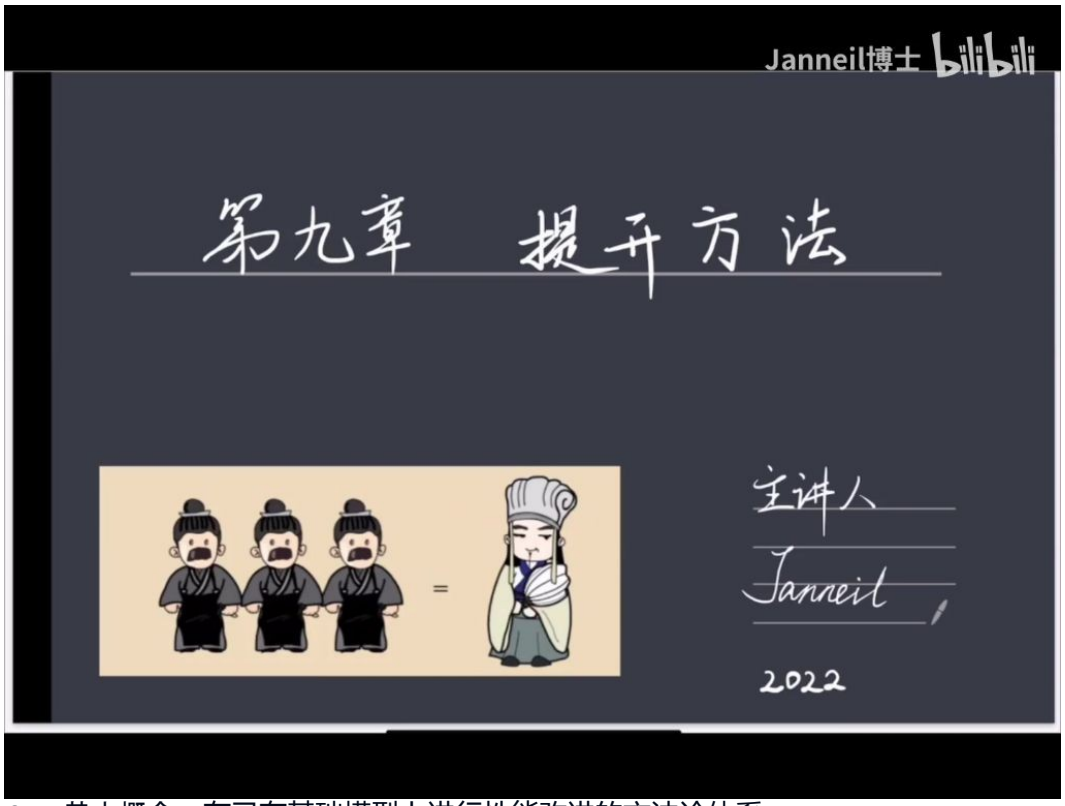
一、提升方法 ***00:01***

1. 提升方法概述 00:10

1）提升方法三要素 00:21

- **核心问题**：学习提升方法需要明确三个关键问题：What（定义）、Why（原因）、How（实现方法）
- **学习路径**：从概念理解到应用场景再到具体实现，形成完整认知链条

2）提升方法定义 00:42

- 

- 基本概念：在已有基础模型上进行性能改进的方法论体系
- **核心特征**：通过迭代优化逐步提高模型精度，具有渐进式改进的特点
- **形象比喻**：类似"站在巨人肩膀上"的程，每个改段都建立在前一段成果基上

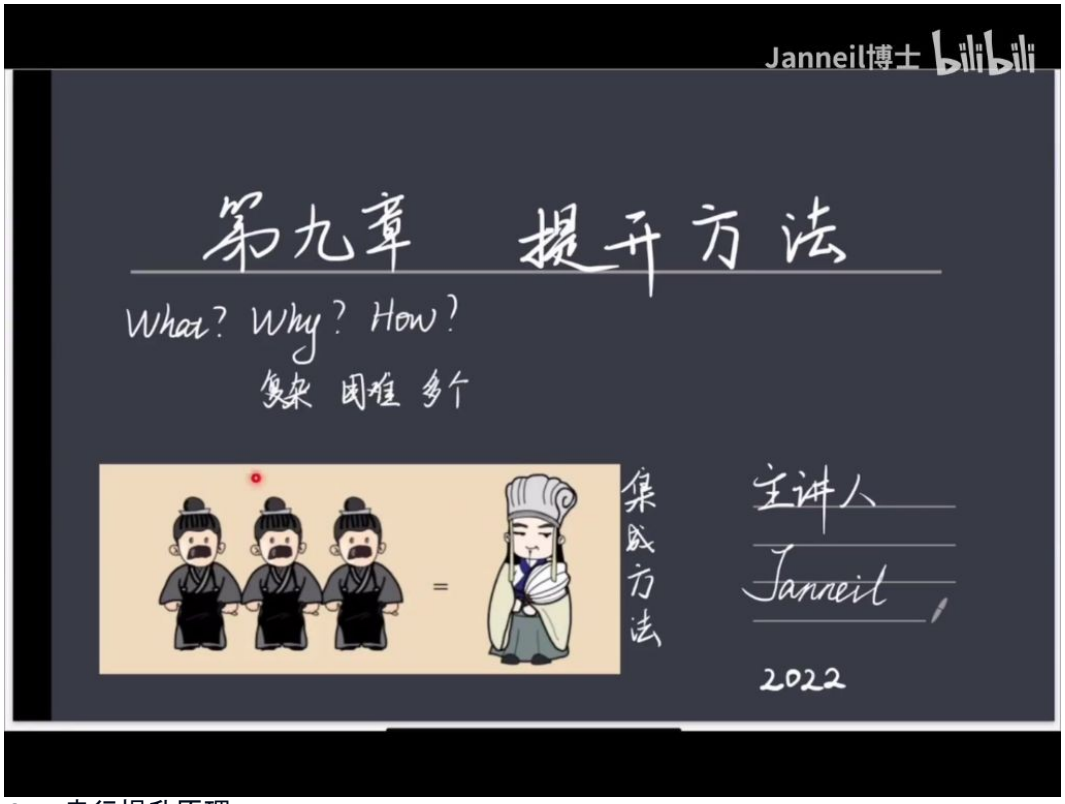
2. 集成方法 01:01

1）集成方法分类 02:34

- 同质集成方法 02:52
  - 基本定义：使用相同类型基学习器构建的集成系统
  - **结构特点**：可分为串行和并行两种拓扑结构
  - 典型代表：
    - 串行结构：Boosting 系列（AdaBoost、GB 等）
    - 并行结构：Bagging（如随机森林）
- 并行集成方法 04:27
  - 工作机制：多个基学习器独立训练后结果聚合
  - **技术特点**：通过 Bootstrap 抽样生成多样化的训练子集
  - **优势分析**：天然适合并行计算，训练效率较高
  - 典型应用：随机森林（由多棵决策树构成的"森林"）

3. 提升方法原理 05:18

1）三个臭皮匠原理 05:24



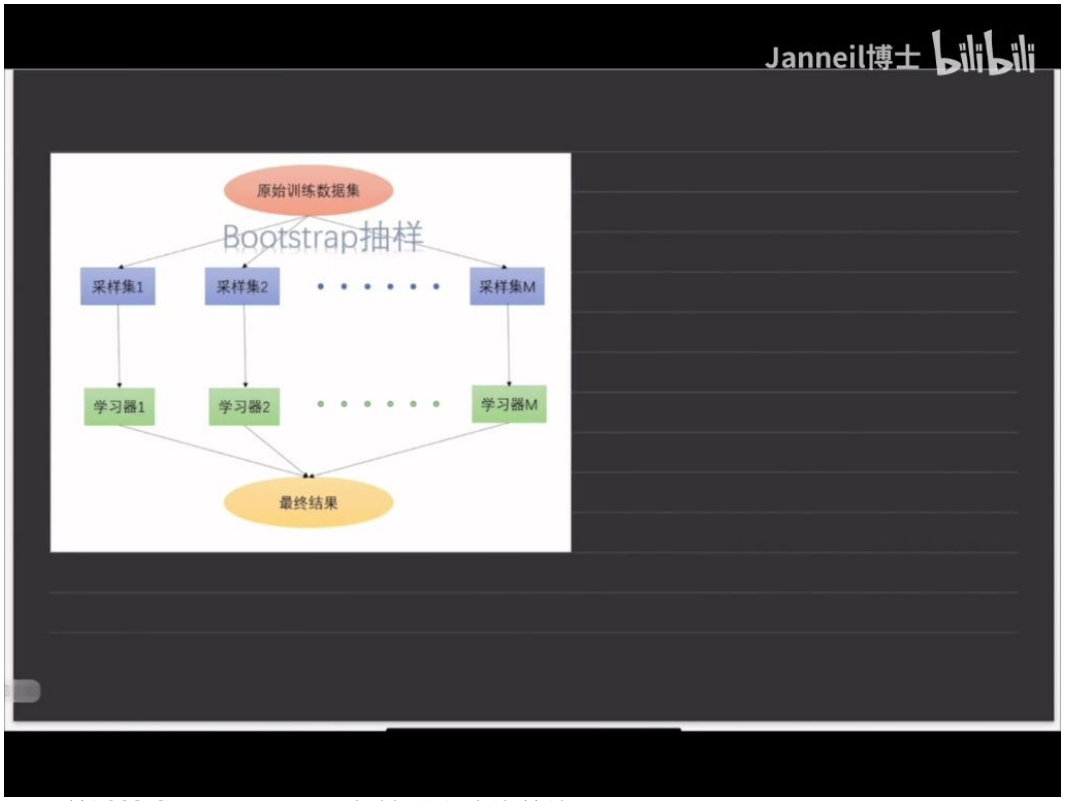
- 串行提升原理：
  - 通过车轮战式迭代，每个后续模型修正前序模型的错误
  - 示例：将领作战胜率从 0.7→0.8→0.9 的渐进提升过程
  - 数学本质：误差逐步修正的递推优化
- 并行提升原理：
  - 采用概率乘积法则整合多个弱分类器
  - 示例：三个独立分类器（ 0.6,0.55,0.45 ）通过 1-(1-0.6)(1-0.55)(1-0.45)≈0.9 实现效果提升
  - 物理类比：并联电路"任一通路导通即成功"的工作模式

2）异质集成方法 08:13

- **核心特征**：组合不同类型的基础学习器
- 工作流程：前层模型输出作为后层模型输入
- **结构类比**：类似神经网络的多层处理架构
- 典型应用：Stacking 等高级集成技术

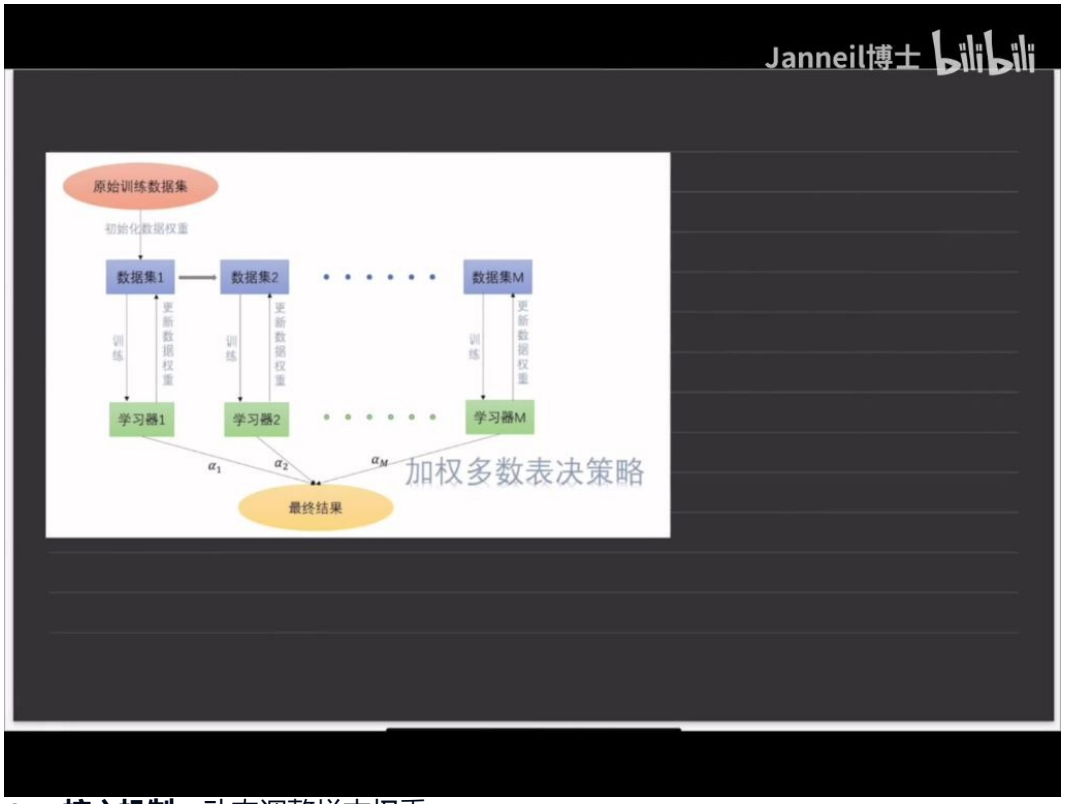
4. 集成方法流程图 09:18

1）Bagging 流程图 09:25

- 

- **关键技术**：Bootstrap 重复抽样构建伪总体
- **实现步骤**：
  - 从原始数据集进行有放回抽样
  - 生成 M 个独立训练子集
  - 并行训练基学习器
  - 通过投票或平均得到最终结果
- **优势特点**：有效降低模型方差，增强泛化能力

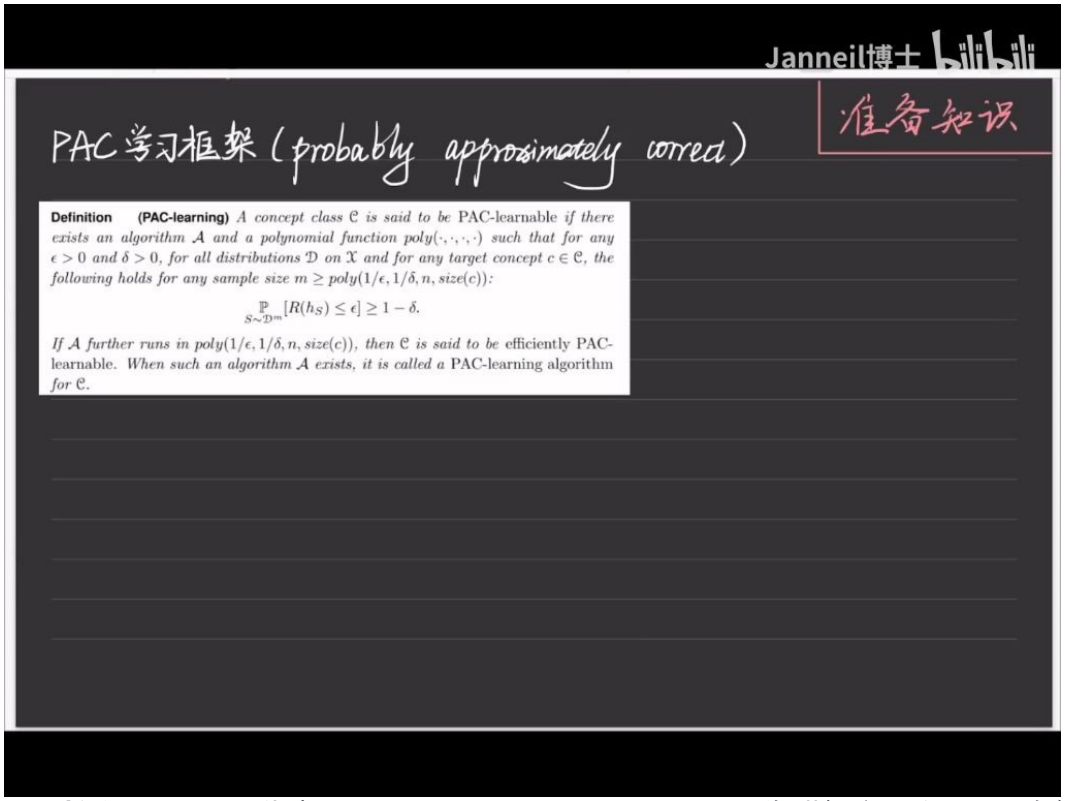
2）AdaBoost 流程图 10:45

- 

- **核心机制**：动态调整样本权重
- 工作流程：
  - 初始化样本权重分布
  - 迭代训练过程中：
    - 增加错分样本权重
    - 降低正确分类样本权重
  - 最终采用加权多数表决
- **数学特点**：通过指数损失函数实现误差最小化
- **优势分析**：特别适合处理分类边界不清晰的问题

