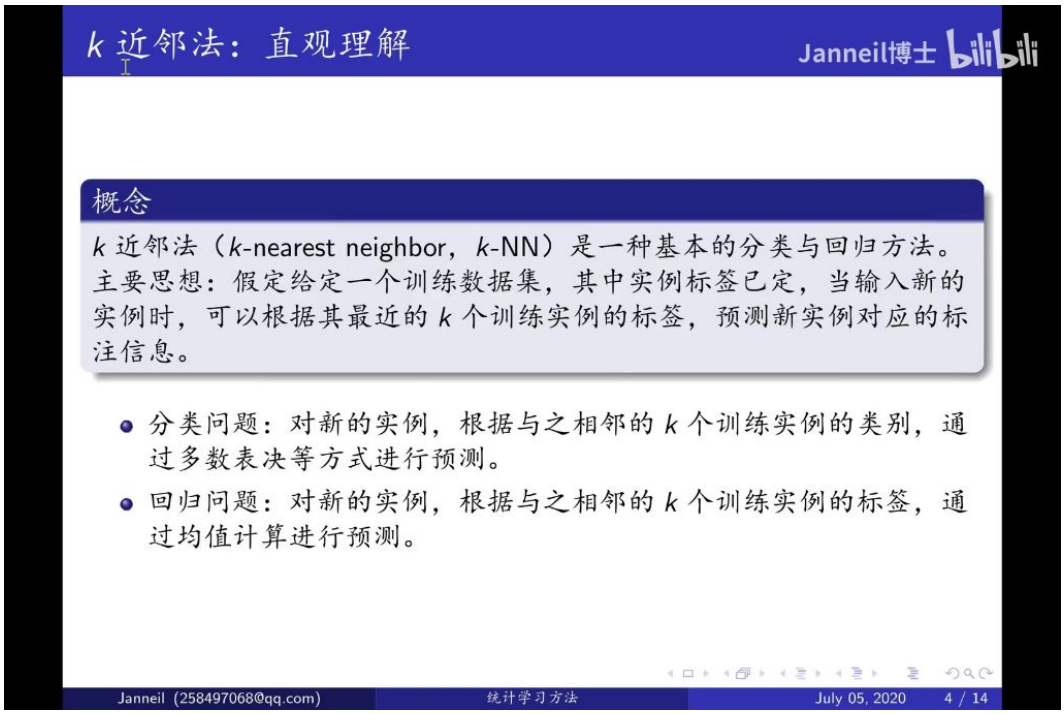
一、统计学习方法 00:02

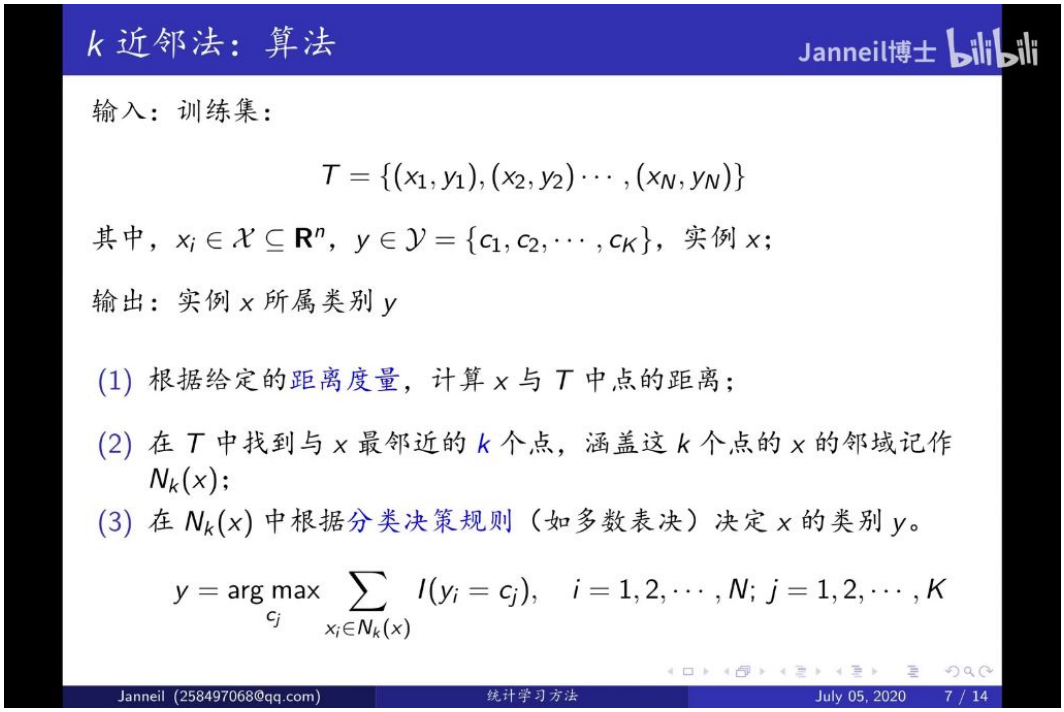
1. k近邻法 00:07

1）直观理解 00:36



- **基本概念**：k近邻法(k-nearest neighbor,k-NN)是一种基本的分类与回归方法，由Cover和Hart于1968年提出。由于没有显式的学习过程，也被称为"懒惰学习"方法。
- **核心思想**：给定训练数据集$T = \{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，其中实例标签已定。当输入新实例x时，根据其最近的k个训练实例的标签预测新实例的标注信息。
- **分类应用**：通过多数表决方式预测新实例类别，即选择k个最近邻中出现最多的类别。
- **回归应用**：通过计算k个最近邻标签的均值进行预测。
- **示例说明**：当k=3时，绿色圆点属于红色三角形类(2/3)；当k=5时，则属于蓝色正方形类(3/5)。说明k值选择会影响分类结果。

