数据挖掘作业02

knn分类和估计分类器的精度

- 14051435 叶梅北宁
- 14051440 周贤杰
- 14051409 陈品维

实验分析

实验开始之前，先明确实验中可能会碰到的问题和需要注意的地方:

> 计算两个实验数据距离所使用的距离算法

通常欧氏距离作为计算N维向量之间的距离最常用的算法。但是，DNA位点之间距离我个人认为欧氏距离不是最好的算法，所以用了以下几种算法备用

1. Cosine 余弦函数
2. Adjust_Cosine 修正余弦函数
3. Pearson_Metric 皮尔森系数
4. Euclidean_Metric 欧几里得距离

> K临近分类其中K的值

K值的选择关系到实验的结果，查阅部分资料后了解到：

- K太小，分类结果易受噪声点影响
- K太大，近邻中又可能包含太多的其它类别的点。

一般K的值不超过训练样本数的平方根，对于本次试验我们一般认为 1 <= k <= 8

实验将基于10折交叉验证和留一交叉验证来决定K值和距离算法的选择

> 10折交叉验证的验证组和训练组的选择

实验中，我们将样本均分后**随机**抽取训练组的实验组，为了减小实验误差，对于每个K值和距离函数，进行10次10折交叉验证，之后取平均值作为结果

实验过程

实验代码基于Python3.5,使用xlrd解析xls文件。

项目GitHub地址: Data Mining

代码运行方法:

```
python3 knn.py -l {label file} -d {data file(*.xls)} [dist func]
dist func:
          -c cosine
          -a adjust cosine
          -p pearson
          -e euclidean
```

实验环境

本次试验在MacOS 10.12 Serria下进行，硬件配置为:

- CPU: 2.2 GHz Intel Core i7
- 内存: 16 GB 1600 MHz DDR3

实验结果

> 使用余弦函数作为样本间距离函数

```
kanouumekitas-MacBook-Pro:HomeWork_2_KNN lostmoonkin$ python3 knn.py -l data/dmr_labe
l.txt -d data/dmr_data.xls -c
Start cross_validation Test:
ten fold cross validation(knn: k = 1 ): 0.560
ten fold cross validation(knn: k = 2 ): 0.571
ten fold cross validation(knn: k = 3 ): 0.642
ten fold cross validation(knn: k = 4 ): 0.617
ten fold cross validation(knn: k = 5 ): 0.642
ten fold cross validation(knn: k = 6 ): 0.633
ten fold cross validation(knn: k = 7 ): 0.624
ten fold cross validation(knn: k = 8 ): 0.615
ten fold cross validation test finished.
best k for the test data: 3
Start loocv test:
k = 1
num of data: 96
passed: 54 56.25%
k = 2
num of data: 96
passed: 54 56.25%
k = 3
num of data: 96
passed: 62 64.58%
k = 4
num of data: 96
passed: 61 63.54%
k = 5
num of data: 96
passed: 61 63.54%
k = 6
num of data: 96
passed: 62 64.58%
k = 7
num of data: 96
passed: 61 63.54%
k = 8
num of data: 96
passed: 61 63.54%
loocv test finished.
best k for the loocv test: 3
```

结论: k = 3 - 5时，10次10折交叉验证和留一交叉验证的正确率都在最高，都在63%左右

> 使用修正余弦函数作为样本间距离函数

```
kanouumekitas-MacBook-Pro:HomeWork_2_KNN lostmoonkin$ python3 knn.py -l data/dmr_label.txt -d data/dmr_data.xls -a
Start cross_validation Test:
ten fold cross validation(knn: k = 1 ): 0.584
ten fold cross validation(knn: k = 2 ): 0.482
ten fold cross validation(knn: k = 3 ): 0.513
ten fold cross validation(knn: k = 4 ): 0.518
ten fold cross validation(knn: k = 5 ): 0.567
ten fold cross validation(knn: k = 6 ): 0.558
ten fold cross validation(knn: k = 7 ): 0.551
ten fold cross validation(knn: k = 8 ): 0.554
ten fold cross validation test finished.
best k for the test data: 1
Start loocv test:
k = 1
num of data: 96
passed: 51 53.12%
k = 2
num of data: 96
passed: 46 47.92%
k = 3
num of data: 96
passed: 47 48.96%
k = 4
num of data: 96
passed: 51 53.12%
k = 5
num of data: 96
passed: 54 56.25%
k = 6
num of data: 96
passed: 54 56.25%
k = 7
num of data: 96
passed: 52 54.17%
k = 8
num of data: 96
passed: 54 56.25%
loocv test finished.
best k for the loocv test: 5
```

结论: 对于此实验，修正余弦函数的识别率还不如普通余弦函数。个人感觉样本属性之间并不是相互独立的，才导致修正属性之后的识别率下降。

> 使用皮尔森指数作为样本间距离函数

```
kanouumekitas-MacBook-Pro:HomeWork_2_KNN lostmoonkin$ python3 knn.py -l data/dmr_labe
l.txt -d data/dmr_data.xls -p
Start cross_validation Test:
ten fold cross validation(knn: k = 1 ): 0.557
ten fold cross validation(knn: k = 2 ): 0.572
ten fold cross validation(knn: k = 3 ): 0.639
ten fold cross validation(knn: k = 4 ): 0.613
ten fold cross validation(knn: k = 5 ): 0.631
ten fold cross validation(knn: k = 6 ): 0.627
ten fold cross validation(knn: k = 7 ): 0.614
ten fold cross validation(knn: k = 8 ): 0.604
ten fold cross validation test finished.
best k for the test data: 3
Start loocv test:
k = 1
num of data: 96
passed: 53 55.21%
k = 2
num of data: 96
passed: 55 57.29%
k = 3
num of data: 96
passed: 62 64.58%
k = 4
num of data: 96
passed: 60 62.50%
k = 5
num of data: 96
passed: 61 63.54%
k = 6
num of data: 96
passed: 61 63.54%
k = 7
num of data: 96
passed: 61 63.54%
k = 8
num of data: 96
passed: 60 62.50%
loocv test finished.
best k for the loocv test: 3
```

结论: 皮尔森指数的结果和余弦函数结果十分相似，都在k = 3 - 5时取得最大值

> 使用欧几里得距离作为样本间距离函数

```
kanouumekitas-MacBook-Pro:HomeWork_2_KNN lostmoonkin$ python3 knn.py -l data/dmr_label.txt -d data/dmr_data.xls -e
Start cross_validation Test:
ten fold cross validation(knn: k = 1 ): 0.544
ten fold cross validation(knn: k = 2 ): 0.446
ten fold cross validation(knn: k = 3 ): 0.491
ten fold cross validation(knn: k = 4 ): 0.436
ten fold cross validation(knn: k = 5 ): 0.474
3ten fold cross validation(knn: k = 6 ): 0.507
ten fold cross validation(knn: k = 7 ): 0.544
ten fold cross validation(knn: k = 8 ): 0.519
ten fold cross validation test finished.
best k for the test data: 7
Start loocv test:
k = 1
num of data: 96
passed: 53 55.21%
k = 2
num of data: 96
passed: 42 43.75%
k = 3
num of data: 96
passed: 47 48.96%
k = 4
num of data: 96
passed: 42 43.75%
k = 5
num of data: 96
passed: 44 45.83%
k = 6
num of data: 96
passed: 47 48.96%
k = 7
num of data: 96
passed: 53 55.21%
k = 8
num of data: 96
passed: 52 54.17%
loocv test finished.
best k for the loocv test: 1
```

结论: 使用欧氏距离的实验结果十分不理想，也验证了之前的猜测。

实验结论

距离函数	K值	10折交叉验证正确率	留一交叉验证正确率
余弦函数	3, 5	64.2%	64.6%
修正余弦函数	1	58.4%	56.5%
皮尔森指数	3	63.9%	64.5%
欧几里得距离	1	54.4%	55.2%

本次试验中，识别正确率最高两个距离函数为余弦函数和皮尔森指数，之后为修正余弦函数，欧式距离最低。

但是，实验总体的识别率并不高，最高也没超过70%，组员讨论之后，可能的原因如下:

1. k值选择问题，我个人认为这个结论不成立，在实验中我们已经对可能的K值进行了枚举，基本排除
2. 距离公式计算/选择存在问题
3. 未对实验数据进行处理，因为实验数据中，A和C列的数值相对其他列来说，差距过大。但尝试一些处理方法之后，并没有产生效果，暂时也未想到更好的方法
4. 1,2,3共同影响

Update 11.19.2016

对数据做了归一化处理，归一化之后使用欧氏距离准确率提升较大:

基本达到了皮尔森指数和余弦函数的准确率。

```
kanouumekitas-MacBook-Pro:HomeWork_2_KNN lostmoonkin$ python3 knn.py -l data/dmr_labe
l.txt -d data/dmr_data.xls -c
Start cross_validation Test:
ten fold cross validation(knn: k = 1 ): 0.571
ten fold cross validation(knn: k = 2 ): 0.606
ten fold cross validation(knn: k = 3 ): 0.638
ten fold cross validation(knn: k = 4 ): 0.627
ten fold cross validation(knn: k = 5 ): 0.637
ten fold cross validation(knn: k = 6 ): 0.614
ten fold cross validation(knn: k = 7 ): 0.621
ten fold cross validation(knn: k = 8 ): 0.595
ten fold cross validation test finished.
best k for the test data: 3
Start loocv test:
k = 1
num of data: 96
passed: 54 56.25%
k = 2
num of data: 96
passed: 58 60.42%
k = 3
num of data: 96
passed: 62 64.58%
k = 4
num of data: 96
passed: 60 62.50%
k = 5
num of data: 96
passed: 61 63.54%
k = 6
num of data: 96
passed: 60 62.50%
k = 7
num of data: 96
passed: 61 63.54%
k = 8
num of data: 96
passed: 58 60.42%
loocv test finished.
best k for the loocv test: 3
```

基本达到了皮尔森指数和余弦函数的准确率。

归一化函数使用 new*x = (x - min*x) / (max*x - min*x)