**模式识别大作业**

题 目 手写数字识别

学 院 信息科学与工程

专 业 控制科学与工程

组 员 彭竞超、陈书田、李宜儒

指导教师 赵海涛

**完成日期： 2019 年 12 月10日**

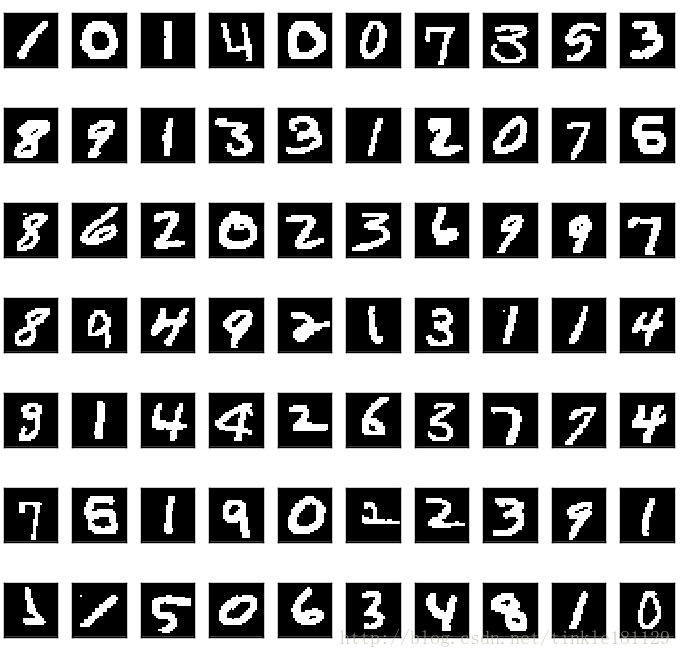
**模式识别大作业报告**

组员：彭竞超、陈书田、李宜儒

研一上半学期的学习进程告一段落，模式识别课程也进行了结课考试，在赵老师的辛苦授课和指导下，我们小组在彭竞超同学的组织和带领下，进行了有关模式识别和机器学习方向的实践，我们认为只有将理论和实践结合起来，才算真正掌握了这门课所传授的知识，而赵老师所布置的大作业也是很好的锻炼我们。

经过小组分工，我们顶下了题目，拟定了解决问题的方法，然后分工组员去实现。下面则是我们遇到的问题和解决问题的过程。

**一、手写数字识别**

手写数字识别这个项目是在Kaggle平台上的一个竞赛项目。而Kaggle是由联合创始人、首席执行官安东尼·高德布卢姆2010年在墨尔本创立的，主要为开发商和数据科学家提供举办机器学习竞赛、托管数据库、编写和分享代码的平台。该平台已经吸引了80万名数据科学家的关注，这些用户资源或许正是吸引谷歌的主要因素。主要是为开发商和数据科学家提供举办机器学习竞赛、托管数据库、编写和分享代码的平台。这一平台已经吸引了许多科学家和开发者的关注，他们也纷纷入驻这一平台。这些科学家和开发者资源正是[谷歌](https://baike.so.com/doc/108804-114835.html" \t "https://baike.so.com/doc/_blank)看中他们的地方 。

**二、准备知识**

手写数字识别的核心就是识别出0~9。解决方案：PCA降维+svm分类。

**2.1 主成分分析**

主成分分析（Principal Component Analysis）法作为一种强有力的正交化工具，对存在相关性的数据具有非常好的去相关性作用。并且在利用主成分分析的过程中，可以对特征信息进行提取，对数据的维数进行降低。

实际工业过程中，会遇到许多变量同时进行检测和分析，如设备的流量、压力、温度、物料浓度、设备老化程度等等，变量数量可能多达几十个甚至几百个，对于如此多的数据进行检测，会发现有的数据变化呈现同增同减，亦或是此消彼长的关系。而且在数据量十分庞大的时候，分析起来非常不便，因此希望采用较少的变量进行对原来数据信息进行替代，并且希望彼此互不相关，由此产生了主成分分析法的基本想法：能否对原来多个变量进行合适的组合，组合成一些综合指标，用较少的综合指标，来近似替代原来多个变量，这种由原来多个变量组合而成的综合指标，就称为主成分。

**PCA算法步骤**

输入：数据，降维维数，数据个数，

输出：，

求解的均值

求解矩阵

对矩阵进行特征分解，，，

令即为我们要求的变换矩阵

，，

**2.2 支持向量机**

支持向量机 (Support Vector Machine，SVM)是一种广义上的线性分类器。对于一个两类的分类问题，一个线性分类器的学习目标是在数据空间里寻找到一个超平面

在这个超平面的两侧，样本属于不同的类。

分类器的设计思路，是从分界面所留下的裕度大小来入手的，希望所设计的分类器，尽可能给数据留下最大的裕度。

换言之，我们认为数据确实有可能是“波动的”，但是变化的很小的情况下，我们应该能够给出更好的分类结果。而不至于发生越界情况。

因此我们希望最后能获得**线性分类器，到数据集合的最短距离，尽可能最大**。

以线性分类器，在分类器边界上的样本则有，。根据图示，令分界面外一点到到分界面的距离为，在分界面上的投影为，分界面的单位法向量为。假设(的情况，讨论是类似的)，根据上图有：

