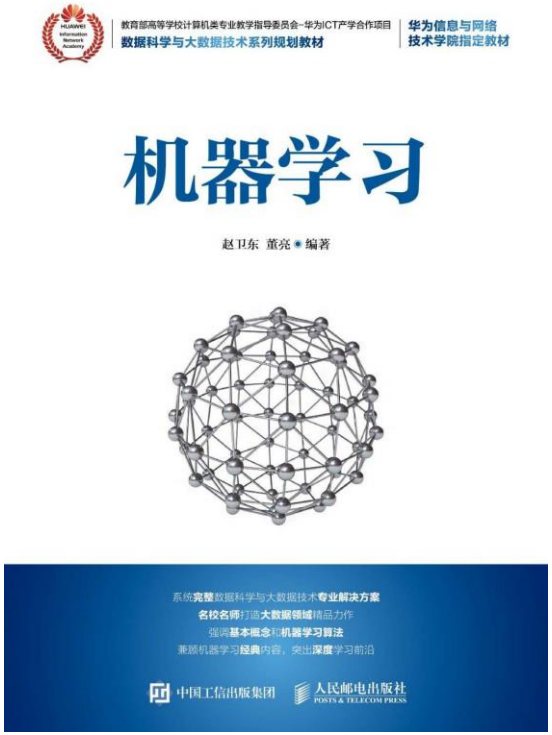


机器学习
支持向量机

复旦大学 **赵卫东** 博士

wdzhao@fudan.edu.cn

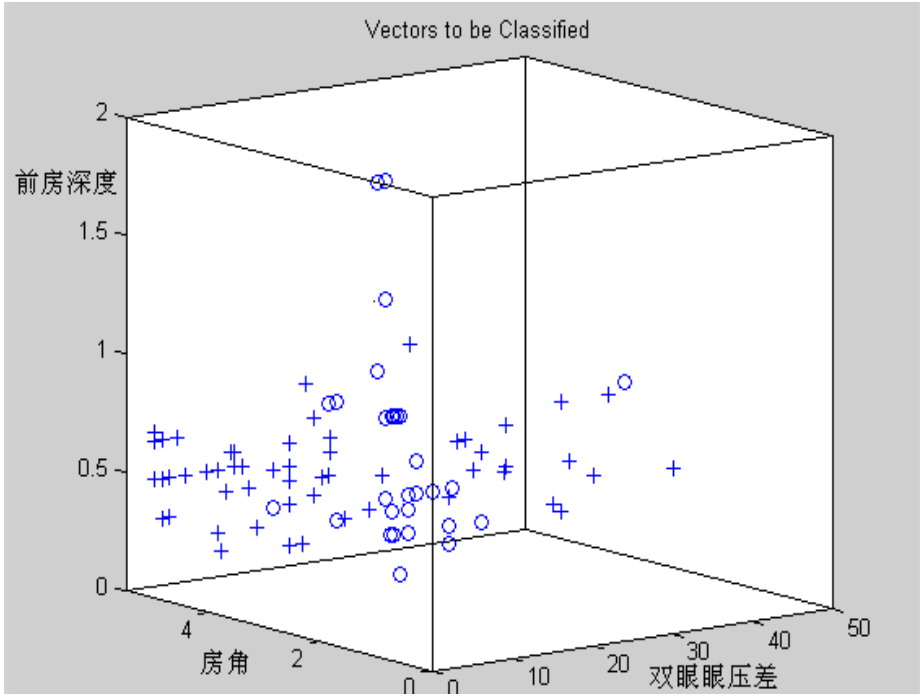
章节介绍

- 支持向量机（Support Vector Machine，SVM）属于有监督学习模型，主要用于解决数据分类问题。通常SVM用于二元分类问题，对于多元分类可将其分解为多个二元分类问题，再进行分类，主要应用场景有图像分类、文本分类、面部识别和垃圾邮件检测等领域。
- 本章共划分为两个小节，分别介绍支持向量机模型的基础以及支持向量机的应用过程。

