个人资料

zouxy09

访问： 3692429次
积分： 23414
等级： BLOG 7
排名： 第156名

原创： 116篇 转载： 11篇
译文： 1篇 评论： 2942条

个人简介

关注：机器学习、计算机视觉、人机交互和人工智能等领域。
邮箱：zouxy09@qq.com
微博：Erik-zou
交流请发邮件，不怎么看博客私信^-^

文章搜索

文章分类

OpenCV (29)
机器学习 (46)
计算机视觉 (73)
Deep Learning (18)
语音识别与TTS (13)
图像处理 (55)
Linux (15)
Linux驱动 (4)
嵌入式 (18)
OpenAL (3)
Android (1)
C/C++编程 (18)
摄像头相关 (5)
数学 (5)
Kinect (9)
神经网络 (8)

机器学习算法与Python实践之（一）k近邻（KNN）

2013-11-26 00:38 41148人阅读 评论(19) 收藏 举报

分类： C/C++编程（17） 机器学习（45）

机器学习算法与**Python**实践之（一）**k**近邻（**KNN**）

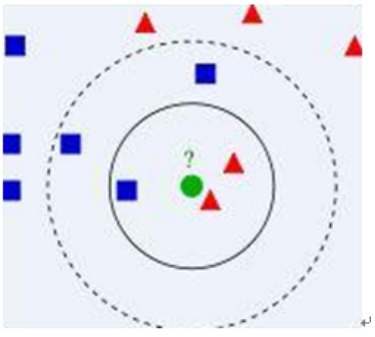
zouxy09@qq.com

http://blog.csdn.net/zouxy09

机器学习算法与Python实践这个系列主要是参考《机器学习实战》这本书。因为自己想学习Python，然后也想对一些机器学习算法加深下了解，所以就想通过Python来实现几个比较常用的机器学习算法。恰好遇见这本同样定位的书籍，所以就参考这本书的过程来学习了。

一、**kNN**算法分析

K最近邻（k-Nearest Neighbor，KNN）分类算法可以说是最简单的机器学习算法了。它采用测量不同特征值之间的距离方法进行分类。它的思想很简单：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。



比如上面这个图，我们有两类数据，分别是蓝色方块和红色三角形，他们分布在一个上图的二维中间中。那么假如我们有一个绿色圆圈这个数据，需要判断这个数据是属于蓝色方块这一类，还是与红色三角形同类。怎么做呢？我们先把离这个绿色圆圈最近的几个点找到，因为我们觉得离绿色圆圈最近的才对它的类别有判断的帮助。那到底要用多少个来判断呢？这个个数就是k了。如果k=3，就表示我们选择离绿色圆圈最近的3个点来判断，由于红色三角形所占比例为2/3，所以我们认为绿色圆是和红色三角形同类。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。从这里可以看到，k的值还是很重要的。

随谈 (2)

文章存档

2015年10月 (4)
2015年04月 (2)
2014年12月 (1)
2014年08月 (1)
2014年05月 (2)

展开

阅读排行

Deep Learning（深度学 (244808)
Deep Learning（深度学 (201628)
Deep Learning（深度学 (180423)
Deep Learning（深度学 (118741)
Deep Learning（深度学 (113259)
Deep Learning（深度学 (112887)
Deep Learning论文笔记 (112564)
计算机视觉、机器学习相 (110393)
从最大似然到EM算法浅解 (99557)
Deep Learning（深度学 (93579)

评论排行

Deep Learning论文笔记 (211)
基于Qt的P2P局域网聊天 (150)
从最大似然到EM算法浅解 (123)
时空上下文视觉跟踪（S (115)
计算机视觉、机器学习相 (113)
Deep Learning（深度学 (102)
机器学习算法与Python实 (78)
压缩跟踪Compressive T (67)
Deep Learning（深度学 (65)
压缩跟踪Compressive T (64)

最新评论

机器学习算法与Python实践之（
瘦子什么时候才能长胖: 98.84%，get。

TLD（Tracking-Learning-Detec
qinfenziqiang: @world_hope:添加的链接库问题，Debug模式下只添加带d的lib

标签传播算法（Label Propagati
无间虚者: @liyaohhh:在loadCircleData里写着。

标签传播算法（Label Propagati
无间虚者: @sophiahls:随机初始化无标签数据的标签

标签传播算法（Label Propagati
无间虚者: labelPropagation函数中的第99行，无标签的数据不是label_function在nu...

