

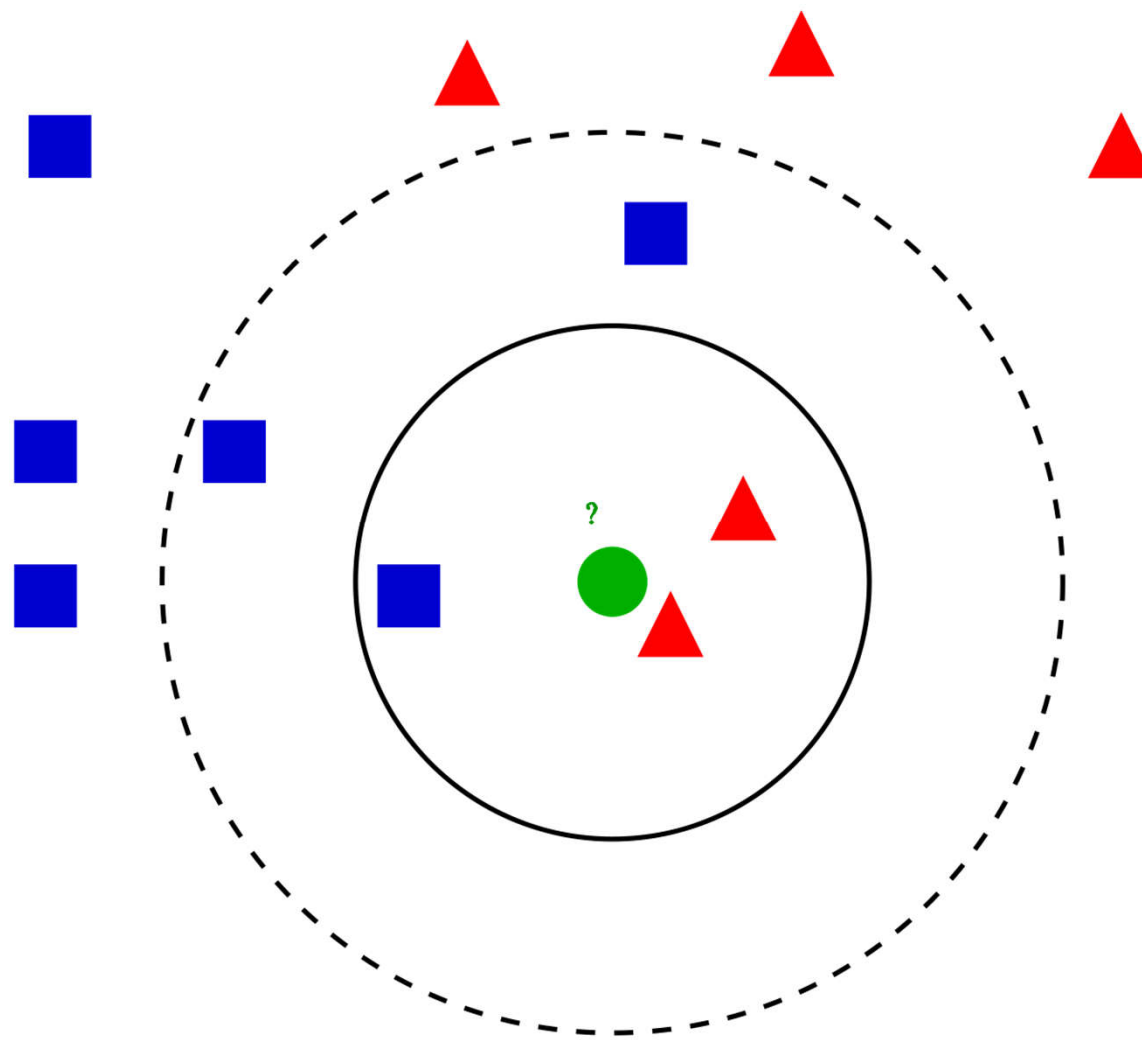
# k 近邻法

Peng Li

<https://simplelp.github.io/>

2019/06/03

# K 近邻算法



# K 近邻算法

## 算法 3.1 ( $k$ 近邻法)

输入：训练数据集

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

其中， $x_i \in \mathcal{X} \subseteq \mathbf{R}^n$  为实例的特征向量， $y_i \in \mathcal{Y} = \{c_1, c_2, \dots, c_K\}$  为实例的类别， $i = 1, 2, \dots, N$ ；实例特征向量  $x$ ；

