華中科技大學

课 程 实 验 报 告

课程名称： 机器学习结课实验

专业班级：校交 1802

学　　号：U201811187

姓　　名：李俊

指导教师：何琨

报告日期：2020 年 6 月 1 日

计算机科学与技术学院

目 录

基于语音识别的性别识别

摘要：本次实验主要使用手写 SVM 模型解决非线性分类问题，注重通过多种方法进一步优化 SVM 模型以达到预期的正确率。同时辅以决策树，随机森林算法等模型作为基准参考进行对比，给出 SVM 的模型评估，分析分类错误的数据，找到分类出错的原因，最后结合训练好的模型进行实际应用。

1. 引言

1.1 问题来源

问题来自于 Kaggle，要求训练一个模型，使之能够根据声音识别说话人的性别。根据声音识别一个人的性别对人来说是一个很容易的事情，人可以通过仅仅数个词语，或者语音片段就可以判断出说话的人是男是女。然而，设计实现一个计算机程序来完成相同的事情却有些棘手。

1.2 问题背景

性别识别是语音信号处理中一个很重要的课题,他与语音识别、说话人识别、语音通信等都有很大的联系。在语音识别和说话人识别实验中发现，事先知道说话人性别时所得到的正确识别率要比不知道说话人性别时高。在语音通信中,可以基于性别识别建立性别有关的语音特征参数提取方案,减少特征参数的维数,减少传输带宽。由此可见性别识别是语音识别研究中的一个重要问题，具有重要意义。

2. 算法设计

2.1 SVM 算法简介

支持向量机（support vector machines, SVM）是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机；SVM 还包括核技巧，这使它成为实质上的非线性分类器。SVM 的的学习策略就是间隔最大化，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。SVM 的的学习算法就是求解凸二次规划的最优化算法。

2.2 SVM 算法原理

对于输入空间中的非线性分类问题，可以通过非线性变换将它转化为某个维特征空间中的线性分类问题，在高维特征空间中学习线性支持向量机。由于在线

性支持向量机学习的对偶问题里，目标函数和分类决策函数都只涉及实例和实例之间的内积，所以不需要显式地指定非线性变换，而是用核函数替换当中的内积。核函数表示，通过一个非线性转换后的两个实例间的内积。具体地，$k(x,z)$ 是一个函数或正定核，意味着存在一个从输入空间到特征空间的映射，对任意输入空间中的x,z，有

$$K(x,z)=\phi(x)\cdot\phi(z)$$

在线性支持向量机学习的对偶问题中，用核函数 $k(x,z)$替代内积，求解得到的就是非线性支持向量机。

2.3 SVM 算法实现步骤

输入：训练数据集：

$$T=\{(x_1,y_1),(x_1,y_1),\dots,(x_N,y_N)\}$$

输出：分离超平面和分类决策函数

①选取适当的核函数$k(x,z)$和惩罚参数$C>0$，构造并求解凸二次规划问题。

$$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(\boldsymbol{x_i}\cdot\boldsymbol{x_j})-\sum_{i=1}^{N}\alpha_i$$

$$\text{s.t.}\quad\sum_{i=1}^{N}\alpha_i y_i=0$$

得到最优解：$\alpha^*=(\alpha_1^*,\alpha_2^*,\dots,\alpha_N^*)^T$

②选择$\boldsymbol{\alpha}^*$的一个分量$\boldsymbol{\alpha}_j^*$满足条件 $0<\boldsymbol{\alpha}_j^*<C$，计算

$$b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i K(x_i,x_j)$$

③计算分类决策函数

$$f(x)=sign\left(\sum_{i=1}^{N}\alpha_i^* y_i K(x,x_i)+b^*\right)$$

以高斯核函数$K(x,z)=\exp\left(-\frac{|x-z|^2}{2\sigma^2}\right)$为例

对应的 SVM 是高斯径向基函数分类器，在此情况下，分类决策函数为

$$f(x)=sign\left(\sum_{i=1}^{N}\alpha_i^* y_i\exp\left(-\frac{|x-z|^2}{2\sigma^2}\right)+b^*\right)$$

