
数据挖掘

第4章 分类-最近邻分类器

教师：王东京

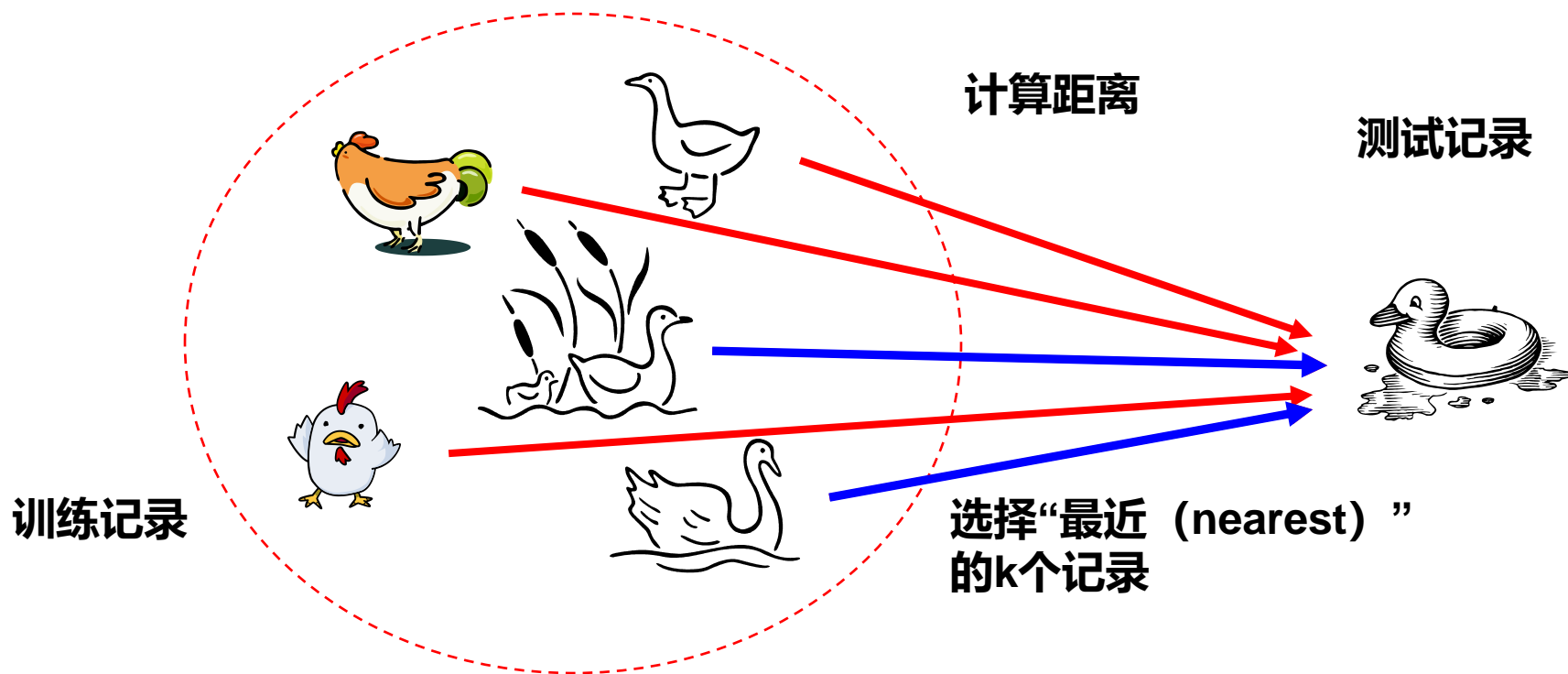
学院：计算机学院

邮箱：dongjing.wang@hdu.edu.cn

最近邻分类器Nearest Neighbor Classifiers

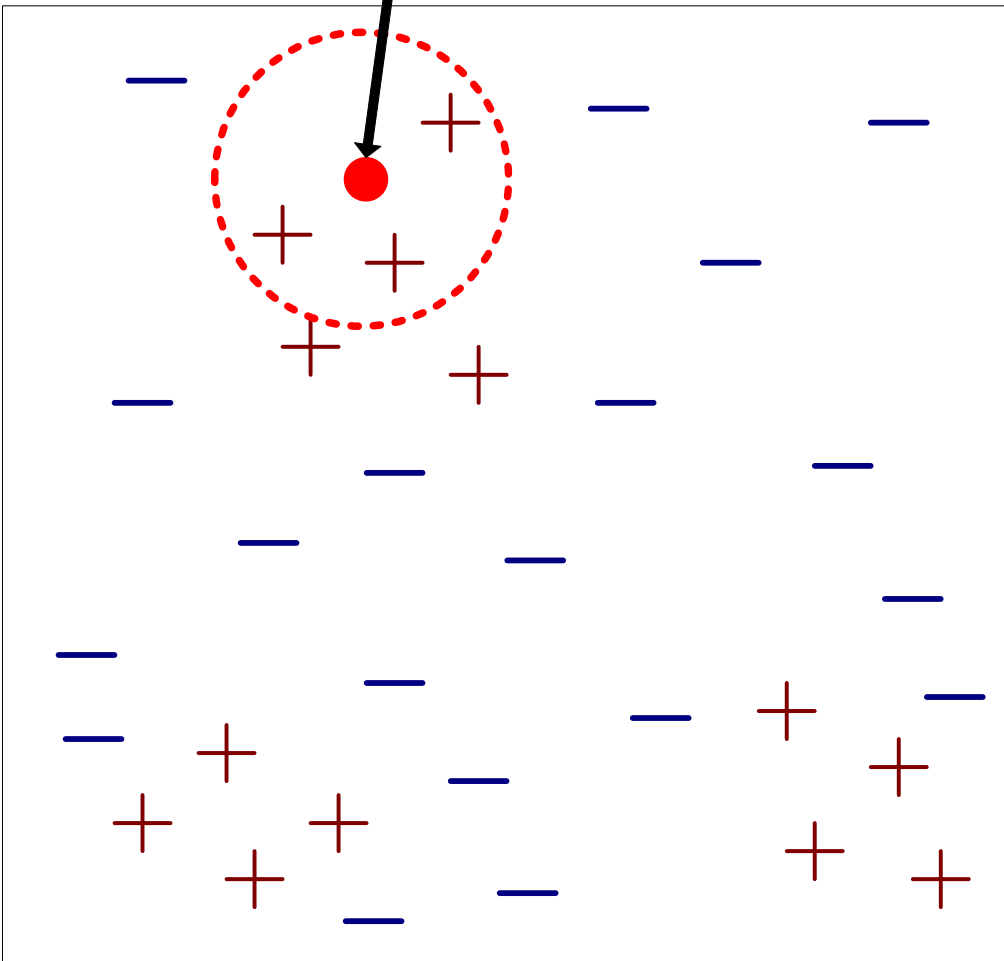
基本想法:

- 鸭子?
- 如果它走路像鸭子，嘎嘎叫像鸭子，看起来也像鸭子，那它可能就是只鸭子



Nearest-Neighbor Classifiers

Unknown record



◆ 需要三个要素

- 已经标记好的记录 (record) 集
- 距离度量标准, 用于计算记录之间的距离
- K 的值, 即要检索的最近邻居的数量