章节结构

- 支持向量机模型
  - 核函数
  - 模型原理分析
- 支持向量机应用
  - 基于SVM进行新闻主题分类
  - 基于支持向量机和主成分分析的人脸识别

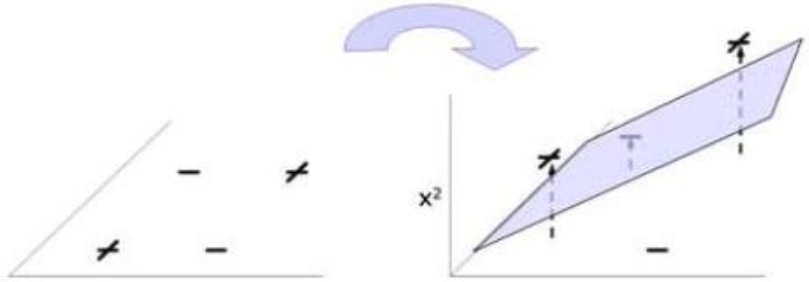
一个例子：青光眼诊断



图中“+”表示开角型青光眼样本点，“○”表示闭角型青光眼型样本点。样本数据相互交叉较多，不易进行线性可分。

支持向量机模型

- 支持向量机在高维或无限维空间中构造超平面或超平面集合，将原有限维空间映射到维数高得多的空间中，在该空间中进行分离可能会更容易。它可以同时最小化经验误差和最大化集合边缘区，因此它也被称为最大间隔分类器。直观来说，分类边界距离最近的训练数据点越远越好，因为这样可以缩小分类器的泛化误差。

