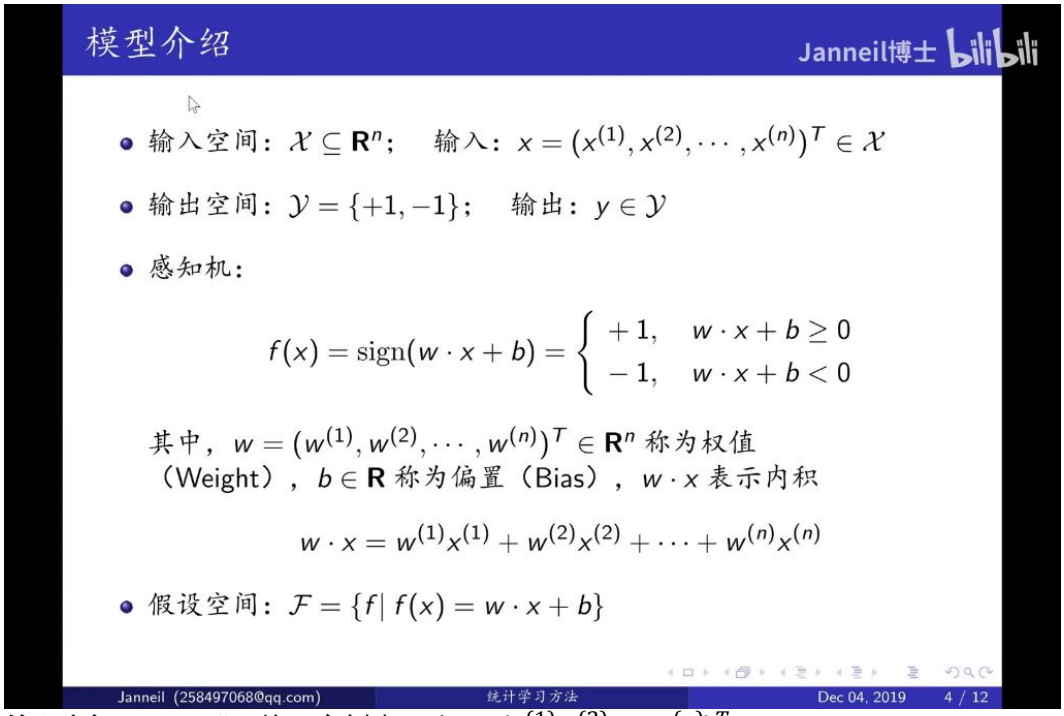
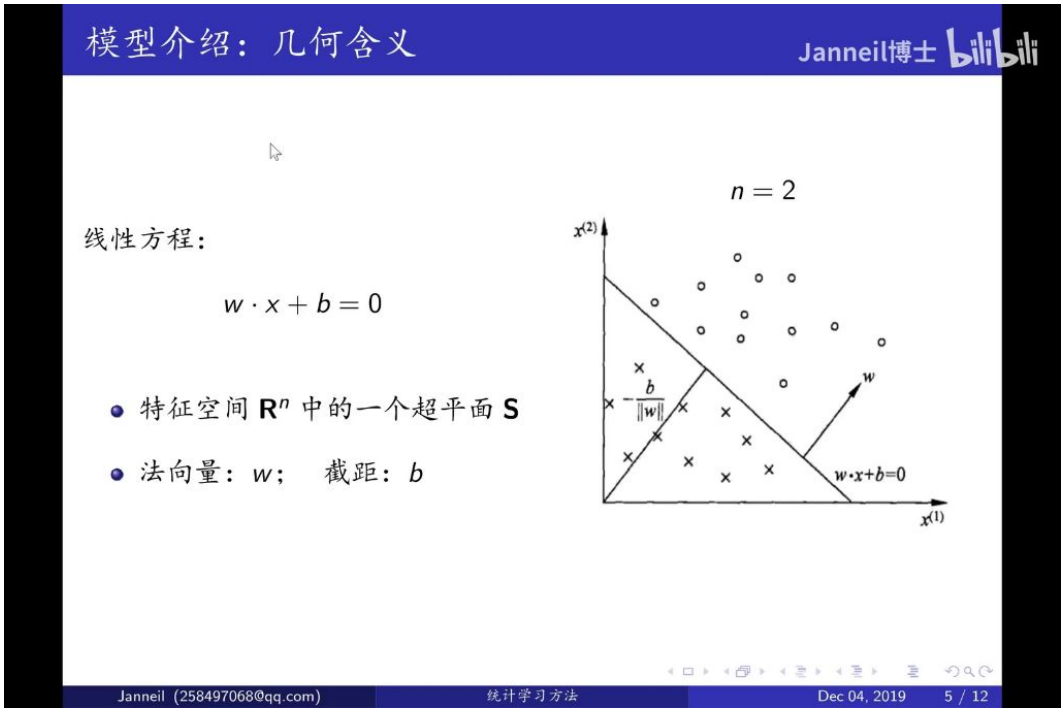
一、感知机 00:03

1. 模型介绍 00:21



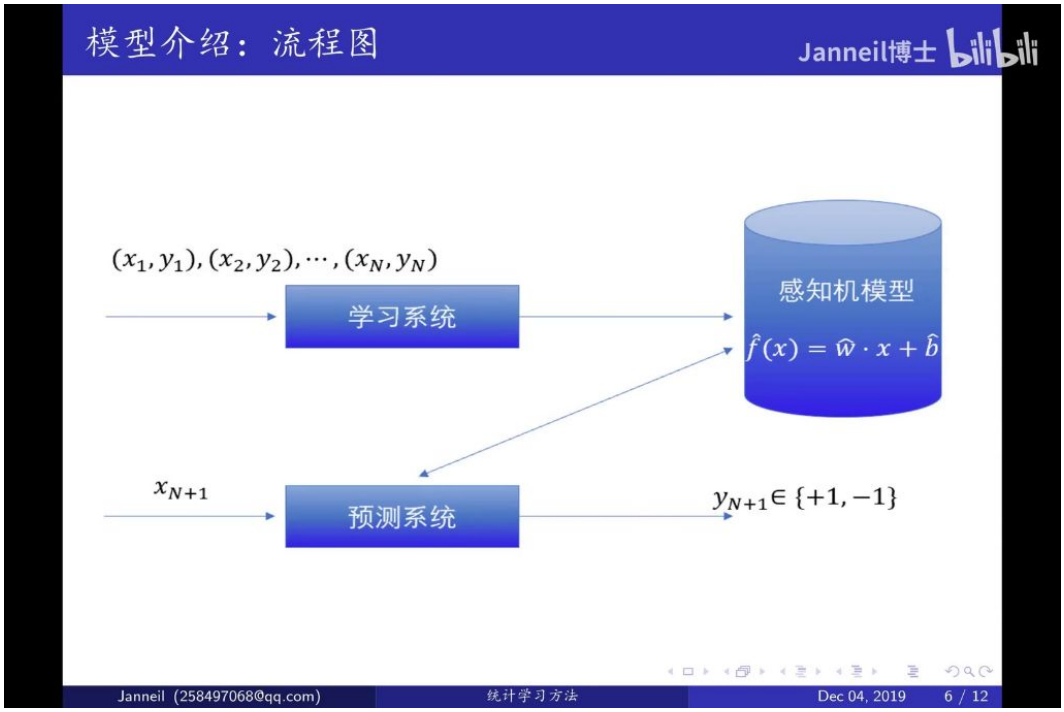
- **输入空间**：$X \subseteq \mathbf{R}^n$，输入实例表示为$x = (x^{(1)}, x^{(2)}, \cdots, x^{(n)})^T \in X$
- **输出空间**：$Y = \{+1, -1\}$，+1代表正类，-1代表负类
- **模型定义**：$f(x) = sign(w \cdot x + b) = \begin{cases} +1, & w \cdot x + b \geq 0 \\ -1, & w \cdot x + b < 0 \end{cases}$
- **参数说明**：
  - **权值向量**：$w = (w^{(1)}, w^{(2)}, \cdots, w^{(n)})^T \in \mathbf{R}^n$
  - **偏置项**：$b \in \mathbf{R}$
  - **内积运算**：$w \cdot x = \sum_{i=1}^{n} w^{(i)}x^{(i)}$
- **假设空间**：$F = \{f \mid f(x) = w \cdot x + b\}$，包含所有可能的线性分类器

1）模型介绍的几何含义 03:17



- **超平面定义**：$w \cdot x + b = 0$是特征空间$\mathbf{R}^n$中的超平面S
  - **法向量**：w垂直于超平面
  - **截距**：b决定超平面位置
- **空间划分**：
  - 正类区域：$w \cdot x + b \geq 0$
  - 负类区域：$w \cdot x + b < 0$
- **维度关系**：
  - 一维特征空间：分离点为实数轴上的点
  - 二维特征空间：分离超平面为直线
  - 三维特征空间：分离超平面为平面
  - n维特征空间：分离超平面为n-1维子空间

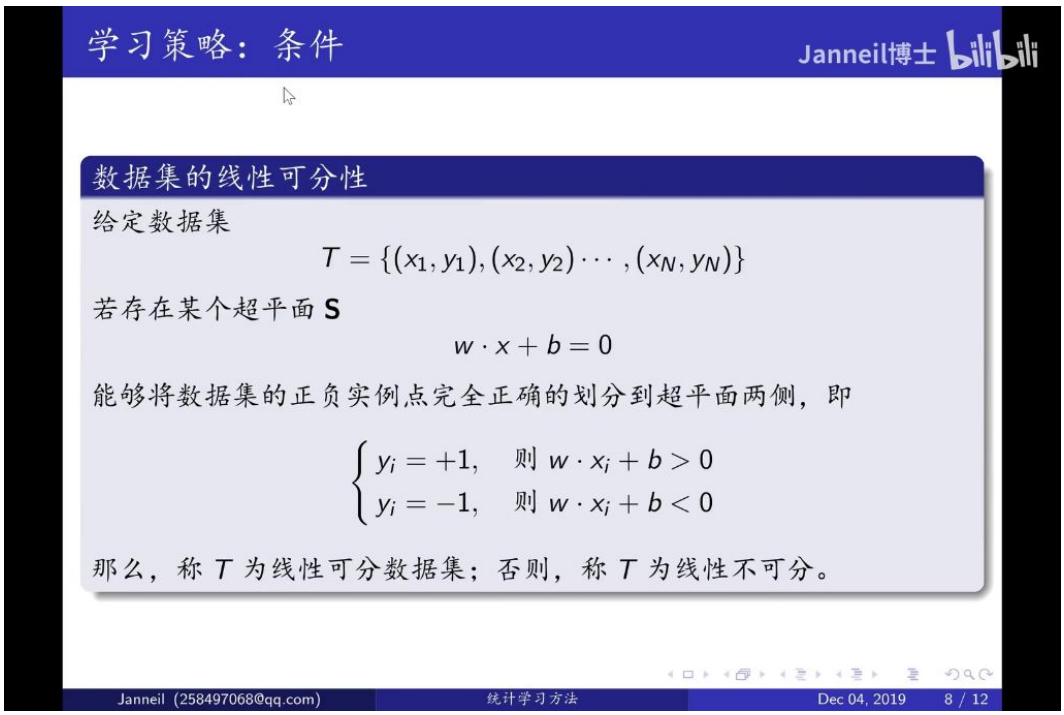
2）模型介绍的流程图 06:15



- **训练阶段**：输入训练集$\{(x_1,y_1),...,(x_N,y_N)\}$，学习系统输出模型参数w,b
- **预测阶段**：新实例x_{N+1}通过模型$f(x) = sign(w \cdot x + b)$预测类别$y_{N+1}$