◆ 对未知记录进行分类:

- 计算与其他训练记录的距离
- 识别 k个最近的邻居
- 使用最近邻居的类别标签来确定未知记录的类别标签 (例如, 以投票的方式)

Nearest Neighbor Classification

计算两点之间的接近度 (proximity) :

- 例如: 欧氏距离

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

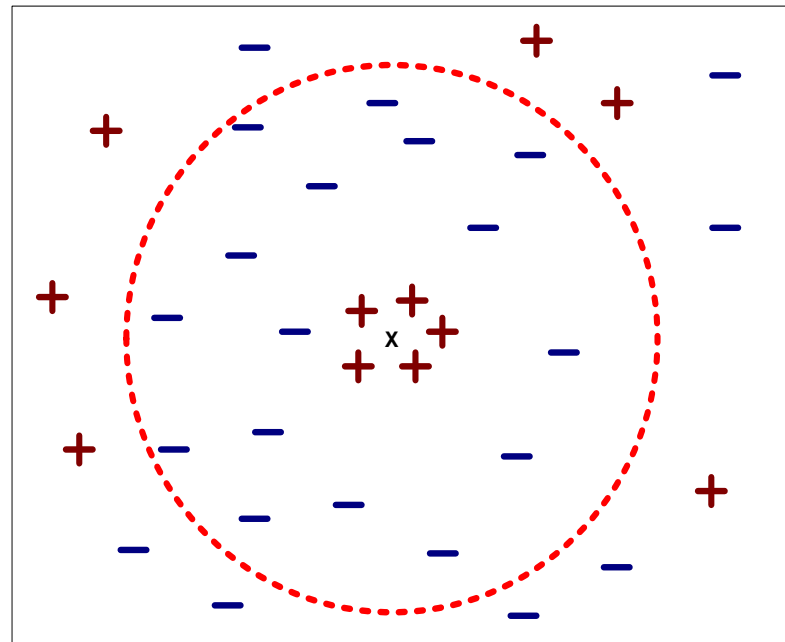
根据最近邻列表确定类别

- 选择 k 个最近邻中类别标签最多的
- 根据距离对投票结果进行加权
 - ◆ weight factor, $w = 1/d^2$

Nearest Neighbor Classification...