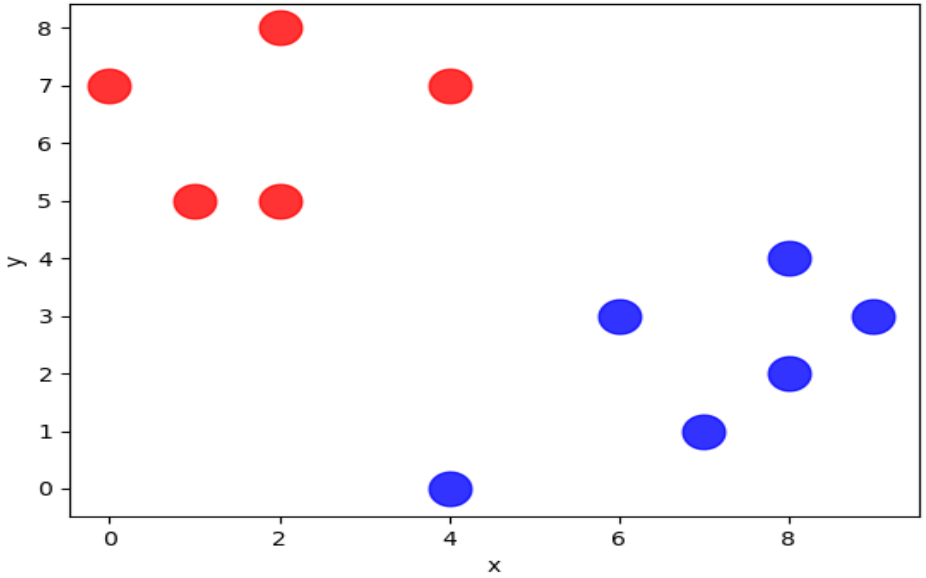


二维直线不可分离　　用三维平面可分离

低维不可分问题高维未必不可分

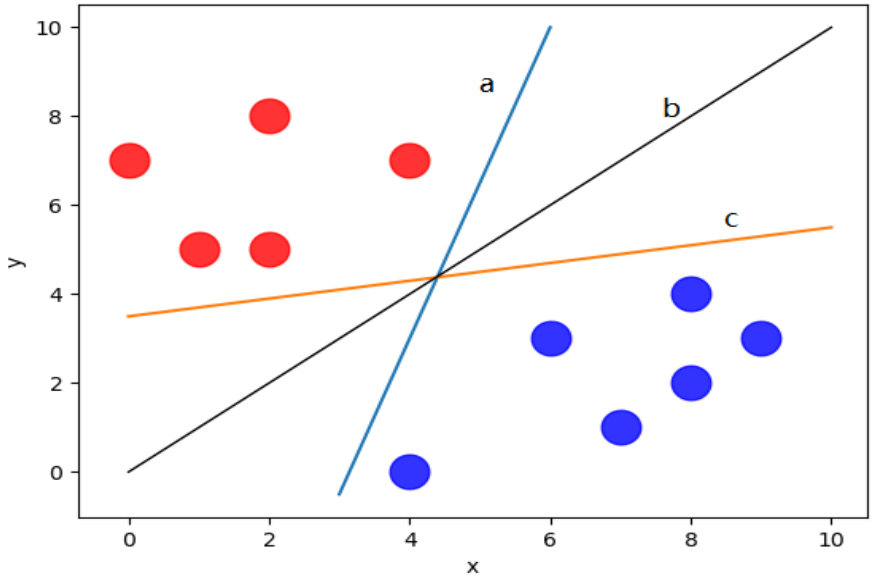
模型基本思想

- 以一个二元分类问题为例讲解模型原理。首先假设有两类数据，如图需要找出一条边界来将两类数据分隔开来。



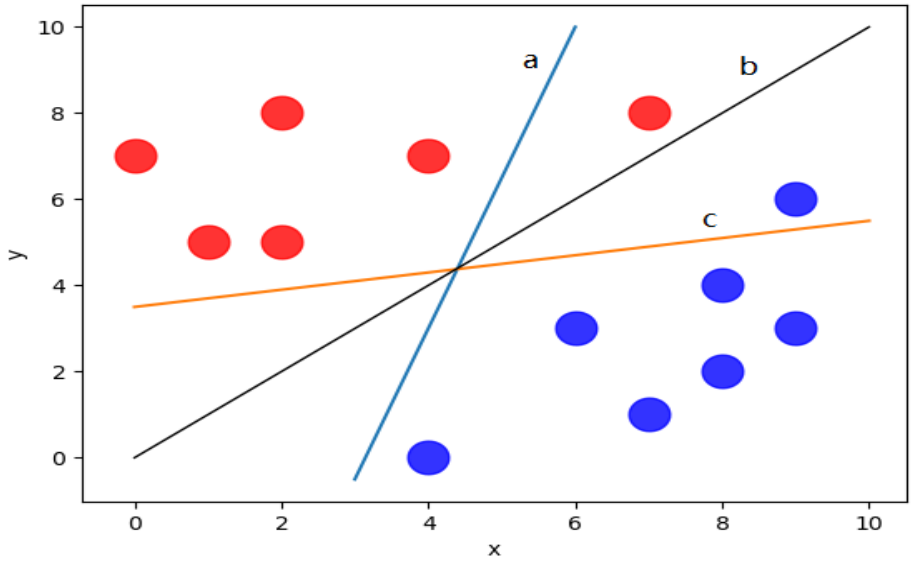
模型基本思想

- 下图中列出一些可行的分隔方式。在当前的数据集的条件下，三种分隔方式都是可行的，我们该如何做选择？



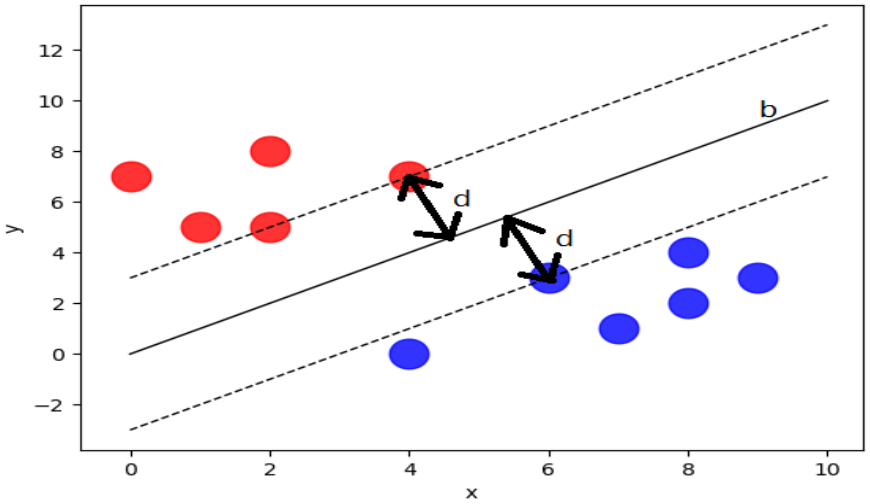
模型基本思想

- 一般说来，需要选择的是具有较强分类能力的直线，有较稳定的分类结果和较强的抗噪能力，比如在数据集扩展之后如下图所示。在这三种分隔方式中，b的分隔效果更好。



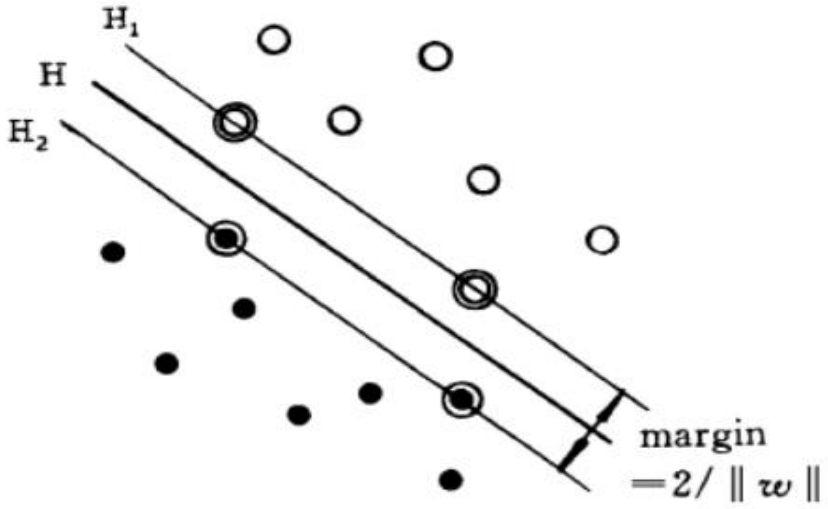
模型基本思想

- 找到最优分类数据的分界线，使得对样本数据的分类效果更好的方法就是要尽可能地远离两类数据点，即数据集的边缘点到分界线的距离d最大，这里虚线穿过的边缘点称作支持向量，分类间隔为2d。如下图所示。



支持向量机原理

SVM是从线性可分情况下的最优分类面发展而来的。



- 分类超平面：(w.x)+b=0
- 判决函数：

$$y_i = \mathrm{sgn}(wx_i + b) \quad y_i \in \{-1, 1\}$$

- 最大间隔问题：

在间隔固定为1时，寻求最小的‖w ‖

支持向量机原理

- 容易看出，最优化目标就是最大化几何间隔，并且注意到几何间隔与‖w‖反比，因此只需寻找最小的‖w‖，即

$$\min \|w\|$$

- 对于这个目标函数，可以用一个等价的目标函数来替代：

$$\min \frac{1}{2}\|w\|^2$$

支持向量机原理

- 为使分类对所有样本正确分类，要求满足如下约束：

$$y_i[(\boldsymbol{w}\cdot\boldsymbol{x}_i)+b]\geqslant 1 \qquad i=1,2,\cdots,l$$

支持向量机原理

- 优化问题：$\min \ \frac{1}{2}\|w\|^2$

$$s.t. \quad y_i[(wx_i)+b]-1\geq 0 \quad (i=1,2,...,n)$$

- 为解决这个约束问题的最优解，引入Lagrange函数：

