实验介绍

1.实验内容

介绍如何使用kNN进行乳腺癌诊断。

2.实验目标 ¶

通过本实验掌握kNN算法的原理，熟悉如何运用kNN算法解决真实世界问题。

3.实验知识点

- kNN算法原理
- kNN算法流程

4.实验环境

- R

5.预备知识

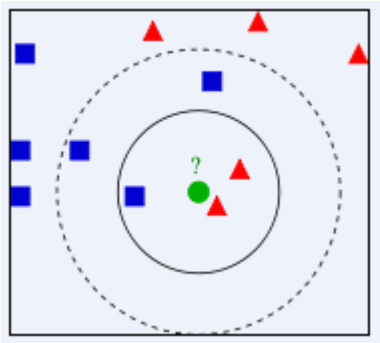
- 初等数学知识
- Linux命令基本操作
- R语言编程基础
- kNN算法原理

实验原理

1.kNN算法简介

k近邻法(k-nearest neighbor, kNN)是1967年由Cover T和Hart P提出的一种基本分类与回归方法。它的工作原理是：存在一个样本数据集合，也称作为训练样本集，并且样本集中每个数据都存在标签，即我们知道样本集中每一个数据与所属分类的对应关系。输入没有标签的新数据后，将新的数据的每个特征与样本集中数据对应的特征进行比较，然后算法提取样本最相似数据(最近邻)的分类标签。一般来说，我们只选择样本数据集中前k个最相似的数据，这就是k-近邻算法中k的出处，通常k是不大于20的整数。最后，选择k个最相似数据中出现次数最多的分类，作为新数据的分类。

所谓K最近邻，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。该方法在确定分类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。 kNN方法在类别决策时，只与极少量的相邻样本有关。由于kNN方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因此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。



上图中，绿色圆要被决定赋予哪个类，是红色三角形还是蓝色四方形？如果K=3，由于红色三角形所占比例为2/3，绿色圆将被赋予红色三角形那个类，如果K=5，由于蓝色四方形比例为3/5，因此绿色圆被赋予蓝色四方形类。

K最近邻(k-Nearest Neighbor，KNN)分类算法，是一个理论上比较成熟的方法，也是最简单的机器学习算法之一。该方法的思路是：如果一个样本在特征空间中的k个最相似(即特征空间中最邻近)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。KNN算法中，所选择的邻居都是已经正确分类的对象。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。 KNN方法虽然从原理上也依赖于极限定理，但在类别决策时，只与极少量的相邻样本有关。由于KNN方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因此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。

KNN算法不仅可以用于分类，还可以用于回归。通过找出一个样本的k个最近邻居，将这些邻居的属性的平均值赋给该样本，就可以得到该样本的属性。更有用的方法是将不同距离的邻居对该样本产生的影响给予不同的权值(weight)，如权值与距离成反比。

2.kNN算法流程

a.计算已知类别数据集中的点与当前点之间的距离；
b.按照距离递增次序排序；
c.选取与当前点距离最小的k个点；
d.确定前k个点所在类别的出现频率；
e.返回前k个点所出现频率最高的类别作为当前点的预测分类。

【练习】基于kNN的乳腺癌诊断-背景

乳腺癌是女性常见的恶性肿瘤之一,是威胁女性健康常见肿瘤。

数据解析

数据来源是UCI机器学习数据仓库的威廉康星乳腺癌诊断数据集。这个数据集包含569例细胞活检案例，每个案例有32个乳房肿块活检图像显示的细胞核的特征。第一个特征是ID，第二个是这个案例的癌症诊断结果，癌症诊断结果用编码"M"表示恶性，B表示良性。其他30个特征是数值型的其他指标，包括细胞核的半径(Radius)、质地(Texture)、周长(Perimeter)、面积(Area)和光滑度(Smoothness)等的均值、标准差和最大值。部分数据如下：

	ID	Diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
1	842302	M	17.990	10.38	122.80	1001.0	0.11840	0.27760
2	842517	M	20.570	17.77	132.90	1326.0	0.08474	0.07864
3	84300903	M	19.690	21.25	130.00	1203.0	0.10960	0.15990
4	84348301	M	11.420	20.38	77.58	386.1	0.14250	0.28390
5	84358402	M	20.290	14.34	135.10	1297.0	0.10030	0.13280
6	843786	M	12.450	15.70	82.57	477.1	0.12780	0.17000
7	844359	M	18.250	19.98	119.60	1040.0	0.09463	0.10900
8	84458202	M	13.710	20.83	90.20	577.9	0.11890	0.16450
9	844981	M	13.000	21.82	87.50	519.8	0.12730	0.19320
10	84501001	M	12.460	24.04	83.97	475.9	0.11860	0.23960
11	845636	M	16.020	23.24	102.70	797.8	0.08206	0.06669
12	84610002	M	15.780	17.89	103.60	781.0	0.09710	0.12920
13	846226	M	19.170	24.80	132.40	1123.0	0.09740	0.24580
14	846381	M	15.850	23.95	103.70	782.7	0.08401	0.10020
15	84667401	M	13.730	22.61	93.60	578.3	0.11310	0.22930
16	84799002	M	14.540	27.54	96.73	658.8	0.11390	0.15950
17	848406	M	14.680	20.13	94.74	684.5	0.09867	0.07200
18	84862001	M	16.130	20.68	108.10	798.8	0.11700	0.20220
19	849014	M	19.810	22.15	130.00	1260.0	0.09831	0.10270
20	8510426	B	13.540	14.36	87.46	566.3	0.09779	0.08129
21	8510653	B	13.080	15.71	85.63	520.0	0.10750	0.12700