因此有：

因为有

则有：

从而可得,

用表示数据集，表示对应元素所属类的判别值。则有

按照上述思路则有如下目标。

目标：找到分界直线（分界面），并且有

或者可以写成：

那么所要求的直线参数问题，可以化为如下的优化问题：

其中，。

假设有：

那么就有：

即：

那么优化问题可以转化为：

然而从众多个，找到一个最小值，然后除以，其中的最小值由约束条件可知为1。因此可以将问题转化成：

由于要求上述式子的极大，相当于求解

易见， 和 表示的是同一条直线，因此在下面的推导中仍用和作为我们待求的未知参数。

也就相当于求解：

此时运用拉格朗日数乘法有：

由KKT条件，可得：

从约束条件可知，当时，为0。换言之，此时的并不是支持向量，只有当不为零的时候，向量才是支持向量。将KKT条件得到的关系式代入原优化问题，可得

**三、解决方案**

**2.1 数据结构分析**

首先在Kaggle上把需要的数据集下载下来。如表2-1所示，

**表2-1 训练集表（部分示例）**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| label | pixel0 | pixel1 | pixel2 | pixel3 | pixel4 | pixel5 | pixel6 | pixel7 | pixel8 | pixel9 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

该train.csv是训练样本集，大小为60001\*784。第一行为文字描述，第一列是标签label对应的数字，所以实际的样本数据大小60000\*784。第一列单独列出得到60000\*1的向量label，剩下的则为60000\*783的特征向量集。

而test.csv是测试集，数据内容和训练集相似，就不介绍相同点。在测试集中，实际的样本数据大小50000\*784，也不存在label这一列。我们要做的工作就是为这些测试样本找出正确的label。

根据我们所拥有的数据集（训练集和测试集）得到的矩阵有：trainDate、TrainLabel、TeatDate、TestResult。

**2.2 主成分保留选取实验**

from sklearn.decomposition import PCA

import matplotlib.pyplot as plt

import pandas as pd

pca = PCA(whiten=True)

digit = pd.read\_csv('train.csv')

train = digit.values[1:, :].astype(int)

pca.fit(train)

exr = pca.explained\_variance\_ratio\_

x = []

y = []

line85 = []

line98 = []

for i in range(len(exr)) :

x.append(i)

line85.append(0.85)

line98.append(0.98)

if i == 0:

y.append(exr[0])

else:

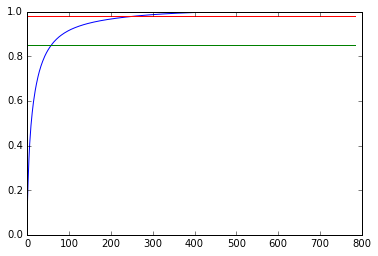
y.append(exr[i]+y[i-1])

plt.plot(x, y)

plt.plot(x, line85)

plt.plot(x, line98)

plt.show()

从图中可以得出保留98%的能量大概需要154个左右的维度

**2.3 识别实验**

# -\*- coding: utf-8 -\*-

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)

digit = pd.read\_csv('train.csv')

test = pd.read\_csv('test.csv')

label = digit.values[:, 0].astype(int)

train = digit.values[:, 1:].astype(int)

test\_data = test.values[:, :].astype(int)

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')

调参数总共调了三次：

第一次用0.95，结果是0.9737。

第二次用0.85，结果是0.9815。

第三次直接试保留10个维度，结果是0.9341，比之前低了许多。

**2.4 实验结果**

参数选择95%时，前50个测试数据真实值和预测值如下:

|  |  |  |
| --- | --- | --- |
| 序号 | 真实值 | 实验值 |
| 1 | 2 | 2 |
| 2 | 1 | 1 |
| 3 | 0 | 0 |
| 4 | 4 | 4 |
| 5 | 1 | 1 |
| 6 | 4 | 4 |
| 7 | 9 | 9 |
| 8 | 5 | 5 |
| 9 | 9 | 9 |
| 10 | 0 | 0 |
| 11 | 6 | 6 |
| 12 | 9 | 9 |
| 13 | 0 | 0 |
| 14 | 1 | 1 |
| 15 | 5 | 5 |
| 16 | 9 | 9 |
| 17 | 7 | 7 |
| 18 | 3 | 3 |
| 19 | 4 | 4 |

**三、小组分工**

程序设计及编写：李宜儒、陈书田

程序调试及修改：彭竞超、陈书田、李宜儒

最初实验报告编写：李宜儒

最终报告修改和讨论：全体成员

**四、 作业总结**

**4.1 实验中的困难及不足**

在本次实验中，我们遇到的困难及大致解决方法如下：

1）环境的配置问题。在写代码跑程序之前，配置环境和安装Numpy、Scipy等包是一件麻烦的事。最开始我跟着教程在网上下载whl文件，到后面可能是因为版本问题不相互匹配而一直没有配置好环境。在组员的帮助下，利用anaconda配置完成环境并安装好程序所需要的包。

2）调用函数。在写代码的时候，虽然说调用函数是一个方便且不大会出错的方法，但是，如果我们还可以更好的话，可以自己编写算法函数代替所调用的函数。

3）保留主要的特征维度，能提高分类器的鲁棒性，但是选取维度过少，会丢失了一些特征信息。过度调参数可能会增加过拟合。

**4.2 实验心得体会**

通过本次模式识别实验，我们对PCA这种算法和结合python的强大功能有了更深入的了解。在实验最开始的时候，我们自认为对数字识别这一块还是有信心的，但是，在实际操作的时候，可能是因为环境配置的问题或者是对算法的理解不够深刻而丧失了一些些耐心，开始变得焦躁。经过几天的讨论和网上的查找资料，情况在一点一点变好，我们也做的越来越顺畅。

虽然题目是有些简单，但是大作业让我们直接接触到了用程序语言来让机器学习，解决一些生活中所看到的小问题。让我们在今后的机器学习研究中，积攒了许多经验。如果完成一项小任务，对我们在机器学习方面也会产生兴趣的。最后想感谢赵老师在这一个学期无微不至的照顾和教学，也很感激他一直在课上亲自示范敲代码的举措。