K近邻分类器

翟婷婷

扬州大学
信息工程（人工智能）学院
zhtt@yzu.edu.cn

2023春

课程目标

- 理解k-近邻分类的原理，能够熟练运用k-近邻方法进行分类。
- 了解k-近邻方法的限制、缺陷以及可能的解决办法。
- 掌握常见的特征归一化方法，会实际应用。

引言

- 前面我们一直运用的是贝叶斯分类器：根据后验概率做出分类决策。
- 我们讨论了如何使用训练数据集对贝叶斯分类器中的先验概率和类条件概率分布进行估计。
- 本节我们研究新的分类技术，不需要对数据中的概率结构进行估计。

最近邻分类器(Nearest Neighbour classifier)

- 给定一个训练样本集$\{(\boldsymbol{x}_1, y_1), \cdots (\boldsymbol{x}_n, y_n)\}$，其中$\boldsymbol{x_i} \in \mathbb{R}^d$称为训练实例(sample, instance)，$y_i \in \{1,2 \cdots c\}$称为实例类标(label)。
- 给定一个距离度量函数$d(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}$：能度量两个实例$\boldsymbol{x}$和$\boldsymbol{y}$之间的距离，或不相似的程度。
- 最近邻分类规则：

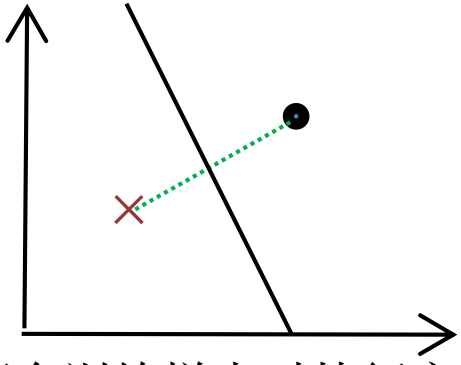
  对测试实例$\boldsymbol{x}$，将$\boldsymbol{x}$指派到训练样本集中与$\boldsymbol{x}$距离最近的那个实例所在的类别中：

  ① $i^* = \underset{i=1,\cdots n}{\operatorname{argmin}} d(\boldsymbol{x}, \boldsymbol{x}_i)$

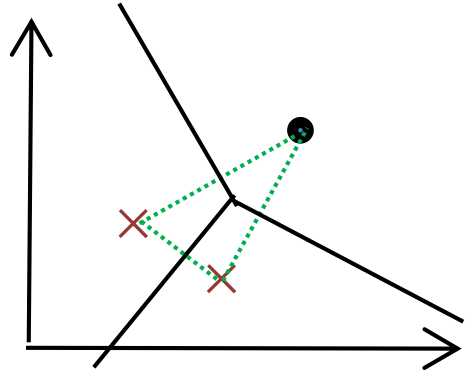
  ② 输出对$\boldsymbol{x}$分类的预测y_{i^*}

最近邻分类器的原理

- 最近邻分类器不需要训练分类器。
- 最近邻分类器隐含地利用训练样本集将整个特征空间划分成了很多个小区域，使得每个区域中只包含一个训练实例，然后把每个区域中包含的那个训练实例的类标作为该区域中所有点的类标记。
- 一个待分类的实例点落在哪个区域中，它到该区域中包含的训练实例的距离均小于它到其它训练实例的距离。



