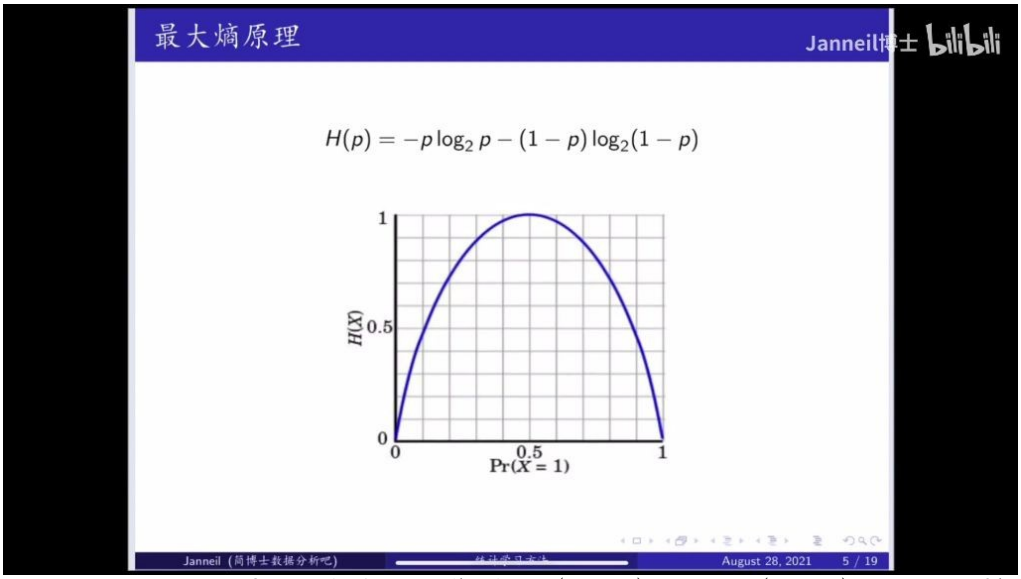
一、最大熵模型 00:01

1. 最大熵原理 00:09

1）熵的定义 00:31

- 起源：熵起源于热力物理学，后被引入信息论中用于度量信息量的多少。
- 离散形式：对于离散随机变量 X，熵定义为$H=-\sum p(x)logp(x)$
- 形式：对于连续随机变量，求和变为积分，即$H=-\int p(x)logp(x)dx$
- **最大熵含义**：指包含最多信息量的概率分布情况。

2）离散情况最大熵 02:20

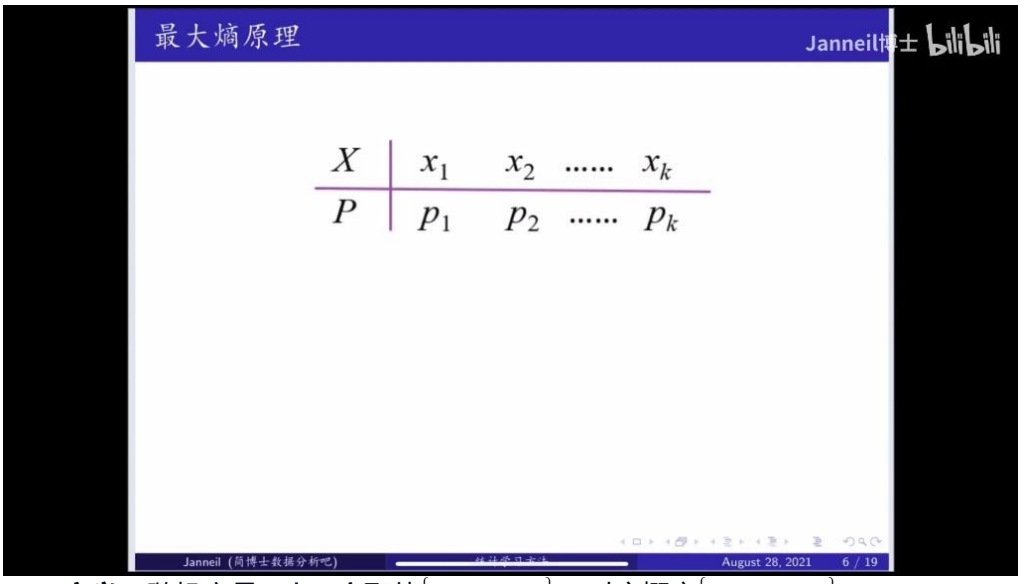
- 两点分布示例
  - 

  - 表达式：对于伯努利分布$P(X=1)=p$，$P(X=0)=1-p$，熵为$H(p)=-p\log_2 p-(1-p)\log_2(1-p)$
  - **极值分析**：
    - 当$p=0$或$p=1$时，熵达到最小值 0
    - 当$p=0.5$时，熵达到最大值 1（即$\log_2 2$）
  - **物理意义**：等概率分布时信息量最大，确定性事件（$p=0$或 1）信息量为零

3）多项分布最大熵 05:14

- 多项分布的基本概念 05:19

-

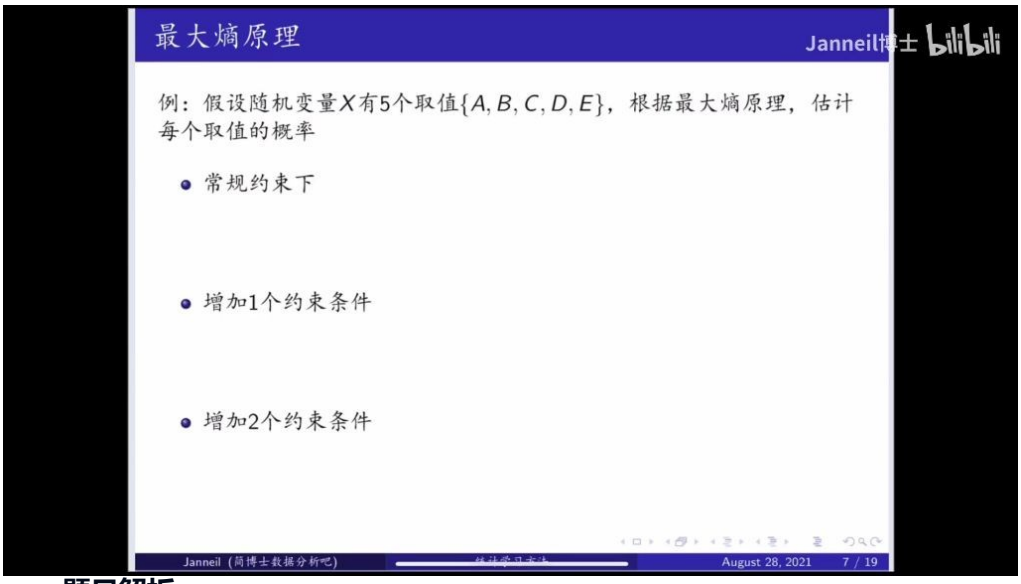


  - **定义**：随机变量 X 有 k 个取值$\{x_1,\dots,x_k\}$，对应概率$\{p_1,\dots,p_k\}$
- 多项分布的熵 05:59
  - 表达式：$H=-\sum_{i=1}^{k} p_i \log p_i$
  - **约束条件**：$\sum_{i=1}^{k} p_i=1$
- 限制条件与拉格朗日乘子法 06:21
  - **优化问题**：在$\sum p_i=1$约束下最大化 H
  - 拉格朗日函数：$L=-\sum p_i \log p_i+\lambda\left(\sum p_i-1\right)$
- 熵的最大值的求解 07:32
  - **求导结果**：$\frac{\partial L}{\partial p_i}=-\log p_i-1+\lambda=0$
  - 解的形式：$p_i¿2^{\lambda-1}=\frac{1}{k}$（等概率分布）
- 熵的最大值与最小值 08:52
  - **最大值**：$logk$（当$p_i=1/k$时）
  - **最小值**：0（当某个$p_i=1$时）
- 等概率分布的假设 09:32
  - **古典概型基础**：解释了为什么古典概率中假设各样本点等概率

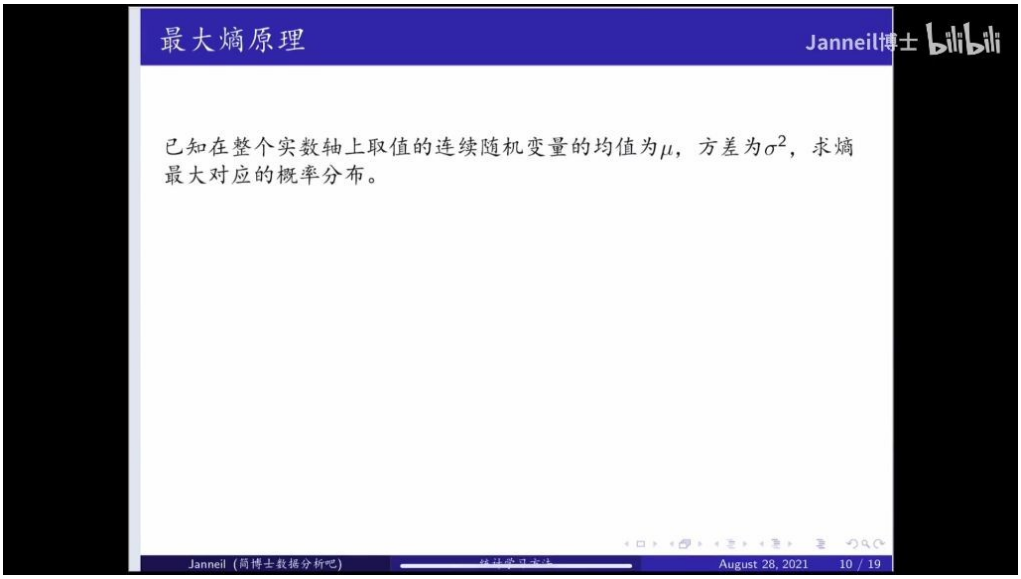
4）连续情况最大熵 09:54

- 最大熵原理概述与离散情况示例 09:58
  - 核心思想：在给定约束条件下寻找使熵最大的概率分布
- 例题 1: 常规约束下的最大熵 10:13

-



  - **题目解析**：
    - 随机变量 X 有 5 个取值$\{A, B, C, D, E\}$
    - 仅常规约束$\sum p_i = 1$时，最大熵分布为$p_i = 1/5$
- 例题 2: 增加一个约束条件下的最大熵 10:54
  - **新增约束**：$p_1 + p_2 = 0.3$
  - 解法：
    - 将变量分为两组：$\{A, B\}$和$\{C, D, E\}$
    - 组内等概率：$p_1 = p_2 = 0.15$，$p_3 = p_4 = p_5 \approx 0.233$
- 例题 3: 增加两个约束条件下的最大熵 12:17
  - **新增约束**：$p_1 + p_3 = 0.5$
  - 解法：
    - 建立方程组表达各概率关系
    - 通过求导找到使熵最大的$p_1 \approx 0.1859$
    - 其他概率通过关系式递推得出
- 连续随机变量的最大熵原理 16:14
  - **正态分布合理性**：解释了为什么正态分布是日常最常用的连续分布假设
- 例题 4: 已知均和方差的随机量的最大分布 16:49
  -



- **约束条件**：
  - $\int p(x)dx=1$
  - $\int xp(x)dx=\mu$
  - $\int(x-\mu)^2 p(x)dx=\sigma^2$
- 解法：
  - 构建拉格朗日函数
  - 对 p(x)求泛函导数
  - 解得$p(x)=\frac{1}{\sqrt{\square}}$
- **结论**：在给定均值和方差约束下，正态分布使熵达到最大

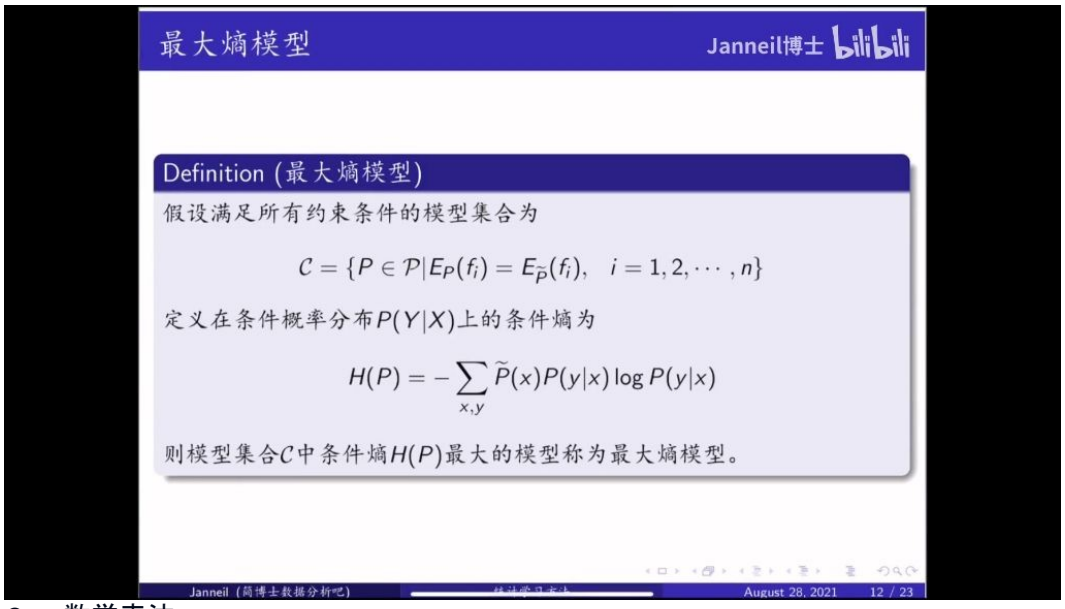
2. 最大熵模型定义 27:00

- **模型集合**：，表示满足所有约束条件的概率分布集合
- 核心思想：从所有可能的概率分布中，筛选出满足特定约束条件的分布集合
- **应用场景**：主要用于分类问题，通过条件概率分布 P(Y|X)进行预测

1）模型集合定义 27:11

- **集合构成**：花体C表示从所有概率分布P中，满足期望约束的那些分布
- **约束条件**：共有 n 个约束条件，确保模型保留训练数据中的统计特征
- **筛选过程**：类似于"弱水三千，只取一瓢"，从无限可能中选取符合要求的有限分布

2）条件熵定义 28:39

- 

- 数学表达：
- **特殊性**：不同于联合熵，这是针对条件概率分布 P(Y|X)的特有条件熵
- **选择原因**：
  - 最大熵模型是判别方法，需要直接建模条件概率
  - 输入变量 X 的信息必须充分利用
  - 联合熵$P(X,Y)$不适合判别任务的需求
- **优化目标**：在约束集合C中寻找使条件熵 H(P)最大的模型

3）模型应用方法

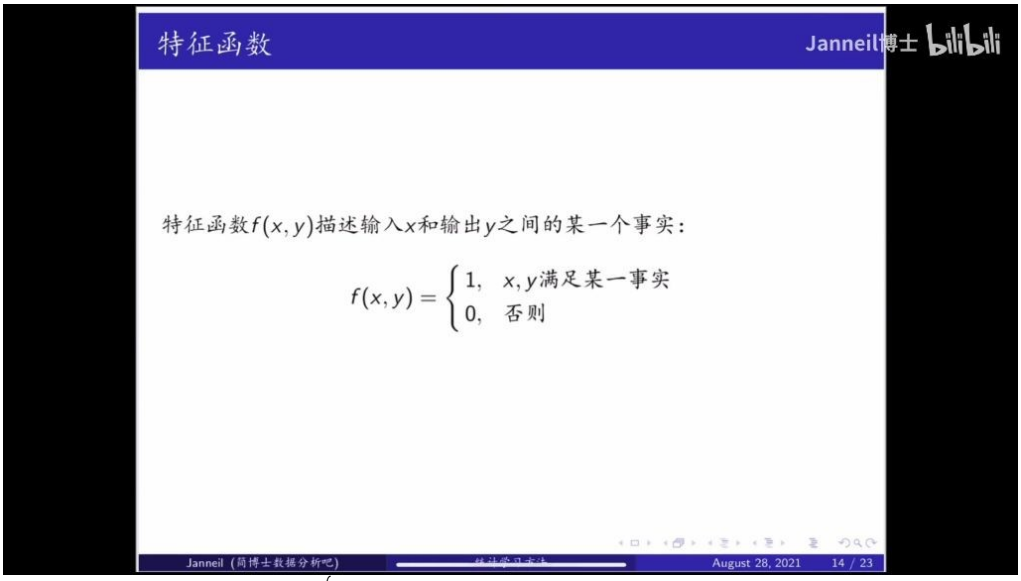
- **二分类应用**：
  - 给定 x=5 时，若$P(y=1\vee x=5)=0.7>P(y=0\vee x=5)=0.3$
  - 则预测输出$\hat{y}=1$
- **多分类应用**：
  - 对于 k 个类别$c_1,c_2,\ldots,c_k$
  - 算所有$P(y=c_i\vee x)$概率值
  - 选择概率最大的类别作为预测结果
- **平局处理**：

- 当多个类别概率相同时
- 可随机选择一个作为输出
- 类似决策树中叶子节点类别确定方法

3. 最大熵模型应用 35:12

1）特征函数 36:44

- 例题 1：汉字"打"的词性与发音 36:51
  - 

  - **多音字特性**：汉字"打"是一个提手旁加"丁"的多音字，具有二声"dá"和三声"dǎ"两种发音
  - **词性区分**：
    - 量词（二声）：一打鸡蛋、两打扑克、三打啤酒
    - （三声）：打蛋、打扑克、打、打球
  - **输入输出关系**：输入 x 为文字串（如一打鸡蛋），输出 y 为对应的发音（dá/dǎ）
- 特征函数的定义 40:09
  - 

  - **基本形式**：$f(x,y)=\begin{cases}1\\0\end{cases}$
  - 功能描述：描述输入 x 和输出 y 之间的特定事实关系
  - **应用场景**：在"打"字例子中，特征函数判断 x 前面是否有表示数量的词
- 例题 2：汉字"打"的特征函数 40:43

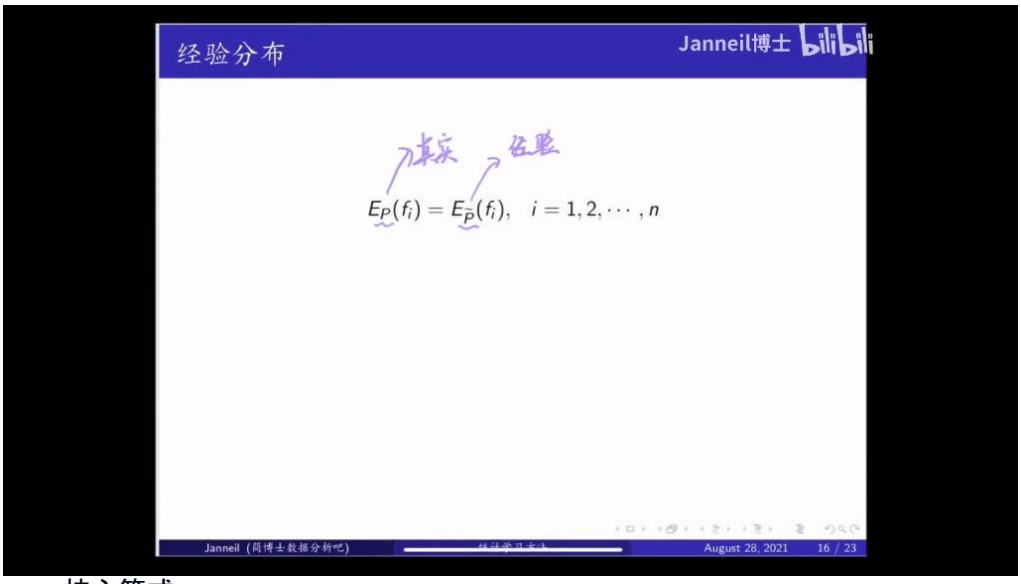
-

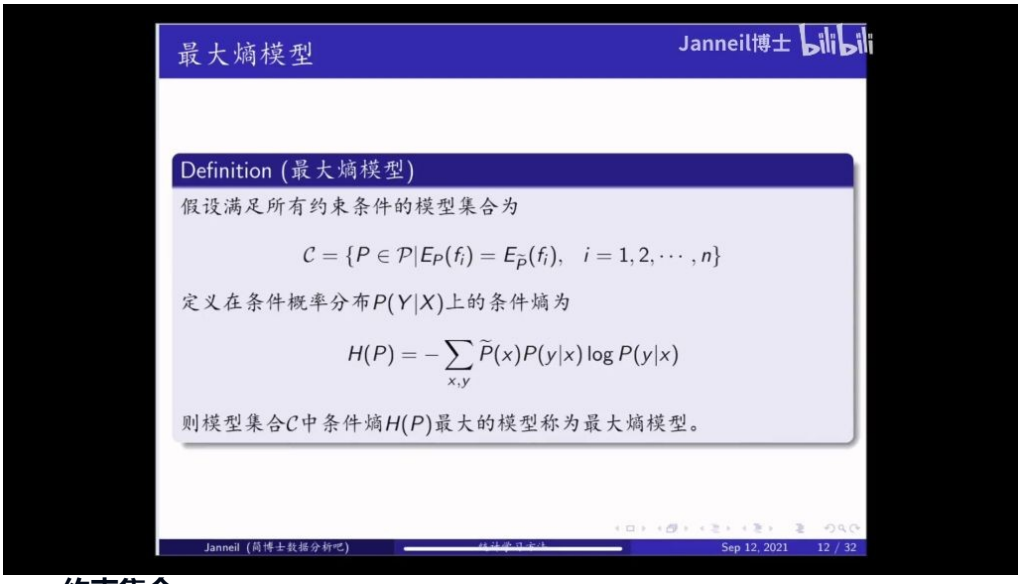

  - 特征**值计**算：
    - $f(x_1, y_1)=1$（一打鸡蛋）
    - $f(x_4, y_4)=0$（打蛋）
  - **规律总结**：当 x 前有数字（一、二、三等）时特征值为 1，否则为 0
- 例外情况与多特征函数 42:04
  - **例外处理**：如"三打白骨精"中"打"是动词而非量词
  - **多特征扩展**：
    - f_1：判断前面是否有数字
    - f_2：判断是否来自特定语义环境（如西游记）
  - 特征函数数量：n=2（当使用两个特征函数时）

2）经验分布 44:51

- 分布与上帝角下的真分布 44:57
  -


  - **上帝视角**：理论真实分布（如硬币正反面各 0.5）
  - **经验分布**：通过实验数据得到的估计分布（如 100 次抛掷 48 正 52 反）
  - **关系**：经验分布是真实分布的估计值
- 特征函数的约束条件 47:10

-


  - 核心等式：
  - **符号说明**：
    - P：真实分布（上帝视角）
    - ：经验分布（训练数据）
- 条件概率分布与联合概率分布的关系 50:02
  - 公式：$P(Y \vee X) = \frac{P(X, Y)}{P(X)}$
  - **目标**：求条件概率分布 P(Y|X)
- 条件概率分布的估计 50:52
  - 近似方法：用经验分布替代真实 P(X)
  - 表达式：
- 约束条件与条件概率分布求解 52:43
  -


