深 圳 大 学 实 验 报 告

课程名称 机器学习

项目名称 实验四：SVM 算法

学　　院 计算机与软件学院

专　　业 软件工程（腾班）

指导教师 赖志辉

报 告 人 黄亮铭 学号 2022155028

实验时间 2024 年 4 月 8 日至 2024 年 4 月 21 日

实验报告提交时间 2024 年 4 月 21 日

教务处制

一、实验目的与要求

1. 熟悉 SVM 算法，要求理解 SVM 算法的基本原理。
2. 实现简单的 SVM 算法。
3. 使用实际数据集测试所写的 SVM 算法。
4. 分析 SVM 算法的分类效果。

二、实验内容与方法

1. 分别给出经典的/ 软间隔/核-SVM 的优化问题并推导其求解优化过程，实现经典的 SVM 算法进行图像识别；在二维平面对二类问题给出 support vector 的一个示例。
2. 用 PCA、LDA 算法提取前 10,20,30,...,160 维的图像特征，然后再用不同 SVM 进行识别，并比较识别率。

三、实验步骤与过程

1. 分别给出经典的/ 软间隔/核-SVM 的优化问题并推导其求解优化过程，实现经典的 SVM 算法进行图像识别；在二维平面对二类问题给出 support vector 的一个示例。

   a) **SVM 算法简介**

   支持向量机是一种经典的机器学习算法，常用于分类和回归分析。它在许多实际问题中都表现出色，特别是在高维空间中的数据分类任务上。

   - **基本原理：**SVM 的核心思想是找到一个最优的超平面，将不同类别的数据点分开。对于二分类问题，这个超平面可以被看作是一个将特征空间划分为两个部分的决策边界。最优的超平面是指能够最大化类别间间隔（margin）的超平面，即使得不同类别的数据点离超平面的距离最大化。
   - **关键概念：**

     ① 支持向量（Support Vectors）： 离超平面最近的数据点被称为支持向量。这些支持向量决定了超平面的位置和方向。

     ② 间隔（Margin）： 类别间间隔是指在超平面和支持向量之间的距离，SVM 试图最大化这个间隔，以提高分类器的泛化能力。

     ③ 核技巧（Kernel Trick）： SVM 可以通过引入核函数将数据映射到高维空间中，从而使得在原始特征空间中线性不可分的数据在新的空间中变得线性可分。
   - **优点：**1）SVM 对于高维空间中的数据分类效果好，适用于特征维度较高的情况。

     2）通过核技巧，SVM 能够处理非线性可分的数据。

     3）SVM 的训练过程中通常只涉及到一部分训练样本，因此对内存消耗较小。
   - **小结：**

   支持向量机是一种强大且灵活的机器学习算法，适用于各种分类和回归任务。它的优雅数学基础以及在高维空间中的优良性能使得它成为了许多实际问题中的首选算法之一。

b) **经典 SVM 优化问题及推导优化过程**

样本点(x_i, x_j)与超平面的距离为：$r = \frac{|w^T x_i + b|}{\|w\|}$

若分类正确，则有：$y_i(w^T x_i + b) \geq 1$

两异类支持向量到超平面的距离之和为：$\frac{2}{\|w\|}$

我们的目标是找到最大间隔以分类，即：

$$\max_{w,b} \frac{2}{\|w\|}, s.t. y_i(w^T x_i + b) \geq 1, \forall i = 1,2,\ldots,n$$

问题等价于如下形式：$\min_{w,b} \frac{1}{2}\|w\|^2, s.t. y_i(w^T x_i + b) \geq 1, \forall i = 1,2,\ldots,n$

令$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i[y_i(w^T x_i + b) - 1], s.t. y_i(w^T x_i + b) \geq 1, \forall i = 1,2,\ldots,n$

对 L 求偏导：$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i$　　$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_1 y_i$

令$\frac{\partial L}{\partial w} = 0$，$\frac{\partial L}{\partial b} = 0$得：$w = \sum_{i=1}^{n} \alpha_i y_i x_i$　$\sum_{i=1}^{n} \alpha_1 y_i = 0$

将结果回代 L:

$$\begin{aligned} L(w,b,\alpha) &= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i[y_i(w^T x_i + b) - 1] \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_1\alpha_j y_1 y_j x_i^T x_j - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_1\alpha_j y_1 y_j x_i^T x_j + \sum_{i=1}^{n} \alpha_i \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_1\alpha_j y_1 y_j x_i^T x_j \end{aligned}$$

原问题可以转化为：

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_1\alpha_j y_1 y_j x_i^T x_j, s.t. \sum_{i=1}^{n} \alpha_1 y_i = 0, \alpha_i \geq 0, \forall i = 1,2,\ldots,n$$

c) **软间隔 SVM 优化问题及推导优化过程**

软间隔即允许某些样本不满足要求，为此我们引入松弛变量ζ_i

$$\forall i = 1,2,\ldots,n, 有 \zeta_i \geq 0$$

目标函数为：$\min_{w,b,\zeta_i} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \zeta_i, s.t. y_i(w^T x_i + b) \geq 1, \zeta_i \geq 0, \forall i = 1,2,\ldots,n$

令$L(w,b,\zeta_i,\alpha,\mu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \zeta_i - \sum_{i=1}^{n} a_i[y_i(w^T x_i + b) - 1] -$

$\sum_{i=1}^{n} \mu_i\zeta_i, s.t. y_i(w^T x_i + b) \geq 1, \zeta_i \geq 0, \forall i = 1,2,\ldots,n$

对 L 求偏导：$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i$　$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_1 y_i$　$\frac{\partial L}{\partial \zeta_i} = C - a_i - \mu_i$

令$\frac{\partial L}{\partial w} = 0 \frac{\partial L}{\partial b} = 0$　$\frac{\partial L}{\partial \zeta_i} = 0$，得：$w = \sum_{i=1}^{n} \alpha_i y_i x_i$　$\sum_{i=1}^{n} \alpha_1 y_i = 0$　　$C = a_i + \mu_i$

将结果回代 L:

$$L(w,b,\xi_i,\alpha,\mu)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\zeta_i-\sum_{i=1}^{n}a_i[y_i(w^Tx_i+b)-1]-\sum_{i=1}^{n}\mu_i\zeta_i$$

$$=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_1\alpha_j y_1 y_j x_i^T x_j+\sum_{i=1}^{n}(C-a_i-\mu_i)\zeta_i$$

$$-\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_1\alpha_j y_1 y_j x_i^T x_j+\sum_{i=1}^{n}\alpha_i$$

$$=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_1\alpha_j y_1 y_j x_i^T x_j$$

原问题可以转化为:

$$\max_{\alpha}\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_1\alpha_j y_1 y_j x_i^T x_j\ ,s.t.\sum_{i=1}^{n}\alpha_1 y_i=0, C\geq\alpha_i\geq 0,\forall i=1,2,\dots,n$$

d) 核-SVM 优化问题及推导优化过程

目标函数为: $\min_{w,b}\frac{1}{2}\|w\|^2, s.t. y_i(w^T\phi(x_i)+b)\geq 1,\forall i=1,2,\dots,n$

令$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}a_i[y_i(w^T\phi(x_i)+b)-1], s.t. y_i(w^T\phi(x_i)_i+b)\geq 1,\forall i=1,2,\dots,n$

对 L 求偏导: $\frac{\partial L}{\partial w}=w-\sum_{i=1}^{n}\alpha_i y_i\phi(x_i)$ $\frac{\partial L}{\partial b}=-\sum_{i=1}^{n}\alpha_1 y_i$

令$\frac{\partial L}{\partial w}=0$, $\frac{\partial L}{\partial b}=0$得: $w=\sum_{i=1}^{n}\alpha_i y_i\phi(x_i)$ $\sum_{i=1}^{n}\alpha_1 y_i=0$

将结果回代 L:

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{n}a_i[y_i(w^T\phi(x_i)+b)-1]$$

$$=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_1\alpha_j y_1 y_j\phi(x_i)^T\phi(x_j)-\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_1\alpha_j y_1 y_j\phi(x_i)^T\phi(x_j)$$

$$+\sum_{i=1}^{n}\alpha_i$$

$$=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_1\alpha_j y_1 y_j\phi(x_i)^T\phi(x_j)$$

由核方法可知: $\kappa(x_i,x_j)=\phi(x_i)^T\phi(x_j)$

原问题可以转化为:

$$\max_{\alpha}\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_1\alpha_j y_1 y_j\kappa(x_i,x_j)\ ,s.t.\sum_{i=1}^{n}\alpha_1 y_i=0,\alpha_i\geq 0,\forall i=1,2,\dots,n$$

e) 经典 SVM 算法的实现

```
% 划分训练集和测试集
% 训练集
train_set = [dataset(1:5,:);dataset(11:15,:)];
train_set_labels = [lableset(1:5);lableset(11:15)];
% 测试集
test_set = [dataset(6:10,:);dataset(16:20,:)];
test_set_labels = [lableset(6:10);lableset(16:20)];

% 数据预处理（归一化）
[mtrain,ntrain] = size(train_set);
[mtest,ntest] = size(test_set);
test_dataset = [train_set;test_set];
[dataset_scale,ps] = mapminmax(test_dataset',0,1);
dataset_scale = dataset_scale';
train_set = dataset_scale(1:mtrain,:);
test_set = dataset_scale( (mtrain+1):(mtrain+mtest),: );

% SVM训练
model = fitcsvm(train_set,train_set_labels);

% SVM预测
[predict_label] = predict(model,test_set);
```