2.4 核函数及优缺点

线性核（Linear Kernel）：

$$k(x,y)=x^Ty+c$$

优点：做什么问题可以从最简单的线性开始并且简单，可以求解较快一个 QP 问题。

缺点：只能解决线性可分问题

多项式核（Polynomial Kernel）：

$$k(x,y)=(ax^Ty+c)^d$$

优点：可解决非线性问题，可通过主观设置幂数来实现总结的预判。

缺点：对于大数量级的幂数，不太适用；比较多的参数需要选择。

高斯核（Gaussian Kernel）：

$$k(x,y)=\exp\left(-\frac{|x-y|^2}{2\sigma^2}\right)$$

优点：可以映射到无限维；决策边界更为多样；只有一个参数，相比多项式核容易选择

缺点：可解释性差，无限多维的转换；解一个对偶问题计算速度比较慢；参数选不好时容易过拟合。

2.5 决策树简介

决策树是一种常见的机器学习算法。它基于树结构来进行决策，与人类在面临决策问题时的思考方式类似。决策树是一种十分常用的分类方法，需要监管学习（Supervised Learning），给出一堆样本，每个样本都有一组属性和一个分类结果，也就是分类结果已知，那么通过学习这些样本得到一个决策树，这个决策树能够对新的数据给出正确的分类。

2.6 随机森林简介

随机森林顾名思义，是用随机的方式建立一个森林，森林里面有很多的决策树组成，随机森林的每一棵决策树之间是没有关联的。在得到森林之后，当有一个新的输入样本进入的时候， 就让森林中的每一棵决策树分别进行一下判断，看看这个样本应该属于哪一类（对于分类算法），然后看看哪一类被选择最多，就预测这个样本为那一类。

3. 实验环境与平台

3.1 实验环境与平台

实验操作系统为：windows10 professional,硬件方面 CPU 为：core i7-8565U,内存为：8G。编程环境为：Jupyter Notebook，python 版本为：python-3.7.7，所需运行库有 matplotlib、numpy、pandas 及 sklearn。

3.2 实验数据集

实验数据集来自 Kaggle(voiceGender)，集合中共有 3168 条数据，男女各 1584 条，每条数据可视作一个长度为 21 的一维数组。其中前 20 个数值是这条语

音的 20 个特征值，这些特征值包括了语音信号的长度、基频、标准差、频带中值点/一分位频率/三分位频率等；最后一个数值是性别标记。元数据集中直接以字符串,即 male 和 female 进行标注。

4. 程序实现

4.1 基准模型（benchmark）

在 kaggle 的问题描述页上，给出了到目前为止不同模型最好的成绩。

表 1 基准模型

监督学习算法	训练集准确率	测试集准确率
BaseLine(总预测男性)	50%	50%
Logistic Regression	97%	98%
CART	96%	97%
Random Forest	100%	98%
SVM	100%	99%
XGBoost	100%	99%

所有的机器学习算法训练出的最终模型都得到了近乎完美的准确率。Kaggle 上提供的一个快速入门，使用决策树以 meanfun 和 IQR 为区分特征，实现了 96.0%的准确率。因为 96.0%已经是一个非常高的准确率，已经非常接近人根据声音识别性别的准确率，所以本项目尝试通过对 SVM 进行优化再将准确率提高 2%，也就是将准确率提高到 98%左右。

4.2 数据预处理

①数据集中，label 栏下的数据是字符串形式的，为了便于处理，需要将这些字符串映射成数字，’male’映射到 1, ‘female’映射到 2。

②在划分训练集和测试集之前，需要将数据进行随机打乱，因为原始数据文件中，数据是按照性别排序的，不具有随机性。在将数据随机打乱后，将其中 20%的数据作为测试集，80%的数据作为训练集。为了评估模型的泛化能力，使用交叉验证。即将数据分为 5 组互斥子集，每组占总数据的 20%，每次使用一组数据作为测试集，剩下 4 组做训练集，重复 5 次，使得每一组数据都充当过测试集的角色。取 5 个测试集最终准确率的平均数作为最终准确率。

③对于本实验，数据集一共包括 3168 条数据，数据集的 20%也就是 634

条数据，也就是测试集包括 634 条数据，训练集包括 2534 条数据。

④将 voice.csv 中已经数值化的数据转换成 person-feature 矩阵，矩阵的值为特征值。

	feat1 (meanfreq)	feat2 (sd)	feat3 (median)	...	feat21 (label)
person1	val1	val1	val1		val1
person2	val2	val2	val2		val2
person3	val3	val3	val3		val3
	.	.	.		.
	.	.	.		.
	.	.	.		.
person3167	val3167	val3167	val3167		val3167
person3168	val3168	val3168	val3168		val3168