2. 学习策略 07:06

1）学习策略的条件 07:13

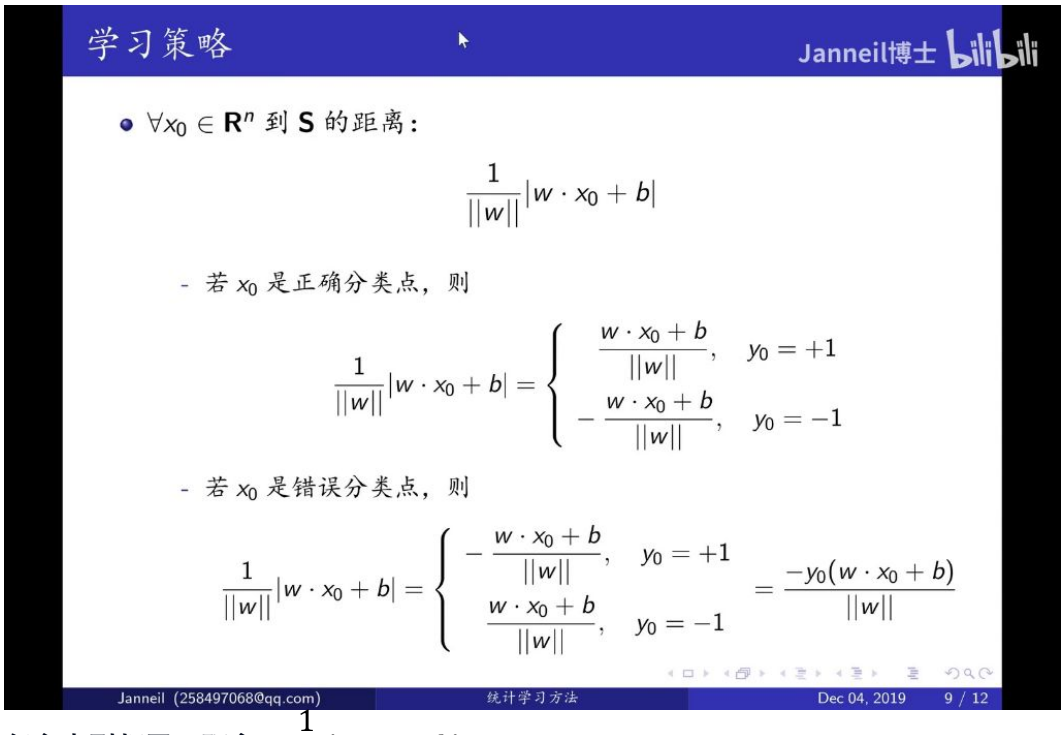


- **线性可分性**：数据集$T = \{(x_1,y_1),...,(x_N,y_N)\}$存在超平面S使得：
  - 正例点满足$w \cdot x_i + b > 0$且$y_i = +1$
  - 负例点满足$w \cdot x_i + b < 0$且$y_i = -1$
- **不可分情况**：无法找到满足上述条件的超平面

2）学习策略的定义 08:33

- **核心目标**：寻找使所有训练样本正确分类的w,b参数
- **方法选择**：通过最小化损失函数来确定最优参数

3）学习策略的误分类点距离 10:39

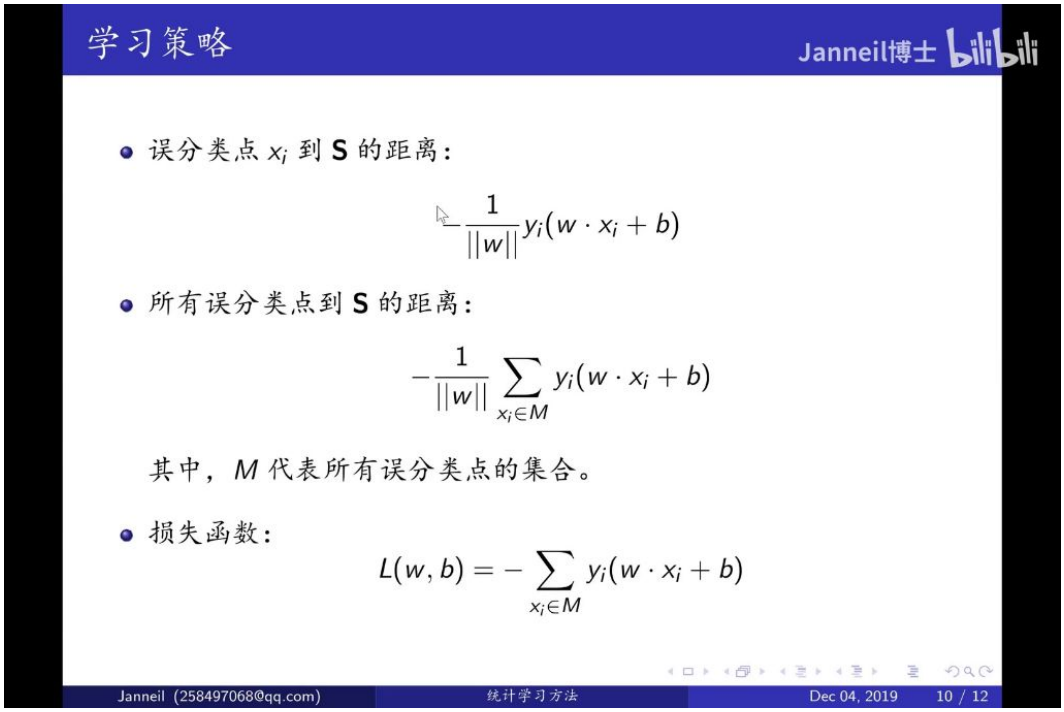


- **任意点到超平面距离**：$\frac{1}{|w|}|w \cdot x_0 + b|$
- **正确分类点距离**：
  - 正类点：$\frac{w \cdot x_0 + b}{|w|}$
  - 负类点：$-\frac{w \cdot x_0 + b}{|w|}$
- **错误分类点距离**：
  - 统一表达式：$\frac{-y_0(w \cdot x_0 + b)}{|w|}$

4）学习策略的所有误分类点距离 11:03

- **单个误分类点距离**：$\frac{-y_i(w \cdot x_i + b)}{|w|}$
- **总距离公式**：$-\frac{1}{|w|}\sum_{x_i \in M} y_i(w \cdot x_i + b)$
  - M为所有误分类点的集合
- **优化考虑**：$|w|$不影响分类结果判断，可忽略

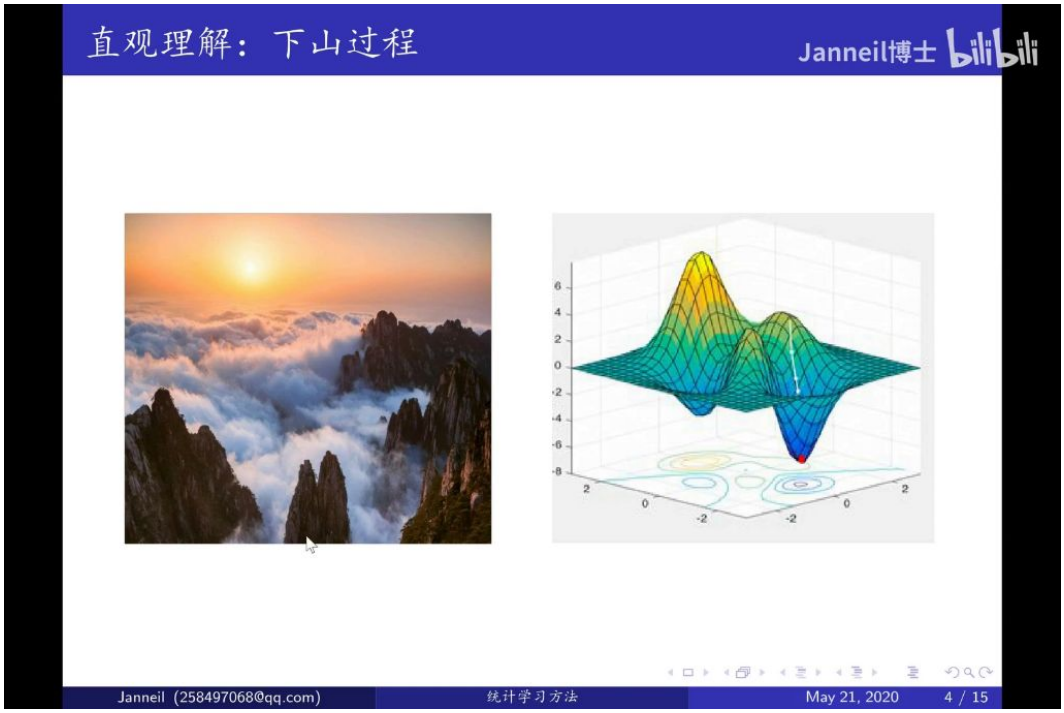
5）学习策略的损失函数 11:21



- **定义**：$L(w,b) = -\sum_{x_i \in M} y_i(w \cdot x_i + b)$
- **性质**：
  - 非负函数，当无误分类点时L(w,b)=0
  - 误分类点越少，损失函数值越小
- **优化目标**：通过最小化L(w,b)寻找最优w,b参数

