第7章
模型选择
Model Selection

向 世 明
smxiang@nlpr.ia.ac.cn
中科院自动化研究所 模式识别国家重点实验室

助教： 何文浩 (wenhao.he@nlpr.ia.ac.cn)
杨红明 (hongming.yang@nlpr.ia.ac.cn)

内容提要

- 引言
- 模型选择的相关定理和原则
- 模型选择的评价标准
- 分类器设计中的重采样技术
- 分类器集成
  - Adaboost，…

7.1 引言

- **模型**
  - 一是指**模型类别**，比如“所有的含有两个或三个成分的高斯混合模型之集合”。
  - 二是指**模型类别固定，但模型带有一些参数**，这些参数可变。
- **模型选择**
  - 估计不同模型的性能，以便从中选择最好的模型
- **模型评估**
  - 从选定的模型中，估计新样本的预测值（泛化误差）

7.1 引言

- **机器学习**
  - **从数据中学习决策函数、规则、知识**
  - **学习=表示 + 评价+ 优化**
  - **表示：**
    - 一个分类器必须用计算机可以处理的某种形式语言来表示。**为学习器选择一种表示，就意味选择一个特定的分类器集合**。
    - 学习器可能学出的分类器只能在这个集合中。这个集合被称为学习器的假设空间（hypothesis space）。如果某个分类器不在该空间中，它就不可能被该学习器学到。

7.1 引言

- **机器学习**
  - **评价：**
    - 需要一个评价函数（亦称为目标函数或打分函数）来判断分类器的优劣。
  - **优化：**
    - 需要一个搜索方法，能够在假设空间中找到评价函数得分最高的那个分类器。

7.1 引言

- 机器学习

表示	准则	优化
—近邻法	—准确率/错误率	—组合优化
—支持向量机	—召回率	—贪心搜索
—超平面分类器	—平方误差	—连续优化
—朴素贝叶斯分类	—后验概率	—无/有约束优化
—逻辑斯蒂回归	—似然	—梯度下降
—神经网络	—信息增益	—共轭梯度
—决策树	—KL距离	—线性规划
—图模型	—利润	—二次规划
—贝叶斯网络	—成本/效用	—半正定规划
—命名规则		

7.1 引言

- **机器学习：泛化**
  - 机器学习的基本目标是对训练集合中样例的泛化。这是因为，不管我们有多少训练数据，在测试阶段这些数据都不太可能会重复出现。
  - **泛化的目的是使学习器在处理新数据时表现出色**。
  - 将泛化作为目标带来的另外一个重要结果是，仅有数据还不够，无论你有多少。

每个学习器都需要包含一些**数据之外的知识或假设**（**assumption**），才能够将数据泛化。这一概念被沃尔伯特（**Wolpert**）形式化为“**没有免费的午餐**”定理

7.1 引言

- **机器学习：过拟合**
  - 如果拥有的知识和数据并不足以学习出正确的分类器，将会怎样呢？
    - 所建分类器并非建立在现实基础上，这是一个风险。
    - 比如，训练数据上的正确率达到100%，测试数据上的正确率只有50%。

7.1 引言

- **机器学习：维数灾难**
  - 许多在低维空间表现很好的算法，在处理高维数据时**性能急剧下降**。
  - 随着样本维度的增长，正确泛化的难度会以指数级增加，因为**同等规模的训练集只能覆盖越来越少的空间比例**。
  - 感谢“**非均匀性的祝福**”（blessing of nonuniformity）！
    - 在大多数应用中，样本在空间中并非均匀分布，而是集中在一个低维流形（manifold）上面或附近。

7.1 引言

- **机器学习：理论上的保证**
  - 泛化所需样本数目的边界（bound）
  - 渐进（asymptotic）正确：给定无穷数据，学习器将保证输出正确的分类器
    - 但是如果仅仅因为有渐进保证就选择一个分类器则是非常草率的。**在实践中，我们很少处于渐进状态。**
    - 如果对无穷数据，学习器 A 比学习器B好，那么在有限数据的情况下 B 可能（通常）比 A 好。
  - 理论保证的主要作用**并不是在实践中作为决策的标准，而是作为理解算法的动机和来源**。

7.1 引言

- **机器学习：相关并不意味因果**
  - 通常，人们学习预测模型的目的是将其作为某种行动指南 (比如广告推荐、路径规划与推荐)。
  - 如果我们发现超市里的**啤酒和尿布**经常被一起购买，那将啤酒放在尿布旁边将会提高销售量。
  - 但除非真的做实验，不然很难发现这一点。

7.2 模型选择原则

- **有很多机器学习方法，哪一个最好？**
  - 一些机器学习方法**计算复杂度低**，一些**可更好地应用先验知识**，另一些则具有**更好的分类或聚类性能**。
  - 没有任何一个分类方法一定优于其它方法！
    - 之所以有些要做得好一些，与问题本身、先验分布以及其它知识有关
  - **实践：对于在训练集上表现同样良好的两个分类器，人们会更喜欢越简单的那个分类器**，甚至期望它对于测试数据会做出更好的结果。

有无一些独立于算法的分类器设计规则？

7.2 模型选择原则

- **没有免费的午餐定理（No Free Lunch, NFL)**
  - 1995年，David H. Wolpert和William G. Macready 提出NFL定理：对“寻找代价函数极值”的算法，**在平均到所有可能的代价函数上时，其表现都恰好相同**。
    - **由于对“所有可能的函数”的相互补偿，最优化算法的性能是等价的。**暗示：没有其它任何算法能够比搜索(有限)空间的线性列举或者纯随机搜索更优。
    - **对于整个函数集(类)而言，不存在万能的最佳算法**。所有算法在整个函数集的平均表现度量是一样的。
      - 特别地，如果算法A在一些代价函数上优于算法B，那么存在一些其它函数，使B优于A。

7.2 模型选择原则

- 没有免费的午餐定理
  - NFL定理的主要价值：观念性启示作用
    - 研究目标不应该走向“寻找万能算法”之路。
    - 以算法为导向，从算法到问题。对于每一个算法，都有其适用和不适用的问题；我们要尽可能通过理论分析，给出其适用问题的范围。
    - 以问题为导向，从问题到算法。对于一个小的特定的函数集，或者一个特定的实际问题，应设计专门适用的算法。

7.2 模型选择原则

- **没有免费的午餐定理：对于机器学习**
  - 学习算法必须要引入一些**与问题领域有关的假设**。
  - **不存在一个与具体应用无关的、普遍适用的“最优分类器”**。
  - 在没有假设的前提下，我们没有理由**偏爱某一学习或分类算法而轻视另一个**。
  - 要想在某些指标上得到性能的提高，必须在另一些指标上付出相应的代价。因此，**要求我们深刻地理解所研究对象的本质**：
    - 先验知识、数据分布、训练数据量、代价准则

7.2 模型选择原则

- **丑小鸭定理（Ugly Ducking)**
  - **1969年，**Watanabe Satosi 提出。
  - **丑小鸭与白天鹅。**
  - **人们对苹果的认知——“我认为最重要的特征”，只代表个人立场，其他人没有赞同的理由。**
  - 世界上不存在分类的客观标准，一切分类的标准都是主观的。
  - 机器学习
    - **不存在与问题无关的最优的特征/属性集合。**
    - 不存在与问题无关的模式之间的**“相似性度量”。**

7.2 模型选择的相关定理和原则

- **Occam剃刀原理（Occam's Razor）**
  - 由14世纪逻辑学家**Occam提出。**
  - 如无必要，勿增实体——即简单有效原理。
  - 切勿浪费较多东西去做“用较少的东西也可以做好的事情”。
  - 无情地剔除所有累赘。
    - 牛顿：如果某一原因既真实又足以解释自然事物的特性，则我们不应当接受比这个更多的其它原因
    - 爱因斯坦：万事万物都应尽可能简洁，但不能过于简单（简约而不简单）

7.2 模型选择原则

- **Occam剃刀原理：对于机器学习**
  - 设计者不应该选用比“必要”更加复杂的分类器。
    - “必要：是指由训练数据的拟合情况决定的。
  - 如果对训练数据分类的效果相同，“简单的”分类器往往优于“复杂的”分类器。

7.2 模型选择原则

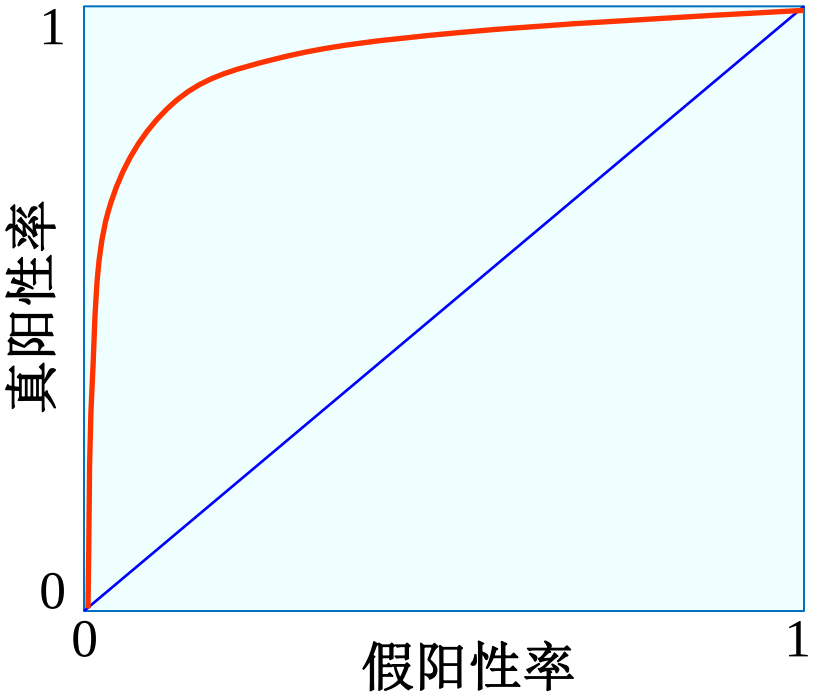
- **最小描述长度原理**（Minimum Description Length, MDL）
  - 我们必须使**模型的算法复杂度**、以及**与该模型相适应的训练数据的描述长度**之和最小。也就是说，我们应该选择尽可能简单的分类器或模型
    - 赤池信息量准则，即Akaike information criterion (AIC)
    - 贝叶斯信息准则，即Bayesian information criterion (BIC)
    - 网络信息准则，即Network Information Criterion (NIC)

7.3 模型评价标准

- **统计检验方法**
  - 应该意识到**两个分类器的准确率上的差异可能不是统计显著的**，因为分类器的准确率与测试样本的数量有很大的关系。
  - 比如，采用统计检验方法－ **student t-分布检验**
  - 又如，采用 **Critical difference (CD) diagram of different models**
- **偏差和方差**
  - 这两个量是评价学习算法与给定分类问题的“匹配”和“校准”程度的通用方法。
  - 偏差用于评价“匹配”的准确性和质量；
  - 方差用于评价“匹配”的精确性。