输出：实例  $x$  所属的类  $y$ 。

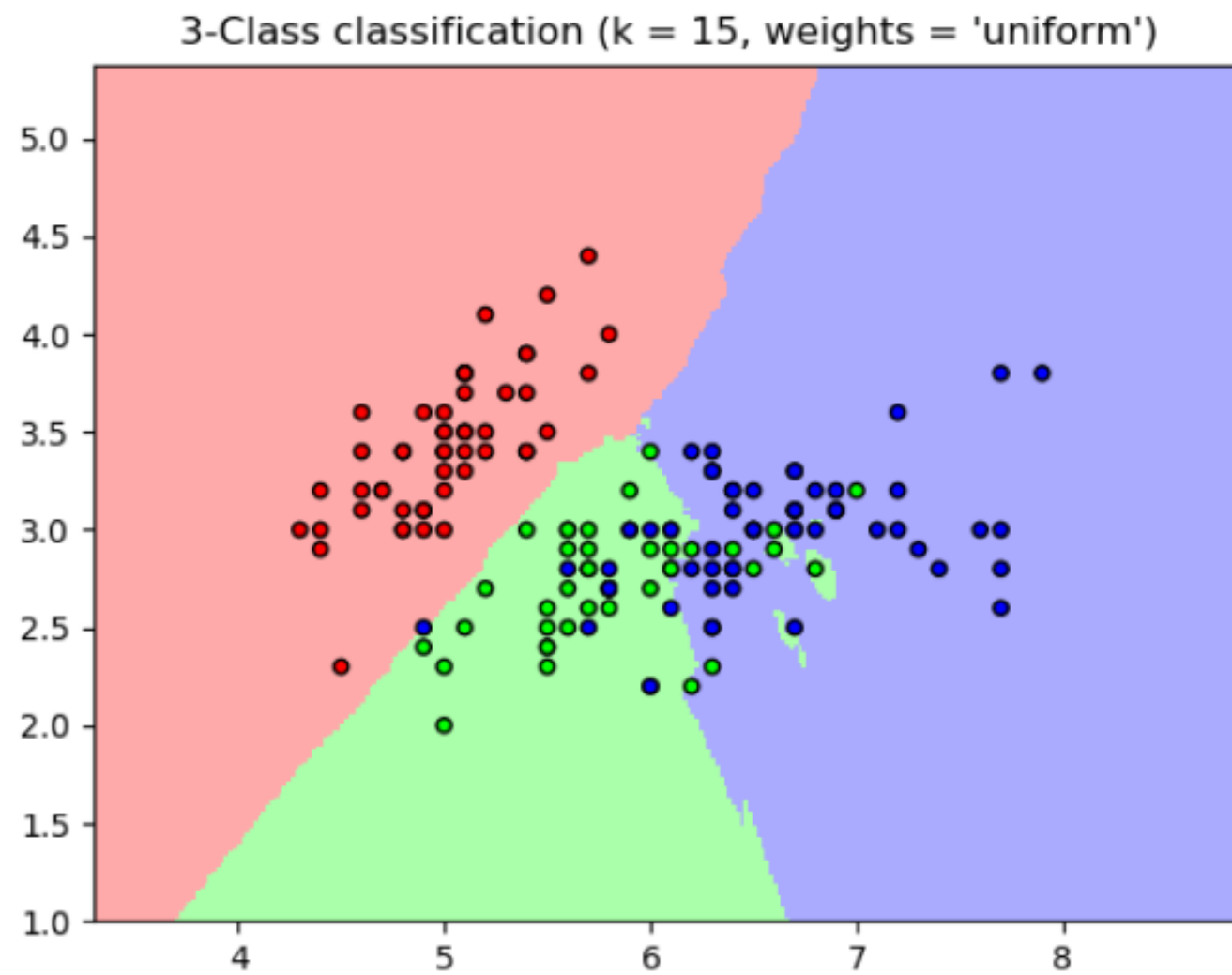
(1) 根据给定的距离度量，在训练集  $T$  中找出与  $x$  最邻近的  $k$  个点，涵盖这  $k$  个点的  $x$  的邻域记作  $N_k(x)$ ；

