西安电子科技大学

模式识别 课程实验报告

人工智能学院 1920032 班

姓名 付凯文 学号 19170100004

实验日期 2021 年 10 月 21 日

成绩：
实验报告内容基本要求及参考格式 一、实验目的 二、实验基本原理及步骤 三、实验仿真结果与分析 四、实验中遇到的问题及解决方法（至少 3 个，每人至少写 1 个，写清楚谁的问题和解决方法）

-

实验一　分析 k 近邻的错误率

一、　实验目的

1、学习并掌握 KNN 算法
2、使用 KNN 算法预测鸢尾花的种类
3、分析 KNN 算法的错误率
4、熟悉 python 编程

二、　实验基本原理及步骤

K 近邻算法简介：

k 近邻算法是指给定一个未知标签的样本，在已有的训练集样本集中，找到与该待分类的样本距离最邻近的 k 个训练样本，随后根据这 k 个训练样本的类别，通过一定的决策规则决定该位置样本的类别。具体的流程如下所示：

设训练样本$D=\{(x_1,y_1),(x_2,y_2),\dots\dots(x_m,y_m)\}$，其中$x_i\in X\in R^n$为样本的特征向量，$y_i\in\gamma\in\{c_1,c_2\dots\dots c_s\}$为样本标签，$i=1,2,\dots.m$，$S$为类别个数，对于待分类的未知样本$x$, 求解所属的类$y$步骤如下：

（1）首先确定一种距离度量，计算x与训练集中每个样本x_i的距离，选出距离最小，即与x最邻近的k个点，这k个点集合记作$N_k(x)$。

（2）在$N_k(x)$中，对每个样本的标签投票决定x的类别y, 例如少数服从多数：

$$y=\arg\max_j\sum_{x_i\in N_k(x)} I(y_i=c_j)\quad i=1,2,\cdots,m;\ j=1,2,\cdots,S$$

式中，I为指示函数，即当且仅当$y_i=c_j$时，$I=1$，否则$I=0$

当$k=1$时，k近邻法又称为最近邻算法。显然，对于输入的待分类样本x,其类别为在训练样本集与之距离最小的训练样本类别。

k近邻法的模型有三个要素，分别是距离度量、k值大小与分类决策规则。

K 近邻算法模型：

k近邻法模型中，当给定训练样本集，且确定了距离度量，k值大小及分类决策规则时，对于任意未知标签的样本，通过k近邻法模型，它的预测类别是唯一的。

k近邻法模型根据上述要素进行建模，相当于把特征空间完备划分为一个子空间，每个子空间都有唯一所属的类别。在特征空间中，对于每个训练样本点x_i都有一个单元，即相比于其他训练样本，距离当前训练样本更近的所有点组成的区域。对落入一个单元的所有样本点，其类别都会与该单元所属类别一样。

K 近邻算法中距离度量：

两个样本的特征距离反映了彼此之间的相似程度。常见的距离度量有欧氏距离、具有一般性的L_p距离和曼哈顿距离等。

假设样本空间X属于n维实数向量空间R^n，$x_i,x_j\in X$, 则他们之间的距离定义为

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

其中，p大于等于 1。当$p=2$时，称为欧式距离：

$$L_2(\boldsymbol{x}_i,\boldsymbol{x}_j)=\left(\sum_{l=1}^{n}\left|x_i^{(l)}-x_j^{(l)}\right|^2\right)^{\frac{1}{2}}$$

当$p=1$时，称为曼哈顿距离：

$$L_1(\boldsymbol{x}_i,\ \boldsymbol{x}_j)=\sum_{l=1}^{n}\left|x_i^{(l)}-x_j^{(l)}\right|$$

当$p=\infty$时，它代表各个坐标距离中的最大值，即：

$$L_\infty(\boldsymbol{x}_i,\ \boldsymbol{x}_j)=\max_{l}\left|x_i^{(l)}-x_j^{(l)}\right|$$

K 近邻算法分类决策规则：

k近邻算法中最后一步是利用分类决策规则对未知样本进行分类。其中最常见的分类决策为多数表决：已知未知样本的k个近邻训练样本，统计其中最多的类别作为未知样本的类别，即少数服从多数。