7.3 模型评价标准

- 操作接收者特征曲线 ROC (对两类分类问题)
  - Receiver Operating Characteristic（越高越好）

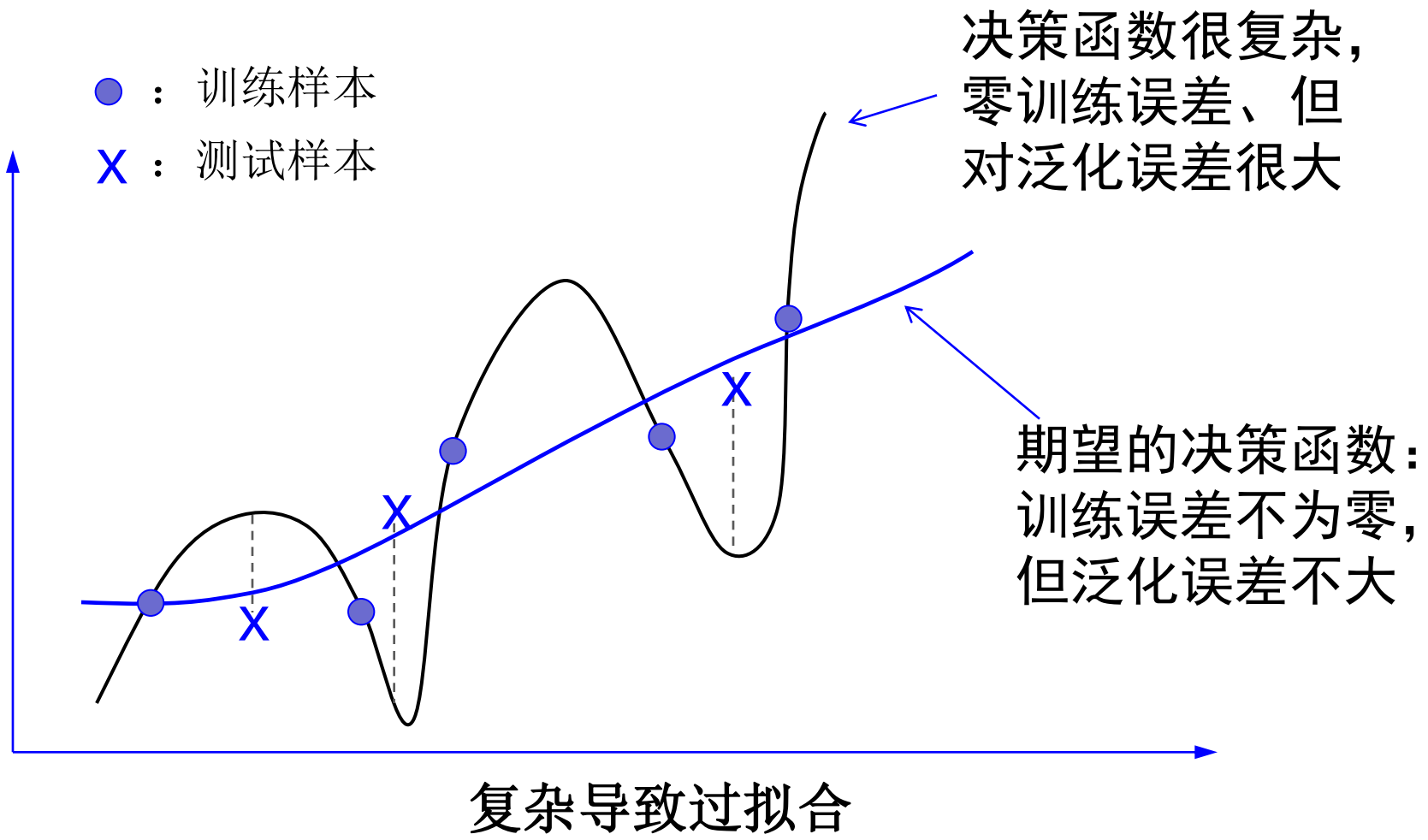


$$\text{真阳性率} = \frac{\text{真阳性个数}}{\text{真阳性}+\text{假阴性个数}}$$

$$\text{假阳性率} = \frac{\text{真阴性个数}}{\text{真阴性}+\text{假阳性个数}}$$

7.3 模型评价标准

- **模型的复杂度**



7.3 模型评价标准

- **模型的复杂度**

训练误差：

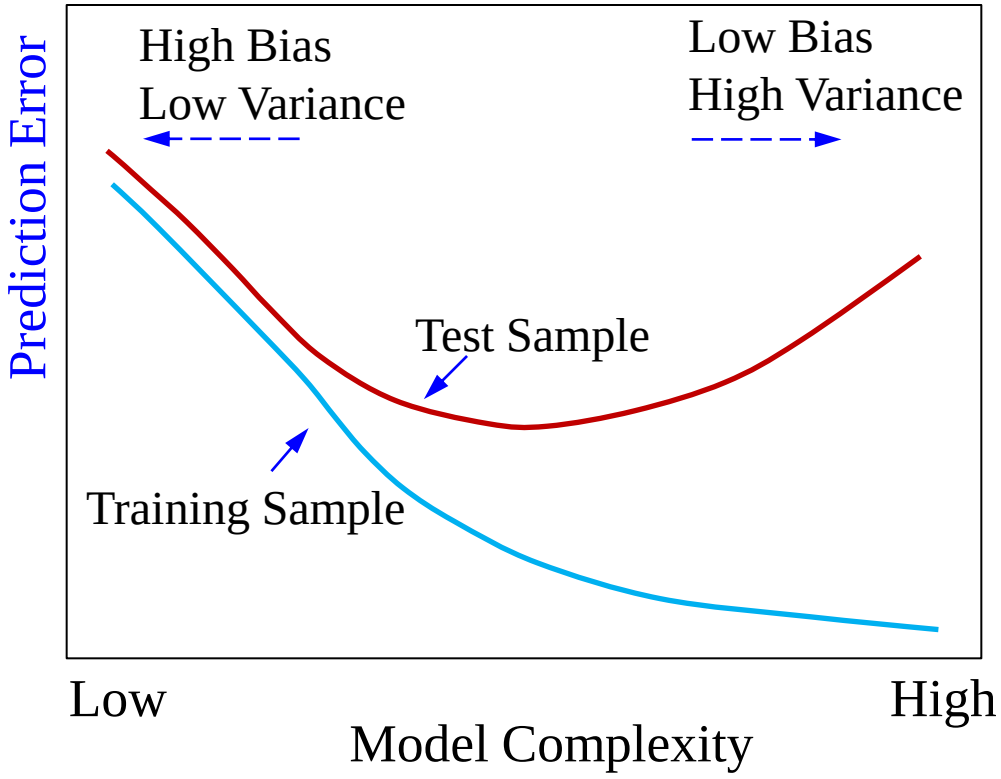
$$e_{train} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i))$$

L(.): 损失函数
f: 学习器
$(\mathbf{x}_i, y_i)$: 训练样本

泛化误差：

$$e_{test} = E(L(Y, f(X))$$

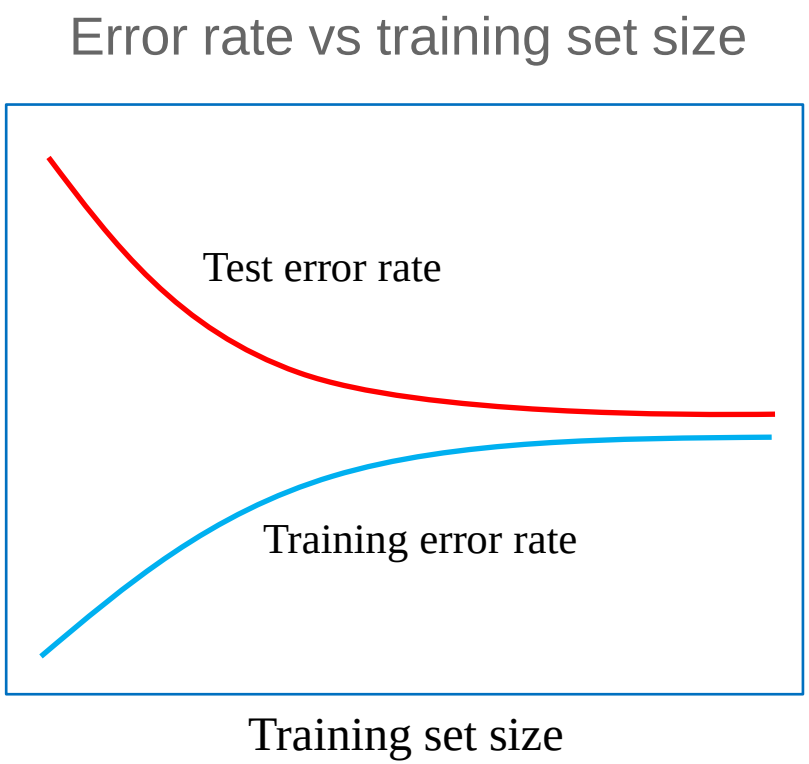
训练误差不是“泛化误差的一种好的估计”



7.3 模型评价标准

- **训练数据的多少对泛化性能的影响**

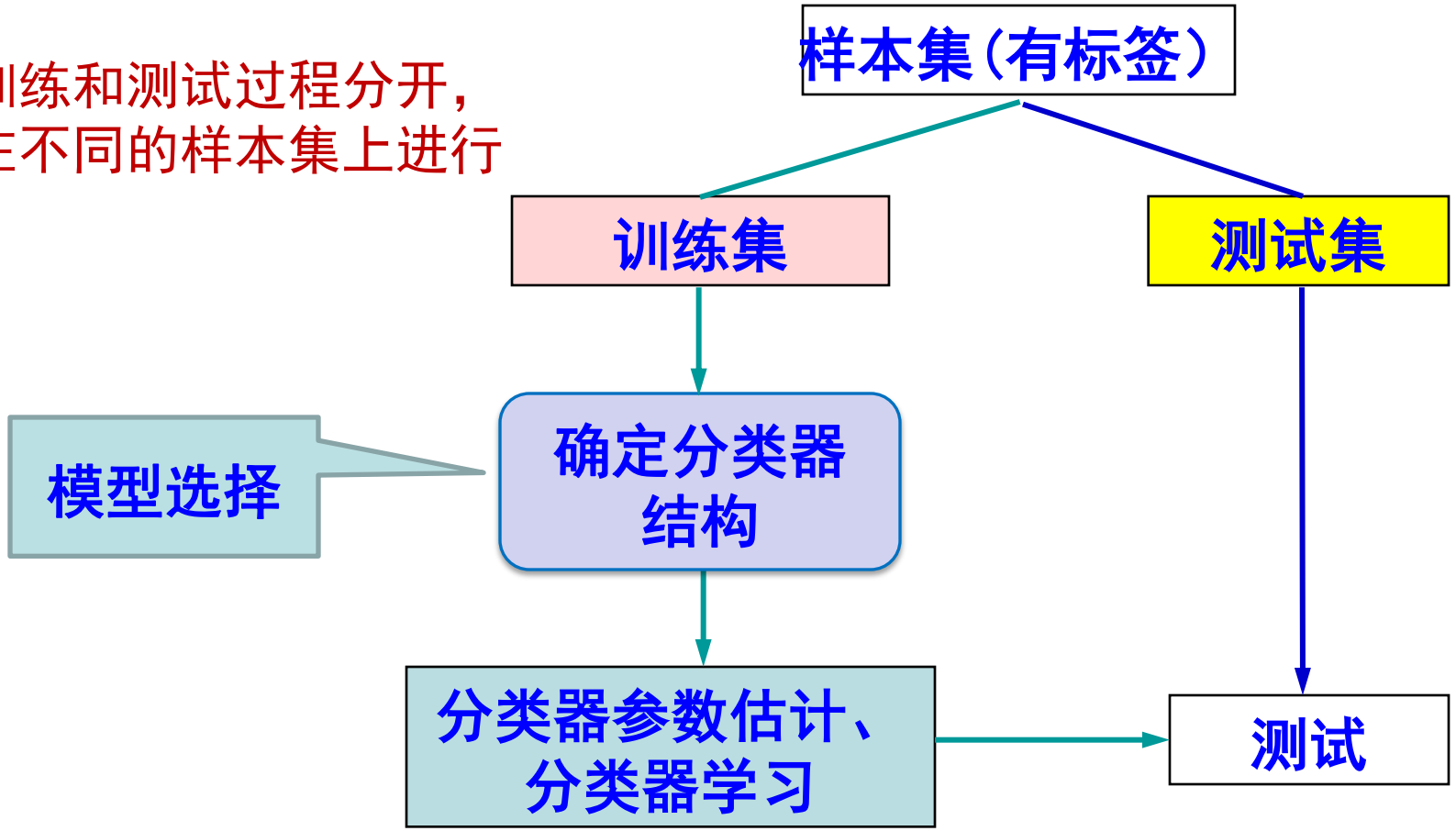
  - ✓ 泛化性能(Generalization Performance)：测试数据上的分类性能
  - ✓ 测试错误率跟训练错误率往往是有差异的
  - ✓ 过拟合/过学习：用复杂分类器能将训练数据分类错误率降到极低
  - ✓ 训练数据越多、越有代表性，则泛化性能越好



