b$ ，使得误差总和 $L(a, b)$ 值最小。在线性回归中，解决如此目标的方法叫最小二乘法。

一般而言，要使函数具有最小值，可对 $L(a, b)$  参数 $a$ 和 $b$ 分别求导，令其导数值为零，再求取参数 $a$ 和 $b$ 的取值。

# 线性回归：参数学习

回归模型参数求取： $y_i = ax_i + b$  ( $1 \leq i \leq n$ )

$$\frac{\partial L(a, b)}{\partial a} = \sum_{i=1}^n 2(y_i - ax_i - b)(-x_i) = 0$$

将  $b = \bar{y} - a\bar{x}$  ( $\bar{y} = \frac{\sum_{i=1}^n y_i}{n}$ ,  $\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ )

代入上式

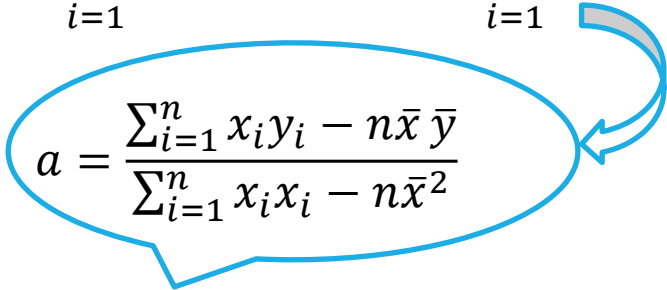
$$\rightarrow \sum_{i=1}^n (y_i - ax_i - \bar{y} + a\bar{x})(x_i) = 0$$

$$\rightarrow \sum_{i=1}^n (y_i x_i - ax_i x_i - \bar{y} x_i + a\bar{x} x_i) = 0$$

$$\min_{a, b} L(a, b) = \sum_{i=1}^n (y_i - a \times x_i - b)^2$$

$$\rightarrow \sum_{i=1}^n (y_i x_i - \bar{y} x_i) - a \sum_{i=1}^n (x_i x_i - \bar{x} x_i) = 0$$

$$\rightarrow \left( \sum_{i=1}^n x_i y_i - n\bar{x}\bar{y} \right) - a \left( \sum_{i=1}^n x_i x_i - n\bar{x}^2 \right) = 0$$


$$a = \frac{\sum_{i=1}^n x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^n x_i x_i - n\bar{x}^2}$$



# 线性回归：参数学习

回归模型参数求取： $y_i = ax_i + b$  ( $1 \leq i \leq n$ )  $\min_{a,b} L(a,b) = \sum_{i=1}^n (y_i - a \times x_i - b)^2$

$$\frac{\partial L(a,b)}{\partial b} = \sum_{i=1}^n 2(y_i - ax_i - b)(-1) = 0$$

$$\rightarrow \sum_{i=1}^n (y_i - ax_i - b) = 0$$

$$\rightarrow \sum_{i=1}^n (y_i) - a \sum_{i=1}^n x_i - \sum_{i=1}^n b = 0$$

$$\rightarrow n\bar{y} - an\bar{x} - nb = 0$$

$$b = \bar{y} - a\bar{x}$$

$$a = \frac{\sum_{i=1}^n x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^n x_i^2 - n\bar{x}^2}$$

可以看出：只要给出了训练样本 $(x_i, y_i)$  ( $i = 1, \dots, n$ )，我们就可以从训练样本出发，建立一个线性回归方程，使得对训练样本数据而言，该线性回归方程预测的结果与样本标注结果之间的差值和最小。

# 线性回归：参数学习

回归模型参数求取： $y_i = ax_i + b$  ( $1 \leq i \leq n$ )  $\min_{a,b} L(a,b) = \sum_{i=1}^n (y_i - a \times x_i - b)^2$

$$b = \bar{y} - a\bar{x}$$

$$a = \frac{\sum_{i=1}^n x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^n x_i^2 - n\bar{x}^2}$$

年份(x)	1970	1975	1980	1985	1990	1995	2000	2005
CO2(y)	325.68	331.15	338.69	345.90	354.19	360.88	369.48	379.67