(2) 在  $N_k(x)$  中根据分类决策规则（如多数表决）决定  $x$  的类别  $y$ ：

$$y = \arg \max_{c_j} \sum_{x_i \in N_k(x)} I(y_i = c_j), \quad i = 1, 2, \dots, N; \quad j = 1, 2, \dots, K \quad (3.1)$$

式 (3.1) 中， $I$  为指示函数，即当  $y_i = c_j$  时  $I$  为 1，否则  $I$  为 0. ■

# K 近邻算法

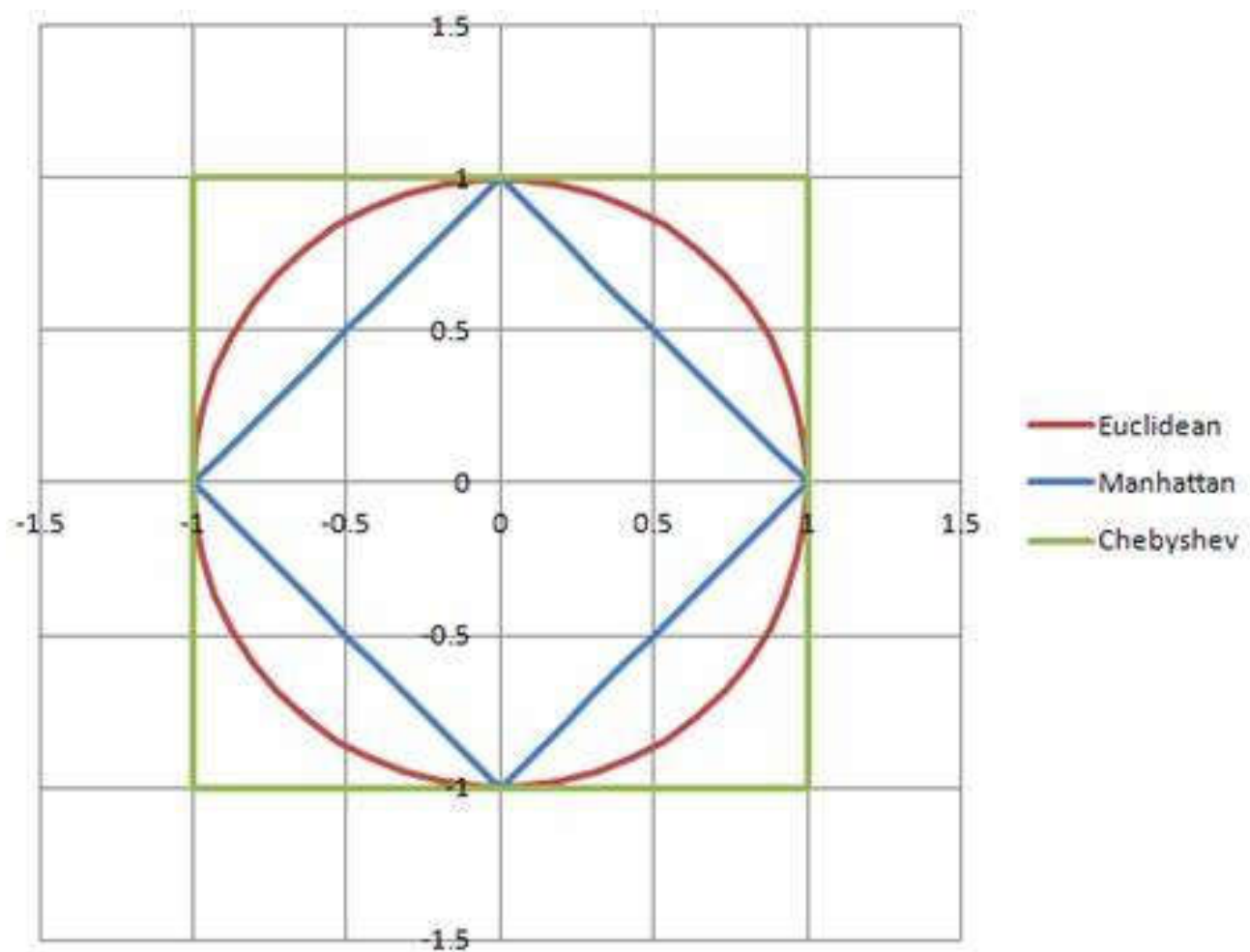


# K 近邻算法三要素

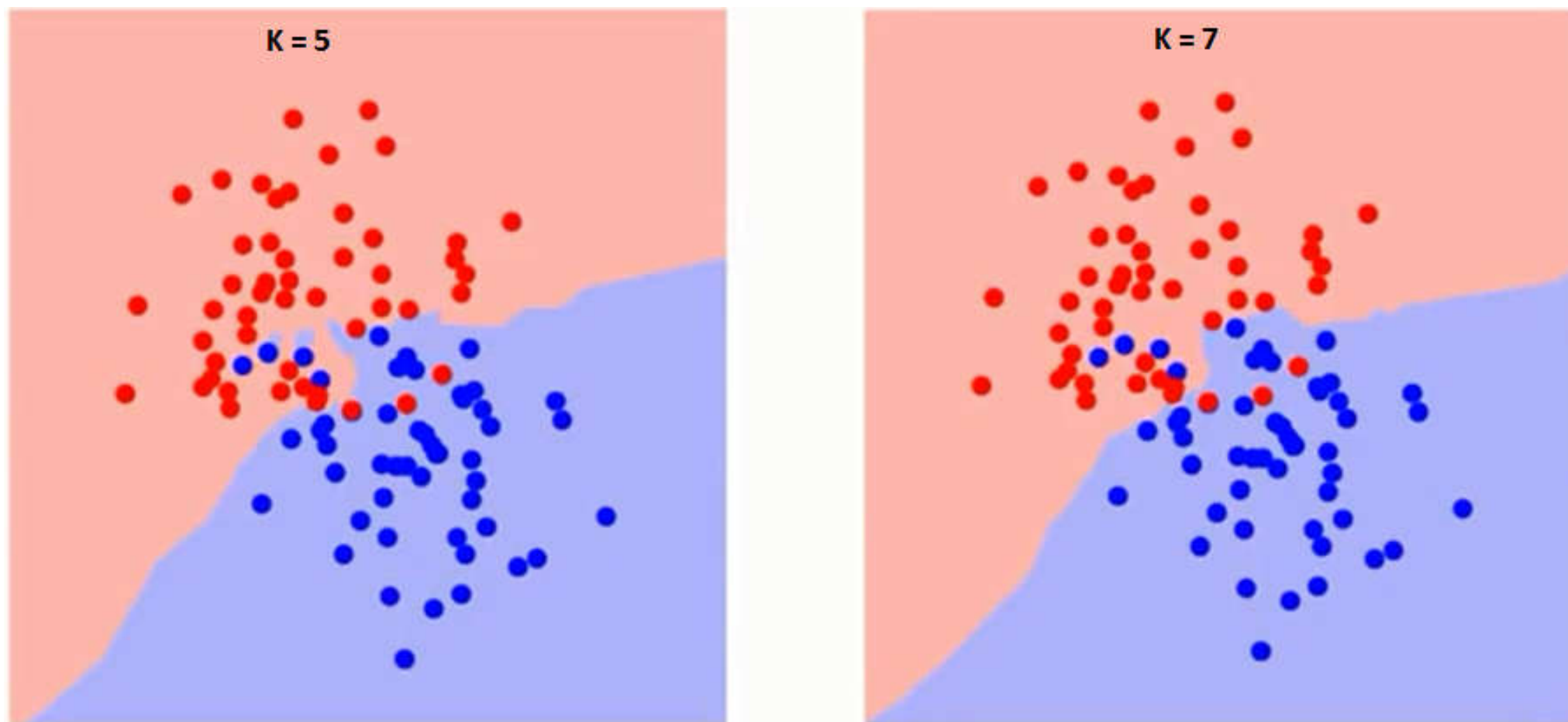