二、PAC 学习框架 13:42

1. PAC 学习定义 13:54



- **核心思想**：PAC 代表"Probably Approximately Correct"，即渐进概率正确，强调在概率意义下达到近似正确的学习效果。

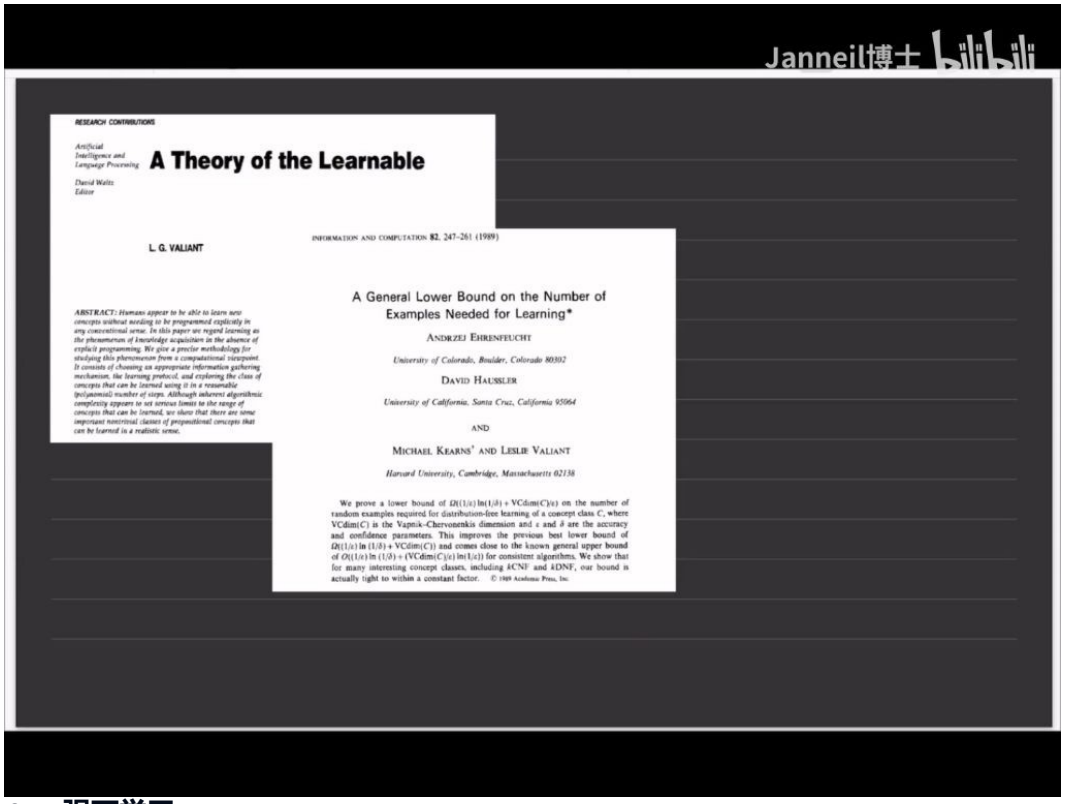
形式化定义：概念类C是 PAC 可学的，当存在算法 A 和多项式函数$poly(\cdot)$，使得对于任意$\epsilon>0$、$\delta>0$、任意分布 D 和任意目标概念$c \in C$，当样本量$m \geq poly(1/\epsilon, 1/\delta, n, ¿c)$¿时，有：

- $P_{S\ D^m}[R(h_S) \leq \epsilon] \geq 1-\delta$
- 高效 **PAC 学习**：若算法 A 的运行时间也是多项式复杂度，则称C是高效 PAC 可学习的。

2. PAC 学习符号说明 14:35

- **输入输出空间**：
  - X：输入空间（实例空间）
  - Y：输出空间（标签空间）
- 概念类：
  - C：希望学习的概念类（人类视角）
  - H：所有可能概念的集合（上帝视角）
- 本表示：
  - 观测样本：$T=\{(x_1, y_1), \ldots, (x_m, y_m)\}$
  - 映射样本：$S=\{(x_1, c(x_1)), \ldots, (x_m, c(x_m))\}$
- **泛化误差**：
  - $R(h)$¿$P_{x\ D}[h(x) \neq c(x)]$，表示概念 h 的错误率
- 多式参数：
  - $poly(1/\epsilon, 1/\delta, n, ¿c)$¿中：
    - ϵ：精度参数
    - δ：置信参数
    - n：输入维度
    - size(c)：概念 c 的表示复杂度

3. 强可学习与弱可学习 17:06

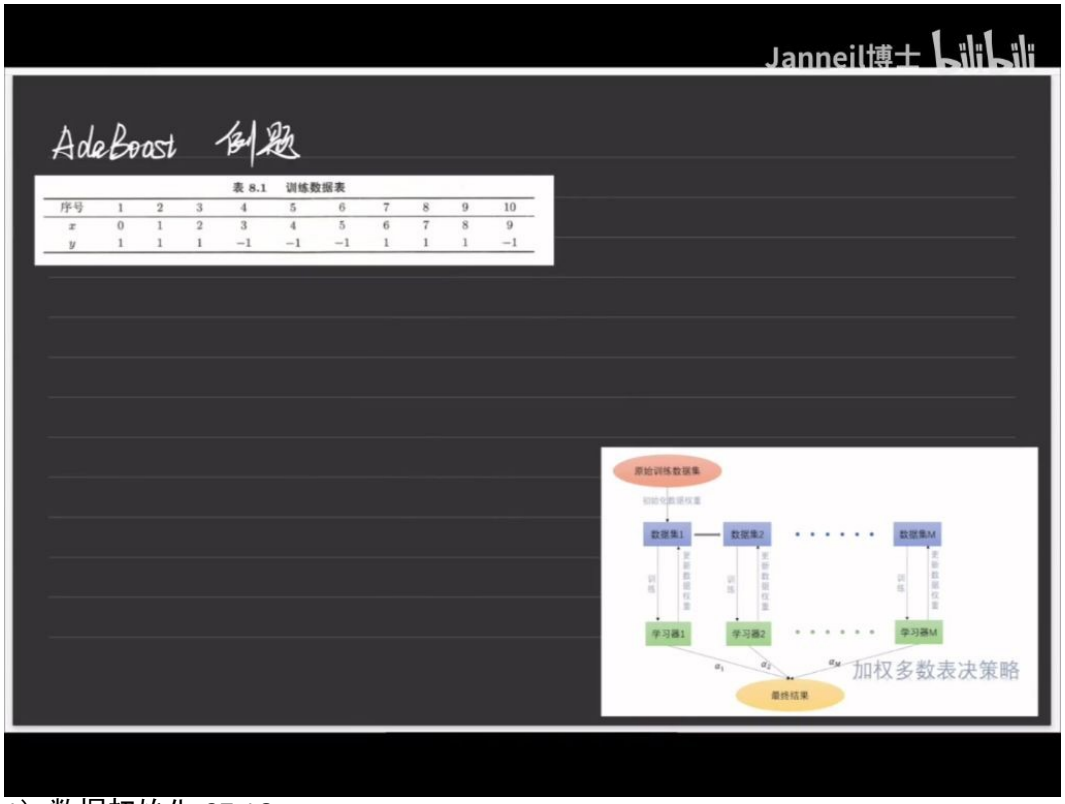


- **强可学习**：
  - 等同于 PAC 可学习
  - 要求错误率可以任意小（$R(h) \leq \epsilon$）
- **弱可学习**：
  - 只需比随机猜测略好
  - 正确率只需超过 50%一个微小量
- **等价关系**：
  - 由 Schapire 在 1990 年证明：概念类 C 是弱可学习的当且仅当它是强可学习的
  - 这意味着可以通过组合多个弱学习器来构造强学习器
- 提升方法核心：
  - 两个关键问题：
    - 弱学习器的选择
    - 如何组合弱学习器成为强学习器
  - 这是 Boosting 方法的基础理论支撑

三、AdaBoost 算法 26:53

1. 例题:二分类问题 27:01

-



1）数据初始化 27:16

- **问题类型**: 解决的是二分类问题，标签 y 取值为{+1,-1}
- **初始权重设置**:
  - 样本数量为 10 个
  - 每个样本初始权重$\omega_{1i}=\frac{1}{10}$（$i=1,2,\ldots,10$）
  - 原因：最初对所有样本一无所知，故采用等权重分配

2）弱分类器选择 28:00

- 模型: 采用"一刀切"的简单分类器
  - 决策规则：在某个阈值点将数据分为正类和负类
  - 特点：这种简单模型被称为弱分类器，为构建强分类器做准备
- 方法:
  - 尝试所有可能的分割点（共 9 个）
  - 选择误分类样本最少的分割点
  - 示例：当分割点为 2.5 时，误分类 3 个样本（误判率 0.3）；分割点为 5.5 时误分类 6 个样本
- 最优分类器:
  - 确定分割点为 2.5
  - 决策规则：$x<2.5$为正类，$x\geq 2.5$为负类
  - 误判样本为第 7、8、9 号样本

3）权重更新过程 30:03

- **误差率计算**:
  - $e_1=\frac{误判数}{总数}=\frac{3}{10}=0.3$
- **分类器权重计算**:
  - $\alpha_1=\frac{1}{2}\ln(\frac{1-e_1}{e_1})=0.4236$
  - 公式意义：误差率越小，分类器权重越大
- **样本权重更新**:

- 更新公式：$\omega_{2i}=\frac{\omega_{1i}}{Z_1}e^{-\alpha_1 y_i g_1(x_i)}$
- 判断正确：权重降低（$e^{-0.4236}=0.6547$倍）
- 判断错误：权重增加（$e^{0.4236}=1.5275$倍）
- 计算结果：
  - 正确分类样本新权重：0.07143（1-6 号及 10 号）
  - 错误分类样本新权重：0.1667（7-9 号）

- **决策函数构建**:
  - 第一轮决策函数：$sign(f_1(x))=sign(\alpha_1 g_1(x))$
  - 其中$g_1(x)$为第一轮弱分类器

2. AdaBoost 算法流程 39:01

1）算法输入输出 39:07

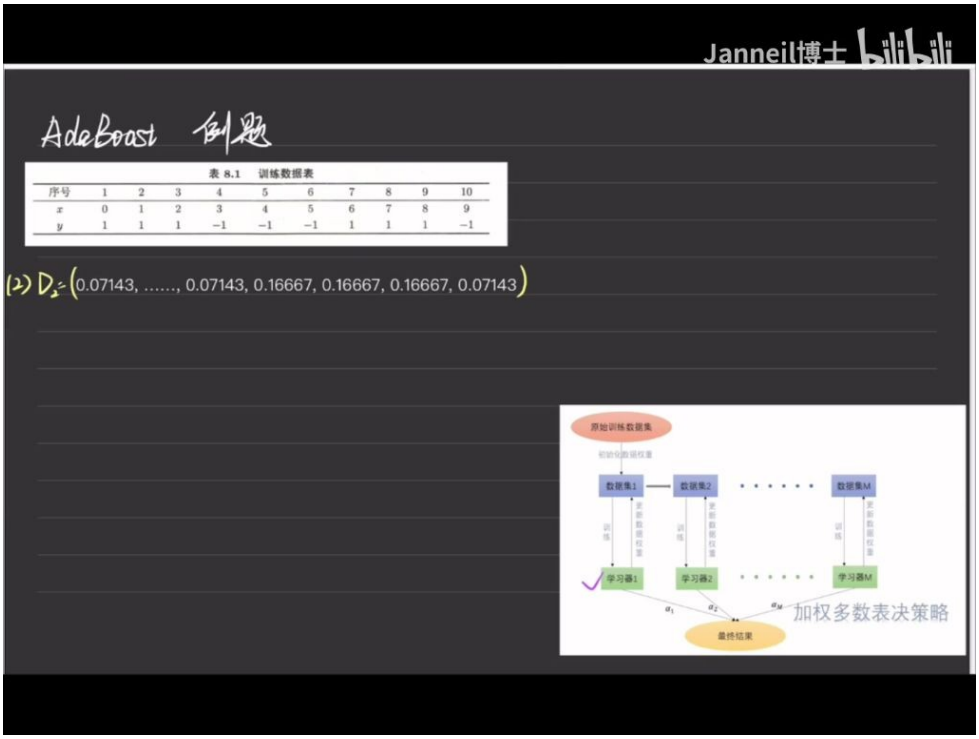
- 数据保持: 训练过程中原始数据 X 和 Y 保持不变
- **权重变化**: 样本权重会随着迭代轮次动态调整，这是 Adaboost 的核心机制

2）初始化步骤 39:41

- 第一轮权重设置
  - **初始值**: 每个样本点的初始权重均为 0.1（假设样本量为 10）
  - **均匀分布**: 首轮采用等权重分配，体现"无知先验"原则
- 权重更新规则
  - 正确本: 权重从 0.1 降至 0.07143，降幅约 28.57%
  - 本: 权重从 0.1 升至 0.1667，增幅达 66.7%
  - **调整目的**: 增强误分类样本的重视程度，削弱正确分类样本的影响
- 误差率计算
  - **加权计算**: 使用更新后的权重计算误差率，而非简单计数
  - 应用场景: 权重直接影响弱分类器的选择标准，决定其在最终模型中的话语权
  - 数学体**现**: 误差率公式为$e_t=\sum_{i=1}^{N} w_i^{(t)} I(y_i \neq h_t(x_i))$，其中$w_i^{(t)}$为第 t 轮第 i 个样本的权重

3）迭代过程 40:27

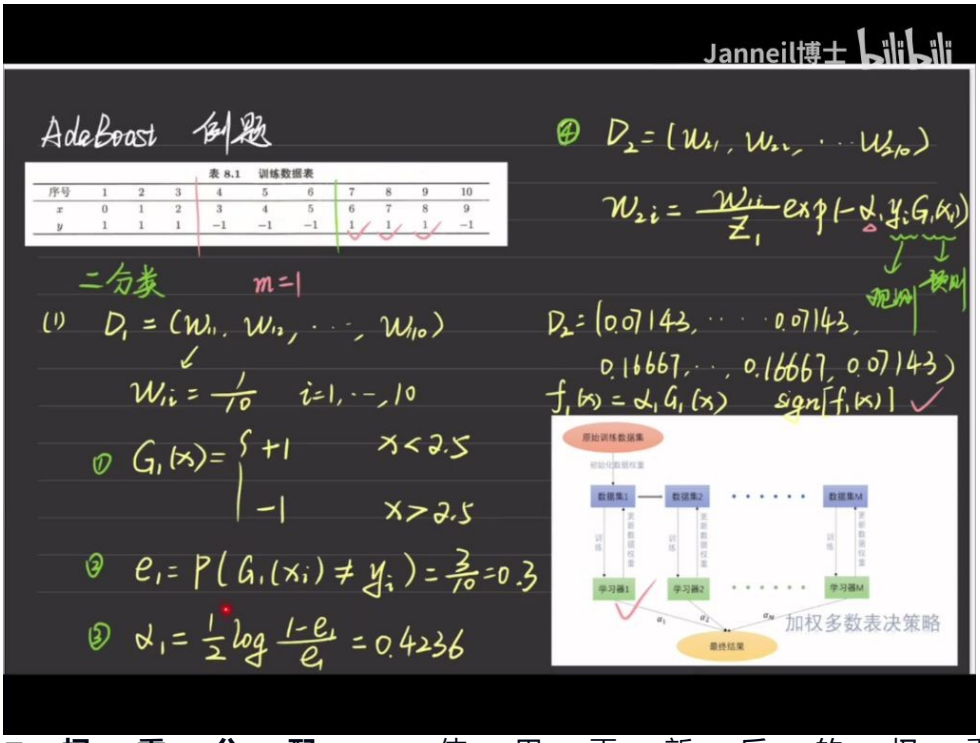
- 例题 1:分类器 g2x 的构建与误差率计算 40:28
  - 分类器构建原理
    -



- 决策边界选择：在 x=8.5 处进行切分，形成正负类区域
- **多数表决原则**：当 x<8.5 时取正类（因该区域 6 个正样本>3 个负样本），x>8.5 时取负类
- **分类器表达式**：$g_2(x)=\begin{cases}+1 & x<8.5\\ -1 & x>8.5\end{cases}$

○ 误差率计算方法

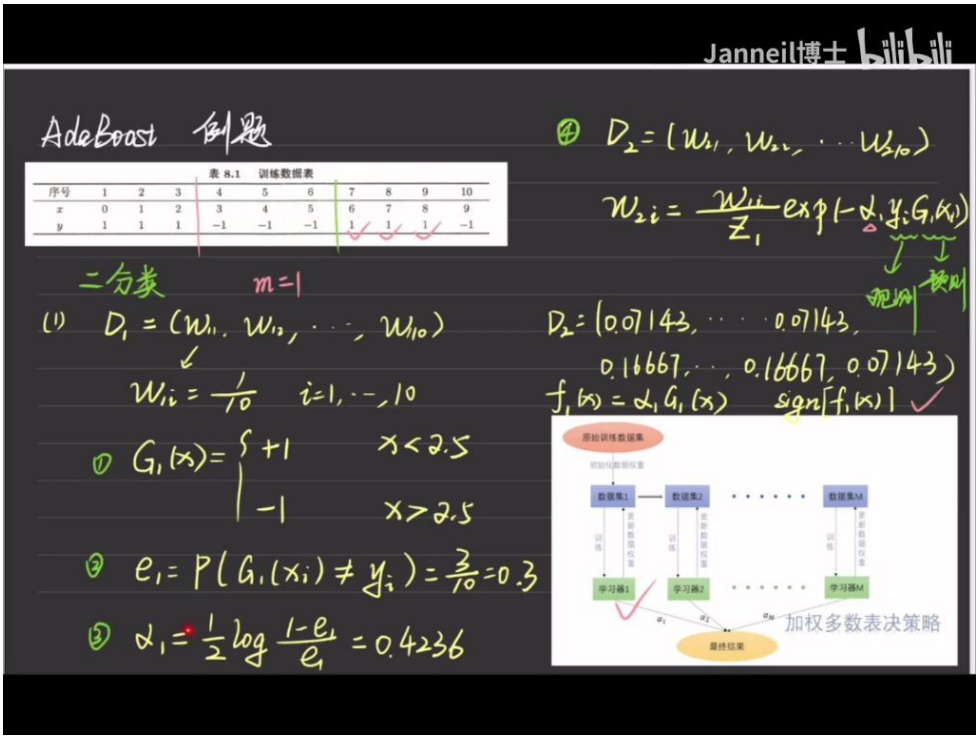
■



- **权重分配**：使用更新后的权重 $D_2=(0.07143,0.07143,0.16667,0.16667,0.16667,0.07143)$
- **误差计算规则**：
  - 正确分类样本：权重×0
  - 错误分类样本：权重×1
- **具体计算**：
  - 样本 4-6 分错：0.07143×1×3=0.2143
  - 其余样本正确：权重×0
  - 总误差率：0.2143