训练样本数据



$$a = \frac{x_1 y_1 + x_2 y_2 + \dots + x_8 y_8 - 8\bar{x}\bar{y}}{x_1^2 + x_2^2 + \dots + x_8^2 - 8\bar{x}^2} = 1.5344$$

$$b = \bar{y} - a\bar{x} = -2698.9$$

预测莫纳罗亚山地区二氧化碳浓度的一元线性回归模型为“二氧化碳浓度 =  $1.5344 \times \text{时间年份} - 2698.9$ ”，即  $y = 1.5344x - 2698.9$ 。

# 提纲

- 1、机器学习基本概念
- 2、线性回归
- 3、提升算法 (Ada Boost)

# Boosting (adaptive boosting, 自适应提升)

From [Adaptive Computation and Machine Learning](#)

## Boosting

Foundations and Algorithms

By [Robert E. Schapire](#) and [Yoav Freund](#)

### Overview

*Boosting* is an approach to machine learning based on the idea of creating a highly accurate predictor by combining many weak and inaccurate “rules of thumb.” A remarkably rich theory has evolved around boosting, with connections to a range of topics, including statistics, game theory, convex optimization, and information geometry. Boosting algorithms have also enjoyed practical success in such fields as biology, vision, and speech processing. At various times in its history, boosting has been perceived as mysterious, controversial, even paradoxical.

This book, written by the inventors of the method, brings together, organizes, simplifies, and substantially extends two decades of research on boosting, presenting both theory and applications in a way that is accessible to readers from diverse backgrounds while also providing an authoritative reference for advanced researchers. With its introductory treatment of all material and its inclusion of exercises in every chapter, the book is appropriate for course use as well.

The book begins with a general introduction to machine learning algorithms and their analysis; then explores the core theory of boosting, especially its ability to generalize; examines some of the myriad other theoretical viewpoints that help to explain and understand boosting; provides practical extensions of boosting for more complex learning problems; and finally presents a number of advanced theoretical topics. Numerous applications and practical illustrations are offered throughout.

- 对于一个复杂的分类任务，可以将其分解为若干子任务，然后将若干子任务完成方法综合，最终完成该复杂任务。
- 将若干个弱分类器(weak classifiers)组合起来，形成一个强分类器(strong classifier)。
- 能用众力，则无敌于天下矣；**能用众智，则无畏于圣人矣**(语出《三国志 吴志 孙权传》)

Freund, Yoav; Schapire, Robert E (1997), A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences (original paper of Yoav Freund and Robert E.Schapire where AdaBoost is first introduced.)

# 计算学习理论 ( Computational Learning Theory)

- 可计算：什么任务是可以计算的？图灵可停机
- 可学习：什么任务是可以被学习的、从而被学习模型来完成？
- Leslie Valiant (2010年图灵奖获得者)和其学生Michael Kearns 两位学者提出了这个问题并进行了有益探索，逐渐完善了计算学习理论。

## 计算学习理论：霍夫丁不等式(Hoeffding's inequality)

- 学习任务：统计某个电视节目在全国的收视率。
- 方法：不可能去统计整个国家中每个人是否观看电视节目、进而算出收视率。  
只能抽样一部分人口，然后将抽样人口中观看该电视节目的比例作为该电视节目的全国收视率。
- 霍夫丁不等式：全国人口中看该电视节目的人口比例（记作 $x$ ）与抽样人口中观看该电视节目的人口比例（记作 $y$ ）满足如下关系：

$$P(|x - y| \geq \epsilon) \leq 2e^{-2N\epsilon^2} (N \text{ 是采样人口总数、} \epsilon \in (0,1) \text{ 是所设定的可容忍误差范围})$$

当 $N$ 足够大时，“全国人口中电视节目收视率”与“样本人口中电视节目收视率”差值超过误差范围 $\epsilon$ 的概率非常小。

# 计算学习理论：概率近似正确 (probably approximately correct, PAC)

- 对于统计电视节目收视率这样的任务，可以通过不同的采样方法（即不同模型）来计算收视率。
- 每个模型会产生不同的误差。
- 问题：如果得到完成该任务的若干“弱模型”，是否可以将这些弱模型组合起来，形成一个“强模型”。该“强模型”产生误差很小呢？这就是概率近似正确（PAC）要回答的问题。

# 计算学习理论：概率近似正确 (probably approximately correct, PAC)

在概率近似正确背景下，有“强可学习模型”和“弱可学习模型”