- 距离度量
- K值的选择
- 分类决策规则

# 距离度量

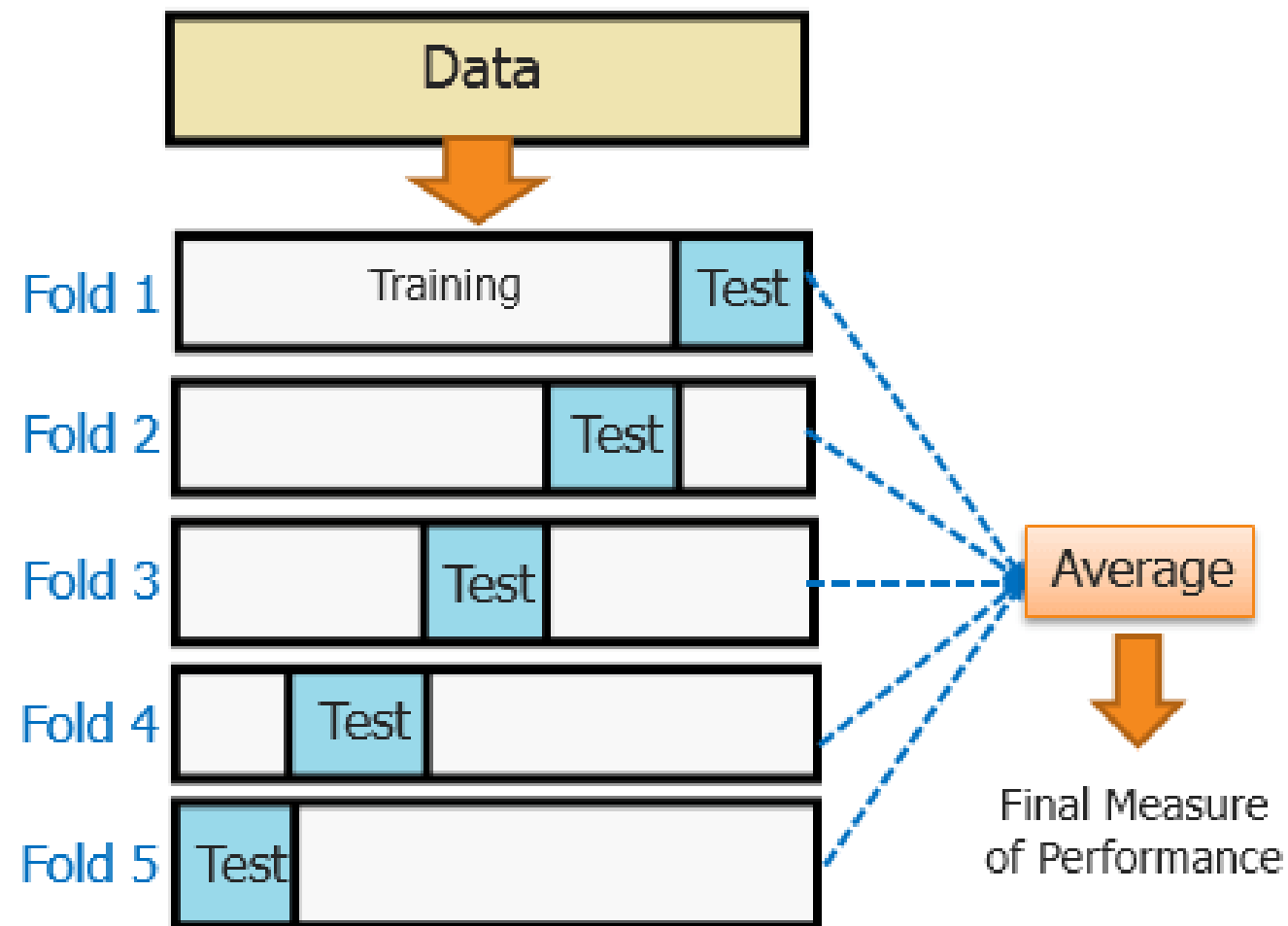
$$L_p(x_i, x_j) = \left( \sum_{l=1}^n |x_i^{(l)} - x_j^{(l)}|^p \right)^{\frac{1}{p}}$$