  - **约束集合**：
  - **求解目标**：在C中找到条件熵最大的模型
- x 的经验分布 52:56
  - **计算方法**：
  - **示例**：若 x 有 m 个取值，统计每个x_i出现的频率
- x 和 y 联合的经验分布 54:03

- **计算方法**：
- **注意**：需要统计所有(x,y)组合的出现频率

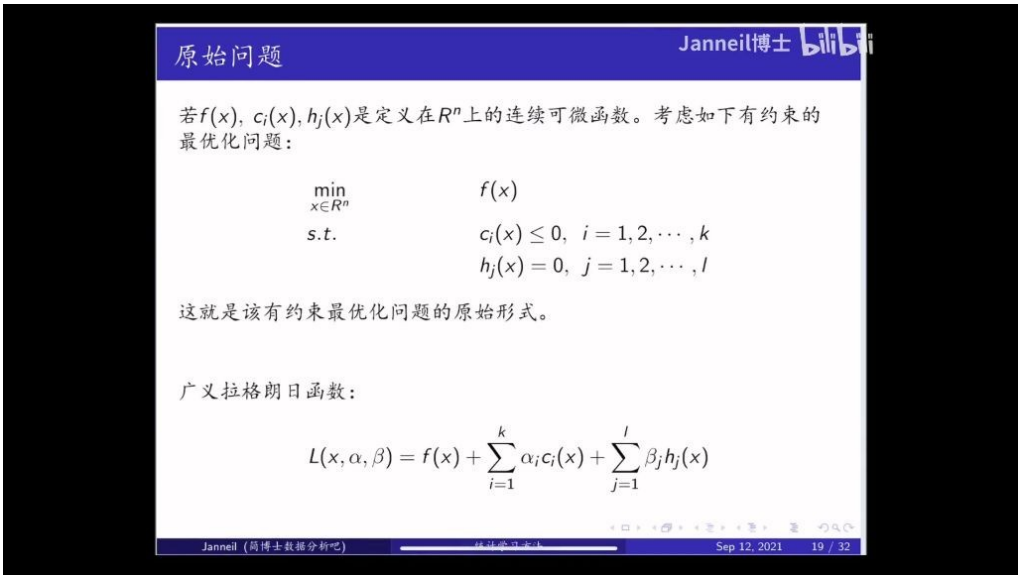
- 条件熵的计算 55:19
  - **定义式**：
  - **展开形式**：考虑所有 x 取值和对应 y 的条件概率
  - **物理意义**：衡量在已知 x 情况下 y 的不确定性

4. 最大熵模型学习 59:24

1）优化问题 01:00:01

2）原始问题 01:01:29

- 原始问题定义
  -



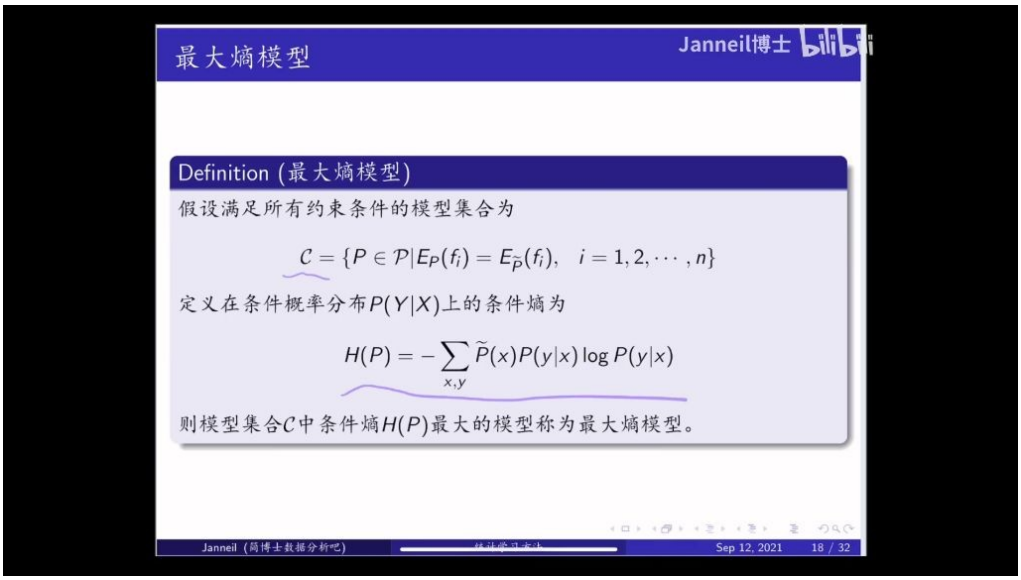
形式: 考虑定义在R^n上的连续可微函数 f(x)，带约束的最优化问题为：

$min_{x\in R^n} f(x)$

$s.t. c_i(x)\le 0, i=1,2,...,k$

  - $h_j(x)=0, j=1,2,...,l$
  - **约束分类**: 约束条件分为不等式约束(≤ 0)和等式约束(=0)两类，共 k+l 个约束条件
  - **连续可微要求**: 目函数和束函数都要求可微，便于后求偏数

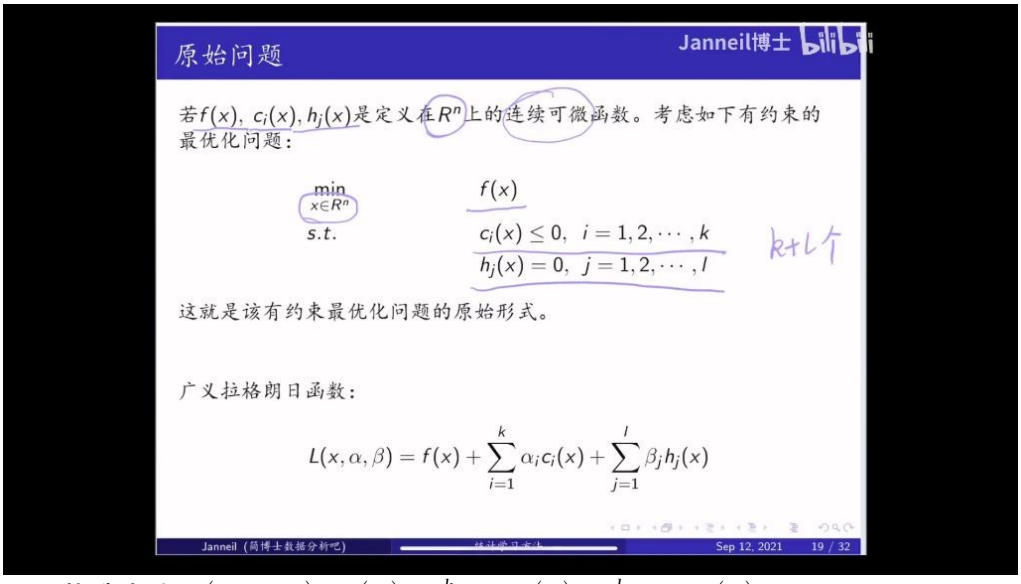
- 最大熵模型 01:03:26
  -



  - 模型集合:

- **条件熵定义**:
- **优化目标**: 在模型集合C中寻找使条件熵 H(P)最大的模型
- **等价转换**: 求最大熵等价于求负熵的最小值，可统一为最小化问题形式

- 广拉格朗日函数 01:03:35
  -



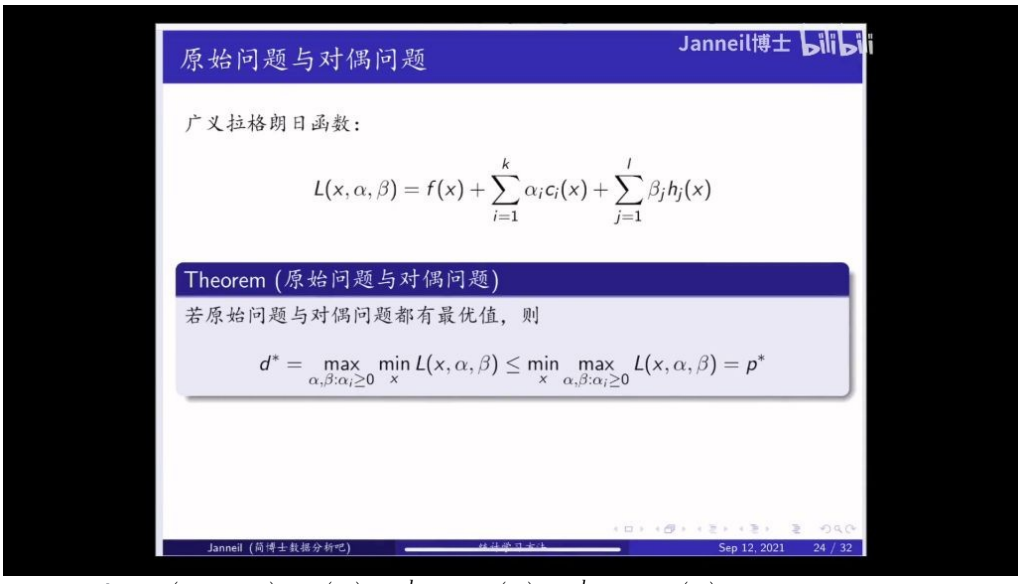
  - 构造方法:$L(x,\alpha,\beta)=f(x)+\sum_{i=1}^{k}\alpha_i c_i(x)+\sum_{j=1}^{l}\beta_j h_j(x)$
  - **变量说明**:
    - x: n 维向量(原始变量)
    - α: k 维向量(不等式约束乘子)
    - β: l 维向量(等式约束乘子)
  - 函数特点: 将约束条件通过拉格朗日乘子整合到目标函数中
- 拉格朗日小八卦 01:06:22
  - **生平背景**: 1736 年出生于意大利的法国数学家
  - **趣闻轶事**:
    - 出身富裕家庭但不愿继承家业
    - 17 岁时因热爱数学而希望家道中落
    - 18 岁即获得欧拉赏识
  - 数学贡献: 拉格朗日中值定理、拉格朗日公式等多项重要成果
- 原始问题的最优值 01:08:57
  - 构造函数:$\theta_P(x)=max_{\alpha,\beta:\alpha_i\ge 0}L(x,\alpha,\beta)$
  - **最优值表示**:$p^*=min_x\theta_P(x)=min_x max_{\alpha,\beta:\alpha_i\ge 0}L(x,\alpha,\beta)$
  - 函数性**质**:
    - 满足约束时$\theta_P(x)=f(x)$
    - 不满足约束时$\theta_P(x)=+\infty$
- 为什么有约束的情况下，大 px 等于 fx 01:12:20
  - **约束满足时**:
    - 不等式约束$c_i(x)\le 0$，当$\alpha_i=0$时第二项为 0
    - 等式约束$h_j(x)=0$，第三项恒为 0
    - 最终$\theta_P(x)=f(x)$
  - **约束违反时**:
    - 任一$c_i(x)>0$，可取$\alpha_i\to+\infty$使整体$+\infty$
    - 任一$h_j(x)\ne 0$，可取$\beta_j\to\pm\infty$使整体$+\infty$
- 原始问题总结 01:17:43
  - **求解思路**: 通过构造广义拉格朗日函数将约束优化问题转化为无约束问题

- **关键步骤**:
  - 定义拉格朗日函数
  - 构造$\theta_P(x)$函数
  - 求解极小极大问题
- **优势**: 简化了带约束的优化问题求解过程

3）对偶问题 01:17:57

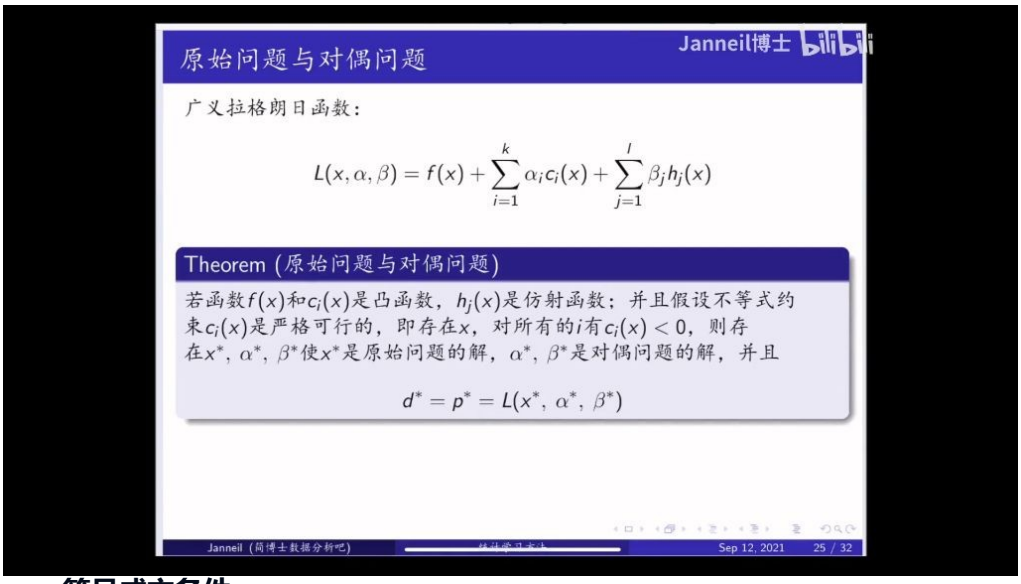
- **定义**：为解决原始优化问题而提出的转化形式，通过对广义拉格朗日函数$L(x,\alpha,\beta)$先求关于 x 的最小值，再求关于α,β的最大值
- 表示形式：$d^*=max_{\alpha,\beta:\alpha_i\geq 0}min_x L(x,\alpha,\beta)$
- **优势**：相比原始问题p^*更易求解，因当α,β已知时，求函数最小值相对简单
- **关系**：对偶问题最优值d^*与原始问题最优值p^*满足$d^*\leq p^*$

4）KKT 条件 01:20:44

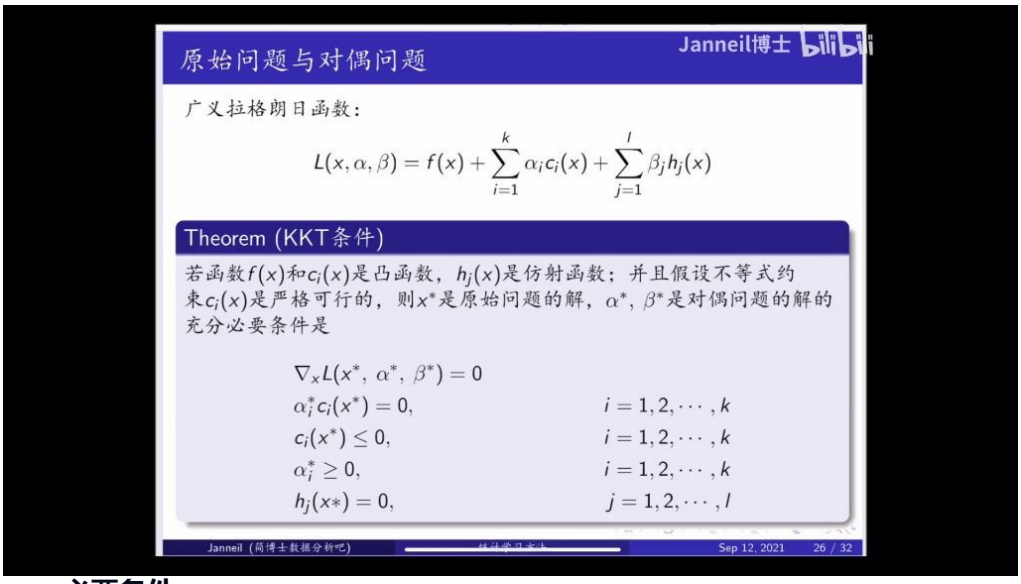
- 广义拉格朗日函数
  - 

  - 形式：$L(x,\alpha,\beta)=f(x)+\sum_{i=1}^{k}\alpha_i c_i(x)+\sum_{j=1}^{l}\beta_j h_j(x)$
  - **组成**：包含目标函数 f(x)、不等式约束$c_i(x)$（对应乘子α_i）和等式约束$h_j(x)$（对乘子β_j）
- 原始问题与对偶问题：定理 1 01:21:02
  - **原始问题**：$min_x max_{\alpha,\beta:\alpha_i\geq 0}L(x,\alpha,\beta)\Leftrightarrow p^*$
  - **对偶问题**：$max_{\alpha,\beta:\alpha_i\geq 0}min_x L(x,\alpha,\beta)\Leftrightarrow d^*$
  - **强弱关系**：$d^*\leq p^*$，即对偶问题的最优值不超过原始问题的最优值
- 原始问题与对偶问题：定理 2 01:24:04

-



  - **等号成立条件**：
    - f(x)和$c_i(x)$是凸函数
    - $h_j(x)$是仿射函数
    - 不等式约束严格可行（存在x使所有$c_i(x)$<0）
  - **结论**：满足上述条件时，存在x^*,α^*,β^*使得$d^*=p^*=L(x^*,\alpha^*,\beta^*)$
- 凸函数与仿射函数 01:24:31
  - **凸函数示例**：$y=x^2$（呈现"smile"形状），任意两点连线位于函数图像上方
  - 仿射函数：形如$Ax+b$的线性变换，包含线性回归模型作为特例
  - **严格可行条件**：存在至少一个x使得所有不等式约束$c_i(x)$严格小于0
- 原始问题与对偶问题：KKT 条件 01:26:40
  -



  - **必要条件**：
    - $\nabla_x L(x^*,\alpha^*,\beta^*)=0$（梯度为零）
    - $\alpha_i^* c_i(x^*)=0$（互补松弛条件）
    - $c_i(x^*)\le 0$（原始不等式约束）
    - $\alpha_i^*\ge 0$（对偶不等式约束）
    - $h_j(x^*)=0$（原始等式约束）

- **充分性**：在凸优化问题中，KKT 条件既是必要条件也是充分条件
- **应用**：通过 KKT 条件可以验证找到的解是否为最优解

二、最大熵模型 01:28:33

1. 最大熵模型学习 01:28:41

1）学习目标 01:29:16

- 核心目**标**：通过训练数据集和特征函数，建立条件概率分布模型用于新实例的类别预测

2）已知信息 01:29:28

- **训练数据集**：记作 T，包含 N 个样本，用于模型训练
- 特征函数：记作 f，共 n 个，描述 x 与 y 之间的事实关系
- **双重约束**：模型需同时满足训练数据拟合和 n 个特征函数约束

3）模型目的 01:30:25

- 核心**应**用：当新实例仅知 x 时，通过条件概率分布 P(y|x)预测类别 y
- **预测机制**：将 x 代入分布函数计算各类别概率，取最大概率作为预测结果

4）实现方法 01:31:29

- 方法论：采用最大熵模型进行概率分布训练
- **理论基础**：熵最大化原则保证模型在最不确定情况下做出最合理的推断

5）熵函数 01:31:38

- **定义式**：$H(P)=-\sum P(y\vee x)logP(y\vee x)$
- **优化目标**：求熵 H(P)的最大值对应概率分布

6）约束条件 01:32:25

- 特征约束：n 个特征函数对应 n 个期望约束
- **常规约束**：$\sum_y P(y\vee x)=1$（概率归一化条件）
- **总约束数**：共$n+1$个约束条件（n 个特征+1 个常规）

7）优化问题转化 01:33:26

- **问题转换**：通过负号将最大化问题转化为最小化问题
- **标准形式**：$min-H(P)$s.t.$h_j(P)=0$（全部为等式约束）
- **约束处理**：通过移项使等式右边归零，符合标准优化问题形式

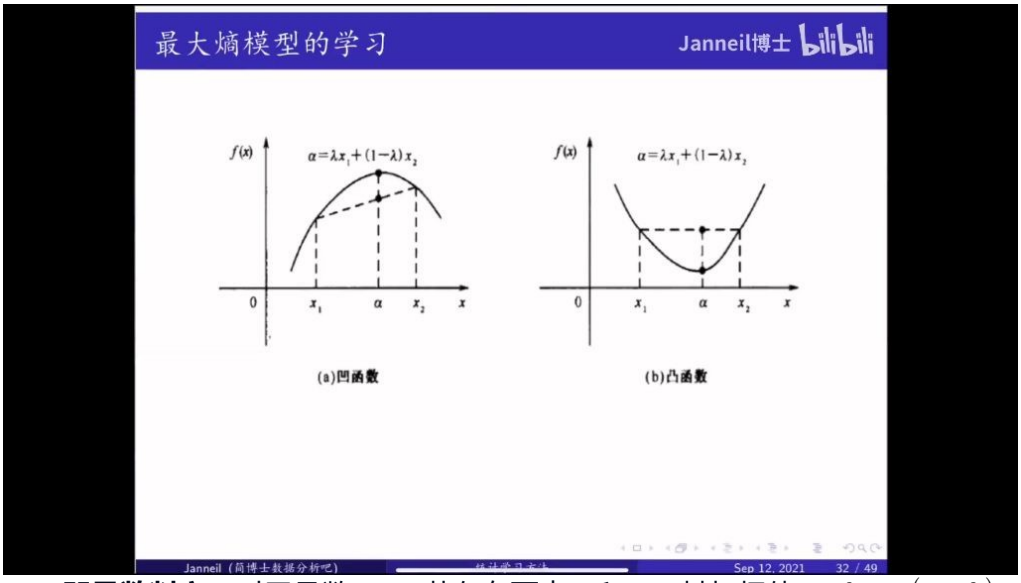
8）拉格朗日函数 01:35:30

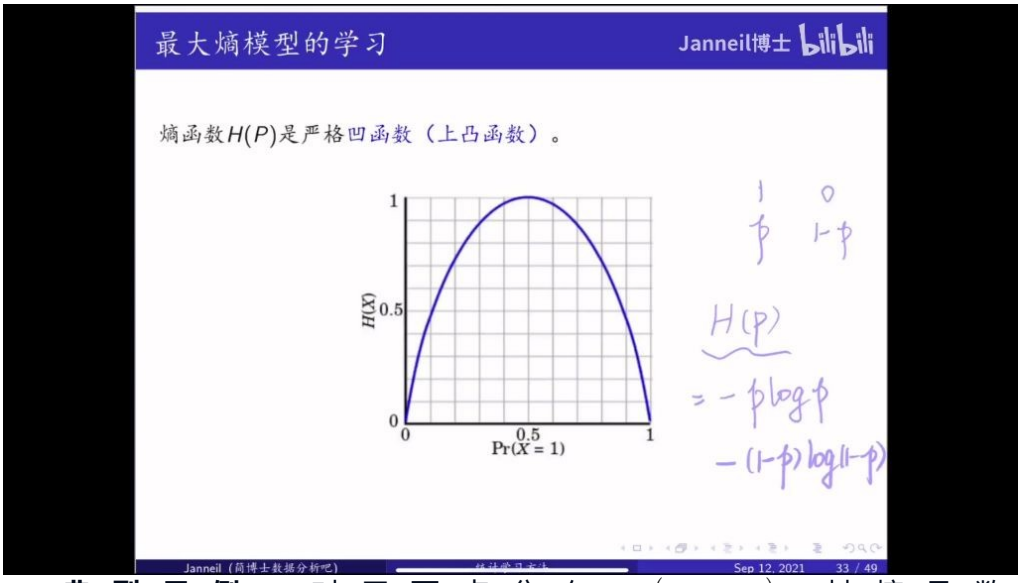
- 有约束的最小化问题与拉格朗日乘子法 01:35:40
  - 构造方法：将目标函数与约束条件线性组合
  - 函数形式：$L(P,\omega)=-H(P)+\sum\omega_j h_j(P)$
- 拉格朗日乘子：欧米伽零与欧米伽 I 01:36:19
  - **乘子分配**：常规约束乘子记作ω_0，特征约束乘子记作ω_i
  - 参数数：共$n+1$个拉格朗日乘子对应全部约束
- 条件概率的求和式与特征函数的期望 01:36:45
  - 两种等价形式：
    - $1-\sum P(y\vee x)$或$\sum P(y\vee x)-1$
    - $E_{\ P}[f]-E_P[f]$或$E_P[f]-E_{\ P}[f]$
  - **经验分布**： $P(x)$表示训练数据的经验分布
- 条件概率分布与经验分布 01:38:02
  - **分布替代**：用经验分布 $P(x)$近似真分布 P(x)
  - **理论依据**：大数定律保证经验分布收敛于真实分布