7.3 模型评价标准

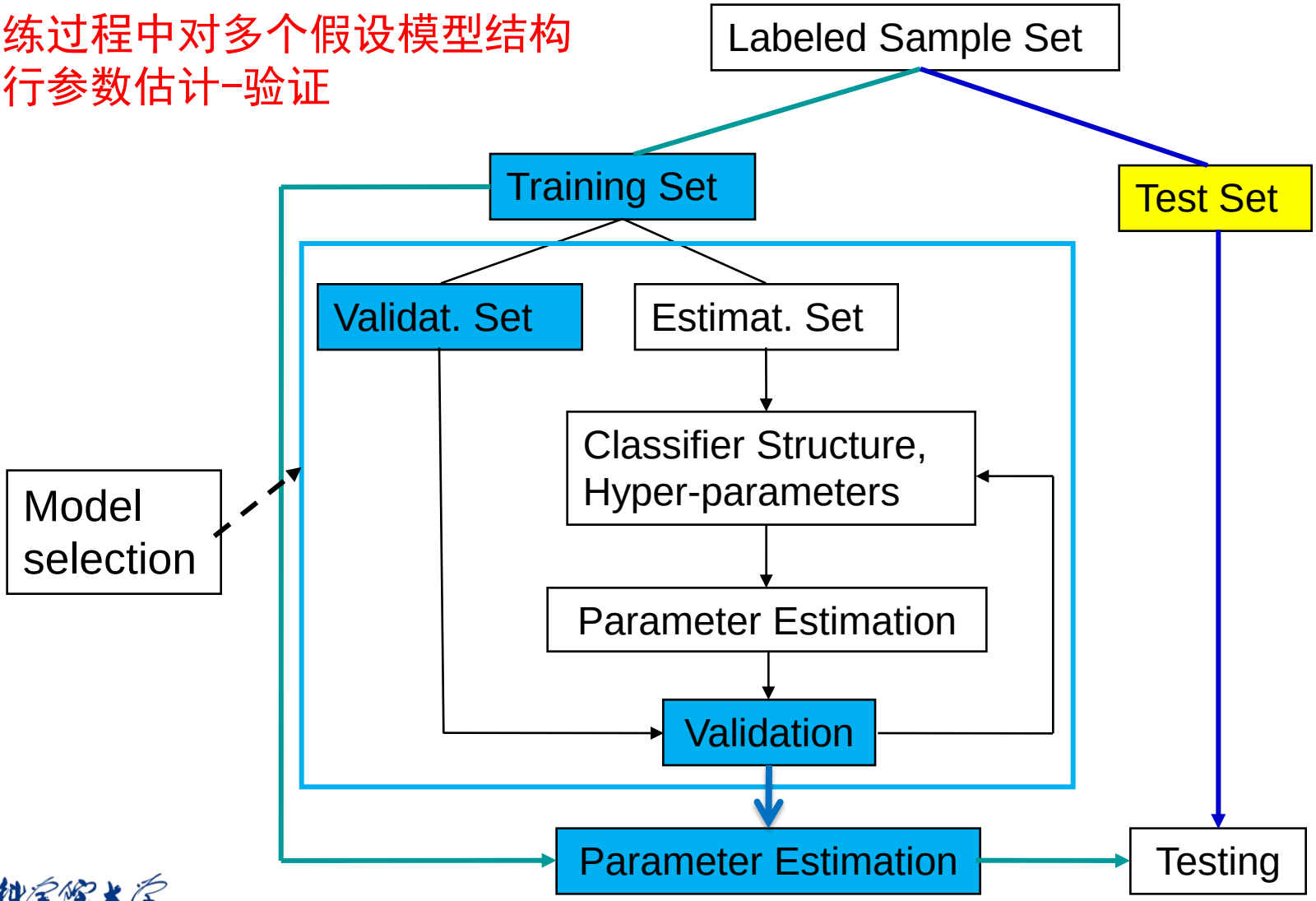
- 分类器训练的评价过程

训练和测试过程分开，
并在不同的样本集上进行



7.3 模型评价标准

训练过程中对多个假设模型结构
进行参数估计-验证



7.3 模型评价标准

- **训练样本的划分**
  - **刀切法（留一法）：**每次从样本集中删除一个或者多个样本，用剩余的样本做为**“刀切样本”**
  - **自助法（ Bootstrap）：**每次**有放回地随机抽取*n*个样本**
  - **保持方法 （Holdout）：一部分用于训练，一部分用于测试**
  - **交叉验证 (cross-validation)**
    - 将数据平分为 k 个子集，用 k-1个子集进行训练，余下的部分用于验证，并计算验证误差。重复这一过程 k 次，得到 k 次结果的平均。
    - 交叉验证是目前最常用的一种**模型选择和评估**方法 **(特别是对模型的参数进行选择)**。

7.3 模型评价标准

- **交叉验证：模型参数选择**
  - 给出一个参数的候选集合
    - 对每一个参数，将训练数据平分为 k 个子集，用 k-1 个子集进行训练，余下一个子集用于验证，并计算验证误差。
    - 根据验证误差，选择统计性能最优的那个参数作为模型的参数，对未来的样本，模型将使用该参数。

7.5 分类器集成概述

- **描述**
  - 将若干单个分类器集成起来，共同完成最终的分类任务，期望取得比单个分类器更好的性能。
- **统计上：**对于一般的学习任务，**要搜索的假设空间通常十分巨大**。但能够用于训练分类器的**样本个数却不足够**用来精确地学习到目标假设。
  - 即使学习到能够满足训练集的假设，但在应用中不一定就具有同样优秀的表现。
  - 因此，输出一个能够很好地满足**单个假设**的学习结果会面临一些风险。**将多个假设集成起来能够降低这种风险（误差抵消）**。

7.5 分类器集成概述

- **计算上**
  - 分类器模型训练通常面临高的计算复杂度。比如最优人工神经网络学习和决策树学习是一个NP难问题。
  - 可采用一些启发式方法来**降低**寻找目标假设的**复杂度**。尽管单个假设可以很简单，也并非最优，**多假设的集成可以使最终结果更加接近实际的目标函数值**。
- **表示上**
  - 学习时所使用的目标假设可能并不在应用时的假设空间之中。**假设空间越开放（不封闭在某一特定类型或特定参数），将其设置为一系列假设的集成，有可能表示出不在假设空间中的目标假设。**

7.5 分类器集成概述

- **集成学习的有效条件**
  - **每个单一的学习器错误率都应当低于0.5**，否则集成的结果反而会提高错误率。
  - **进行集成学习的每个分类器还应当各不相同。**如果每个分类器分类结果相差不大，则集成后的分类器整体和单个分类器做出的决策实际上没有什么差异，性能得不到提高。

7.5 分类器集成概述

- **集成学习的常用技术手段**
  - **通过处理训练数据**（bagging, boosting），比如，对训练样本进行随机分组，对错分样本进行加权。
  - **通过处理特征**，比如，每次只选择一部分特征来训练分类器
  - **通过处理类别标号**，比如，对多类问题，通过类别标号将数据（随机）划分为两个不相交的子集，将多类问题转化为一系列的两类分类问题，最后统计投票
  - **通过改进学习方法**，比如，变更学习参数(如多核学习)或者模型结构(如神经网络结构)、初始参数等

7.5 分类器集成概述

- **分类器集成算法分类**
  - 按基本分类器类型是否相同
    - **异态集成**—基本分类器类型不同
      - **叠加法：**将基本学习器分布在多个层次上。第一层学习器按照某种规则对分类进行预测，然后第一层的预测结果作为第二层的输入，…。
      - **元学习法：采用(学习)一个元分类器**对所有基本分类器的结果进行处理，比如仲裁、合并、投票。
    - **同态集成**—基本分类器类型相同
      - 基本分类器多采用神经网络、二叉树、K—近邻

7.5 分类器集成概述

- 分类器集成算法分类
  - 按训练数据处理方式
    - Bagging
    - Random subspace （随机子空间）
    - Boosting/adaboost
    - 随机森林（讲了决策树之后才能展开）

7.6 Stacked Generalization (层叠泛化)

- **Stacked Generalization**
  - 采用多层结构。第一层的学习器配置不同的学习算法，由训练数据集生成。第一层的输出作为第二层的输入。**第二层学习层称为“元学习器”**。
  - 基本过程
    - 给定训练数据 $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
    - 第一层：学习算法 $L_1, L_2, \ldots, L_T$
    - 第二层：学习算法 L

Stacked Generalization

1 Input data set $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$

2 for $t = 1, \ldots, T$

3 $h_t = L_t(D_t)$ // 学习T个不同的基分类器

4 end

5 $D' = \varnothing$ // 准备生成一个新的数据集合

6 for $i = 1, \ldots, n$,

7 for $t = 1, \ldots, T$, $z_{it} = h_t(\mathbf{x}_i)$, end

8 $D' = D' \cup \{ ((z_{i1}, z_{i2}, \ldots, z_{iT}), y_i) \}$

9 end

10 $h = L(D')$ // 采用收集到的数据，训练h

11 return $H(\mathbf{x}) = h (h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_T(\mathbf{x}))$

7.7 Bagging （装袋）

- **Bagging**
  - 训练一组基分类器，每个基分类器通过一个bootstrap训练样本集来训练。
  - 一个bootstrap训练样本集是通过**有放回地随机**从一个给定的数据中抽样得到。
  - 获得基本分类器之后，bagging通过投票进行统计，被投票最多的类则确定为预测类。

