Chapter 3
Classification

DuLi

2019 年 4 月 9 日

目录

第一章我们提到过，监督学习中最常见的任务是回归问题（预测）和分类问题。第二章我们探索了回归问题，这一章我们重点关注分类问题。

1 MNIST

In this chapter,we will be using the MNIST dataset,which is a set of 70000 small images of digits handwritten by high school students and employees of the US Census Bureau.Each image is labeled with the digit it represents.这个数据集经常被当作Machine Learning中的Hello World问题：这个数据集也经常被用来测试一个新的分类算法。

Scikit-Learn provides many helper functions to download popular datasets.MNIST is one of them.The following code fetches the MNIST dataset:

```
from sklearn.datasets import fetch_mldata

mnist = fetch_mldata('MNIST original',data_home='./')
mnist
```

不知道为什么，下载一直有问题，远程连接关闭，所以直接下载下来放在文件夹就OK了。

```
In [7]: from sklearn.datasets import fetch_mldata

In [11]: mnist = fetch_mldata('MNIST original',data_home='./')

In [12]: mnist

Out[12]: {'COL_NAMES': ['label', 'data'],
          'DESCR': 'mldata.org dataset: mnist-original',
          'data': array([[0, 0, 0, ..., 0, 0, 0],
                 [0, 0, 0, ..., 0, 0, 0],
                 [0, 0, 0, ..., 0, 0, 0],
                 ...,
                 [0, 0, 0, ..., 0, 0, 0],
                 [0, 0, 0, ..., 0, 0, 0],
                 [0, 0, 0, ..., 0, 0, 0]], dtype=uint8),
          'target': array([0., 0., 0., ..., 9., 9., 9.])}
```

图 1: Overview of the MNIST dataset

先简单说一下数据集的结构：

- DESCR关键字存储的是数据集的描述。也就是一句话，mldata.org dataset:mnist-original.
- data关键字储存的一行一行的数据，每一行存储一个字符的像素.
- target关键字保存的是对应列代表的数字。

There are 70,000 images and each image has 784 features(28*28=784 pixels),and each feature simply represents one pixel's intensity ,from 0(white) to 255(black).Let's take a peek at one digit from the dataset.We grab the vector and reshape it to a 28X28 array.

```
X,y = mnist['data'],mnist['target']

import matplotlib
import matplotlib.pyplot as plt

some_digit = X[1123]
some_digit_image = some_digit.reshape(28,28)

plt.imshow(some_digit_image,cmap=matplotlib.cm.binary,interpolation="nearest")
plt.axis("off")
plt.show()
```

图 2: Random digits in the data set

很明显，这是一个0，值得注意的是，MNIST数据集已经分割好了训练集和测试集，分别是前60000和后10000张图像。

```
X_train, X_test, y_train, y_test = X[:60000], X[60000:], y[:60000], y[60000,]
```

同时，我们还要shuffle数据集。主要原因在于这样在做cross-vaildation的时候，可以让每一次fold都很接近。并且，一些算法对于训练集的先后顺序是敏感的，如果连续的输入相同的训练集将会影响算法的性能。

```
import numpy as np

shuffle_index = np.random.permutation(60000)
X_train, y_train = X_train[shuffle_index], y_train[shuffle_index]
```

2 Training a Binary Classifier