南开大学
Nankai University

南　开　大　学

计 算 机 学 院

机器学习实验报告

实验一 基于KNN的手写数字识别

姓名：王泳鑫

学号：1911479

年级：2019级

专业：计算机科学与技术

指导教师：卫金茂

2021 年 10 月 8 日

摘要

关键字：**KNN， Machine Learning， Deep Learning**

目录

一、 实验描述

（一） 实验内容

给定semeion手写数字数据集，实现KNN分类算法

（二） 实验要求

基本要求：编程实现kNN算法；给定不同k值(1,3,5)情况下，kNN算法对手写数字的识别精度（要求采用留一法）中级要求：将实验过程结果等图示展出

二、 代码实现

（一） kNN分类算法

该函数的功能就是使用k-邻近算法将每组数据划分到某个类中，其算法伪代码如下：

1. 计算已知类别数据集中的点和当前点的距离；
2. 按照距离递增次序排序；
3. 选取与当前点距离最小的k个点；
4. 确定前k个点所在类别的出现频率；
5. 返回前k个点出现频率最高的类别作为当前点的预测分类。

```
def classify0(inX,dataSet,labels,k):#分类器knn
#首先计算已知类别数据集与当前点的距离
dataSetSize=dataSet.shape[0]#读取数据集的行数，声明为dataSetSize
diffMat = tile(inX,(dataSetSize,1))-dataSet#
sqDiffMat = diffMat**2#平方
sqDistances = sqDiffMat.sum(axis=1)#相加
distances = sqDistances**0.5#开方


#按照距离递增次序排序
sortedDistindicies = distances.argsort()#返回排序结果的索引
classCount = {}#新建一个词典用来计数

#选取与当前点距离最小的个点并且确定出现频率k
for i in range(k):
    voteIlabel = labels[sortedDistindicies[i]]
    classCount[voteIlabel] = classCount.get(voteIlabel,0)+1

#将出现频率进行排序
sortedClassCount = sorted(classCount.items(),key=operator.itemgetter(1),reverse=
                                            True)
return sortedClassCount[0][0]
```

（二） 数据集处理

本次实验采用semeion数据集，该数据集每一条数据项，都是由256个0或1组成（相当于16*16的阴影），同时后面紧跟着一个大小为10的独热码来标注它的分类，因此对于该数据集，我们需要将前面256大小的数据作为计算距离的dataset，后面的独热码作为可以判断分类的label。

```
def datareading(filename):
 #读取文件
 fo=open(filename)
 data = fo.readlines()
 fo.close()

 row = len(data)#计算数据项的数量
 dataMat = zeros((row,256))#初始化为零矩阵
 dataLabels = []#新建一个列表，用来存储label

 for i in range(row):
     vals = data[i].split()
     dataMat[i,:] = vals[0:256]
     dataLabels.append(vals[256:].index('1'))

 return dataMat,dataLabels
```

（三） 测试与验证

实验要求使用留一法来进行验证，留一法就是每次只留下一个样本做测试集，其它样本做训练集，如果有n个样本，则需要训练n次，测试n次。

```
def test():
 dataMat,dataLabels = datareading('D:\\Data\\semeion.data')
 m = dataMat.shape[0]
 testnum = int(m*0.1)
 TNum = 0
 k=3
 for i in range(testnum):
     ans = classify0(dataMat[i,:],dataMat[testnum:,:],dataLabels[testnum:],k)
     if(ans==dataLabels[i]): TNum+=1
 print("When k is ",k,",the Precision is",TNum/testnum)
```