Deep Learning论文笔记之（五）
kisshunter: @aslinyufang:横坐标应该是代表的的所有样本吧，60000个训练样本，每50个作为一批进行...

Python机器学习库scikit-learn实
LvZhongkai: 不错不错

Deep Learning（深度学习）学习

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。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。

该算法在分类时有个主要的不足是，当样本不平衡时，如一个类的样本容量很大，而其他类样本容量很小时，有可能导致当输入一个新样本时，该样本的K个邻居中大容量类的样本占多数。因此可以采用权值的方法（和该样本距离小的邻居权值大）来改进。该方法的另一个不足之处是计算量较大，因为对每一个待分类的文本都要计算它到全体已知样本的距离，才能求得它的K个最近邻点。目前常用的解决方法是事先对已知样本点进行剪辑，事先去除对分类作用不大的样本。该算法比较适用于样本容量比较大的类域的自动分类，而那些样本容量较小的类域采用这种算法比较容易产生误分[参考机器学习十大算法]。

总的来说就是我们已经存在了一个带标签的数据库，然后输入没有标签的新数据后，将新数据的每个特征与样本集中数据对应的特征进行比较，然后算法提取样本集中特征最相似（最近邻）的分类标签。一般来说，只选择样本数据库中前k个最相似的数据。最后，选择k个最相似数据中出现次数最多的分类。其算法描述如下：

1）计算已知类别数据集中的点与当前点之间的距离；

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。

二、Python实现

对于机器学习而已，Python需要额外安装三件宝，分别是Numpy，scipy和Matplotlib。前两者用于数值计算，后者用于画图。安装很简单，直接到各自的官网下载回来安装即可。安装程序会自动搜索我们的python版本和目录，然后安装到python支持的搜索路径下。反正就python和这三个插件都默认安装就没问题了。

另外，如果我们需要添加我们的脚本目录进Python的目录（这样Python的命令行就可以直接import），可以在系统环境变量中添加：PYTHONPATH环境变量，值为我们的路径，例如：E:\Python\Machine Learning in Action

2.1、kNN基础实践

一般实现一个算法后，我们需要先用一个很小的数据库来测试它的正确性，否则一下子给个大数据给它，它也很难消化，而且还不利于我们分析代码的有效性。

首先，我们新建一个kNN.py脚本文件，文件里面包含两个函数，一个用来生成小数据库，一个实现kNN分类算法。代码如下：

```python
[python]

01. ########################################
02. # kNN: k Nearest Neighbors
03.
04. # Input:      newInput: vector to compare to existing dataset (1xN)
05. #             dataSet:  size m data set of known vectors (NxM)
06. #             labels:   data set labels (1xM vector)
07. #             k:        number of neighbors to use for comparison
08.
09. # Output:     the most popular class label
10. ########################################
11.
12. from numpy import *
```

```
13. import operator
14.
15. # create a dataset which contains 4 samples with 2 classes
16. def createDataSet():
17.     # create a matrix: each row as a sample
18.     group = array([[1.0, 0.9], [1.0, 1.0], [0.1, 0.2], [0.0, 0.1]])
19.     labels = ['A', 'A', 'B', 'B'] # four samples and two classes
20.     return group, labels
21.
22. # classify using kNN
23. def kNNClassify(newInput, dataSet, labels, k):
24.     numSamples = dataSet.shape[0] # shape[0] stands for the num of row
25.
26.     ## step 1: calculate Euclidean distance
27.     # tile(A, reps): Construct an array by repeating A reps times
28.     # the following copy numSamples rows for dataSet
29.     diff = tile(newInput, (numSamples, 1)) - dataSet # Subtract element-wise
30.     squaredDiff = diff ** 2 # squared for the subtract
31.     squaredDist = sum(squaredDiff, axis = 1) # sum is performed by row
32.     distance = squaredDist ** 0.5
33.
34.     ## step 2: sort the distance
35.     # argsort() returns the indices that would sort an array in a ascending order
36.     sortedDistIndices = argsort(distance)
37.
38.     classCount = {} # define a dictionary (can be append element)
39.     for i in xrange(k):
40.         ## step 3: choose the min k distance
41.         voteLabel = labels[sortedDistIndices[i]]
42.
43.         ## step 4: count the times labels occur
44.         # when the key voteLabel is not in dictionary classCount, get()
45.         # will return 0
46.         classCount[voteLabel] = classCount.get(voteLabel, 0) + 1
47.
48.     ## step 5: the max voted class will return
49.     maxCount = 0
50.     for key, value in classCount.items():
51.         if value > maxCount:
52.             maxCount = value
53.             maxIndex = key
54.
55.     return maxIndex
```