<b>强可学习</b> (strongly learnable)	● 学习模型能够以较高精度对绝大多数样本完成识别分类任务
<b>弱可学习</b> (weakly learnable)	学习模型仅能完成若干部分样本识别与分类，其精度略高于随机猜测。
强可学习和弱可学习是等价的，也就是说，如果已经发现了“弱学习算法”，可将其提升（boosting）为“强学习算法”。Ada Boosting算法就是这样的方法。具体而言，Ada Boosting将一系列弱分类器组合起来，构成一个强分类器。	



## Ada Boosting : 思路描述

- Ada Boosting算法中两个核心问题：
  - 在每个弱分类器学习过程中，如何改变训练数据的权重：提高在上一轮中分类错误样本的权重。
  - 如何将一系列弱分类器组合成强分类器：通过加权多数表决方法来提高分类误差小的弱分类器的权重，让其在最终分类中起到更大作用。同时减少分类误差大的弱分类器的权重，让其在最终分类中仅起到较小作用。

## Ada Boosting : 算法描述---数据样本权重初始化

- 给定包含 $N$ 个标注数据的训练集合 $\Gamma$ ,  $\Gamma = \{(x_1, y_1), \dots, (x_N, y_N)\}$ 。

$$x_i (1 \leq i \leq N) \in X \subseteq R^n, y_i \in Y = \{-1, 1\}$$

- Ada Boosting算法将从这些标注数据出发, 训练得到一系列弱分类器, 并将这些弱分类器线性组合得到一个强分类器。

### 1. 初始化每个训练样本的权重

$$D_1 = (w_{11}, \dots, w_{1i}, \dots, w_{1N}), \text{ 其中 } w_{1i} = \frac{1}{N} (1 \leq i \leq N)$$

## Ada Boosting : 算法描述---第 $m$ 个弱分类器训练

2. 对 $m = 1, 2, \dots, M$

a) 使用具有分布权重 $D_m$ 的训练数据来学习得到第 $m$ 个基分类器（弱分类器） $G_m$ ：

$$G_m(x): X \rightarrow \{-1, 1\}$$

b) 计算 $G_m(x)$ 在训练数据集上的分类误差

$err_m = \sum_{i=1}^N w_{mi} I(G_m(x_i) \neq y_i)$  这里：  $I(\cdot) = 1$ ，如果 $G_m(x_i) \neq y_i$ ； 否则为0

c) 计算弱分类器 $G_m(x)$ 的权重：  $\alpha_m = \frac{1}{2} \ln \frac{1-err_m}{err_m}$

d) 更新训练样本数据的分布权重：  $D_{m+1} = w_{m+1,i} = \frac{w_{m,i}}{Z_m} e^{-\alpha_m y_i G_m(x_i)}$ ，其中 $Z_m$ 是归一

化因子以使得 $D_{m+1}$ 为概率分布，  $Z_m = \sum_{i=1}^N w_{m,i} e^{-\alpha_m y_i G_m(x_i)}$

## Ada Boosting : 算法描述---弱分类器组合成强分类器

3. 以线性加权形式来组合弱分类器 $f(x)$

$$f(x) = \sum_{i=1}^M \alpha_m G_m(x)$$

得到强分类器 $G(x)$

$$G(x) = \text{sign}(f(x)) = \text{sign}\left(\sum_{i=1}^M \alpha_m G_m(x)\right)$$

# Ada Boosting : 算法解释

第 $m$ 个弱分类器 $G_m(x)$ 在训练数据集上产生的分类误差：

该误差为被错误分类的样本所具有权重的累加

$$err_m = \sum_{i=1}^N w_{m,i} I(G_m(x_i) \neq y_i) \quad \text{这里: } I(\cdot) = 1, \text{ 如果 } G_m(x_i) \neq y_i; \text{ 否则为 } 0$$

# Ada Boosting : 算法解释

计算第 $m$ 个弱分类器 $G_m(x)$ 的权重 $\alpha_m$ :  $\alpha_m = \frac{1}{2} \ln \frac{1-err_m}{err_m}$

(a) 当第 $m$ 个弱分类器 $G_m(x)$  错误率为1, 即 $err_m = \sum_{i=1}^N w_{mi} I(G_m(x_i) \neq y_i) = 1$ , 意味每个