多数表决规则有如下解释：定义样本空间到类别空间映射函数为

$$f:\mathbf{R}^n\rightarrow\{c_1,\ c_2,\ \cdots,\ c_S\}$$

那么错分概率为

$$P(\boldsymbol{y}\neq f(\boldsymbol{x}))=1-P(\boldsymbol{y}=f(\boldsymbol{x}))$$

给定未知样本$x\in X$，求得最近邻的k个样本集合$N_k(x)$，那么c_j为样本集合$N_k(x)$中样本的决策类别，则错分概率为

$$\frac{1}{k}\sum_{\boldsymbol{x}_i\in N_k(\boldsymbol{x})}I(y_i\neq c_j)=1-\frac{1}{k}\sum_{\boldsymbol{x}_i\in N_k(\boldsymbol{x})}I(y_i=c_j)$$

错分概率最小等同经验风险最小，即使

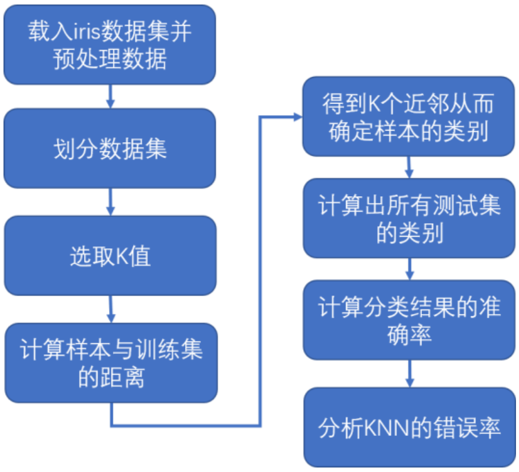
$$\sum_{\boldsymbol{x}_i\in N_k(\boldsymbol{x})}I(y_i=c_j)$$

最大，从这个角度上将，多数表决规则等价于经验风险最小化。

实验步骤：

1. 首先载入 iris 数据集，并通过$Xi=(Xi-Xmin)/(Xmax-Xmin)$将特征向量归一化。
2. 将归一化后的数据集进行划分，本实验中训练集：测试集=7；3
3. 选取合适的 K 值。
4. 计算样本与训练集的距离矩阵。
5. 通过决策规则将测试集中的数据划分到相应的类别。
6. 计算 KNN 的分类正确率。
7. 对 KNN 的错误率进行分析。

实验步骤如下图所示：



三、 实验仿真结果及分析

利用 python 对实验进行多次仿真，预测的争取率在 0.93-1.00 之间，平均正确率为 0.96（十次仿真的结果见附件）

KNN 与贝叶斯最优分类器的期望错误率分别为

$$err = 1 - \sum_{c \in Y} P^2(c|x) \text{ ,} \quad err* = 1 - \max_{c \in Y} P(c|x)$$

设 $c^* = \arg\max_{c \in Y} P(c|x)$，则 $\max_{c \in Y} P(c|x) = P(c^*|x)$

因为：

$$\begin{aligned}
\sum_{c \in Y} P^2(c|x) &= \max_{c \in Y} P^2(c|x) + \sum_{c \in Y} P^2(c|x) - \max_{c \in Y} P^2(c|x) \\
&= P^2(c^*|x) + \sum_{c \in Y} P^2(c|x) - P^2(c^*|x) \\
&= P^2(c^*|x) + \sum_{c \in Y, c \neq c^*} P^2(c|x)
\end{aligned}$$

所以：

$$\begin{aligned}
\max_{c \in Y} P(c|x) - \sum_{c \in Y} P^2(c|x) &= \max_{c \in Y} P(c|x) - P^2(c^*|x) - \sum_{c \in Y, c \neq c^*} P^2(c|x) \\
&= P(c^*|x) - P^2(c^*|x) - \sum_{c \in Y, c \neq c^*} P^2(c|x) \\
&= P(c^*|x)(1 - P(c^*|x)) - \sum_{c \in Y, c \neq c^*} P^2(c|x) \\
&= P(c^*|x) \sum_{c \in Y, c \neq c^*} P(c|x) - \sum_{c \in Y, c \neq c^*} P^2(c|x) \\
&= \sum_{c \in Y, c \neq c^*} P(c|x) \left(P(c^*|x) - P(c|x)\right)
\end{aligned}$$