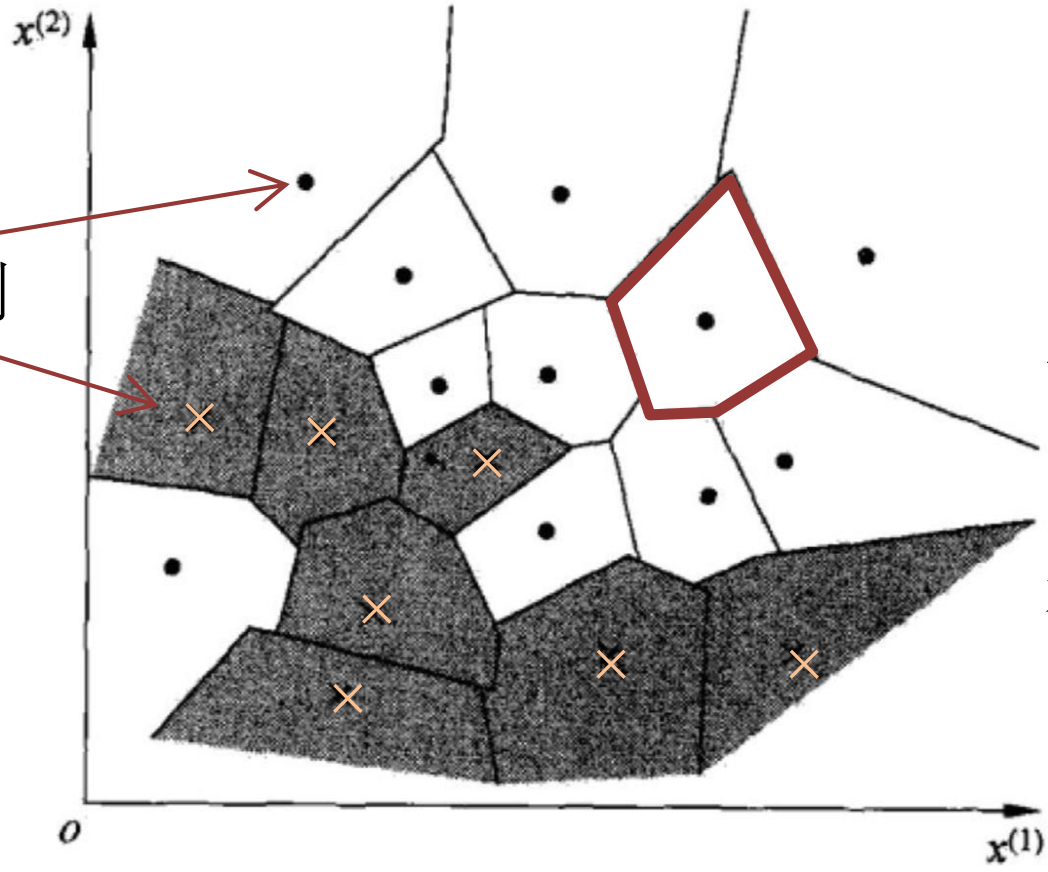
只有两个训练样本时特征空间的划分



三个训练样本时特征空间的划分

最近邻分类器的原理

- 在二维特征空间中，有很多个训练样本时，最近邻分类器对应的特征空间的一个划分：



训练实例

测试实例位于哪个区域中，就把它指派到该区域中的训练实例所在的类别中。

距离函数(distance function)

- 最近邻分类规则依赖于一个度量点与点之间距离的距离函数$d(\boldsymbol{x},\boldsymbol{y})$。
- 一个距离度量函数$d(\cdot,\cdot)$必须满足如下的性质：

① 非负性：$d(\boldsymbol{x},\boldsymbol{y}) \geq 0,\ \forall \boldsymbol{x},\boldsymbol{y}$

② 自反性：$d(\boldsymbol{x},\boldsymbol{y}) = 0$ 当且仅当$\boldsymbol{x} = \boldsymbol{y}$。

③ 对称性：$d(\boldsymbol{x},\boldsymbol{y}) = d(\boldsymbol{y},\boldsymbol{x}), \forall \boldsymbol{x},\boldsymbol{y}$

④ 三角不等式：$d(\boldsymbol{x},\boldsymbol{y}) + d(\boldsymbol{y},\boldsymbol{z}) \geq d(\boldsymbol{x},z), \forall \boldsymbol{x},\boldsymbol{y},\boldsymbol{z}$

常见的距离函数

$$\boldsymbol{x} = (x_1, \ldots, x_d), \quad \boldsymbol{y} = (y_1, \ldots y_d)$$

- 闵可夫斯基度量(Minkowski metric)称为ℓ_p范数距离:

$$\ell_p(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^p\right)^{1/p}, \quad p \geq 1$$

- ℓ_2范数距离，就是欧式距离(Euclidean):

$$\ell_2(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^2\right)^{1/2} = \|\boldsymbol{x} - \boldsymbol{y}\|_2$$