# K值的选择



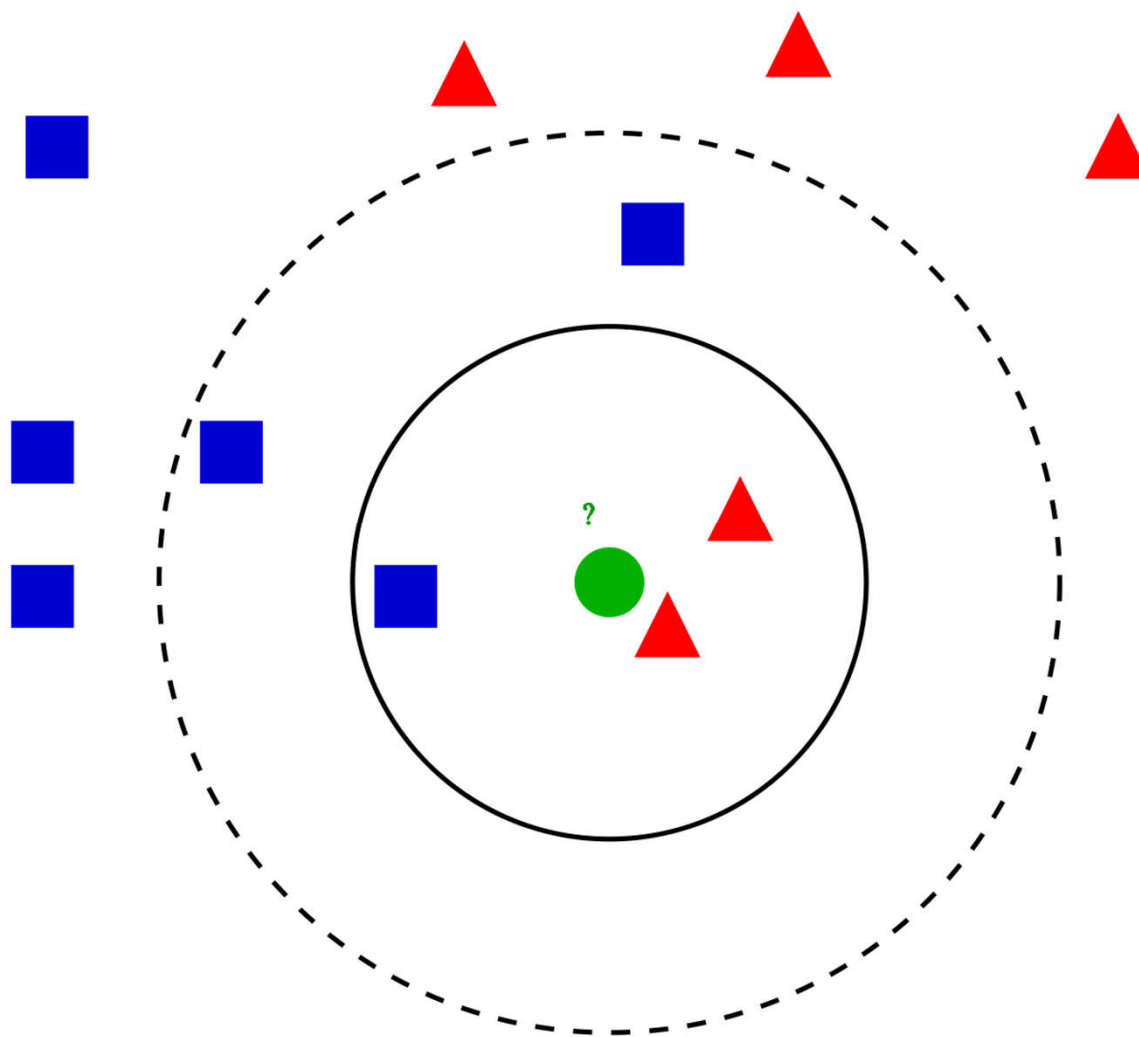
# K值的选择



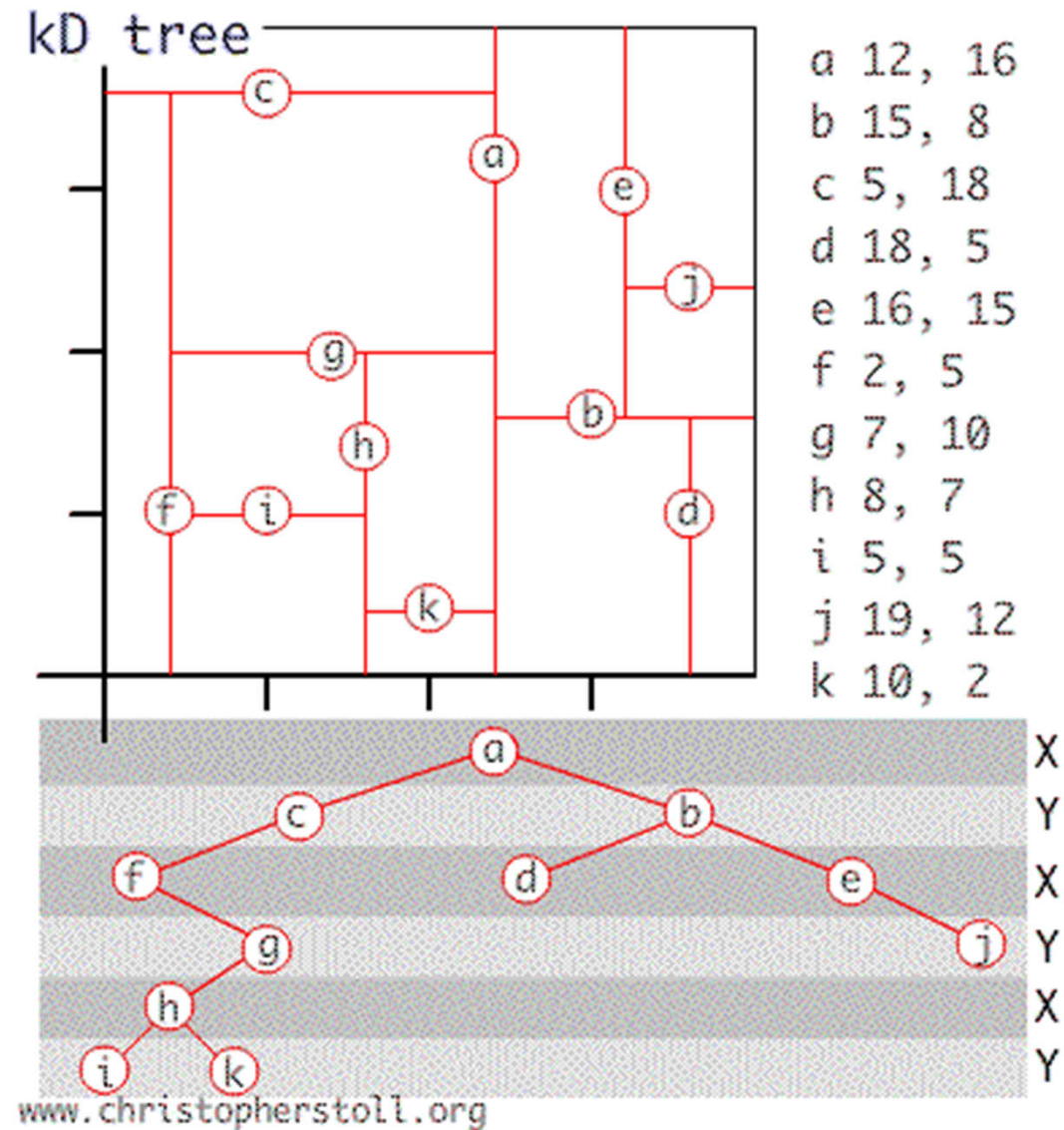


# 分类决策规则

- 多数表决规则



# K近邻法的实现： kd树



# 思考题

- Kd树的建立与搜索（重点）
- kNN算法的复杂度
- kNN算法距离度量上的改进方式
- kNN算法的实现

**Thanks !**