- 例题 1:分类器 g2x 的权重计算与更新 45:10
  - 分类器权重计算

■



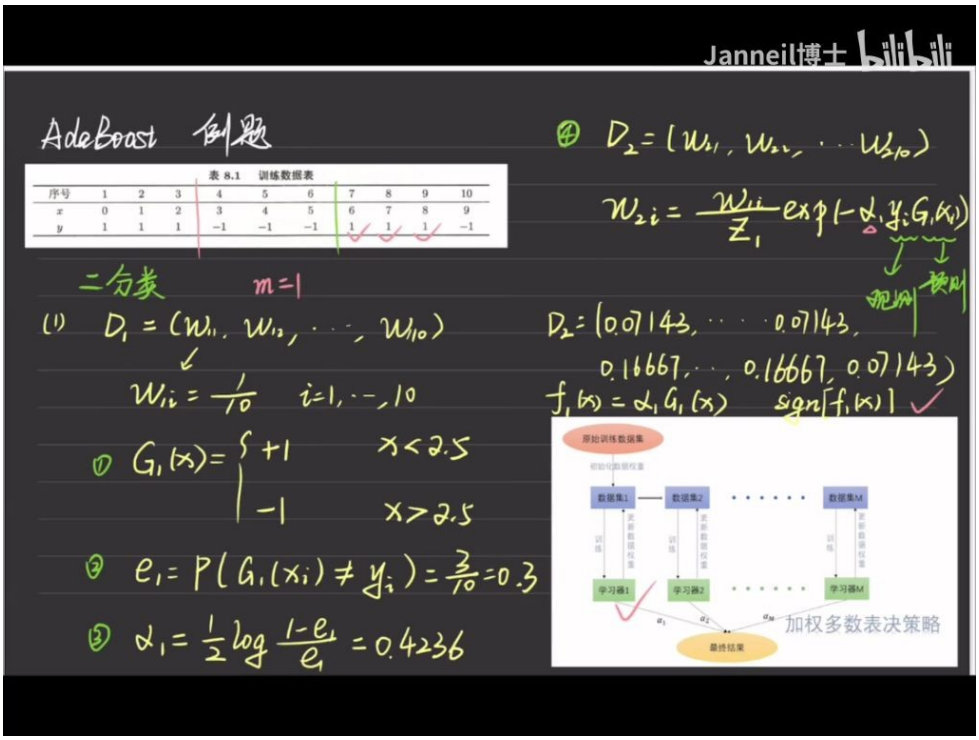
■ **计算公式**：$\alpha_2=\frac{1}{2}\ln\frac{1-e_2}{e_2}$

■ **具体数值**：代入$e_2=0.2143$得$\alpha_2=0.6496$

■ 物理意义：误差率越小，该分类器在最终决策中的权重越大

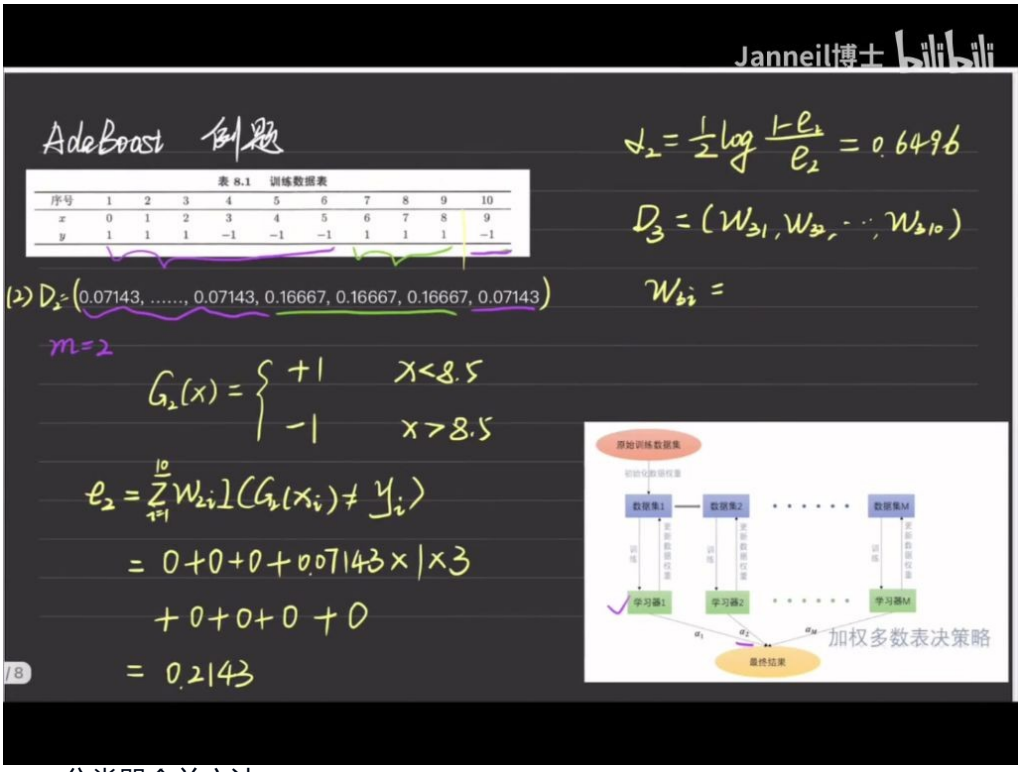
○ 权重更新机制

■



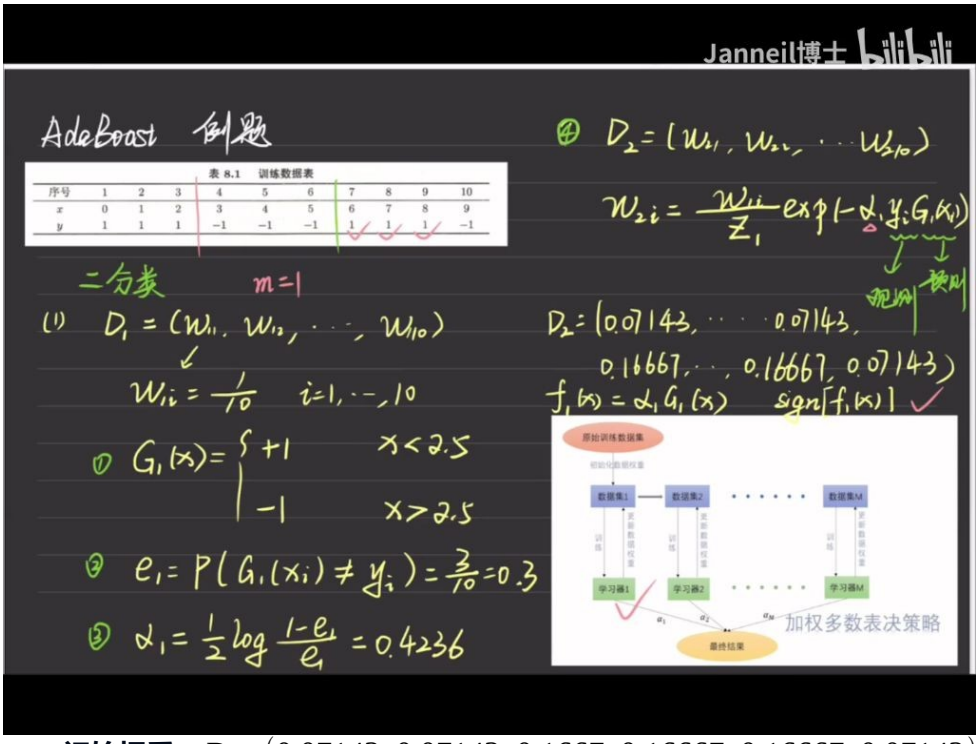
■ **更新公式**：$\omega_{3i}=\frac{\omega_{2i}}{Z_2}e^{-\alpha_2 y_i g_2(x_i)}$

- Z_2：归一化因子

        - 正确分类样本：权重降低（指数项为负）
        - 错误分类样本：权重升高（指数项为正）
    - 更新效果：使下一轮分类器更关注当前分错的样本
- 例题 1:分类器合并与决策函数构建 47:05
    - 

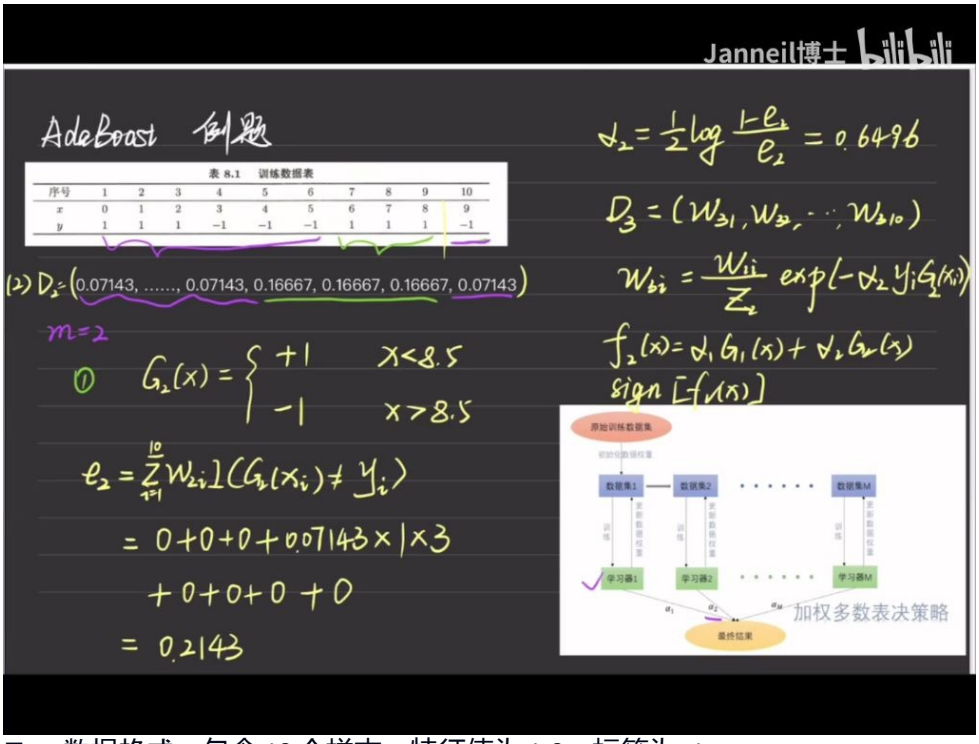
    - 分类器合并方法
        - **合并公式**：$f_2(x)$¿$f_1 g_1(x)+\alpha_2 g_2(x)$，其中$g_1(x)$和$g_2(x)$分别是两个基础学习器，$\alpha$为权重系数
        - 决策函数构建：最终决策函数为符号函数$sgn(f(x))$，通过取符号函数值实现分类决策
        - **权重计算**：$\alpha_1=\frac{1}{2}\ln\frac{1-\theta}{\theta}=0.4236$，其中$\theta$为误差率
    - 训练数据权重更新

-

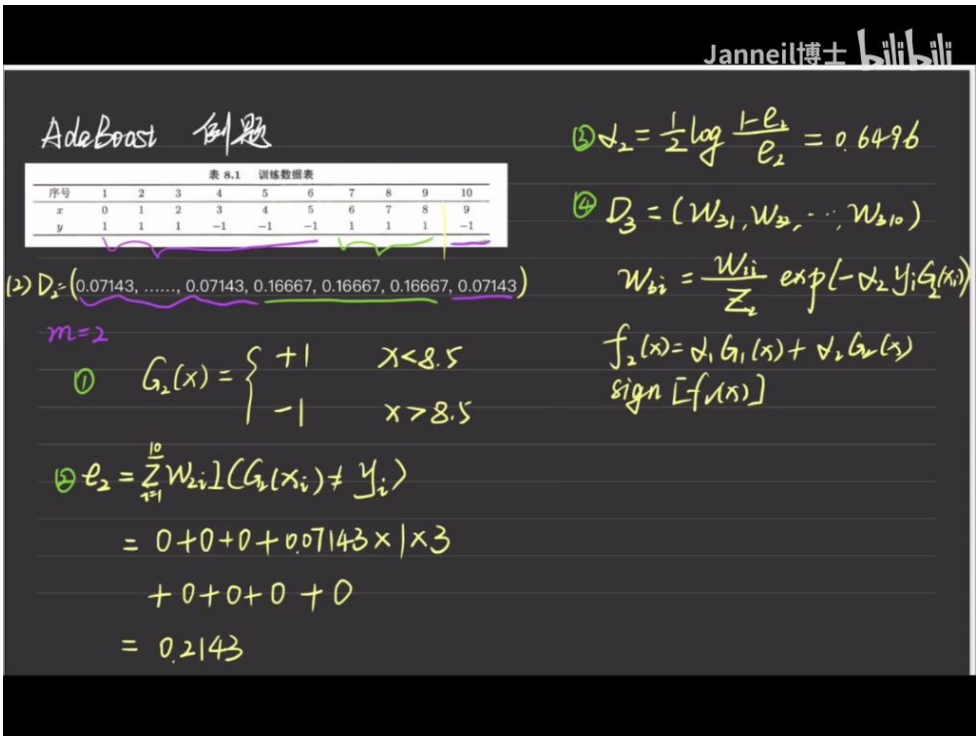


  - **初始权重**：$D_1=(0.07143,0.07143,0.1667,0.16667,0.16667,0.07143)$
  - **误差率计算**：$B_1=P(G_1(x_i)\neq y_i)=\frac{3}{10}=0.3$
  - **权重调整**：根据误差率调整样本权重，误分类样本权重增加
- 例题 1:分类器构建过程总结与问题解答 48:00
  - 分类器构建四步骤
    - 第一步：基学器：通过"一刀切"方法遍历所有可能划分，选择误差率最小的划分点
    - **第二步：权重计算**：根据误差率计算分类器权重α
    - **第三步：样本权重更新**：增加误分类样本权重，减少正确分类样本权重
    - **第四步：模型合并**：将新分类器加权合并到集成模型中
  - 关键问题解答
    - **学习器生成方法**：类似第一轮，遍历所有可能划分（共 9 种切法），选择误差率最小的划分作为$g_2(x)$
    - **误差率优化**：每次选择使加权误差率最小的划分点，保证每轮新增的分类器能有效改进模型
    - **迭代特性**：通过多轮迭代逐步改进模型，每轮关注前一轮误分类的样本
  - 训练数据表示例

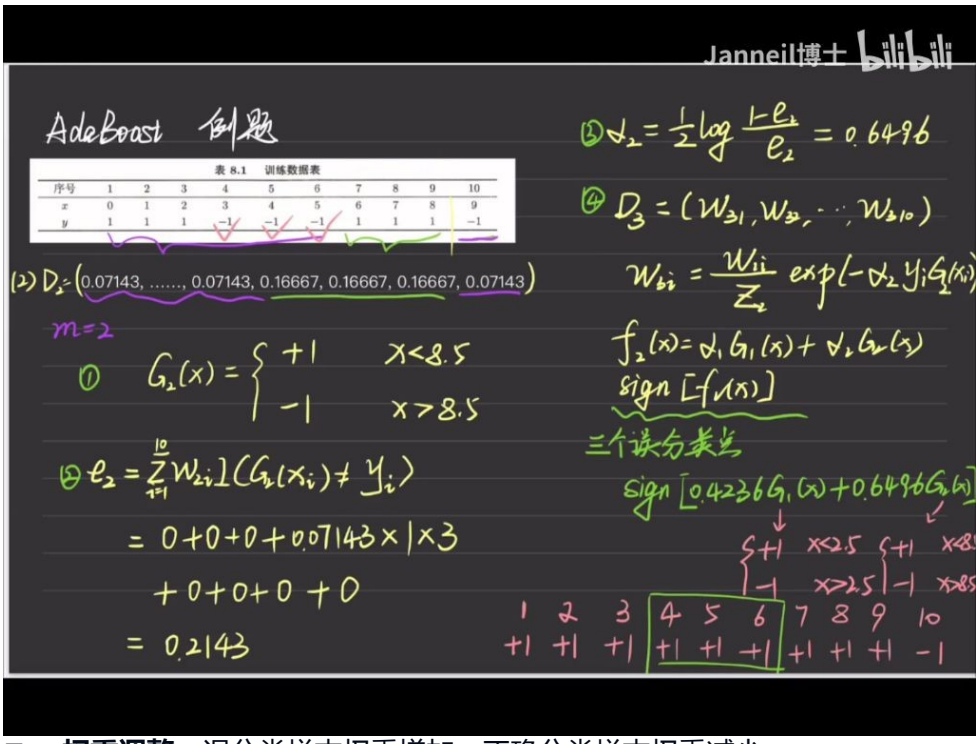
- ■



  - ■ 数据格式：包含 10 个样本，特征值为 1-9，标签为±1
  - ■ **样本分布**：正例集中在特征值较大区域（ 6,7,8 ），负例集中在特征值较小区域（ 3,4,5 ）
  - ■ **权重变化**：展示了两轮迭代后的权重分布变化情况
- 例题 1:分类器误分类点寻找与验证 49:17
  - AdaBoost 算法误分类点分析
  - ■