Bagging

1 Input data set $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$

2 Given basic learner L , number of learners T

3 for $t = 1, \ldots, T$

4 $D_t = \text{Bootstrap}(D)$ // 从D中生成一个Bootstrap集

5 $h_t = L(D_t)$ // 学习基分类器

6 end

7 return h_t , $t = 1, \ldots, T$

最终的分类器（即投票最大者）：

$$H(\mathbf{x}) = \arg\max_{y \in Y} \sum_{i=1}^{T} I(y == h_t(\mathbf{x}))$$

取遍所有的标签

真值函数

7.8 Random Subspace （随机子空间）

- **Random Subspace**
  - 随机子空间通常也被称为**属性装袋** (attribute bagging)
  - 随机子空间的基分类器通常由**线性分类器、支持向量机**等组成。

7.8 Random Subspace （随机子空间）

- **随机子空间算法**
  - 令 n 为训练样本的个数，D 为训练数据的特征维数
  - 选择 d 为在每个分类器使用的子特征的维数，$d < D$, 每次可以不同。
  - 令 T 为所有分类器的个数
  - 对于每一个分类器，通过在 D 个特征中选择 d 个子特征来创建一个训练集合，同时学习一个分类器
  - 对新样本，通过多数投票法则来预测其类别。

7.9 Adaboost

- **提升方法**
  - Boosting是一种**提高任意给定学习算法准确度**的方法，是一种常用的统计学习方法，应用广泛且有效。
  - 在分类问题中，**它通过改变训练样本的权重，学习多个分器，并将这些分类器进行组合，提高分类性能。**
  - 提升方法中最具代表性的算法是 **Adaboost**。
  - 基本思路
    - **对于一个复杂任务，将多个专家的判断进行适当的综合所得出的判断，要比其中任何一个专家单独的判断要好。**
    - 这就是**“三个臭皮匠顶个诸葛亮”**。

7.9 Adaboost

- **发展历程**
  - 它的思想起源于 Valiant提出的 PAC (Probably Approximately Correct)学习模型。
  - **1988年**，Kearns和 Valiant提出**强可学习**和**弱可学习**概念。在PAC学习框架中，一个概念（一个类），如果存在一个多项式的学习算法能够学习它，并且正确率很高，则称这个概念是强可学习的；一个概念，如果存在一个多项式的学习算法能够学习它，学习的正确率仅比随机猜测略好，即这个概念是弱可学习的。
  - 但他们留下了一问题：**强可学习与弱可学习是否等价**。如果等价，则可以将弱可学习通过“提升”变成强可学习，这样可以避免直接寻找强可学习算法。

- **发展历程 (续)**
  - **1990年**，Schapire通过一个**构造性方法对“等价性”问题作出了肯定的回答**。证明多个弱分类器可以集成为一个强分类器。这就形成了集成学习的理论基础。
  - **1991年**，Freund提出了boost-by-majority 策略，其亮点思想是：增加**“对难学习部分进行学习的可能性”**，迫使学习者提出新的假设，使其在“难学习部分”少犯错误。但该算法需要知道弱学习算法学习正确率的下限，这一点，在实际应用中难以回答。
  - **1995年**， Freund和Schapire提出adaboost算法，避免这这些难点，还将其推广至分类问题和回归问题
  - 最近，有诸多改进形式， **gentle Adaboost**，real AdaBoost, logit AdaBoost,….

7.9 Adaboost

- **处理分类问题的思想**
  - 给定训练集，**寻找比较粗糙的分类规则（弱分类器）要比寻找精确的分类规则要简单得多。**
  - **提升算法的核心**是从弱学习算法出发，反复学习，得到一系列弱分类器。然后组合这些弱分类器，构成一个强分类器。
  - **基本做法：**改变训练数据的概率（权重）分布，针对不同的训练数据的分布，调用弱学习算法来学习一系列分类器。
  - **需要回答两个问题：**
    - 在每轮训练中，如何改变训练数据的权值或分布？
    - 如何将一系列的弱分类器组合成一个强分类器？

7.9 Adaboost

- **处理分类问题的思想**
  - **关于第一个问题：**
    - Adaboost的做法是：**提高那些被前一轮弱分类器分错的样本的权重，降低已经被正确分类的样本的权重。**错分的样本将在下一轮弱分类器中得到更多关注。于是分类问题被一系列弱分类器**“分而治之”**。
  - **关于第二个问题：**
    - 关于弱分类器的组合， Adaboost的做法是：采用加权(多数)表决的方法。具体地，加大分类误差率较小的弱分类器的权重，使其在表决中起更大的作用。
  - **Adaboost的巧妙之处在于将这些想法融合于一个算法之中！**

7.9 Adaboost

- **Adaboost算法**
  - **给定一个两类分类训练数据集**

$$T=\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$$

其中，每个样本点由实例和标记组成。实例(即样本) $\mathbf{x}_i \in R^d$，标记 $y_i \in \{-1,+1\}$。记 X 为实例空间，Y 为标记集合。

AdaBoost从训练数据中学习一系列弱分类器或者基本分类器，并将这些弱分类器线性地组合成一个强分类器。

7.9 Adaboost

- **Adaboost算法步骤**
  - 输入训练数据集：

$$T=\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$$

  - 输入弱学习算法
  - (1) 初始化训练数据的权值分布

$$D_1=\{w_{11}, w_{12}, \ldots, w_{1n}\},\ w_{1i}=1/n,\ \ i=1,2, \ldots, n$$

  - (2) 对 $m=1, 2, \ldots, M$
  - (2a) 使用具有权值分布 $\boldsymbol{D_m}$ 的训练数据，学习基本分类器

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

Adaboost算法步骤(续)

– (2b) 计算 $G_m(\mathbf{x})$ 在训练数据集上的分类错误率(加权)：

$$e_m = P(G_m(\mathbf{x}_i) \neq y_i) = \sum_{i=1}^{n} w_{mi} I(G_m(\mathbf{x}_i) \neq y_i)$$

真值函数

– (2c) 计算 $G_m(\mathbf{x})$ 的系数：

$$\alpha_m = \frac{1}{2} \ln \frac{1 - e_m}{e_m}$$

α_m 表示 $G_m(\mathbf{x})$ 在最终分类器中的重要性。当 $e_m \leq 0.5$ 时，$\alpha_m \geq 0$。同时，**α_m 将随着 e_m 的减小而增大。**

所以，分类误差率越小的基本分类器在最终分类器中的作用越大。

Adaboost算法步骤(续)

– (2d) 更新训练数据集的权重分布：

$$D_{m+1}=\{w_{m+1,1}, w_{m+1,2}, \cdots, w_{m+1,n}\}$$

具体计算如下：

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} \times \begin{cases} \exp(-\alpha_m), & \text{if } G_m(\mathbf{x}_i) = y_i \\ \exp(\alpha_m), & \text{if } G_m(\mathbf{x}_i) \neq y_i \end{cases}$$

若正确分类，减少权重；否则，增加权重

$$= \frac{w_{mi}}{Z_m} \times \exp(-\alpha_m y_i G_m(\mathbf{x}_i))$$

其中， Z_m 是规范化因子，它使 D_{m+1} 成为一个概率分布：

$$Z_m = \sum_{i=1}^{n} w_{mi} \exp(-\alpha_m y_i G_m(\mathbf{x}_i))$$

Adaboost算法步骤(续)

– (3) 构建基本分类器的线性组合：

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

对于两类分类问题，得到最终的分类器：

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

7.9 Adaboost

- **对Adaboost算法的说明**
  - 步骤(1)：假设训练数据集具有均匀分布的权重，保证第一步能在原始数据上学习到基本分类器。
  - 步骤(2)：AdaBoost 反复学习多个基本分类器，并顺序执行以下操作：
    - (a) 使用当前加权分布 D_m 训练数据，学习基本分类器 $G_m(\mathbf{x})$。

- **对Adaboost算法的说明(续)**
  - 步骤(2)：
    - (b) **计算当前基本分类器在加权训练数据集上的分类误差率：**

$$e_m = P(G_m(\mathbf{x}_i) \neq y_i) = \sum_{G_m(\mathbf{x}_i) \neq y_i} w_{mi}$$

其中，w_{mi} 为第 m 轮中第 i 个实例的权值，且$\sum_i w_{mi}$=1。

这表明，**$G_m(\mathbf{x})$ 在加权的训练数据集上的分类误差率是被其错分样本的权值之和**。由此可以看出“数据权值分布 D_m 与基本分类器 $G_m(\mathbf{x})$ 的分类误差率之间的关系”。

- **对Adaboost算法的说明(续)**
  - **步骤(2)：**
    - (c) 计算基本分类器 $G_m(\mathbf{x})$ 的系数 α_m。
      - α_m 表示 $G_m(\mathbf{x})$ 在最终分类器中的**重要性**。α_m 是 e_m 的**单调递减函数**。
      - 当 $e_m \le 0.5$ 时，$\alpha_m \ge 0$。同时，α_m 将随着 e_m 的减小而增大。**所以分类误差率越小的基本分类器在最终分类器中的作用越大。**
      - 当 $e_m > 0.5$ 时，$\alpha_m < 0$ (重要性小于0)。此时，该基分类器将起到**负面作用**。因此必须保证每个基本分类器要优于随机猜测。

- **对Adaboost算法的说明(续)**
  - **步骤(2)：**
    - (d) 更新训练数据的权值分布，为下一轮作准备：

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} \times \begin{cases} \exp(-\alpha_m), & \text{if } G_m(\mathbf{x}_i) = y_i \\ \exp(\alpha_m), & \text{if } G_m(\mathbf{x}_i) \neq y_i \end{cases}$$

可见： 被基本分类 $G_m(\mathbf{x})$ 误分的样本的权重得以扩大。相对于正确分类的样本，扩大 $\exp(2\alpha_m) = e_m/(1\text{-}e_m)$倍。因此，误为样本在下一轮学习中起的作用会更大。

不断地改变训练样本的权值，使其在基本分类器中起不同的作用，这正是AdaBoost的一个特点。

- **对Adaboost算法的说明(续)**
  - **步骤(3)：线性组合 $f(\mathbf{x})$ 实现了对M个基本分类器的加权表决。系数** α_m 表示基本分类器 $G_m(\mathbf{x})$ 的重要性。注意，所有 α_m 之和并不为 1。
    - $f(\mathbf{x})$ 的符号决定了实例(即样本) $\mathbf{x}$ 的类别， $f(\mathbf{x})$ 的绝对值表示分类的确信度。

利用基本分类器的线性组合构建最终的分类器，也是**AdaBoost**的另一个特点。

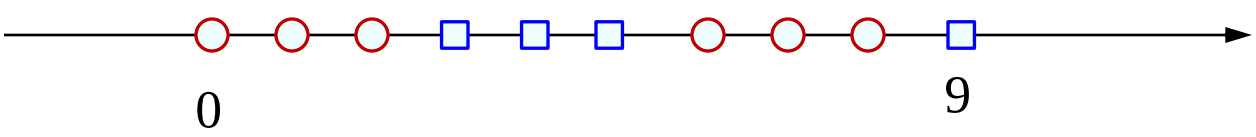
7.9 Adaboost

$sign(x-v)$

- **例子：** 给定如表所示训练数据。假设弱分类器由“如果 $x<v$，则 x 属于第一类；如果 $x>v$，则 x 属于第二类”产生，其阈值使该分类器在训练数据集上分类误差率最低。 试用Adaboost算法学习一个强分类器。