样本分类出错, 则 $\alpha_m = \frac{1}{2} \ln \frac{1-err_m}{err_m} \rightarrow -\infty$ , 给予第 $m$ 个弱分类器 $G_m(x)$ 很低权重。

(b) 当第 $m$ 个弱分类器 $G_m(x)$ 错误率为 $\frac{1}{2}$ ,  $\alpha_m = \frac{1}{2} \ln \frac{1-err_m}{err_m} = 0$ 。如果错误率 $err_m$ 小于 $\frac{1}{2}$ , 权

重 $\alpha_m$ 为正( $err_m < \frac{1}{2}$ 、 $\alpha_m > 0$ )。可知权重 $\alpha_m$ 随 $err_m$ 减少而增大, 即错误率越小的弱分

类器会赋予更大权重。

(c) 如果一个弱分类器的分类错误率为 $\frac{1}{2}$ , 可视为其性能仅相当于随机分类效果。

# Ada Boosting : 算法解释

在开始训练第 $m + 1$ 个弱分类器 $G_{m+1}(x)$ 之前对训练数据集中数据权重进行调整

$$w_{m+1,i} = \begin{cases} \frac{w_{m,i}}{Z_m} e^{-\alpha_m}, & G_m(x_i) = y_i \\ \frac{w_{m,i}}{Z_m} e^{\alpha_m}, & G_m(x_i) \neq y_i \end{cases}$$

- 可见，如果某个样本无法被第 $m$ 个弱分类器 $G_m(x)$ 分类成功，则需要增大该样本权重，否则减少该样本权重。这样，被错误分类样本会在训练第 $m + 1$ 个弱分类器 $G_{m+1}(x)$ 时会被“重点关注”。
- 在每一轮学习过程中，Ada Boosting算法均在划重点（重视当前尚未被正确分类的样本）

# Ada Boosting : 算法解释

## 弱分类器构造强分类器

$$f(x) = \sum_{i=1}^M \alpha_m G_m(x)$$

$$G(x) = \text{sign}(f(x)) = \text{sign}\left(\sum_{i=1}^M \alpha_m G_m(x)\right)$$

- $f(x)$ 是 $M$ 个弱分类器的加权线性累加。分类能力越强的弱分类器具有更大权重。
- $\alpha_m$ 累加之和并不等于1。
- $f(x)$ 符号决定样本 $x$ 分类为1或-1。如果 $\sum_{i=1}^M \alpha_m G_m(x)$ 为正，则强分类器 $G(x)$ 将样本 $x$ 分类为1；否则为-1。



# Ada Boosting : 回看霍夫丁不等式

假设有 $M$ 个弱分类器 $G_m(1 \leq m \leq M)$ , 则 $M$ 个弱分类器线性组合所产生误差满足如下条件:

$$P\left(\sum_{i=1}^M G_m(x) \neq \zeta(x)\right) \leq e^{-\frac{1}{2}M(1-2\epsilon)^2}$$

- $\zeta(x)$  是真实分类函数、 $\epsilon \in (0,1)$ 。上式表明, 如果所“组合”弱分类器越多, 则学习分类误差呈指数级下降, 直至为零。
- 上述不等式成立有两个前提条件: 1) 每个弱分类器产生的误差相互独立; 2) 每个弱分类器的误差率小于50%。因为每个弱分类器均是在同一个训练集上产生, 条件1) 难以满足。也就是说, “准确性(对分类结果而言)”和“差异性(对每个弱分类器而言)”难以同时满足。
- Ada Boosting 采取了序列化学习机制。

# Ada Boosting : 优化目标

Ada Boost实际在最小化如下指数损失函数(minimization of exponential loss ):

