大数据基础与应用
BIG DATA FOUNDATION AND APPLICATION

数据分析算法

北京理工大学计算机学院　孙新

2019年1月

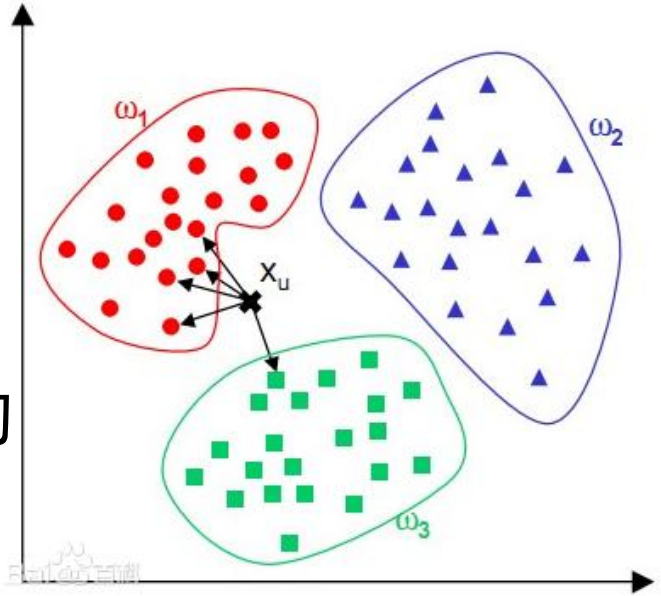
目　录

4、经典的机器学习方法

- 4.1 分类算法原理
- 4.2 决策树算法
- 4.3 K-近邻分类算法（KNN算法）
- 4.4 K-均值聚类算法（K-means算法）
- 4.5 Apriori关联规则算法

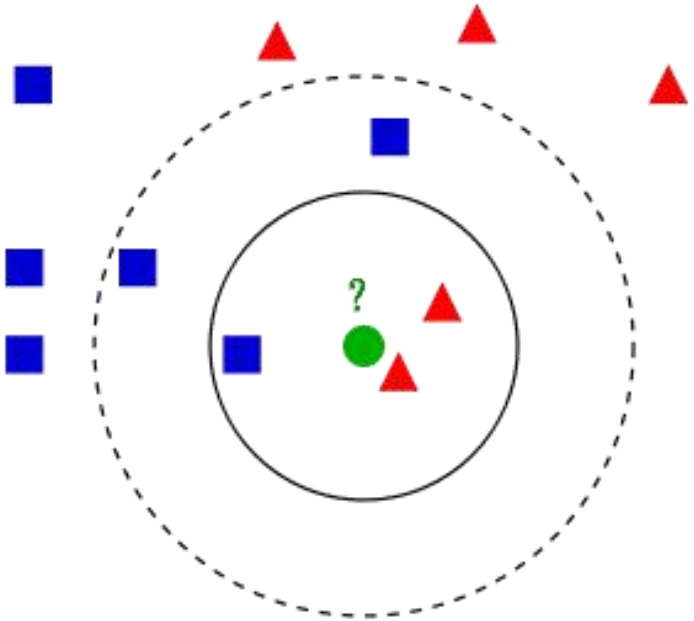
4.3 K-近邻分类算法（KNN算法）

- K近邻(KNN，k-NearestNeighbor)分类算法是分类技术中最简单的方法之一
  - 所谓K近邻，就是k个最近的邻居的意思，说的是每个样本都可以用它最接近的k个邻居来代表。
  - kNN算法的核心思想是如果一个样本在特征空间中的k个最相邻的样本中的大多数属于某一个类别，则该样本也属于这个类别，并具有这个类别上样本的特性。



KNN算法思想

- 如图所示，有两类不同的样本数据，分别用蓝色的小正方形和红色的小三角形表示问题：图中的绿色的圆属于哪一类？
  - 如果K=3，绿色圆点的最近的3个邻居是2个红色小三角形和1个蓝色小正方形，少数从属于多数，基于统计的方法，判定绿色的这个待分类点属于红色的三角形一类。
  - 如果K=5，绿色圆点的最近的5个邻居是2个红色三角形和3个蓝色的正方形，还是少数从属于多数，基于统计的方法，判定绿色的这个待分类点属于蓝色的正方形一类。



- 给定一个未知样本，k-最临近分类法搜索模式空间，找出最接近未知样本的k个训练样本；然后使用k个最临近者中最公共的类来预测当前样本的类标号
  - 产生训练集，使得训练集按照已有的分类标准划分成离散型数值类，或者是连续型数值类输出。
  - 以训练集的分类为基础，对测试集每个样本寻找K个近邻，采用欧式距离作为样本间的相似程度的判断依据，相似度大的即为最近邻。一般近邻可以选择1个或者多个
  - 当类为连续型数值时，测试样本的最终输出为近邻的平均值；当类为离散型数值时，测试样本的最终为近邻类中个数最多的那一类。

