**SVM课程报告**

学号：20307110315

姓名：周训哲

# 1.任务描述

利用SVM支持向量机分类算法实现手写数字的识别与分类，分别尝试使用线性核、多项式核、sigmoid核、高斯核进行支持向量的多分类，并且尝试使用pca算法对多维数据进行降维分析以及可视化输出。解决了多样化模糊图片的识别分类问题。除此而外，还利用学习的SVM数学模型自己仿真实现了自己的SVM二分类模型（具体实现详见源代码）。

# 2.数据集描述

**采用何种数据集开展实验？**

利用MNIST手写数字集作为学习以及测试的数据集。里面包含了图片集作为识别对象，标签集作为学习判断的标准。

**数据集有什么特点？**

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。具有极大的测试数据量。

数据集分为训练集和测试集两个部分，其中又细分为图片集和标签集。

图片集主要存储的是图像信息，每一张图片包含28\*28个像素，在数据集中展开转换为一个长度为784的向量数据。在Mnist的训练数据集中mnist.train.images是一个形状为[60000, 784]的张量，第一个维度数字用来索引图片，第二个维度数字用来索引每张图片中的像素点，图片里的某个像素的强度值介于0-1之间。

标签集主要存储的是分类的信息，将每张图片所表示的数字进行存储。Mnist数据集中所有的标签mnist.train.labels是一个[60000, 10]的数字矩阵。在训练集中主要用来学习，以及kNN算法中的分类（利用距离）。在测试集中主要用来对比计算算法的识别准确率。

**数据集的各项统计量是多少？**

训练集中的数据量有60000个，测试集中数据量有10000个。

# 3.方法介绍

## 3.1数据预处理（如有）

**从原始数据到输入模型的数据之间，经过了哪些处理？这些处理有何作用？**

首先，原始数据为ubyte二进制文件，需要对二进制数据进行处理。需要利用open（）函数打开，其中参数为 ‘rb’，其含义为将文件以二进制文件形式打开。

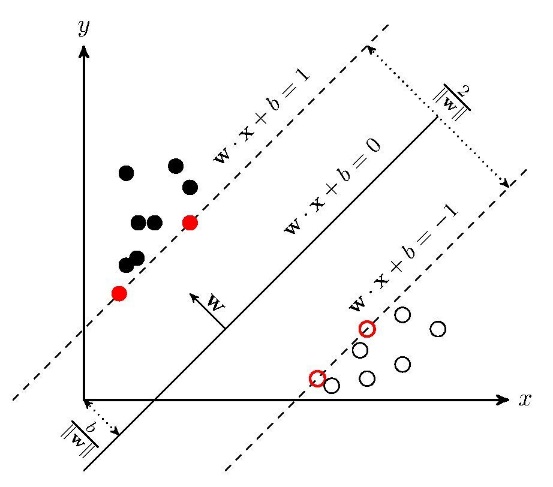
对于图像数据（image）是一个3维的数据集（60000\*28\*28）。开头为信息标识，有四个变量，魔数（magic）为2051，数据集大小（size）为60000，图像的行、列都是28，均通过unpack（）函数读取。然后将图像数据从28\*28的二维数据进行转换为长度为748的一维向量images。

对于标签数据（label）是一个一维的数据集（60000），开头也是信息标识，有两个变量，魔数（magic）为2049，数据集大小（size）为60000，也是通过unpack（）函数读取，使用uint8解码，将标签存储在向量labels中。