然后我们在命令行中这样测试即可：

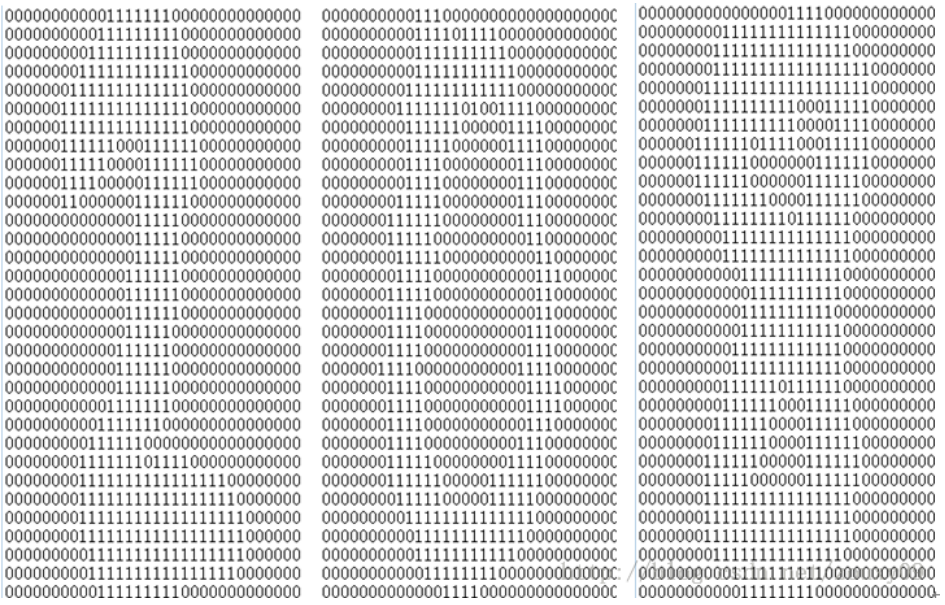
```
[python]
01. import kNN
02. from numpy import *
03.
04. dataSet, labels = kNN.createDataSet()
05.
06. testX = array([1.2, 1.0])
07. k = 3
08. outputLabel = kNN.kNNClassify(testX, dataSet, labels, 3)
09. print "Your input is:", testX, "and classified to class: ", outputLabel
10.
11. testX = array([0.1, 0.3])
12. outputLabel = kNN.kNNClassify(testX, dataSet, labels, 3)
13. print "Your input is:", testX, "and classified to class: ", outputLabel
```

这时候会输出：

```
[python]
01. Your input is: [ 1.2  1.0] and classified to class:  A
02. Your input is: [ 0.1  0.3] and classified to class:  B
```

2.2、kNN进阶

这里我们用kNN来分类一个大点的数据库，包括数据维度比较大和样本数比较多的数据库。这里我们用到一个手写数字的数据库，可以到这里下载。这个数据库包括数字0-9的手写体。每个数字大约有200个样本。每个样本保持在一个txt文件中。手写体图像本身的大小是32x32的二值图，转换到txt文件保存后，内容也是32x32个数字，0或者1，如下：



