机器学习实战书籍代码讲解

废话部分

自我介绍：

本人，性名：**黄海安**，网名：**深度眸**，南京航空航天大学硕士，业余工程师

录制视频的缘故：

(1) 本人懒，不愿意写字

(2) 本人爱好聊天，愿意说

本系列视频都会基于某一篇优秀博客或者某一本书来讲，视频适合对机器学习理论已经比较清楚，但是对代码掌握程度不够的同学，由于本人水平有限，如果有哪里说的不好，望大家见谅！！(欢迎提意见)。

讨论和交流：

为了让大家可以对我的视频进行评论以及后续的改进(例如推荐些优秀博客和书籍，我来讲解其代码部分，或者大家觉得有哪些知识点需要我讲解的也可以提，我会单独制作一个附属视频对问题进行讲解)，我特意建立了一个 QQ 群：ML 和 DL 视频分享群(**678455658**)，本视频以及以后的视频都会放置在里面，也会上传一些其他地方收集来的 ML 和 DL 的资料，欢迎各位加入，并提出宝贵意见！！！

一、机器学习实战第二章

开发环境：

(1) PyCharm-2017.2.3
(2) Anaconda3-4.4.0
(3) Window7-64bit

观看视频建议：本视频对原理只是简单说说，请务必先阅读对应的博客或者机器学习实战书籍，否则会听不懂！

1、本章需要掌握的知识点

本章视频对应机器学习实战书籍的第二章，或者
Jack-Cui 的博客链接：
http://blog.csdn.net/c406495762/article/details/75172850#三-k-近邻算法实战之sklearn手写数字识别

1.1 KNN 原理

1.2 KNN 优缺点

1.3 常用的向量距离度量准则

1.3.1 欧式距离

1.3.2 曼哈顿距离

1.3.3 切比雪夫距离

1.3.4 马氏距离

1.3.5 巴氏距离

1.3.6 汉明距离

1.3.7 皮尔逊系数

1.3.8 信息熵

1.4 归一化数据的重要性

以神经网络曲线拟合为例：

$$y = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n \tag{1-1}$$

θ_i 为第 i 个样本权重，x_i 为第 i 个样本特征，设优化的每一时刻均为 $|\theta_i| <= 1$，如果特征样本不进行归一化或者说特征间差距较大，例如：

$$y = \theta_0 0.1 + \theta_1 100 + \theta_2 1000 + \cdots + \theta_n 0.2 \tag{1-2}$$

则上述公式可以等效为：

$$y = \theta_2 1000 \tag{1-3}$$

可以看出，所有样本的特征只剩下第 2 个特征起作用，必然会导致欠拟合问题。

1.5 UCI 机器学习仓库

UC Irvine Machine Learning Repository 是加州大学欧文学院机器学习数据集仓库，里面有大量的数据集，主要用于机器学习算法验证

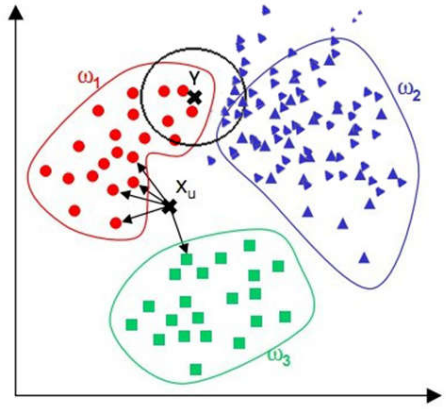
链接：http://archive.ics.uci.edu/ml/index.php

1.6 sklearn.neighbors 模块使用

KNneighborsClassifier 重要参数说明

(1) n_neighbors：默认为 5，就是 k-NN 的 k 的值，选取最近的 k 个点

(2) weights：默认是 uniform，参数可以是 uniform、distance，也可以是用户自己定义的函数。uniform 是均等的权重，就说所有的邻近点的权重都是相等的。distance 是不均等的权重，距离近的点比距离远的点的影响大。用户自定义的函数，接收距离的数组，返回一组维数相同的权重



图(1) 权重影响

上述图片来自麦子学院-彭亮的课程，感谢。

2、代码分析中不好理解部分

2.1 KNN_test01.py

diffMat = np.tile([101,20],(4,1))-dataSet

np.tile 函数作用是按照某个方向复制元素，np.tile([101,20],(4,1))意思将[101,20]在行方向上复制 4 行，列方向复制一行，转换为：

$$\begin{bmatrix} 101 & 20 \\ 101 & 20 \\ 101 & 20 \\ 101 & 20 \end{bmatrix} \tag{1-4}$$