9）原始问题与对偶问题 01:38:49

- 解等价条件：
  - 目标函数 f(x)凸函数
  - 约束$h_j(x)$仿射函数
- **验证结论**：
  - 熵函数 H(P)的负数为凸函数
  - 所有约束均为线性（仿射）函数
- **求解保证**：满足 KKT 条件时原始问题与对偶问题同解

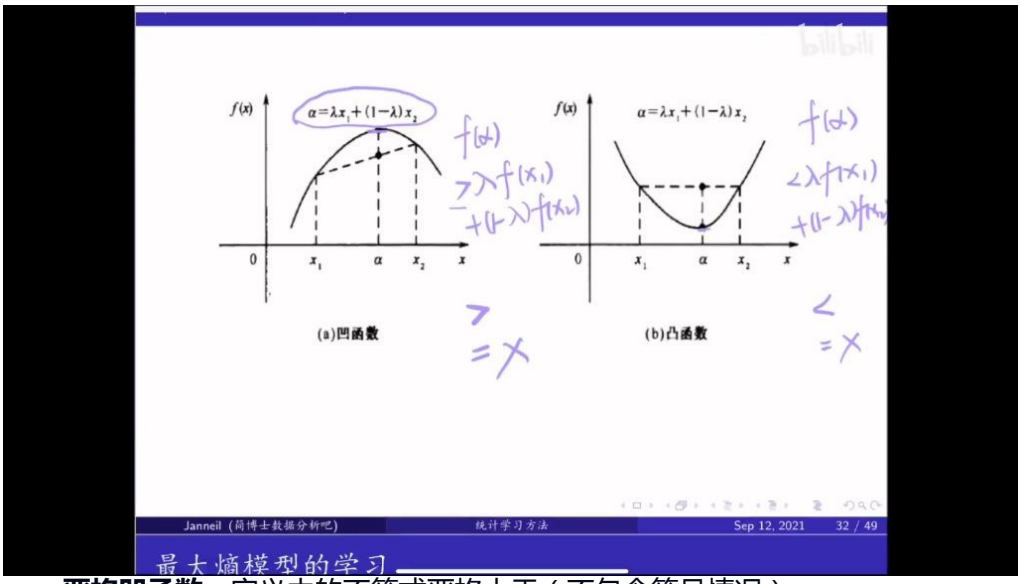
10）熵函数性质 01:41:09

- 凸函数与凹函数的定义 01:41:16
  - 

  - **凹函数判定**：对于函数 f(x)，若存在两点x_1和x_2，其加权值$\alpha=\lambda x_1+(1-\lambda)x_2$满足$f(\alpha)>\lambda f(x_1)+(1-\lambda)f(x_2)$，则称 f(x)为凹函数
  - 凸函数判定：当不等式方向相反，即$f(\alpha)<\lambda f(x_1)+(1-\lambda)f(x_2)$时，称为凸函数
  - **几何意义**：凹函数图像呈"上凸"形状，凸函数图像呈"下凸"形状
- 凸函数与凹函数的极值点 01:42:33
  - **极值特性**：
    - 凸函数的极值点为全局最小值点，此时偏导数为零
    - 凹函数的极值点为全局最大值点，此时偏导数也为零
  - **应用价值**：研究函数凹凸性可帮助确定优化问题的极值点性质
- 熵函数 H(P)是严格凹函数 01:42:59
  - 

  - **典型示例**：对于两点分布$X(p,1-p)$，其熵函数为$H(P)=-plogp-(1-p)\log(1-p)$
  - 图像特征：绘制出的熵函数曲线呈现明显的上凸形状，验证其凹函数性质
  - **严格凹性**：该函数在定义域内是严格凹函数（上凸函数）
- 严格凹函数与严格凸函数的定义 01:44:28

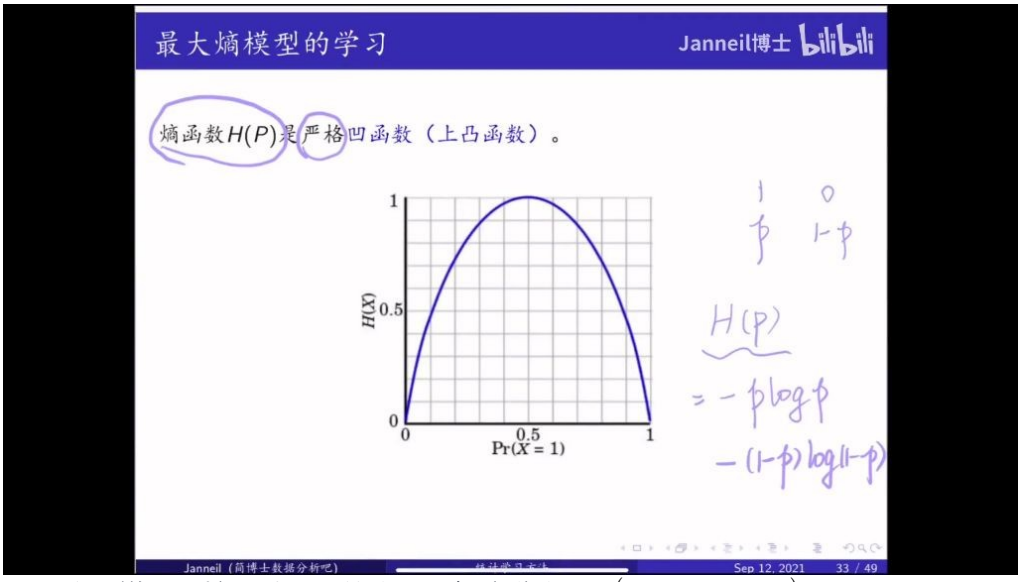
-



  - **严格凹函数**：定义中的不等式严格大于（不包含等号情况）
  - **严格凸函数**：定义中的不等式严格小于（不包含等号情况）
  - **重要性**：严格性保证了极值点的唯一性

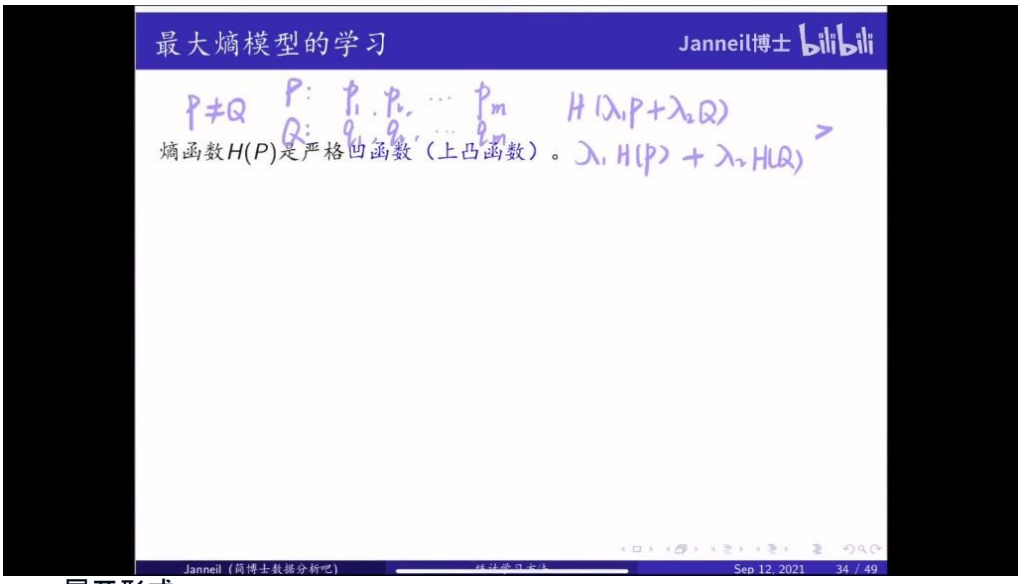
11）严格凹函数证明 01:44:58

- 熵函数的定义及变量 01:45:09
  -



  - **变量说明**：熵函数 H(P)的变量是概率分布$P=(p_1,p_2,\dots,p_m)$
  - **比较对象**：设另有概率分布$Q=(q_1,q_2,\dots,q_m)$，且$P\neq Q$
- 熵函数加权求和性质 01:45:53
  - **目标关系式**：需证明$H(\lambda_1 P+\lambda_2 Q)>\lambda_1 H(P)+\lambda_2 H(Q)$
  - **权重条件**：$\lambda_1+\lambda_2=1$且$\lambda_1,\lambda_2>0$
  - **新分布定义**：记$O=\lambda_1 P+\lambda_2 Q=(o_1,o_2,\dots,o_m)$
- 熵函数不等式证明思路 01:47:33

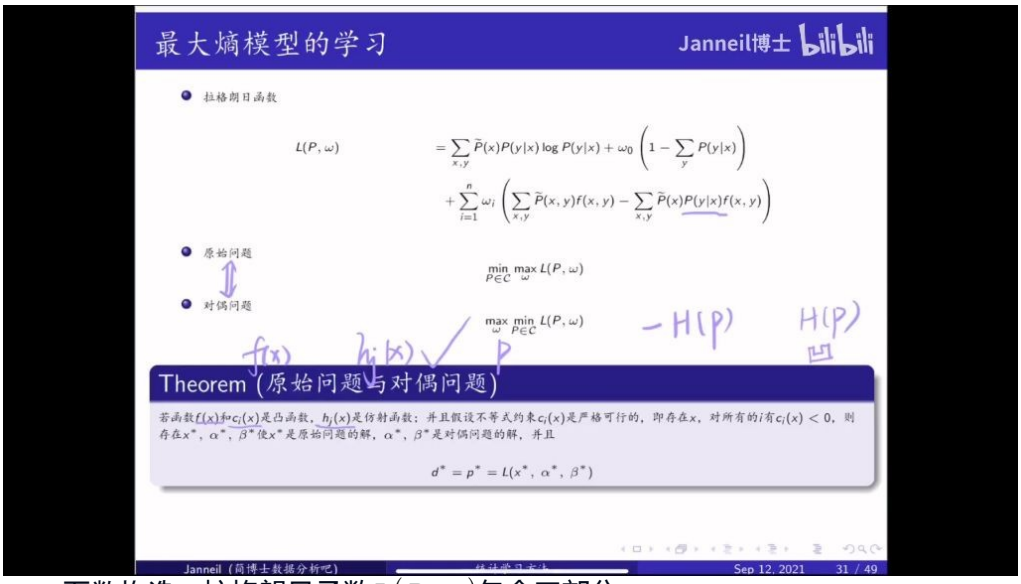
-

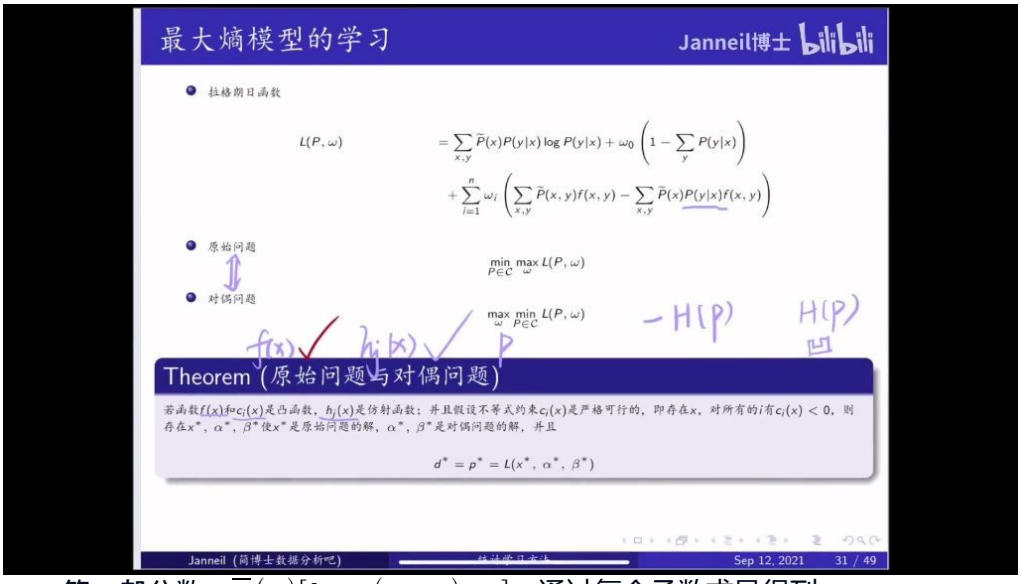


  - 展开形式：
    - 左边：$-\sum[\lambda_1 p_i+\lambda_2 q_i]\log[\lambda_1 p_i+\lambda_2 q_i]$
    - 右边：$-\lambda_1\sum p_i\log p_i-\lambda_2\sum q_i\log q_i$
  - **转化目标**：证明$\sum p_i\log\frac{o_i}{p_i}+\sum q_i\log\frac{o_i}{q_i}<0$
- 熵函数不等式证明过程 01:49:13
  - **对数不等式应用**：利用$logt\le t-1$（当且仅当$t=1$取等）
  - **放缩处理**：
    - $p_i\log\frac{o_i}{p_i}\le p_i(\frac{o_i}{p_i}-1)¿o_i-p_i$
    - 求和得$\sum o_i-\sum p_i=1-1=0$
  - **严格不等条件**：因$P\neq Q$，故$o_i\neq p_i$，等号不成立
- 熵函数不等式证明结论 01:57:16
  - **最终结论**：熵函数 H(P)是严格凹函数
  - 推论：负熵$-H(P)$是严格凸函数
  - **应用意义**：该性质保证了最大熵原理中极值点的唯一性和全局最优性

12）对偶问题求解 01:57:41

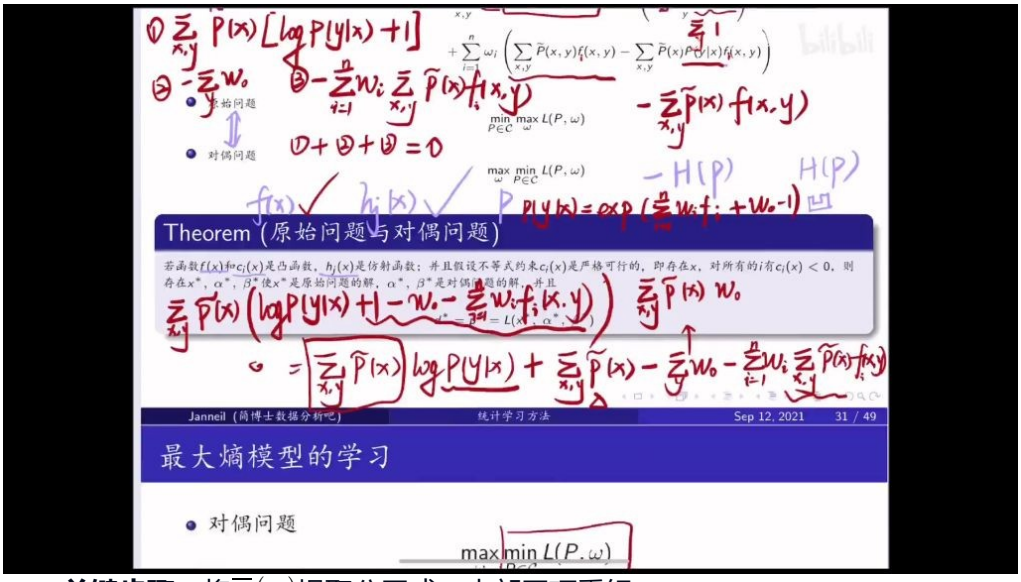
- 对偶问题与内部极小化问题概述 01:57:50

-


  - 函数构造：拉格朗日函数$L(P,\omega)$包含三部分：
    - 熵项：$\sum_{x,y}\overline{P}(x)P(y\vee x)logP(y\vee x)$
    - 概率归一约束：$\omega_0(1-\sum_y P(y\vee x))$
    - 特征期望约束：$\sum_{i=1}^{n}\omega_i(\sum_{x,y}\overline{P}(x,y)f(x,y)-\sum_{x,y}\overline{P}(x)P(y\vee x)f(x,y))$
  - **优化目标**：原始问题转化为$min_{P\in C}max_{\omega}L(P,\omega)$的对偶问题
- 定理：原始问题与对偶问题 01:58:28
  - 凸性条件：当 f(x)和$c_i(x)$是凸函数，$h_j(x)$是仿射函数，且不等式约束严格可行时
  - 解的存在性：存在$x^{¿},\alpha^{¿},\beta^{¿}$使得原始问题与对偶问题解相等，且$d^{¿}=p^{¿}=L(x^{¿},\alpha^{¿},\beta^{¿})$
  - **求解策略**：将 min-max 问题转化为 max-min 问题，先求内部极小化再求外部极大化
- 偏导数的计算过程 01:59:01
  -


  - 第一部分数：$\overline{P}(x)[logP(y\vee x)+1]$，通过复合函数求导得到
  - **第二部分导数**：$-\omega_0$，直接对线性项求导
  - **第三部分导数**：$-\sum_i\omega_i\overline{P}(x)f_i(x,y)$，特征函数相关项

- **合并结果**：$\frac{\partial L}{\partial P(y \vee x)} = \overline{P}(x)[logP(y \vee x) + 1 - \omega_0 - \sum_i \omega_i f_i(x,y)]$

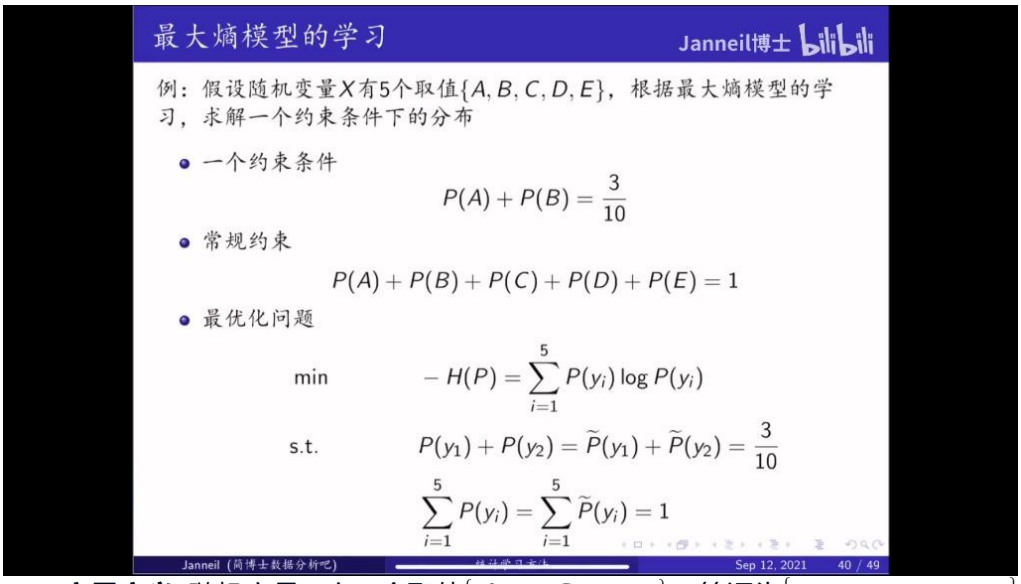
- 偏导数为零的整理 02:01:37
  - 

  - **关键步骤**：将$\overline{P}(x)$提取公因式，内部四项重组
  - **变形技巧**：对ω_0乘以$\sum_y P(y \vee x) = 1$的恒等式
  - **中间结果**：$logP(y \vee x) = -1 + \omega_0 + \sum_i \omega_i f_i(x,y)$
- 条件概率的表达式 02:06:27
  - 指数形式解：$P(y \vee x) \propto e^{-1+\omega_0+\sum \omega_i f_i(x,y)}$
  - **参数解释**：完全由$n+1$个拉格朗日乘子$\{\omega_0, \omega_1, \ldots, \omega_n\}$和特征函数$f_i$决定
- 内部极小化问题所得函数与条件概率分布 02:07:04
  - 函数定**义**：
  - **条件分布**：
  - **参数关系**：每个ω对应一个特定的条件概率分布
- 约束条件与规范化因子 02:08:17
  - **概率归一化**：利用$\sum_y P(y \vee x) = 1$约束确定规范化因子
  - **Z 函数定义**：$Z_\omega(x) = \sum_y e^{\sum_i \omega_i f_i(x,y)}$，消除$\omega_0$项
  - **最终形式**：$P_\omega(y \vee x) = \frac{e^{\sum_i \omega_i f_i(x,y)}}{Z_\omega(x)}$
- 外部极大化与拉格朗日乘子的求解 02:11:01
  - **优化目标**：，寻找最优拉格朗日乘子ω^*
  - **求解方法**：通过偏导计算或数值优化方法
  - 模型完成：将ω^*代回得到最终条件概率分布$P_{\omega^*}(y \vee x)$