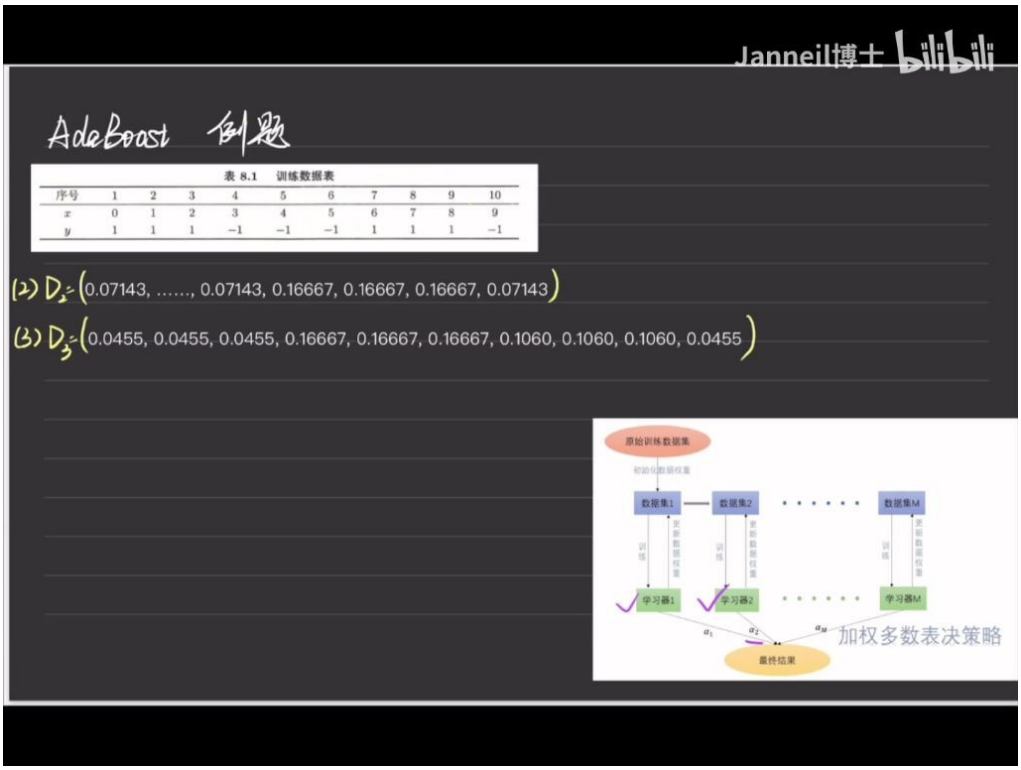
  - ■ **初始权重分配**：在第二轮迭代中，数据权重 D_2为(0.07143,...,0.07143,0.16667,0.16667,0.16667,0.07143)，其中第 4-6 个样本权重较高

- **分类器组合**：当前组合分类器为$f_2(x)$¿$0.4236G_1(x)+0.6496G_2(x)$，其中：
  - $G_1(x)$：当x<2.5时输出+1，否则-1
  - $G_2(x)$：当x<8.5时输出+1，否则-1
- 误分类点验证过程
  -
  - 方法：逐个样本代入组合分类器计算符号函数值
    - **样本 1-3**（x=0,1,2）：$G_1=+1$，$G_2=+1\rightarrow f_2(x)>0\rightarrow$ 预测+1（正确）
    - **样本 4-6**（x=3,4,5）：$G_1=-1$，$G_2=+1\rightarrow f_2(x)=0.226>0\rightarrow$ 预测+1（实际-1，错误）
    - **样本 7-9**（x=6,7,8）：$G_1=-1$，$G_2=+1\rightarrow$ 预测+1（部分正确）
    - **样本 10**（x=9）：$G_1=-1$，$G_2=-1\rightarrow$ 预测-1（正确）
  - **误分类点**：第 4、5、6 号样本被错误分类，对应$x=3,4,5$的特征值
  - **错误率计算**：$\epsilon_2=0+0+0+0.07143\times1\times3+0+0+0+0=0.2143$
- 权重更新原理
  -
  - 

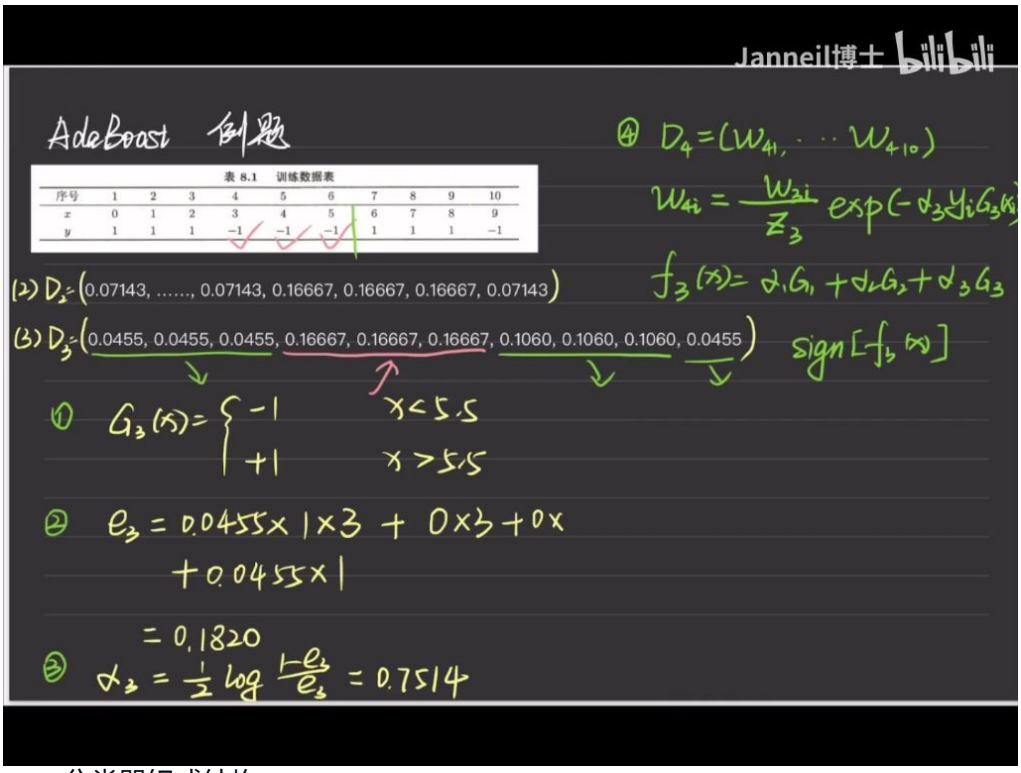
  - **权重调整**：误分类样本权重增加，正确分类样本权重减少
  - **分类器权重**：$\alpha_2=\frac{1}{2}\ln\frac{1-\epsilon_2}{\epsilon_2}=0.6496$
  - 迭代效果：经过两轮迭代后，两个基分类器分别有 3 个误分类点，通过加权组合逐步提升整体分类准确率
- 例题 1:分类器权重更新与验证 54:25
  - 权重更新机制
    - **权重调整原则**：分类错误的样本权重增加（加强重视），分类正确的样本权重减少（削弱重视）
    - **具体变化示例**：
      - 第 4-6 个样本（误分类点）：权重从 0.0743 提升至 0.1667
      - 第 1-3 个样本（正确分类）：权重从 0.0743 降至 0.0455
      - 第 7-9 个样本（本轮正确）：权重从 0.1667 降至 0.1060
      - 第 10 个样本（持续正确）：权重从 0.07143 降至 0.0455
  - 权重更新原理

    - 数学表达：$w_{t+1}^{(i)}=\begin{cases} w_t^{(i)}\cdot\alpha & 正确分类 \\ w_t^{(i)}/\alpha & 错误分类 \end{cases}$
    - **动态平衡**：通过权重调整使模型持续关注难分类样本
    - 迭代特性：第二轮权重基于第一轮结果计算，第三轮基于第二轮结果计算
- 例题 1:新分类器构建与误差率计算 56:22
  - 分类器生成过程
    - 决策边界选择：通过遍历所有可能切分点（如$x=5.5$），选择误差率最小的划分
    - **分类规则建立**：
      - $x<5.5$时判为负类（3 正 3 负难以判断）
      - $x\geq 5.5$时判为正类（3 正 1 负明显倾向）
  - 误差率计算
    - **计算过程**：
      - $x<5.5$误判 3 个：$3\times 0.0455\times 1=0.1365$
      - $x\geq 5.5$正确 3 个：$3\times 0.1667\times 0=0$
      - 第 10 个误判：$1\times 0.0455\times 1=0.0455$
    - **总误差率**：$e_3=0.1365+0.0455=0.1820$
    - **权重作用**：误差计算时每个样本的贡献由其当前权重决定
  - 迭代优化特征
    - **自适应调整**：每次迭代后分类器会针对前一轮的弱点进行优化
    - **误差收敛性**：通过权重更新机制，系统会逐步降低整体误差率
    - **关键参数**：当前轮次误差率$e_3=0.1820$将用于计算下一轮分类器权重
- 例题 1:新分类器权重计算与更新 59:03
  - 权重系数 α 的计算
    - **计算公式**: 第 t 个弱分类器的权重系数α_t计算公式为$\alpha_t=\frac{1}{2}\ln\left(\frac{1-e_t}{e_t}\right)$，其中$e_t$是分类误差率
    - **具体计算**：根据例中$e_3=0.1820$，代入公式得$\alpha_3=\frac{1}{2}\ln\left(\frac{1-0.1820}{0.1820}\right)=0.7514$
  - 样本权重更新
    - 更新规则: 第 t+1 轮样本权重$\omega_{t+1,i}=\frac{\omega_{t,i}}{Z_t}e^{-\alpha_t y_i G_t(x_i)}$，其中$Z_t$是规范化因子
    - **计算要素**:
      - $\omega_{3,i}$: 当前轮次样本权重
      - Z_3: 保证权重总和为 1 的规范化因子
      - α_3=0.7514: 当前分类器权重
      - y_i: 样本真实标签
      - $G_3(x_i)$: 当前弱分类器预测结果
  - 集成分类器构建
    - 性合:$f_3(x)¿\alpha_1 G_1(x)+\alpha_2 G_2(x)+\alpha_3 G_3(x)$
    - **最终决策**:$sign(f_3(x))$，取线性组合结果的符号作为最终分类结果
    - **特点**: 通过加权投票机制，强分类器由多个弱分类器组合而成
- 例题 1:最终分类器构建与验证 01:01:33

○



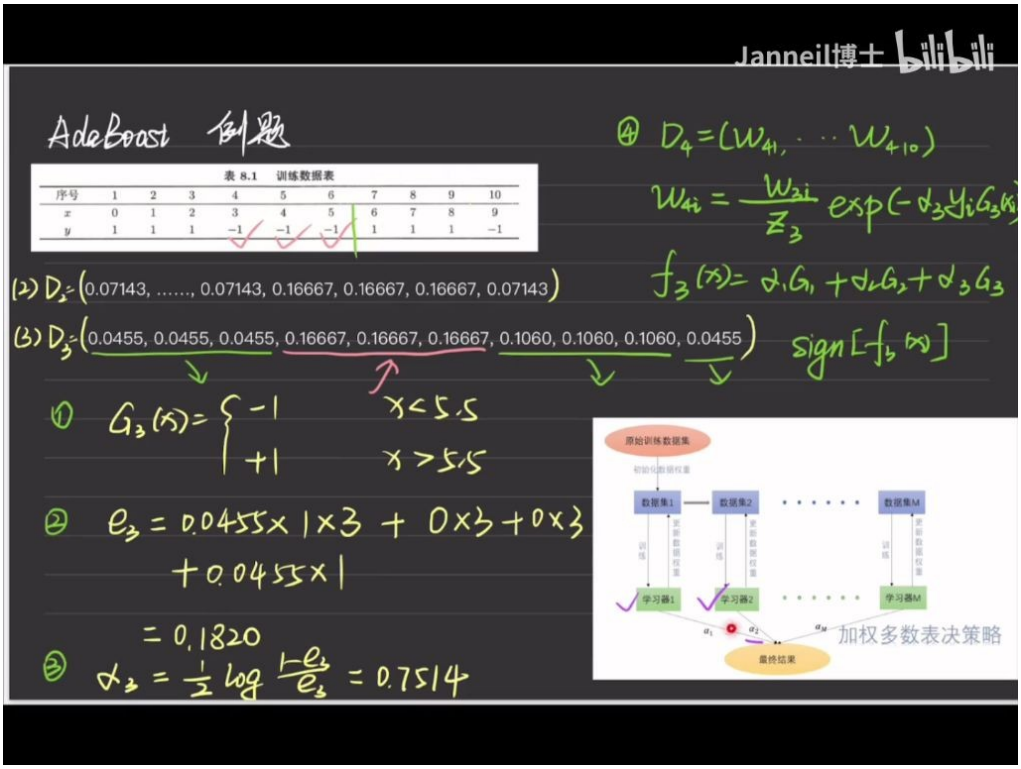
○



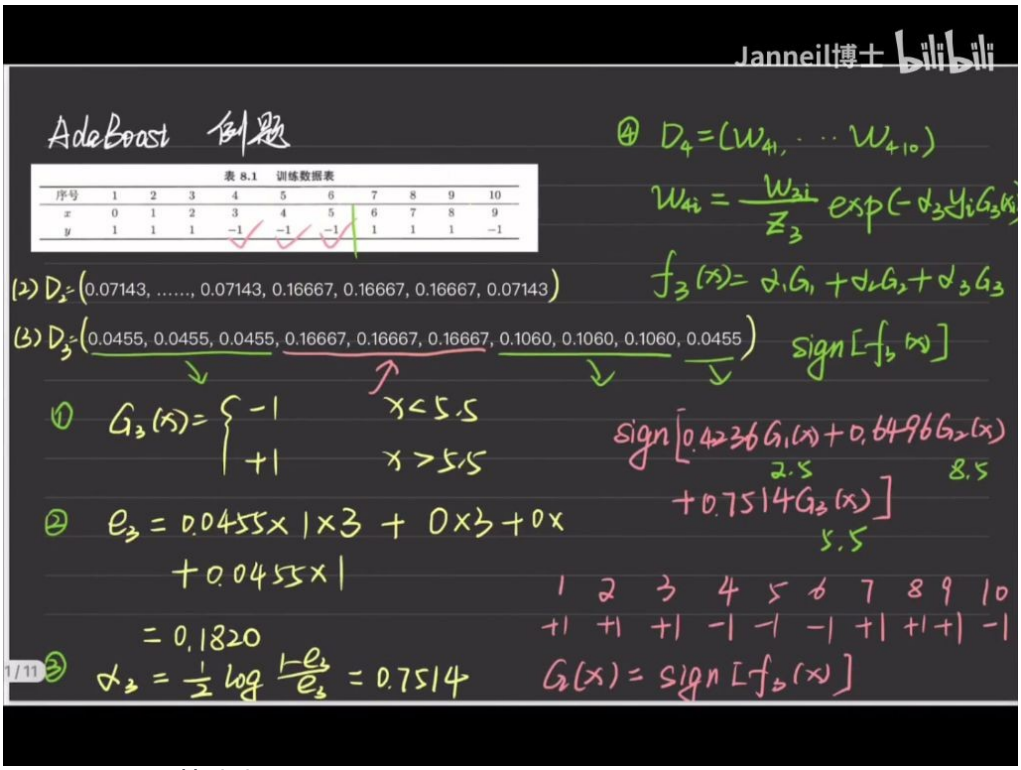
○ 分类器组成结构

- 决策函数形式：最终分类器由三个弱分类器加权组合而成，具体形式为 $G(x)=sign[f_3(x)]$，其中 $f_3(x)¿0.4236\,g_1(x)+0.6496\,g_2(x)+0.7514\,g_3(x)$
- **弱分类器阈值**：
  - g_1以 2.5 为分界点
  - g_2以 8.5 为分界点