图 1 转换矩阵

4.3 初步实现

4.3.1 支持向量机

使用手写的 SVM 模型，在使用 Cost Function 为：

$$\max W(\alpha) = \sum_{i=1}^{n} \alpha - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j a_i a_j \left(K(x_i, x_j)\right)$$

线性核函数和固定惩罚参数 C 情况下，使用训练集进行训练一个模型。

SVM 初步实现代码：

```
class SVM(object):
    """
    SVM model
    """
    def __init__(self, max_iter=10000, kernel_type='linear', C=1.0, epsilon=0.00001):
        self.max_iter = max_iter
        self.kernel_type = kernel_type
        self.kernel_func_list = {
            'linear': self._kernel_linear,
            'quadratic': self._kernel_quadratic,
        }
        self.kernel_func = self.kernel_func_list[kernel_type]
        self.C = C
        self.epsilon = epsilon
        self.alpha = None

    def train_raw(self, X_train, Y_train):
        """
        Training model
        :param X_train: shape = num_train, dim_feature
        :param Y_train: shape = num_train, 1
        :return: loss_history
        """
        n, d = X_train.shape[0], X_train.shape[1]
        self.alpha = np.zeros(n)
        # Iteration
        for i in range(self.max_iter):
            diff = self._iteration(X_train, Y_train)
```

```
                if i % 100 == 0:
                    print('Iter %r / %r, Diff %r' % (i, self.max_iter, diff))
                if diff < self.epsilon:
                    break

        def predict_raw(self, X):
            return np.dot(self.w.T, X.T) + self.b

        def predict(self, X):
            return np.sign(np.dot(self.w.T, X.T) + self.b).astype(int)

        def _iteration(self, X_train, Y_train):
            alpha = self.alpha
            alpha_prev = np.copy(alpha)
            n = alpha.shape[0]
            for j in range(n):
                # Find i not equal to j randomly
                i = j
                for _ in range(1000):
                    if i != j:
                        break
                    i = random.randint(0, n - 1)
                x_i, x_j, y_i, y_j = X_train[i, :], X_train[j, :], Y_train[i], Y_train[j]
                # Define the similarity of instances. K11 + K22 - 2K12
                k_ij = self.kernel_func(x_i, x_i) + self.kernel_func(x_j, x_j) - 2 * self.ke
rnel_func(x_i, x_j)
                if k_ij == 0:
                    continue
                a_i, a_j = alpha[i], alpha[j]
                # Calculate the boundary of alpha
                L, H = self._cal_L_H(self.C, a_j, a_i, y_j, y_i)
                # Calculate model parameters
                self.w = np.dot(X_train.T, np.multiply(alpha, Y_train))
                self.b = np.mean(Y_train - np.dot(self.w.T, X_train.T))
                # Iterate alpha_j and alpha_i according to 'Delta W(a_j)'
                E_i = self.predict(x_i) - y_i
                E_j = self.predict(x_j) - y_j
                alpha[j] = a_j + (y_j * (E_i - E_j) * 1.0) / k_ij
                alpha[j] = min(H, max(L, alpha[j]))
                alpha[i] = a_i + y_i * y_j * (a_j - alpha[j])
            diff = np.linalg.norm(alpha - alpha_prev)
            return diff

        def _kernel_linear(self, x1, x2):
            return np.dot(x1, x2.T)

        def _kernel_quadratic(self, x1, x2):
            return np.dot(x1, x2.T) ** 2

        def _cal_L_H(self, C, a_j, a_i, y_j, y_i):
            if y_i != y_j:
                L = max(0, a_j - a_i)
                H = min(C, C - a_i + a_j)
            else:
                L = max(0, a_i + a_j - C)
                H = min(C, a_i + a_j)
            return L, H
```

4.3.2 决策树

实现一个简单的决策树，查看预测准确率。从 sklearn 库中引入决策树模块，设置 random_state 为 1，训练决策树。查看训练结果，测试集准确率为 95.8%，训练集的准确率为 1.0。从测试集的准确率上看，已经非常接近实验的目标 98%