序号	1	2	3	4	5	6	7	8	9	10
x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1

上表中：共10个一维空间的样本，x表示样本，y表示标签。



解： 初始化数据权值分布：

$$D_1 = (w_{11}, w_{12}, \cdots, w_{110}), \qquad w_{1i} = 0.1, \quad i = 1, 2, \cdots, 10$$

对m=1:

(a) 在权值分布为 D_1 的训练数据上，阈值 v 取2.5时分类误差率最低 (取不同的阈值，然后计算加权错误率，选择最小者)，故基本分类器 $G_1(x)$ 为：

$$G_1(x) = \begin{cases} 1, & x < 2.5 \\ -1, & x > 2.5 \end{cases}$$

(b) $G_1(x)$在训练数据集上的误差率:

$$e_1 = P(G_1(x_i) \neq y_i) = 0.3$$

解(续)：

对m=1:

(c) 计算 $G_1(x)$ 的系数：$\alpha_1 = \frac{1}{2}\log\frac{1-e_1}{e_1} = 0.4236$

(d) 更新训练数据的权值分布：

$$D_2 = (w_{21}, \cdots, w_{2i}, \cdots, w_{210})$$

$$w_{2i} = \frac{w_{1i}}{Z_1}\exp(-\alpha_1 y_i G_1(x_i)), \quad i = 1, 2, \cdots, 10$$

$$D_2 = (0.0715, 0.0715, 0.0715, 0.0715, 0.0715, 0.0715, 0.1666, 0.1666, 0.1666, 0.0715)$$

(e) 获得 $f_1(x) = 0.4236\, G_1(x)$。

此时，分类器 sign[$f_1(x)$] 在训练集上有3个误分样本。

解(续)：

对m=2:

(a) 在权值分布为 D_2 的训练数据上，阈值 v 取8.5时分类误差率最低(取不同的阈值，然后计算加权错误率，选择最小者)，故基本分类器 $G_2(x)$ 为：

$$G_2(x)=\begin{cases}1, & x<8.5\\ -1, & x>8.5\end{cases}$$

(b) $G_2(x)$ 在训练数据集上的误差率:

$$e_2=P(G_2(x_i)\neq y_i)=0.2143$$

解(续)：

对m=2:

(c) 计算 $G_2(x)$ 的系数：$\alpha_2 = \frac{1}{2}\log\frac{1-e_2}{e_2} = 0.6496$

(d)更新训练数据的权值分布：

$$D_3 = (w_{31}, \cdots, w_{3i}, \cdots, w_{310})$$

$$w_{3i} = \frac{w_{2i}}{Z_2}\exp(-\alpha_2 y_i G_2(x_i)), \quad i = 1, 2, \cdots, 10$$

$$D_2 = (0.0455, 0.0455, 0.0455, 0.1667, 0.1667, 0.1667, 0.1060, 0.1060, 0.1060, 0.0455)$$

(e) 获得 $f_2(x)$ = 0.4236 $G_1(x)$ +0.6496 $G_2(x)$ 。

此时，分类器 sign[$f_2(x)$] 在训练集上有3个误分样本。

解(续)：

对m=3:

(a) 在权值分布为 D_3 的训练数据上，阈值 v 取5.5时分类误差率最低(取不同的阈值，然后计算加权错误率，选择最小者)，故基本分类器 $G_3(x)$ 为：

$$G_3(x)=\begin{cases}1, & x>5.5\\ -1, & x<5.5\end{cases}$$

(b) $G_3(x)$在训练数据集上的误差率:

$$e_3=P(G_3(x_i)\neq y_i)=0.1820$$

解(续)：

对m=3:

(c) 计算 $G_3(x)$的系数： $\alpha_3 = \frac{1}{2}\log\frac{1-e_3}{e_3} = 0.7514$

(d) 更新训练数据的权值分布：

$$D_4 = (w_{41}, \cdots, w_{4i}, \cdots, w_{410})$$

$$w_{4i} = \frac{w_{3i}}{Z_3}\exp(-\alpha_3 y_i G_3(x_i)), \quad i = 1, 2, \cdots, 10$$

$$D_2 = (0.125, 0.125, 0.125, 0.102, 0.102, 0.102, 0.065, 0.065, 0.065, 0.125)$$

(e) 获得 $f_3(x)$ = 0.4236 $G_1(x)$ + 0.6496 $G_2(x)$ + 0.7514 $G_3(x)$ 。

此时，分类器 sign[$f_3(x)$] 在训练数据集误分样本个数为0。

获得最终分类器： sign[$f_3(x)$]

7.10 对Adaboost的理论解释

- **关于Adaboost算法的理论解释，有如下三个定理：**
  - 定理1：最终分类器关于训练误差的界
  - 定理2：最终分类器关于训练误差的界 (更具体)
  - 定理3：Adaboost与前向分步算法(加法模型)之间的关系

7.10 对Adaboost的理论解释

- **定理1：AdaBoost算法最终分类器的训练误差界为：**

$$\frac{1}{n}\sum_{i=1}^{n} I(G(\mathbf{x}_i) \neq y_i) \leq \frac{1}{n}\sum_{i=1}^{n} \exp\left(-y_i f(\mathbf{x}_i)\right) = \prod_{m=1}^{M} Z_m$$

训练误差：
错误个数除以总数

上界：
基本分类器误差之乘积

Adaboost最基本的性质是它能在学习过程中不断地减少训练误差。

- **定理1证明：第一部分**

  - 当$G_m(\mathbf{x}) \neq y_i$，$y_i f(\mathbf{x}_i) < 0$，此时$\exp(-y_i f(\mathbf{x}_i)) \geq 1$。
    综合起来，自然有：

$$\frac{1}{n}\sum_{i=1}^{n} I(G(\mathbf{x}_i) \neq y_i) \leq \frac{1}{n}\sum_{i=1}^{n} \exp\left(-y_i f(\mathbf{x}_i)\right)$$

- **定理1证明：第二部分**

注意到样本权重的更新方式:

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} \times \exp(-\alpha_m y_i G_m(\mathbf{x}_i))$$

所以，有：$w_{m+1,i} Z_m = w_{mi} \exp(-\alpha_m y_i G_m(\mathbf{x}_i)),\ i = 1,..,n$

$$\frac{1}{n}\sum_{i=1}^{n}\exp\left(-y_i f(\mathbf{x}_i)\right)$$

$$=\frac{1}{n}\sum_{i=1}^{n}\exp\left(-\sum_{m=1}^{M}\alpha_m y_i G_m(\mathbf{x}_i)\right)$$

$$=\sum_{i=1}^{n} w_{1i}\prod_{m=1}^{M}\exp\left(-\alpha_m y_i G_m(\mathbf{x}_i)\right) \qquad \because w_{1i}=\frac{1}{n}$$

$$=\sum_{i=1}^{n} w_{1i}\exp\left(-\alpha_1 y_i G_1(\mathbf{x}_i)\right)\prod_{m=2}^{M}\exp\left(-\alpha_m y_i G_m(\mathbf{x}_i)\right)$$

$$=\sum_{i=1}^{n} Z_1 w_{2i}\prod_{m=2}^{M}\exp\left(-\alpha_m y_i G_m(\mathbf{x}_i)\right)$$

$$=Z_1\sum_{i=1}^{n} w_{2i}\exp\left(-\alpha_1 y_i G_1(\mathbf{x}_i)\right)\prod_{m=3}^{M}\exp\left(-\alpha_m y_i G_m(\mathbf{x}_i)\right)$$

$$=Z_1\sum_{i=1}^{n} Z_2 w_{3i}\prod_{m=3}^{M}\exp\left(-\alpha_m y_i G_m(\mathbf{x}_i)\right)$$

$$=Z_1 Z_2\sum_{i=1}^{n} w_{3i}\prod_{m=3}^{M}\exp\left(-\alpha_m y_i G_m(\mathbf{x}_i)\right)$$

$$=Z_1 Z_2\cdots Z_{M-1}\sum_{i=1}^{n} w_{Mi}\exp\left(-\alpha_M y_i G_M(\mathbf{x}_i)\right)$$

$$=\prod_{m=1}^{M} Z_m$$

(后续见右上)

证毕

7.10 对Adaboost的理论解释

- **定理2：AdaBoost的训练误差界可进一步写为:**

$$\prod_{m=1}^{M} Z_m = \prod_{m=1}^{M}\left(2\sqrt{e_m(1-e_m)}\right) = \prod_{m=1}^{M}\left(\sqrt{1-4\gamma_m^2}\right) \le \exp\left(-2\sum_{m=1}^{M}\gamma_m^2\right)$$

其中，$\gamma_m=0.5-e_m$。

- **定理2证明:**

$$Z_m = \sum_{i=1}^{n} w_{mi} \exp(-\alpha_m y_i G_m(\mathbf{x}_i))$$

$$= \sum_{y_i = G_m(\mathbf{x}_i)} w_{mi} \, \mathrm{e}^{-\alpha_m} + \sum_{y_i \neq G_m(\mathbf{x}_i)} w_{mi} \, \mathrm{e}^{\alpha_m}$$

$$= (1 - e_m)\mathrm{e}^{-\alpha_m} + e_m e^{\alpha_m}$$

$$= 2\sqrt{e_m(1-e_m)} \quad = \quad \sqrt{1-4\gamma_m^2} \qquad \because \alpha_m = \frac{1}{2}\ln\frac{1-e_m}{e_m}$$

对于不等式： $\prod_{m=1}^{M}\left(\sqrt{1-4\gamma_m^2}\right) \leq \exp\left(-2\sum_{m=1}^{M}\gamma_m^2\right)$

可先由 e^x 和 $\sqrt{1-x}$ 在 x=0 的泰勒展开式推出不等式：

$$\sqrt{1-4\gamma_m^2} \leq \exp\left(-2\gamma_m^2\right)$$

7.10 对Adaboost的理论解释

- **推论：如果存在一个$\gamma > 0$，对所有m有$\gamma_m \geq \gamma$，则**

$$\frac{1}{n}\sum_{i=1}^{n} I(G(\mathbf{x}_i) \neq y_i) \leq \exp\left(-2M\gamma^2\right)$$

这表明， Adaboost的训练误差**随着基本分类器的增多而以指数速率下降**。(γ 总是可以大于0的）

同时，要注意这是算法的一个性质。而实际应用中 Adaboost 并不需要知道这个下界 γ。这正是Freund和Schapire设计 Adaboost 时所考虑的。

因此，与早期的方法不同， Adaboost具有自适应性，即它能适应弱分类器各自的训练误差。**这正是其名称的由来。**

7.10 对Adaboost的理论解释

- 从加法模型的角度对Adaboost的解释
  - Adbboost算法是一种学习模型为加法模型、损失函数为指数函数、学习方法为前向分步算法的两类分类学习方法。
  - 考虑如下加法模型：

$$f(\mathbf{x}) = \sum_{m=1}^{M} \beta_m b(\mathbf{x}; \gamma_m)$$

其中，$b(\mathbf{x}; \gamma_m)$ 为基函数，γ_m 为基函数的参数，β_m 为基函数的系数。

7.10 对Adaboost的理论解释

- **对加法模型的学习**
  - **在给定训练数据及损失函数** $L(y, f(\mathbf{x}))$ **，** 学习加法模型 $f(\mathbf{x})$ 可以通过最小化经验风险（即所有样本引起的损失之和）来实现：

$$\min_{\beta_m,\gamma_m} \sum_{i=1}^{n} L\left(y_i, \sum_{m=1}^{M} \beta_m b(\mathbf{x}_i;\gamma_m)\right)$$