        - g_3以 5.5 为分界点
    - 样本点分类验证
        - **样本 1(x=0)**:
            - $g_1=+1, g_2=-1, g_3=-1$
            - 计算结果 0.4236-0.6496-0.7514>0→分类为+1
        - **样本 4(x=3)**:
            - $g_1=-1, g_2=+1, g_3=-1$
            - 计算结果-0.4236+0.6496-0.7514<0→分类为-1
        - **样本 7(x=6)**:
            - $g_1=-1, g_2=+1, g_3=+1$
            - 计算结果-0.4236+0.6496+0.7514>0→分类为+1
        - **样本 10(x=9)**:
            - $g_1=+1, g_2=-1, g_3=-1$
            - 计算结果 0.4236-0.6496-0.7514<0→分类为-1
    - 分类效果评估
        - 完美分类: 所有样本点均被正确分类，无分类错误点
        - **权重计算**:
            - 误差率$e_2=0.1820$
            - 系数计算$\alpha_2=\frac{1}{2}\log\frac{1-e_2}{e_2}=0.7514$
        - **权重分布**:
            - 初始权重D_1为均匀分布
            - 迭代后权重D_4呈现差异化分布
- 例题 1:分类器升级过程总结 01:05:55
    -



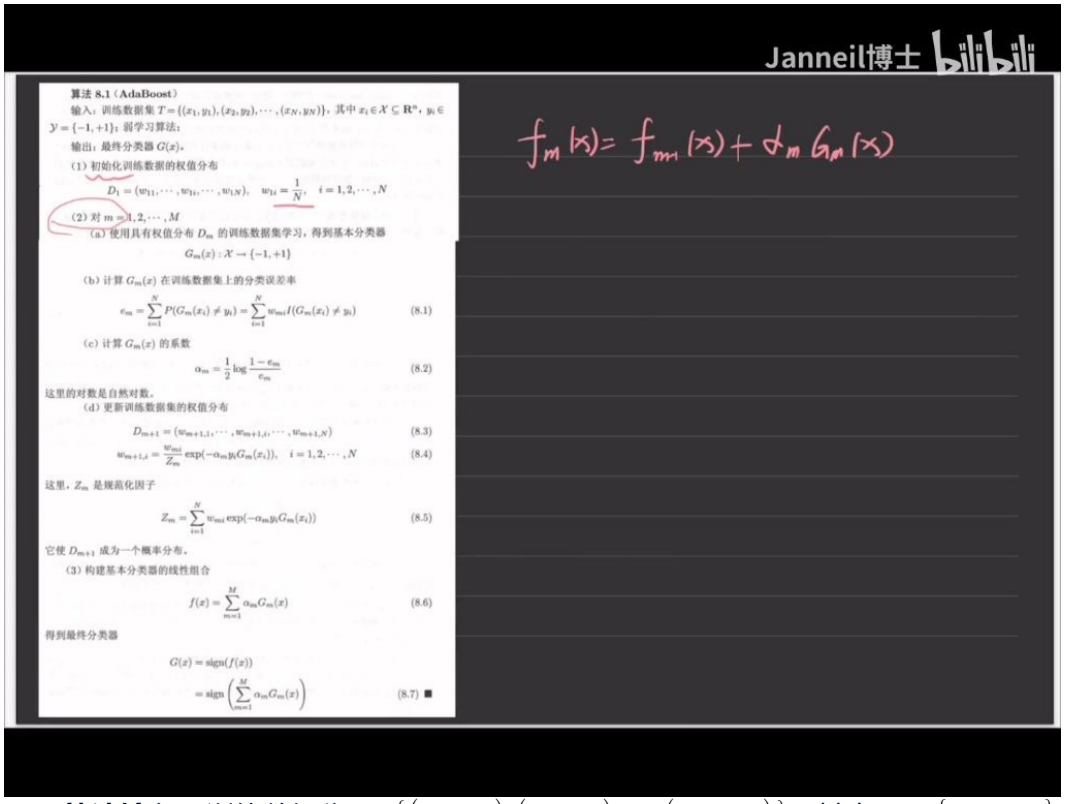
○



- Adaboost 算法步骤
  - **权重初始化**: 初始权重D_1为均匀分布，示例中为(0.07143, 0.07143, 0.1667, 0.1667, 0.167, 0.07143)
  - **错误率计算**：计算当前分类器的加权错误率e_2，示例中$e_2=0.0455\times1\times3+0\times3+0.0455\times1=0.1820$
  - **分类器权重**: 通过公式$\alpha=\frac{1}{2}\log\frac{1-e}{e}$计算，示例中$\alpha_2=0.7514$
  - **权重更新规则**: 使用$D_{t+1}=\frac{D_t}{Z_t}e^{-\alpha_t y_i h_t(x_i)}$更新样本权重，其中$Z_t$为归一化因子
- 分类器组合过程
  - 逐步升: 从学习器 1 升级到学习器 2，再到学习器 3，最终组合成强学习器 G(x)
  - **加权投票策略**: 最终分类器采用加权多数表决，$G(x)=sign(\sum_{t=1}^{T}\alpha_t h_t(x))$
  - **训练数据变化**: 展示了不同轮次中样本权重的变化过程，如第二轮权重变为(0.15,0.045,0.15,0.1667,0.16687,0.167,0.1060,0.1060,0.1060,0.045)
- 关键公式说明
  - **权重更新公式**:$D_2=\frac{D_1}{E}e_1\in O_2U_2$，其中 E 为归一化因子
  - **组合分类器**:$f_3(x)=\alpha_1h_1+\alpha_2h_2+\alpha_3h_3$
  - **最终决策**:$G(x)=sign[f_3(x)]$，取符号函数作为最终输出

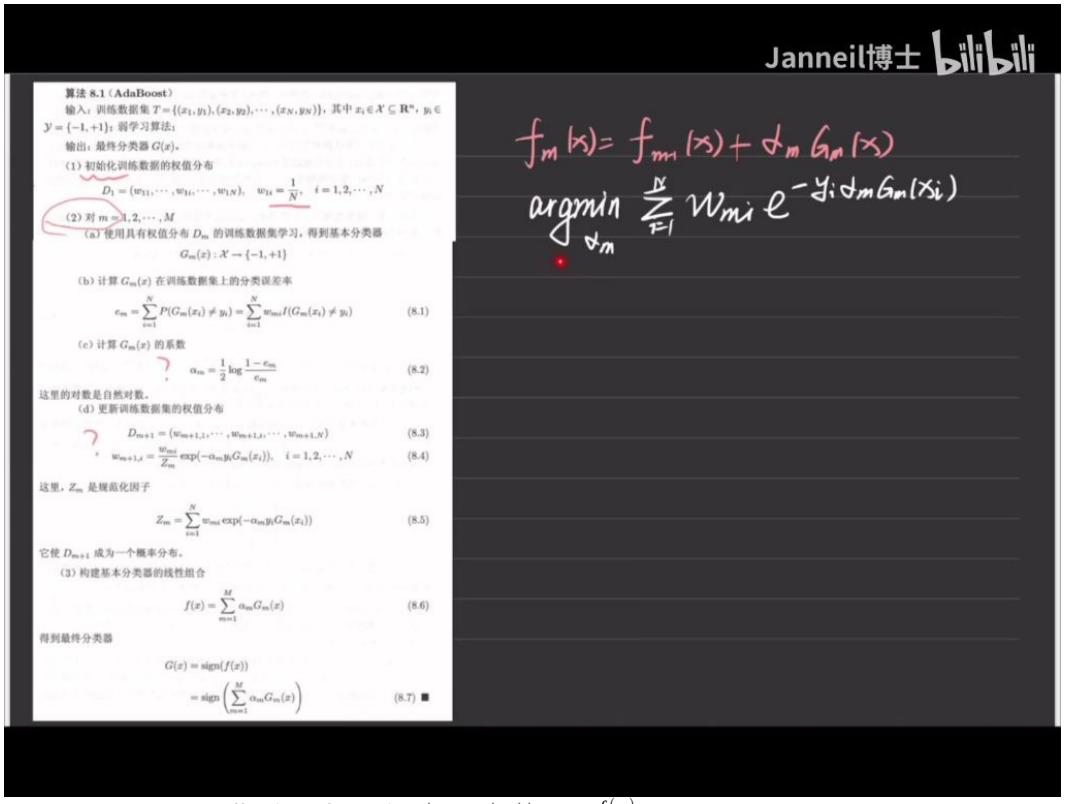
3. 算法理论分析 01:06:25

1）指数损失函数 01:06:34



- **算法输入**：训练数据集$T=\{(x_1,y_1),(x_2,y_2),...,(x_N,y_N)\}$，其中$y_i\in\{-1,+1\}$，以及弱学习算法
- **算法输出**：最终分类器 G(x)
- **初始化权重**：$D_1=(w_{11},...,w_{1N})$，其中$w_{1i}=\frac{1}{N}$，采用等概率权重
- 迭代过程：
  - **训练弱分类器**：使用带权重的训练数据学习得到$G_m(x):X\to\{-1,+1\}$
  - **计算误差率**：$e_m=\sum_{i=1}^{N}w_{mi}I(G_m(x_i)\neq y_i)$
  - **计算系数**：$\alpha_m=\frac{1}{2}\log\frac{1-e_m}{e_m}$（自然对数）
  - 更新**权**重：$w_{m+1,i}=\frac{w_{mi}}{Z_m}\exp(-\alpha_m y_i G_m(x_i))$，其中$Z_m$为规范化因子
- **最终分类器**：$G(x)=sign(\sum_{m=1}^{M}\alpha_m G_m(x))$

2）目标函数优化 01:18:57



- 失函数：采用指数损失函数$L(y,f(x))=e^{-yf(x)}$
- **总损失最小化**：目标是使$\sum_{i=1}^{N} e^{-y_i f(x_i)}$最小
- 迭代公式**推导**：
  - 将$f_m(x)=f_{m-1}(x)+\alpha_m G_m(x)$代入损失函数
  - 拆分为正确分类和错误分类两部分：
    - 正确分类：$e^{-\alpha_m}$乘以对应权重
    - 错误分类：e^{α_m}乘以对应权重
  - 最终表达式：$e^{-\alpha_m}\sum_{正确分类} w_{mi}+e^{\alpha_m}\sum_{错误分类} w_{mi}$
- **权重更新原理**：
  - 错误分类样本权重增加（e^{α_m}项）
  - 正确分类样本权重减少（$e^{-\alpha_m}$项）
- 系数α_m的由来：通过最小化损失函数推导得出，与分类误差率e_m相关

四、AdaBoost 算法 01:22:46

1. 算法步骤 01:22:50

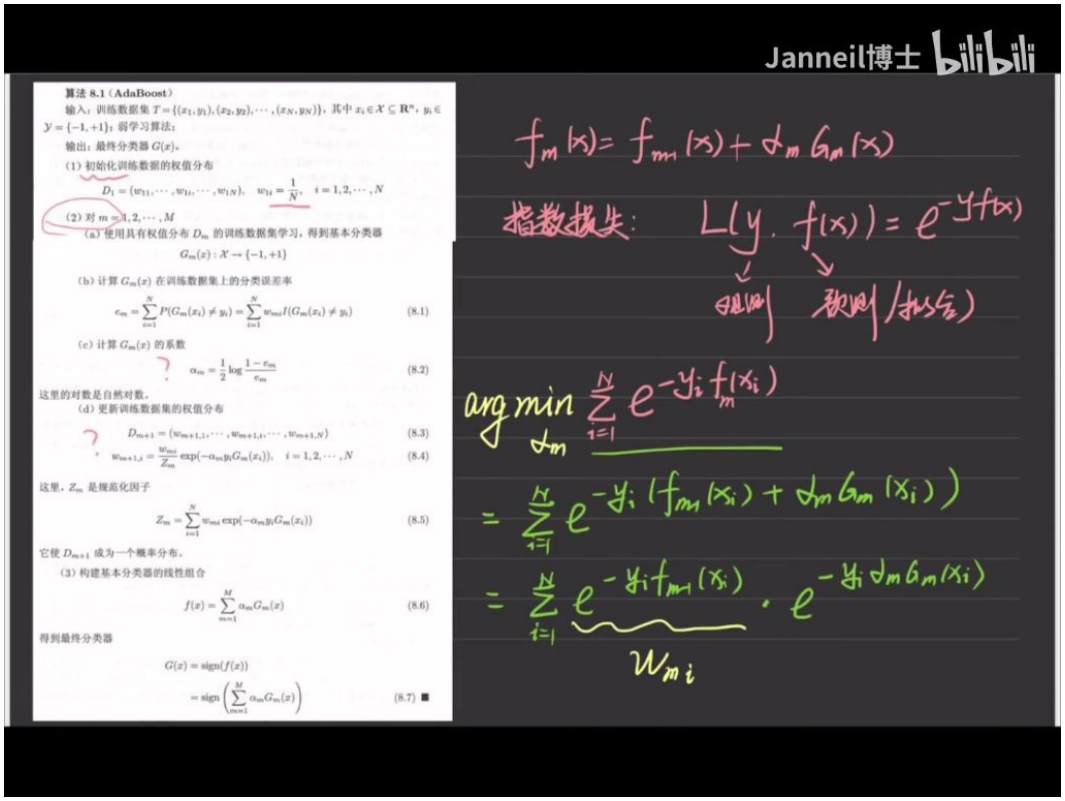
1）初始化权值分布

- **权值设定**：初始时所有样本权重相等，$w_{1i}=\frac{1}{N}$，其中 N 为样本数量
- **分布表示**：用$D_1=(w_{11},\ldots,w_{1N})$表示初始权值分布

2）分类误差率计算 01:23:13

- **计算方法**：$e_m=\sum_{i=1}^{N} w_{mi} I(G_m(x_i)\neq y_i)$，即所有误分类样本的权重之和
- **弱分类器要求**：误差率必须小于 0.5（比随机猜测要好），这是算法有效的前提条件
- 指标意义：反映了当前弱分类器在加权训练集上的错误率

3）分类器系数计算 01:25:47



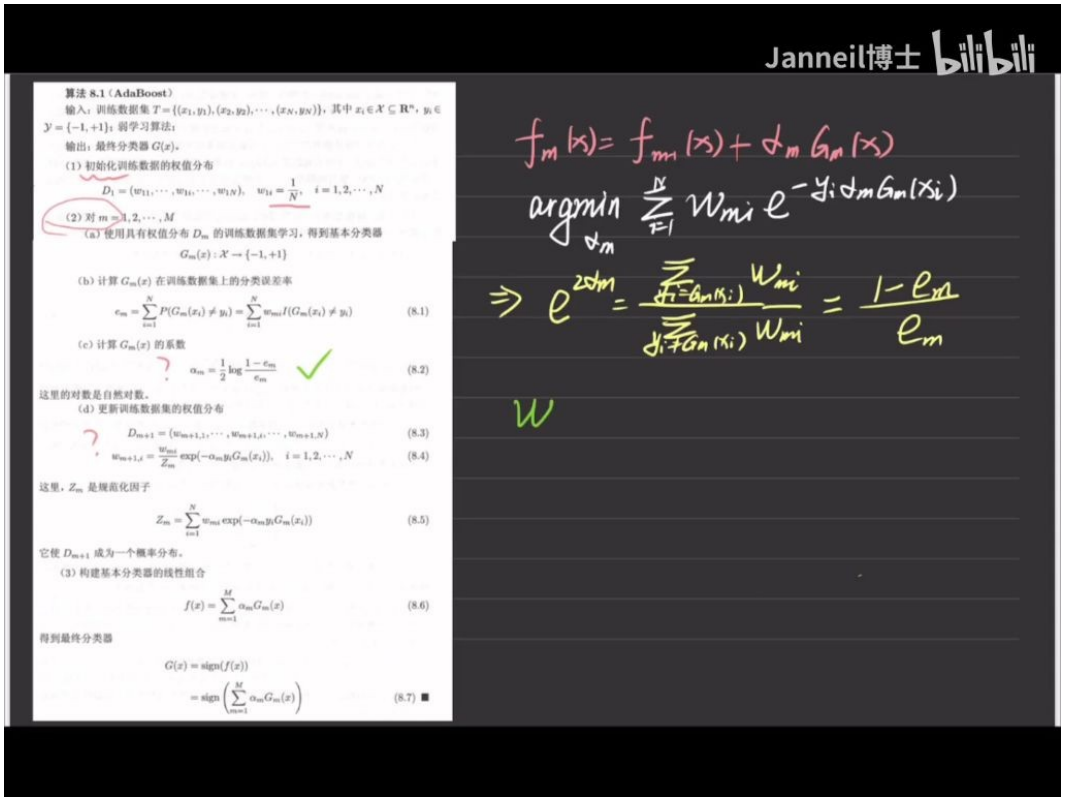
- **系数公式**：$\alpha_m=\frac{1}{2}\ln\frac{1-e_m}{e_m}$，使用自然对数
- 推导过程：通过最小化指数损失函数，利用费马原理求导得到
- **特性分析**：
  - 当e_m<0.5时，α_m>0
  - 误差率越小，系数越大，表示该分类器越重要
  - 系数决定了该弱分类器在最终组合中的权重