KNN算法步骤

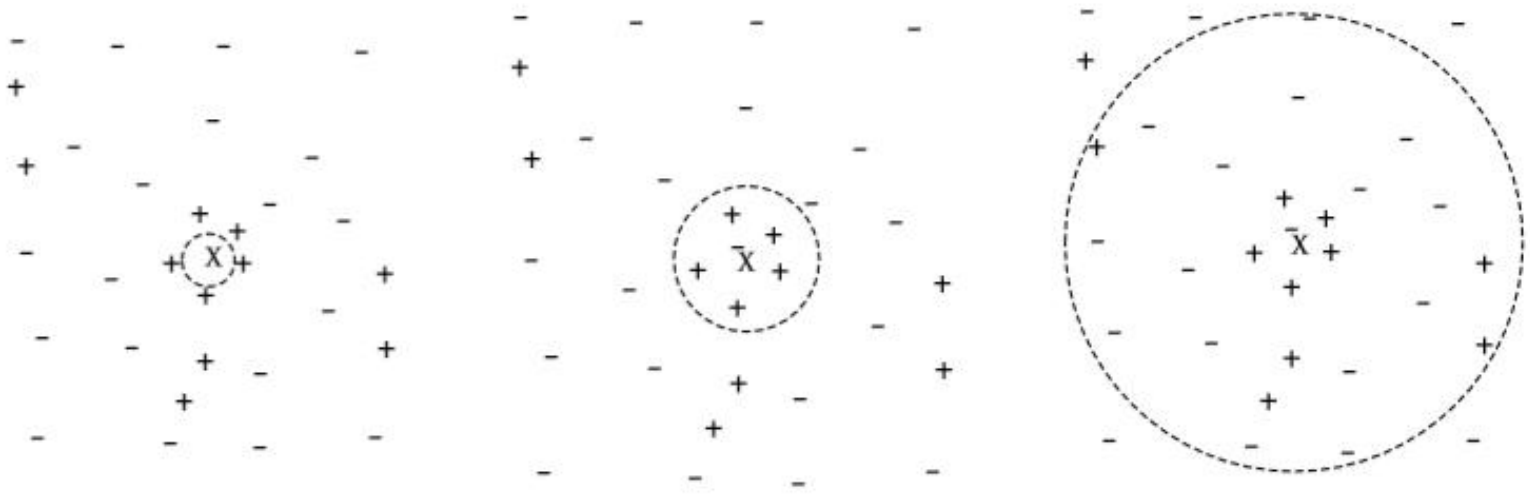
step.1---初始化距离为最大值
step.2---计算未知样本和每个训练样本的距离dist
step.3---得到目前K个最临近样本中的最大距离maxdist
step.4---如果dist小于maxdist，则将该训练样本作为K-最近邻样本
step.5---重复步骤2、3、4，直到未知样本和所有训练样本的距离都算完
step.6---统计K个最近邻样本中每个类别出现的次数
step.7---选择出现频率最大的类别作为未知样本的类别

- K 值的选择
- 距离度量
- 类别的判定

- K 值的选取对KNN学习模型有很大的影响，
  - 若K值过小，得到的近邻数过少，会降低分类精度，同时也会放大噪声数据的干扰
  - 若k值选择过大，会有较大的邻域训练样本进行预测，可以减少噪音样本点，但是距离较远的训练样本对预测结果会有贡献，以至于实际上并不相似的数据也可能被包含进来，造成噪声增加而导致分类或者预测效果的降低

KNN算法——K 值的选择

- 下图给出K值的选取对于预测结果的影响



(a) Neighborhood too small.

(b) Neighborhood just right.

(c) Neighborhood too large.

K 值的选择

- 通常，K值的设定采用交叉检验的方式（以K=1为基准）
- 经验规则：K一般低于训练样本数的平方根。
  - 交叉验证的概念：有时亦称循环估计， 是一种统计学上将数据样本切割成较小子集的实用方法。于是可以先在一个子集上做分析， 而其它子集则用来做后续对此分析的确认及验证。 开始的子集被称为训练集。而其它的子集则被称为验证集或测试集。
  - 交叉验证误差统计选择法就是比较不同K值时的交叉验证平均误差率，选择误差率最小的那个K值。例如选择K=1,2,3,…， 对每个K=i做100次交叉验证，计算出平均误差，然后比较、选出最小值。

- K 值的选择介绍之后，我们再来看：
- 距离度量
  - 计算距离有许多种不同的方法，如欧氏距离、余弦距离、汉明距离、曼哈顿距离等等，传统上，kNN算法采用的是欧式距离。
  - 也称欧几里得距离，它是一个采用的距离定义，它是在维空间中两个点之间的真实距离。

$$d=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

- 类别的判定，往往是多数表决，即由输入实例的 K 个最临近的训练实例中的多数类决定输入实例的类别
  - 投票决定：少数服从多数，近邻中哪个类别的点最多就分为该类
  - 如果训练数据大部分都属于某一类，投票算法就有很大问题了。这时候就需要考虑设计每个投票者票的权重了
  - 加权投票法：根据距离的远近，对近邻的投票进行加权，距离越近则权重越大（权重为距离平方的倒数）
  - 若样本到测试点距离为d，则选1/d为该邻居的权重（也就是得到了该邻居所属类的权重），接下来统计统计k个邻居所有类标签的权重和，值最大的那个就是新数据点的预测类标签。

4.3 KNN算法

算法优点：

（1）简单，易于理解，易于实现，无需估计参数，无需训练；

（2）适合对稀有事件进行分类；

（3）特别适合于多分类问题(multi-modal,对象具有多个类别标签)

（4）对于类域的交叉或重叠较多的待分样本集来说，KNN方法较其他方法更为适合。

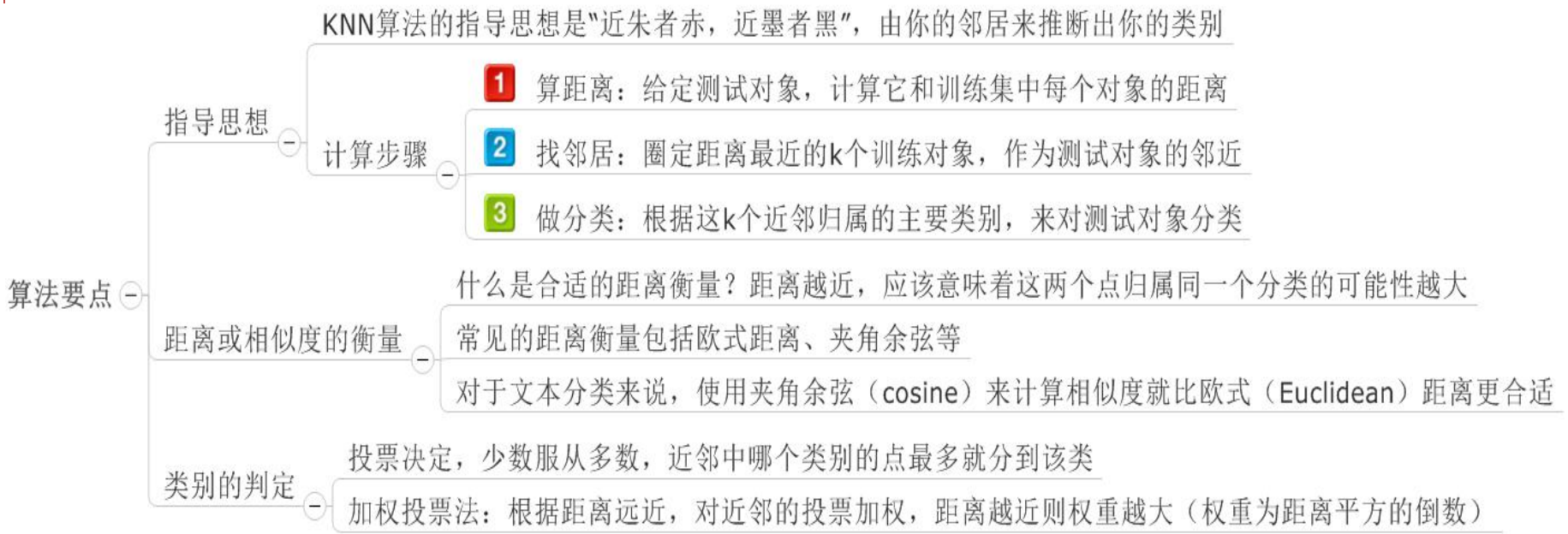
算法的不足：

（1）需要存储全部训练样本，计算量较大

（2）可解释性较差，无法给出决策树那样的规则。

（3）当样本不平衡时（如一个类的样本容量很大，而其他类样本容量很小时）有可能导致当输入一个新样本时，该样本的K个邻居中大容量类的样本占多数。

4.3 KNN算法


算法要点
指导思想
KNN算法的指导思想是"近朱者赤，近墨者黑"，由你的邻居来推断出你的类别
计算步骤
1 算距离：给定测试对象，计算它和训练集中每个对象的距离
2 找邻居：圈定距离最近的k个训练对象，作为测试对象的邻近
3 做分类：根据这k个近邻归属的主要类别，来对测试对象分类
距离或相似度的衡量
什么是合适的距离衡量？距离越近，应该意味着这两个点归属同一个分类的可能性越大
常见的距离衡量包括欧式距离、夹角余弦等
对于文本分类来说，使用夹角余弦（cosine）来计算相似度就比欧式（Euclidean）距离更合适
类别的判定
投票决定，少数服从多数，近邻中哪个类别的点最多就分到该类
加权投票法：根据距离远近，对近邻的投票加权，距离越近则权重越大（权重为距离平方的倒数）