$$L(\boldsymbol{\omega}, b, \alpha) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 - \sum_{i=1}^{n}\alpha_i[y_i(\boldsymbol{\omega}\cdot\boldsymbol{x}_i+b)-1]$$

式中α_i>=0为Lagrange乘子。为求函数的最小值，分别对w、b、 α_i求偏微：

支持向量机原理

- 分别对w、b、α求偏微：

$$\begin{cases} \dfrac{\partial L}{\partial \boldsymbol{\omega}} = 0 \Rightarrow \boldsymbol{\omega} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i \\ \dfrac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \dfrac{\partial L}{\partial \alpha_i} = 0 \Rightarrow \alpha_i [y_i (\boldsymbol{\omega} \cdot \boldsymbol{x}_i + b) - 1] = 0 \end{cases}$$

可以将上述求最优平面的问题转化为对偶问题：

支持向量机原理

$$\begin{cases} \max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j) \\ \text{s. t.} \quad \alpha_i \geqslant 0, \quad i = 1, \cdots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

支持向量机原理

- 这是一个二次函数寻优的问题，存在唯一的解。若a*为最优解：

$$\boldsymbol{\omega}^* = \sum_{i=1}^{n} \alpha_i^* y_i \boldsymbol{x}_i$$

式中 α_i^* 为不为零的样本，即支持向量。b^* 是分类阈值，可由约束条件：$\alpha_i[y_i(\boldsymbol{\omega} \cdot \boldsymbol{x}_i + b) - 1] = 0$

得到最优分类函数为

$$f(\boldsymbol{x}) = \mathrm{sng}\{(\boldsymbol{\omega} \cdot \boldsymbol{x}) + b\} = \mathrm{sgn}\left\{\sum_{i=1}^{n} \alpha_i^* y_i (\boldsymbol{x}_i \cdot \boldsymbol{x}) + b^*\right\}$$