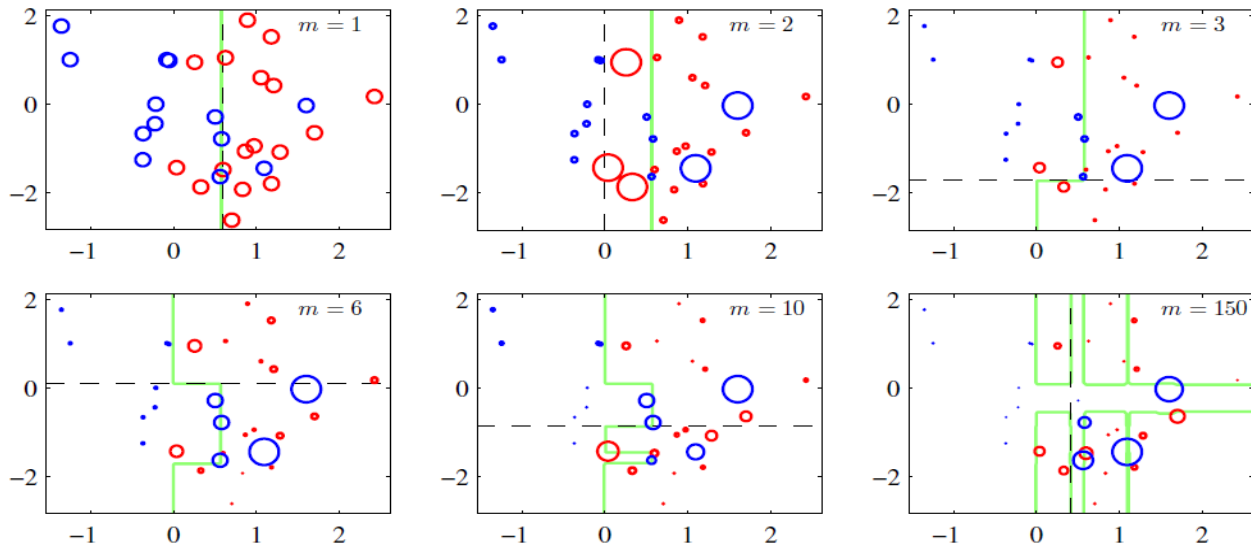
$$\sum_i e^{-y_i f(x_i)} = \sum_i e^{-y_i \sum_{m=1}^M \alpha_m G_m(x_i)}$$

Ada Boost的分类误差上界如下所示:

$$\frac{1}{N} \sum_{i=1}^N I(G(x_i) \neq y_i) \leq \frac{1}{N} \sum_i e^{-y_i f(x_i)} = \prod_m Z_m$$

- 在第 $m$ 次迭代中, Ada Boosting总是趋向于将具有最小误差的学习模型选做本轮生成的弱分类器  $G_m$ , 使得累积误差快速下降。

# Ada Boosting : 算法例子

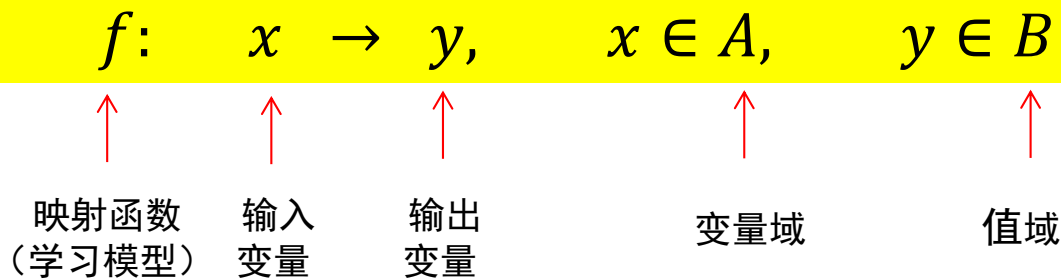


**Figure 14.2** Illustration of boosting in which the base learners consist of simple thresholds applied to one or other of the axes. Each figure shows the number  $m$  of base learners trained so far, along with the decision boundary of the most recent base learner (dashed black line) and the combined decision boundary of the ensemble (solid green line). Each data point is depicted by a circle whose radius indicates the weight assigned to that data point when training the most recently added base learner. Thus, for instance, we see that points that are misclassified by the  $m = 1$  base learner are given greater weight when training the  $m = 2$  base learner.

摘自《Pattern Recognition and Machine Learning  
(by Christopher M. Bishop)》第14章

# 回归与分类的区别

- 两者均是学习输入变量和输出变量之间潜在关系模型，基于学习所得模型将输入变量映射到输出变量。



- 监督学习分为回归和分类两个类别。
- 在回归分析中，学习得到一个函数将输入变量映射到连续输出空间，如价格和温度等，即值域是连续空间。
- 在分类模型中，学习得到一个函数将输入变量映射到离散输出空间，如人脸和汽车等，即值域是离散空间。