4.3.3 随机森林

同理，从库中训练一个随机森林模型，参数设置为默认参数。查看模型在测试集上的表现，准确率达到 98%，这一个非常高的分数，和 kaggle 上最佳随机森林模型的准确率几乎相同。

4.4 SVM 优化

因为决策树和随机森林的表现都很好，对这两种算法进行改进难以看出改进效果，所以本文主要研究手写支持向量机的改进方法。可以考虑从不同角度优化支持向量机，本项目主要从两方面优化支持向量机，即数据标准化和参数方面。

4.4.1 数据标准化

数据的标准化（normalization）是将数据按比例缩放，使之落入一个小的特定区间。在某些比较和评价的指标处理中经常会用到，去除数据的单位限制，将其转化为无量纲的纯数值，便于不同单位或量级的指标能够进行比较和加权。

目前数据标准化方法有多种，归结起来可以分为直线型方法(如极值法、标准差法)、折线型方法(如三折线法)、曲线型方法(如半正态性分布)。不同的标准化方法，对系统的评价结果会产生不同的影响，然而不幸的是，在数据标准化方法的选择上，还没有通用的法则可以遵循。

本次实验的标准化是将每个属性的分布偏移为具有零的平均值和标准偏差为 1（单位方差）。它对标准化模型的属性很有用。数据集的标准化是在 scikit-learn 中实现的许多机器学习估计器的常见要求;如果单个功能没有或多或少看起来像标准的正态分布数据，它们可能表现不好。对数据标准化之后，再一次训练模型，查看训练结果。

4.4.2 调式核函数

在数据标准化后，分别尝试 SVM 结合线性核（Linear Kernel），多项式核（Polynomial Kernel），高斯核（Gaussian Kernel）进行不同核函数的对比，最后选择正确率最佳的核函数进行进一步优化。

4.4.3 调试其他参数

除了核函数之外，支持向量机还有两个重要参数，C、gamma。首先调试参数 C。

参数 C 相当于惩罚变量或松弛参数，当 C 的值越大，对错误分类的惩罚越严重，此时模型趋向于训练集全分类对的情况。这时，训练集的准确率很高，不过模型的泛化能力会下降，也就是倾向于过拟合。C 的值越小，对错误分类

的惩罚越小，允许出错，将一些分类出错的点当成噪声忽略。

若最终采取的是 RBF 核函数，那么 gamma 参数是支持向量机众多参数中出参数 C 之外另一个比较重要的参数。Gamma 参数是 rbf 核函数（高斯函数）的标准差的倒数。gamma 的值用于评估两个点的相似性。小的 gamma 值意味着模型有一个大的方差。在这种情况下，即使两个点相距较远也会被认为是相似的。另一方面，大的gamma 值以为一个小方差的模型，这种情况下，当两个点距离比较近时才会被认为是相似的。

5.实验结果

5.1 数据集简要分析

查看前 7 个样本的数据，观察识别声音的特征值。

	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	sfm	...
0	0.059781	0.064241	0.032027	0.015071	0.090193	0.075122	12.863462	274.402906	0.893369	0.491918	...
1	0.066009	0.067310	0.040229	0.019414	0.092666	0.073252	22.423285	634.613855	0.892193	0.513724	...
2	0.077316	0.083829	0.036718	0.008701	0.131908	0.123207	30.757155	1024.927705	0.846389	0.478905	...
3	0.151228	0.072111	0.158011	0.096582	0.207955	0.111374	1.232831	4.177296	0.963322	0.727232	...
4	0.135120	0.079146	0.124656	0.078720	0.206045	0.127325	1.101174	4.333713	0.971955	0.783568	...
5	0.132786	0.079557	0.119090	0.067958	0.209592	0.141634	1.932562	8.308895	0.963181	0.738307	...
6	0.150762	0.074463	0.160106	0.092899	0.205718	0.112819	1.530643	5.987498	0.967573	0.762638	...