核函数

- 支持向量机通过线性变换A(x)将输入空间X映射到高维特征空间Y，如果低维空间存在函数K，x,y∈X，使得K(x,y)=A(x)·A(y)，则称K(x,y)为核函数。核函数方法可以与不同的算法相结合，形成多种不同的基于核函数的方法，常用的核函数有：
  - 线性核函数
  - 多项式核函数
  - 径向基核函数
  - Sigmoid核

线性核函数

- 线性核函数(Linear Kernel)是最简单的核函数，主要用于线性可分的情况，表达式如下：

$$K(x,y) = x \cdot y + c$$

其中c是可选的常数。线性核函数是原始输入空间的内积，即特征空间和输入空间的维度是一样的，参数较少运算速度较快。适用的情景是在特征数量相对于样本数量非常多时。

多项式核函数

- 多项式核函数(Polynomial Kernel)是一种非稳态核函数，适合于正交归一化后的数据，表达式如下：

$$K(x,y) = [a \cdot x \cdot y + c]^{d}$$

其中a是调节参数，d是最高次项次数，c是可选的常数。

径向基核函数

- 径向基核函数(Radial Basis Function Kernel)具有很强的灵活性，应用广泛。与多项式核函数相比参数较少。因此大多数情况下都有较好的性能。径向基核函数类似于高斯函数，所以也被称为高斯核函数。在不确定用哪种核函数时，可优先验证高斯核函数。表达式如下：

$$K(x,y) = \exp\{-[(||x-y||^2)/(2\cdot a^2)]\}$$

其中a^2越大。高斯核函数就会变得越平滑，此时函数随输入x变化较缓慢，模型的偏差和方差大，泛化能力差，容易过拟合。 a^2越小，高斯核函数变化越剧烈，模型的偏差和方差越小，模型对噪声样本比较敏感。

Sigmoid核

- Sigmoid核(Sigmoid Kernel)来源于MLP中的激活函数，SVM使用Sigmoid相当于一个两层的感知机网络，表达式如下：

$$K(x,y) = \tanh(a \cdot x \cdot y + c)$$

其中a表示调节参数，c为可选常数，一般情况c取1/n，n是数据维度。

支持向量机应用

- 支持向量机（SVM）算法比较适合图像和文本等样本特征较多的应用场合。基于结构风险最小化原理，对样本集进行压缩，解决了以往需要大样本数量进行训练的问题。它将文本通过计算抽象成向量化的训练数据，提高了分类的精确率。

新闻主题分类

- 新闻的分类是根据新闻中与主题相关的词汇来完成的。应用SVM对新闻分类可以划分为五个步骤：
  - 获取数据集
  - 将文本转化为可处理的向量
  - 分割数据集
  - 支持向量机分类
  - 分类结果显示

获取数据集

- 数据集来自于sklearn官网上的20组新闻数据集，下载地址为：http://scikit-learn.org/stable/datasets/index.html#the-20-newsgroups-text-dataset
  数据集中一共包含20类新闻，选择其中三类新闻，对应的target依次为0,1,2。部分代码如下：

```
select = ['alt.atheism', 'talk.religion.misc', 'comp.graphics']
newsgroups_train_se = fetch_20newsgroups(subset='train', categories=select)
```

文本转化为向量

- sklearn中封装了向量化工具TfidfVectorizer，它统计每则新闻中各个单词出现的频率，并进行TF-IDF处理，其中TF（term frequency）是某一个给定的词语在该文件中出现的次数。IDF（inverse document frequency）是逆文档频率，用于降低其它文档中普遍出现的词语的重要性，TF-IDF倾向于过滤掉常见的词语，保留重要的词语。通过TF-IDF来实现文本特征的选择，也就是说，一个词语在当前文章中出现次数较多，但在其它文章中较少出现，那么可认为这个词语能够代表此文章，具有较高的类别区分能力。使用TfidfVectorizer实例化、建立索引和编码文档的过程如下:

```
vectorizer = TfidfVectorizer()
vectors = vectorizer.fit_transform(newsgroups_train_se.data)
print(vectors.shape)
```

支持向量机分类

- 使用sklearn中的SVM工具包SVC（C-Support Vector Classification）来进行分类，核函数采用的是线性核函数，代码如下：

```
svc = SVC(kernel='linear')
svc.fit(x_train, y_train)
```

