

**模式识别大作业**

题 目 数字识别器(Digit recognizer)

学 院 信息科学与工程

专 业 控制科学与工程

组 员 蒋嘉石

指导教师 赵海涛

**完成日期： 2019 年 12 月12日**

**模式识别大作业报告——数字识别器（Digit recognizer）**

组员：蒋嘉石

经过半个学期的对模式识别课程的学习，在赵海涛老师的辛勤指导下，我对模式识别有了一定的了解。由于仅有32课时的课程，在课堂上仅能了解简单的原理，并且理解也不深刻，难以真正学到精髓。故通过本次大作业对上课所学的知识进行一些应用，以期将课上学到的知识巩固加强。

本次课题选择了Kaggle上一个入门级别的课题，数字识别器（Digit reconizer），辨识经典的MNIST手写数据，应用PCA的方法对手写数据的灰度进行降维，再用SVM实现分类。经过了一段时间的研究，最终识别的准确率达到了97.45%，并且能够输出CSV用于上传至Kaggle。

**1 Kaggle与Digit recognizer简介**

Kaggle上有着各种高质量的数据集，基于这些数据集会有一些大公司进行赞助，举办一些算法竞赛，在竞赛时还可以通过社区进行讨论，使得全球的顶尖高手都会对参加比赛产生兴趣。同时Kaggle对模式识别与机器学习的新手也十分友好，在社区中可以学到许多知识与解决方案。

Kaggle中的比赛主要可以分为两类：

竞赛——Competitions：要求在指定期限内，使用出题方的数据完成指定的任务，如果赢得比赛，可以获得奖金，模型也可能会被竞赛赞助商应用到商业实践中，获奖者可能可以直接进入赞助企业工作或者获得一些重大会议邀请等殊荣。

数据集——datasets：数据集是赞助商为了解决某些问题，免费公开了自己的内部的一些脱敏数据，所有人都可以使用这些数据集进行研究，来改进现有的模型或者优化现有问题的解决方案。解决方案可以不公开，故datasets的内容不提供奖金，仅用于与他人的方案效果进行比较。

Digit recognizer这一题目是属于竞赛类别的课题，并且其属于Knowledge的分类，Kaggle不仅提供了MNIST的手写数据集，也提供了一些示例算法以供学习。MNIST数据集来自美国国家标准与技术研究所, National Institute of Standards and Technology (NIST)。训练集(training set)由来自250个不同人手写的数字构成, 其中50%是高中学生, 50% 来自人口普查局(the Census Bureau)的工作人员。测试集(test set)也是同样比例的手写数字数据。该实验的目的就是通过训练集对模型进行训练，再对测试集进行分类，尽可能提高辨识精度。

**2 整体解决方案**

**2.1数据结构分析**

MINST\_train数据是一个42000行，785列的数据矩阵。MINST\_test数据是一个42000行，784列的数据矩阵。Train数据中第一列是标签列（label），也就是这一行数据所表示的一个0-9数字。Test数据则不含标签。后784列数据表示的是表示手写数据的灰度矩阵，每个数值在0-255之间。

考虑到时间以及变成复杂度问题，测试集我们选择10000×784的数据大小进行求解。

**2.2数据读入**

本次程序应用Python进行编程，读取数据使用了Numpy模块，可以将csv格式的数据集读成数int格式，便于接下来的操作。

|  |
| --- |
| from sklearn.decomposition import PCA  from sklearn.svm import SVC  import pandas as pd  import numpy as np  pca = PCA(n\_components=0.95, whiten=True)  a=np.loadtxt("E:\BaiduNetdiskDownload\mnist\_train.csv",dtype=int,delimiter=',')  #加载数据为ndarray  digit=a[np.arange(0,42000)] #提取多行  digit = pd.DataFrame(digit)  b=np.loadtxt("E:\BaiduNetdiskDownload\mnist\_test.csv",dtype=int,delimiter=',')  #加载数据为ndarray  test1=b[np.arange(0,10000)] #提取多行  test = pd.DataFrame(test1) |





由图可以看出，读取训练集为digit，42000×785的数据大小；读取测试集为test，10000×785的数据大小。前一节中提到，test数据中是不含label行的，但是由于Kaggle下载数据集需要翻墙，于是从其他来源获得了MNIST的数据。这组数据的test集中含有label列，故如图所示，test也有785列。在之后需要降label去除。再上图代码中，首先定义了pca函数，令剩余方差百分比为95%，并使得数据白化，也就是使数据方差相等。由于需要提取数据集中的多行数据，DataFrame格式难以操作，故先使用np.loadtxt获得int格式的数据，再提取训练集和测试集的前42000行和10000行，再使用Pandas模块中的pd.Dataframe函数将数据集转化回Dataframe进行后续操作。

|  |
| --- |
| label = digit.values[:, 0].astype(int)  train = digit.values[:, 1:].astype(int)  test\_data = test.values[:, 1:].astype(int) |