数据库解压后有两个目录：目录trainingDigits存放的是大约2000个训练数据，testDigits存放大约900个测试数据。

这里我们还是新建一个kNN.py脚本文件，文件里面包含四个函数，一个用来生成将每个样本的txt文件转换为对应的一个向量，一个用来加载整个数据库，一个实现kNN分类算法。最后就是实现这个加载，测试的函数。

```python
[python]

##########################################
# kNN: k Nearest Neighbors

# Input:      inX: vector to compare to existing dataset (1xN)
#             dataSet: size m data set of known vectors (NxM)
#             labels: data set labels (1xM vector)
#             k: number of neighbors to use for comparison

# Output:     the most popular class label
##########################################

from numpy import *
import operator
import os


# classify using kNN
def kNNClassify(newInput, dataSet, labels, k):
    numSamples = dataSet.shape[0] # shape[0] stands for the num of row

    ## step 1: calculate Euclidean distance
    # tile(A, reps): Construct an array by repeating A reps times
    # the following copy numSamples rows for dataSet
    diff = tile(newInput, (numSamples, 1)) - dataSet # Subtract element-wise
    squaredDiff = diff ** 2 # squared for the subtract
    squaredDist = sum(squaredDiff, axis = 1) # sum is performed by row
    distance = squaredDist ** 0.5

    ## step 2: sort the distance
    # argsort() returns the indices that would sort an array in a ascending order
    sortedDistIndices = argsort(distance)

    classCount = {} # define a dictionary (can be append element)
    for i in xrange(k):
        ## step 3: choose the min k distance
        voteLabel = labels[sortedDistIndices[i]]

```

```
        ## step 4: count the times labels occur
        # when the key voteLabel is not in dictionary classCount, get()
        # will return 0
        classCount[voteLabel] = classCount.get(voteLabel, 0) + 1

    ## step 5: the max voted class will return
    maxCount = 0
    for key, value in classCount.items():
        if value > maxCount:
            maxCount = value
            maxIndex = key

    return maxIndex

# convert image to vector
def  img2vector(filename):
    rows = 32
    cols = 32
    imgVector = zeros((1, rows * cols))
    fileIn = open(filename)
    for row in xrange(rows):
        lineStr = fileIn.readline()
        for col in xrange(cols):
            imgVector[0, row * 32 + col] = int(lineStr[col])

    return imgVector

# load dataSet
def loadDataSet():
    ## step 1: Getting training set
    print "---Getting training set..."
    dataSetDir = 'E:/Python/Machine Learning in Action/'
    trainingFileList = os.listdir(dataSetDir + 'trainingDigits') # load the training set
    numSamples = len(trainingFileList)

    train_x = zeros((numSamples, 1024))
    train_y = []
    for i in xrange(numSamples):
        filename = trainingFileList[i]

        # get train_x
        train_x[i, :] = img2vector(dataSetDir + 'trainingDigits/%s' % filename)

        # get label from file name such as "1_18.txt"
        label = int(filename.split('_')[0]) # return 1
        train_y.append(label)

    ## step 2: Getting testing set
    print "---Getting testing set..."
    testingFileList = os.listdir(dataSetDir + 'testDigits') # load the testing set
    numSamples = len(testingFileList)
    test_x = zeros((numSamples, 1024))
    test_y = []
    for i in xrange(numSamples):
        filename = testingFileList[i]

        # get train_x
        test_x[i, :] = img2vector(dataSetDir + 'testDigits/%s' % filename)

        # get label from file name such as "1_18.txt"
        label = int(filename.split('_')[0]) # return 1
        test_y.append(label)

    return train_x, train_y, test_x, test_y

# test hand writing class
def testHandWritingClass():
    ## step 1: load data
    print "step 1: load data..."
    train_x, train_y, test_x, test_y = loadDataSet()

    ## step 2: training...
    print "step 2: training..."
    pass

    ## step 3: testing
    print "step 3: testing..."
    numTestSamples = test_x.shape[0]
    matchCount = 0
```

```
    for i in xrange(numTestSamples):
        predict = kNNClassify(test_x[i], train_x, train_y, 3)
        if predict == test_y[i]:
            matchCount += 1
    accuracy = float(matchCount) / numTestSamples

    ## step 4: show the result
    print "step 4: show the result..."
    print 'The classify accuracy is: %.2f%%' % (accuracy * 100)
```

测试非常简单，只需要在命令行中输入：

```
[python]
import kNN
kNN.testHandWritingClass()
```

输出结果如下：

```
[python]
step 1: load d
---Getting training set...
---Getting tes
step 2: traini...
step 3: testin
step 4: show t... ...
The classify accuracy is: 98.84%
```