- ℓ_1范数距离，也称曼哈顿距离，又称城市街区距离:

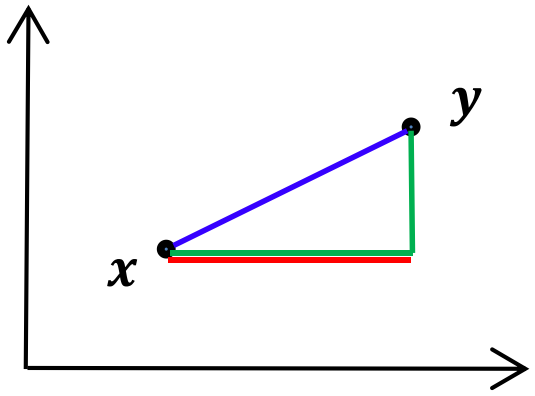
$$\ell_1(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} |x_i - y_i| = \|\boldsymbol{x} - \boldsymbol{y}\|_1$$

- ℓ_∞范数距离：各维距离的最大值:

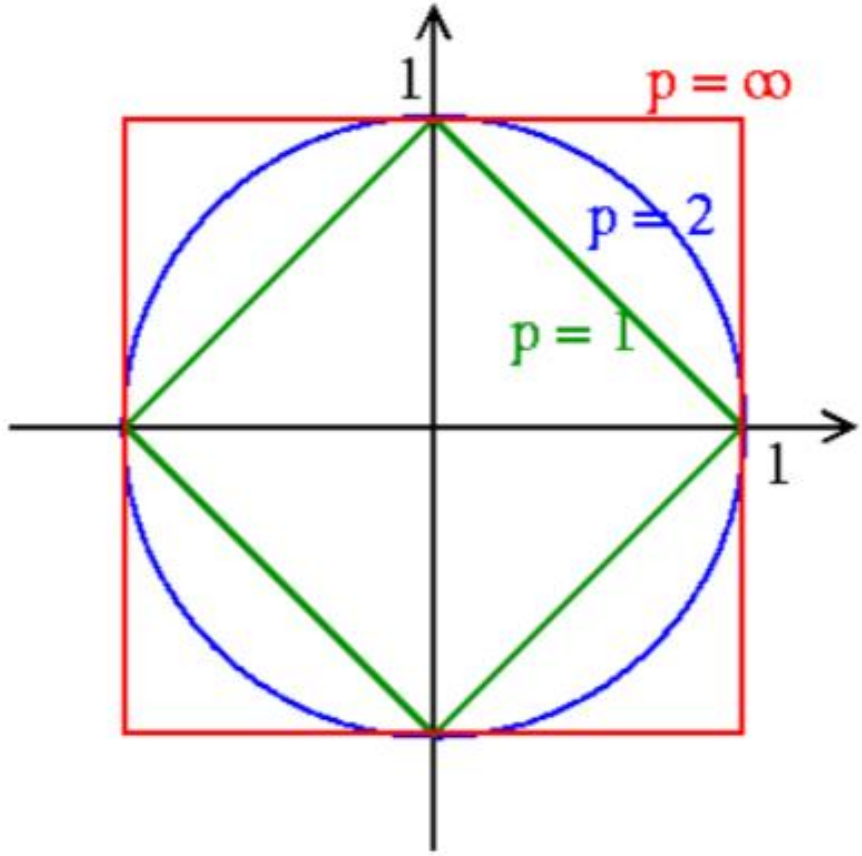
$$\ell_\infty(\boldsymbol{x}, \boldsymbol{y}) = \max_{i=1,\cdots d} |x_i - y_i| = \|\boldsymbol{x} - \boldsymbol{y}\|_\infty$$

常见的距离函数

- 左图：ℓ_p范数距离图示。
- 右图：距离原点的ℓ_p范数距离等于1的点的轨迹。



蓝线：$\ell_2(\boldsymbol{x},\boldsymbol{y})$距离
绿线：$\ell_1(\boldsymbol{x},\boldsymbol{y})$距离
红线：$\ell_\infty(\boldsymbol{x},\boldsymbol{y})$距离