3. 准备知识 13:01

1）直观理解 13:39

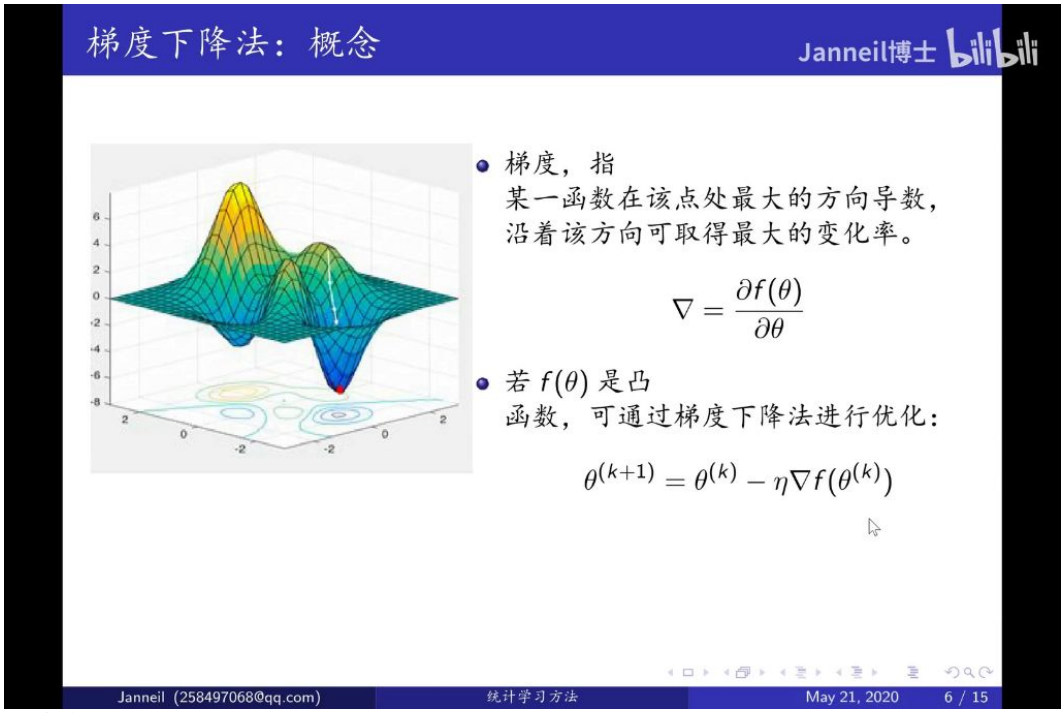


- **核心思想**：通过下山过程的类比理解梯度下降法，每次选择当前位置最陡峭的方向（负梯度方向）前进一小步（由步长η控制）
- **关键要素**：

- **方向**：函数在该点的梯度方向（最陡上升方向），取其反方向即为下降最快方向
- **步长**：每次迭代的移动距离，直接影响收敛速度和精度

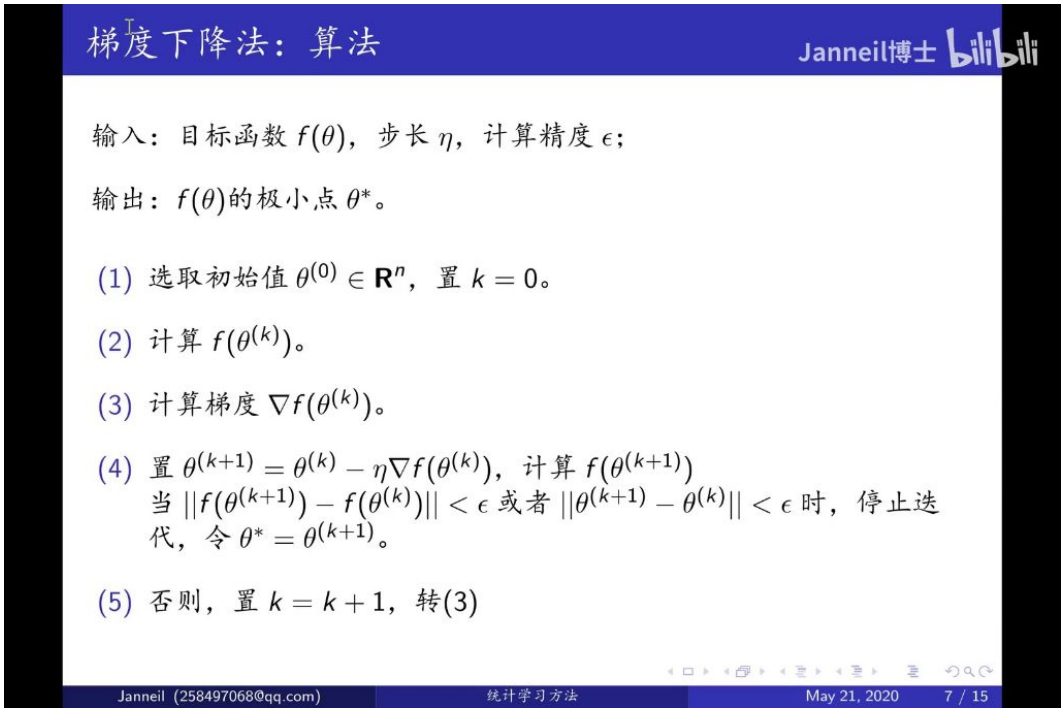
- **类比说明**：如同在浓雾中下山，只能通过局部信息（当前位置的坡度）决定下一步走向

2）梯度下降法概念 15:19



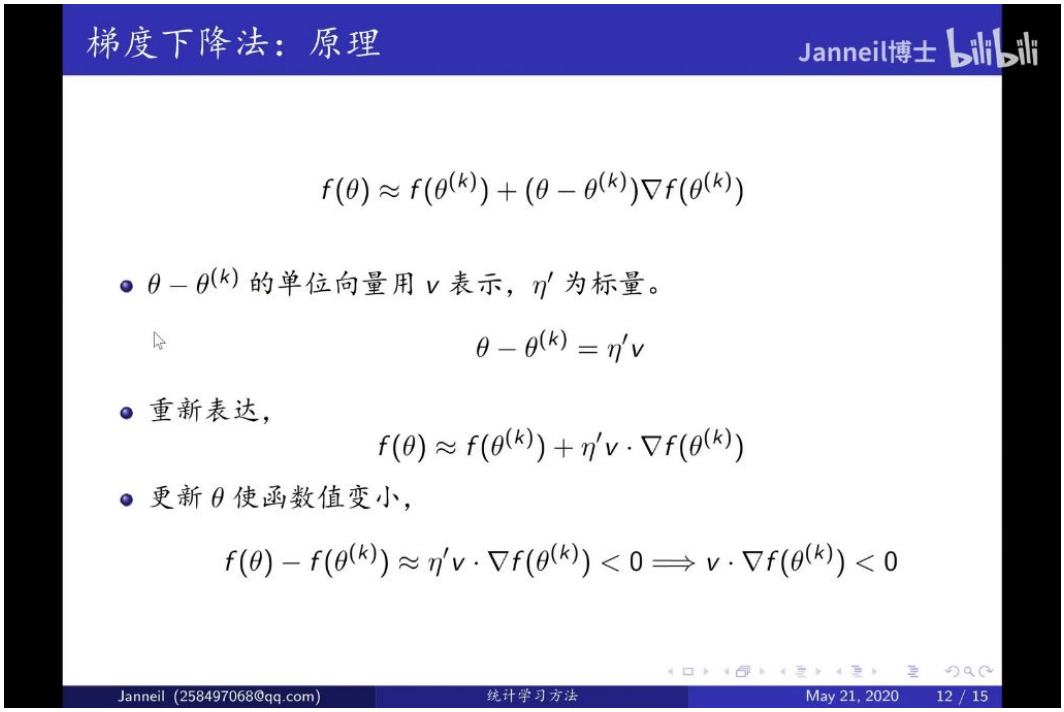
- **梯度定义**：
  - **数学表达**：$\nabla = \frac{\partial f(\theta)}{\partial \theta}$
  - **物理意义**：函数在某点处方向导数最大的向量，包含方向（最大增长率方向）和大小（变化率）
- **适用条件**：要求目标函数$f(\theta)$是凸函数
- **迭代公式**：$\theta^{(k+1)} = \theta^{(k)} - \eta \nabla f(\theta^{(k)})$
  - $\theta^{(k)}$：第k次迭代的参数值
  - η：学习率（步长）
  - $\nabla f(\theta^{(k)})$：当前点的梯度

3）梯度下降法算法 16:34



- **算法流程**:
  - 初始化：选择初始值$\theta^{(0)}$，设k=0
  - 计算当前函数值$f(\theta^{(k)})$
  - 计算当前梯度$\nabla f(\theta^{(k)})$
  - 参数更新：$\theta^{(k+1)} = \theta^{(k)} - \eta \nabla f(\theta^{(k)})$
  - 终止条件：
    - 相邻函数值差$|f(\theta^{(k+1)}) - f(\theta^{(k)})| < \epsilon$
    - 或参数变化量$|\theta^{(k+1)} - \theta^{(k)}| < \epsilon$
- **计算实例**:
  - 单变量函数：$f(x) = x^2$的梯度为$2x$
  - 多变量函数：$f(\theta_1,\theta_2,\theta_3)$的梯度为$[10\theta_1,2,4\theta_3^3]$
  - 收敛示例：当$\eta = 0.4$，$\epsilon = 0.01$时，迭代序列为1→0.2→0.04→0.008→0.0016

4）梯度下降法原理 20:44



- **数学基础**：基于函数的一阶泰勒展开：
- $f(\theta) \approx f(\theta^{(k)}) + (\theta - \theta^{(k)}) \nabla f(\theta^{(k)})$
- **方向选择**：
  - 要使$f(\theta) - f(\theta^{(k)}) < 0$，需满足$v \cdot \nabla f(\theta^{(k)}) < 0$
  - 最优方向：$v = -\frac{\nabla f(\theta^{(k)})}{|\nabla f(\theta^{(k)})|}$（单位负梯度方向）

公式推导：

$$\theta^{(k+1)} = \theta^{(k)} - \eta' \frac{\nabla f(\theta^{(k)})}{|\nabla f(\theta^{(k)})|} = \theta^{(k)} - \eta \nabla f(\theta^{(k)})$$