三、 实验结果展示与分析

对于指定k进行测试，当k为(1,3,5)时，测试结果如图2所示

```
When k is  1 ,the Precision is 0.723404255319149
When k is  1 ,the Precision is 0.8556485355648535  by sklearn
When k is  2 ,the Precision is 0.723404255319149
When k is  2 ,the Precision is 0.799163179916318  by sklearn
When k is  3 ,the Precision is 0.723404255319149
When k is  3 ,the Precision is 0.8410041841004184  by sklearn
When k is  4 ,the Precision is 0.8085106382978723
When k is  4 ,the Precision is 0.8263598326359832  by sklearn
When k is  5 ,the Precision is 0.7872340425531915
When k is  5 ,the Precision is 0.8389121338912134  by sklearn
When k is  6 ,the Precision is 0.7872340425531915
When k is  6 ,the Precision is 0.8347280334728033  by sklearn
When k is  7 ,the Precision is 0.7872340425531915
When k is  7 ,the Precision is 0.8389121338912134  by sklearn
When k is  8 ,the Precision is 0.723404255319149
When k is  8 ,the Precision is 0.8389121338912134  by sklearn
When k is  9 ,the Precision is 0.7659574468085106
When k is  9 ,the Precision is 0.8472803347280334  by sklearn
When k is  10 ,the Precision is 0.7659574468085106
When k is  10 ,the Precision is 0.8389121338912134  by sklearn
When k is  11 ,the Precision is 0.7446808510638298
When k is  11 ,the Precision is 0.8242677824267782  by sklearn
When k is  12 ,the Precision is 0.7446808510638298
When k is  12 ,the Precision is 0.8200836820083682  by sklearn
When k is  13 ,the Precision is 0.7659574468085106
When k is  13 ,the Precision is 0.8138075313807531  by sklearn
When k is  14 ,the Precision is 0.7021276595744681
When k is  14 ,the Precision is 0.805439330543933  by sklearn
When k is  15 ,the Precision is 0.7659574468085106
When k is  15 ,the Precision is 0.805439330543933  by sklearn
When k is  16 ,the Precision is 0.7659574468085106
```

图 1: kNN测试结果

这幅图中也包含了利用sklearn中knn分类算法得到预测结果，我所写的程序在不同k值的情况下得到的结果如下图2所示：

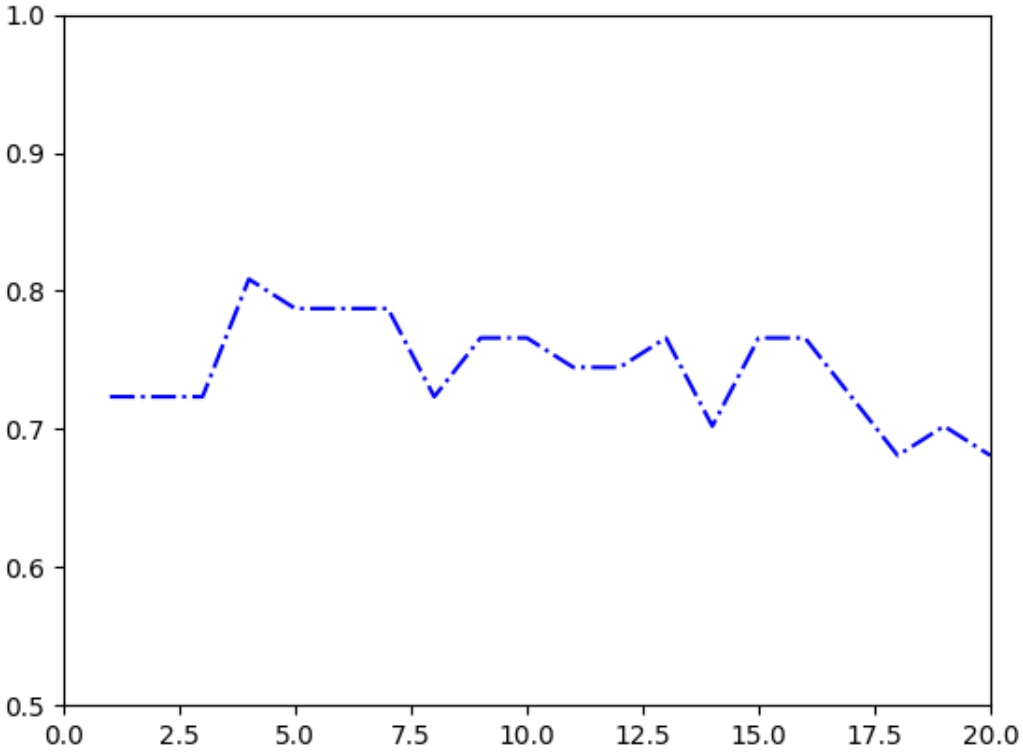

图 2: myplot