常见的距离函数

- 余弦相似性度量：用两个向量之间夹角的余弦值来衡量两个向量间的相似度。

$$Simi(\boldsymbol{x},\boldsymbol{y}) = cos\theta = \frac{\boldsymbol{x}^{\top}\boldsymbol{y}}{\|\boldsymbol{x}\|_2 \cdot \|\boldsymbol{y}\|_2}$$

$$= \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} |x_i|^2} \cdot \sqrt{\sum_{i=1}^{d} |y_i|^2}}$$

夹角越小，余弦值越接近1，表明两个向量越相似。
相比距离度量函数，余弦相似度度量更加注重两个向量在方向上的差异，而不关注向量的长度。

- 其它距离度量：海明距离、编辑距离、Jaccard距离等。

最近邻分类器的错误率

- 给定一个训练样本集，选择一个合适的距离度量函数，就可以使用最近邻分类规则来进行分类。
- 最近邻分类器的分类性能有保证吗？

记$P_n(e)$为最近邻分类器在一个包含n个样本的训练集上取得的分类平均错误率；记$\lim_{n\to\infty} P_n(e) = P$；

记P^*为贝叶斯分类器在同一个训练集上取得的错误率；

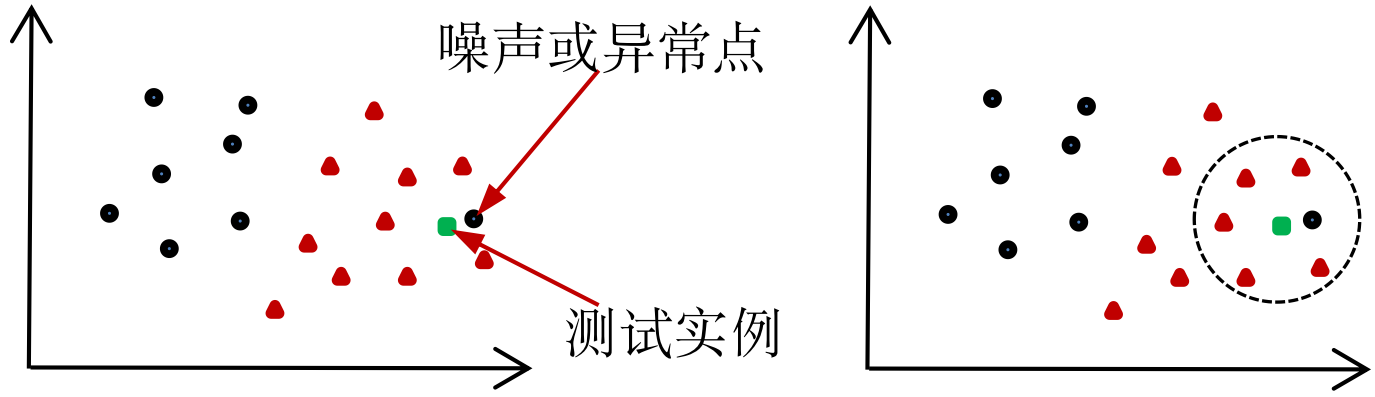
则最近邻分类器与贝叶斯分类器的错误率满足：

$$P^* \le P \le P^*\left(2 - \frac{c}{c-1}P^*\right) < 2P^*$$

其中c是类别总数。也就是说，当训练样本数足够多的情况下，最近邻分类器的分类错误率上界总小于 $2P^*$，当P^*非常小时，则逼近 $2P^*$。

K近邻分类器

- 最近邻分类规则只依赖于与测试实例距离最近的训练实例，很容易受到噪声或异常点数据的影响。



- 为了避免噪声或异常点的干扰，可以考虑与测试实例距离最近的前K个训练实例的类标。这就是K近邻分类规则的思想。

K近邻分类器

- 给定一个训练样本集，选择一个合适的距离函数。
- K近邻分类规则：

  对测试实例$\boldsymbol{x}$，找到与$\boldsymbol{x}$距离最近的前K个训练实例所对应的类标，将$\boldsymbol{x}$指派到这K个类标中的多数类中。