2）算法 05:08



- **输入**：训练集$T = \{(x_1,y_1),\cdots,(x_N,y_N)\}$，其中$x_i \in X \subseteq \mathbf{R}^n$，$y \in Y = \{c_1,\cdots,c_K\}$，以及实例x。
- **输出**：实例x所属类别y。
- **算法步骤**：
  - 根据给定距离度量，计算x与T中所有点的距离
  - 在T中找到与x最邻近的k个点，记作$N_k(x)$

在$N_k(x)$中根据分类决策规则(如多数表决)决定x的类别：

  -
- **三要素**：
  - 距离度量：影响最近邻的选择
  - k值选择：影响分类结果
  - 分类决策规则：如多数表决

3）误差率 08:07

最近邻法误差：考虑k=1时，新实例x与最近邻x_i的类别分别为a和b，误差率：

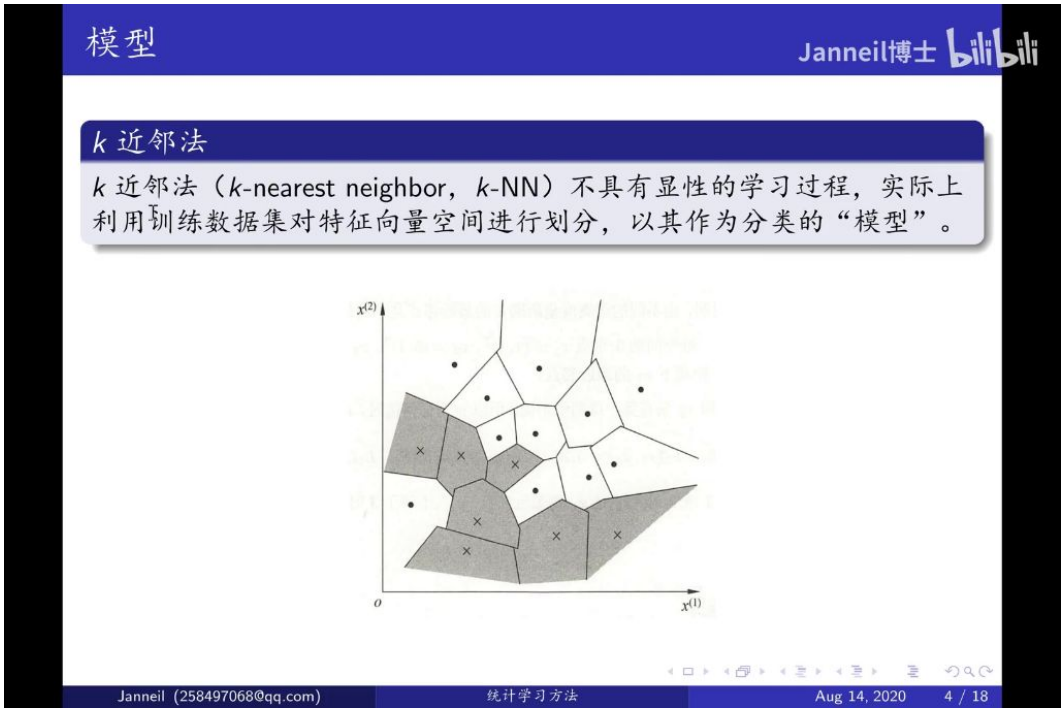
- $Err(x,x_i) = P(a \neq b|x,x_i) = \sum_{j=1}^{K} P(a = c_j|x)(1 - P(b = c_j|x_i))$
- **极限情况**：当$K \to \infty$时，$P(b = c_j|x_i) \to P(a = c_j|x)$，误差率趋近于1 - $\sum_{j=1}^{K} P^2(a = c_j|x)$

误差率边界：设最优类别，贝叶斯误差率$P^* = 1 - P(c^*|x)$，则：

- $P^* \leq Err(x,x_i) \leq 2P^*$
- **推广结论**：当$N \to \infty$且$K \to \infty$时，k近邻法的误差率$Err(x) \to P^*$，即在大样本情况下近似最优决策。