分类结果显示

- print(svc.score(x_test, y_test))

  Result: 0.955017301038

  可以看到训练正确率约为95.5%

基于支持向量机和主成分分析的人脸识别

- 主成分分析（Principal Component Analysis , PCA）是一种降维方法，可以从多种特征中解析出主要的影响因素，使用较少的特征数量表示整体。PCA的目标就是找到方差大的维度作为特征。本案例可以被划分为六个步骤：

  - 获取数据集
  - 将图片转化为可处理的n维向量
  - 分割数据集
  - PCA主成分分析，降维处理
  - 支持向量机分类
  - 查看训练后的分类结果

主成分分析

- 主成分分析是最常用的线性降维方法，它的目标是通过某种线性投影，将高维的数据映射到低维的空间中，并期望在所投影的维度上数据的方差最大，以此使用较少的维度，同时保留较多原数据的维度
- 尽可能如果把所有的点都映射到一起，那么几乎所有的区分信息都丢失了，而如果映射后方差尽可能的大，那么数据点则会分散开来，特征更加明显。PCA是丢失原始数据信息最少的一种线性降维方法，最接近原始数据
- PCA算法目标是求出样本数据的协方差矩阵的特征值和特征向量，而协方差矩阵的特征向量的方向就是PCA需要投影的方向。使样本数据向低维投影后，能尽可能表征原始的数据。协方差矩阵可以用散布矩阵代替，协方差矩阵乘以（n-1）就是散布矩阵，n为样本的数量。协方差矩阵和散布矩阵都是对称矩阵，主对角线是各个随机变量（各个维度）的方差

主成分分析

- 设有m条n维数据，PCA的一般步骤如下
  - 将原始数据按列组成n行m列矩阵X
  - 计算矩阵X中每个特征属性（n维）的平均向量M（平均值）
  - 将X的每行（代表一个属性字段）进行零均值化，即减去M
  - 按照公式$C = \frac{1}{\mathrm{m}} XX^T$求出协方差矩阵
  - 求出协方差矩阵的特征值及对应的特征向量
  - 将特征向量按对应特征值从大到小按行排列成矩阵，取前k（k < n）行组成基向量P
  - 通过$Y = PX$计算降维到k维后的样本特征

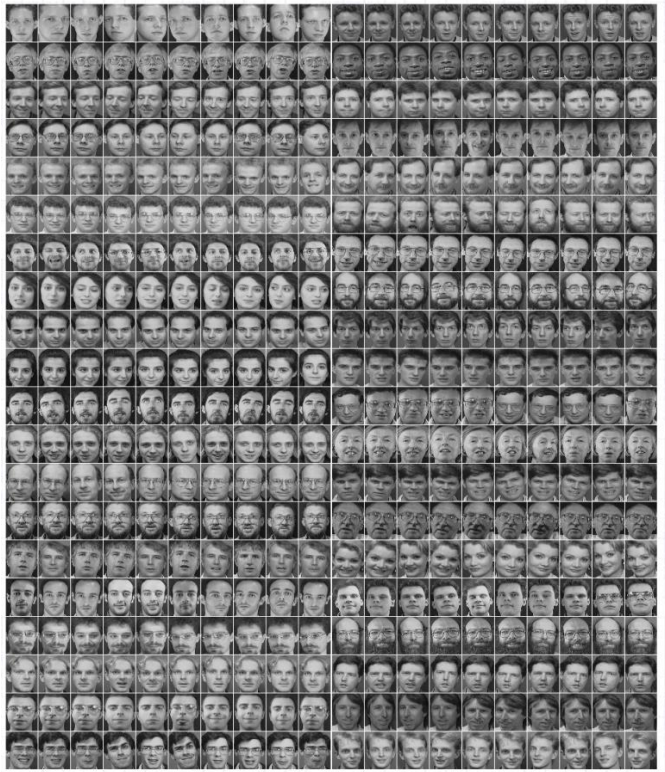
主成分分析

- 基于sklearn（Python语言下的机器学习库）和numpy随机生成2个类别共40个3维空间的样本点，生成的代码如下：

```
mu_vec1 = np.array([0,0,0])
cov_mat1 = np.array([[1,0,0],[0,1,0],[0,0,1]])
class1_sample = np.random.multivariate_normal(mu_vec1, cov_mat1, 20).T
mu_vec2 = np.array([1,1,1])
cov_mat2 = np.array([[1,0,0],[0,1,0],[0,0,1]])
class2_sample = np.random.multivariate_normal(mu_vec2, cov_mat2, 20).T
```