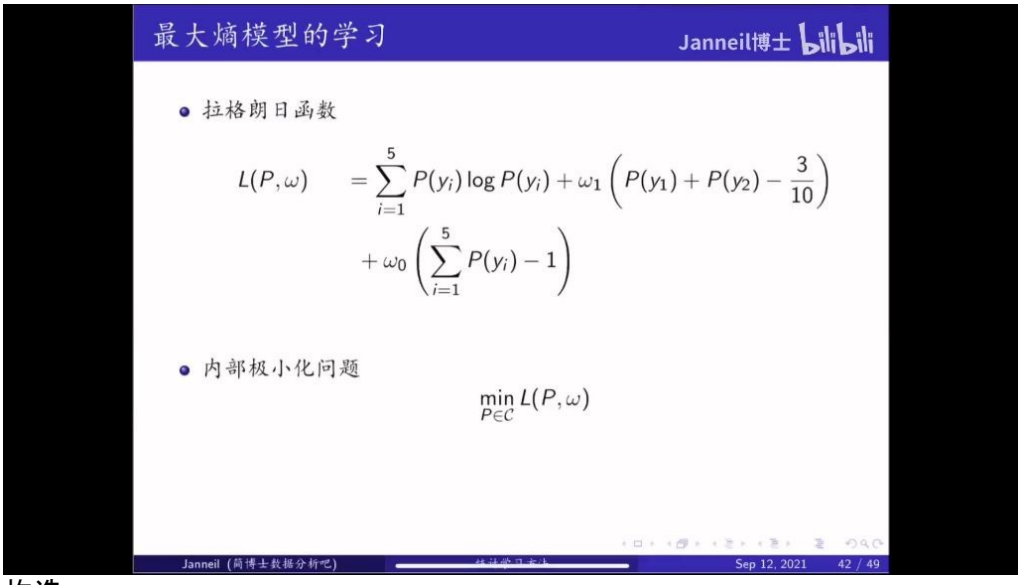
13）例题:约束条件下分布 02:12:33

- 问题描述与建模

-



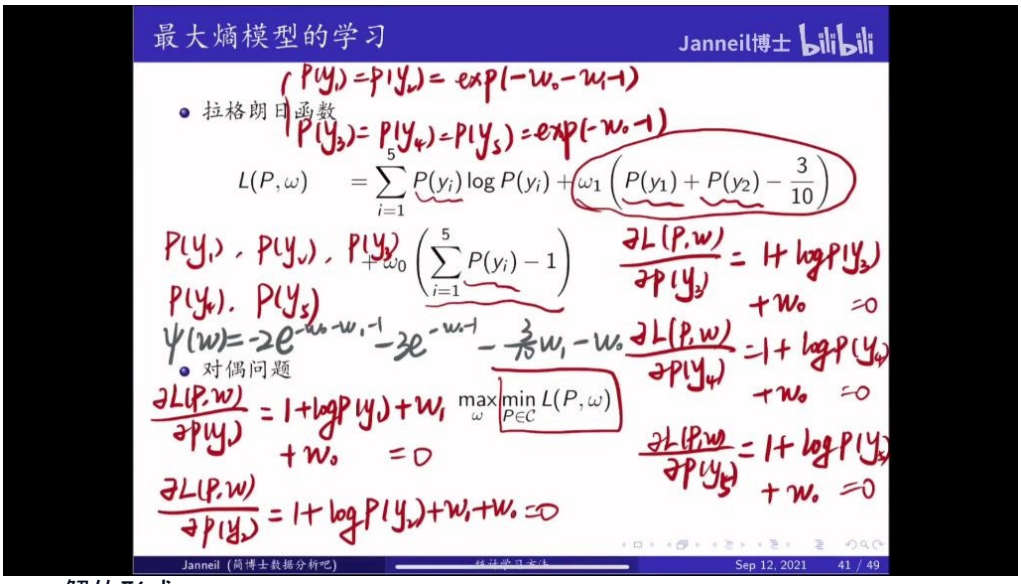
  - **变量定义**: 随机变量 X 有 5 个取值$\{A,B,C,D,E\}$，简记为$\{y_1,y_2,y_3,y_4,y_5\}$
  - **约束条件**:
    - 特征约束：$P(y_1)+P(y_2)=\frac{3}{10}$
    - 常规约束：$\sum_{i=1}^{5}P(y_i)=1$
  - **优化目标**: 最小化负熵$-H(P)=\sum_{i=1}^{5}P(y_i)logP(y_i)$
- 拉格朗日函数构建
  -



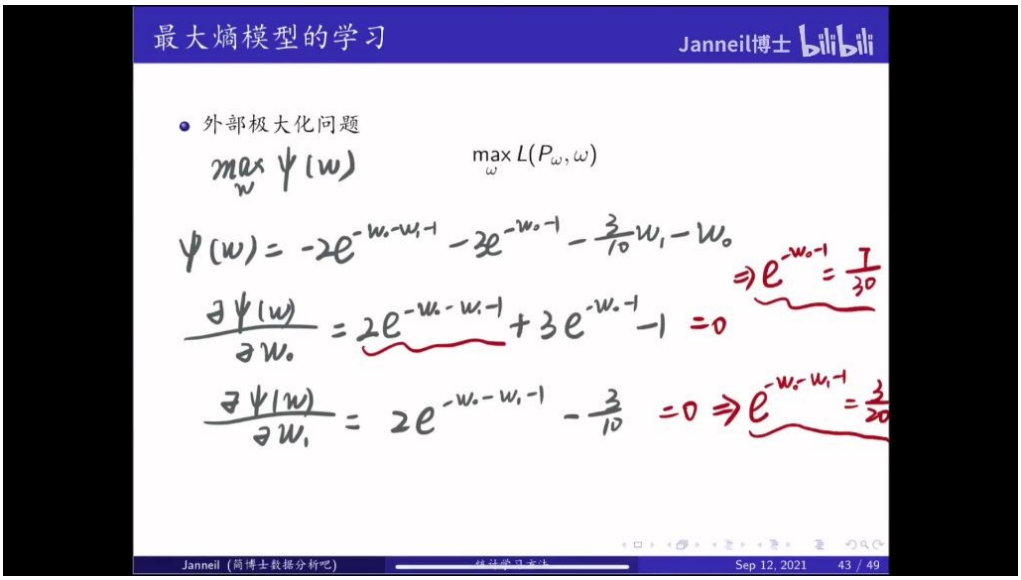
函数构造:

  - $L(P,\omega)=\sum_{i=1}^{5}P(y_i)logP(y_i)+\omega_1\left(P(y_1)+P(y_2)-\frac{3}{10}\right)+\omega_0\left(\sum_{i=1}^{5}P(y_i)-1\right)$
  - **内部极小化**:
    - 对$P(y_1)$求偏导：$1+logP(y_1)+\omega_1+\omega_0=0$
    - 对$P(y_2)$求偏导：$1+logP(y_2)+\omega_1+\omega_0=0$
    - 对其他变量求偏导：$1+logP(y_i)+\omega_0=0$（$i=3,4,5$）
- 概率分布求解

-



  - 解的形式:
    - $P(y_1)=P(y_2)=e^{-\omega_0-\omega_1-1}$
    - $P(y_3)=P(y_4)=P(y_5)=e^{-\omega_0-1}$
  - **参数关系**:
    - 由约束条件可得：$e^{-\omega_0-\omega_1-1}=\frac{3}{20}$
    - $e^{-\omega_0-1}=\frac{7}{30}$
- 外部极大化问题
  -



目函数:

  - $\psi(\omega)=-2e^{-\omega_0-\omega_1-1}-3e^{-\omega_0-1}-\frac{3}{10}\omega_1-\omega_0$
  - **求解过程**:
    - 令偏数零得到方程
    - 解得：$e^{-\omega_0-\omega_1-1}=\frac{3}{20}$，$e^{-\omega_0-1}=\frac{7}{30}$
- 最终概率分布

- **分布结果**:
  - $P(y_1)=P(y_2)=\frac{3}{20}$
  - $P(y_3)=P(y_4)=P(y_5)=\frac{7}{30}$
- **验证**:
  - 满足原始约束条件：$\frac{3}{20}+\frac{3}{20}=\frac{3}{10}$
  - 概率总和为 1：$2\times\frac{3}{20}+3\times\frac{7}{30}=1$

三、极大似然估计 02:25:13

1. 基本原理

- 估方法：逻辑回归和最大熵模型都可以通过极大似然方法来估计条件概率分布 P(y|x)
- 核心思想：在已知样本集的情况下，找到使似然函数最大的条件概率分布

2. 对数似然函数 02:27:09

1）离散分布与连续分布的概率问题 02:27:15

- 离散分布：可以直接讨论概率值
- **连续分布**：只能讨论概率密度，需要将密度最大化
- **最大熵模型**：因为是分类问题，属于离散概率分布，可以直接求概率

2）四然函数的研究对象 02:28:24

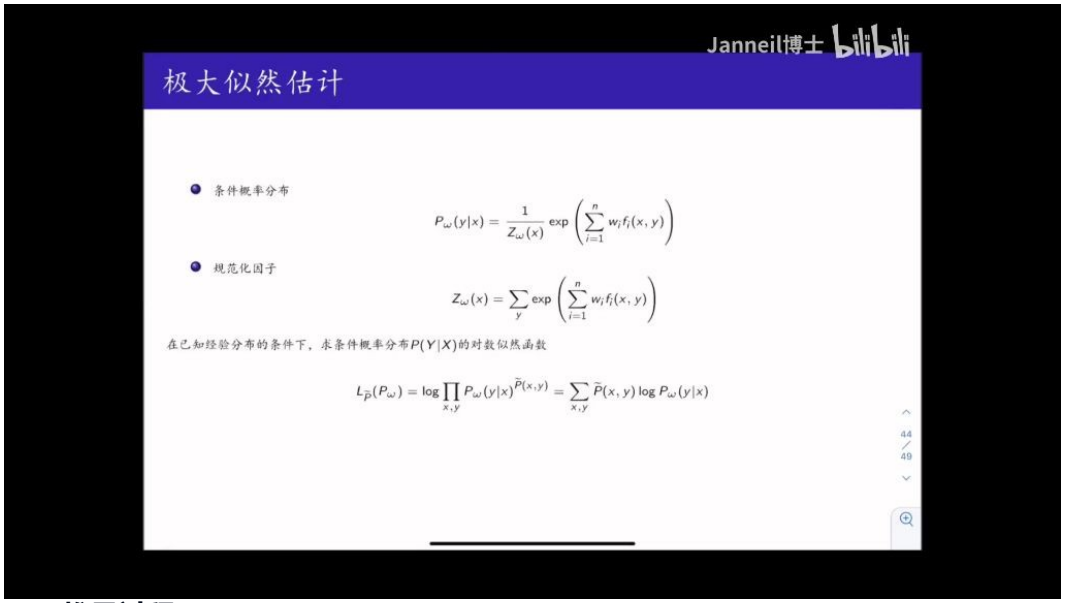
- **研究对象**：条件概率分布 P(y|x)，即似然函数中的未知量
- **目标**：在已知 x 情况下求 y 的条件概率分布

3）表达式形式与对数似然函数 02:29:01

- **原始形式**：样本出现概率的连乘形式
- **对数转换**：为简化运算，取对数将乘除变为加减，得到求和式
- **对数似然函数**：

4）例题 1:最大商模型的对数似然函数 02:30:02

-

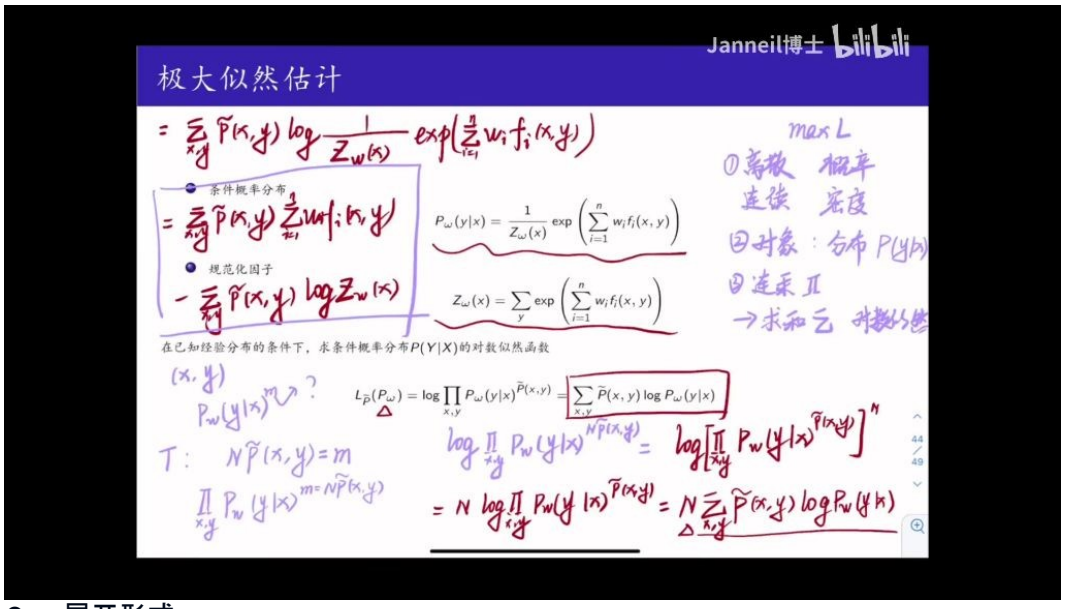


- **推导过程**：
  - 个本点(x,y)的概率为$P_\omega(y\vee x)$
  - 样本点出现次数
  - 所有样本点的联合概率为$\prod P_\omega(y\vee x)^m$
  - 取对数后得到对数似然函数

5）聚焦参数简化问题 02:34:06

- **参数化表示**：将条件概率表示为$P_\omega(y\vee x)=\frac{1}{Z_\omega(x)}\exp(\sum_{i=1}^n w_i f_i(x,y))$
- **规范化因子**：$Z_\omega(x)=\sum_y \exp(\sum_{i=1}^n w_i f_i(x,y))$

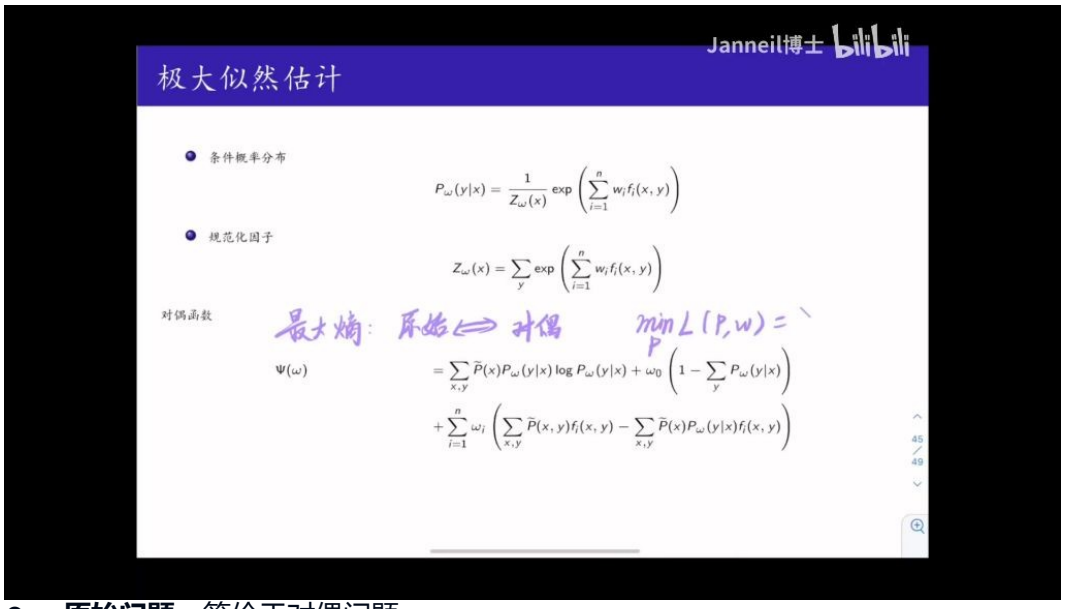
6）例题 2:用欧米伽表示的条件概率的对数似然函数 02:34:29

- 

- 展开形式：
  - 将对数除法展开为两项相减
  - 第一项：$\sum_{i=1}^n w_i f_i(x,y)$
  - 第二项：$\log Z_\omega(x)$

3. 最大熵模型应用 02:36:23

1）最大熵模型的对偶问题 02:36:39

- 

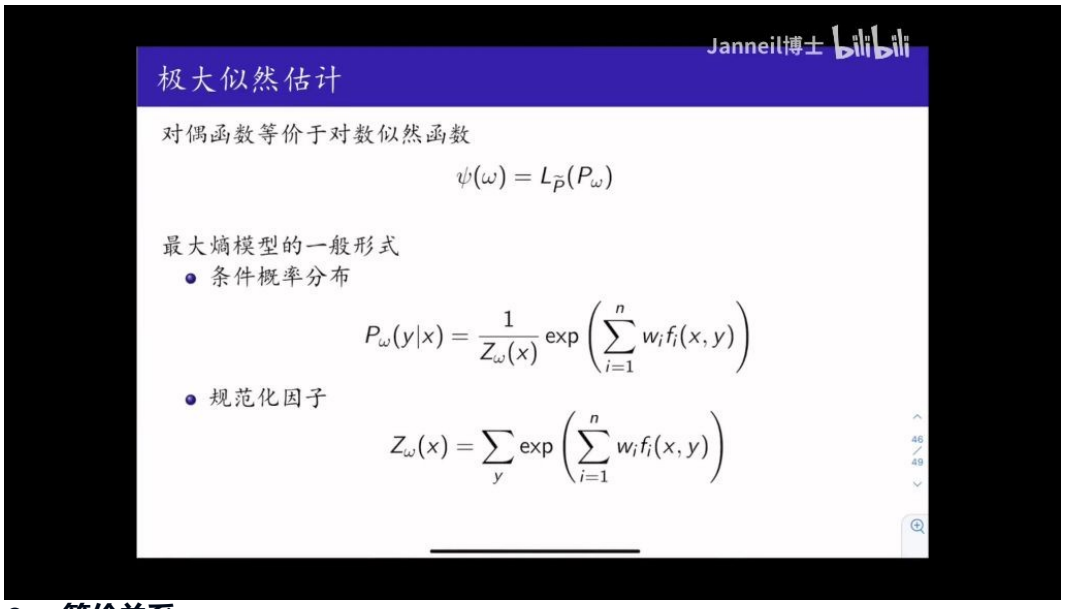
- **原始问题**：等价于对偶问题
- **对偶函数**：$\psi(\omega)=\sum_{x,y}\overline{P}(x)P_\omega(y\vee x)\log P_\omega(y\vee x)+$*约束项*

2）对偶函数的简化过程 02:38:02

- 化依据：$\sum_y P_\omega(y\vee x)=1$
- **简化结果**：保留两项
  - $\sum_{i=1}^n w_i \sum_{x,y}\overline{P}(x,y)f_i(x,y)$

- $\sum_x \overline{P}(x)\log Z_\omega(x)$

3）最大熵模型的对偶函数与对数似然函数等价 02:42:32



- **等价关系**：
- **优化问题**：转化为求$\psi(\omega)$的最大值问题

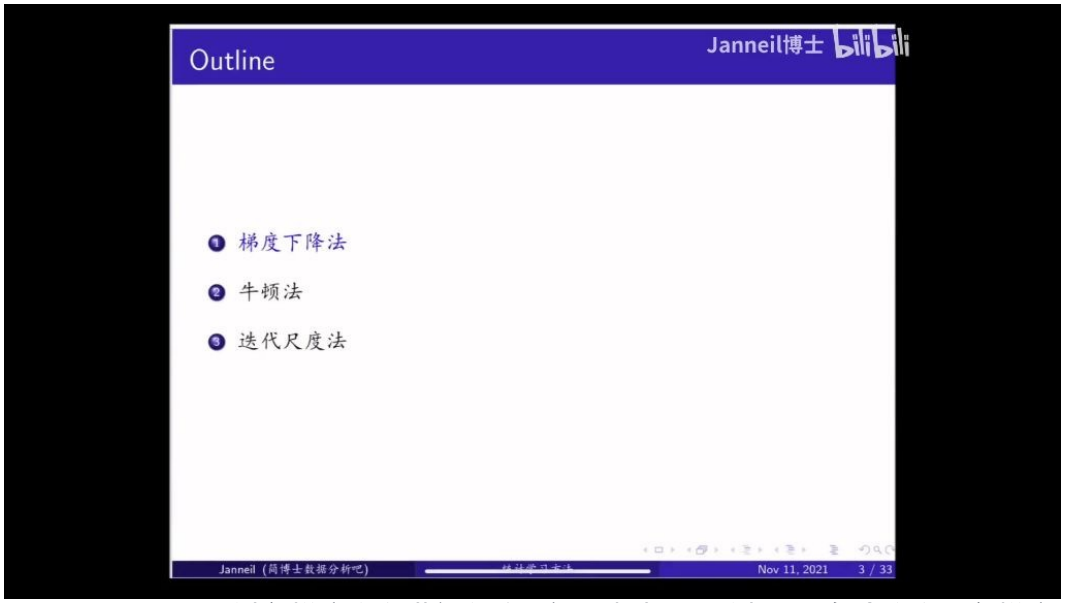
4）最大熵模型的特点 02:43:34

- 本质特征：在满足所有约束条件下，找到信息熵最大的条件概率分布
- **约束条件影响**：约束条件数量影响模型拟合和预测能力，需要平衡
- **计算复杂度**：条件数量多时计算量大，实际应用可能困难

四、迭代算法 02:45:50

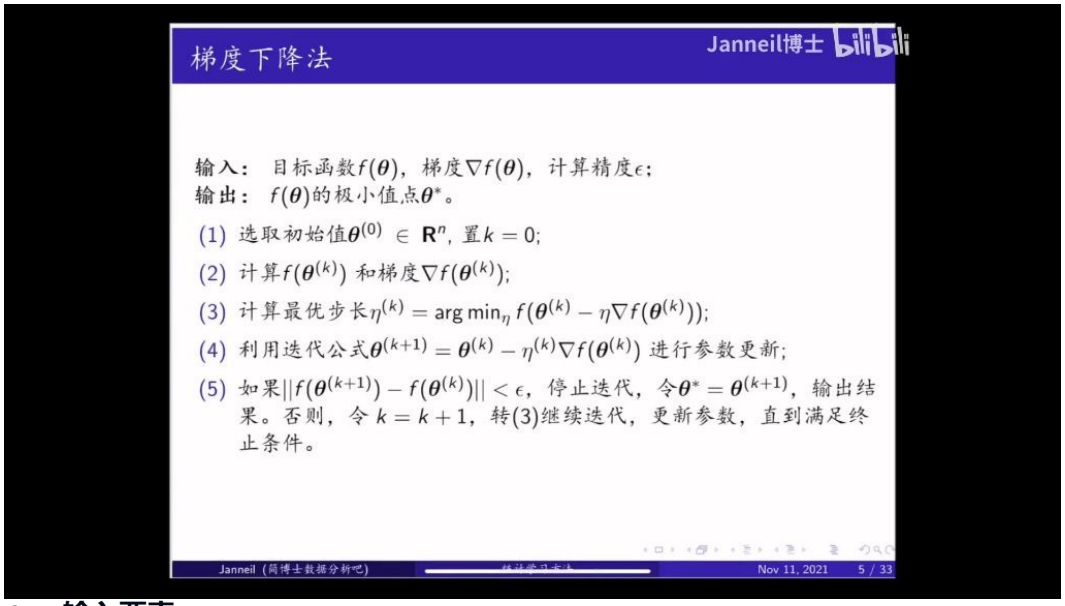
1. 梯度下降法

1）算法原理 02:47:12



- 核心思想：通过负梯度方向进行参数更新，类比"下山过程"，包含方向（负梯度）和步长两个关键因素。
- 迭代公式：$\theta^{(k+1)}=\theta^{(k)}-\eta^{(k)}\nabla f(\theta^{(k)})$，其中$\eta^{(k)}$为第 k 次迭代的最优步长。
- 数学基础：基于一阶泰勒展开式，通过函数值变化量¿$f(\theta^{(k+1)})-f(\theta^{(k)})$∨¿判断收敛。