2.2 UCI 手写数字数据集

数据集名称：Optical Recognition of Handwritten Digits

原始数据集特点：

(1) 训练样本 optdigits.tra Training 3823

(2) 测试样本 optdigits.tes Testing 1797

(3) 每个样本：64 input+1 class attribute

(4) 每个像素的取值范围：0..16
(5) 数字范围：0..9
(6) 其中对于每个样本，64 是指每个数字是 8X8 的像素，将其转换为 0,1 矩阵，则为 32 X 32 大小---这句话写错了

2.3 视频勘误

不好意思，由于本人第一次讲课，比较激动，录制完视频后发现最后 1 分钟时候话讲错了！

错误的位置是：48:10 分钟后，讲错了由 64 转 32x32 的方式，并不是视频中的那样，2.2(6)的写法是错误的，下面是正确的：

其中对于每个样本，64 是指每个数字是 8X8 的像素，其中最原始的数据就是图片格式，是 32X32 的黑白 bitmap 图片（每个点只有 0 和 1），这些图片来源于 44 位不同的人的手写数字，图片已归一化为以手写数字为中心的 32*32 规格的图片。

(1) 将 32X32 图片切割为不重叠的以 4X4 像素为单位的 64 个小块，从而转换为 8X8 的像素值
(2) 8X8 的像素值范围是 0…16，原因是 4X4 的小块，范围就是 0(16 个 0 相加)…16(16 个 1 相加)
(3) 然后将 8X8 的像素值平铺，从而转换为 64 长度的向量，即视频中所展示的那样,例如 0,1,6,15,12,1,0,0,0,7…
(4) 所以，如果我们最初拿到的是 64 长度的数据，反向即可还原为 32X32 的黑白图片

二、机器学习实战第三章

1、本章需要掌握的知识点

本章视频对应机器学习实战书籍的第三章，或者

Jack-Cui 的博客链接：

http://blog.csdn.net/c406495762/article/details/75663451

http://blog.csdn.net/c406495762/article/details/76262487#六-sklearn 之使用决策树预测隐形眼镜类型

请先确保看过博客或者书籍，否则可能听不懂

1.1 决策树原理

适用场景：决策树能够生成清晰的基于特征(feature)选择不同预测结果的树状结构，希望更好的理解手上的数据的时候往往可以使用决策树，在实际应用中，受限于它的简单性，决策树更大的用处是作为一些更有用的算法的基石，例如随机森林。

1.2 决策树优缺点

(1) 计算复杂度不高，输出结果易于理解

以 ID3 为例，每次运算都是基于某一列特征，特征计算完后，下次计算不考虑该最优特征，并且通过适当剪支可以简化复杂度

(2) 对中间值的缺失不敏感

这个不好理解，下次补上

(3) 可以处理不相关特征数据

是基于每一列特征来计算，不考虑特征之间的依赖关系

详情见博客：http://blog.csdn.net/xuxiatian/article/details/54340428

1.3 决策树算法种类

(1) ID3

以信息增益作为树的分裂准则，该算法存在的不足：

(a) ID3 没有考虑连续特征，比如长度，密度都是连续值，无法在 ID3 运用，如果一定要运用 ID3 出来连续属性，那么要自己将连续特征离散化(办法非常多)

(b) 对于缺失值的情况没有做考虑

(c) 偏向于多值属性。例，如果存在唯一标识属性 ID(每个样本的 ID 属性值都不相同)，则 ID3 会选择它作为优先分裂属性，这样虽然使得划分充分纯净，但这种划分对分类几乎毫无用处

(2) C4.5

(a)以基于信息增益的增益率（gain ratio）作为树的分裂准则，解决了 ID3 的偏向于多值属性问题

(b)内部自己考虑了连续属性离散化过程，所以克服了 ID3 的没有考虑连续特征问题

(c)内部考虑了缺失值的自动处理策略

(3) CART

ID3 和 C4.5 只能处理分类问题，而 CART 可以处理分类和回归问题，CART 考虑问题非常全面，有较多优点，可以自行深入研究

详细分析，请见优秀系列博客：http://www.cnblogs.com/pinard/p/6050306.html

1.4 信息熵的深入理解

信息熵用于度量信息的混乱程度，信息越混乱说明能够包含的信息量越多，则熵越大，例如一个声波，我们可以通过傅里叶变换分析其频谱看到其中的大量的信息；信息越有序说明包含的信息量越少，则熵越小，例如一条直线，这个信息太少了，则它的熵也非常小。