2. k近邻法的三要素 15:02

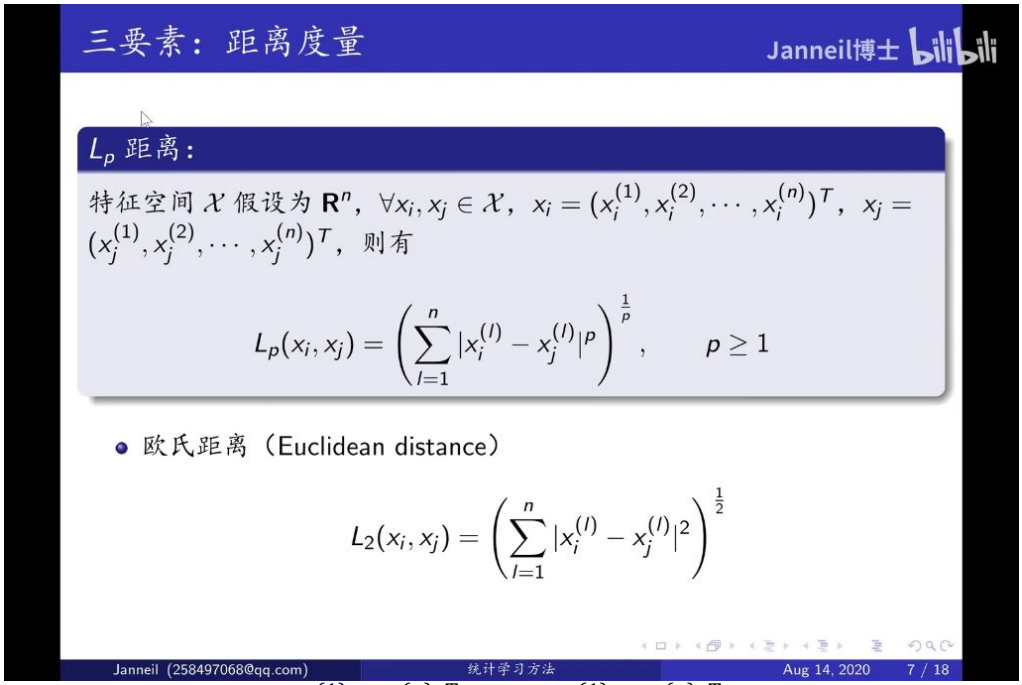
1）模型 15:45



- **本质特征**：k近邻法(k-NN)不具有显性学习过程，通过对特征向量空间的划分实现分类
- **工作原理**：给定新实例时，根据其k个最近邻的训练实例的类别决定自身类别
- **空间划分**：每个训练实例周围形成决策区域（Voronoi单元），新实例落入哪个区域就属于对应类别