4）权值分布更新 01:25:57

- 更新规则：$w_{m+1,i}=\frac{w_{mi}}{Z_m}\exp(-\alpha_m y_i G_m(x_i))$
- 规范化因子：$Z_m=\sum_{i=1}^{N}w_{mi}\exp(-\alpha_m y_i G_m(x_i))$，保证新权值构成概率分布
- **权重变化规律**：
  - 正确分类样本：权重乘以$e^{-\alpha_m}$（减小）
  - 错误分类样本：权重乘以e^{α_m}（增大）
- 数学解**释**：通过指数损失函数推导，体现了"重视错误样本"的思想

5）线性组合构建 01:31:36

●



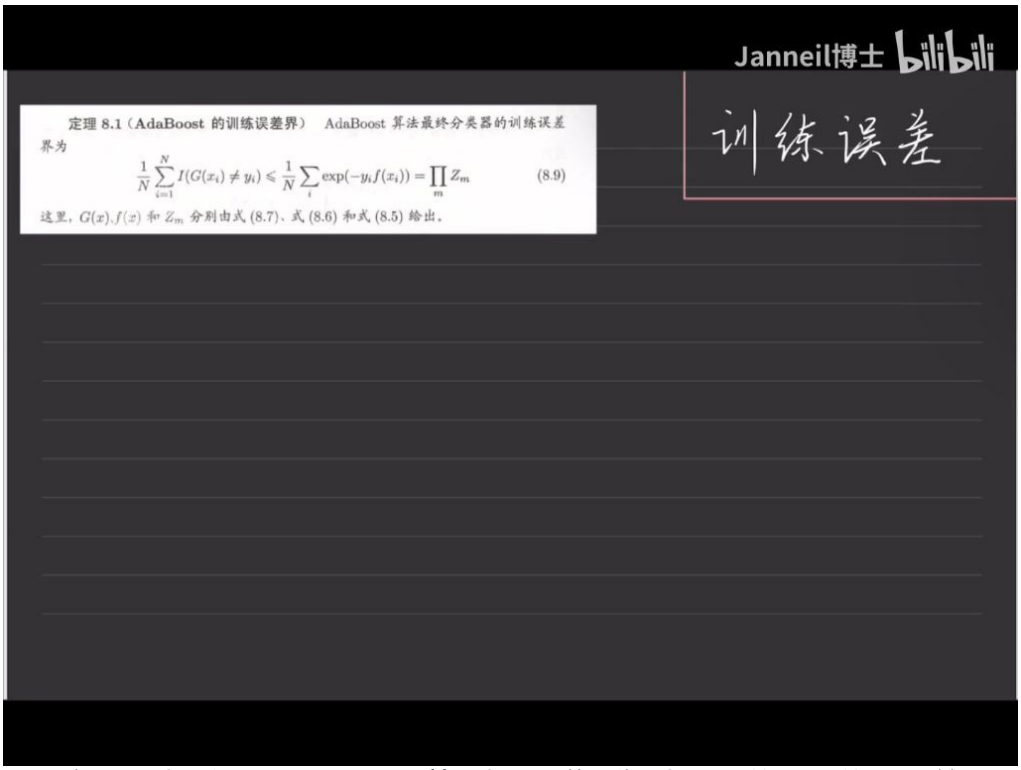
- **组合方式**：$f(x)=\sum_{m=1}^{M}\alpha_m G_m(x)$，加权求和所有弱分类器
- **最终分类器**：$G(x)=sign(f(x))$，通过符号函数输出类别
- **特点**：
  - 弱分类器系数α_m不要求归一化
  - 组合后的强分类器具有更好的泛化能力
  - 通过迭代可以不断提升分类性能

2. 训练误差界定理 01:32:22

1）定理 8.1 介绍

- 定理 8.1 概述 01:32:30

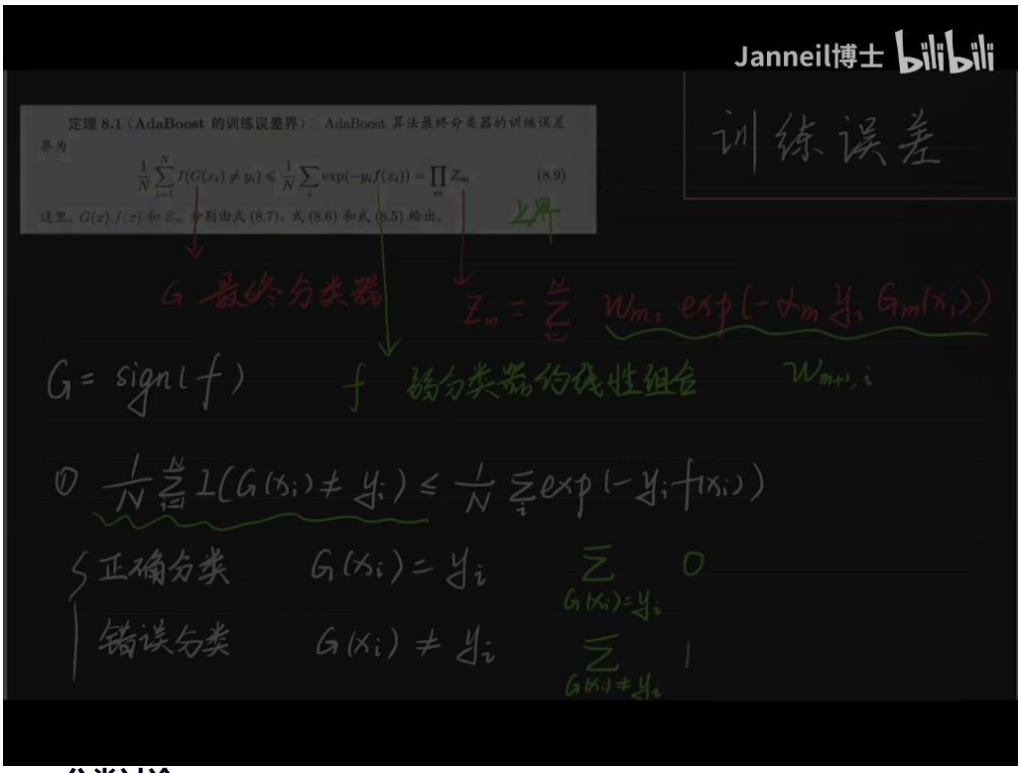
○



- 定理内容：AdaBoost 算法最终分类器的训练误差界为 $\frac{1}{N}\sum_{i=1}^{N} I(G(x_i)\neq y_i)\leq \frac{1}{N}\sum_i \exp(-y_i f(x_i))=\prod_m Z_m$
- **符号说明**：
  - G(x)：最终分类器，由式(8.7)给出
  - f(x)：弱分类器的线性组合，由式(8.6)给出
  - Z_m：规范化因子，由式(8.5)给出

- 定理 8.1 不等式左边解读 01:33:05
  - **训练误差定义**：$\frac{1}{N}\sum_{i=1}^{N} I(G(x_i)\neq y_i)$表示分类错误样本的比例
    - $I(\cdot)$为示性函数，分类错误时取 1，正确时取 0
    - 求和后除以 N 得到错误率
  - **分类器关系**：$G(x)=sign(f(x))$，即最终分类器是弱分类器线性组合的符号函数
- 定理 8.1 不等式右边解读 01:34:31
  - **指数损失**：$\frac{1}{N}\sum_i \exp(-y_i f(x_i))$是训练集上的指数损失
  - 规范化因子：$\prod_m Z_m$是各轮规范化因子的连乘积
    - Z_m保证每轮权重更新后仍为概率分布
    - $Z_m=\sum_i \omega_{m,i}\exp(-\alpha_m y_i G_m(x_i))$
- 定理 8.1 不等式等号解读 01:35:47
  - 等号条件：当且仅当所有误分类点满足$f(x_i)=0$时取等号
    - 即所有误分类点都位于决策边界上
    - 实际应用中极少出现这种情况
- 定理 8.1 不等式成立过程 01:37:28

○



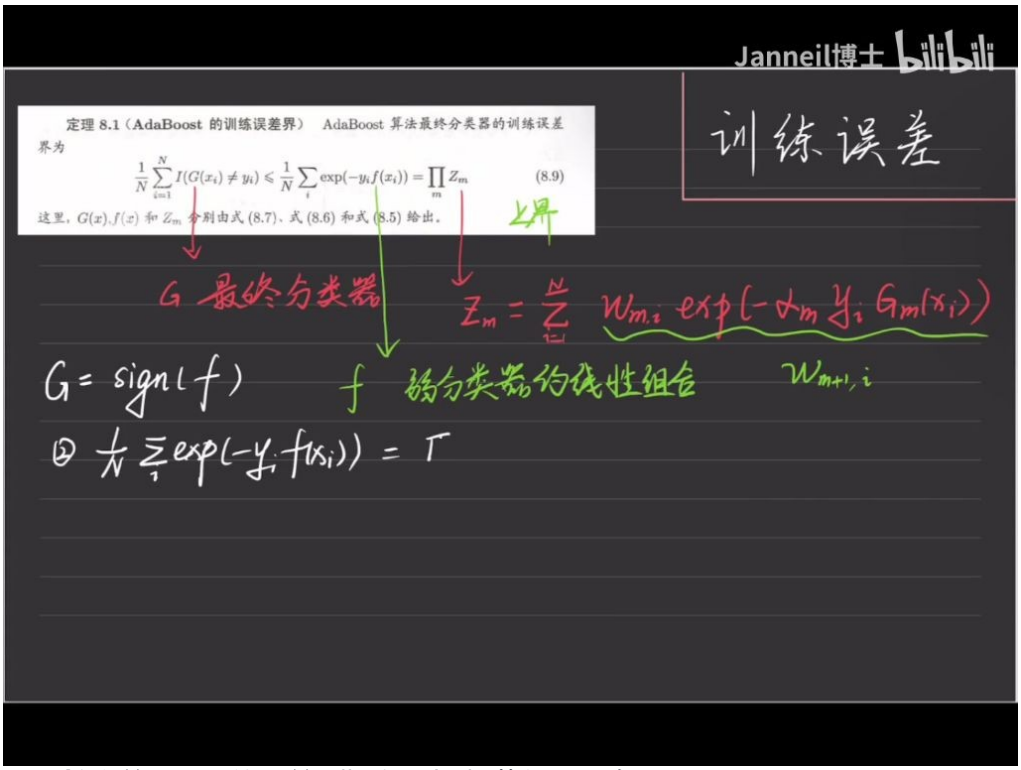
- **分类讨论**：
  - 正确分类点：$y_i f(x_i) \geq 0$，指数损失≤ 1，示性函数=0
  - 错误分类点：$y_i f(x_i) \leq 0$，指数损失≥ 1，示性函数=1
- **结论**：示性函数值总是小于等于对应的指数损失值

- 定理 8.1 不等式取等号情况 01:46:43
  - **严格条件**：需要同时满足：
    - 所有样本点都被误分类
    - 所有误分类点都满足$f(x_i)=0$
  - **实际意义**：表明定理给出的上界在极端情况下可以达到

2）定理 8.1 证明 01:48:11

- 定理 8.1 内容：AdaBoost 的训练误差界

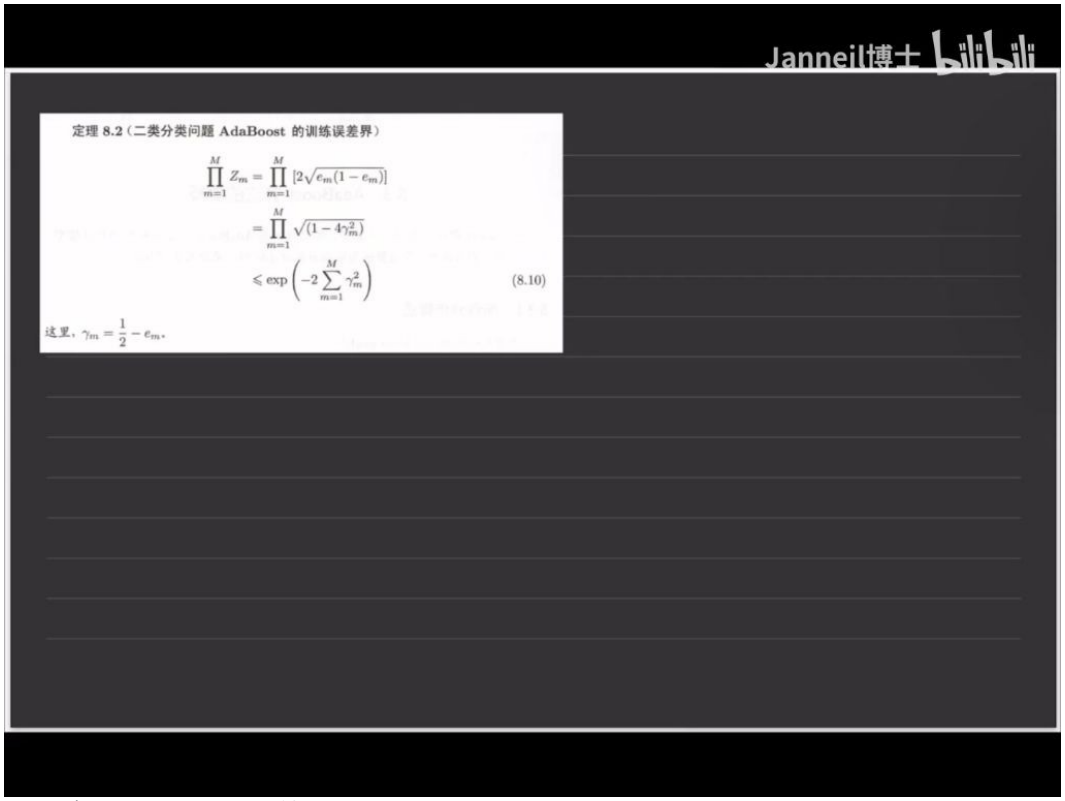
○



- **核心关系**：训练误差≤指数损失=规范化因子连乘积

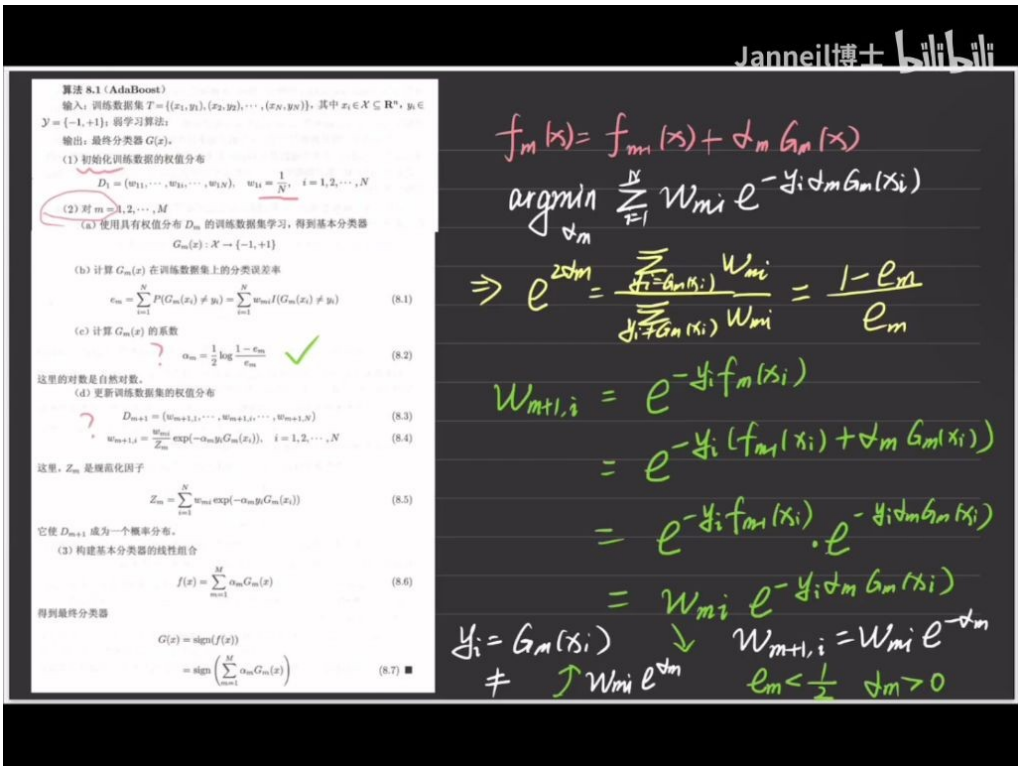
- 训练误差上界的展开 01:48:45
  - **线性组合展开**：将$f(x)=\sum_{m=1}^{M}\alpha_m G_m(x)$代入指数损失
  - **权重表示**：初始权重$\omega_{1,i}=\frac{1}{N}$，可用其表示第一项
  - **递推关系**：$\omega_{m+1,i}=\frac{\omega_{m,i}\exp(-\alpha_m y_i G_m(x_i))}{Z_m}$
- 权重更新与规范化因子关系 01:53:50
  - **权重递推**：
    - $Z_1\omega_{2,i}=\omega_{1,i}\exp(-\alpha_1 y_i G_1(x_i))$
    - 依次展开可得$Z_1 Z_2 \cdots Z_{M-1}\omega_{M,i}=\omega_{1,i}\exp(-\sum_{m=1}^{M-1}\alpha_m y_i G_m(x_i))$
  - **最终形式**：通过递推将指数损失表示为$\prod_m Z_m$
- 定理 8.1 的用处 01:59:10
  - **算法指导**：通过最小化Z_m来降低训练误差上界
    - 每轮选择使Z_m最小的弱分类器G_m
    - Z_m越小表示该轮分类效果越好
  - 理**论保证**：解释了为什么 AdaBoost 能有效降低训练误差