上一篇 Python基础学习笔记之（二）
下一篇 机器学习算法与Python实践之（二）支持向量机（SVM）初级

查看评论

10楼 瘦子什么时候才能长胖 4小时前发表

98.84%，get。

9楼 拾毅者 2015-05-22 09:19发表

向大神们学习

8楼 昱光 2015-04-15 14:58发表

楼主好，在第二个应用中，我把下载下来的digits.zip文件解压到一个文件夹后，然后更改代码中的路径后，python还是提示windows error[error 3],请问还需要修改其他地方才能够正常编译？

7楼 A_wen_A 2014-12-14 16:34发表

step 3: choose the min k distance
41. voteLabel = labels[sortedDistIndices[i]]
这个为什么能够取得K个最小的对应的标签？感觉取得标签对不上

Re: sinat_25627735 2015-04-15 23:48发表

回复A_wen_A：可以的， 你研究下argsort()函数就知道了

6楼 WYYAHU 2014-06-07 14:25发表

博主，看了您的帖子，感觉写的很好。我是个初学者，我也正在看那本机器学习的书，发现博主的程序都是自己写的，而且写的很好。有个问题想不通，就是2.2 kNN进阶这块程序第41行，这行程序在2.1也有，就是第一次get时，肯定没有值，然后值加1，第二次如果get获取不到，好像也加1了，如果第二次get获取到了，则是加了2，这个有点无法理解，是不是应该判断一下首次get？

Re: zhuqi12580 2014-09-09 17:58发表

回复WYYAHU：step 4: count the times labels occur
when the key voteLabel is not in dictionary classCount, get()
will return 0
注释说的很清楚，没有则得到0.
建议对熟悉一下语法

5楼 自由之畔 2014-06-02 11:13发表

博主你好，不知道怎么关注你的博客呢？

我这个学期需要做一个课题，关于LCS，XCS，FIXCS的应用的，简单一句话就是，用Python实现一个模糊XCS分类系统的应用。

问题是我觉得很难，不知道从哪里入手，我甚至连“状态动作对”是什么都不是很清楚。

由于在海外留学，自学要求非常高，小弟我很难找到人问，而外国同学讲解毕竟不是母语我也不是十分清楚！所以真心希望博主能拯救拯救我！

能提供一些书或者阅读资料真是最好不过了

谢谢！

4楼 大帅爷 2013-12-07 11:50发表

您好！我也在学习这本书，有个问题想请教您！
书中在说 dating 的例子的时候，用了matplotlib scatter画图。
经过下列语句后，图内的点就变成largeDoses，smallDoses，didntLike三种颜色和大小的样式了。
>>> ax.scatter(datingDataMat[:,1], datingDataMat[:,2]
15.0*array(datingLabels), 15.0*array(datingLabels))
我在使用的时候调用失败，TypeError: unsupported operand type(s) for *: 'float' and 'numpy.ndarray'。
按照我查到的scatter的属性格式http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.scatter，完全想不明白这样做为什么可行。
您能解答下吗？非常感谢~

Re: WYYAHU 2014-06-08 17:23发表

回复大帅爷：我也遇到相同问题，怎么能转换成array的呢，奇怪，你解决这个问题了吗？

3楼 sniper517 2013-11-28 10:14发表

你这是什么版本的PYTHON啊

Re: zouxy09 2013-11-28 15:06发表

回复sniper517：不好意思啊，忘了注明了。我的是Python 2.7.5

2楼 zhangzhihao66 2013-11-26 21:12发表

ValueError: invalid literal for int() with base 10: ''

Re: laocan_shi 2014-04-15 20:00发表

回复zhangzhihao66：遇到同样的问题，应该是在做int("?")时内部?不是0~9

Re: WYYAHU 2014-06-08 17:10发表

回复laocan_shi：这个问题我解决了，就是把classLabelVector.append(int(listFromLine[-1]))里面的int去掉，明显不能这样转换，因为都是文字

Re: WYYAHU 2014-06-07 19:26发表

回复laocan_shi：你们这个问题解决了吗？int()是对一段字符进行处理的，明显不行啊，不知作者为何这么写

Re: 张new 2015-10-23 21:07发表

回复WYYAHU：因为你们用错 txt 文件了，我开始也是那样，后来发现用datingTestSet2.txt就可以了。

1楼 鹳狸媛 2013-11-26 10:38发表

您的文章已被推荐到CSDN首页，感谢您的分享。

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap