机器学习实战

Stephen CUI[1]

February 17, 2023

[1]cuixuanStephen@gmail.com

Contents

Part I

分类

前两部分主要探讨监督学习（supervised learning）。在监督学习的过程中，我们只需要给定输入样本集，机器就可以从中推演出指定目标变量的可能结果。

监督学习一般使用两种类型的目标变量：标称型和数值型。标称型目标变量的结果只在有限目标集中取值，如真与假、动物分类集合 { 爬行类、鱼类、哺乳类、两栖类 } ；数值型目标变量则可以从无限的数值集合中取值，如 0.100、42.001、1000.743 等。数值型目标变量主要用于回归分析。

Chapter 1

k-近邻算法

众所周知，电影可以按照题材分类，然而题材本身是如何定义的?由谁来判定某部电影属于哪个题材?也就是说同一题材的电影具有哪些公共特征?这些都是在进行电影分类时必须要考虑的问题。那么动作片具有哪些共有特征，使得动作片之间非常类似，而与爱情片存在着明显的差别呢？动作片中也会存在接吻镜头，爱情片中也会存在打斗场景，我们不能单纯依靠是否存在打斗或者亲吻来判断影片的类型。但是爱情片中的亲吻镜头更多，动作片中的打斗场景也更频繁，基于此类场景在某部电影中出现的次数可以用来进行电影分类。

1.1 k-近邻算法概述

简单地说，k-近邻算法采用测量不同特征值之间的距离方法进行分类。

k-近邻算法

优点：精度高、对异常值不敏感、无数据输入假定。
缺点：计算复杂度高、空间复杂度高。
适用数据范围：数值型和标称型。

k-近邻算法（kNN），它的工作原理是：存在一个样本数据集合，也称作训练样本集，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。

1.1.1 实施kNN算法

k-近邻算法给出k-近邻算法的伪代码：

代码清单

kNN.py
kNN.ipynb

Algorithm 1 k-近邻算法

对未知类别属性的数据集中的每个点依次执行以下操作：

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。

classify0()函数有4个输入参数：用于分类的输入向量是inX，输入的训练样本集为dataSet，标签向量为labels，最后的参数k表示用于选择最近邻居的数目，其中标签向量的元素数目和矩阵dataSet的行数相同。

计算完所有点之间的距离后，可以对数据按照从小到大的次序排序。然后，确定前k个距离最小元素所在的主要分类，输入k总是正整数；最后，将classCount字典分解为元组列表，然后使用程序第二行导入运算符模块的itemgetter方法，按照第二个元素的次序对元组进行排序。此处的排序为逆序，即按照从最大到最小次序排序，最后返回发生频率最高的元素标签。

1.1.2 如何测试分类器

分类器并不会得到百分百正确的结果，我们可以使用多种方法检测分类器的正确率。此外分类器的性能也会受到多种因素的影响，如分类器设置和数据集等。不同的算法在不同数据集上的表现可能完全不同。

为了测试分类器的效果，我们可以使用已知答案的数据，当然答案不能告诉分类器，检验分类器给出的结果是否符合预期结果。通过大量的测试数据，我们可以得到分类器的错误率——分类器给出错误结果的次数除以测试执行的总数。错误率是常用的评估方法，主要用于评估分类器在某个数据集上的执行效果。

1.2 示例：使用 k-近邻算法改进约会网站的配对效果

我的朋友海伦一直使用在线约会网站寻找适合自己的约会对象。海伦希望我们的分类软件可以更好地帮助她将匹配对象划分到确切的分类中。此外海伦还收集了一些约会网站未曾记录的数据信息，她认为这些数据更有助于匹配对象的归类。

1.2.1 准备数据：从文本文件中解析数据

海伦收集约会数据已经有了一段时间，她把这些数据存放在文本文件datingTestSet.txt中，每个样本数据占据一行，总共有1000行。海伦的样本主要包含以下3种特征：

- 每年获得的飞行常客里程数
- 玩视频游戏所耗时间百分比
- 每周消费的冰淇淋公升数

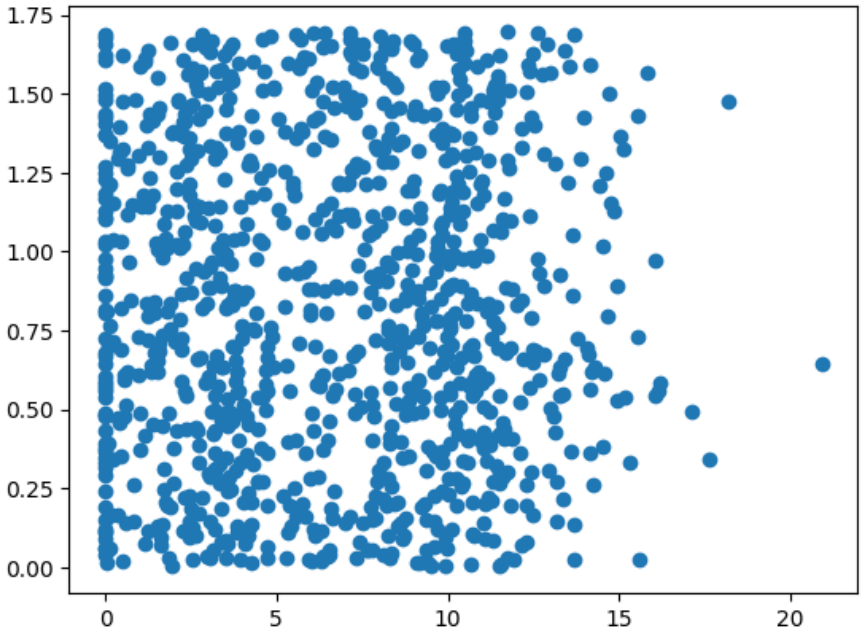

Figure 1.1: Dating data without class labels

在将上述特征数据输入到分类器之前，必须将待处理数据的格式改变为分类器可以接受的格式。在kNN.py中创建名为file2matrix的函数，以此来处理输入格式问题。该函数的输入为文件名字符串，输出为训练样本矩阵和类标签向量。