- 其中$\eta = \eta' / |\nabla f(\theta^{(k)})|$

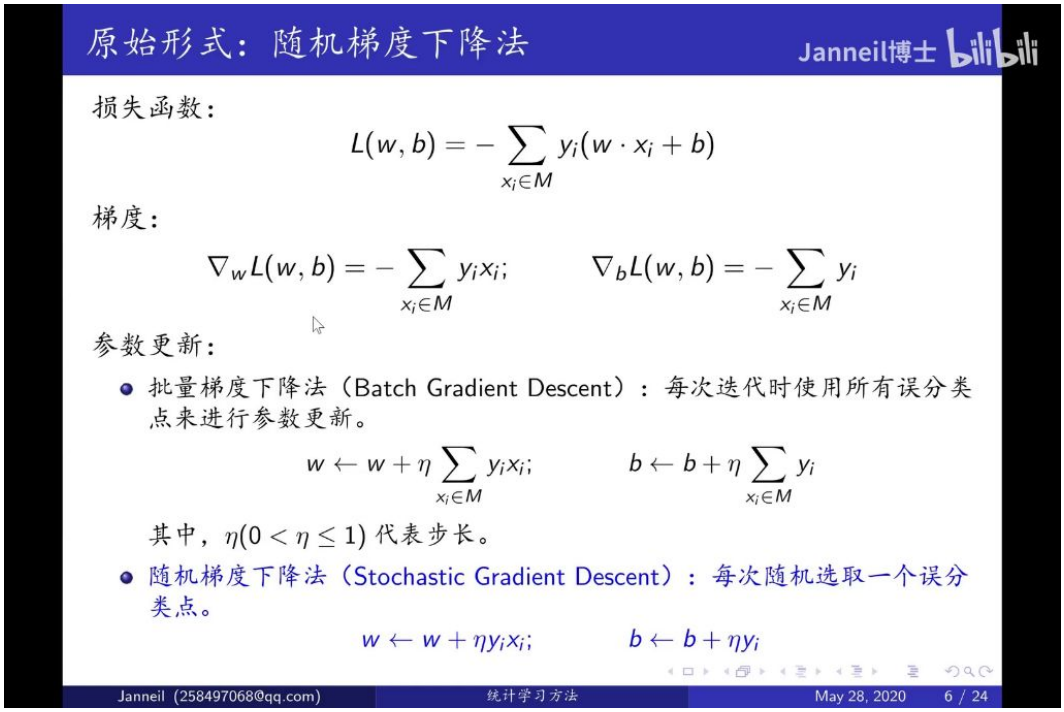
二、感知机的学习算法之原始形式 24:25

1. 学习问题 24:33

- **训练数据集**：$T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$，其中$x_i \in X \subseteq \mathbf{R}^n$为n维特征向量，$y_i \in Y = \{+1, -1\}$表示类别标签（+1为正类，-1为负类）
- **损失函数**：$L(w,b) = -\sum_{x_i \in M} y_i (w \cdot x_i + b)$，其中M为误分类点集合，w·x表示向量内积
- **优化目标**：，即寻找使损失函数最小的超平面参数w和b

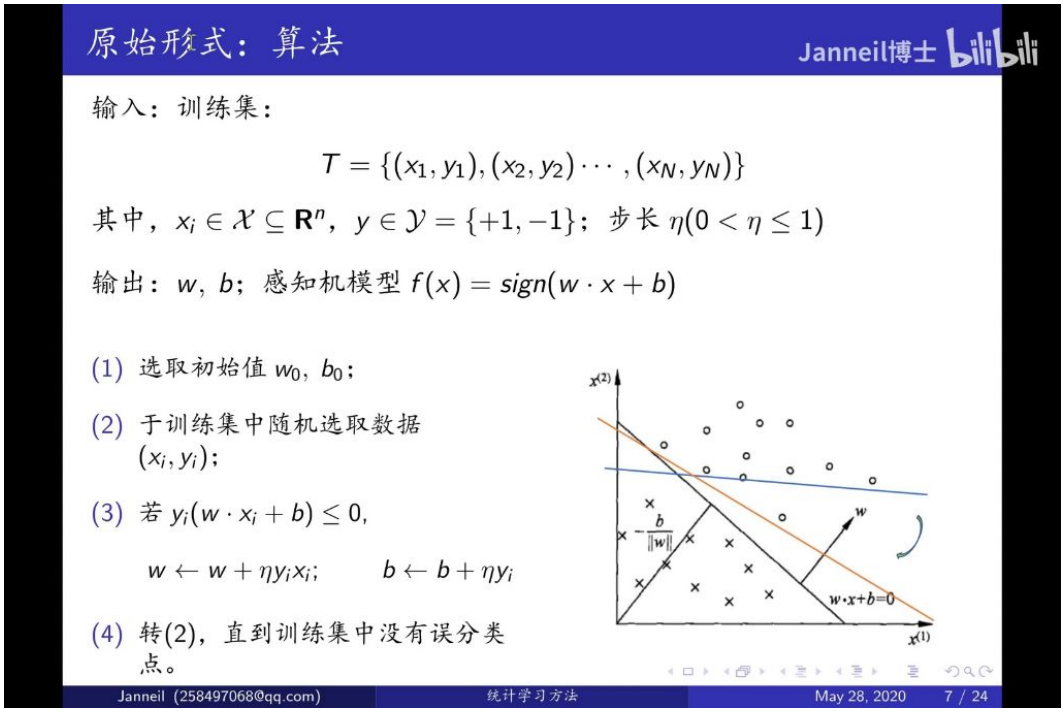
2. 原始形式 26:26

1）随机梯度下降法 26:33



- **梯度计算**:
  - $\nabla_w L(w,b) = -\sum_{x_i \in M} y_i x_i$
  - $\nabla_b L(w,b) = -\sum_{x_i \in M} y_i$
- **批量梯度下降**:
  - 每次迭代使用所有误分类点
  - 更新规则：$w \leftarrow w + \eta \sum_{x_i \in M} y_i x_i$，$b \leftarrow b + \eta \sum_{x_i \in M} y_i$
- **随机梯度下降**:
  - 每次随机选取一个误分类点
  - 更新规则：$w \leftarrow w + \eta y_i x_i$，$b \leftarrow b + \eta y_i$
  - 优势：迭代速度更快，可能减少后续误分类点数量

2）原始形式的算法 30:06



- **输入**：训练集T，学习率$\eta(0 < \eta \leq 1)$
- **输出**：参数w,b；感知机模型$f(x) = sign(w \cdot x + b)$
- **算法步骤**：
  - 初始化$w_0 = 0$，$b_0 = 0$
  - 随机选取数据(x_i, y_i)
  - 若$y_i(w \cdot x_i + b) \leq 0$（误分类），则更新：
    - $w \leftarrow w + \eta y_i x_i$
    - $b \leftarrow b + \eta y_i$
  - 重复2-3直到无误分类点
- **特点**：解不唯一，依赖初始值和误分类点选取顺序

3）应用案例 32:50

- 例题:感知机模型求解

例题分析

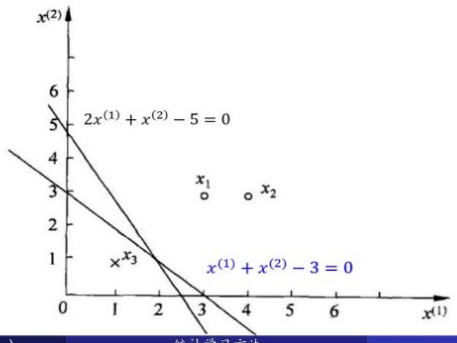
Janneil博士 bilibili

输入：训练集：

$$T = \{(x_1, +1), (x_2, +1), (x_3, -1)\}$$

其中，$x_1 = (3,3)^T$，$x_2 = (4,3)^T$，$x_3 = (1,1)^T$，假设 $\eta = 1$。

输出：w，b；感知机模型 $f(x) = sign(w \cdot x + b)$



例题分析

Janneil博士 bilibili

(4) 重复以上步骤，直到没有误分类点

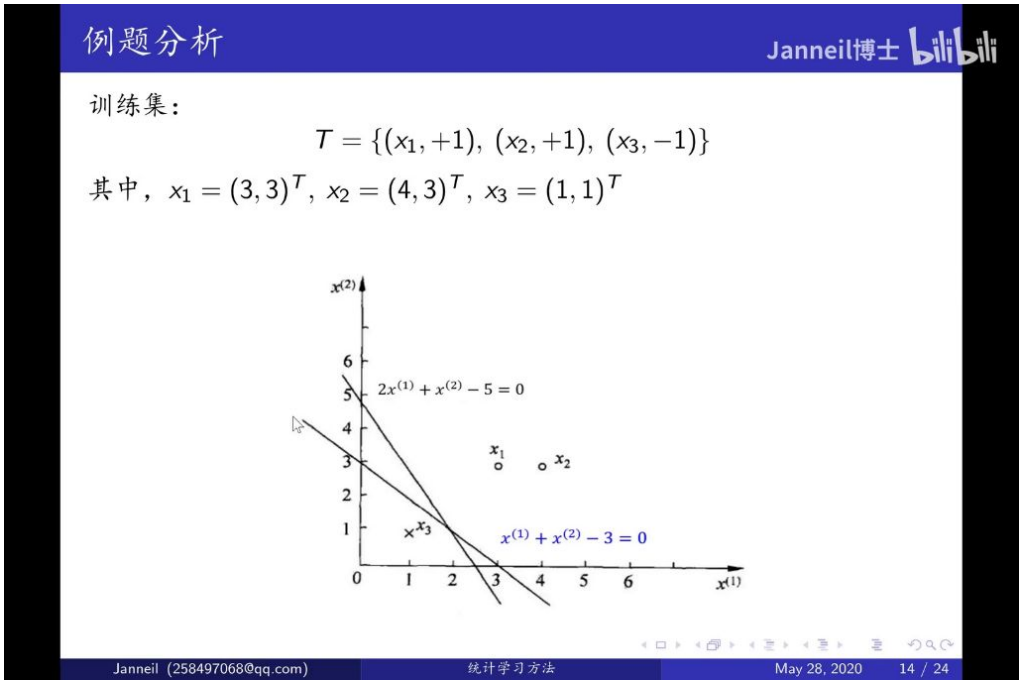
迭代次数	误分类点	w	b	$w \cdot x + b$
0		$(0,0)^T$	0	0
1	x_1	$(3,3)^T$	1	$3x^{(1)} + 3x^{(2)} + 1$
2	x_3	$(2,2)^T$	0	$2x^{(1)} + 2x^{(2)}$
3	x_3	$(1,1)^T$	-1	$x^{(1)} + x^{(2)} - 1$
4	x_3	$(0,0)^T$	-2	-2
5	x_1	$(3,3)^T$	-1	$3x^{(1)} + 3x^{(2)} + 1$
6	x_3	$(2,2)^T$	-2	$2x^{(1)} + 2x^{(2)} - 2$
7	x_3	$(1,1)^T$	-3	$x^{(1)} + x^{(2)} - 3$
8		$(1,1)^T$	-3	$x^{(1)} + x^{(2)} - 3$

得到参数

$$w_7 = (1,1)^T, \qquad b_7 = -3$$

模型，

$$w_7 \cdot x + b_7 = x^{(1)} + x^{(2)} - 3$$

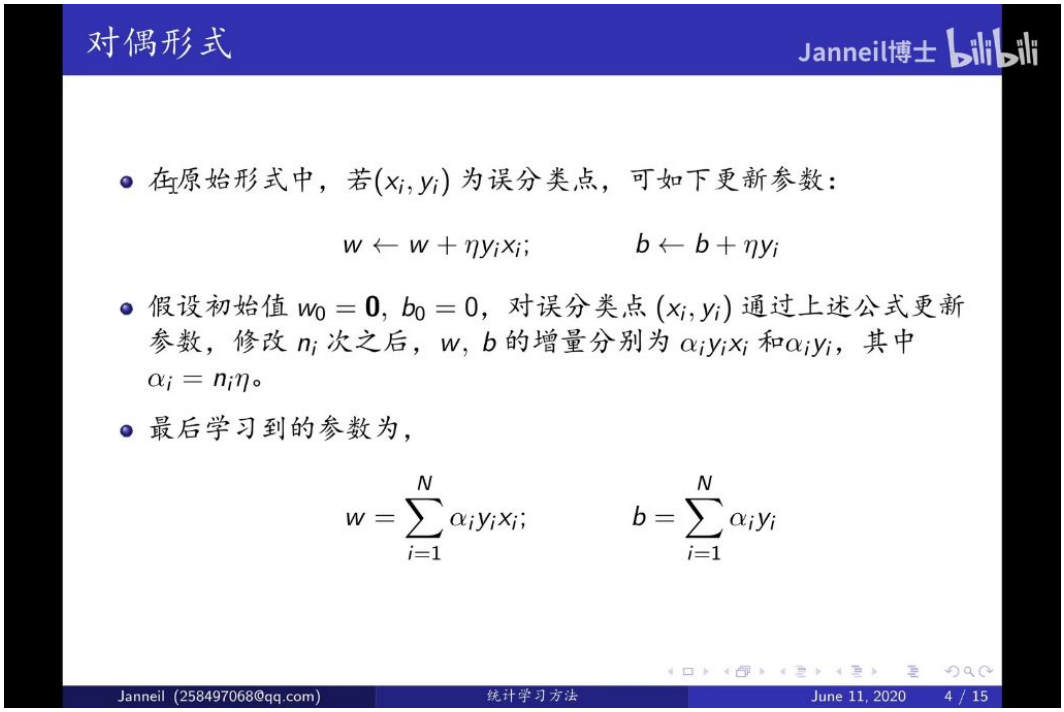


- **题目解析**：
  - 训练集：$T = \{(3,3)^T, +1), ((4,3)^T, +1), ((1,1)^T, -1)\}$，$\eta = 1$
  - **迭代过程**：
    - 初始$w_0 = (0,0)^T$，$b_0 = 0$
    - 误分类点x_1：更新后$w_1 = (3,3)^T$，$b_1 = 1$，模型$3x^{(1)} + 3x^{(2)} + 1 = 0$
    - 误分类点x_3：更新后$w_2 = (2,2)^T$，$b_2 = 0$，模型$2x^{(1)} + 2x^{(2)} = 0$
    - 经过7次迭代得最终解：$w_7 = (1,1)^T$，$b_7 = -3$，模型$x^{(1)} + x^{(2)} - 3 = 0$
  - **结果验证**：
    - 分离超平面：$x^{(1)} + x^{(2)} - 3 = 0$
    - 感知机模型：$f(x) = sign(x^{(1)} + x^{(2)} - 3)$
  - **注意事项**：
    - 不同误分类点顺序可能得到不同解（如$2x^{(1)} + x^{(2)} - 5 = 0$）
    - 解的不唯一性是感知机算法的特点

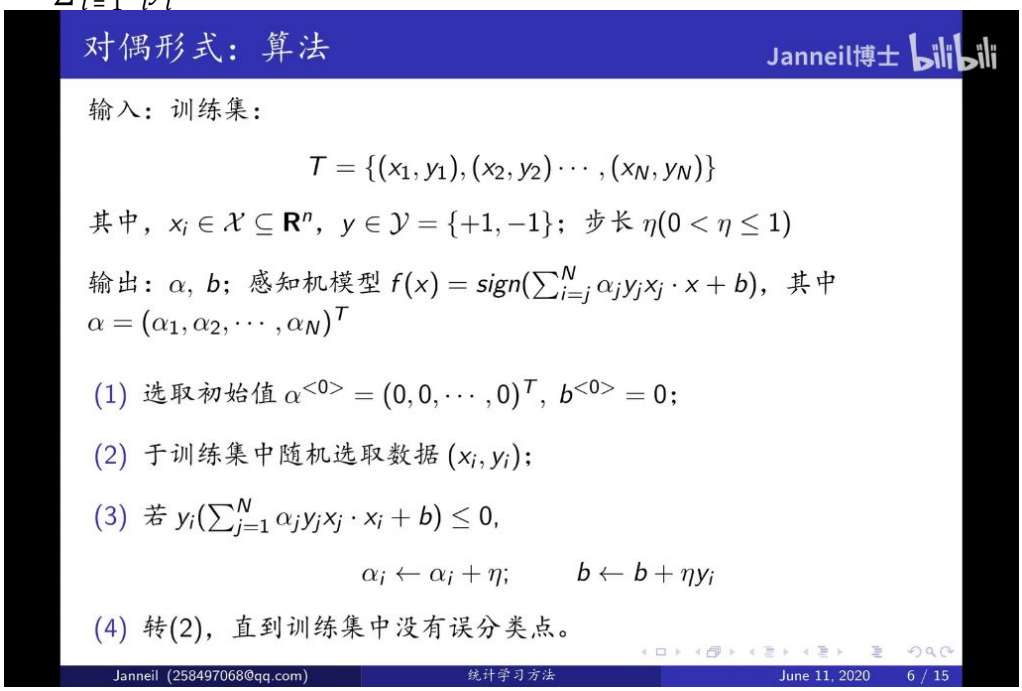
三、感知机的学习算法之对偶形式 39:01

1. 对偶形式 39:23

1）对偶形式的算法 45:33



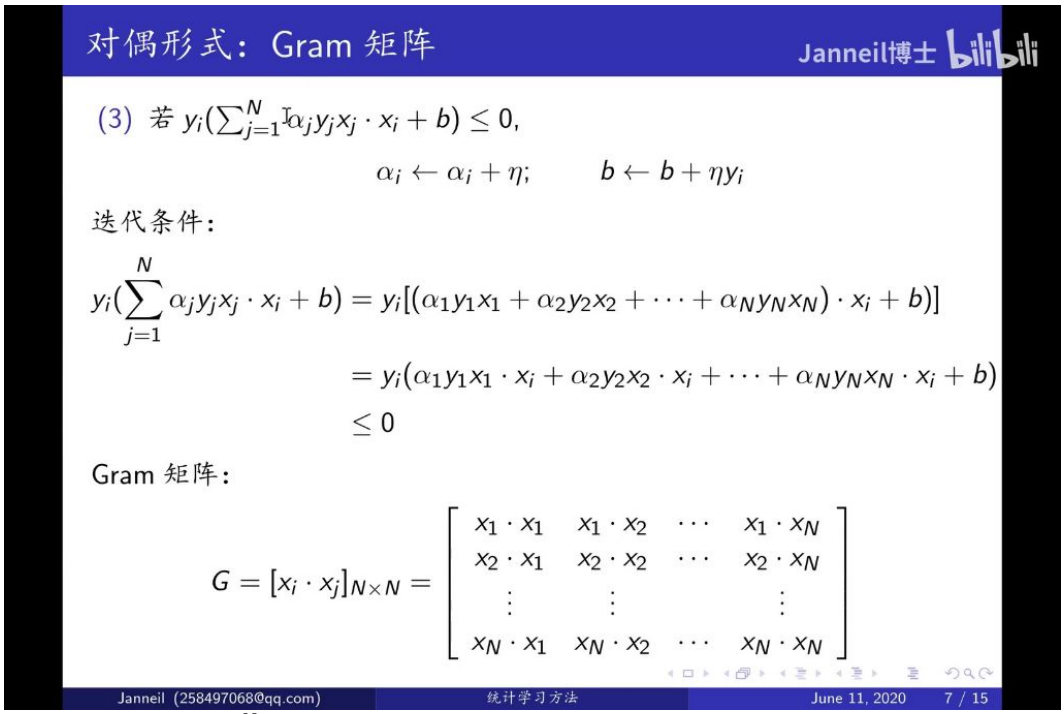
- **参数更新规则**：当(x_i,y_i)为误分类点时，更新公式为：
  - $w \leftarrow w + \eta y_i x_i$
  - $b \leftarrow b + \eta y_i$
- **初始条件**：假设$w_0 = 0$，$b_0 = 0$，对误分类点(x_i,y_i)更新n_i次后：
  - w的增量为$\alpha_i y_i x_i$
  - b的增量为$\alpha_i y_i$
  - 其中$\alpha_i = n_i \eta$
- **最终参数表达式**：
  - $w = \sum_{i=1}^{N} \alpha_i y_i x_i$
  - $b = \sum_{i=1}^{N} \alpha_i y_i$



- **输入**：训练集$T = \{(x_1,y_1),...,(x_N,y_N)\}$，步长$\eta$
- **输出**：α和b，感知机模型$f(x) = sign(\sum_{j=1}^{N} \alpha_j y_j x_j \cdot x + b)$

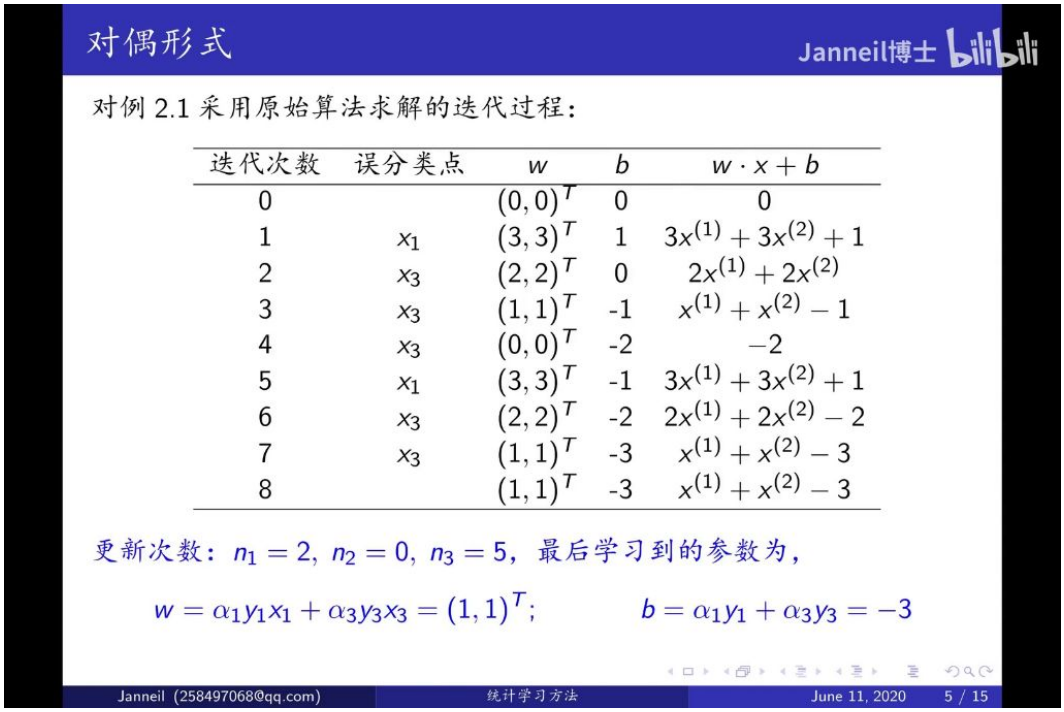
- **算法步骤**:
  - 初始化$\alpha^{(0)}=(0,...,0)^T$，$b^{(0)}=0$
  - 随机选取数据(x_i,y_i)
  - 若$y_i(\sum_{j=1}^{N}\alpha_j y_j x_j \cdot x_i+b)\leq 0$，则更新:
    - $\alpha_i \leftarrow \alpha_i+\eta$
    - $b \leftarrow b+\eta y_i$
  - 重复步骤2-3直到无分类错误

2）对偶形式的迭代条件 47:59



- **误分类条件**：$y_i(\sum_{j=1}^{N}\alpha_j y_j x_j \cdot x_i+b)\leq 0$
- **Gram矩阵**：$G=[x_i \cdot x_j]_{N\times N}$，存储所有样本间的内积
  - 第一行：$x_1\cdot x_1, x_1\cdot x_2, ..., x_1\cdot x_N$
  - 第二行：$x_2\cdot x_1, x_2\cdot x_2, ..., x_2\cdot x_N$
  - ...
  - 第N行：$x_N\cdot x_1, x_N\cdot x_2, ..., x_N\cdot x_N$

3）例题：例2.1的原始算法求解

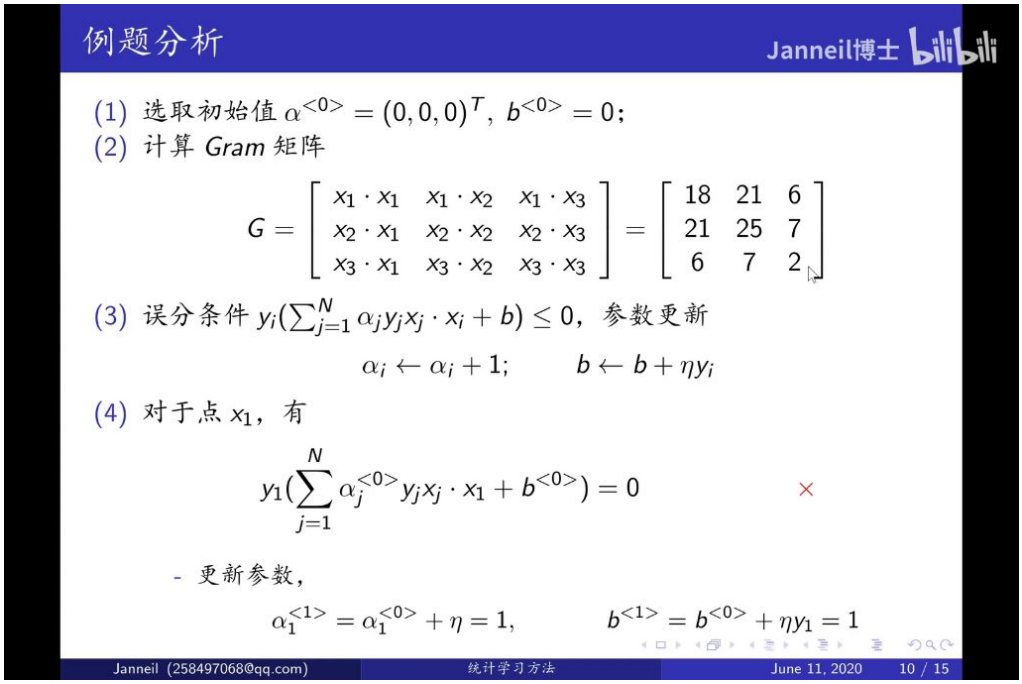


- **题目解析**:
  - 实例x_1作为误分类点出现2次，$n_1 = 2$
  - 实例x_2未出现，$n_2 = 0$
  - 实例x_3作为误分类点出现5次，$n_3 = 5$
  - 总迭代次数为7次（$n_1 + n_3$）
- **参数计算**:
  - 设$\eta = 1$，则$\alpha_1 = 2$，$\alpha_3 = 5$
  - $w = 2 \times 1 \times (3,3) + 5 \times (-1) \times (1,1) = (6,6) + (-5,-5) = (1,1)$
  - $b = 2 \times 1 + 5 \times (-1) = -3$
- **最终模型**：$f(x) = sign(x_1 + x_2 - 3)$

2. 应用案例 50:16

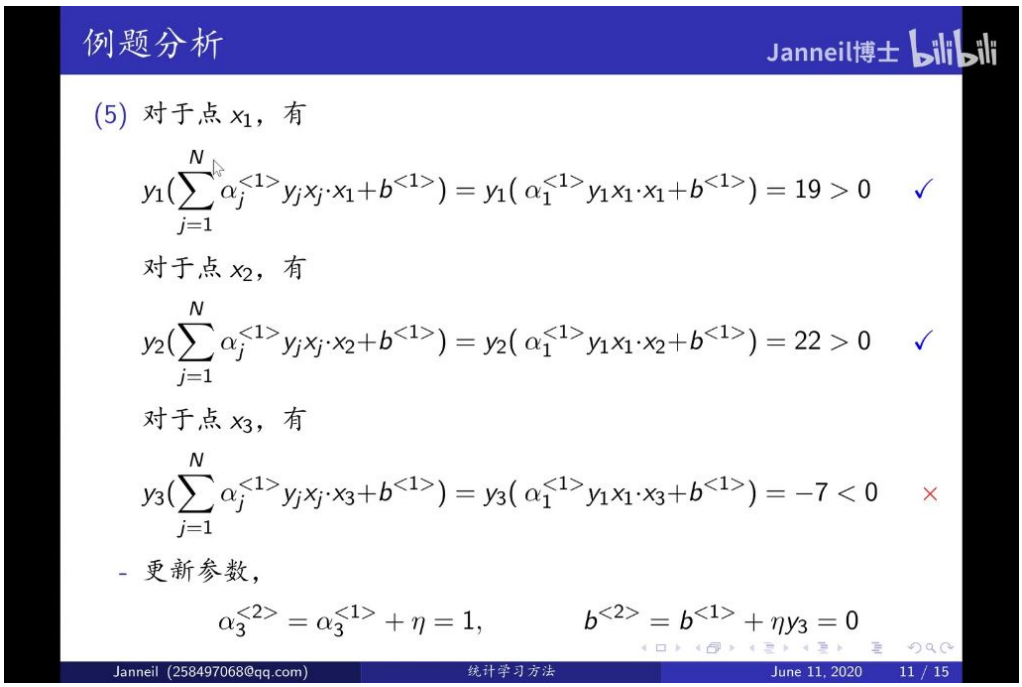
1）例题:感知机模型求解

- **输入数据**:
  - 训练集：$T = \{(x_1, +1),(x_2, +1),(x_3, -1)\}$
  - 其中：$x_1 = (3,3)^T$，$x_2 = (4,3)^T$，$x_3 = (1,1)^T$
  - 学习率：$\eta = 1$
- **输出目标**:
  - 参数α和b
  - 感知机模型：$f(x) = sign(\sum_{j=1}^{N} \alpha_j y_j x_j \cdot x + b)$
- 算法初始化



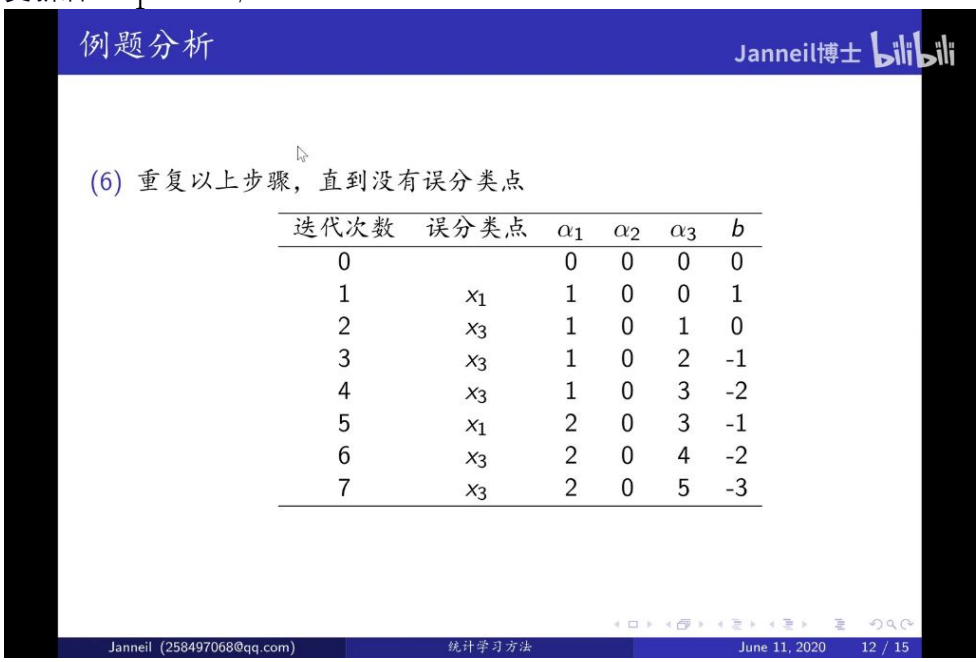
- **初始参数设置**:
  - $\alpha^{<0>}=(0,0,0)^T$
  - $b^{<0>}=0$
- **Gram矩阵计算**:
  - $G=\begin{bmatrix}18 & 21 & 6\\ 21 & 25 & 7\\ 6 & 7 & 2\end{bmatrix}$
  - 计算说明：$x_i\cdot x_j$表示向量内积，如$x_1\cdot x_1=3\times 3+3\times 3=18$

- 迭代过程

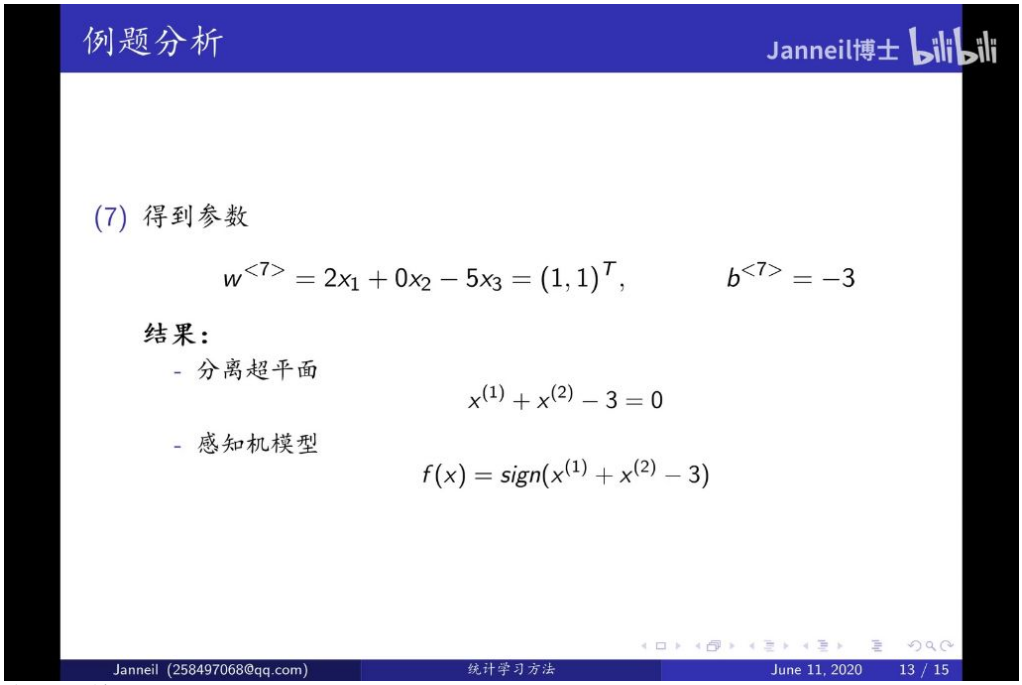


- **误分条件**:
  - $y_i(\sum_{j=1}^{N}\alpha_j y_j x_j\cdot x_i+b)\le 0$
- **参数更新规则**:
  - $\alpha_i\leftarrow\alpha_i+1$

- $b \leftarrow b + \eta y_i$

- **第一次迭代**：
  - 误分点：x_1
  - 更新后：$\alpha_1^{<1>} = 1$，$b^{<1>} = 1$
  - 

  -
- **完整迭代记录**：

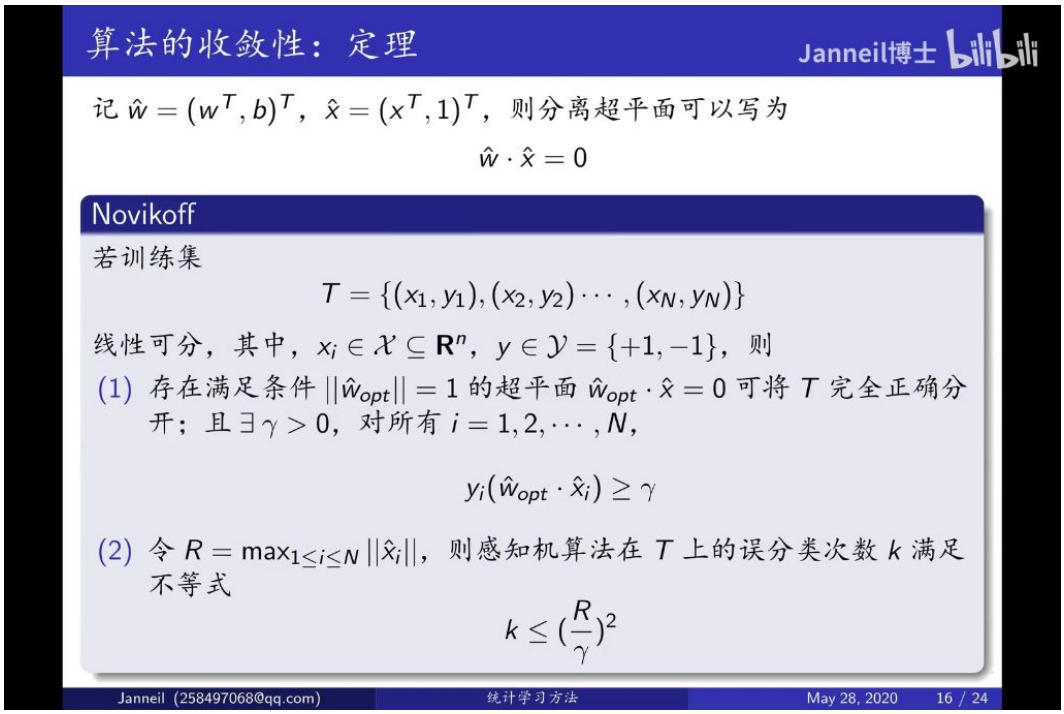
- 最终结果
  - 

  -
- **参数计算**：
  - $w^{<7>} = 2x_1 + 0 \times x_2 - 5x_3 = (1,1)^T$
  - $b^{<7>} = -3$
- **模型输出**：
  - 分离超平面：$x^{(1)} + x^{(2)} - 3 = 0$
  - 感知机模型：$f(x) = sign(x^{(1)} + x^{(2)} - 3)$
- **算法特性**：

- 对偶形式与原始形式最终得到的分离超平面相同
- 算法具有收敛性但解不唯一，需加约束条件才能确定唯一解

四、算法的收敛性 59:13

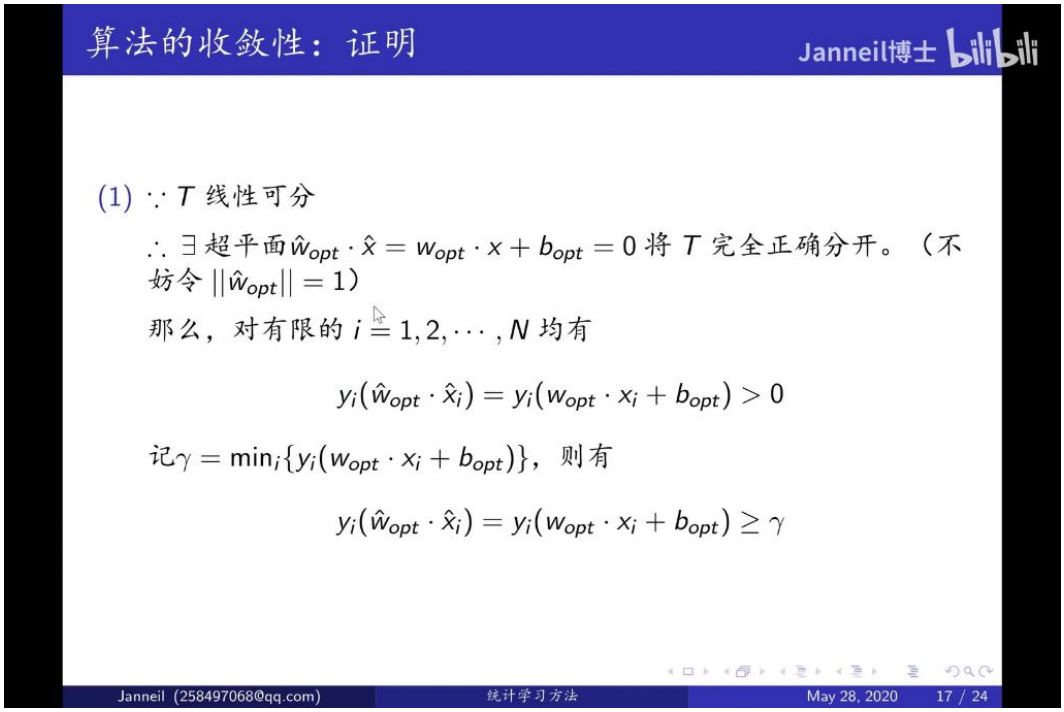
1. 定理 59:42



- **扩充向量表示**:
  - $\hat{w}=(w^T,b)^T$表示扩充权重向量
  - $\hat{x}=(x^T,1)^T$表示扩充样本向量
  - 分离超平面可表示为$\hat{w}\cdot\hat{x}=0$
- **定理内容**:
  - 给定线性可分训练集$T=\{(x_1,y_1),...,(x_N,y_N)\}$，其中$x_i\in R^n,y_i\in\{+1,-1\}$
  - (1) 存在满足条件$||\hat{w}_{opt}||=1$的超平面$\hat{w}_{opt}\cdot\hat{x}=0$将T完全正确分开，且存在$\gamma>0$使得对所有$i=1,...,N$有$y_i(\hat{w}_{opt}\cdot\hat{x}_i)\geq\gamma$
  - (2) 令$R=max_{1\leq i\leq N}||\hat{x}_i||$，则感知机算法在T上的误分类次数k满足$k\leq(\frac{R}{\gamma})^2$