  形式化的描述：

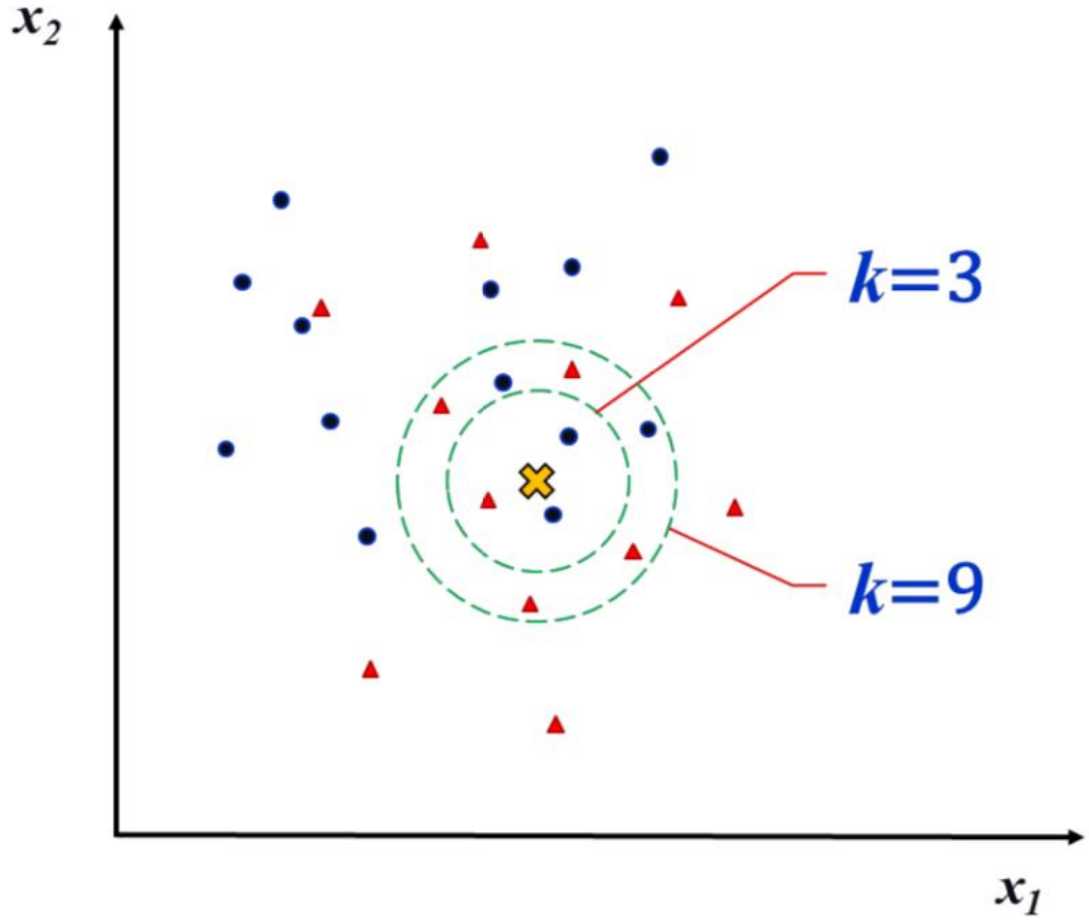
  记$N_K(\boldsymbol{x})$为与$\boldsymbol{x}$距离最近的前K个训练实例的集合；

  记n_i为$N_K(\boldsymbol{x})$中第i类样本的数目，$i = 1,2,\cdots c$；

  则将$\boldsymbol{x}$指派到第i^*类中，其中$i^* = \max\limits_{i=1,\cdots c} n_i$

K值的选择问题

- 在K近邻分类器中，K值的选择至关重要。如图所示，K 取不同的值，得到的分类决策结果可能是不同的。



K值的选择问题

- 如何选取一个合适的K值呢？考虑两个极端情况：

  ①K最小取值为K=1：分类器容易受到噪声或异常点的影响。

  ②K最大取值为K=n，即训练样本的总数：分类器总是将测试实例预测为训练样本集中的多数类，而不论测试实例是什么，造成分类模型过于简单。

- 可以总结出：

  K值越小，分类器越复杂，易发生过拟合。

  K值越大，分类器越简单，易造成欠拟合。

  综上，K值既不能太大，也不能太小。实际应用中，需要不断尝试选择一个合适的值。

K近邻分类器的时间和空间代价

- 如果距离度量函数$d(\boldsymbol{x},\boldsymbol{y})$采用欧式距离:

  ①计算一次距离的时间复杂度为：$O(d)$。

  ②最朴素的最近邻分类器实现方法需要计算n次距离，其时间复杂度为$O(nd)$。

  ③最朴素的K近邻分类器实现方法的时间复杂度为：$O(nd+Kn)$。

- 空间复杂度:

  最朴素的实现方法是将所有的训练样本都存储起来，这种方法的空间复杂度为$O(nd)$。

  综上，当n和d非常大时，时间和空间代价太大!

降低K近邻分类器的时间和空间代价

- 很多技术都可以使用，包括但不限于：
  - ✓ 计算部分距离：计算距离时没有必要对所有的维度计算完后再比较。如果使用一部分维度计算出的距离比当前的K近邻列表中的K个距离都大，这样的点就可以淘汰了。
  - ✓ 预先构建某种形式的搜索树，使得在搜索的过程中仅仅考虑训练集的一个子集，例如构造kd树。
  - ✓ 对训练样本集加以裁剪：将那些周围被同类样本点环绕的样本点删除。

特征的归一化/标准化

- 在计算两个样本之间的距离时，我们要特别注意样本每个维度上的特征值的取值范围。

  当样本不同维度的特征值的取值范围有很大差异时，前面所讲的ℓ_p范数距离度量实际上隐含地将更多的权重分配给大范围的特征。

- 特征归一化的目的是近似地均衡每维特征的影响，使每个维度上的特征在距离计算中有近似相同的作用。

特征归一化/标准化

- min-max标准化

  对某一维的特征,找到其最小值x_{min}和最大值x_{max}，一个样本在该维度原始的特征值为x，经过min-max标准化后为$x^{'}$:

$$x^{'} = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (x^{'} \in [0,1])$$

- z-score标准化

  对某一维的特征,找到其样本均值μ和样本标准差σ，一个样本在该维度原始的特征值为x，经过z-score标准化后为$x^{'}$:

$$x^{'} = \frac{x - \mu}{\sigma}$$

  z-score标准化后，该维特征的均值为0，方差为1。

特征归一化/标准化

- 排名归一化(rank normalization)

  对某一维的特征，其在所有样本中的取值依次为$x_1, x_2 \cdots x_n$，对每一个值x_i找到其排名$rank(x_i)$，x_i经过排名归一化后的值为：

$$x_i' = \frac{rank(x_i) - 1}{n - 1}$$

  排名归一化后，该维的特征值均匀地缩放到[0,1]。

总结

- K近邻分类规则是：对测试样本$\boldsymbol{x}$，找到与$\boldsymbol{x}$距离最近的前K个训练样本所对应的类标，将$\boldsymbol{x}$指派到这K个类标中的多数类中。

重点

- 当$K=1$时，K近邻分类规则就变成最近邻分类规则。最近邻分类易受噪声、异常点的影响。
- K近邻分类算法不需要事先训练分类器，只需在分类时，计算测试样本的K近邻。最朴素的K近邻分类器的实现，时间和空间复杂度都比较大。
- 为了降低K近邻分类器的时空代价，常采用的方法包括：计算部分距离、预先构建某种形式的搜索树、对训练数据集加以裁剪等方法。

总结

- 使用K-近邻算法分类前，需要选择一个合适的距离度量/相似性度量函数$d(\boldsymbol{x}, \boldsymbol{y})$。学会计算几种常见的距离度量：欧式距离、曼哈顿距离、ℓ_∞范数距离、余弦相似度。
- 当样本不同维度的特征值的取值范围有很大差异时，会利用特征对一化的方法，均衡每个维度上特征的影响。