选择 k 值:

- 如果 k 太小
 - ◆ 则对噪声点敏感
- 如果 k 太大
 - ◆ 则邻域可能包含其他类别的点



Nearest Neighbor Classification...

相近度 (proximity) 的选择很重要

- For documents, **cosine** is better than correlation or Euclidean

1 1 1 1 1 1 1 1 1 1 1 0



VS

0 0 0 0 0 0 0 0 0 0 0 1

0 1 1 1 1 1 1 1 1 1 1 1

1 0 0 0 0 0 0 0 0 0 0 0

Euclidean distance = 1.4142 for both pairs

Nearest Neighbor Classification...

— 一般需要数据预处理

- 可能必须对属性进行缩放，以防止距离度量值被其中一个属性支配

◆ 示例:

- height of a person may vary from 1.5m to 1.8m
- weight of a person may vary from 90lb to 300lb
- income of a person may vary from \$10K to \$1M

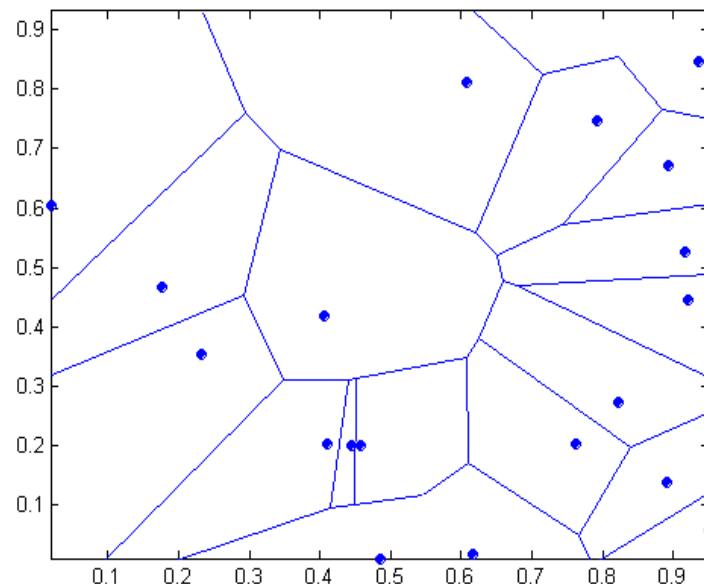


- 时间序列通常标准化为：均值为0，而标准偏差为1

Nearest-neighbor classifiers

- ◆ 最近的邻居分类器是**局部 (local) 分类器**
- ◆ 它们可以产生**任意形状的决策边界 (decision boundary)**。

1-nn 的决策边界是一个沃罗诺伊图
(Voronoi Diagram)



Nearest Neighbor Classification...

如何处理训练和测试集中的缺失值？

- 接近度计算通常要求所有属性都存在
- 一些方法使用两个实例中存在的属性子集
 - ◆这可能不会产生好的结果，因为它对每对实例样本使用了不同的接近度度量
 - ◆因此，邻近度变成不可比的

Nearest Neighbor Classification...

无关和多余的属性

- 不相关的属性会给邻近度量增加噪音
- 冗余属性使接近度偏向某些属性

如何处理？

- 可以使用变量选择或降维来解决不相关和冗余的属性

改进 KNN 的效率

KNN属于消极学习方法，而之前的决策树属于积极学习方法。

- 消极学习方法不需要建立模型，但是测试时开销很大。

避免必须计算到训练集中所有对象的距离

- 多维访问方法 (k-d树)
- 快速近似相似搜索
- 局部敏感哈希 (Locality Sensitive Hashing, LSH)

压缩 (Condensing)

- 确定提供相同性能的较小对象集

编辑 (Editing)

- 移除对象以提高效率

下述关于最近邻分类器（KNN）的说法**错误**的是？

- ☐ A KNN使用具体的训练实例进行预测，不需要维护源自数据的抽象模型
- ☐ B 当k值很小时，KNN对噪声很敏感
- ☒ C 相比决策树等积极学习方法，KNN的训练速度慢，但测试速度更快，开销低，消耗资源更少

提交