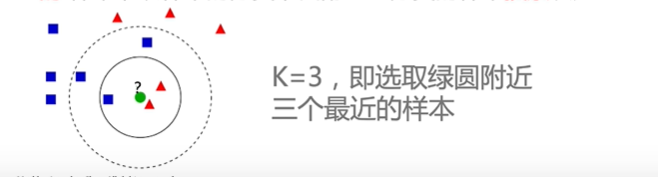
# KNN(K-Nearest-Neighbors Algorithm)

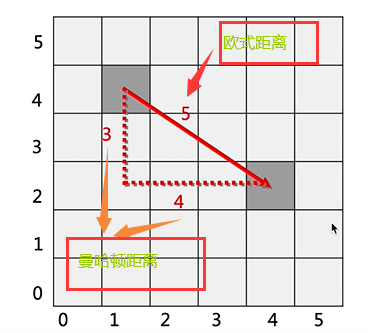
## **KNN简介**

简称：K近邻算法，属于监督学习，中心思想是找到未分类的测试样本附近K个**最相近的 已分类的** 样本，按照附近已分类的样本**投票**决定**。**



**图1：KNN示意图**

**如何计算距离：**



**图2：距离计算示意图**

**K值如何选择？**

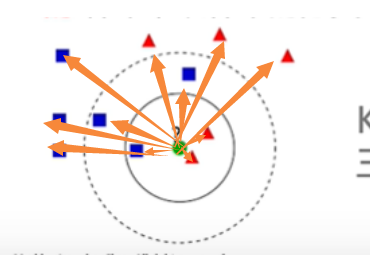
1. Ｋ为＂经验值＂。
2. Ｋ通常取奇数。
3. 一般情况下，，ｎ为样本总数
4. 当ｎ较少时，Ｋ可以尝试。
5. 当ｎ较大时，

**Ｋ值过大或者过小会产生的影响**：

1. 如果Ｋ过小取１的话，预测结果对近邻的实例点非常敏感，容易受到噪声的影响，发生过拟合。
2. Ｋ值过大的话，可以避免上述情况，但是现有训练集误差会增大，输入不相似的训练实例会导致预测错误。

## **算法流程**

1. 选择距离公式：欧式距离等等。
2. 计算未分类样本点与其它所有已分类样本之间的距离。
3. 针对计算出的距离　进行升序排序。
4. 选择前Ｋ个距离的最短的点。
5. 前Ｋ个距离最短的点中，未分类的样本点所属分类，由附近已分类的样本投票决定。



**图3：计算示意图**

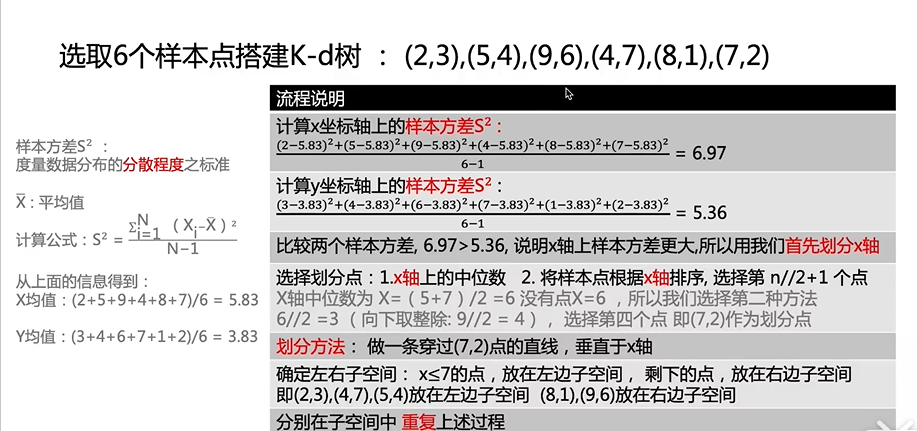
## 缺陷

1. 缺陷１：计算未分类样本点与所有已分类点距离，再进行排序计算量非常大。

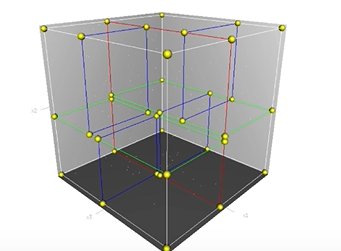
***解决方法***：采用Ｋ-Ｄ（K-Dimension Tree）树数据结构，优化搜索操作，减少计算量。

***原理***：对数据点在K维空间中划分的一种数据结构，K-d树是一种空间划分树，把整个空间划分为特定的几个部分，然后在特定空间的部分内进行相关搜索操作。

1. ***D树的构建：***

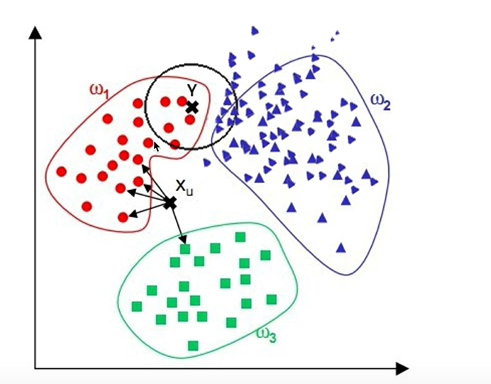


**图4：p-d树流程**

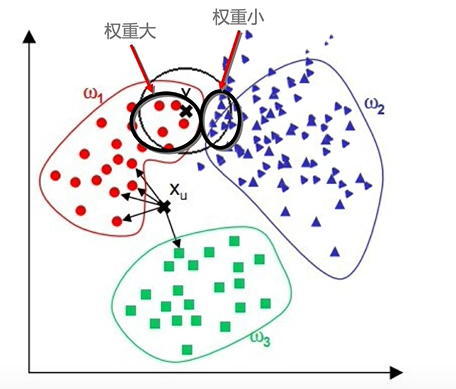


**图5：p-d树示意图**

1. 缺陷２：样本不平衡问题会影响分类结果



**图6：权重示意图**



**图7：权重示意图**

***解决方法***：采用距离1/d权重法解决。

https://github.com/hehaiyang111/MLAlgorithm/tree/main/KNN