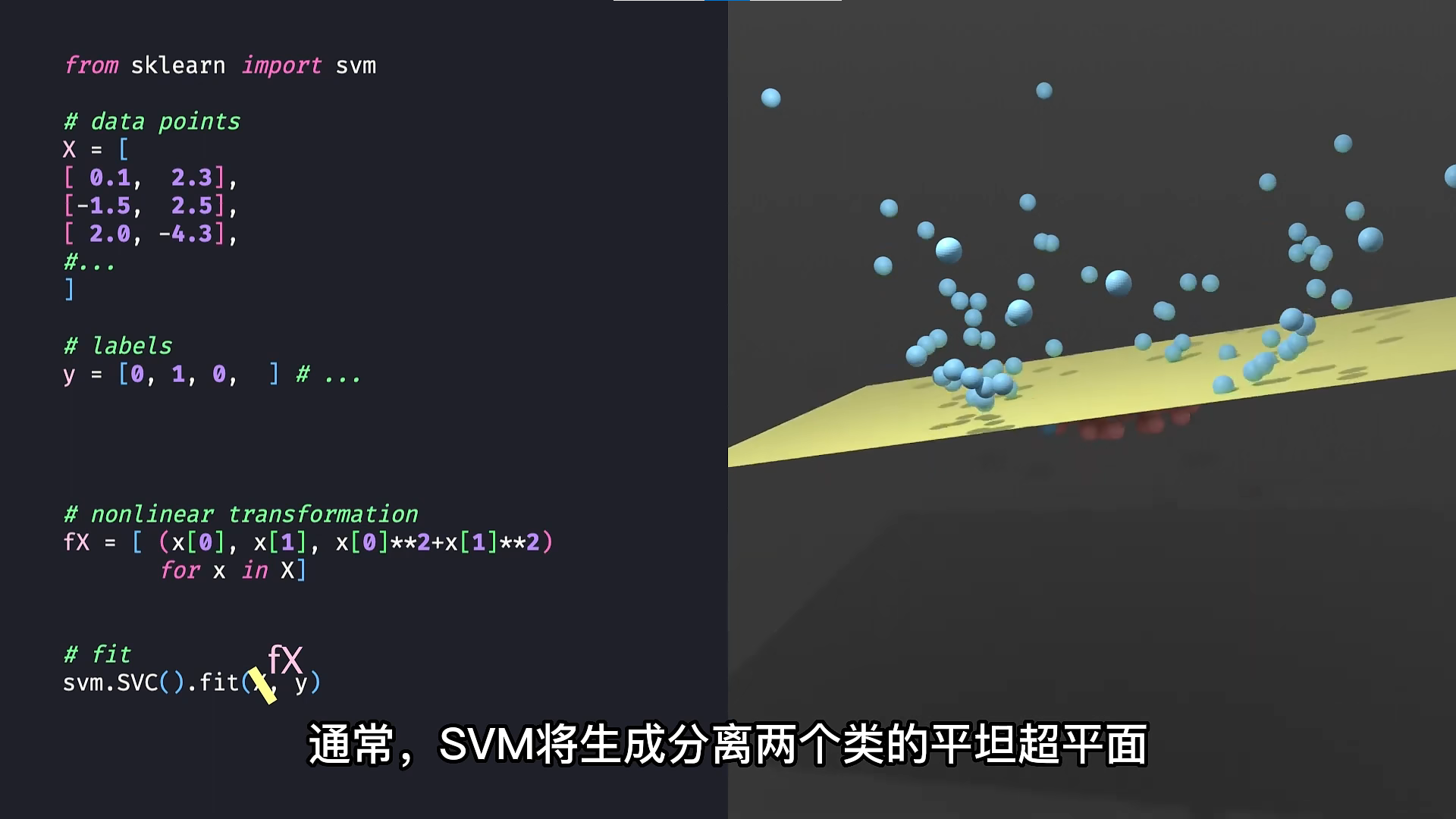
## 3.2算法描述

**核心思想：**

SVM算法的核心思想是，要对一堆数据进行分类，就要找到数据的分界点，也即决策边界。而决策边界的寻找则要取决于支持向量的选择。假设数据集线性可分，对于线性SVM来说，其决策边界可以假设为wx+b，则支持向量间的间隔投影，也即两类数据之间的间隔为2/||w||，为了分类的有效性，则需要找到最大的间隔，也就找到最小的||w||。利用已有的约束条件，y（wx+b）>=1，使用拉格朗日乘子法进行计算，由于原问题的解难以求解，则需要引入一个对偶问题，然后对对偶问题进行求解。最后得到KKT条件，将条件带入后使用SMO算法即可求得拉格朗日乘子法中引入的lamda，然后代入即可求得w和b，最后代入测试集进行分类测试，即可完成模型的生成。



对于线性不可分的情况，则使用非线性核进行升维处理，即利用核技巧将一部分数据升维，然后利用维度对数据进行分类。

****

对于多分类模型，目前，构造SVM多类分类器的方法主要有两类：一类是直接法，直接在目标函数上进行修改，将多个分类面的参数求解合并到一个最优化问题中，通过求解该最优化问题“一次性”实现多类分类。这种方法看似简单，但其计算复杂度比较高，实现起来比较困难，只适合用于小型问题中；另一类是间接法，主要是通过组合多个二分类器来实现多分类器的构造，常见的方法有one-against-one（一对一）和one-against-all（一对多）两种。（a.一对多法（one-versus-rest,简称1-v-r-SVMs）。训练时依次把某个类别的样本归为一类,其他剩余的样本归为另一类，这样k个类别的样本就构造出了k个SVM。分类时将未知样本分类为具有最大分类函数值的那类。b.一对一法（one-versus-one,简称1-v-1 SVMs）。其做法是在任意两类样本之间设计一个SVM，因此k个类别的样本就需要设计k(k-1)/2个SVM。当对一个未知样本进行分类时，最后得票最多的类别即为该未知样本的类别。Libsvm中的多类分类就是根据这个方法实现的。）

**整体流程：**

总体来说，SVM分类算法包括以下4个步骤：

①准备数据，对数据进行预处理。

②选择数据及参数，代入多分类模型。

③对模型进行回归，得到系数

④将系数代入测试集，即可测算准确率。

**缺点：**

（1）由于SVM在进行SMO计算时，需要对所有数据进行多次循环迭代以更新lamda，所以当数据集较大时，则可能耗费时间较长。

（2）由于SVM本身没有缺失值处理的策略，特征向量本身及特征空间的构建，对SVM的结果有影响，以及超平面的确定由少量支持向量决定，缺失数据在超平面附近会影响模型的结果，导致SVM对缺失数据敏感。