图 1 样本属性

可见 voice.csv 中的每个样本包含 20 个特征值，这些特征值包括了语音信号的长度、基频、标准差、频带中值点/一分位频率/三分位频率等等数据。

5.2 特征之间的关系

为了进一步查看特征之间的关系，将两个特征作为一组，遍历所有的特征组合，使用 matplotlib 库输出性别关于这些特征组合的二维分布图。一共得到 210 副图。

通过观察这 210 副图像，可以看出平均基音频率（meanfun）是一个非常有效的特征，任何特征和平均基音频率组合都能够将男女分为两个明显的簇。

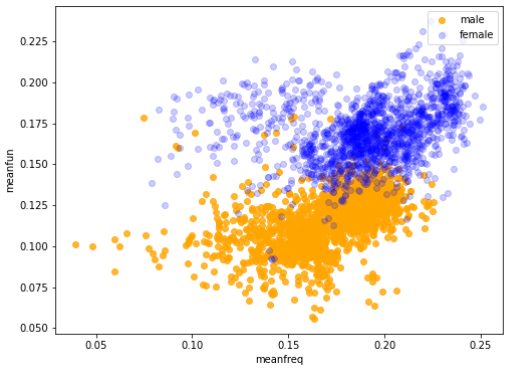

图 2 平均基音频率（meanfun）分簇

5.3 优化前的结果

通过在交叉验证运行优化前的程序发现未经优化的结果不太理想。因此需要进行优化，否则达不到预期的准确率。

```
In [16]: def main(x_train, y_train, x_test, y_test, learning_rate , para):
             # initialize
             model = SVM()
             dimension =  x_train.shape[0]
             w,b = initialize_weights_and_bias(dimension)
             parameters, gradients, cost_list = update(w, b, x_train, y_train, learning_rate,para)
             y_prediction_test = predict(parameters["weight"],parameters["bias"],x_test)
             print("test accuracy: {} %".format(100 - np.mean(np.abs(-0.1+y_prediction_test - y_test)) * 100))

         main(x_train, y_train, x_test, y_test,learning_rate = 1, para = 300)

         test accuracy: 87.38170347003154 %
```

图 3 优化前运行结果

由上图可见，优化前正确率只有 87.38%

与实现的基准模型对比：

表 2 基准模型对比

模型	交叉验证
SVM	87.38%
Random forest	98.79%
Decision tree	95.39%

5.4 数据标准化

通过将数据标准化，即将每个属性的分布偏移为具有零的平均值和标准偏差置为 1 后，发现支持向量机预测准确率达到了 96%。使用交叉验证查看预测准确率为 94%。

表 3 数据标准化前后准确率

方案	训练集准确率	测试集准确率	交叉验证准确率
标准化后	96.31%	96.42%	94.55%
标准化前	87.45%	87.96%	87.38%

从表格中的数据可以看出，交叉验证的准确率要明显低于使用一个测试集和训练集时的模型准确率。这是因为交叉验证将数据集分成了多个不同的训练集和测试集，并将模型在所有测试集上的准确率的平均值作为最终结果。因此交叉验证的准确率会低一些，不过这也使得交叉验证的准确率更接近模型的真实准确率。

5.5 调试核函数

在数据标准化后，分别尝试 SVM 结合线性核（linear），多项式核（poly），高斯核（rbf）进行不同核函数的对比。

调试支持向量机的参数，查看当核函数（kernel）不同时，交叉验证的准确

率。使用的数据是经过标准化处理的数据。

表 4 采用不同核函数支持向量机的表现

方案	交叉验证
kernel='linear'	96.71%
kernel='rbf'	96.79%
kernel='poly'	93.39%

从表 3 中可以看出，采用'linear'和'rbf'核函数的模型表现比较好，而使用'poly'核函数的模型相对差一些。所以接下来在调试其他参数时，核函数（kernel）均使用 rbf。其中，不同核函数的分类超平面如下图所示。

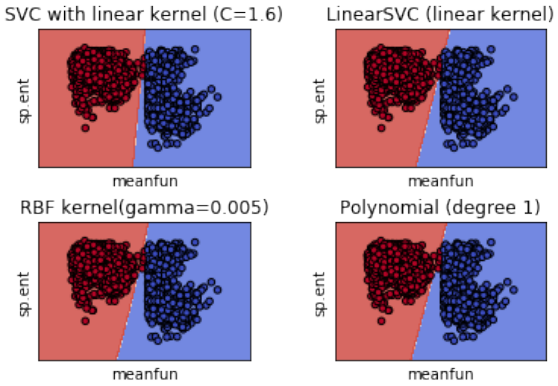

图 4 不同核函数的分类超平面

5.6 调试惩罚变量

参数 C 为惩罚变量，当 C 的值越大，对错误分类的惩罚越严重，此时模型趋向于训练集全分类对的情况。C 的值越小，对错误分类的惩罚越小，允许出错，将一些分类出错的点当成噪声忽略。