可以看出各个特征的范围相差较大，直接使用原始数据计算距离将使分类器出现问题，所以应该使用标准化方法重新调整特征的值。

【练习】准备数据数据归一化

首先使用R从文件wdbc.data.txt中读入数据并进行整理，代码及注释如下：

In [1]:
```
rm(list=ls())
# 读入数据:
wbcd<-read.table("wdbc.data.txt",stringsAsFactors = F,header = T)
# 删除ID:
wbcd<-wbcd[,-1]
# 查看肿瘤和良性的比例:
prop.table(table(wbcd$Diagnosis))
# 将Diagnosis转为因子并给标签
wbcd$Diagnosis<-factor(wbcd$Diagnosis,levels=c("B","M"),labels = c("Benign","Malignant"))
prop.table(table(wbcd$Diagnosis))
```

```
        B         M 
0.6274165 0.3725835 


   Benign Malignant 
0.6274165 0.3725835 
```

可以看出数据案例中恶性肿瘤占62.7%，良性占37.2%。
接着进行min-max标准化：

In [2]:
```
# 由于不同特征的范围差别很大，所以需要Min-Max归一化:
normalize<-function(x){  return((x-min(x))/(max(x)-min(x)))}
# 使用lapply对每一列进行标准化
wbcd_n<-as.data.frame(lapply(wbcd[,2:31],normalize))
```

将数据分为两部分：一部分是训练集（随机抽取469条）；一部分是是评估模型准确性的测试集（剩下的100条）

In [3]:
```
# 随机取样100个作为测试集：
test_100<-sample(1:nrow(wbcd_n),100)
# 剩下的469个是训练集：
train_w<-setdiff(1:nrow(wbcd_n),test_100)
# 提取测试集数据：
wbcd_test<-wbcd_n[test_100,]
# 提取训练集数据：
wbcd_train<-wbcd_n[train_w,]
# 测试集诊断结果标签：
wbcd_test_labels<-wbcd[test_100,1]
# 训练集诊断结果标签：
wbcd_train_labels<-wbcd[train_w,1]
```

使用class包的knn()函数来实现本次训练和预测：

In [4]:
```
# 使用class包的knn()函数：
# 载入包
library(class)
# 进行训练预测：
wbcd_test_pred<-knn(train = wbcd_train,test=wbcd_test,cl=wbcd_train_labels,k=21)
```

【练习】测试算法验证分类器

创建双向交叉表（gmodels包的CrossTable()函数）来评估模型的性能：

In [5]:
```
library(gmodels)
CrossTable(x=wbcd_test_labels,y=wbcd_test_pred,prop.chisq = F)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  100


                 | wbcd_test_pred
wbcd_test_labels |    Benign | Malignant | Row Total |
-----------------|-----------|-----------|-----------|
          Benign |        55 |         0 |        55 |
                 |     1.000 |     0.000 |     0.550 |
                 |     0.917 |     0.000 |           |
                 |     0.550 |     0.000 |           |
-----------------|-----------|-----------|-----------|
       Malignant |         5 |        40 |        45 |
                 |     0.111 |     0.889 |     0.450 |
                 |     0.083 |     1.000 |           |
                 |     0.050 |     0.400 |           |
-----------------|-----------|-----------|-----------|
    Column Total |        60 |        40 |       100 |
                 |     0.600 |     0.400 |           |
-----------------|-----------|-----------|-----------|
```

上图左上角的格子表示真阴性(True Negative)，这里表示分类器结果和临床结果一致认为是良性的情况，100个预测中有63个真阴性结果。相反，右下角就是真阳性(True Positive)结果，有32个案例分类器和临床一致认为是恶性的。左下的5个是假阴性(False Negative)，预测为良性实际是恶性，这可能导致病人错过治疗，这是最糟糕的预测结果。右上角的是假阳性(False Positive)，预测是恶性实际是良心，会导致病人虚惊一场。

其他评估模型预测结果的方法如敏感性、特异性、ROC曲线及检验。

参考文献及延伸阅读

参考资料：

1.哈林顿，李锐. 机器学习实战 : Machine learning in action[M]. 人民邮电出版社, 2013.
2.周志华. 机器学习:Machine learning[M]. 清华大学出版社, 2016.

延伸阅读：

1.李航. 统计学习方法[M]. 清华大学出版社, 2012.

看不懂，直接搬的答案[笑哭]