**改进策略：**

经过搜查，找寻到一种改进传统SVM的方法，仅供参考：（<https://www.zhihu.com/question/31211585>）

第一步：实现传统的 SMO 算法

第二步：实现核函数缓存

第三步：优化误差值求解

第四步：实现冷热数据分离

第五步：支持 Ensemble

第六步：继续优化核函数计算

第七步：支持稀疏向量和非稀疏向量

第八步：针对线性核进行优化

**训练和预测过程:（具体实现思路）**

训练过程即输入调试参数以及数据，代入模型即可（具体算法在之前已做概述），

svc = svm.SVC()

parameters = {'kernel': ['poly'], 'C': [1]}

clf = GridSearchCV(svc, parameters, n\_jobs=-1)

clf.fit(train\_data, train\_label)

测试过程，即将clf模型代入测试集数据，进行数据的比对，然后根据label进行类型的判别。

prediction = clf.predict(test\_data)

accurate = [0] \* 10  
sumall = [0] \* 10  
i = 0  
while i < len(test\_label):  
 sumall[test\_label[i]] += 1  
 if prediction[i] == test\_label[i]:  
 accurate[test\_label[i]] += 1  
 i += 1

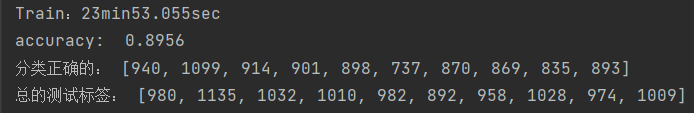
# 4.实验结果分析

## 4.1评价指标

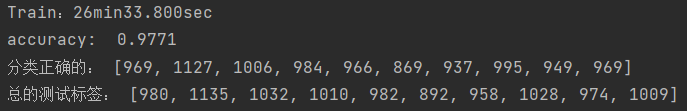
该算法训练集学习之后，对测试集样例的图像文件进行识别与预测，将预测结果与标签集比对，最后计算得到10000个测试数据的准确率。

## 4.2定量评价结果

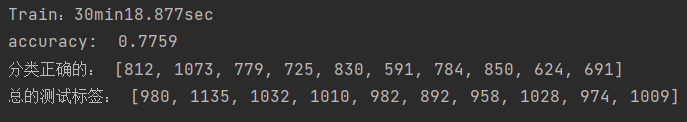
**基于线性核SVM的MNIST数据集分类准确率**

****

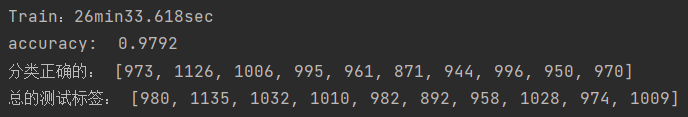
**基多项式核SVM的MNIST数据集分类准确率**



**基于sigmoid核SVM的MNIST数据集分类准确率**



**基于高斯核SVM的MNIST数据集分类准确率**

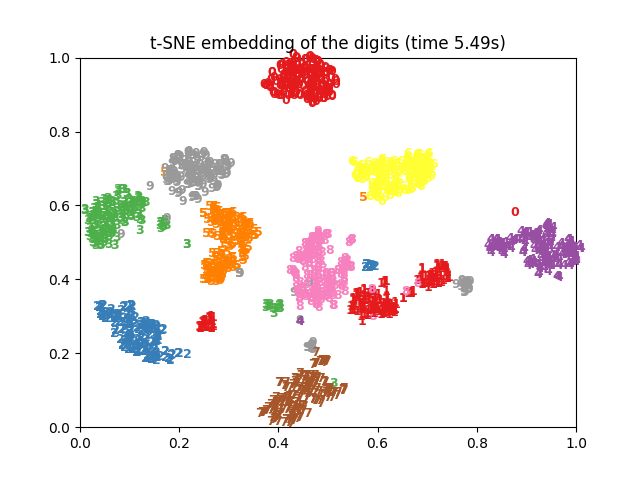


注：其中标签中数量分别对应的是0-9的数字图片集

可见，高斯核和多项式核的表现最好，sigmoid核的表现最差，线性核的表现居中。

## 4.3可视化结果

利用PCA降维分析方法，将28\*28=784维的数据成功降维到2维，并在图标中实现可视化输出。



# 总结

对于MNIST数据集来说，非线性核中的高斯核表现是明显优于线性核，说明MNIST数据集并非完全线性可分。并且使用sigmoi函数的效果也不是很好，说明其分类函数的选择十分重要。

通过本次实验，我学会了SVM的基本原理以及其套用到具体实例的python代码实现。除此而外，通过本次学习，我了解到PCA降维方法以及一些数据处理的技巧，可以说是收获颇丰。