3）定理 8.2 介绍 02:01:41

- 

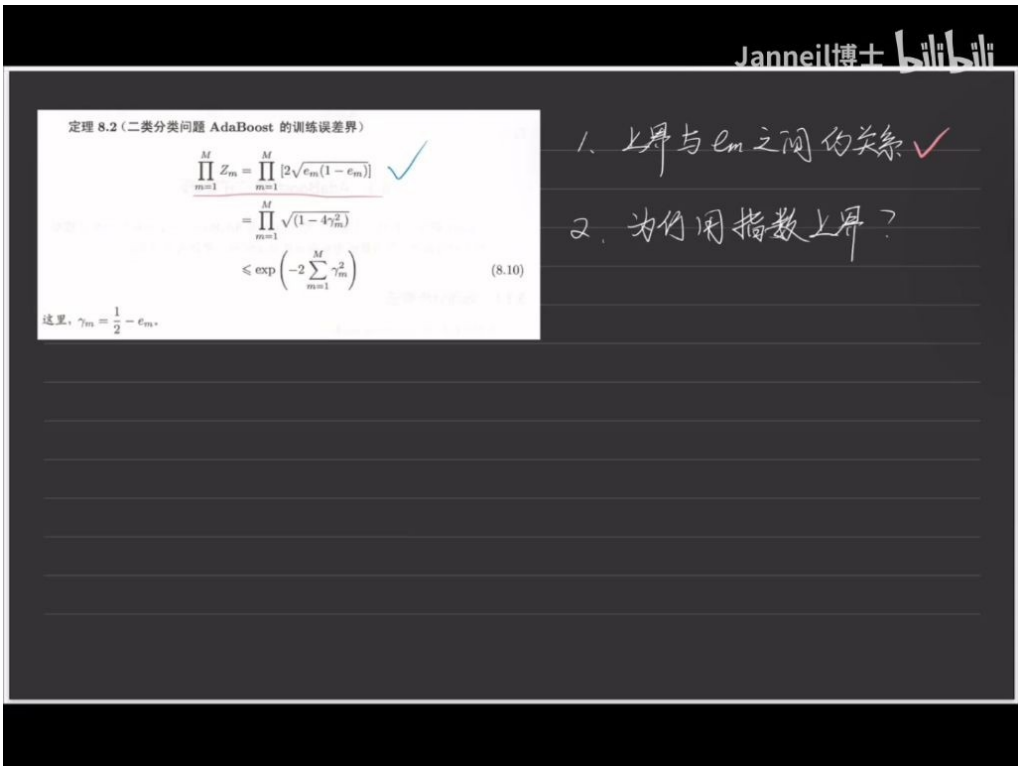
- 定理 8.2 内容：训练误差上界的上界 02:01:53
  - **核心不等式**：$\prod_{m=1}^{M} Z_m = \prod_{m=1}^{M}$¿
  - **参数定义**：$\gamma_m = \frac{1}{2} - e_m$，其中$e_m$为第 m 个弱分类器的错误率
- 定理 8.2 相关问题 02:02:16
  - **关键问题 1**：训练误差上界与错误率e_m的关系
  - **关键问题 2**：为何最终采用指数形式表示上界
- 定理 8.2 第一行推导：上界与 em 之的关系 02:02:58
  - 规范化因子分解：Z_m可拆分为正确分类部分$\sum_{y_i = G_m(x_i)} w_{mi} e^{-\alpha_m}$和错误分类部分$\sum_{y_i \neq G_m(x_i)} w_{mi} e^{\alpha_m}$
  - **错误率表示**：$e_m = \sum_{y_i \neq G_m(x_i)} w_{mi}$，正确分类权重和为$1 - e_m$
  - 等号条件：当且仅当$(1-e_m)e^{-\alpha_m}$¿$e_m e^{\alpha_m}$时成立，此时$\alpha_m = \frac{1}{2}\ln\frac{1-e_m}{e_m}$
- 算法 8.1 回顾：AdaBoost 02:10:13

-



  - **权重更新**：$w_{m+1,i}=\frac{w_{mi}}{Z_m}\exp(-\alpha_m y_i G_m(x_i))$
  - **分类器组合**：$f(x)=\sum_{m=1}^{M}\alpha_m G_m(x)$，最终分类器$G(x)=sign(f(x))$
- 定理 8.2 第一行证明结论 02:11:01
  - **关键发现**：算法中α_m的表达式与推导结果完全一致，验证了等号成立的合理性
  - 数学关系：$\alpha_m=\frac{1}{2}\ln\frac{1-e_m}{e_m}$是使不等式取等的必要条件
- 定理 8.2 第二行推导：为何用指数上界 02:11:17
  - 代数**变**形：$2\sqrt{\square}$，其中$\gamma_m=\frac{1}{2}-e_m$
  - **弱分类器性质**：要求$e_m<\frac{1}{2}$（优于随机猜测），故$\gamma_m>0$
  - **指数化优势**：便于分析误差界的收敛速度，$\exp(-2\sum\gamma_m^2)$直观显示误差随迭代次数指数下降

○

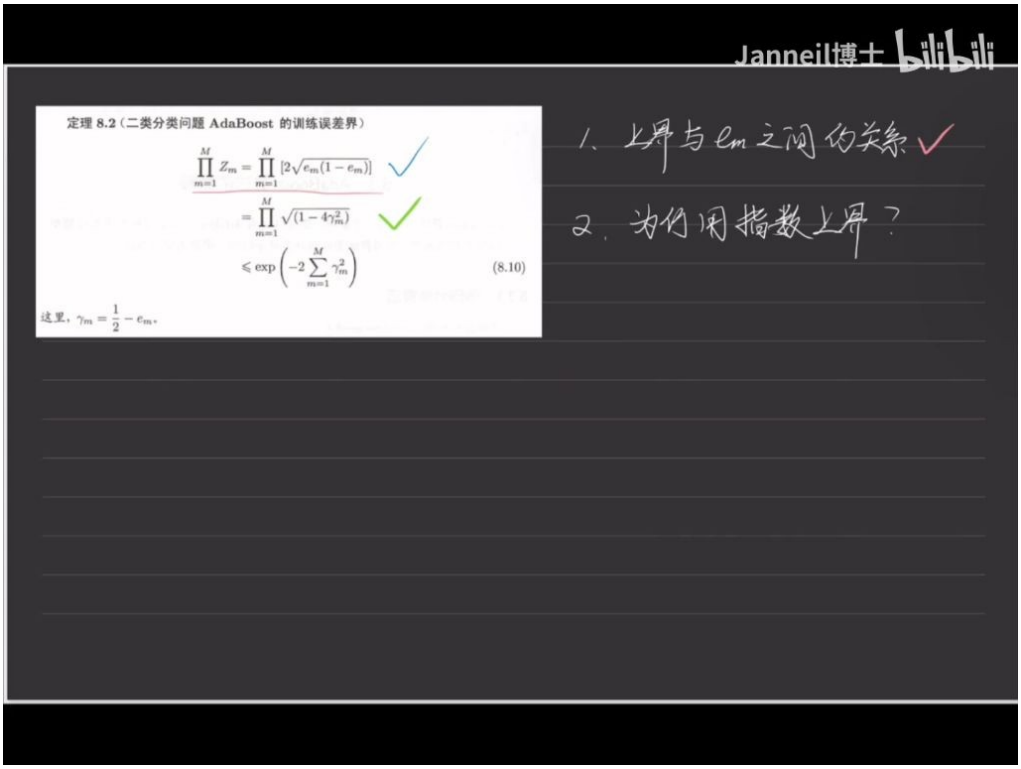


- **实际意义**：当每个弱分类器仅比随机猜测略好（γ_m很小），需要更多迭代才能使误差足够小；若存在显著优势的弱分类器（γ_m较大），误差会快速下降

4）定理 8.2 证明 02:13:30

- 定理 8.2 的证明目标 02:13:48

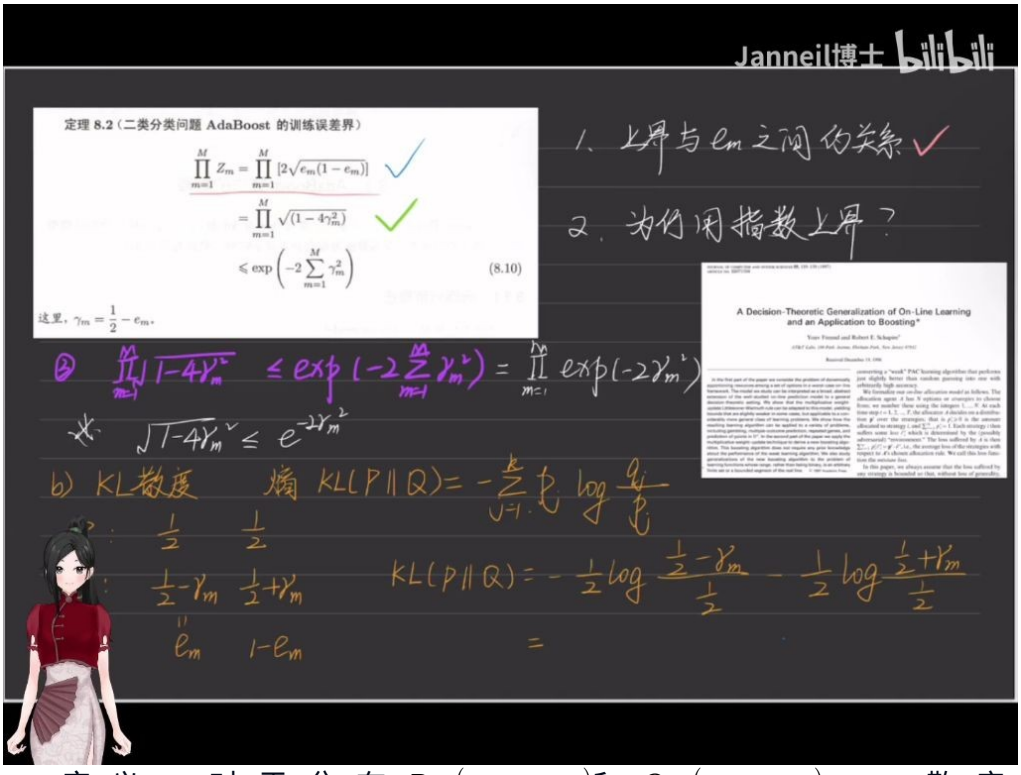
○



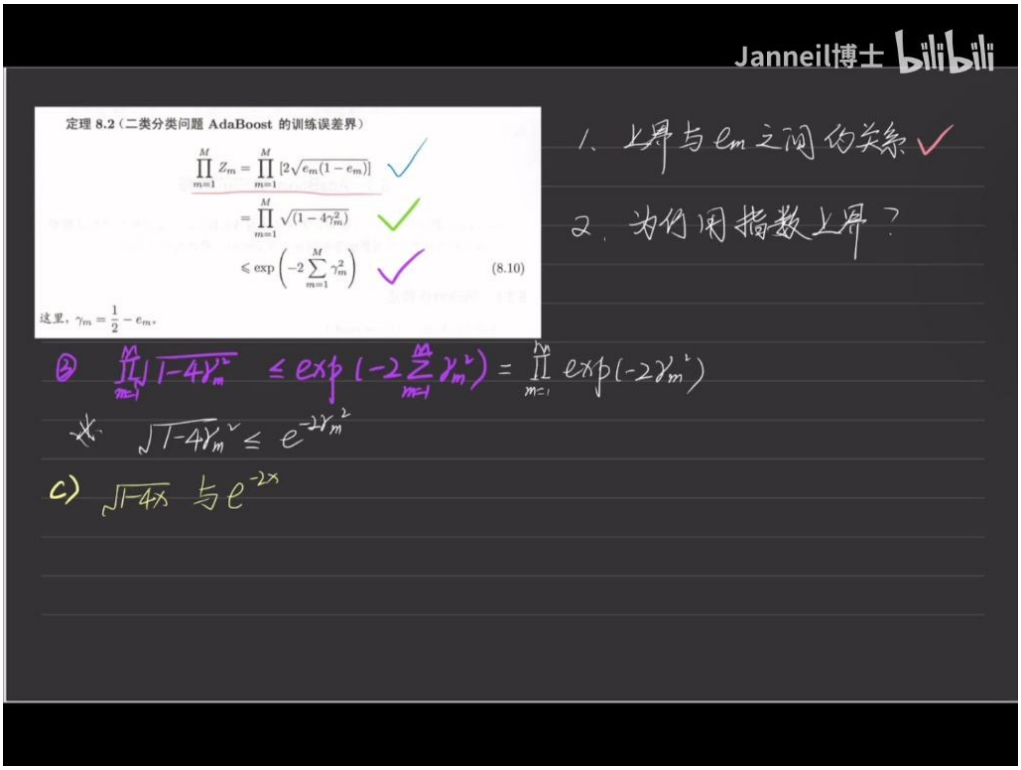
- **核心不等式**：需要证明$\prod_{m=1}^{M}\sqrt{\square}$，其中$\gamma_m = \frac{1}{2} - e_m$
- **转化目标**：只需证明$\sqrt{\square}$（令$x = \gamma_m^2$）

- **物理意义**：当弱分类器仅略优于随机猜测（γ_m接近 0）时，AdaBoost 的训练误差上界

- 泰勒展开证明 02:15:20
  - 函数展开
    - **展开函数**：在$x=0$处对$f(x)=\sqrt{\square}$和$g(x)¿e^{-2x}$进行泰勒展开
    - **导数计算**：
      - f(x)前三阶导：$f'(0)=-2, f''(0)=-4, f'''(0)=-24$
      - g(x)前三阶导：$g'(0)=-2, g''(0)=4, g'''(0)=-8$
  - 展开式比较
    - **f(x)展开**：$1-2x-2x^2-4x^3+O(x^4)$（各项系数均为负）
    - **g(x)展开**：$1-2x+2x^2-\frac{4}{3}x^3+O(x^4)$（系数正负交替）
    - **关键结论**：由于 f(x)展开后各项均为减项，而 g(x)有增有减，故$f(x)\le g(x)$在$x\ge 0$时成立
- KL 散度介绍 02:22:10
  -



  - 定义：对于分布$P=(p_1,\dots,p_k)$和$Q=(q_1,\dots,q_k)$，KL 散度为$D_{KL}(P\vee ¿Q)=-\sum p_i\log\frac{q_i}{p_i}$
  - **物理意义**：用 P 的信息熵度量 Q 分布相对于 P 分布的偏离程度
  - **特性**：非对称性（$D_{KL}(P\vee ¿Q)\neq D_{KL}(Q\vee ¿P)$），非负性
- KL 散度在定理 8.2 证明中的应用 02:25:09
  - 分布设定
    - **基准分布 P**：随机猜测的分布$(\frac{1}{2},\frac{1}{2})$
    - **比较分布 Q**：弱分类器分布$(\frac{1}{2}-\gamma_m,\frac{1}{2}+\gamma_m)$
  - 不等式转化

    - **KL 散度计算**：$D_{KL}(Q\vee ¿P)=\frac{-1}{2}\log(1-4\gamma_m^2)$
    - **关键步骤**：
      - 将√❑表示为$e^{-D_{KL}}$
      - 证明$-D_{KL}\leq -2\gamma_m^2$（通过对log √❑泰勒展开）
      - 最终得到√❑
  - 方法对比
    - 泰勒展开法：直接但计算量大，需要展开多阶导数
    - **KL 散度法**：概念更深刻，建立了概率分布与误差界的联系
    - **共同点**：都需要在γ_m接近 0（弱分类器）的假设下成立
- 为什么用指数上界 02:31:10
  -


  - 定理表达式：$\prod_{m=1}^{M} Z_m = \prod_{m=1}^{M} ¿$
  - **不等式验证**：通过图形比较√❑（绿色曲线）与e^{-2x}（蓝色曲线）在$x\in[0,0.25]$区间的关系，验证绿色曲线始终低于蓝色曲线
  - **中间曲线分析**：存在更精确的上界$1-\log(1+2x)$（红色曲线），但会导致连乘形式复杂化
  - **核心原因**：利用指数函数性质将连乘∏化求和∑，简化表达式形式
- 指数函数的选择 02:35:40
  -
  - **参数确定条件**：
    - 在$x=0$处与√❑相切（一阶导数相等）
    - 全局满足$e^{f(x)}\geq$ √❑
  - **推导过程**：
    - 设指数函数为e^{ax^b}，求导得$abe^{ax^b}x^{b-1}$
    - 在$x=0$处必须满足$ab=-2$（因√❑在$x=0$处导数为-2）
    - 取$b=1$保证非零导数，从而确定$a=-2$
  - 函数**验证**：通过动态图示展示当$a=-2$时，e^{-2x}与√❑在$x=0$处相切且全局上界成立
  - **最优性说明**：当a←2时函数会与√❑相交，不再满足全局上界条件