修改一下kNN.py文件，向其添加file2matrix()函数。

首先我们需要知道文本文件包含多少行。打开文件，得到文件的行数。然后创建以零填充的矩阵NumPy。将该矩阵的另一维度设置为固定值3，你可以按照自己的实际需求增加相应的代码以适应变化的输入值。循环处理文件中的每行数据，首先使用函数line.strip()截取掉所有的回车字符，然后使用tab字符\t将上一步得到的整行数据分割成一个元素列表。接着，选取前3个元素，将它们存储到特征矩阵中。需要注意的是，我们必须明确地通知解释器，告诉它列表中存储的元素值为整型，否则Python语言会将这些元素当作字符串处理。

成功导入datingTestSet.txt文件中的数据之后，可以简单检查一下数据内容。

接着我们需要了解数据的真实含义。当然我们可以直接浏览文本文件，但是这种方法非常不友好，一般来说，我们会采用图形化的方式直观地展示数据。

1.2.2 分析数据：使用 Matplotlib 创建散点图

由于没有使用样本分类的特征值，我们很难从Figure 1.1中看到任何有用的数据模式信息。一般来说，我们会采用色彩或其他的记号来标记不同样本分类，以便更好地理解数据信息。Matplotlib库提供的scatter函数支持个性化标记散点图上的点。

1.2.3 准备数据：归一化数值

Table 1.1给出了提取的四组数据，如果想要计算样本3和样本4之间的距离，可以使用下面的方法：

$$\sqrt{(0-67)^2+(20000-32000)^2+(1.1-0.1)^2}$$

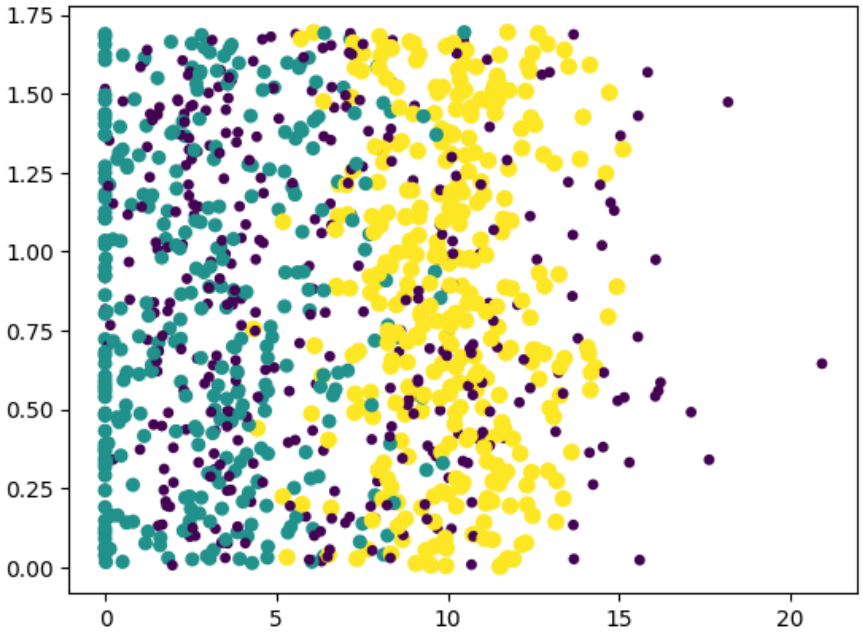

Figure 1.2: Dating data with markers changed by class label

Table 1.1: 约会网站原始数据改进之后的样本数据

	玩视频游戏所耗时间百分比	每年获得的飞行常客里程数	每周消费的冰淇淋公升数	样本分类
1	0.8	400	0.5	1
2	12	134, 000	0.9	3
3	0	20, 000	1.1	2
4	67	32, 000	0.1	2

我们很容易发现，上面方程中数字差值最大的属性对计算结果的影响最大，也就是说，每年获取的飞行常客里程数对于计算结果的影响将远远大于表2-3中其他两个特征——玩视频游戏的和每周消费冰淇淋公升数——的影响。而产生这种现象的唯一原因，仅仅是因为飞行常客里程数远大于其他特征值。但海伦认为这三种特征是同等重要的，因此作为三个等权重的特征之一，飞行常客里程数并不应该如此严重地影响到计算结果。

在处理这种不同取值范围的特征值时，我们通常采用的方法是将数值归一化，如将取值范围处理为0到1或者　1到1之间。下面的公式可以将任意取值范围的特征值转化为0到1区间内的值：

$$newValue = \frac{oldValue - min}{max - min}$$

其中min和max分别是数据集中的最小特征值和最大特征值。虽然改变数值取值范围增加了分类器的复杂度，但为了得到准确结果，我们必须这样做。我们需要在文件kNN.py中增加一个新函数autoNorm()，该函数可以自动将数字特征值转化为0到1的区间。

在函数autoNorm()中，我们将每列的最小值放在变量minVals中，将最大值放在变量maxVals中，其中dataSet.min(0)中的参数0使得函数可以从列中选取最小值，而不是选取当前行的最小值。然后，函数计算可能的取值范围，并创建新的返回矩阵。为了归一化特征值，我们必须使用当前值减去最小值，然

后除以取值范围。需要注意的是，特征值矩阵有1000×3个值，而minVals和range的值都为1×3。为了解决这个问题，我们使用NumPy库中tile()函数将变量内容复制成输入矩阵同样大小的矩阵，注意这是具体特征值相除