该步骤首先将训练集digit集合中的第一列取出，作为label进行备用。将去除label的剩余数据作为训练集train，亦是后续步骤的一个输入参数。由于前面所提到的测试集包含标签的问题，故还需要将测试集中的label去除，以得到正确的数据集，保存为test\_data。

**2.3模型训练与分类结果输出**

该部分的代码如下：

|  |
| --- |
| pca = PCA(n\_components=0.95, whiten=True)  pca.fit(train)  train\_data = pca.transform(train)  svc = SVC()  svc.fit(train\_data, label)  test\_data = pca.transform(test\_data)  ans = svc.predict(test\_data)  a = []  for i in range(len(ans)):  a.append(i+1)  np.savetxt('PCA\_0.95\_SVC.csv', np.c\_[a, ans],  delimiter=',', header='ImageId,Label', comments='', fmt='%d')  test1=test1[:,0] |

其中所用到的PCA模块和SVC模块已经在先前的代码中引入，如下：

|  |
| --- |
| from sklearn.decomposition import PCA  from sklearn.svm import SVC |

考虑到灰度矩阵由784维的数据，可以认为是高维数据。故考虑使用主元分析(PCA)的方法，先对训练集train进行降维，使得SVM的分类速度可以加快。此处将PCA的剩余方差百分比n\_components设置为0.95，即95%。并设置将数据白化。使用pca.fit()函数来训练train数据，随后使用pca.transform()函数再将获得的pca模型train进行降维，得到了降维后的模型train\_data。



可以看到，train\_data从原来784列降维到了154列，有效地提高了效率。

使用PCA完成降维之后，就使用SVM对train\_data数据和label数据进行分类工作。直接使用缺省的SVC()，然后使用函数svc.fit(train\_data,label)完成模型的学习。完成了SVM模型的训练后，将测试集test\_data也进行PCA的降维工作后，使用svc.predict(test\_data)函数，得到SVM对测试集的预测结果，输出到ans当中。如下图所示，ans是一个10000行一列的预测数据，对测试集的每一行都进行了预测。



使用numpy模块中的savetxt函数将得到的ans数据输出成csv格式的数据，至此就完成了可以上传至Kaggle的结果。

|  |
| --- |
| n = 0  acry = 0  for n in range(10000):  if test1[n,] == ans[n,]:  acry =acry+1  else:  acry =acry+0  print(acry/10000) |

由于我的测试集是有标签列的，故可以直接计算模型的预测结果。以上代码表示当预测集中的数与标签集中数据相同时，就进行一次计数，最后可以得到正确预测次数的百分比。



如图所示，在10000个数据中，acry也就是正确预测的数量是9745个，也就是说准确率达到了97.45%。在Kaggle的该项中2200个参赛者中排名1500左右，可以说准确度不是很高。

**三、小组分工**

程序设计及编写：蒋嘉石

程序调试：蒋嘉石

实验报告：蒋嘉石

**四、小结**

本次模式识别大作业，是我第一次独立地思考着手相关的课题。模式识别这一门课程同本科的知识体系相去甚远，需要有扎实的概率统计和矩阵分析的知识体系作为支撑，才能够很好地理解模式识别与机器学习的内容。课堂上的时间确实只够进行一个大概的介绍，故本次作业确实让我收获颇丰。大作业不仅要求对知识的理解，还要求一定的编程能力。本次作业使用了python语言以及其中关于机器学习方法的模块，虽然只是简单的调用模块和缺省参数的函数，但我对这一个过程有了系统的了解。我体会到了python是一种十分易用的语言，正确地调用模块和函数就能够达到一定的效果，十分易学易用。

在Kaggle中选择Digit recognizer这一课题主要是因为这一课题既成熟，网络上的教程和讨论也十分丰富，很适合初学者学习模式识别和python编程。同时这一课题对最后预测结果的评价十分简单，可以直观的知道模型的精度。应用PCA对训练和测试集进行降维在这个案例中十分奏效，将784维的数据降低成154维，数据大小降低了约5倍，这确实提高了程序的效率。由于本人对模式识别仅是一个初学者，故本次只是简单调用了函数，而未对其各种参数进行调试。

这次作业丰富了我对模式识别的认识，知道了Kaggle这样一个可以通过实践来学习知识的优秀平台，对我以后的学习应有很大的裨益。感谢赵海涛老师的教学，给了我一次很好的学习机会。