这是一个复杂的优化问题，期望通过最优化技术一次性求出来通常很困难。取决于基函数的形式。

7.10 对Adaboost的理论解释

- 对加法模型的学习
  - 前向分步算法 (forward stagewise algorithm)
  - 求解该问题的基本思路：
    - 因为学习模型是加法模型，每一步只学习一个基函数及其系数，逐步逼近优化目标函数（见前一页），那么就可以简化优化的复杂度。具体地，每步只优化如下损失函数：

$$\min_{\beta,\gamma} \sum_{i=1}^{n} L\left(y_i, \beta b\left(\mathbf{x}_i;\gamma\right)\right)$$

- **前向分步算法步骤:**
  - 输入训练数据集： $T=\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$
  - 给定损失函数 $L(y, f(\mathrm{x}))$，基函数 $b(\mathrm{x}; \gamma)$;
  - (1) 初始化 $f_0(\mathrm{x}) = 0$;
  - (2) 对 $m = 1, 2, \ldots, M$;
    - (2a) **极小化当前损失函数**，得到参数β_m，γ_m:

$$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{n} L\left(y_i, \boxed{f_{m-1}(\mathbf{x}_i) + \beta b(\mathbf{x}_i; \gamma)}\right)$$

    - (2b) **更新:** $f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \beta_m b(\mathbf{x}; \gamma_m)$
    - (2c) **获得最后模型:** $f(\mathbf{x}) = f_M(\mathbf{x}) = \sum_{m=1}^{M} \beta_m b(\mathbf{x}; \gamma_m)$

7.10 对Adaboost的理论解释

- **定理3：** **AdaBoost是前向分步加法算法的特例，且学习模型是由基本分类器组成的加法模型，损失函数为指数函数。**

我们只需要证明当前向分步算法的基函数为指数函数时，该算法就变为Adaboost，此定理即可获证。

具体的指数函数为： $L(y, f(\mathbf{x})) = \exp(-y f(\mathbf{x}))$。

- **定理3的证明**
  - 假设经过 m-1轮迭代，前向分步算法已经得到 $f_{m-1}(\mathbf{x})$：

$$f_{m-1}(\mathbf{x}) = \alpha_1 G_1(\mathbf{x}) + \alpha_2 G_2(\mathbf{x}) + \ldots + \alpha_{m-1} G_{m-1}(\mathbf{x})$$

  - 进一步，假定在第m轮前向分步算法通过求解如下化优模型得到α_m，$G_m(\mathbf{x})$：

$$\left(\alpha_m, G_m(\mathbf{x})\right) = \arg\min_{\alpha, G(\mathbf{x})} \sum_{i=1}^{n} \exp\left(-y_i f_m(\mathbf{x}_i)\right)$$

其中，$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \alpha_m G_m(\mathbf{x})$。

令 $\quad \bar{w}_{mi} = \exp\left(-y_i f_{m-1}(\mathbf{x}_i)\right)$ ← 与 α 和 $G(\mathbf{x})$无关

  - 因此，上述优化问题可表示为：

$$\left(\alpha_m, G_m(\mathbf{x})\right) = \arg\min_{\alpha, G(\mathbf{x})} \sum_{i=1}^{n} \bar{w}_{mi} \exp\left(-y_i \alpha G(\mathbf{x}_i)\right)$$

- **定理3的证明(续)**
  - 进一步只需证明：

求解如下问题获得的最优解 α^*_m 和 $G^*_m(\mathbf{x})$ 就是AdaBoost算法所得到的 α_m 和 $G_m(\mathbf{x})$：

$$\left(\alpha_m, G_m(\mathbf{x})\right) = \arg\min_{\alpha, G(\mathbf{x})} \sum_{i=1}^{n} \bar{w}_{mi} \exp\left(-y_i \alpha G(\mathbf{x}_i)\right)$$

其中，$\bar{w}_{mi} = \exp\left(-y_i f_{m-1}(\mathbf{x}_i)\right)$

因此，当前任务是如何求解上述优化问题。

- **定理3的证明(续)**
  - 首先求$G^*_m(\mathbf{x})$。对于任意的 $\alpha > 0$，有如下等价的转换：

$$G_m^*(\mathbf{x}) = \arg\min_{G(\mathbf{x})} \sum_{i=1}^{n} \bar{w}_{mi} \exp\left(-y_i \alpha G(\mathbf{x}_i)\right)$$

$$\Updownarrow$$

$$G_m^*(\mathbf{x}) = \arg\min_{G(\mathbf{x})} \sum_{i=1}^{n} \bar{w}_{mi} I\left(y_i \neq G(\mathbf{x}_i)\right)$$

这正是Adaboost第m步时所构造的基分器。因为，当$y_i=G(\mathbf{x}_i)$时，$y_iG(\mathbf{x}_i)=1$，此时$\exp(-\alpha)$为常数，与优化变量无关。

- **定理3的证明(续)**
  - 其次求 α^*_m。考查前述优化问题的目标函数：

$$\sum_{i=1}^{n} \bar{w}_{mi} \exp\left(-y_i \alpha G(\mathbf{x}_i)\right)$$

$$= \sum_{y_i = G(\mathbf{x}_i)} \bar{w}_{mi} \, \mathrm{e}^{-\alpha} + \sum_{y_i \neq G(\mathbf{x}_i)} \bar{w}_{mi} \, \mathrm{e}^{\alpha}$$

$$= \mathrm{e}^{-\alpha} \sum_{i=1}^{n} \bar{w}_{mi} - \mathrm{e}^{-\alpha} \sum_{i=1}^{n} \bar{w}_{mi} I(y_i \neq G(\mathbf{x}_i)) + \mathrm{e}^{\alpha} \sum_{i=1}^{n} \bar{w}_{mi} I(y_i \neq G(\mathbf{x}_i))$$

$$= (\mathrm{e}^{\alpha} - \mathrm{e}^{-\alpha}) \sum_{i=1}^{n} \bar{w}_{mi} I(y_i \neq G(\mathbf{x}_i)) + \mathrm{e}^{-\alpha} \sum_{i=1}^{n} \bar{w}_{mi}$$

真值函数

- **定理3的证明(续)**
  - 将已求得的$G^*_m(\mathbf{x})$代入上述目标函数之中，并对α求导数，并令其等于0，于是有：

$$\alpha^*_m = \frac{1}{2}\ln\frac{1-e_m}{e_m}$$

其中，e_m是分类错误率：

$$e_m = \frac{\sum_{i=1}^{n}\bar{w}_{mi}I(y_i \neq G(\mathbf{x}_i))}{\sum_{i=1}^{n}\bar{w}_{mi}} = \sum_{i=1}^{n}w_{mi}I(y_i \neq G(\mathbf{x}_i))$$

这里的 α^*_m 与Adaboost 算法的第 2(c) 步的 α_m 完全一致。

- **定理3的证明(续)**
  - 关于最后一轮的更新。由如下两式

$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \alpha_m G_m(\mathbf{x}), \quad \bar{w}_{mi} = \exp\left(-y_i f_{m-1}(\mathbf{x}_i)\right)$$

可得：

$$\begin{aligned}\bar{w}_{mi} &= \exp\left(-y_i f_{m-1}(\mathbf{x}_i)\right) \\ &= \exp\left(-y_i \left(f_m(\mathbf{x}) - \alpha_m G_m(\mathbf{x})\right)\right) \\ &= \exp\left(-y_i \left(f_m(\mathbf{x})\right)\right) \cdot \exp\left(y_i \alpha_m G_m(\mathbf{x})\right) \\ &= \bar{w}_{m+1i} \exp\left(y_i \alpha_m G_m(\mathbf{x})\right)\end{aligned}$$

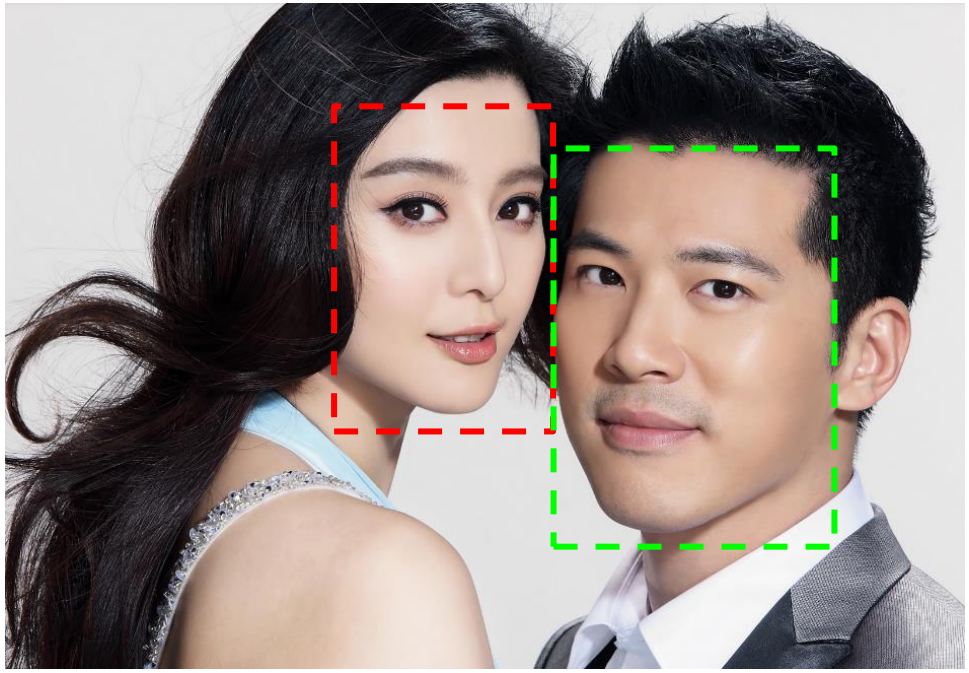
$$\Rightarrow \quad \bar{w}_{m+1i} = \bar{w}_{mi} \exp\left(-y_i \alpha_m G_m(\mathbf{x})\right)$$

这与 Adaboost 算法的第 2(d) 步样本权值更新形式完全相同，只差一个归一化因子，因而等价。

证毕.

7.11 基于Adaboost的人脸检测

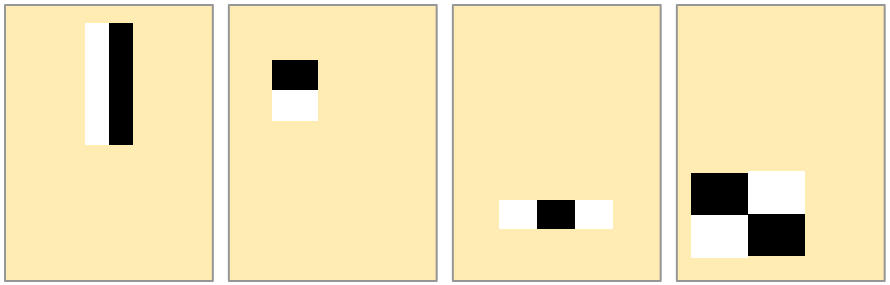
- 人脸检测



Paul Viola, Michael Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features, CVPR, Vol.1, pp. 551-558, 2001

7.11 基于Adaboost的人脸检测

- **Haar特征—矩形特征**
  - 一个haar特征由一组方形滤波器组成
  - 滤波器响应值为对应区域内像素值的和
  - 一个haar特征的响应值为白色滤波器响应值减去灰色滤波器响应值（**即白色区域的像素灰度之和减去黑色区域的像素灰度之和**）
  - 形状与Haar小波类似



Haar(*x*, *y*, *w*, *h*, type)

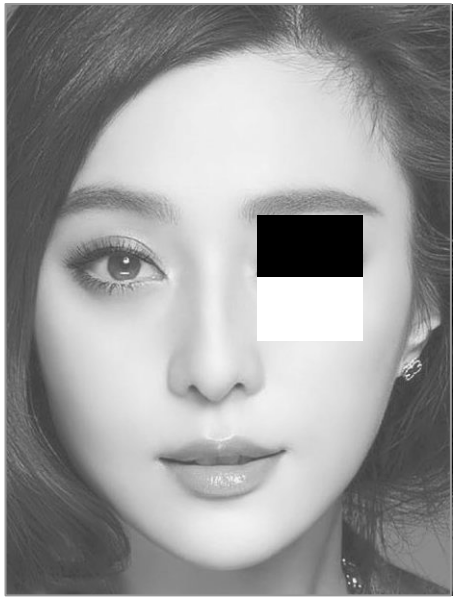
（矩形模板的类型）

（矩形模板的位置和大小）

7.11 基于Adaboost的人脸检测

- **Haar特征—矩形特征**

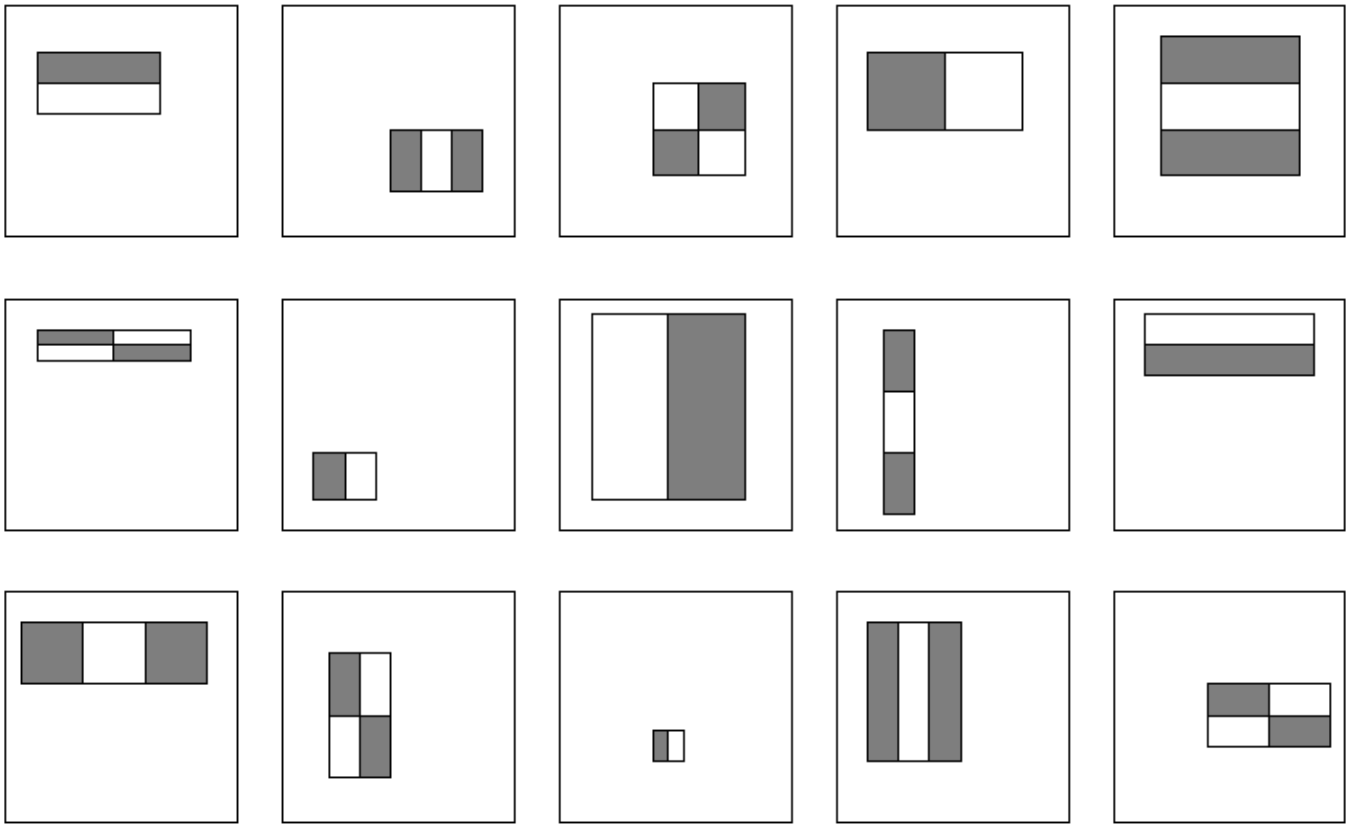




可以很好地描述人脸的眼部区域的灰度分布情况

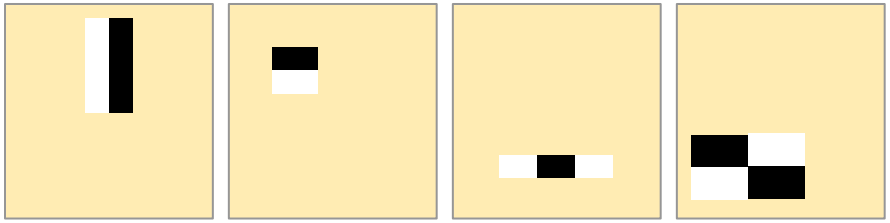
7.11 基于Adaboost的人脸检测

- **Haar特征—能构造的矩形模板是很多的！**



7.11 基于Adaboost的人脸检测

- **Haar特征—能构造的矩形模板是很多的！**



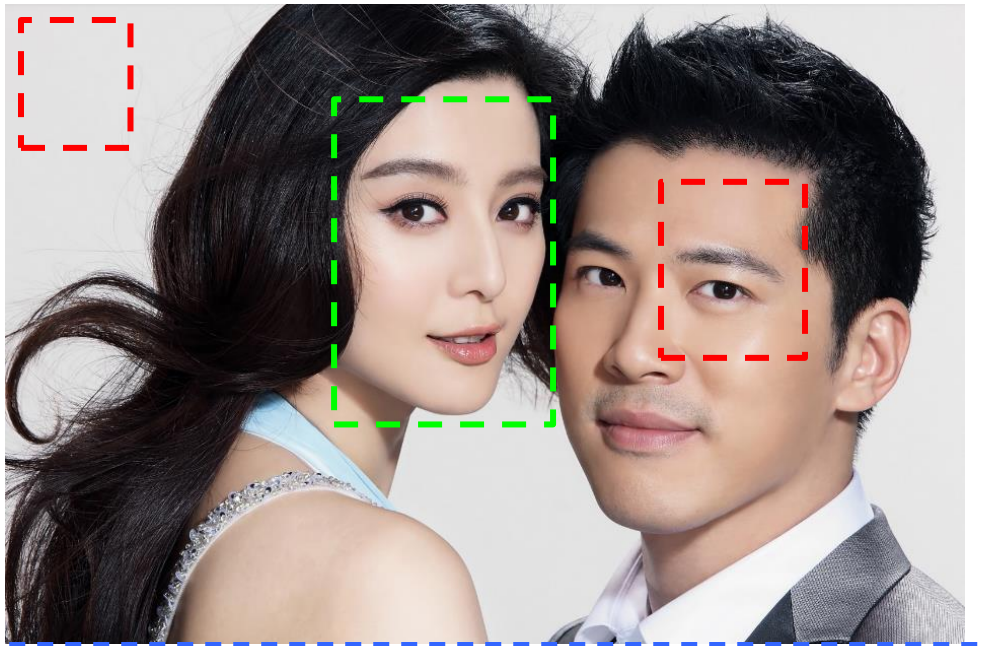
对于一个24x24的图像窗口，从一些基本的矩形模板出发，根据其比例可缩放，位置可移动，角度可旋转，可以构造出多达**几十万甚至上百万个**不同的矩形模板。

这就意味着，从一个人脸图像，我们可以获得一个几十万维甚至上百万维的特征向量。

OK！我们当然可以根据训练集样本（人脸和非人脸），采用SVM来完成此任务！

7.11 基于Adaboost的人脸检测

- **Haar特征—计算量太大！**

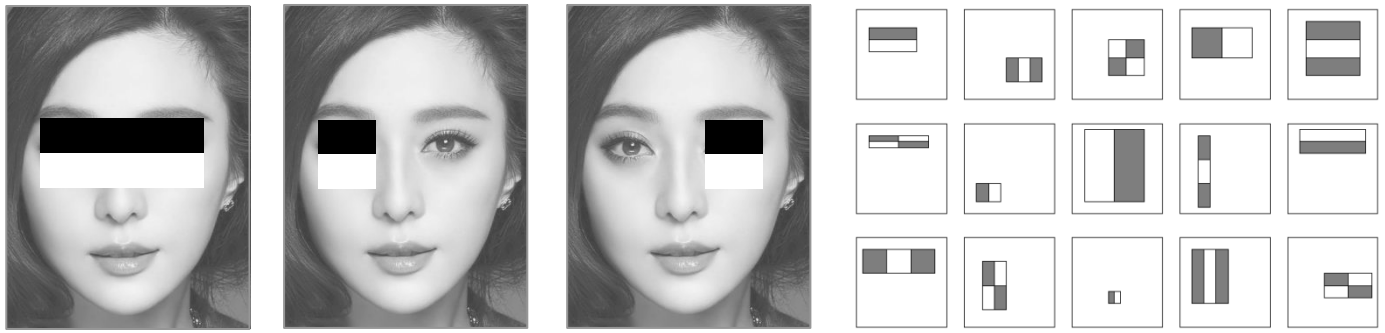


由于不知道人脸的具体尺寸，不同比例的窗口都得扫描一遍！

（做法：每次取固定大小的窗口按行按列扫描图像，每获得一个窗口，首先将其缩放到训练图像的大小，比如：24*24，然后获得上几十万维Haar特征，最后采用SVM进行分类）