- 。
  - **关键参数**：$\gamma_m = \frac{1}{2} - e_m$，反映基分类器与随机猜测的差距
  - **误差界意义**：当γ_m有下界时，训练误差随 M 增大呈指数下降

5）推论 8.1 介绍 02:39:21

- 。
- 定理 8.1：AdaBoost 算法最终分类器的训练误差界 02:39:24
  - **误差界表达式**：$\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leqslant \frac{1}{N}\sum_i \exp(-y_i f(x_i)) = \prod_m Z_m$
  - **组成要素**：式中 G(x)、f(z)和Z_m分别由式(8.7)、(8.6)和(8.5)给出
  - **核心意义**：该定理给出了 AdaBoost 最终分类器训练误差的上界表达式
- 定理 8.2：二类分类问题 AdaBoost 的训误差界 02:39:40
  - **乘积形式表达式**：$\prod_{m=1}^{M} Z_m = \prod_{m=1}^{M}$¿
  - **关键参数**：其中$\gamma_m = \frac{1}{2} - e_m$
  - **指数形式表示**：该定理表明训练误差上界可以用指数形式表示，说明误差以指数速率下降
- 推论 8.1 的证明与结论 02:40:00
  - **前提条件**：存在$\gamma > 0$，对所有 m 有$\gamma_m \geqslant \gamma$
  - **误差界推导**：$\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leqslant \exp(-2M\gamma^2)$
  - **证明过程**：当所有$\gamma_m \geqslant \gamma$时，$\sum_{m=1}^{M} \gamma_m^2 \geqslant M\gamma^2$，代入定理 8.2 即得
  - **实际意义**：训练误差上界随迭代次数 M 呈指数下降，但无需精确计算γ的具体值
- AdaBoost 算法名称的来历 02:42:00
  - 名称含义：Adaptive Boosting 的缩写
  - **适应性体现**：每个基分类器的训练误差率γ_m是独立出现的，具有适应性
  - **统计背景**："adaptive"在统计中常用于表示方法的适应性特征

3. 前向分步算法 02:42:44

1）逐步回归思想

- 。
- **核心思想**：通过逐步添加或剔除变量的方式构建最优回归模型，类似于超级玛丽逐步攀爬的过程。
- **变量选择问题**：当存在 p 个自变量$x_1, x_2, \ldots, x_p$时，直接全部纳入模型会导致模型过于臃肿。
- **系数比较误区**：不能通过比较回归系数β的大小来判断变量重要性，因为系数值会随变量单位变化而改变（如身高单位从厘米变为米时系数会放大 100 倍）。
- **逐步方式分类**：包含向前（Forward）和向后（Backward）两种基本方法，以及结合二者的逐步回归（Stepwise Regression）。

2）逐步向前回归 02:47:43

- **零模型起点**：从不含任何自变量的模型$y = \beta_0 + \epsilon$开始，用因变量均值作为初始预测。
- **逐步添加原则**：
  - 每次添加一个使模型改善最大的变量
  - 通过准则比较（AIC/BIC 越小越好，F 检验显著性）
  - 示例：第一轮从 p 个变量中选出最优x_2，第二轮从剩余$p-1$个中继续筛选
- **终止条件**：当添加新变量不能显著改善模型时停止
- **模型表示**：
  - 第 k 步模型：$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \epsilon$
  - 参数估计：$\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k$

3）逐步向后回归 02:53:50

- 。
- **全模型起点**：从包含所有 p 个自变量的完整模型$y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \epsilon$开始

- **逐步剔除原则**：
  - 每次剔除对模型影响最小的变量
  - 使用与向前回归相同的评价准则
  - 示例：首轮可能剔除x_1，得到$p-1$个变量的模型
- **终止条件**：当继续剔除变量会导致模型显著变差时停止
- **谐音记忆**："剔除"与"T（踢）出"谐音，帮助记忆变量被淘汰的过程
- **逐步回归综合**：
  - 结合向前和向后方法，每添加一个变量后检查是否需要剔除已有变量
  - 类比《论语》"学而时习之"和"吾日三省吾身"的反复检验思想
  - 优势：能发现变量间的交互作用和冗余情况

五、可加模型 ***02:58:34***

1. 逐步回归基础

- **建模思想**：延续逐步回归的前向分步思想，通过逐步添加基函数构建模型
- **翻译差异**：统计学习方法中译作"加法模型"，但为体现逐步添加过程，更推荐"可加模型"译法

2. 可加模型定义 02:58:47

-
- 基本形式：$f(x)=\sum_{m=1}^{M}\beta_m b(x;\gamma_m)$
- **核心特点**：由多个基函数$b(x;\gamma_m)$加权求和构成，每个基函数包含参数γ_m和系数β_m

3. 线性回归对比 02:59:20

- **结构差异**：
  - 性回：明确性关系$y=\beta_0+\beta_1 x_1+...+\epsilon$
  - 可加模型：允许未知非线性关系，基函数形式更自由
- **参数特性**：可加模型的基函数参数γ_m和系数β_m都需要估计

4. 模型结构分析 02:59:56

- 函数**组**成：每个基函数$b(x;\gamma_m)$是与自变量 x 和参数γ_m相关的未知函数
- **扩展性**：可包含任意数量（M 个）基函数，类似主成分分析但不受线性限制

5. 基函数概念 03:00:49

- **多样性**：可以是任意光滑函数（回归问题）或决策树（分类问题）
- 典型应用：AdaBoost 中基函数为简单分类器，参数γ_m对应"分割阈值"

6. 模型参数估计 03:02:04

- 估方法：通过最小化损失函数$L(y,f(x))$确定参数
- **损失计算**：需计算所有样本点的平均损失$\frac{1}{N}\sum_{i=1}^{N}L(y_i,f(x_i))$

7. 前向分布算法 03:02:47

- 迭代过程：
  - 初始化$f_0(x)=0$
  - 逐步添加基函数：$f_m(x)¿f_{m-1}(x)+\beta_m b(x;\gamma_m)$
  - 通过损失最小化确定β_m,γ_m
- **终止条件**：当损失满足要求或达到预设的 M 个基函数

六、前向分步算法 ***03:09:10***

1. 算法输入输出

-
- **输入要素**：
  - 训练数据$T=\{(x_1,y_1),...,(x_N,y_N)\}$
  - 损失函数$L(y,f(x))$
  - 基函数集$\{b(x;\gamma)\}$
- **输出结果**：加法模型 f(x)

2. 初始化步骤 03:09:34

- 零模型：初始设定$f_0(x)=0$（分类问题）或常数（回归问题）

3. 迭代过程 03:10:00

- **参数优化**：每步求解
- **模型更新**：$f_m(x) ¿ f_{m-1}(x)+\beta_m b(x;\gamma_m)$

4. AdaBoost 算法 03:11:57
- **特殊形式**：当损失函数为指数损失$L(y,f(x)) ¿ e^{-yf(x)}$时
- **参数计算**：
  - 基函数$G_m(x)$通过最小化分类误差率得到
  - 系数$\alpha_m=\frac{1}{2}\ln\frac{1-e_m}{e_m}$，其中$e_m$为分类误差率
- **权重更新**：样本权重$w_i^{(m)}$随分类效果动态调整

七、提升 ***03:22:56***

1. 提升方法概述
- 本质：提升方法属于集成方法的一种，由若干个基学习器组成
- **名称解析**："提升"二字体现方法本质，"树"字说明基学习器结构

2. 决策树分类回归 03:23:30
-
- 基学**习**器**结**构：采用二叉树（仅有两个分叉的简单结构）
- 应用场景：
  - 分类问题：如判断水蜜桃"好吃"（+1）或"不好吃"（-1）
  - 回归问题：如量化水蜜桃好吃程度（5.5-9.7 连续值）

3. 众数均值思想 03:24:34
- **分类问题标签确定**：
  - 采用众数思想（多数表决）
  - 示例：甜度>0.1 的 4 个样本中 3 个标记为+1，故该分支标签取+1
- 回归问题标签确定：
  - 采用均值思想（算术平均）
  - 示例：甜度>0.1 的 4 个样本值（7.6,9.5,9.7,8.2）均值为 8.75

4. 中位数应用 03:26:40
- **适用场景**：
  - 顺序数据（如问卷满意度等级）
  - 数值型数据（回归问题）
- 未采用原因：
  - 计算需要排序，效率低于众数/均值
  - 异常值可通过分叉自然隔离
  - 当前方法已足够有效

5. 二叉树结构 03:29:12
- 基学**习**器特点：
  - 极简结构（"木墩子"式二叉划分）
  - 分类回归通用，仅标签确定方式不同
- **分叉示例**：
  - 甜度阈值 0.1：<0.1 左分支，≥0.1 右分支

6. 可加模型实现 03:29:53
- **最终模型**：$f(x)=\sum_{m=1}^{M}T(x;\theta_m)$
  - T 表示决策树基学习器
  - θ_m为第 m 棵树的参数
  - M 为基学习器总数

7. 前向分布算法应用 03:30:27

1）决策树分类与回归确定标签的思想 03:30:28
- **核心区别**：
  - 分类：离散标签→众数决策
  - 回归：连续数值→均值决策

2）集中趋势的度量：众数、平均数、中位数 03:31:35
- **统计量对比**：

- 众数：出现频率最高值
- 平均数：算术平均值
- 中位数：有序序列中间值

3）为什么不用中位数确定标签 03:34:09

- **实践考量**：
  - 计算复杂度较高（需排序）
  - 异常值处理可通过分叉实现
  - 众数/均值已满足大多数场景需求

4）提升树模型结构：基学习器与可加模型 03:36:05

-
- **组成要素**：
  - 基学习器：二叉树决策树
  - 组合方式：线性可加模型

5）提升树算法要素 03:38:52

- **输入**：
  - 训练数据集$T=\{(x_1,y_1),\ldots,(x_N,y_N)\}$
  - 损失函数$L(y,f(x))$：
    - 分类：指数损失
    - 回归：平方损失
  - 基函数集：决策树集合$\{T(x;\theta)\}$

6）提升树算法步骤 03:40:33

- **初始化**：$f_0(x)=0$
- 迭代更新：
  - $f_m(x)¿f_{m-1}(x)+T(x;\theta_m)$
  -
- **输出**：$f_M(x)=\sum_{m=1}^{M}T(x;\theta_m)$

八、*回提升* ***03:43:25***

1. 例题:训练数据表

- 数据范围: x 取值范围为[0.5,10.5]，y 取值范围为[5.0,10.0]
- 数据分布: 包含 10 个样本点，x 取整数值 1-10，对应 y 值分别为 5.56,5.70,...,9.05

2. 树桩基函数 03:44:19

- 定义: 使用二叉树作为基函数组成的回归器
- **特点**: 每个树桩仅包含一个切分点和两个叶节点
- 应用: 本例中采用简单二叉树结构，通过逐步优化提升模型性能

3. 切分点选择 03:45:35

-
- **候选点**: 在 x=1.5,2.5,...,9.5 共 9 个位置尝试切分
- 划分方法: 采用"一刀切"方式，将数据分为左右两部分
- **计算原理**: 左区域取第一个样本 y 值，右区域取剩余样本 y 值的平均值

4. 平方损失计算 03:47:02

-
- **计算公式**:$L=\sum(y_i-\hat{y}_i)^2$
- 示例**计**算: 当 s=1.5 时，$\hat{y}_1=5.56$，其余$\hat{y}_i=7.50$，平方损失为 15.7231
- **比较方法**: 对 9 种切分点分别计算平方损失，选择最小值对应的切分点

5. 最优切分确定 03:48:48

- **最优标准**: 平方损失最小化原则
- 本例结果: 当 s=6.5 时获得最小平方损失 1.93
- **对应划分**:
  - x<6.5 时预测值 6.24（前 6 个样本均值）
  - x≥6.5 时预测值 8.91（后 4 个样本均值）

6. 残差拟合过程 03:50:54

-

- 残差定义:$r_i ¿ y_i - \hat{y}_i$
- **第二轮拟合**: 使用第一轮的残差作为新目标变量
- **最优切分**: 第二轮在 s=3.5 处获得最小平方损失 0.79
- **组合模型**:
  - x<3.5: 5.72 (6.24-0.52)
  - 3.5≤x<6.5: 6.46 (6.24+0.22)
  - x≥6.5: 9.13 (8.91+0.22)

7. 多轮迭代效果 03:53:24
-
- **损失变化**:
  - 第 1 轮: 1.93
  - 第 2 轮: 0.79
  - 第 3 轮: 0.47
  - 第 4 轮: 0.30
  - 第 6 轮: 0.17
- **停止条件**: 当损失小于预设阈值(如 0.2)时终止迭代

8. 过拟合问题 03:57:29
- 提示: 过度细分会导致完美拟合训练数据但泛化能力下降
- 极端情况: 若使用 9 个切分点将数据分为 10 段，可实现零误差但必然过拟合
- **实践建议**: 需要合理设置误差阈值，平衡模型精度与泛化能力

九、梯度提升算法 ***03:59:28***

1. 算法概述
-
- **输入输出**：训练数据集$T=(x_1,y_1),(x_2,y_2),\dots,(x_N,y_N)$，输出提升树$f_M(x)$
- **核心思想**：通过迭代拟合残差来逐步提升模型精度，每轮训练一个新的回归树来拟合当前残差
- **流程特点**：与之前回归问题不同，这里每次迭代都要更新训练数据为上一轮的残差

2. 初始化改进 03:59:53
- **传统初始化**：直接令$f_0(x)=0$
- 梯度提升改：
  - 回归问题：可取$f_0(x)=\bar{y}$（y 的平均值）
  - 分类问题：可取类别众数作为初始预测
- 数学表达：，对平方损失即取均值

3. 残差计算 04:00:27
-
- **计算公式**：$r_{mi}=y_i-f_{m-1}(x_i)$
- **实际意义**：第 m 轮时，用当前模型$f_{m-1}(x)$的预测误差作为新的学习目标
- **计算示例**：第一轮残差=真实值-初始预测值（6.24），第二轮拟合第一轮残差

4. 梯度下降应用 04:01:38
-
- **梯度联系**：当使用平方损失$L=\frac{1}{2}(y-f(x))^2$时，负梯度正好等于残差
- 数学推导：$\frac{\partial L}{\partial f}=-(y-f(x))$，故$-[\frac{\partial L}{\partial f}]_{f=f_{m-1}}=y-f_{m-1}(x)¿r_m$
- **通用性**：梯度提升可适用于任意可微损失函数，不限于平方损失

5. 回归树拟合 04:02:27
-
- 切分点：通过最小化损失函数确定，例中第一轮选 6.5，第二轮选 3.5
- **区域划分**：得到第 m 棵树的叶结点区域$R_{mj},j=1,2,\dots,J$
- **参数计算**：

6. 参数更新 04:04:09
-

- 更新公式：$f_m(x)¿f_{m-1}(x)+\sum_{j=1}^{J} c_{mj} I(x \in R_{mj})$
- **实际应用**：例如 x=6 时，先通过初始预测 6.24，加上第一轮的 0.22 得到 6.46
- **指示函数**：$I(x \in R_{mj})$判断 x 属于哪个区域，决定加上哪个c_{mj}

7. 牛法比 04:05:42

- **梯度下降**：仅使用一阶导数信息
- 牛法：可利用二阶导数信息，可能获得更快收敛
- 应用场景：当损失函数复杂（如含正则项）时，梯度提升更具普适性
- **文献建议**：对高阶优化方法感兴趣可查阅相关文献

十、知小

知识点	核心内容	关键公式/算法	应用示例	难度系数
提升方法基本概念	通过组合多个弱学习器构建强学习器	G(x)=sign(∑αₘgₘ(x))	分类问题中三个臭皮匠顶诸葛亮	⭐⭐
AdaBoost 算法流程	1. 初始化权重 2. 迭代训练弱分类器 3. 更新样本权重 4. 加权组合	αₘ=1/2 ln((1-eₘ)/eₘ) wₘⁱ=wₘ₋₁ⁱ exp(-αₘyᵢgₘ(xᵢ))	一刀切分类器示例（切分点 2.5/8.5/5.5）	⭐⭐⭐⭐
PAC 学习框架	概率近似正确学习理论	Pr[Pr[h(x)≠c(x)]≤ε]≥1-δ	强可学习⇔弱可学习	⭐⭐⭐⭐
前向分步算法	逐步添加基函数优化模型	fₘ(x)=fₘ₋₁(x)+βₘb(x;γₘ)	回归问题中逐步添加决策树	⭐⭐⭐
提升树模型	以决策基函数的提升方	分类树：众数标签 回归树：均值标签	水蜜桃甜度分类案例	⭐⭐⭐
梯度提升(GB)	用梯度下降思想优化任意损失函数	rₘⁱ=-[∂L(yᵢ,f(xᵢ))/∂f(xᵢ)]	平方损失时等价于残差拟合	⭐⭐⭐⭐
训练误差上界	指数级下降的误差边界	∏ₘ²√(eₘ(1-eₘ))≤exp(-2∑γₘ²)	定理 8.1-8.2 证明过程	⭐⭐⭐⭐⭐