2）算法流程 02:51:53

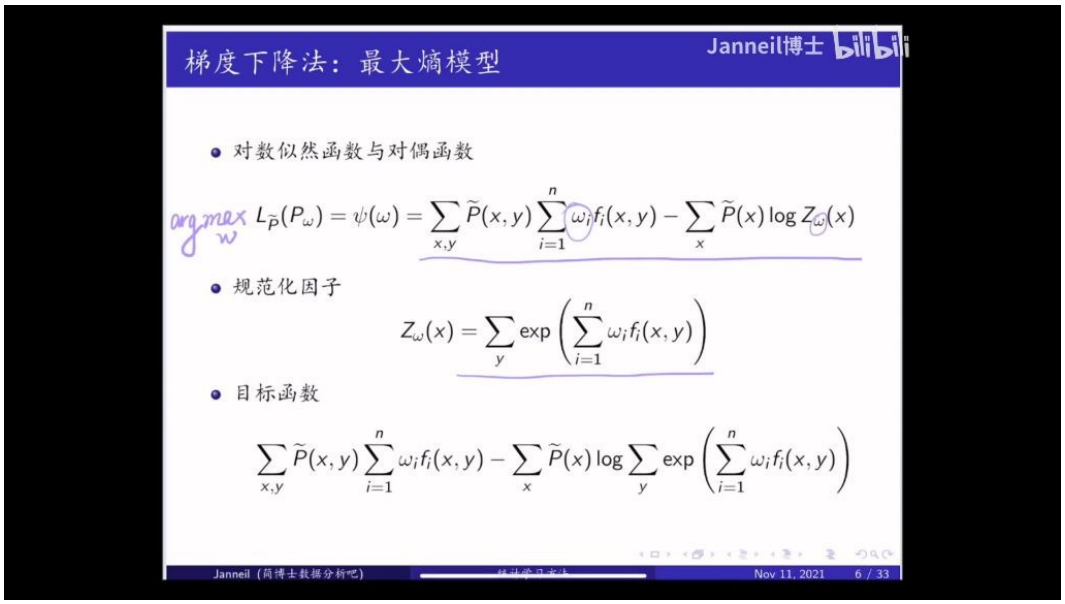
- 

- **输入要素**：
  - 目标函数$f(\theta)$
  - 梯度$\nabla f(\theta)$
  - 计算精度ϵ
- **关键步骤**：
  - 初始化参数$\theta^{(0)}$和迭代计数器 k=0
  - 计算当前点的函数值$f(\theta^{(k)})$和梯度$\nabla f(\theta^{(k)})$
  - 通过线搜索确定最优步长
  - 更新参数$\theta^{(k+1)}$
  - 止条件$¿ f(\theta^{(k+1)}) - f(\theta^{(k)}) \vee ¿ \epsilon$

五、最大熵模型中的梯度下降法 02:58:03

1. 模型构建

1）对数似然函数 02:58:10

- 

- 函数形式：$L_{P}(P_\omega) = \sum_{x,y} P(x,y) \sum_{i=1}^{n} \omega_i f_i(x,y) - \sum_x P(x) \log Z_\omega(x)$

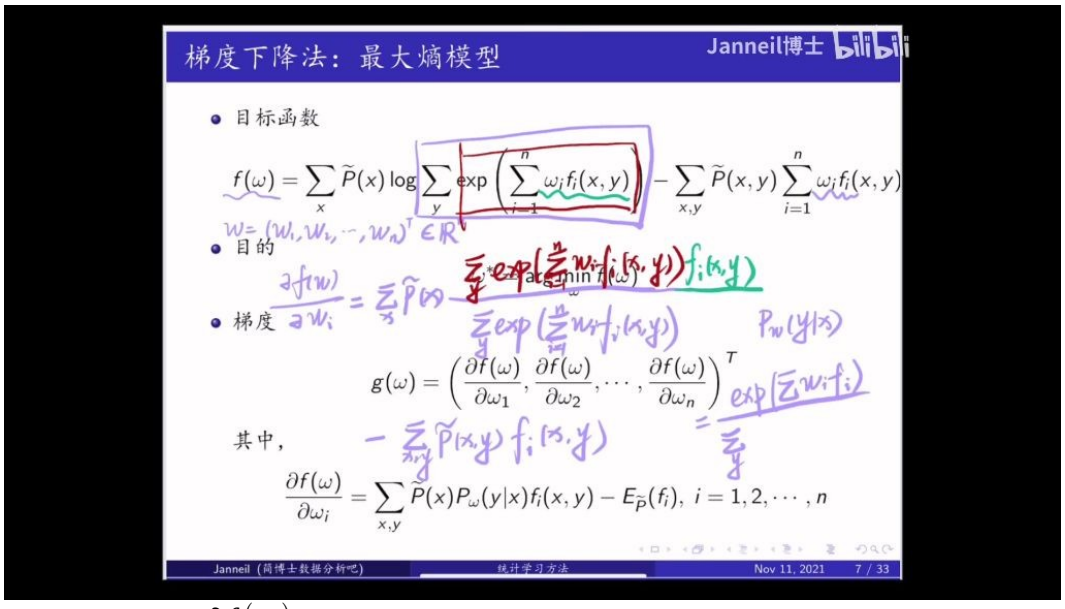
- **规范化因子**：$Z_{\omega}(x)=\sum_{y}\exp\left(\sum_{i=1}^{n}\omega_{i}f_{i}(x,y)\right)$

2）目标函数转化 03:00:05

- **转化方法**：将最大化问题转化为最小化问题，通过添加负号得到新目标函数 $f(\omega)$¿$-L_{P}(P_{\omega})$
- **参数空间**：$\omega\in R^{n}$，表示 n 维参数向量$(\omega_{1},\omega_{2},\ldots,\omega_{n})$

3）梯度向量计算 03:01:06

-

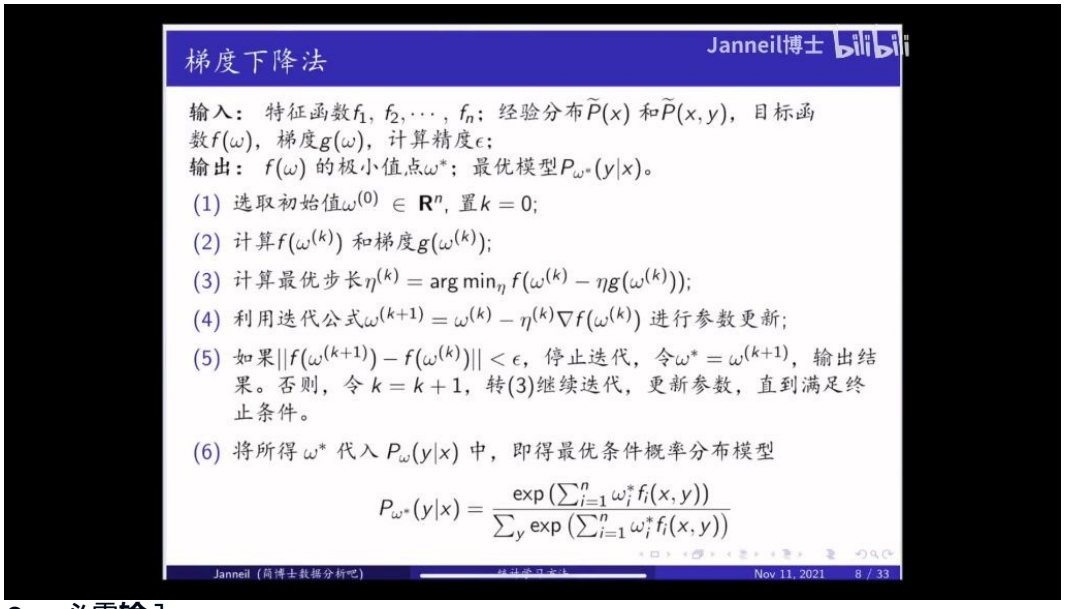


- 偏数公式：$\frac{\partial f(\omega)}{\partial\omega_{i}}=\sum_{x,y}P(x)P_{\omega}(y\vee x)f_{i}(x,y)-E_{P}(f_{i})$
- **梯度向量**：$g(\omega)=\left(\frac{\partial f}{\partial\omega_{1}},\ldots,\frac{\partial f}{\partial\omega_{n}}\right)^{T}$

2. 算法 03:08:14

1）输入参数 03:08:27

-



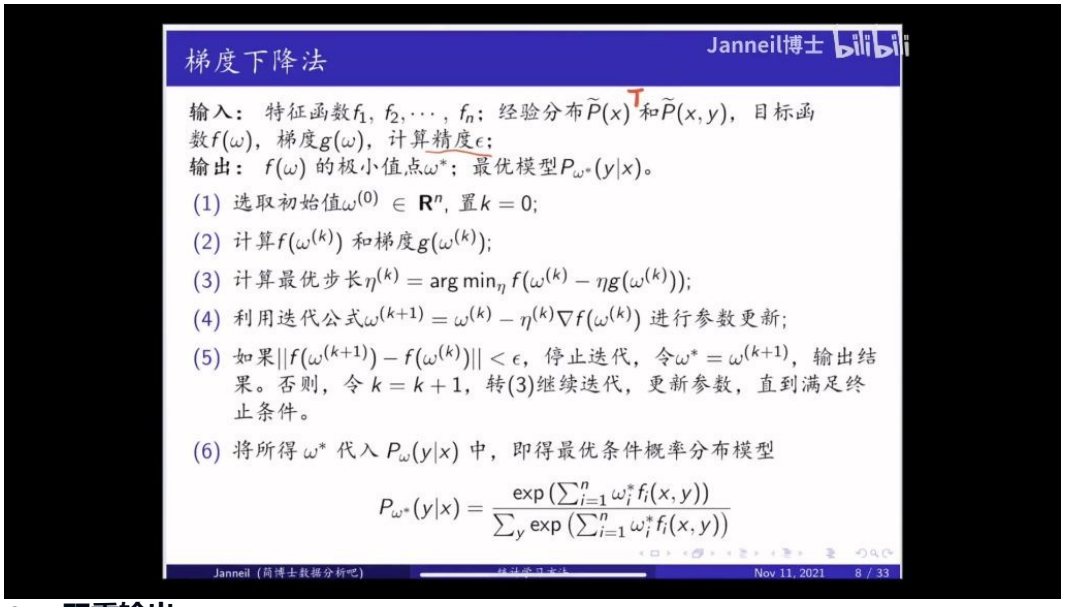
- **必需输入**：
  - n 个特征函数$f_1,\ldots,f_n$
  - 经验分布 $P(x)$和 $P(x,y)$

- 目标函数$f(\omega)$及其梯度$g(\omega)$
- 计算精度ϵ

2）迭代步骤 03:10:11

- 特殊**处**理：
  - 步长$\eta^{(k)}$通过优化子问题动态确定
  - 每次迭代需计算条件概率$P_{\omega}(y \vee x)$
- **终止条件**：相邻迭代点函数值差小于ϵ

3）输出结果 03:12:29

- 

- **双重输出**：
  - 最优参数$\omega^{¿}$

条件概率分布模型：

  - $P_{\omega^{¿}}(y \vee x) = \dfrac{\exp\left(\sum_{i=1}^{n} \omega_i^{¿} f_i(x,y)\right)}{\sum_y \exp\left(\sum_{i=1}^{n} \omega_i^{¿} f_i(x,y)\right)}$
- **应用价值**：得到的条件概率分布可直接用于分类预测

六、牛顿法 03:13:15

1. 历史背景 03:14:06

1）牛顿与苹果树的故事 03:14:10

- **教育意义**: 通过牛顿被苹果砸到的故事，教育人们要善于观察和思考，这个故事启发了万有引力定律的发现。
- **文化关联**: 苹果在多个文化中具有特殊意，如《圣》中当夏娃吃的禁果、格林童话《白雪公主》中的毒苹果，以及现代科技公司苹果的象征意义。

2）斯蒂格勒五称定律 03:16:13

- **定义**: 由科学史学家史蒂格勒提出的现象，指许多以人名命名的概念并非由该人发明。
- **实例**: 牛顿法并非牛顿发明，但以他的名字命名，这正符合斯蒂格勒五称定律。

3）牛顿法的起源：古巴比伦计算平方根的方法 03:16:33

-

- 方法原理:
  - 选择初始近似值x_0（小于$\sqrt{❑}$）
  - 计算$\frac{a}{x_0}$（大于$\sqrt{❑}$）
  - 取平均值$x_1=\frac{1}{2}\left(x_0+\frac{a}{x_0}\right)$
  - 重复迭代直至精度满足要求
- **实例演示**: 计算$\sqrt{❑}$
  - 初始值$x_0=1.4$
  - 第一次迭代：$x_1=\frac{1}{2}\left(1.4+\frac{2}{1.4}\right)=1.4143$
  - 第二次迭代：$x_2=\frac{1}{2}\left(1.4143+\frac{2}{1.4143}\right)=1.41421357$（与真实值 1.41421356 非常接近）

4）牛顿迭代法的真正发明者：辛普森 03:22:10

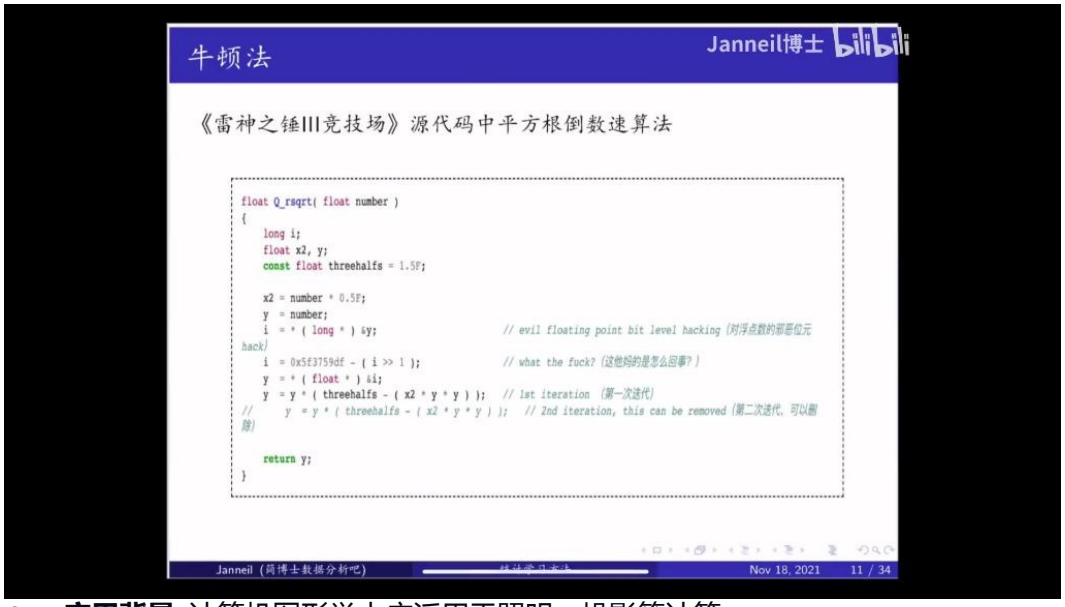
- **关键贡献**: 英国数学家辛普森首次提炼出牛顿迭代法的两个核心要素：
  - 迭代思想
  - 迭代公式（与现代形式基本相同）

5）傅里叶对牛顿迭代法的改进 03:23:01

- 改内容:
  - 使用现代导数符号（f',f''等）重新表达迭代公式
  - 首次将方法命名为"牛顿法"

6）例题：雷神之锤中平方根倒数速算法 03:23:40

-

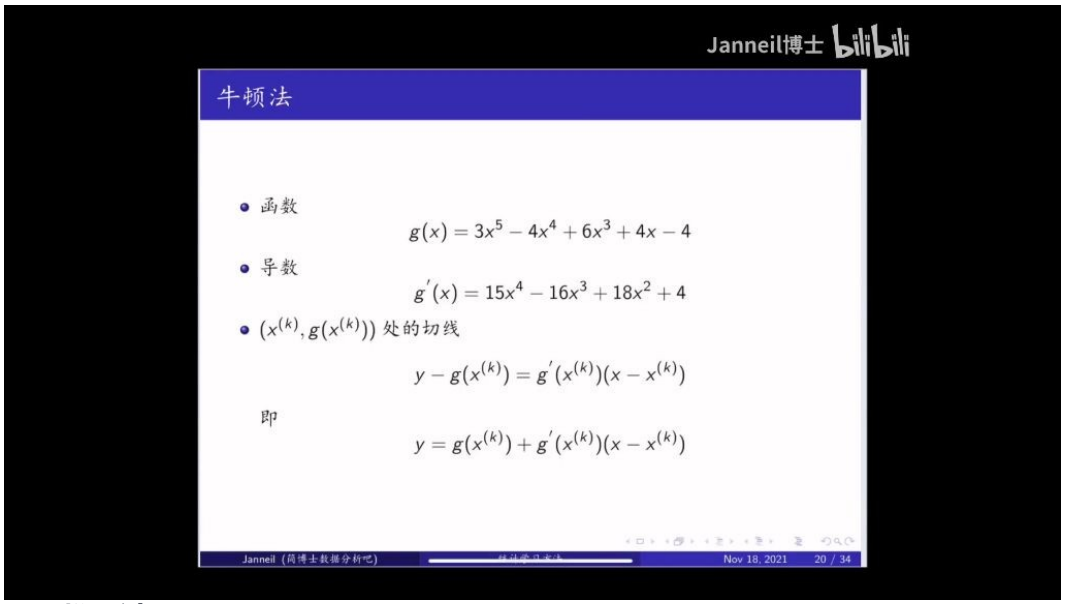


- **应用背景**: 计算机图形学中广泛用于照明、投影等计算
- **算法特点**:
  - 使用牛顿迭代法
  - 初始值为"魔术数字"（0x5f3759df）
  - 只需 1-2 次迭代即可达到所需精度

2. 函数零点求解 03:25:27

1）迭代公式

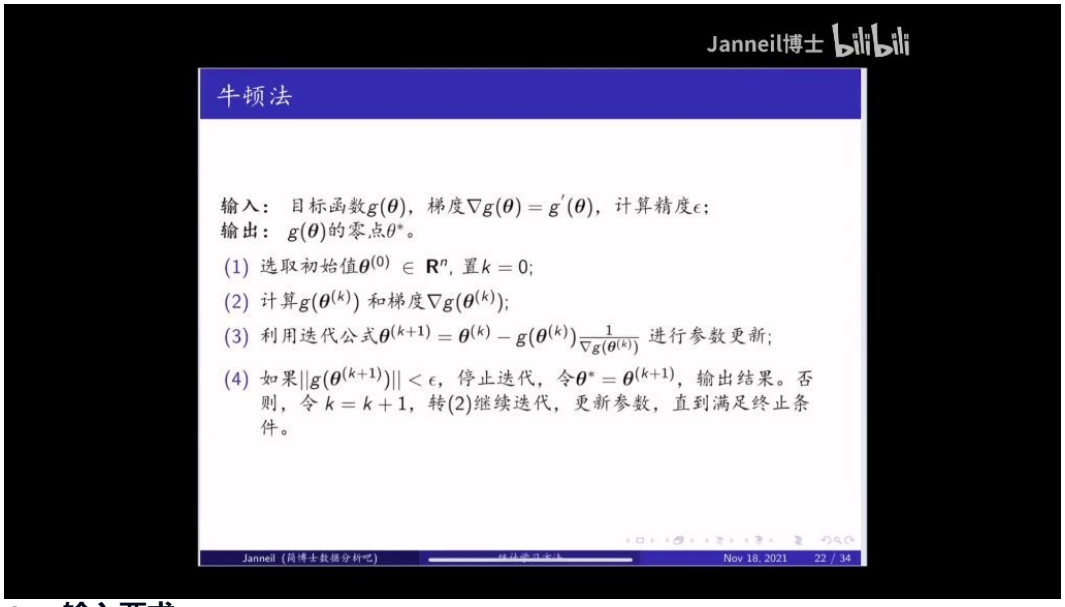
-



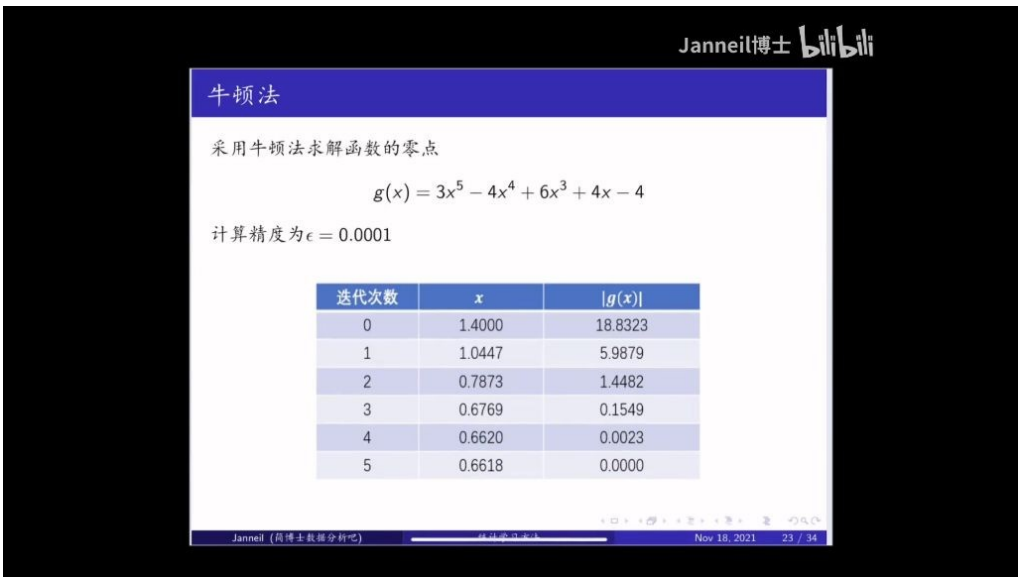
- **推导过程**:
  - 在点$(x^{(k)}, g(x^{(k)}))$处作切线：$y=g(x^{(k)})+g'(x^{(k)})(x-x^{(k)})$
  - 求切线与 x 轴交点：$x^{(k+1)}¿x^{(k)}-\frac{g(x^{(k)})}{g'(x^{(k)})}$
- **几何解释**: 通过不断用切线逼近函数零点

2）算法流程 03:29:53

-



- **输入要求**:
  - 目标函数$g(\theta)$
  - 梯度函数$\nabla g(\theta)$
  - 计算精度ϵ
- **算法步骤**:
  - 选择初始值$\theta^{(0)}$，设$k=0$
  - 计算$g(\theta^{(k)})$和$\nabla g(\theta^{(k)})$
  - 更新参数：$\theta^{(k+1)}=\theta^{(k)}-\frac{g(\theta^{(k)})}{\nabla g(\theta^{(k)})}$
  - 若¿$g(\theta^{(k+1)})\vee$¿ϵ则停止，否则$k=k+1$返回步骤 2
- **实例演示**:
  -



  - 函数：$g(x)$¿$3x^5-4x^4+6x^3+4x-4$
  - 初始值：$x_0=1.4$

迭代过程：

迭代次数	x 值	|g(x)|

0	1.4000	18.8323
1	1.0447	5.9879
2	0.7873	1.4482
3	0.6769	0.1549
4	0.6620	0.0023

    - | 5 | 0.6618 | 0.0000 |
    - 最终结果：$x^{*}=0.6618$（精度$\epsilon=0.0001$）

3. 极值点求解 03:36:18

1）一阶导条件

- 牛顿法求极小值原理 03:36:21
    -
    - 核心思想：将求极值问题转化为求导函数零点问题。对于可微函数 f(x)，其极值点满足$f'(x)=0$，这正是牛顿法求解方程根的适用场景。
    - 数学转换：若要求 f(x)的极小值，只需解方程$g(x)=f'(x)=0$，直接套用牛顿法求根公式。
- 一阶导函数求极值点 03:37:12
    - **求解步骤**：
        - 对目标函数 f(x)求一阶导函数$g(x)=f'(x)$
        - 解方程$g(x)=0$得到临界点
        - 通过二阶导数判断极值性质（极小/极大）
    - 示例函数：以$f(x)=\frac{1}{4}x^4-\frac{1}{8}x$为例，其一阶导$g(x)=x^3-\frac{1}{8}$
- 数学家灭火小笑话 03:38:04
    - **类比说明**：通过数学家应聘消防员的笑话，形象说明"将新问题转化为已解决问题"的数学思维：
        - 已解决问题：灭火（对应求方程根）
        - 新：未着火（求极）
        - 解决方案：点燃货栈再灭火（对应将求极值转化为求导函数零点）

2）二阶导矩阵 03:40:15

- 求极值转化为求零点问题 03:40:22

-

迭代公式推：将牛顿法求根公式中的g(x)替换为$f'(x)$，$g'(x)$替换为$f''(x)$，得到极值点迭代公式：

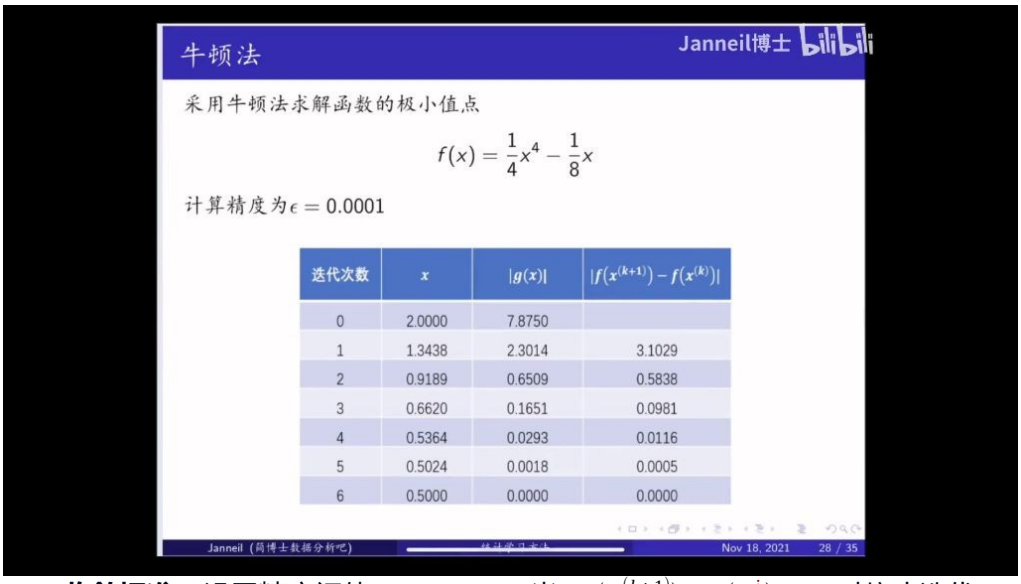
- $x^{(k+1)}=x^{(k)}-\frac{f'(x^{(k)})}{f''(x^{(k)})}$

- 求极小值的迭代公式推导 03:41:30
  - **具体实现**：
    - 计算一阶导数$g(x)=f'(x)¿x^3-\frac{1}{8}$
    - 计算二阶导数$g'(x)=f''(x)¿3x^2$
    - 代入迭代公式：$x^{(k+1)}=x^{(k)}-\frac{(x^{(k)})^3-1/8}{3(x^{(k)})^2}$
- 牛顿法求极小值的算法流程 03:42:54
  - **算法步骤**：
    - 输入目标函数f(x)
    - 计算一阶导（梯度）和二阶导
    - 选择初始点$x^{(0)}$
    - 迭代更新参数直至收敛
    - 输出极小值点$x^{¿}$

3）多元推广 03:47:19

- 牛顿法简介与基础概念
  - **推广思路**：将一元情况推广到n元函数$f(x)$，其中$x=[x_1,x_2,\dots,x_n]^T$
- 牛顿法求解函数的极小值点 03:48:31

○



- **收敛标准**：设置精度阈值$\epsilon=0.0001$，当¿$f(x^{(k+1)})-f(x^{¿})\vee$¿$\epsilon$时停止迭代
- 迭代示例：初始值$x^{(0)}=2.0$，经过 6 次迭代收敛到$x^{¿}=0.5$

● 多元情况下的牛顿迭代法 03:49:10

向量形式：迭代公式推广为：

$x^{(k+1)}=x^{(k)}-[Hf(x^{(k)})]^{-1}\nabla f(x^{(k)})$

- 其中∇f为梯度向量，Hf为 Hessian 矩阵

● 梯度向量与海森矩阵的计算 03:51:22

梯度向量：

- $\nabla f=\left[\frac{\partial f}{\partial x_1},\frac{\partial f}{\partial x_2},\ldots,\frac{\partial f}{\partial x_n}\right]^T$

Hessian 矩阵：

- $[Hf]_{ij}=\frac{\partial^2 f}{\partial x_i \partial x_j}$

● 例题 1:牛顿迭代法的应用 03:55:35

- **计算过程**：
  - 计算梯度向量各分量
  - 构造 Hessian 矩阵各元素
  - 进行矩阵求逆运算
  - 迭代直至收敛（示例中 7 次迭代后差值<0.0001）
- **结果输出**：得到极小值点$x^{¿}=[0.4923,-0.3643]^T$

● 牛顿法的局限性与拟牛顿法简介 03:57:18

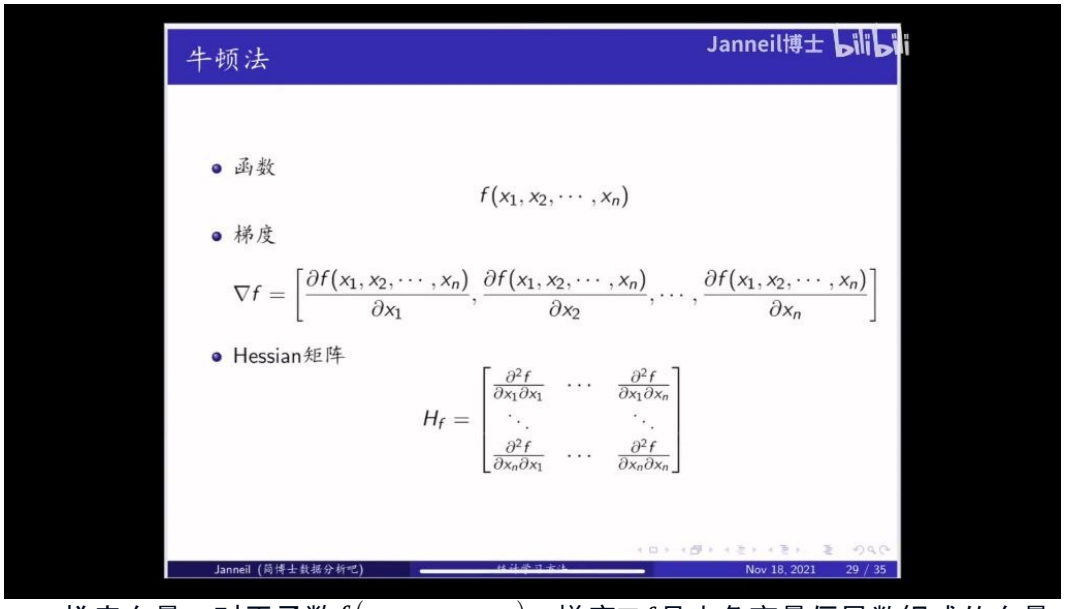
- **主要局限**：
  - 需要计算 Hessian 矩阵及其逆矩阵
  - 高维情况下计算复杂度显著增加
- **改进方向**：拟牛顿法通过近似 Hessian 矩阵来降低计算复杂度

七、拟牛顿法 03:57:29

1. 牛法局限性 03:58:08

1）梯度向量与 Hessian 矩阵介绍 03:58:13

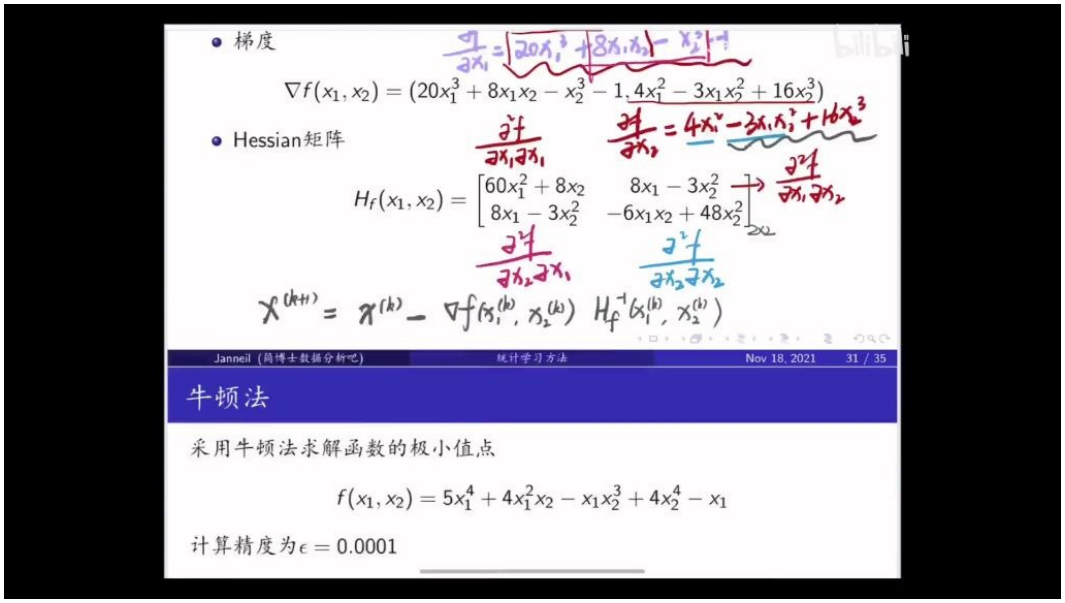
●



- 梯度向量：对于函数$f(x_1, x_2, \cdots, x_n)$，梯度∇f是由各变量偏导数组成的向量，即$\nabla f = \left[\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_n}\right]^T$，在迭代点$x^{(k)}$处的梯度记作$g_k$。
- **Hessian 矩阵**：是二阶偏导数组成的对称矩阵$H_k = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]$，在优化问题中用于描述曲率信息。

2）Hessian 矩阵的逆的求解复杂 03:59:26

●



- **计算复杂度**：当 Hessian 矩阵稠密时，求逆运算时间复杂度高达$O(n^3)$，计算量随维度增加呈指数增长。
- **正定性问题**：Hessian 矩阵可能非正定（行列式为零），此时逆矩阵不存在，如同数字"0"没有倒数。
- **实际案例**：以函数$f(x_1, x_2)$¿$5x_1^4 + 4x_1^2x_2 - x_1x_2^3 + 4x_2^4 - x_1$为例，其 Hessian 矩阵在迭代过程中可能失去正定性。

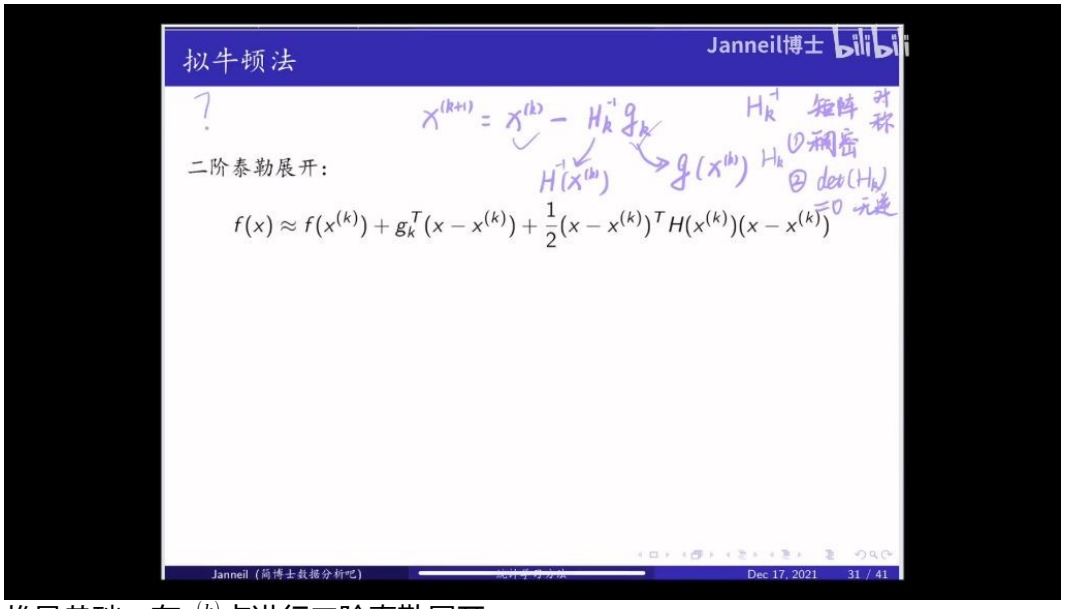
3）拟牛顿法简介 04:01:03

- 核心思想：构造 Hessian 矩阵（或其逆矩阵）的近似矩阵，避免直接计算二阶导数。

- **优势对比**：
  - 牛顿法：需要精确计算H_k^{-1}，计算成本高
  - 拟牛顿法：通过迭代更新近似矩阵，计算效率更高

2. 牛条件 04:01:25

1）泰勒展开

- 

推导基础：在$x^{(k)}$点进行二阶泰勒展开：

- $f(x)\approx f(x^{(k)})+g_k^T(x-x^{(k)})+\frac{1}{2}(x-x^{(k)})^T H_k(x-x^{(k)})$

求：对展开式两边求导得近似关系：

- $g(x)\approx g_k+H_k(x-x^{(k)})$

2）两种形式 04:08:30

基本形式（Hessian 版）：

$y_k ¿ H_k\delta_k$

- 其中$y_k=g_{k+1}-g_k$，$\delta_k=x^{(k+1)}-x^{(k)}$

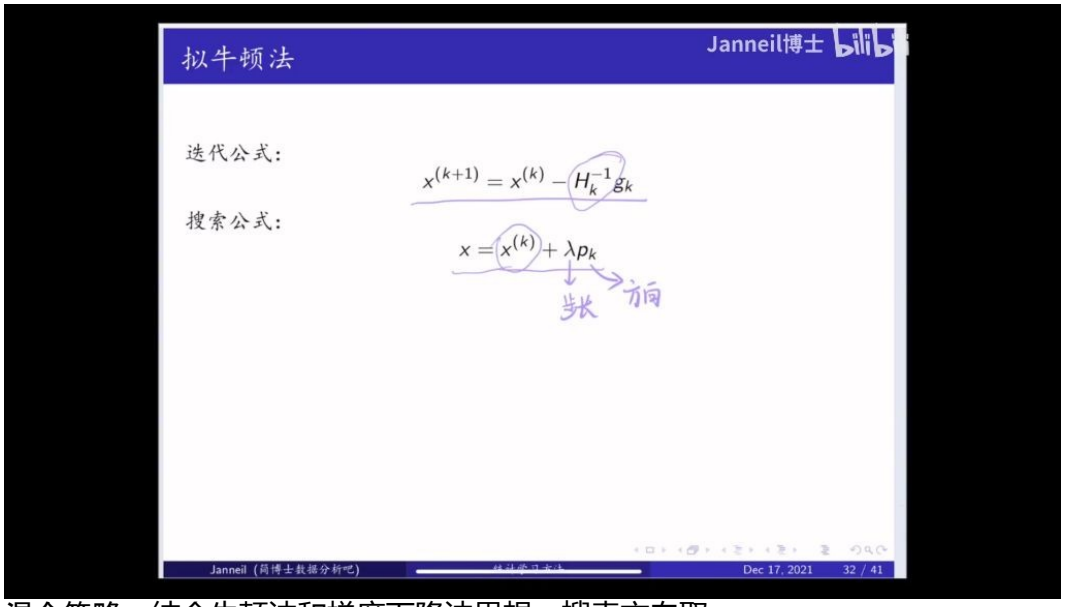
逆矩形式：

- $H_k^{-1}y_k=\delta_k$
- **应用限制**：不能直接求逆，因为y_k是向量（除非一维），且需保证H_k正定。

3. 搜索算法 04:14:41

1）方向确定

●



混合策略：结合牛顿法和梯度下降法思想，搜索方向取：

- p_k ¿$-H_k^{-1} g_k$
- **正定性保证**：初始H_0需取对称正定矩阵（通常取单位阵），确保搜索方向使函数值下降。

2）步长选择 04:15:04

一阶展开验证：通过一阶泰勒展开证明：

$f\left(x^{(k+1)}\right) \approx f\left(x^{(k)}\right) - \lambda\, g_k^T H_k^{-1} g_k$

- 当H_k^{-1}正定时，右边第二项为负，保证函数值下降。
- **精确线搜索**：采用最速下降法思想，通过优化λ使$f\left(x^{(k)} + \lambda\, p_k\right)$最小化。

3）DFP 算法

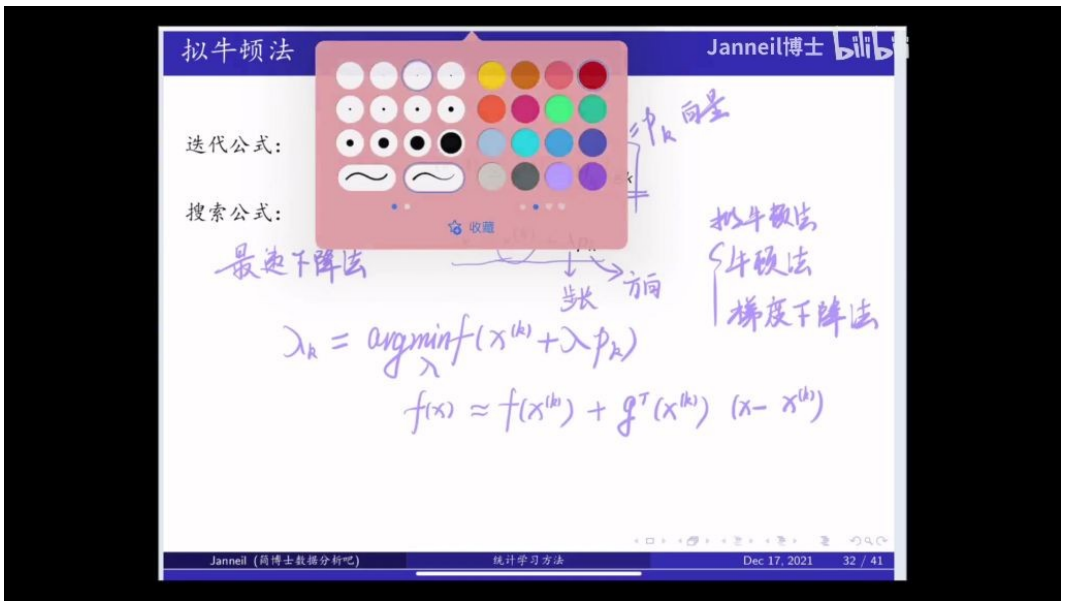
- **实现方式**：基于逆矩阵形式的拟牛顿条件，直接更新H_k^{-1}的近似矩阵。
- **名称说明**：DFP 是三位发明者姓氏首字母缩写，本身无特殊含义。

八、拟牛顿法 04:19:46

1. DFP 算法

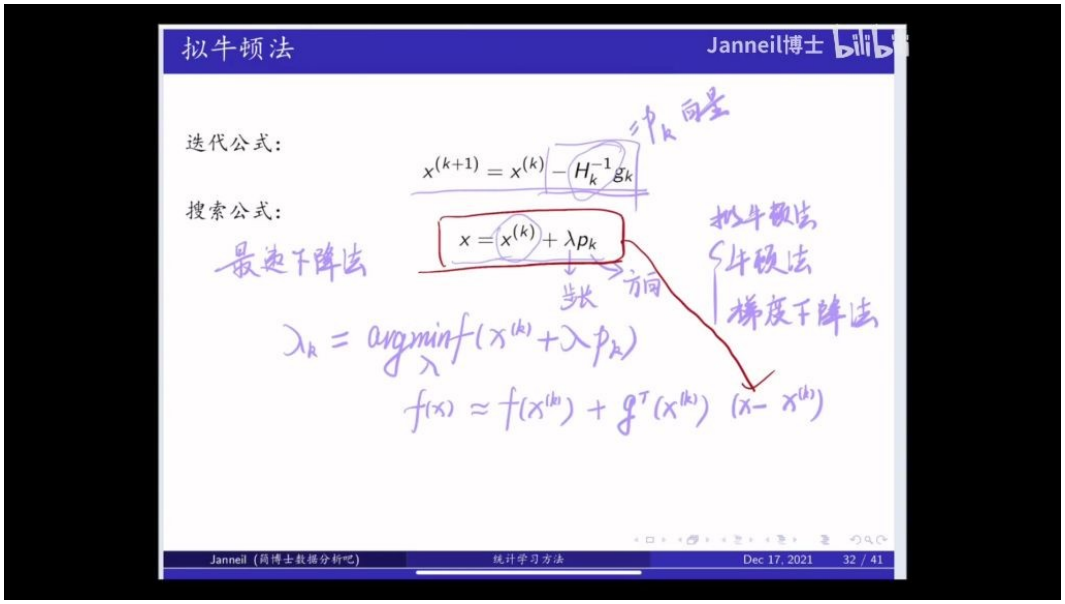
1）算法名称由来

●



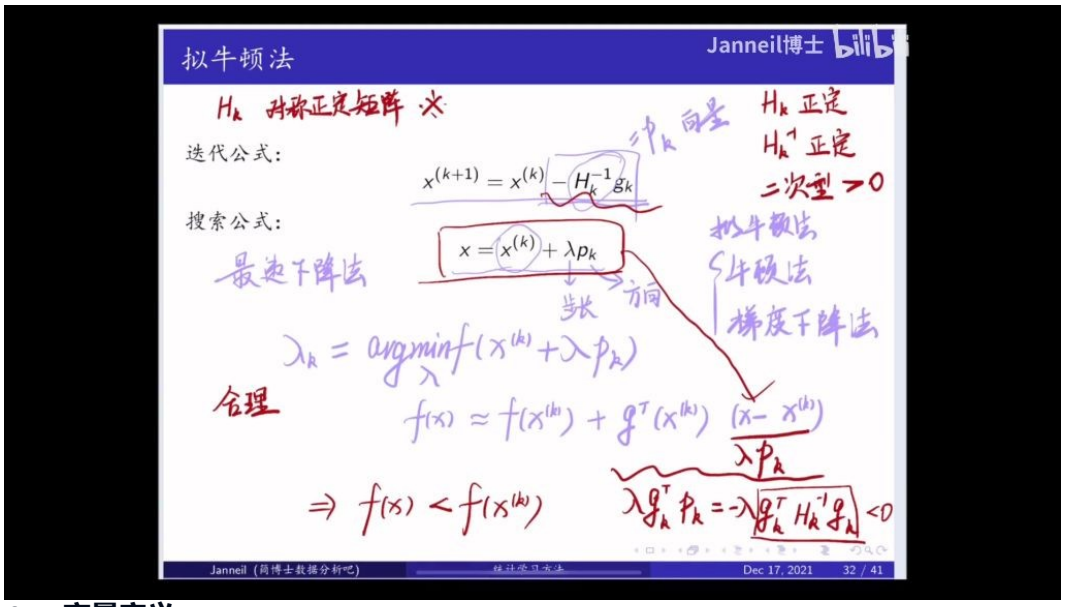
- **名称构成**：由三位学者姓氏首字母组成，D 代表 Dividend（股息），F 代表 Flash（闪存），P 代表 Power（功率）
- **背景说明**：该算法由这三位学者共同提出，因此采用其姓氏首字母命名

2）逆牛顿条件 04:20:21

- 

- 核心思想：直接构造海森矩阵的逆矩阵$G = H^{-1}$的迭代公式
- 迭代形式：$G_{k+1} \leftarrow G_k + P_k + Q_k$，其中$P_k$和$Q_k$为附加修正项
- 数学保证：要求迭代过程中保持矩阵的对称正定性

3）迭代公式 04:20:54

- 

- **变量定义**：
  - $\delta_k \leftarrow x^{(k+1)} - x^{(k)}$：迭代点差值
  - $y_k \leftarrow g_{k+1} - g_k$：梯度差值
- **关键条件**：$G_{k+1} y_k = \delta_k$（拟牛顿条件）
- 修正构造：
  - $P_k = \frac{\delta_k \delta_k^T}{\delta_k^T y_k}$

- $Q_k=\frac{-G_k y_k y_k^T G_k}{y_k^T G_k y_k}$

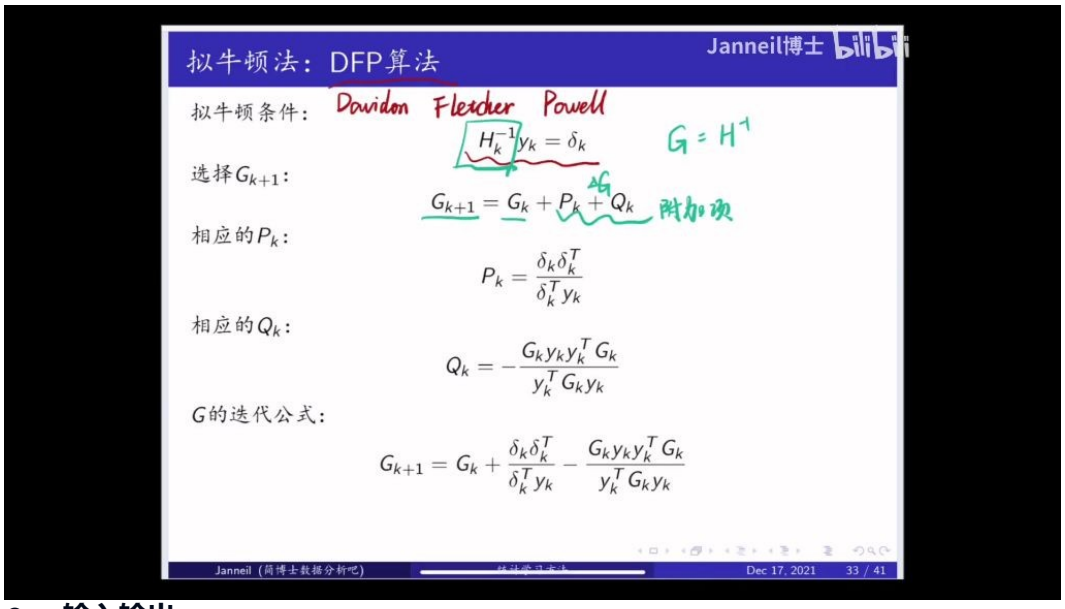
4）拟牛顿条件 04:22:35

- 数学本质：通过构造附加项使迭代矩阵满足$G_{k+1}y_k=\delta_k$
- 构造技巧：
  - 令$P_k y_k=\delta_k$
  - 令$Q_k y_k¿-G_k y_k$
- **对称性保持**：确保P_k和Q_k均为对称矩阵以维持G_k的对称性

5）矩阵变换 04:22:51

- **验证方法**：
  - 对P_k验证：利用结合律$(\delta_k \delta_k^T) y_k¿\delta_k(\delta_k^T y_k)$，分母为标量可约简
  - 对Q_k验证：$(y_k^T G_k y_k)$结果为标量，分子$G_k y_k y_k^T G_k y_k$可简化
- **非唯一性**：满足拟牛顿条件的修正项不唯一，DFP 选择的形式计算简便

6）算法步骤 04:27:30

- 

- **输入输出**：
  - 输入：目标函数 f(x)，梯度向量 g(x)，计算精度ϵ
  - 输出：极小值点$x^¿$
- **实施步骤**：
  - 初始化：选择初始点x_0和初始矩阵G_0（通常取单位矩阵）
  - 计算梯度：$g_k=\nabla f(x^{(k)})$
  - 收判断：若$¿g_k \vee ¿\epsilon$则停止，否则继续
  - 确定搜索方向：$p_k¿-G_k g_k$
  - 一维搜索：求最优步长
  - 更新迭代点：$x^{(k+1)}¿x^{(k)}+\lambda_k p_k$
  - 更新逆矩阵：按 DFP 公式计算G_{k+1}
  - 返回步骤 2 继续迭代
- 特殊**说**明：该算法被称为阻尼牛顿法，结合了牛顿法和最速下降法的优点

2. BFGS 算法 04:31:58

1）算法名称由来 04:33:16

- 命名来源: 由四位数学家姓氏首字母组成：Broyden、Fletcher、Goldfarb、Shanno。与 DFP 算法类似，BFGS 本身没有实际含义。
- **人物关系**:

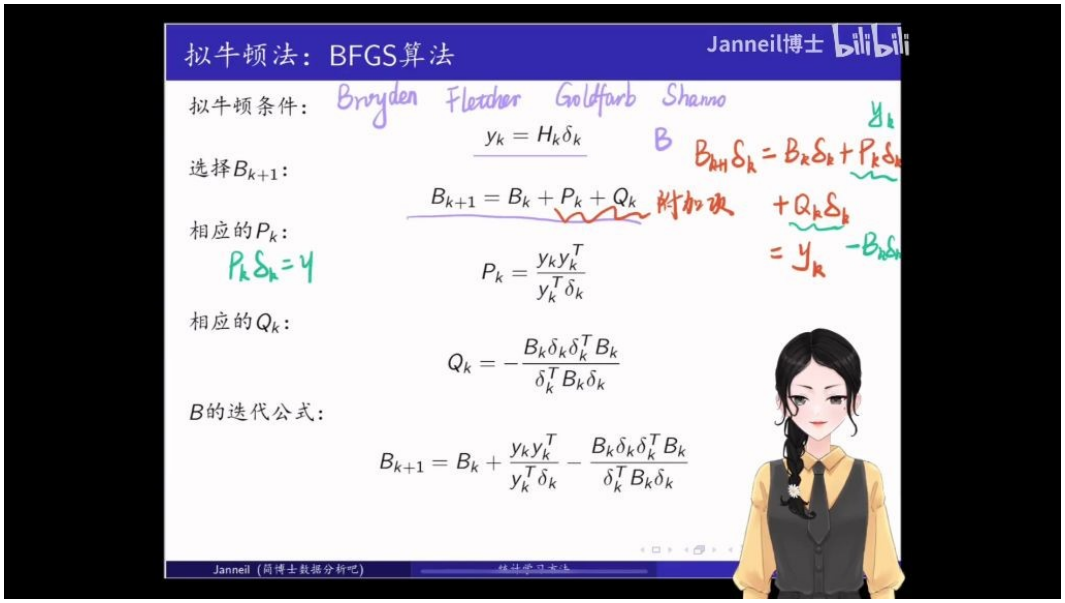
- Fletcher 是 DFP 算法中的"F"
- Goldfarb 是新引入的数学家
- Shanno 是第四位贡献者

2）拟牛顿条件 04:33:46

- 核心条件:$y_k ¿ H_k \delta_k$，但 BFGS 直接对矩阵B_k进行更新而非其逆矩阵
- **矩阵选择**: 将更新矩阵记作 B，构造迭代公式$B_{k+1} ¿ B_k + P_k + Q_k$
- 构造原理:
  - 在等式两边同时乘以δ_k保符合牛条件
  - 通过$P_k \delta_k ¿ y_k$和$Q_k \delta_k ¿ - B_k \delta_k$确定附加项

3）迭代公式 04:34:46

-



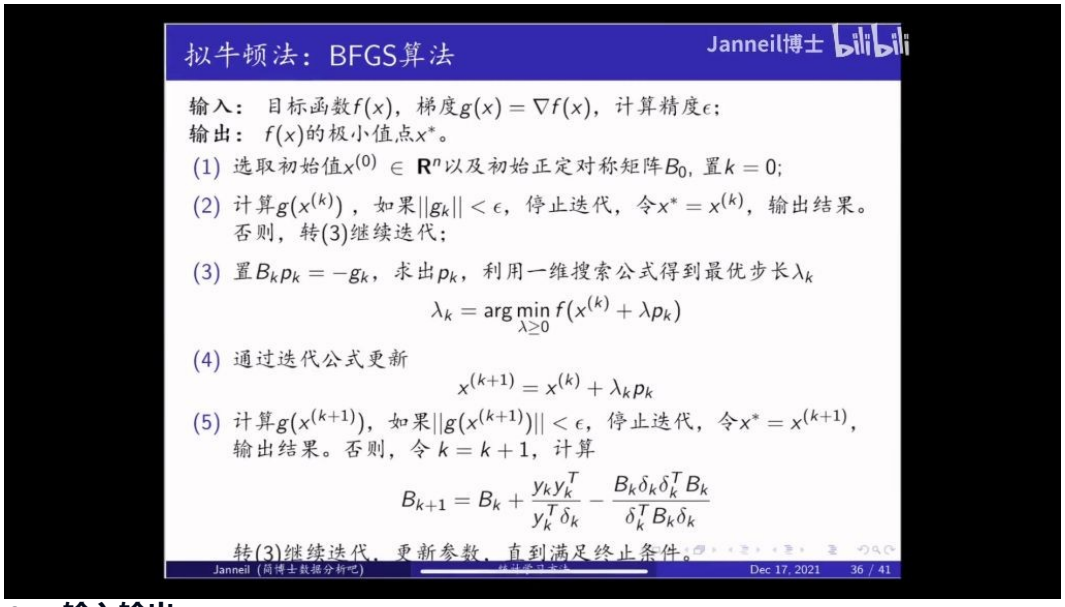
- **P 项构造**:$P_k = \frac{y_k y_k^T}{y_k^T \delta_k}$
- **Q 项构造**:$Q_k = \frac{-B_k \delta_k \delta_k^T B_k}{\delta_k^T B_k \delta_k}$

完整公式:

- $B_{k+1} ¿ B_k + \frac{y_k y_k^T}{y_k^T \delta_k} - \frac{B_k \delta_k \delta_k^T B_k}{\delta_k^T B_k \delta_k}$

4）算法步骤 04:37:06

●

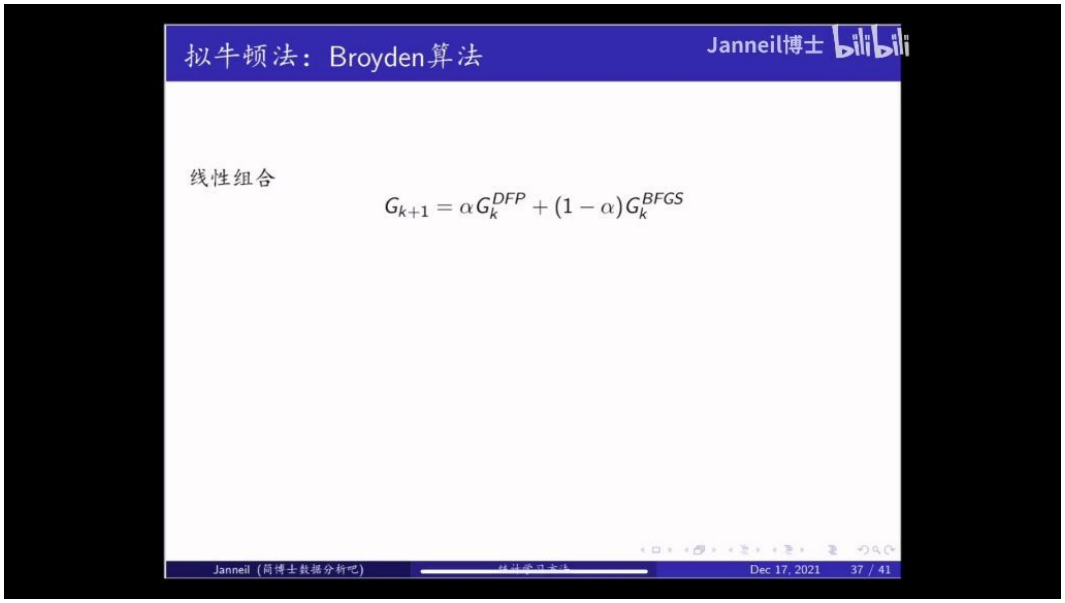


- **输入输出**:
  - 输入：目标函数 f(x)，梯度$g(x)=\nabla f(x)$，计算精度ϵ
  - 输出：极小值点$x^{¿}$
- **具体步骤**:
  - 初始化$x^{(0)}$和正定对称矩阵B_0，设 k=0
  - 计算梯度$g(x^{(k)})$，若¿$g_k \vee$ ¿ϵ则停止
  - 解方程$B_k p_k$¿$-g_k$求方向p_k，进行一维搜索得步长λ_k
  - 更新$x^{(k+1)}$¿$x^{(k)}+\lambda_k p_k$
  - 计算新梯度，若未收敛则更新B_{k+1}后继续迭代
- **关键区别**:
  - 直接操作B_k矩阵而非其逆
  - 方向p_k通过解线性方程组获得
  - 矩阵更新公式与 DFP 不同但结构相似

3. Broyden 算法 04:38:57

1）线性组合 04:39:00

●



基本思想: 将 DFP 和 BFGS 的结果进行线性组合

- $G_{k+1}=\alpha G_k^{DFP}+(1-\alpha)G_k^{BFGS}$
- **参数意义**:
  - $\alpha=0$时为 BFGS 算法
  - $\alpha=1$时为 DFP 算法
- **类比案例**: 类似弹性网(Elastic Net)结合 L1 和 L2 正则项的思想

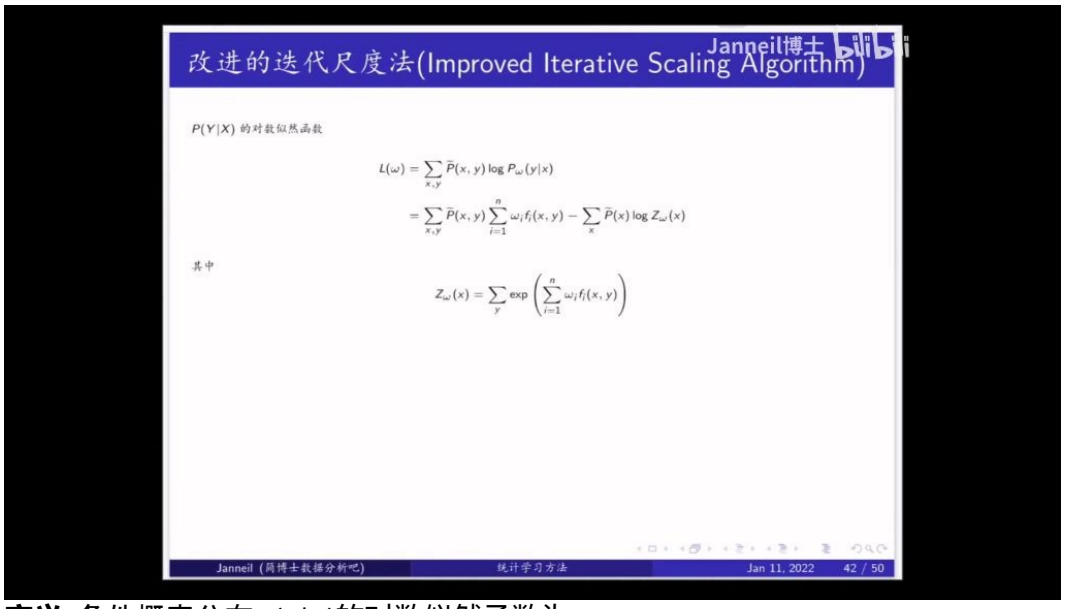
2）算法特点 04:40:11

- **折中策略**: 通过参数α在两种算法间取得平衡
- **理论保证**: 线性组合后的矩阵仍满足拟牛顿条件
- **应用优势**: 可根据问题特性调整算法偏向性

九、最大熵模型 04:45:44

1. 改迭代尺度法 04:45:49

1）对数似然函数 04:46:05

- 

定义: 条件概率分布 P(Y|X)的对数似然函数为：

- 其中$Z_\omega(x)=\sum_y\exp\left(\sum_{i=1}^{n}\omega_i f_i(x,y)\right)$
- 对数似然函数的求解方法 04:47:32
  - **对数化处理**: 将似然函数对数化，将乘法运算转化为加法运算
  - **参数识别**: 经验分布和特征函数f_i已知，唯一未知的是参数向量$\omega=(\omega_1,\omega_2,\ldots,\omega_n)$
  - **极值求解**:
    - **求导法**: 对$L(\omega)$求导并令导数为零
    - **优化算法**: 可采用梯度下降法、拟牛顿法等优化方法
    - 迭代法: 直接迭代计算，保证每次迭代似然值增大
- 直接迭代计算 04:49:10
  - **基本思想**: 通过迭代使似然函数值逐步增大，接近最大值
  - **参数更新**: 在现有参数ω基础上寻找增量δ，使得$L(\omega+\delta)\geq L(\omega)$
  - 与 **EM 算法关系**: 方法思路与 EM 算法相近，都是通过迭代逐步优化

2）算法名称 04:49:24

-

- **全称**: 改的迭代尺度法(Improved Iterative Scaling Algorithm)
- **简称**: IIS
- **背景**: 20 世纪 90 年代由 IBM 机器翻译小组提出，用于解决参数估计问题

3）参数更新 04:50:37

- **增量计算**: 寻找增量δ使得$L(\omega+\delta)-L(\omega)\geq 0$

差表达式:

-

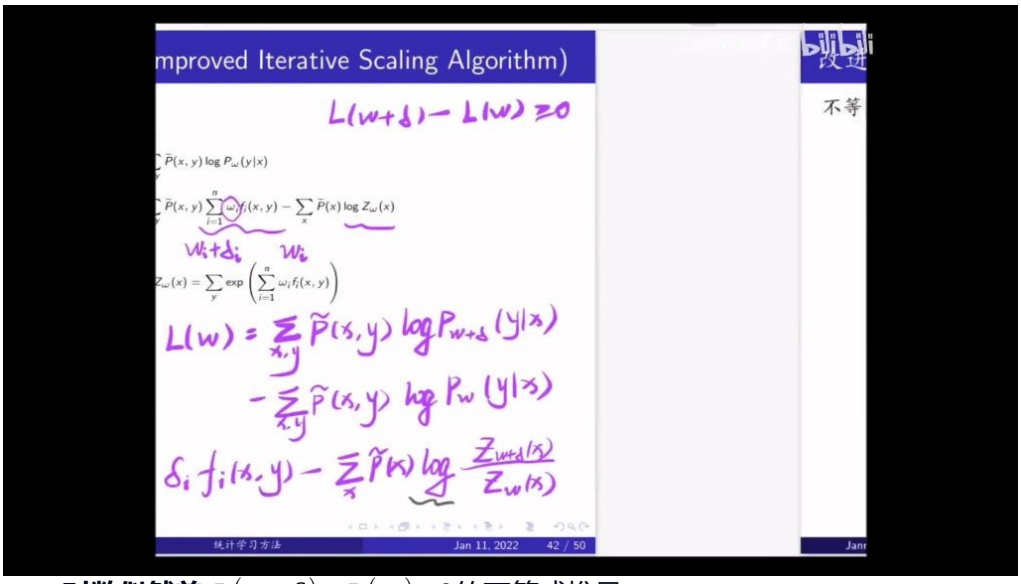
- **对数处理难点**: 表达式中的对数项使计算复杂

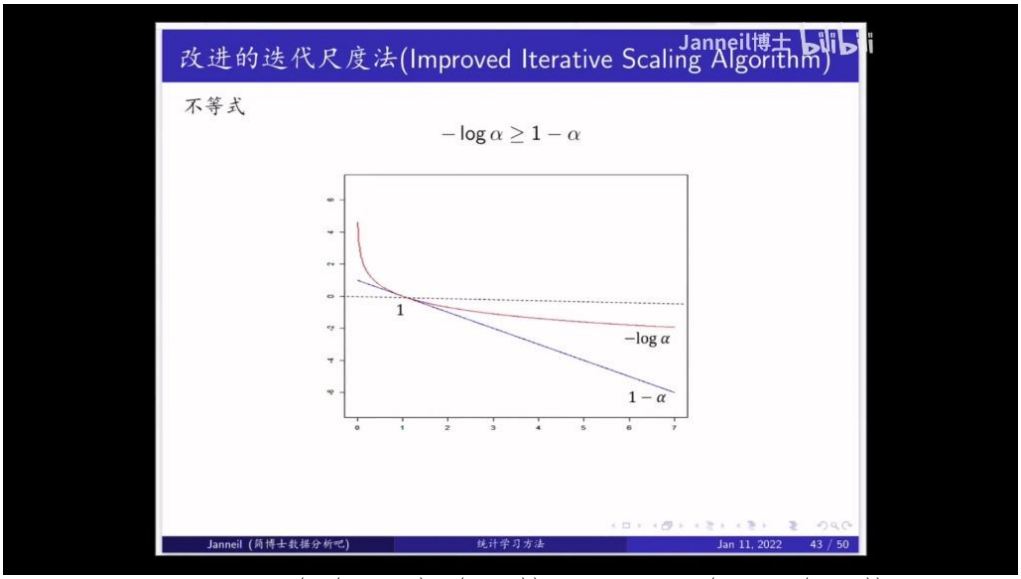
4）不等式应用 04:53:17

- 例题 1:不等式的推导与应用 04:53:20
  - **关键不等式**:$-log\alpha \geq 1-\alpha(\alpha>0)$
  - 明方法:
    - **图像法**: 制$-log\alpha$和$1-\alpha$函数图像，直观比较
    - 函数分析法: 定义$\varphi(\alpha)=-log\alpha-(1-\alpha)$，通过求导分析极值
      - 导数:$\varphi'(\alpha)=\frac{\alpha-1}{\alpha}$
      - 极值点:$\alpha=1$时为最小值点，$\varphi(1)=0$
      - 结论:$\varphi(\alpha)\geq 0$对所有$\alpha>0$成立
  - **应用**: 利用该不等式可以找到对数似然函数差值的下界，简化计算