图 1：经典 SVM

f) 在二维平面对二类问题给出 support vector 的一个示例

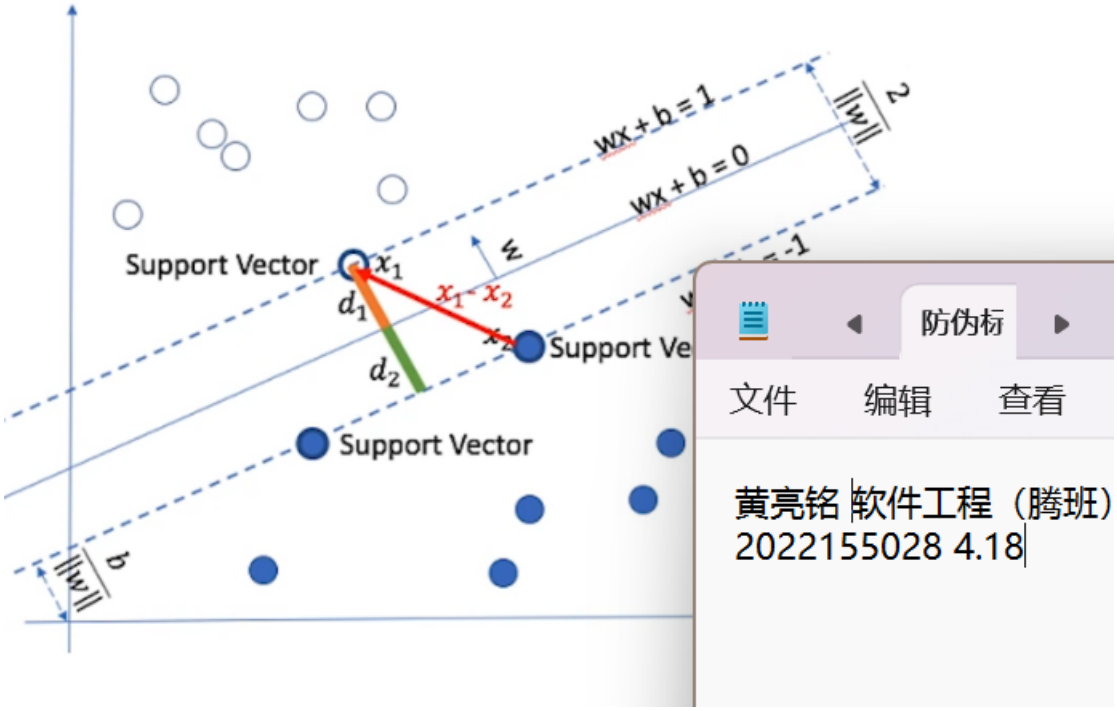


图 2：支持向量示例

2. 用 PCA、LDA 算法提取前 10,20,30,...,160 维的图像特征，然后再用不同 SVM 进行识别，并比较识别率。

a) SVM 核心代码

这里使用经典 SVM 进行人脸识别。对于 c 个类别的情况，我训练了$\frac{c(c-1)}{2}$个 SVM，每个 SVM 用于判断任意个样本是属于 c 种类别中的特定的两种类别。测试时，使用$\frac{c(c-1)}{2}$个 SVM 做$\frac{c(c-1)}{2}$次测试，遵循多数服从少数的原则，样本被分到哪个类别的次数最多，则认为样本属于哪个类别。实现代码如下图：

```
%使用SVM人脸识别
for i=10:10:160
    right_num = 0;
    % 降维得到投影矩阵
    project_matrix = eigen_vectors(:,1:i);
    projected_train_data = project_matrix' * (train_data - all_mean);
    projected_test_data = project_matrix' * (test_data - all_mean);
    % SVM训练过程
    model_num = 1;
    for j = 0:1:people_num - 2
        % 取出每次SVM需要的训练集
        train_img1 = projected_train_data(:,j * each_train_pic_num + 1 : j * each_train_pic_num + each_train_pic_num);
        train_label1 = ones(1,each_train_pic_num)*(j + 1);
        % 取出每次SVM需要的测试集
        test_img1 = projected_test_data(:,j * each_test_pic_num + 1 : j * each_test_pic_num + each_test_pic_num);
        for z = j + 1:1:people_num - 1
            % 取出每次SVM需要的训练集
            train_img2 = projected_train_data(:,z * each_train_pic_num + 1 : z * each_train_pic_num + each_train_pic_num);
            train_label2 = ones(1,each_train_pic_num)*(z + 1);
            train_imgs = [train_img1,train_img2];
            train_label = [train_label1,train_label2];
            % 取出每次SVM需要的测试集
            test_img2 = projected_test_data(:,z * each_test_pic_num + 1 : z * each_test_pic_num + each_test_pic_num);
            test_imgs = [test_img1,test_img2];
            % 数据预处理（归一化）
            [mtrain,ntrain] = size(train_imgs);
            [mtest,ntest] = size(test_imgs);
            test_dataset = [train_imgs,test_imgs];
            [dataset_scale,ps] = mapminmax(test_dataset,0,1);
            train_imgs = dataset_scale(:,1:ntrain);
            test_imgs = dataset_scale( :,(ntrain+1):(ntrain+ntest) );
            % SVM训练
            train_imgs = train_imgs';
            train_label = train_label';
            % fitcsvm默认读取数据为按行，一张一脸为一列，需要转置
            expr = ['model_' num2str(model_num) ' = fitcsvm(train_imgs,train_label);'];
            eval(expr);
            model_num = model_num + 1;
        end
    end
    model_num = model_num - 1;
```


图 3：SVM 核心代码

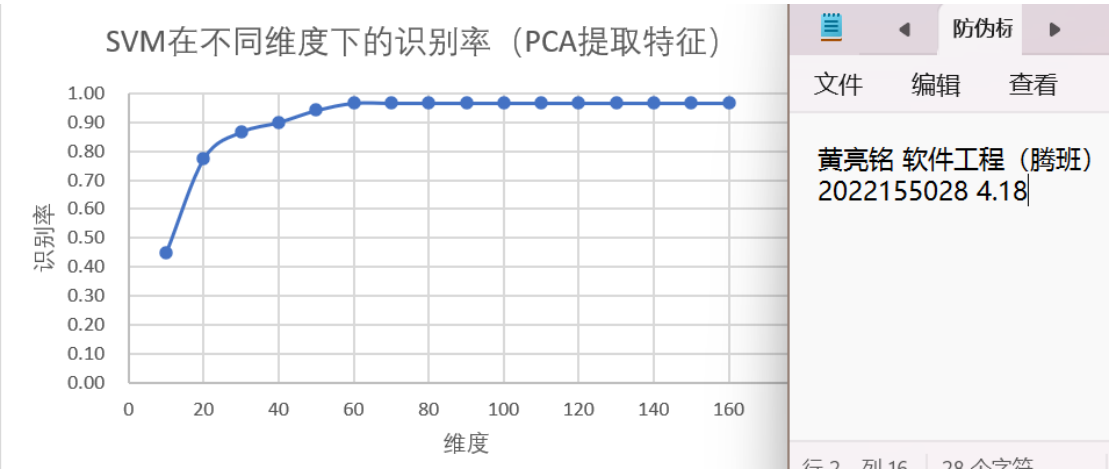
b) 使用 PCA 提取图像特征

```
% PCA
all_mean = mean(train_data, 2); % 全部的平均
centered_face = (train_data - all_mean);
cov_matrix = centered_face * centered_face';
target = cov_matrix;

% 求特征值、特征向量
[eigen_vectors, dianogol_matrix] = eig(target);
eigen_values = diag(dianogol_matrix);

% 对特征值、特征向量进行排序
[sorted_eigen_values, index] = sort(eigen_values, 'descend');
eigen_vectors = eigen_vectors(:, index);
eigen_vectors = real(eigen_vectors);
rate = []; %用于记录人脸识别率
```