因为 $P(c^*|x) - P(c|x) \geq 0$ 且 $P(c|x) \geq 0$，所以 $\max_{c \in Y} P(c|x) \geq \sum_{c \in Y} P^2(c|x)$ 成立，

即 $err^* \leq err$ 成立。

由柯西-施瓦茨不等式

$$\begin{aligned}\sum_{c\in Y,c\neq c^*} P^2(c|x) &\geq \frac{1}{|Y|-1}\left(\sum_{c\in Y,c\neq c^*} P(c|x)\right)^2 \\ &= \frac{1}{|Y|-1}(1-P(c^*|x))^2 \\ &= \frac{1}{|Y|-1}err^{*2}\end{aligned},$$

所以：

$$\begin{aligned}\sum_{c\in Y} P^2(c|x) &\geq \frac{1}{|Y|-1}err^{*2} + P^2(c^*|x) \\ &= \frac{1}{|Y|-1}err^{*2} + (1-err^*)^2\end{aligned}$$

$$err = 1 - \sum_{c\in Y} P^2(c|x)$$

所以：

$$\begin{aligned}err &= 1 - \sum_{c\in Y} P^2(c|x) \\ &\leq 1 - \frac{1}{|Y|-1}err^{*2} - (1-err^*)^2 \\ &= 2err^* - err^{*2}(\frac{|Y|}{|Y|-1}) \\ &= err^*(2 - \frac{|Y|}{|Y|-1} * err^*)\end{aligned}$$

综上所述：

$$err^* \leq err \leq err^*(2 - \frac{Y}{|Y|-1} * err^*).$$

因此，最近邻的渐进错误率最坏时不超过两倍的贝叶斯错误率，最好时接近或达到贝叶斯错误率。

四、 实验中遇到的问题及解决方法

遇到的问题：在实验时不知道如何选取合适的 K 值。K 值选取过小会导致模型过拟合，选取过大的 K 值会使模型变得简单。两种情况都会使预测的准确率变低。

解决方法：利用 K 折交叉验证评估的方式选出最合适的 K 值（本实验中选取五折交叉验证的方式）

工作流程：

1、将数据集分成 5 段。

2、依次选取其中的 4 个子集作为训练集，剩下的 1 个子集作为测试集进行实验。

3、计算每次验证结果的平均值作为最终准确率

4、比较所有 K 值的最终准确率，选出最佳的 K 值。

最终得到，当 K=6 时 KNN 的准确率最高，为 0.980，因此选取 K=6 作为参数的值。

五、 附录

```
import numpy as np
```

```python
import matplotlib.pyplot as plt
from sklearn import datasets
# load 数据集
iris = datasets.load_iris()
# 特征值
diris = iris.data
# 标签
tiris = iris.target
labels = ['setosa', 'versicolor', 'virginica']
# print(airis)
'''
k 值会影响结果
'''
# 定义 k 值,k 个最近的邻居
k = 5
# 数组归一化 x=(x-min)/(max-min)
for i in range(4):
    diris[:, i] = (diris[:, i]-np.min(diris[:, i])) / \
        (np.max(diris[:, i])-np.min(diris[:, i]))
# print(diris)

# 随机选取 105 个样本做训练集，45 个样本做测试集
index = np.arange(150)
np.random.shuffle(index)
traindata, trainlabel = diris[index[:105]], tiris[index[:105]]
testdata, testlabel = diris[index[105:]], tiris[index[105:]]
# print(traindata,trainlabel)
# print(testdata,testlabel)
# result 用来保存结果
result = np.zeros(testlabel.shape[0])

for n, l in enumerate(testdata):
    m = np.square(traindata-l)
    dis = np.sqrt(m.sum(axis=1))  # 计算欧氏距离
    dis = np.c_[dis, trainlabel]
    dis = dis[np.argsort(dis[:, 0])]  # 按照距离大小进行排序
    a = [i[1] for i in dis[:k]]
    result[n] = max(a, key=a.count)
    print('花的实际种类为'+labels[int(testlabel[n]-1)] +
          '  '+'花的预测种类为'+labels[int(result[n]-1)])
print('测试集准确率%.2f' % (list(result-testlabel).count(0)/len(testlabel)))
```