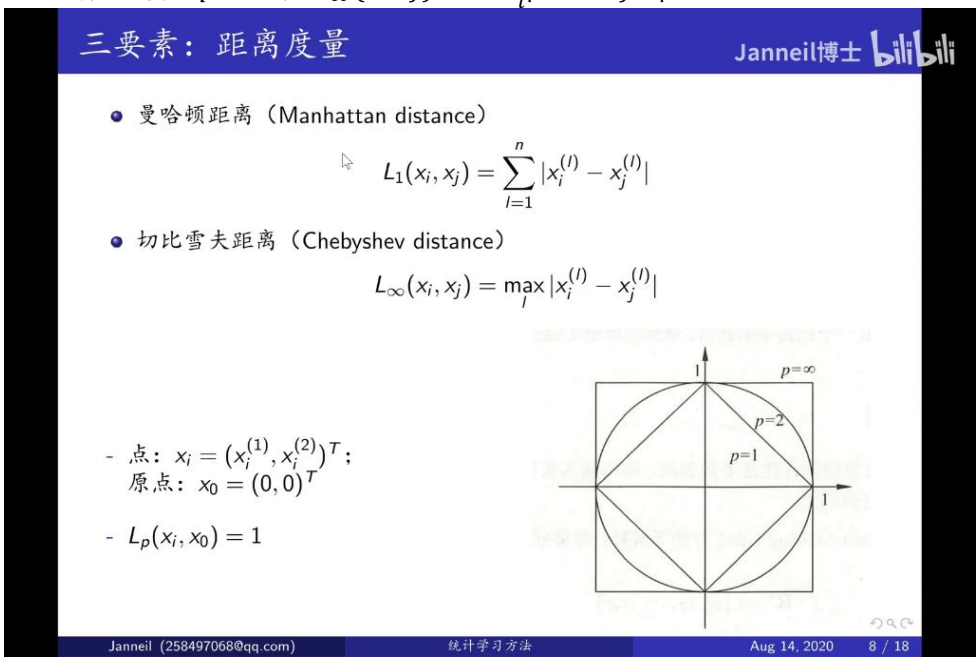
2）三要素 17:37

- 距离度量

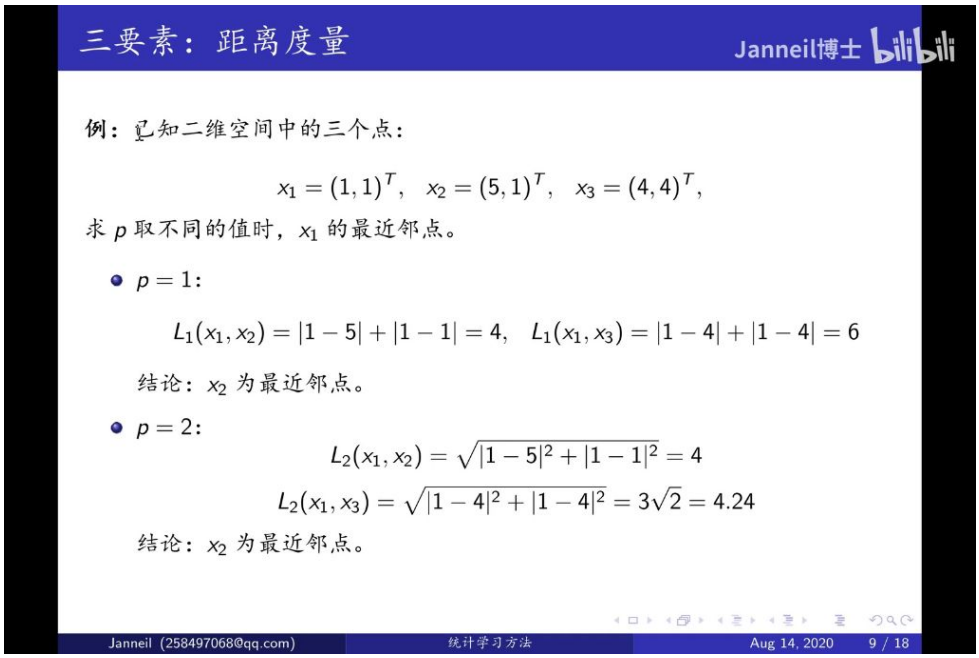


  - **Lp距离定义**：对于$xi = (xi^{(1)},...,xi^{(n)})^T$和$xj = (xj^{(1)},...,xj^{(n)})^T$，距离为$L_p(xi,xj) = (\sum_{l=1}^{n}|xi^{(l)} - xj^{(l)}|^p)^{1/p}$
  - **常见距离类型**：
    - **欧氏距离**：$p = 2$时的特例，$L_2(xi,xj) = (\sum_{l=1}^{n}|xi^{(l)} - xj^{(l)}|^2)^{1/2}$
    - **曼哈顿距离**：$p = 1$，$L_1(xi,xj) = \sum_{l=1}^{n}|xi^{(l)} - xj^{(l)}|$

- **切比雪夫距离**：$p = \infty$，$L_\infty(xi,xj) = max_l|xi^{(l)} - xj^{(l)}|$

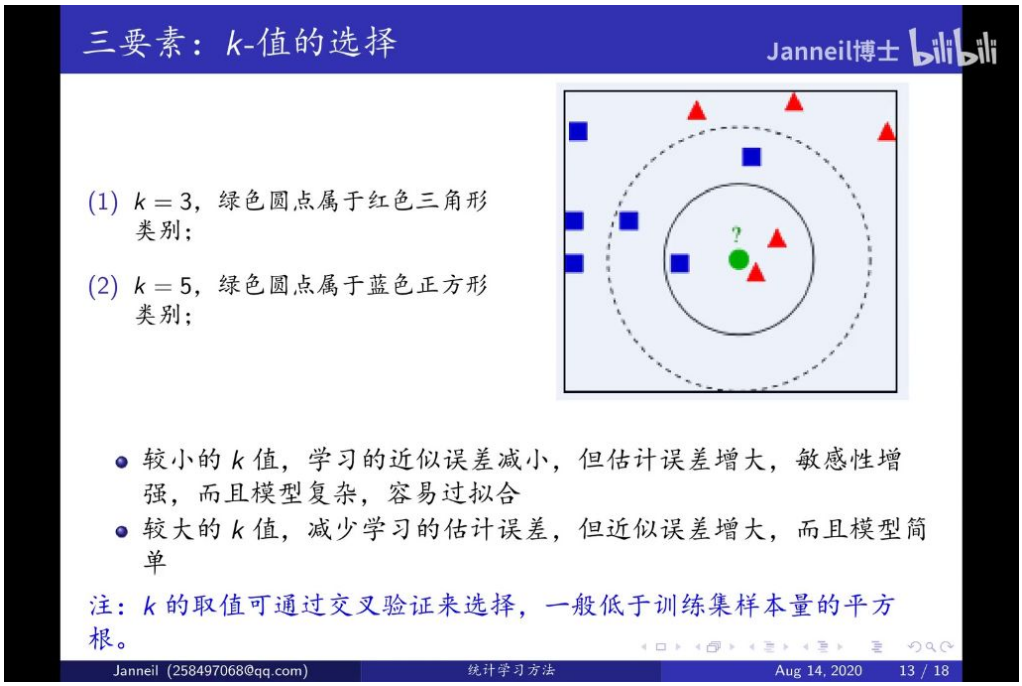


- **几何解释**：对于二维空间中的单位距离点集（$L_p(xi,x0) = 1$）：
  - $p = 2$形成标准单位圆
  - $p = 1$形成边长为$\sqrt{2}$的菱形
  - $p = \infty$ 形成边长为2的正方形
- 例题:最近邻点选择 23:04