图 4：PCA 提取特征

c) 使用 PCA 提取图像特征的 SVM 识别率

由图 5 可以得到信息：维度与识别率的关系为对数关系。维度较低时，识别率较低，这是因为从样本中获得的信息过少，导致 SVM 识别不准确；当维度达到 60 时，识别率稳定在 97%。



5：PCA 提取图像特征的 SVM 识别率

d) **使用 LDA 提取图像特征**

```
% LDA
% 算每个类的平均
k = 1;
class_mean = zeros(dimension, people_num);
for i=1:people_num
    % 求一列（即一个人）的均值
    temp = class_mean(:,i);
    % 遍历每个人的train_pic_num_of_each张用于训练的脸，相加算平均
    for j=1:each_train_pic_num
        temp = temp + train_data(:,k);
        k = k + 1;
    end
    class_mean(:,i) = temp / each_train_pic_num;
end

% 类间散度矩阵Sb
Sb = zeros(dimension, dimension);
all_mean = mean(train_data, 2); % 全部的平均
for i=1:people_num
    % 中心化
    center_data = class_mean(:,i) - all_mean;
    Sb = Sb + center_data * center_data';
end
Sb = Sb / people_num;
% 类内散度矩阵Sw
Sw = zeros(dimension, dimension);
k = 1;
for i=1:people_num
    for j=1:each_train_pic_num
        center_data = train_data(:,k) - class_mean(:,i);
        Sw = Sw + center_data * center_data';
        k = k + 1;
    end
end
Sw = Sw / (people_num * each_train_pic_num);
target = pinv(Sw) * Sb;
```

防伪标 文件 编辑 查看

黄亮铭 软件工程（腾班）
2022155028 4.18

行 2，列 16 28 个字符

```
% 求特征值、特征向量
[eigen_vectors, dianogol_matrix] = eig(target);
eigen_values = diag(dianogol_matrix);

% 对特征值、特征向量进行排序
[sorted_eigen_values, index] = sort(eigen_values, 'descend');
eigen_vectors = eigen_vectors(:, index);
eigen_vectors = real(eigen_vectors);
rate = []; %用于记录人脸识别率
```

防伪标 文件 编辑 查看

黄亮铭 软件工程（腾班）
2022155028 4.18

图 6：LDA 提取特征

e) **使用 LDA 提取图像特征的 SVM 识别率**

由图 5 可以得到信息：维度与识别率的关系为对数关系。维度较低时，识别率较低，这是因为从样本中获得的信息过少，导致 SVM 识别不准确；当维度达到 60 时，识别率稳定在 98%。

