# 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：

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

$\theta_i$ 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

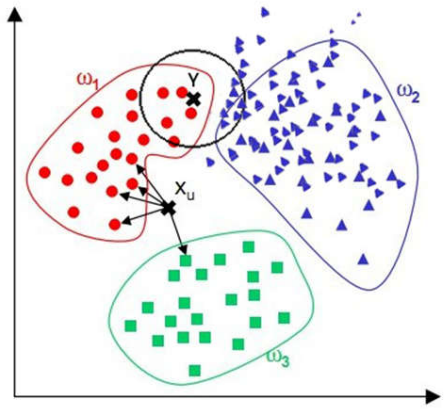
链接：http://archive.ics.uci.edu/ml/index.php

## 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