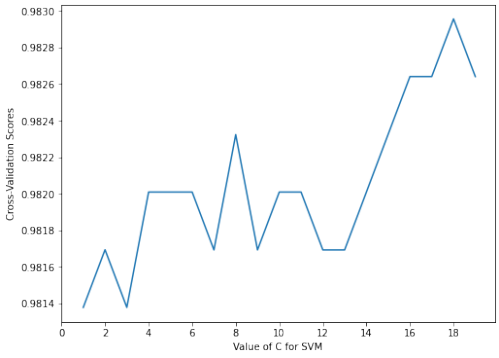

图 5 参数 C 取值 1-20 时的准确率

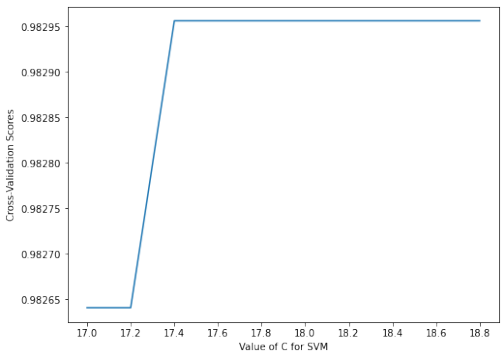

图 6 参数 C 取值 17.0-19.0 时的准确率

通过图 5，查看 C 取值在 1 和 20 之间时模型准确率最大值，可以观察到模型准确率在大约 C=18 时达到最高。

接下来将尝试使用更小的步长， 查看 C 取值在 17.0 和 19.0 之间时，什

么时候能够取得模型准确率最大值。如图 6 所示，当 C 取值 17.4 时，模型准确率最高，为 98.29%左右

5.7 调试 Gamma 参数

Gamma 参数是 rbf 核函数（高斯函数）的标准差的倒数，用于评估两个点的相似性。小的 gamma 值意味着模型有一个大的方差。在这种情况下，即使两个点相距较远也会被认为是相似的。另一方面，大的 gamma 值以为一个小方差的模型，这种情况下，当两个点距离比较近时才会被认为是相似的。

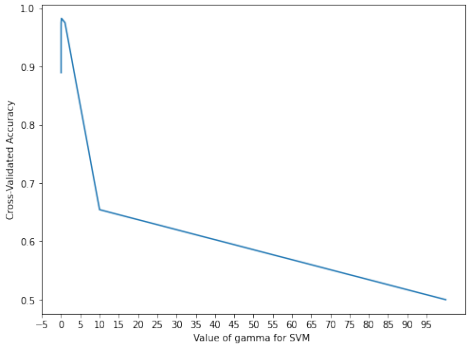

图 7 gamma 取值变换大时的准确率

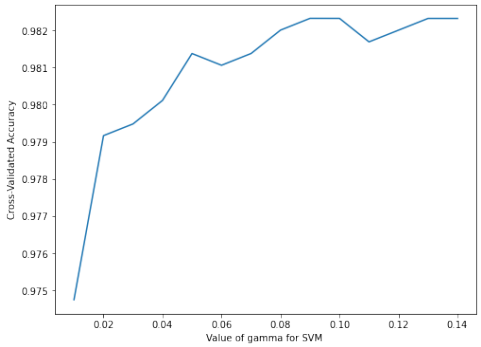

图 8 gamma 取 0.10 左右时的准确率

通过对图 7 的观察，可以发现，当 gamma 取 0 到 100 之间，模型在 gamma 等于 0.1 左右时，准确率达到最高。图 8 显示，当 gamma 等于 0.090 时，模型准确率达到最高值，在 98.20%左右。