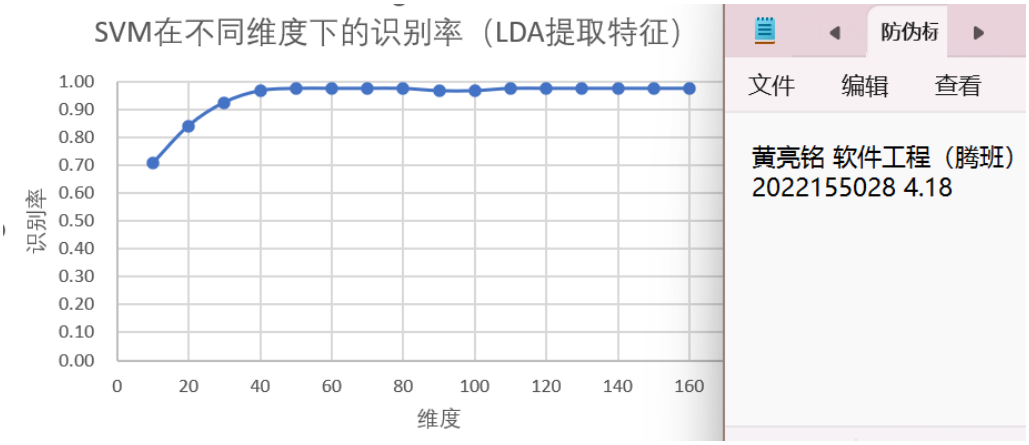


图 7：LDA 提取图像特征的 SVM 识别率

f) **小结**

由图（图 8）我们可以得到信息：① 维度和 PCA 提取特征的 SVM 识别率、LDA 提取特征的 SVM 识别率在图像上均呈现对数关系；② 在维度较低时，LDA 提取特征的 SVM 识别率优于 PCA 提取特征的 SVM 识别率；③ 在维度较高时，LDA 提取特征的 SVM 识别率也略高于 PCA 提取特征的 SVM 识别率。

综上，如果只看识别率，LDA 提取特征的方法优于 PCA 提取特征的方法，但是 LDA 在处理高维数据的计算代价非常大。事实上，执行程序的所用时间也体现了这一点。因此，选择 PCA 提取特征，还是选择 LDA 提取特征，需要根据具体情况确定。

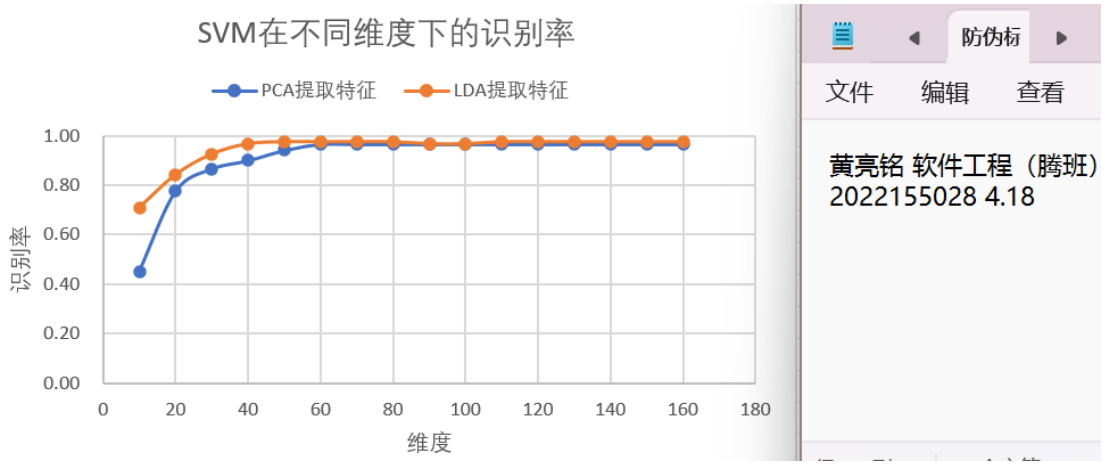


图 8：不同提取特征方法的 SVM 识别率对比

四、实验结论或体会

1. 经典的支持向量机算法在处理线性可分问题时表现良好，但在面对线性不可分问题时效果有限。
2. 软间隔支持向量机通过引入松弛变量允许部分数据点位于间隔内部，提高了模型的鲁棒性，适用于处理一些噪声较多的情况。
3. 核-SVM 通过核技巧将数据映射到高维空间中，从而处理非线性可分问题，但需要注意选择合适的核函数以及调节相关参数。
4. PCA 和 LDA 作为常用的降维算法，能够有效地减少特征维度，提高计算效率，但在处理非线性数据时可能存在局限性。
5. 随着特征维度的增加，图像识别的性能通常会有所提高，因为更多的信息被保留下来，但也存在维度灾难的问题，可能导致过拟合。

指导教师批阅意见： 成绩评定： 指导教师签字： 年　　月　　日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。