在数学上，对于任意一个向量，对其计算信息熵，可以证明出：当向量中每个值都相同的时候，熵最小。这样数学和通俗理解就对应上了。

1.5 信息增益(互信息)的深入理解

信息增益用于度量一个随机变量中包含的关于另一个随机变量的信息量，或者说是一个随机变量由于已知另一个随机变量而减少的不肯定性，也可以简单认为一个随机变量的引入导致了另一个随机变量的混乱性变化(约束)。信息增益是特征选择的重要指标，它定义为一个特征能够为分类系统带来多少信息，带来的信息越多，说明更容易分类，也说明该特征越重要，相应的信息增益也就越大

$$g(D,A)=H(D)-H(D\mid A) \tag{2-1}$$

非对称性：$g(D,A)\neq g(A,D)$ (2-2)

D 是标签对应的列向量，A 是其中的某一列特征，g 是信息增益，从公式来理解更容易：**信息增益越大，特征对最终的分类结果影响也就越大？为何 g 要越大越好**：类标签列信息熵 $H(D)$ 是固定值，g 要大，说明条件熵 $H(D\mid A)$ 要小，条件熵意思是在已知 A 的情况下，对于类标签列带来了多少约束，假设在确定 A 的情况下，类标签列只有一个取值(此时 D 的信息熵最小)，信息类似于直线向量，说明这个约束非常大，$H(D\mid A)$ 达到最小，同时也可以明显看出，这个特征对分类很有用，只要使用了该特征，就可以将类别全部分开，此次我们说，这个特征对分类贡献最大，那么其信息增益也是最大的。

1.6 sklearn 中 tree 模块使用

重要参数分析：

(1) criterion：特征选择标准，可选参数，默认是 gini(CART)，可以设置为 entropy(ID3)

(2) 特征划分点选择标准 splitter：主要考虑计算量

(3) 划分时考虑的最大特征数、决策树最大深、内部节点再划分所需最小样本数、叶子节点最少样本数、叶子节点最小的样本权重和、最大叶子节点数、节点划分最小不纯度：这些策略都是为了防止过拟合

(4) 类别权重:主要是克服样本不平衡问题

还有一些注意事项，请看 **Jack-Cui** 博客

2、代码分析中不好理解部分

2.1 递归算法理解

递归算法中，有两个固定步骤：递归头和递归体，缺一不可。

(1) 递归头：什么时候不调用自己的方法，即递归的结束条件

(2) 递归体：什么时候需要调用自己的方法，即自己调用自己

递归算法的优点：将问题逐渐简单化；缺点：会占用大量的系统堆栈，内存耗用多，早递归调用层数多时，比循环慢很多。

2.2 树的递归划分举例

假设第 2 列是最优特征，使用该特征作为根节点，进行递归，则原来的 dataSet，会变成两个子 dataSet,然后对这两个子 dataSet 分别进行递归创建树,直到满足结束条件

2.3 信息增益计算举例

数据：

```
dataSet = [[0, 0, 0, 0, 'no'],
           [0, 0, 0, 1, 'no'],
           [0, 1, 0, 1, 'yes'],
           [0, 1, 1, 0, 'yes'],
           [0, 0, 0, 0, 'no'],
           [1, 0, 0, 0, 'no'],
           [1, 0, 0, 1, 'no'],
           [1, 1, 1, 1, 'yes'],
           [1, 0, 1, 2, 'yes'],
           [1, 0, 1, 2, 'yes'],
           [2, 0, 1, 2, 'yes'],
           [2, 0, 1, 1, 'yes'],
           [2, 1, 0, 1, 'yes'],
           [2, 1, 0, 2, 'yes'],
           [2, 0, 0, 0, 'no']]
labels = ['年龄', '有工作', '有自己的房子', '信贷情况']
```

$H(D)$的计算：

只看最后一列，其中总共 15 个样本，yes 类别有 9 个，no 类别有 6 个，所以信息熵为：

$$H(D) = -(\frac{9}{15}\log_2\frac{9}{15} + \frac{6}{15}\log_2\frac{6}{15}) = 0.971 \tag{2-3}$$

信息增益的计算：

$$g(D, age) = H(D) - H(D \mid age) \tag{2-4}$$

$$= H(D) - (H(D \mid youth, middle, old)$$

$$= H(D) - [p(youth)H(D \mid A = youth) + p(middle)H(D \mid A = middle) + p(old)H(D \mid A = old)]$$

$$= 0.971 - \frac{5}{15}[H(D \mid A = youth) + H(D \mid A = middle) + H(D \mid A = old)]$$

$H(D \mid A = youth)$ 中，一共 5 个样本类别，yes 有 2 个，no 有 3 个：

$$H(D \mid A = youth) = -(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}) \tag{2-5}$$