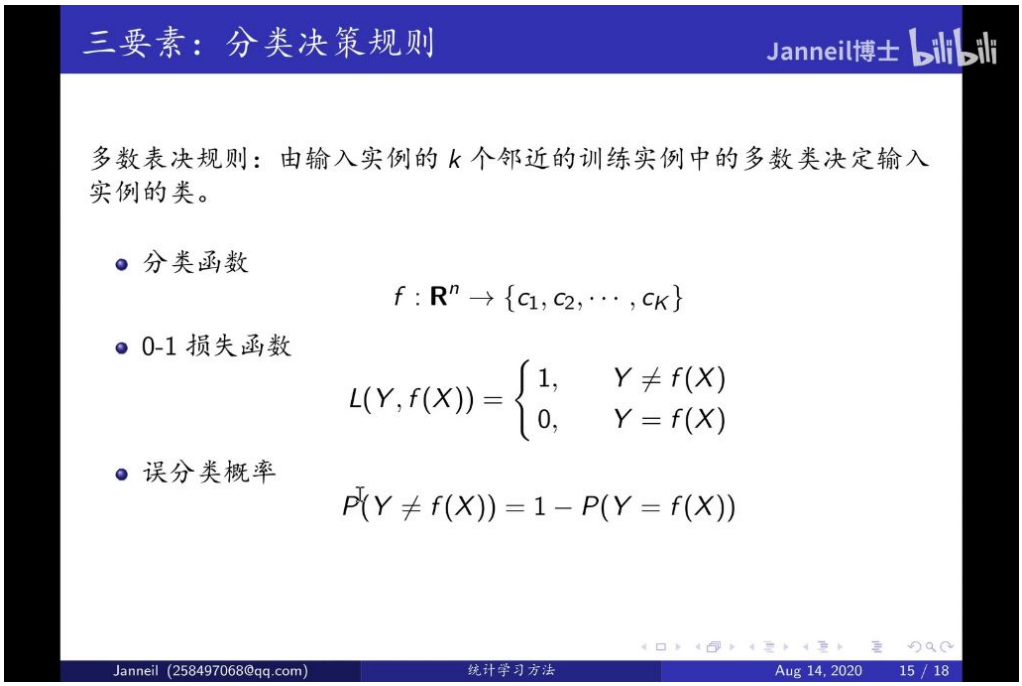


- **题目解析**:
  - 给定点$x_1 = (1,1)^T$， $x_2 = (5,1)^T$， $x_3 = (4,4)^T$
  - 计算不同p值时x_1的最近邻点
    - $p = 1$：$L_1(x1,x2) = 4$，$L_1(x1,x3) = 6 \rightarrow x_2$最近
    - $p = 2$：$L_2(x1,x2) = 4$，$L_2(x1,x3) = 3\sqrt{2} \approx 4.24 \rightarrow x_2$最近
    - $p = 3$：$L_3(x1,x2) = 4$，$L_3(x1,x3) \approx 3.78 \rightarrow x_3$最近
    - $p \geq 4$时x_3始终为最近邻点

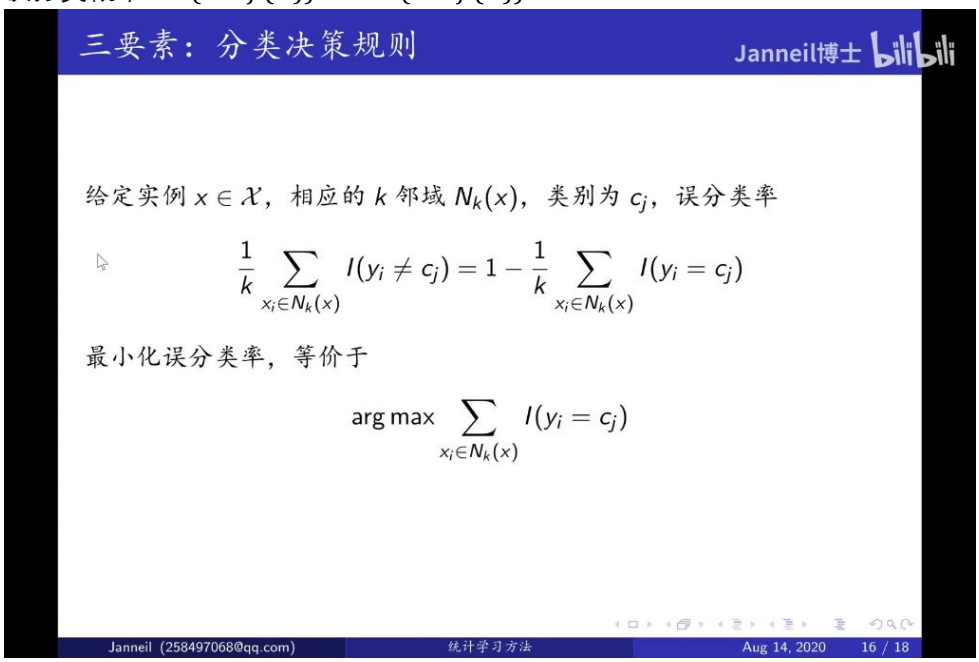
    - **结论**：距离度量方式直接影响最近邻点的选择结果
- k值的选择 26:08



    - **k值影响**：
        - **较小k值**：近似误差小但估计误差大，模型复杂易过拟合
        - **较大k值**：估计误差小但近似误差大，模型简单
    - **选择建议**：
        - 通过交叉验证选择最优k值
        - 一般取值低于训练样本量的平方根
    - **极端情况**：当$k = N$（样本总数）时，所有新实例都被预测为训练集中最多的类别
- 分类决策规则 29:18



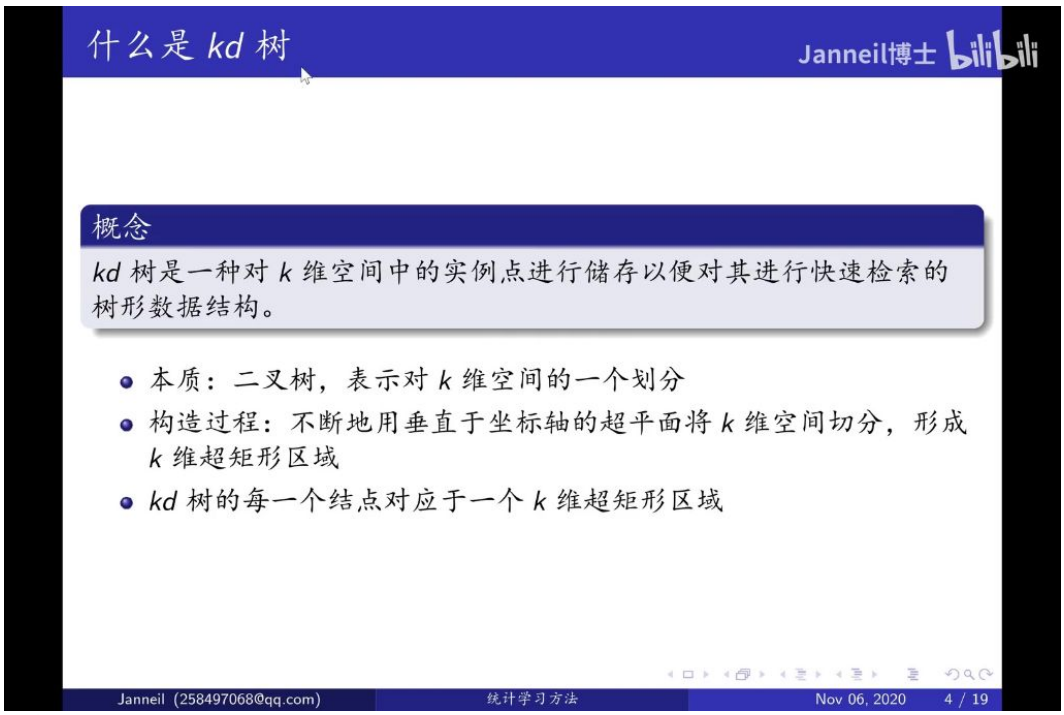
    - **多数表决规则**：新实例的类别由其k个最近邻中出现最多的类别决定
    - **数学表达**：

- 分类函数：$f:R^n \to c_1,\ldots,c_K$
- 0-1损失函数：$L(Y,f(X)) = I(Y \neq f(X))$
- 误分类概率：$P(Y \neq f(X)) = 1 - P(Y = f(X))$
- 

- **优化目标**：
  - 最小化误分类率等价于最大化$\sum_{xi \in N_k(x)} I(yi = c_j)$
  - 经验风险最小化与多数表决规则等价
- **实现方式**：在k邻域$N_k(x)$内统计各类别出现次数，选择出现最多的类别

3. 构造kd树 33:05

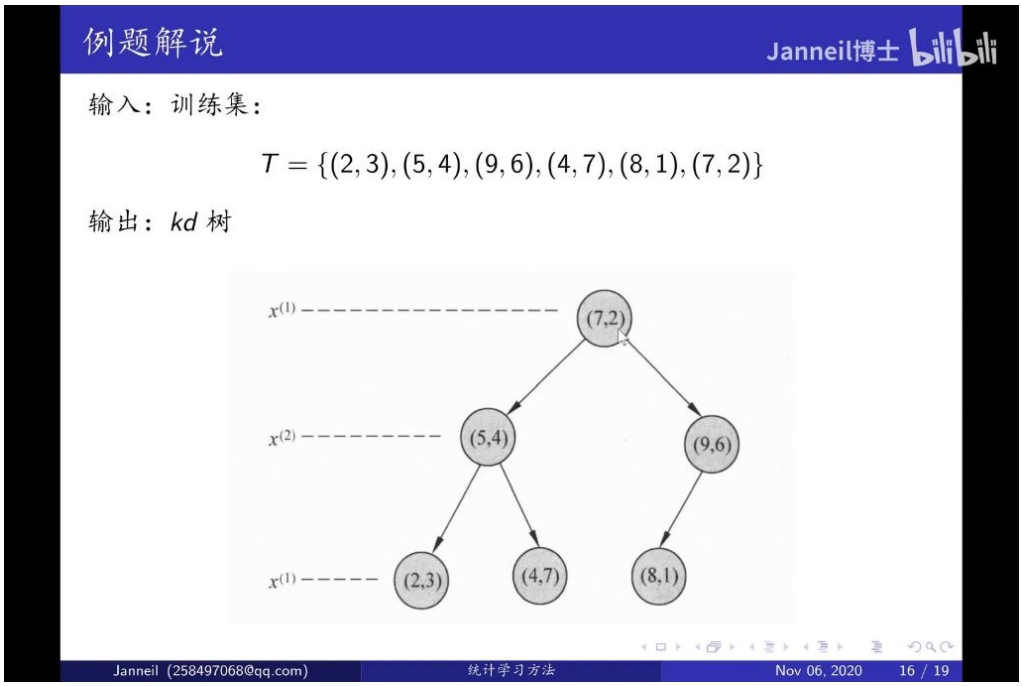
1）什么是kd树 35:42

- 

- **数据结构本质**：二叉树结构，用于对k维空间进行划分
- **核心功能**：存储k维空间中的实例点，实现新实例点的快速检索

- **节点对应关系**：每个节点对应一个k维超矩形区域（k=2时为矩形，k=3时为立方体）
- **维度说明**：k表示数据特征的维度数，区别于k近邻中的k值
- **构造原理**：通过垂直于坐标轴的超平面不断切分空间区域