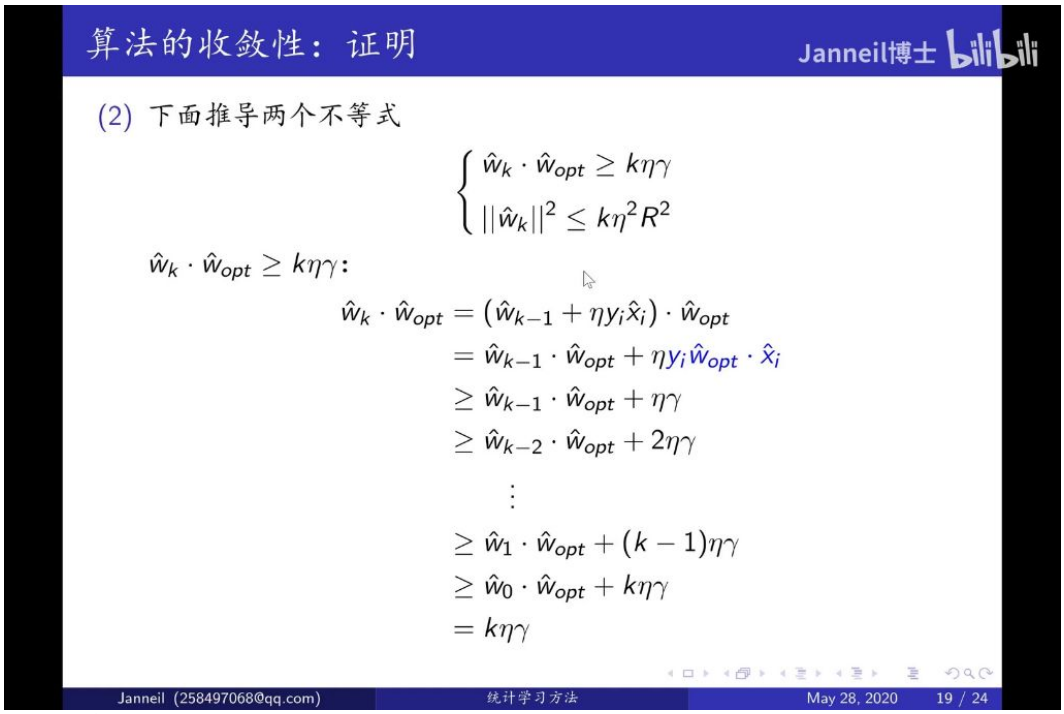
2. 证明 01:02:12

1）第一条证明

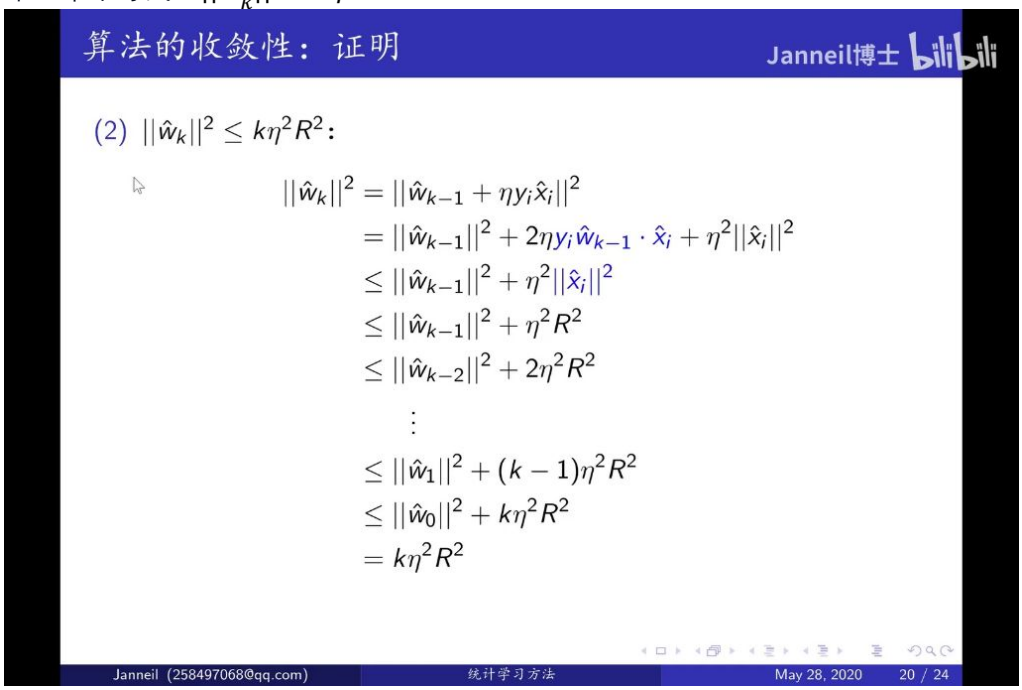


- **超平面存在性**:
  - 由于T线性可分，必存在$\hat{w}_{opt} \cdot \hat{x} = 0$将T完全正确分开
  - 可令$||\hat{w}_{opt}|| = 1$使超平面表示唯一
- **正下限存在性**:
  - 对所有样本有$y_i(\hat{w}_{opt} \cdot \hat{x}_i) > 0$
  - 取$\gamma = min_i\{y_i(w_{opt} \cdot x_i + b_{opt})\}$，则$y_i(\hat{w}_{opt} \cdot \hat{x}_i) \geq \gamma$

2）第二条证明

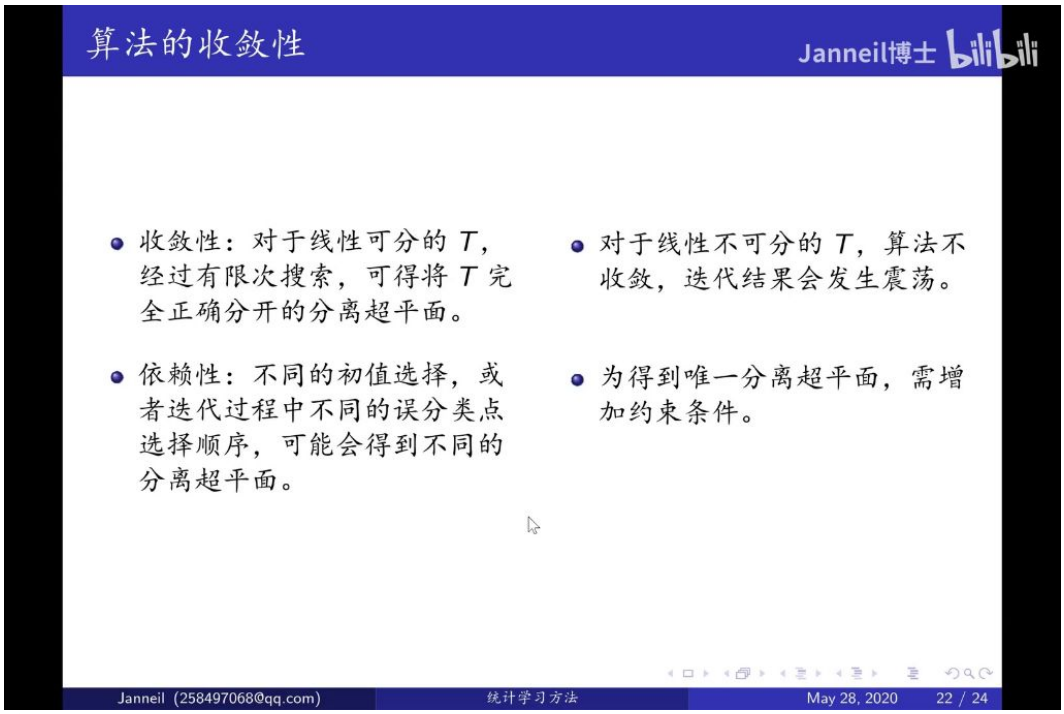


- **权重更新规则**:
  - 初始$\hat{w}_0 = 0$（n+1维零向量）
  - 误分类时更新：$\hat{w}_k = \hat{w}_{k-1} + \eta y_i \hat{x}_i$

- **不等式推导**:
  - 第一个不等式：$\hat{w}_k \cdot \hat{w}_{opt} \geq k\eta\gamma$
  - 第二个不等式：$||\hat{w}_k||^2 \leq k\eta^2 R^2$
  - 

- **最终结论**:
  - 结合两个不等式得到$k\eta\gamma \leq \sqrt{k}\eta R$
  - 化简得$k \leq (\frac{R}{\gamma})^2$，证明误分类次数有上限

3. 收敛性 01:11:24

- 

- **收敛条件**:
  - 线性可分：算法有限次迭代后必收敛，能得到完全正确分类的超平面
  - 线性不可分：算法不收敛，迭代结果会震荡

- **依赖性**:
  - 不同初值选择或误分类点顺序可能导致不同分离超平面
  - 为得到唯一解需增加约束条件（如支持向量机中的约束）

五、知识小结

知识点	核心内容	考试重点/易混淆点	难度系数
感知机模型	二分类线性分类模型，输入空间为n维实数空间子集，输出空间为{+1,-1}	**符号函数定义**: f(x)=sign(w·x+b)	★★★
假设空间与参数空间	特征空间所有可能的线性函数构成假设空间，参数空间为w∈R^n和b∈R组成的n+1维空间	超平面w·x+b=0的几何意义	★★
线性可分性	存在超平面能将数据集完全划分为正负类	**严格定义**：∃超平面s使∀(xi,yi)∈T满足 yi(w·xi+b)>0	★★★★
损失函数	L(w,b)=-∑(yi(w·xi+b))，仅对误分类点求和	**距离推导**：误分类点到超平面距离=-yi(w·xi+b)/‖w‖	★★★★
原始形式算法	随机梯度下降法，参数更新公式：w←w+ηyixi，b←b+ηyi	**迭代特性**：不同初值/误分类顺序会导致不同解	★★★
对偶形式算法	参数表示为αiyixi的线性组合，αi=niη（ni为xi被误分类次数）	**Gram矩阵计算**: G=[xi·xj]_{n×n}	★★★★
收敛性证明	线性可分时迭代次数 k≤(R/γ)²，R=max‖x̂i‖，γ=min yi(w*·x̂i)	**关键不等式**: kηγ≤‖ŵk‖≤ √ kηR	★★★★★
梯度下降法	迭代公式：θ_{k+1}=θ_k-η∇f(θ_ f(θ_k)‖<ε	**步长选择**：过大导致震荡，过小收敛慢	★★★★