7.11 基于Adaboost的人脸检测

- **Haar特征—让我们来换一个角度！**



让我们来评价
每一个Haar特征对人脸的描述能力，
也就是评价其分类判别能力！

对每一个Haar特征（即一个标量特征）
引入一个弱分器！

7.11 基于Adaboost的人脸检测

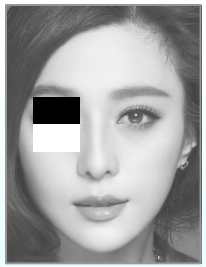
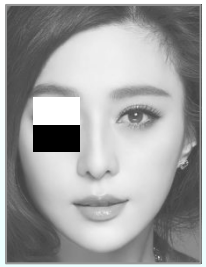
- **从每一个Haar特征构建一个弱分类器！**
  - 首先，对于一个给定子窗口和矩形模板，计算其Haar特征值Haar(x, y, w, h, type)，它是一个标量。
  - 然后，引入一个分类器：

$$G(x,y,w,h,type)=\begin{cases}1, & \text{if } \mathrm{Haar}(x,y,w,h,type)<\theta \\ 0, & \text{otherwise}\end{cases}$$

含义：通过比较Haar特征值是否超过某个阈值来判断是否为人脸

注意，阈值θ 是可变的, 因此它是一个待确定的参数。

- **将弱分类器能否设计得更好，更灵活！**

计算左边矩形模板的一个Haar特征，也同时得到了右边一个模板的Haar特征！

$$G(x, y, w, h, type) = \begin{cases} 1, & \text{if} \quad p \cdot \text{Haar}(x, y, w, h, type) < p \cdot \theta \\ 0, & \text{otherwise} \end{cases}$$

进一步，用一个符号“f”来表示 Haar(x, y, w, h, type)，则该弱分类器可写成明确的参数形式：

$$G(\mathbf{x}, f, p, \theta) = \begin{cases} 1, & \text{if} \quad pf < p\theta \\ 0, & \text{otherwise} \end{cases} \quad (p = \pm 1)$$

矩形窗口所覆盖的灰度数据

7.11 基于Adaboost的人脸检测

- **从训练弱分类器开始**

现在，我们手头有了几十万个弱分类器，
每个弱分类器有两个参数！

回忆一下我们如何构造强分类器！

我们在每次迭代确定一个弱分类器时，需要找一个
错误率最小的弱分类器。
也就是说，此时要通过学习的方式来确定一个弱分类器！

让我们开始吧！
在刚开始的时候，所有的样本的权重都是相同的！

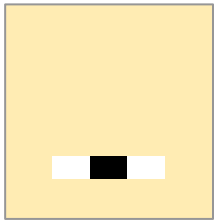
7.11 基于Adaboost的人脸检测

- **弱分类器训练**
  - 训练一个弱分类器（特征 f ）就是在当前权重分布的情况下，确定 f 的最优阈值以及不等号的方向，使得这个弱分类器（特征 f ）对所有训练样本的分类错误率最低。
  - **对于每个特征 f，**计算所有训练样本的特征值，并将其排序(升) 。现在逐元素扫描此排序，计算以下四个量：
    - T^+: 全部人脸样本的权重之和
    - T^-: 全部非人脸样本的权重之和
    - S^+: 在此元素之前的人脸样本的权重之和
    - S^-: 在此元素之前的非人脸样本的权重之和

7.11 基于Adaboost的人脸检测

- **弱分类器训练**

所有样本的特征值：　j



因此，当选取当前元素(第j个)的特征值作为阈值时，所得到的弱分类器在当前元素处将样本分为两类。也就是说，这个阈值对应的弱分类器将该元素之前的所有样本分为**人脸**（或非人脸），而将该元素之后(含)的所有样本分为**非人脸**（或人脸）

根据不等号方向决定：人脸（或非人脸）——非人脸（或人脸）

当然，这一点是根据错误率来决定的

- **弱分类器误差计算**

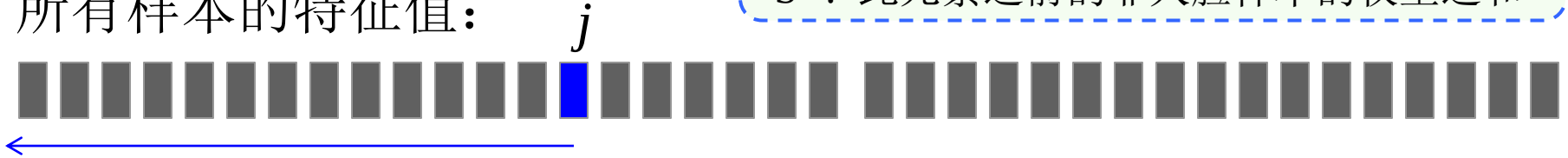
T^+: 全部人脸样本的权重之和

T^-: 全部非人脸样本的权重之和

S^+: 此元素之前的人脸样本的权重之和

S^-: 此元素之前的非人脸样本的权重之和

所有样本的特征值：



如果该元素之前为人脸，该元素之后为非人脸，误差为：

$$S^- + (T^+ - S^+)$$

被错分为人脸的“非人脸”样本错误（权重）总和

被错分为非人脸的“人脸”样本误差（权重）总和

$$p = +1: \qquad G(\mathbf{x}, f, p, \theta) = \begin{cases} 1, & \text{if } f < \theta \\ 0, & \text{otherwise} \end{cases}$$

• 弱分类器误差计算

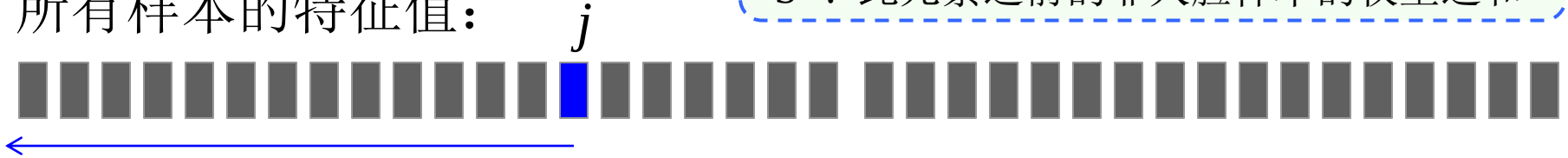
T^+: 全部人脸样本的权重之和

T^-: 全部非人脸样本的权重之和

S^+: 此元素之前的人脸样本的权重之和

S^-: 此元素之前的非人脸样本的权重之和

所有样本的特征值：



如果该元素之前为非人脸，该元素之后为人脸，误差为：

$$S^+ + (T^- - S^-)$$

被错分为非人脸的“人脸”样本错误（权重）总和

被错分为人脸的“非人脸”样本误差（权重）总和

$$p = -1: \quad G(\mathbf{x}, f, p, \theta) = \begin{cases} 1, & \text{if } f > \theta \\ 0, & \text{otherwise} \end{cases}$$

- **弱分类器误差计算**

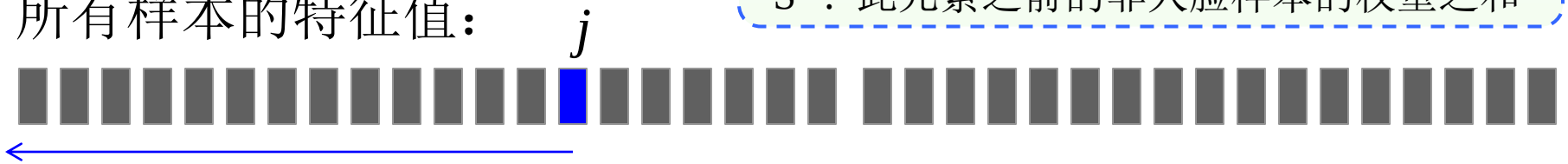
T^+: 全部人脸样本的权重之和

T^-: 全部非人脸样本的权重之和

S^+: 此元素之前的人脸样本的权重之和

S^-: 此元素之前的非人脸样本的权重之和

所有样本的特征值：



现在，取两个误差中的最小者(注意此处添加了一个下标 j)：

$$e_j = \min\left(S^- + (T^+ - S^+),\ S^+ + (T^- - S^-)\right)$$

最后，遍历所有j, 即取所有误差中的最小者，获得一个弱分类器。通过该方法，弱分类器的两个参数都确定了！

7.11 基于Adaboost的人脸检测

- **强分类器构造**

现在，我们知道了如何在当前样本的权值分布下学习一个最优的弱分类器。

那么，如何构造强分类器？

按在9.7节讲的步骤来即可！得到如下组合函数：

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

最后，可以得到一个强分类器（按Viola和 Jones建议）

$$H(x) = \begin{cases} 1, & f(\mathbf{x}) \geq \frac{1}{2}\sum_{i=1}^{M} \delta_i \\ 0, & \text{otherwise} \end{cases} \qquad \left(\delta_t = \log 1/\beta_t, \quad \beta_t = e_t/(1-e_t)\right)$$

7.11 基于Adaboost的人脸检测

- **所得强分类器**

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

$$H(x) = \begin{cases} 1, & f(\mathbf{x}) \geq \dfrac{1}{2}\sum_{m=1}^{M} \delta_m \\ 0, & \text{otherwise} \end{cases}$$

$$\left(\delta_m = \log 1/\beta_m, \quad \beta_m = e_m/(1-e_m)\right)$$

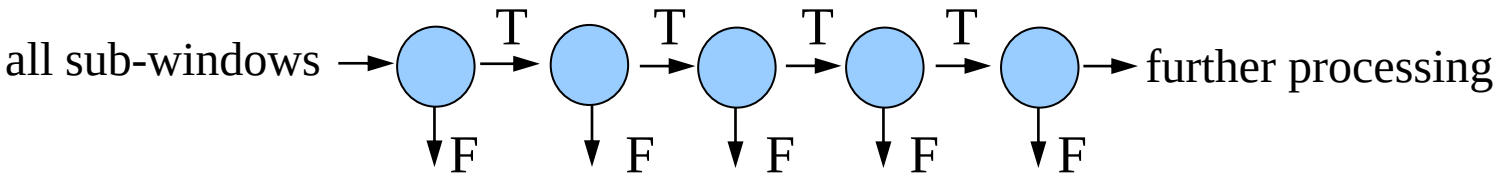
对于一幅待检测图像，该强分类器相当于让所有弱分类器投票，再对投票结果按弱分类器的错误率加权求和，将此和进一步与平均结果比较得出最终的结果。

7.11 基于Adaboost的人脸检测

- **其它补充说明1**
  - 积分图：快速计算指定区域内的像素灰度的总和，因此这个概念只与计算Haar特征Haar(x, y, w, h, type)有关。
  - 我们已经获得了强分类器，按理说就结束了对adaboost的训练。为什么要建立cascade结构？
    - 是的。按经典的adaboost，我们已经获得了一个强分类器。
    - 这个主要是两点考虑。一是速度，级联的时候有出口能迅速过滤掉容易搞定的负样本，这样后面计算量就少很多了; 二是分治，早期的特征表达能力不够强，这样级联能让后面的分类器训练时有针对性地解决当前留下的难样本

7.11 基于Adaboost的人脸检测

- **其它补充说明2**
  - 我们已经获得了强分类器，按理说就结束了对adaboost的训练。为什么要建立cascade结构？
    - 是的。按经典的adaboost，我们已经获得了一个强分类器。
    - 这个主要是两点考虑：
      - 一是速度，级联的时候有出口能迅速过滤掉容易搞定的负样本，这样后面计算量就少很多了；
      - 二是分治，早期的特征表达能力不够强，这样级联能让后面的分类器训练时有针对性地解决当前留下的难样本



7.11 基于Adaboost的人脸检测

- **其它补充说明3**

大家应该看到背后所隐含的特征选择功能！

Thank All of You!