2）构造kd树 38:39

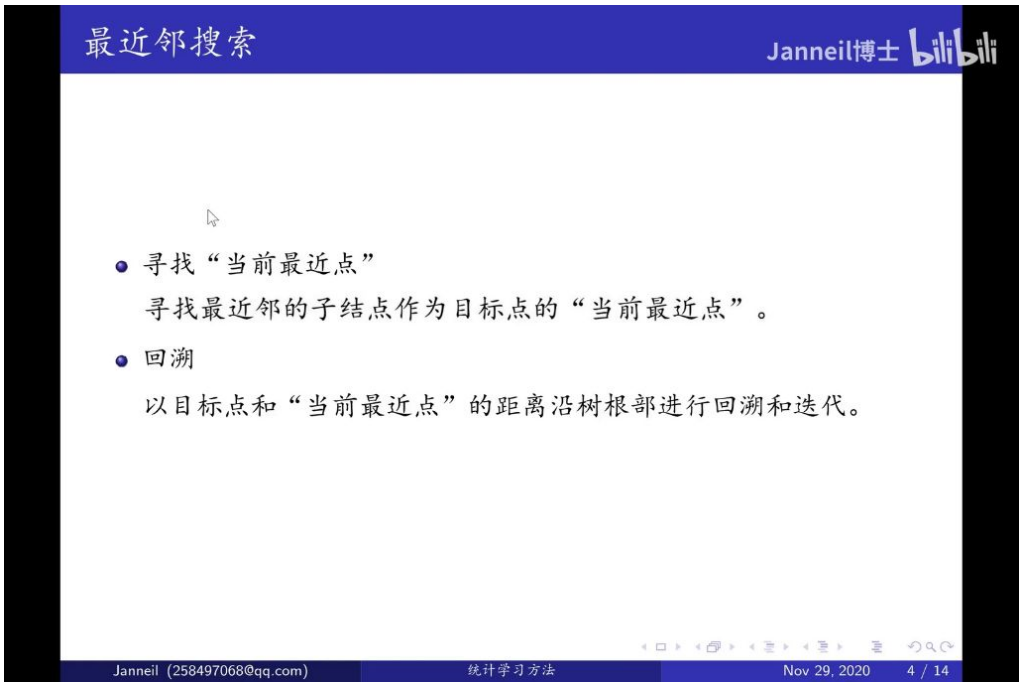
- **输入输出**：
  - 输入：k维数据集$T = \{x_1, x_2, ..., x_N\}$，其中$x_i = (x_i^{(1)}, x_i^{(2)}, ..., x_i^{(k)})^T$
  - 输出：构造完成的kd树结构
- **构造步骤**：
  - **根节点构造**：
    - 坐标轴选择：优先选择方差最大的特征轴（教材默认选$x^{(1)}$轴）
    - 切分点确定：取选定坐标轴上所有数据的中位数
    - 区域划分：将超矩形区域划分为两个子区域
    - 节点关系：切分点作为根节点，生成深度为1的左右子节点
  - **递归划分**：
    - 坐标轴轮换：深度j的节点选择$x^{(l)}$轴，$l = j(mod\ k) + 1$
    - 中位数切分：在当前区域所有实例的选定坐标轴上取中位数
    - 生成子节点：深度j+1的左（小于切分点）、右（大于切分点）子节点
  - **终止条件**：当子区域没有实例需要划分时停止
- **注意事项**：
  - 中位数选择：偶数个数据时可选择中间偏右的值
  - 特征轮换：当$l > k$时需重新从第一个特征开始
  - 节点深度：根节点深度为0，每层子节点深度+1
- 例题:kd树构造示例 46:17
  - 

  - **示例数据**：二维数据集$T = \{(2,3),(5,4),(9,6),(4,7),(8,1),(7,2)\}$
  - **构造过程**：
    - 第一层划分（x轴）：
      - 排序x值：2,4,5,7,8,9
      - 选择切分点：(7,2)
      - 左右区域：左区域x<7，右区域x>7
    - 第二层划分（y轴）：

        - 左区域y值：3,4,7 → 中位数(5,4)
        - 右区域y值：1,6 → 中位数(9,6)
    - 第三层划分（x轴）：
        - 左子区域：上区(4,7)，下区(2,3)
        - 右子区域：(8,1)
- **最终结构**：
    - 根节点：(7,2)
    - 第二层：(5,4)和(9,6)
    - 第三层：(2,3)、(4,7)、(8,1)

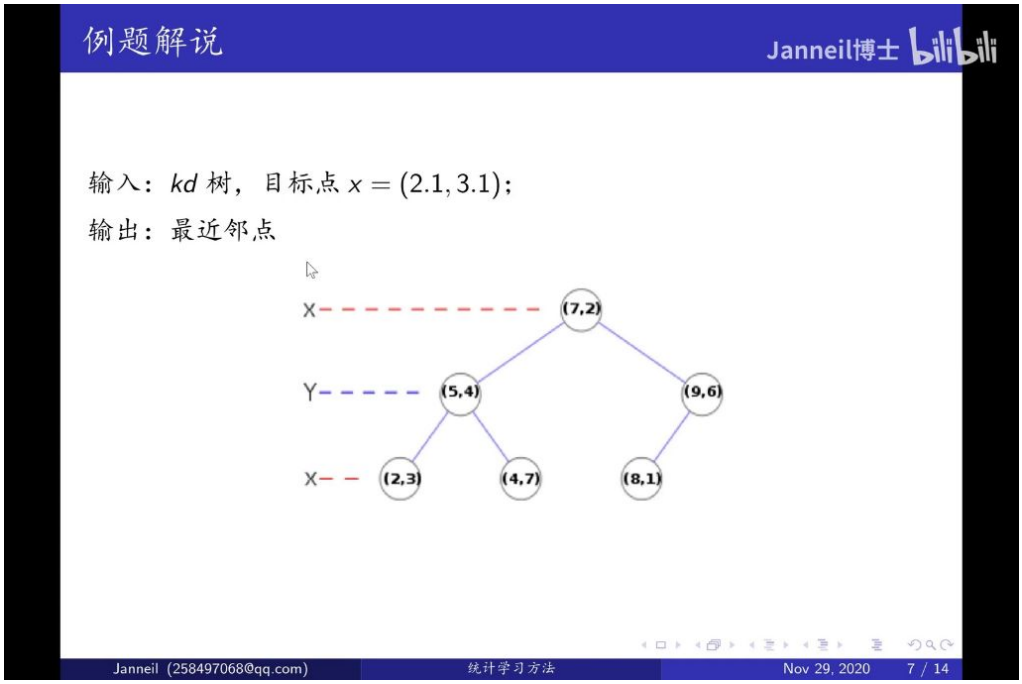
3）搜索kd树 52:15

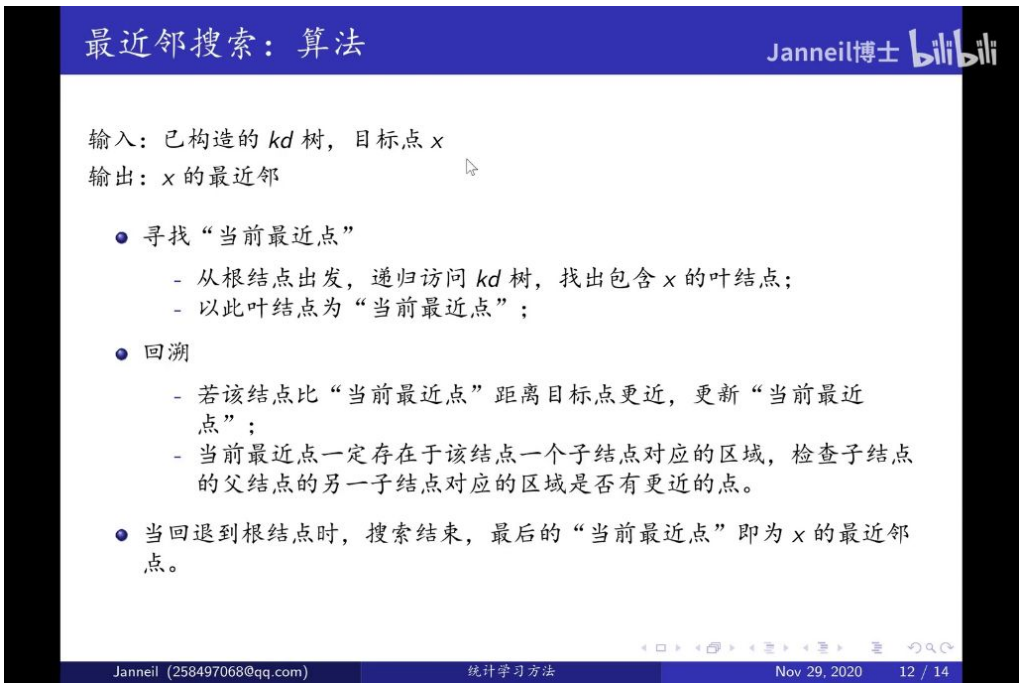
- 最近邻搜索 52:38



- **算法组成**：包含两个核心步骤：寻找当前最近点（初始值确定）和回溯（迭代优化）。
- **当前最近点**：
    - 作用：作为算法迭代的初始基准点
    - 确定方法：从根节点递归访问kd树，找到包含目标点x的叶节点
    - 示例：目标点(2.1,3.1)的初始最近点是叶节点(2,3)
- **回溯过程**：
    - 方向：从叶节点向根节点回溯
    - 更新条件：当发现更近节点时（距离$d_{new} < d_{current}$）
    - 终止条件：回退到根节点时结束
    - 几何验证：通过超球面（二维时为圆）与划分超平面的交集判断潜在近邻区域
- **高维扩展**：在n维空间时，距离判断使用超球面而非圆形

- 例题：二维空间最近邻搜索

- 

- 题目解析
  - **案例1（目标点(2.1,3.1)）**：
    - 搜索路径：根节点(7,2)→(5,4)→(2,3)
    - 回溯验证：超球面未与任何划分超平面相交，直接确认(2,3)为最近邻
  - **案例2（目标点(2,4.5)）**：
    - 初始最近点：(4,7)（距离3.2）
    - 回溯发现：兄弟节点(2,3)距离更近（1.5）
    - 最终确认：(2,3)为最近邻
  - **关键技巧**：
    - 超球面半径动态更新机制
    - 兄弟节点区域的必要性检查
    - 距离比较的数学计算过程
- 算法总结
- 

- **执行流程**:
  - 输入构造好的kd树和目标点x
  - 深度优先搜索确定初始最近点（叶节点）
  - 回溯时检查:
    - 兄弟节点区域与当前超球面的交集
    - 父节点对应区域的潜在更近点
- **效率优势**:
  - 避免全量数据遍历
  - 通过空间划分快速排除无关区域
- **实现要点**:
  - 递归访问的终止条件
  - 距离计算的优化方法
  - 回溯时的剪枝策略

二、知识小结

知识点	核心内容	考试重点/易混淆点	难度系数
k近邻法概念	基于k个最近邻居决定新实例类别，适用于分类与回归问题	**k值选择影响结果**（k=3与k=5可能不同类别）	★★
距离度量	欧式距离（p=2）、曼哈顿距离（p=1）、切比雪夫距离（p=∞）	**不同p值导致最近邻变化**（例：p≥3时x3为最近邻）	★★★
分类决策规则	多数表决：k个邻居中占比最高的类别为新实例类别	**噪声敏感性问题**（k过小易受噪声影响）	★★
误差率分析	最近邻法（k=1）误差率上界为2倍贝叶斯误差率，k增大时趋近贝叶斯误差率	**大样本下k近邻法接近最优决策**	★★★★
kd树构造	递归划分特征空间：选择方差最大特征→中位数切分→生成左右子树	**偶数个样本时中位数选择影响树结构**	★★★
kd树搜索	1. 从根节点递归到叶节点找当前最近点；2. 回溯检查超球面内更近点	**回溯时需检查兄弟节点区域**（例：圆与超平面相交则需比较）	★★★★