获取数据集

- 数据集来自于英国剑桥大学的AT&T人脸数据集，此数据集共有40×10=400张图片，图片大小为112x92，已经经过灰度处理。一共被划分为40个类，每类中包含的是同一个人的10张图像。



图片转化为向量

- 由于每张图片的大小为112x92,每张图片共有10304个像素点，这时需要一个图片转化函数ImageConvert()，将每张图片转化为一个10304维向量，代码如下：

```
def ImageConvert():
 for i in range(1, 41):
  for j in range(1, 11):
    path = picture_savePath + "s" + str(i) + "/" + str(j) + ".pgm"
    # 单通道读取图片
    img = cv2.imread(path, cv2.IMREAD_GRAYSCALE)
    h, w = img.shape
    img_col = img.reshape(h * w)
    data.append(img_col)
    label.append(i)
```

图片转化为向量

- data变量中存储了每个图片的10304维信息,格式为列表变量（list）。变量label中存储了每个图片的类别标签，为数字1~40。应用numpy生成特征向量矩阵，代码如下：

```
import numpy as np
C_data = np.array(data)
C_label = np.array(label)
```

分割数据集

- 将训练集与测试集按照4:1的比例进行随机分割，即测试集占20%，代码如下：

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(C_data, C_label, test_size=0.2,
random_state=256)
```

PCA主成分分析，降维处理

- 引入sklearn工具进行PCA处理：

  from sklearn.decomposition import PCA

  pca = PCA(n_components=15, svd_solver='auto').fit(x_train)

  方法中的15表示处理后保留维度为15个，auto表示PCA会自动选择合适的SVD算法，进行维度转化：

  x_train_pca = pca.transform(x_train)

  x_test_pca = pca.transform(x_test)

支持向量机分类

- 使用sklearn中的SVM工具包SVC（C-Support Vector Classification）来进行分类，核函数采用的是线性核函数，代码如下：

```
svc = SVC(kernel='linear')
svc.fit(x_train, y_train)
```

查看训练后的分类结果

- 使用测试集评估分类器的效果，代码如下：

  print('%.5f' % svc.score(x_test_pca, y_test))

  得到的输出正确率结果如下图所示：

```
/usr/bin/python2.7 /home/gengjia/PycharmProjects/svm_news/svm_pca_face.py
0.96250

Process finished with exit code 0
```

查看训练后的分类结果

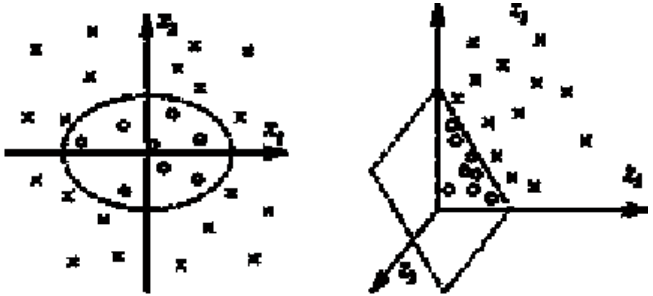
- 进行对比实验，将保留维度为10和20时的效果依次如下面两张图所示：

```
/usr/bin/python2.7 /home/gengjia/PycharmProjects/svm_news/svm_pca_face.py
0.95000

Process finished with exit code 0
```

```
/usr/bin/python2.7 /home/gengjia/PycharmProjects/svm_news/svm_pca_face.py
0.95000

Process finished with exit code 0
```

从图中显示的正确率的情况对比来看，特征数量降为15时，训练的结果是最好的。

线性不可分的情况



二维平面中分类曲线为椭圆（线性不可分）

$$w_1x_1^2 + w_2x_2^2 + \sqrt{2}w_3x_1x_2 + b = 0$$

线性不可分的情况

两维向三维的映射：

$$\Phi : (x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

在三维空间中线性可分

分类面： $w_1'z_1 + w_2'z_2 + w_3'z_3 + b = 0$

根据支持向量机求得决策函数为

$$f(z) = \mathrm{sgn}\{\sum_{i=1}^{l} y_i \alpha_i^* [\phi(z_i) \cdot \phi(z)] + b^*\}$$


THANK
YOU