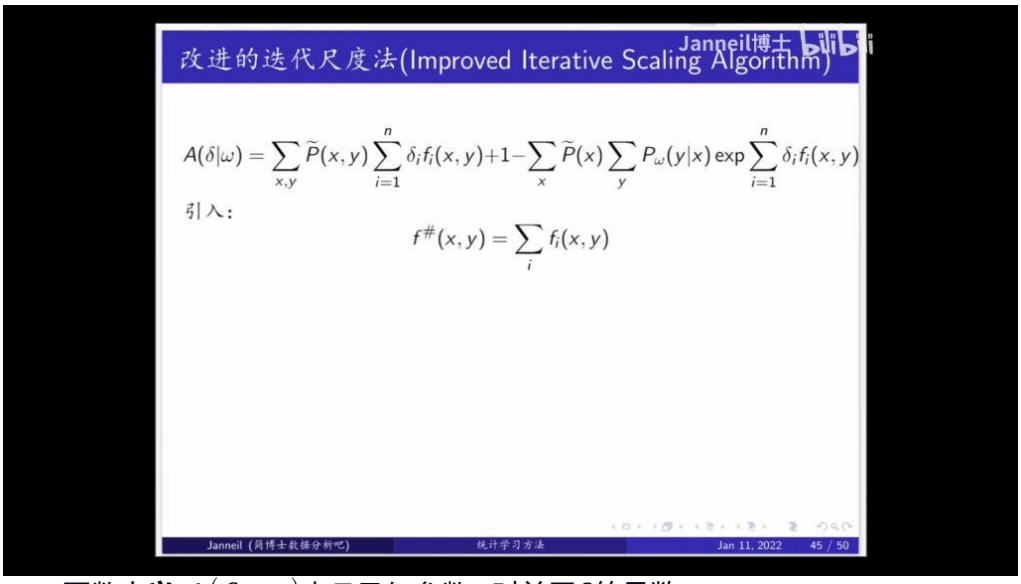
5）下界推导 04:59:36

- 差值下界推导与改写

-


  - **对数似然差**:$L(\omega+\delta)-L(\omega)\geq 0$的不等式推导
  - 初始表达式:$\overline{P}(x,y)\log P_\omega(y\vee x)$和$\overline{P}(x,y)\sum_{i=1}^{n}(\omega,y)f_i(x,y)$
  - **改写技巧**: 使用不等式$-log\alpha\geq 1-\alpha$将对数项转化为线性表达式
- 差值下界推导：取 x 经验分布 05:00:28
  - **经验分布处理**: 当考虑 x 的所有可能时，经验分布总和为 1
  - **变形结果**: 表达式变为 1 减去$Z_{\omega+\delta}(x)¿\,Z_\omega(x)$的比值
- 差值下界推导：求比值 05:01:25
  -


  - **比值计算**: 分子$\exp\left(\sum(\omega_i+\delta_i)f_i(x,y)\right)$，分母为$\exp\left(\sum\omega_i f_i(x,y)\right)$
  - **常见误区**: 不能直接简化为δ_i，因为存在求和符号
- 差值下界推导：分子与分母的确定 05:02:03
  - **分解技巧**: 将分子分解为ω_i和δ_i两部分
  - **条件概率出现**: 分母部分可表示为$Z_\omega(x)$，与条件概率$P_\omega(y\vee x)$相关
- 差值下界推导：条件概率分布与下界表达 05:03:14
  - **最终表达式**: 包含三项：经验分布项、常数 1 和条件概率项
  - **关键转换**: 使用条件概率$P_\omega(y\vee x)$重新表示下界
- 差值下界推导：新表达式与函数定义 05:05:00

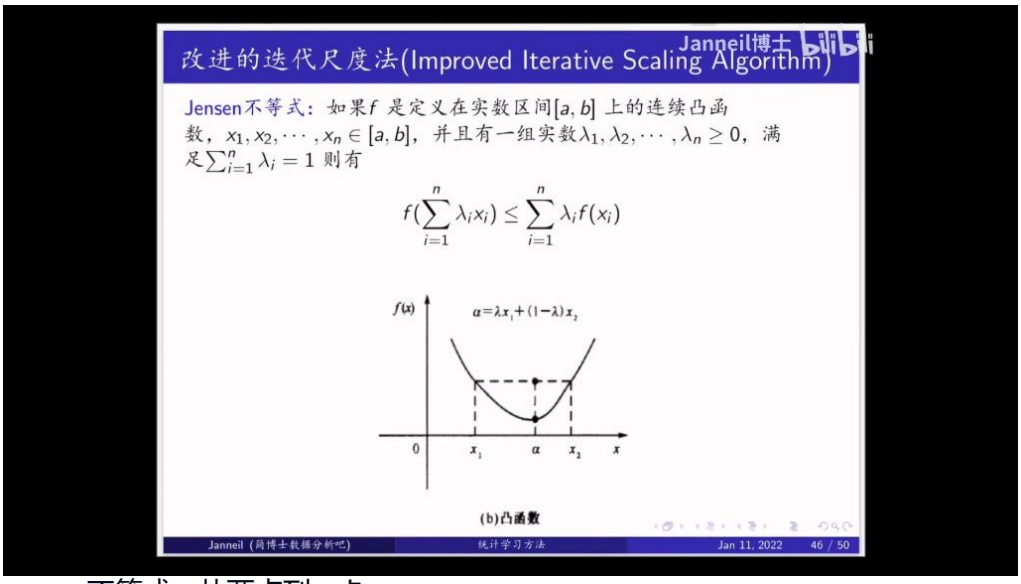
- 

  - 函数定**义**:$A(\delta\vee\omega)$表示已知参数ω时关于δ的函数
  - **组成部分**: 包含经验分布、参数增量δ_i和特征函数$f_i(x,y)$
- 例题 1：找到合适的德尔塔 05:05:46
  - **目标**: 寻找使$A(\delta\vee\omega)$>0的δ
  - 直方法: 尝试对 A 函数求偏导数
- 例题 1 解析：对 a 求偏导数 05:06:20
  - **偏导结果**: 第一项保留$f_i(x,y)$，第三项保留指数部分和f_i
  - **问题发现**: 偏导数包含δ的所有元素，无法独立求解
- 例题 1 总结：直接求偏导不合适 05:08:19
  - **结论**: 直接求导方法不可行，需要寻找新的下界
- 例题 2：对 a 函数找到下界 05:08:31
  - 解决思路: 对 A 函数再寻找一个下界
  - **关键引入**: 特征总数$f^{\#}(x,y)=\sum_i f_i(x,y)$
- 特征函数与特征的个数 05:08:58
  - 特征性质: 二特征函数（0 或 1）
  - **归一化处理**:$f_i/f^{\#}$可视为概率分布
  - **应用优势**: 为引入 Jensen 不等式创造条件
- json 不等式 05:10:46
  - **关键工具**: Jensen 不等式将用于构建新的下界
  - **应用前提**: 需要满足凸函数和概率分布的条件

6）Jensen 不等式 05:11:02

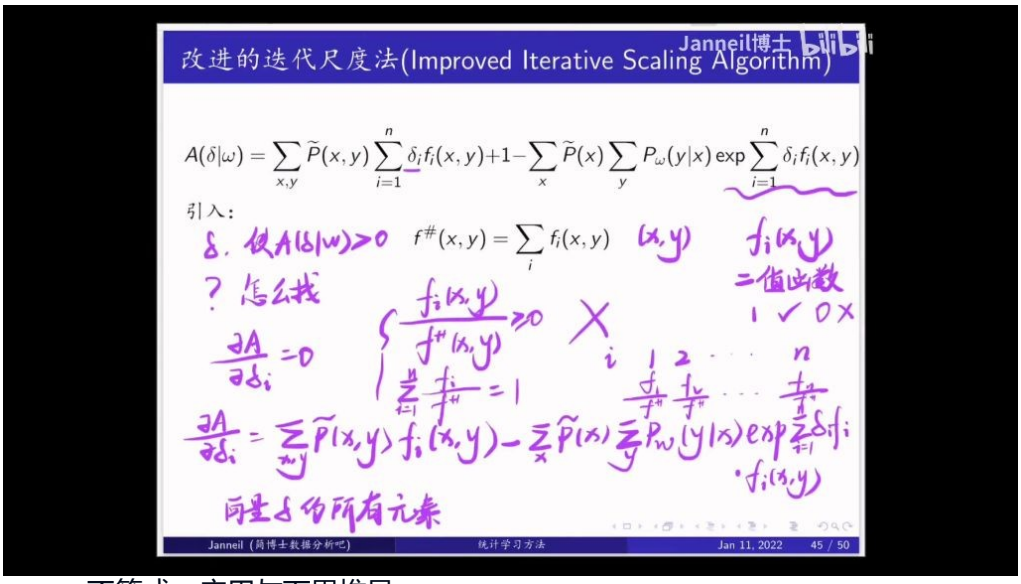
- 改进的迭代尺度法与 EM 算法 05:11:04
  - **算法关联性**：改进的迭代尺度法（Improved Iterative Scaling Algorithm）与 EM 算法思想非常接近，都利用了 Jensen 不等式作为核心数学工具。
  - **应用广泛性**：该不等式在机器学习和人工智能领域应用广泛，不仅用于改进的迭代尺度法，还应用于 EM 算法等其他算法。
- Jensen 不等式与凸函数 05:11:27
  - 函数**类**型区分：注意区分不等式中的普通函数 f 与特征函数f_i，前者是任意凸函数，后者是特定场景下的二值特征函数。
  - 凸函数定**义**：函数 f 在区间$[a,b]$上连续且为凸函数，这是应用 Jensen 不等式的前提条件。

-

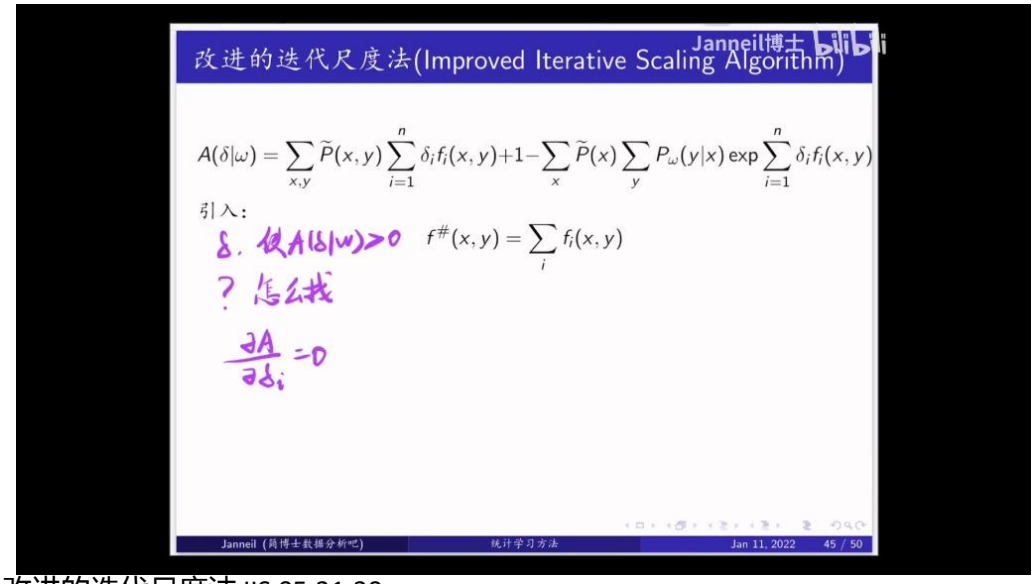


- Jensen 不等式：从两点到 n 点 05:12:39
  - 两点基础：对于凸函数，若取两点x_1, x_2和权重$\lambda \in [0,1]$，有$f(\lambda x_1+(1-\lambda)x_2) \leq \lambda f(x_1)+(1-\lambda)f(x_2)$。
  - n 点推广：推广到 n 个点$x_1, \ldots, x_n$和权重$\lambda_1, \ldots, \lambda_n \geq 0$且$\sum \lambda_i = 1$时，不等式保持相同形式。
- Jensen 不等式：概率期望形式 05:13:32
  - **概率解释**：将λ_i□□x_i发生的概率，则不等式左边是函数在期望值处的取值，右边是函数值的期望。
  - 不等式核心：$f(E[x]) \leq E[f(x)]$，即凸函数在期望处的值不超过函数值的期望。
- Jensen 不等式：去掉指数 05:15:02
  - **问题背景**：在改进的迭代尺度法中，需要处理包含指数函数$\exp(\sum \delta_i f_i(x,y))$的复杂表达式。
  - 解决思路：利用指数函数的凸性，通过 Jensen 不等式将指数内部的求和移到外部，从而简化计算。
- Jensen 不等式：指数函数与凸函数 05:15:17
  - 函数性**质确认**：指数函数e^x是典型的凸函数，其图像呈现向上凸起的形状，满足 Jensen 不等式的应用条件。
  - **权重构造**：在应用中，权重取为$\frac{f_i(x,y)}{f^{\#}(x,y)}$，其中$f^{\#}(x,y)=\sum_i f_i(x,y)$。

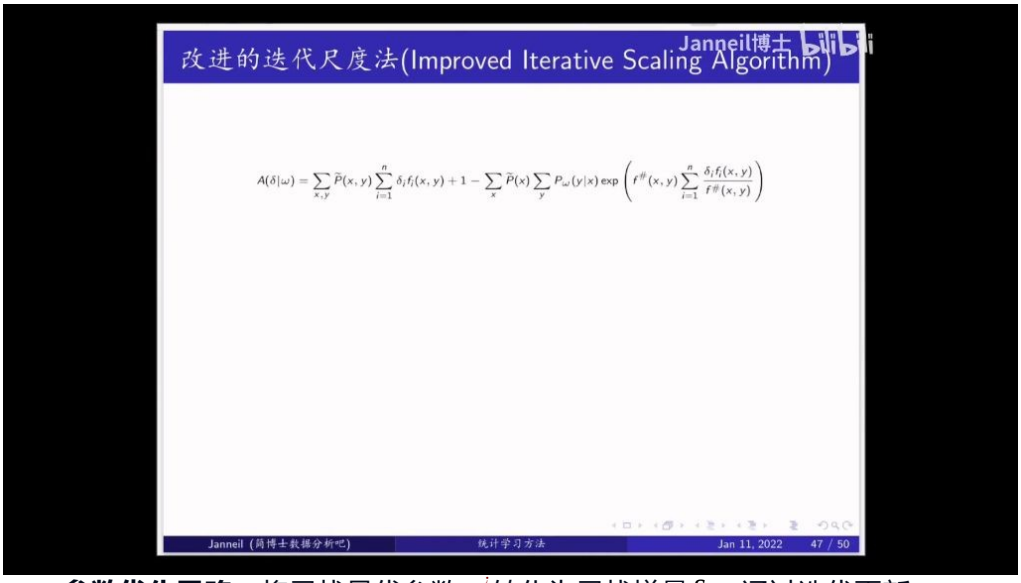
-

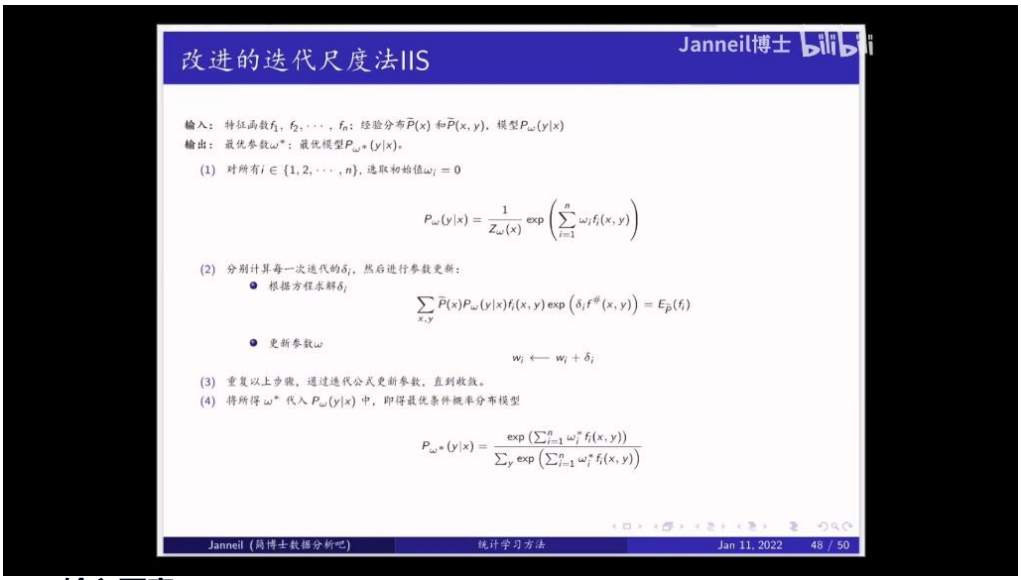


- Jensen 不等式：应用与下界推导 05:16:44
  - 下界构造：将原目标函数$A(\delta\vee\omega)$中的指数部分用 Jensen 不等式替换，得到一个新的下界函数$B(\delta\vee\omega)$。
  - 表达式**转换**：$\exp(\sum\delta_i f_i)\le\sum\frac{f_i}{f^{\#}}\exp(f^{\#}\delta_i)$，这个转换是算法的关键步骤。
- Jensen 不等式：新的下界与偏导 05:18:48
  - **求导简化**：对下界函数 B 求偏导时，由于 Jensen 不等式的应用，使得$\frac{\partial B}{\partial\delta_i}$的表达式中仅含$\delta_i$，而不像原函数那样所有参数耦合。
  - **具体形式**：偏导数包含两项：经验分布下的特征期望 $P(x,y)f_i(x,y)$和模型分布下的调整期望项。
- Jensen 不等式：方程求解与算法 05:20:53
  - 方程求解：令偏导数为零，可以得到关于δ_i的独立方程，避免了原问题中参数相互影响的"牵一发而动全身"的问题。
  - **算法实现**：通过迭代求解这些独立方程，逐步更新参数δ_i，这就是改进的迭代尺度法的核心流程。
  -

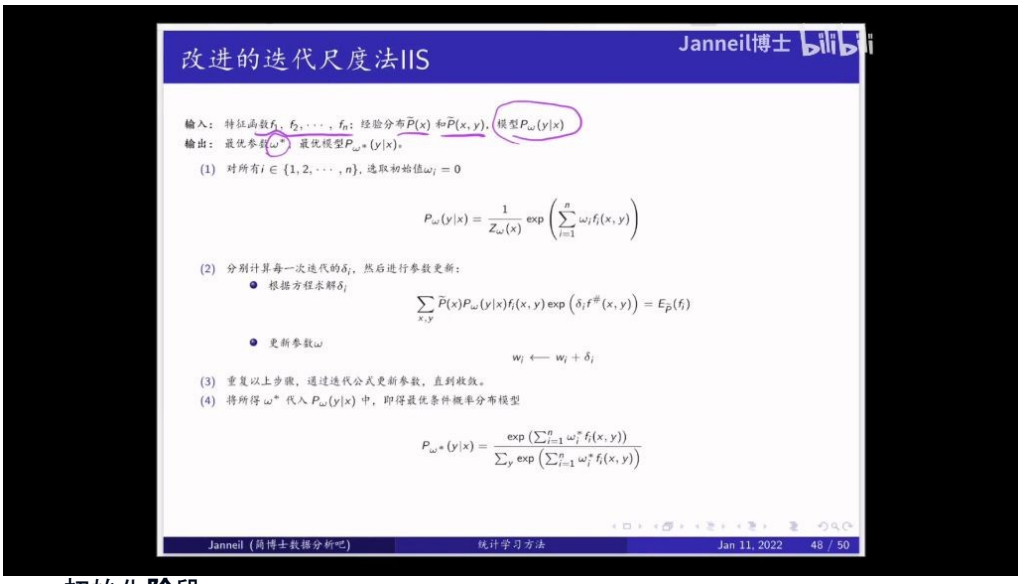


7）改进的迭代尺度法 IIS 05:21:30

- 算法原理 05:21:35
  - 

  - **参数优化思路**：将寻找最优参数$\omega^{¿}$转化为寻找增量δ_i，通过迭代更新$\omega_i \leftarrow \omega_i+\delta_i$逐步逼近最解
  - **目标函数**：$A(\delta\vee\omega)=\sum_{x,y}P(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)+1-\sum_{x}P(x)\sum_{y}P_\omega(y\vee x)\exp\left(f^{¿}(x,y)\sum_{i=1}^{n}\frac{\delta_i f_i(x,y)}{f^{¿}(x,y)}\right)$
  - 数学基础：通过求偏导$\frac{\partial B}{\partial \delta_i}=0$建立参数更新方程
- 算法输入输出 05:21:54
  - 

  - **输入要素**：
    - 特征函数：$f_1, f_2, \ldots, f_n$
    - **经验分布**：$P(x)$和$P(x,y)$
    - 模型**结构**：$P_\omega(y\vee x)=\frac{1}{Z_\omega(x)}\exp\left(\sum_{i=1}^{n}\omega_i f_i(x,y)\right)$
  - **输出结果**：
    - **最优参数**：$\omega^{¿}$

    - **最优模型**：$P_{\omega^{¿}}(y \vee x)$
- 算法步骤详解 05:22:36
  -


    - 初始化**阶**段：
      - **参数设置**：所有$\omega_i = 0$
      - 初始分布特性：此时$P_\omega(y \vee x)$为等概率分布，符合最大熵原则
    - 迭代**过**程：
      - **求解增量**：通过方程$\sum_{x,y} P(x) P_\omega(y \vee x) f_i(x,y) \exp(\delta_i f^{¿}(x,y)) ¿ E_{P}(f_i)$计算$\delta_i$
      - **参数更新**：执行$\omega_i \leftarrow \omega_i + \delta_i$
      - **收敛判断**：重复迭代直到参数收敛
    - **结果输出**：
      - 将最终$\omega^{¿}$代入条件概率模型得到最优分布
- 关键实现细节 05:24:51
  - 初始**值选择**：
    - **零初始化优势**：保证初始模型为等概率分布，符合最大熵原则
    - 数学简化：零初始时分母为 1，分子为 y 的个数
  - **增量计算**：
    - 方程求解：需要解关于δ_i的非线性方程
    - 数**值**方法：通常采用牛顿法等数值计算方法
  - **收敛特性**：
    - **终止条件**：当δ_i变化小于阈值时停止
    - **理论保证**：算法保证收敛到局部最优解

十、知识小结

知识点	核心内容	关公式/原理	应用示例
最大熵原理	在满足所有约束条件下选择熵最大的概率分布	H(p)=-∑p(x)logp(x)	两点分布在 p=0.5 时熵最大
离散变量最大熵分布	多项分布在等概率时熵最大	p_i=1/k	五个取值 abcde 的等概率分布
连续变量最大熵分布	给定均值和方差时正态分布熵最大	N(μ,σ²)	金融数据建模常用正态分布假设
最大熵模型定义	在满足特征函数期望约束的模型集合中选择条件熵最大	H(P)=H(Y|X)	文本分类中利用词性等特征

	的模型		
特征函数	描述输入 x 与输出 y 之间事实的二值函数	f(x,y)∈{0,1}	"打"字前有数词时 f=1
模型训练方法	通过优化算法求解模型参数	梯度下降/牛顿法/拟牛顿法	使用 BFGS 算法迭代更新参数
拟牛顿法	近似海森矩逆的化方法	BFGS/DFP 更新公式	避免直接计算二阶导数
改进迭代尺度法	通过下界逼近优化目标函数	A(δ|ω)≤B(δ|ω)	IBM 机器翻译小组提出