最后，进行整体调试支持向量机的参数，通过上面的探索，可以确定，最优参数 C 的取值在 16.0 到 19.0 之间，而参数 gamma 的最优取值在 0.010 到 0.20 之间，核函数可以选用的有”linear”和”rbf”。所以，可以使用网格搜索，选取一个最优的参数组合。

```
In [16]: from sklearn.model_selection import GridSearchCV

         # 参数字典
         params = tuned_parameters = {
          'C': (np.arange(0.1,1.0,0.1)) , 'kernel': ['linear'],
          'C': (np.arange(15.5,18.5,0.5)) , 'gamma': np.arange(0.01, 0.20, 0.05), 'kernel': ['rbf']
         }
         grid_obj = GridSearchCV(estimator=clf,
                                 param_grid=params,
                                 scoring='accuracy',
                                 cv=5)
         grid_obj.fit(features, gender)
         clf = grid_obj.best_estimator_

         print ('The best model parameters: ', grid_obj.best_params_)
         print ('The best model score: ', grid_obj.best_score_)

         The best model parameters:  {'C': 16.0, 'gamma': 0.060000000000000005, 'kernel': 'rbf'}
         The best model score:  0.983270202020202
```

图 9 网格搜索及最终优化后正确率

由图 9 所示，最终选用 rbf 核函数，参数 C 取 16.0，参数 gamma 取 0.06 时，模型准确率最高，交叉验证准确率在 98.32%左右

6.结果分析

6. 1 模型评估

使用上述讨论得出的最优参数组合{kernel='rbf', C='16.0', gamma=0.06}训练一个支持向量机的模型，并输出学习曲线，即准确率与训练集规模的关系。

训练集中样本至少包含 250 个样本，最多不超过 2500 个样本。从图 10 中可以看出，测试集的准确率随样本数目的增长一直增长，同时增长速率不断减小，最终趋于平缓。训练集的准确率在前半段随样本数目的增长而上升，后半段有些波动。

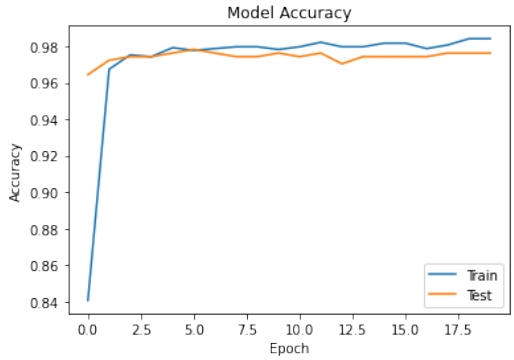

图 10 最优参数组合学习曲线

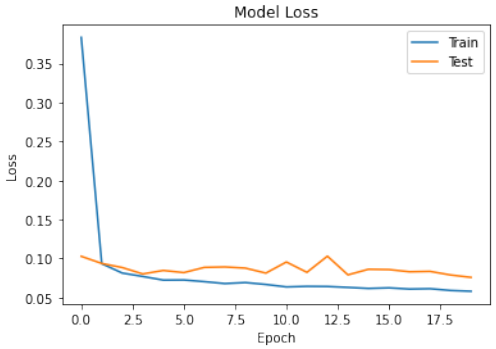

图 11 Loss 曲线

整体来看，训练集的准确率和测试集的准确率在不断接近。图像中没有显示训练集的准确率最终远超过测试集准确率，所以不认为有过度拟合和趋势。

6.2 样本随机检测

在数据集中随机选取 30 个样本，将数据标准化后，使用训练好的支持向量机模型进行预测。预测结果如图 11 所示。

数字是样本编号，编号下第一行是样本的真实性别，第二行是模型预测的性别。

	1	2	3	4	5	6	7	8	9	10
真实性别	female	female	female	male	male	female	male	male	male	male
预测性别	female	female	female	male	male	female	male	male	male	male
	11	12	13	14	15	16	17	18	19	20
真实性别	male	female	female	female	male	male	female	male	female	male
预测性别	male	female	female	female	male	male	female	male	female	male
	21	22	23	24	25	26	27	28	29	30
真实性别	male	male	female	female	female	male	female	male	male	female
预测性别	male	male	female	female	female	male	female	male	male	female

图 12 30 个随机测试样本分类结果

通过结合图 11，可以看出随机选的小规模样本预测结果与真实性别一样。

6.3 错误分类检查

将错误分类的样本保存成一个单独的文件，然后查看这些错误分类的样本在数据集中是怎样分布的。图 12 是样本点关于平均频率（meanfreq）、频率四分

位间距（IQR）和平均基音频率（meanfun）的分布图。

红色代表正确分类的男性，绿色代表正确分类的女性。蓝色代表被 SVM 模型错分为男性的样本，紫色代表被 SVM 模型错分为女性的样本。

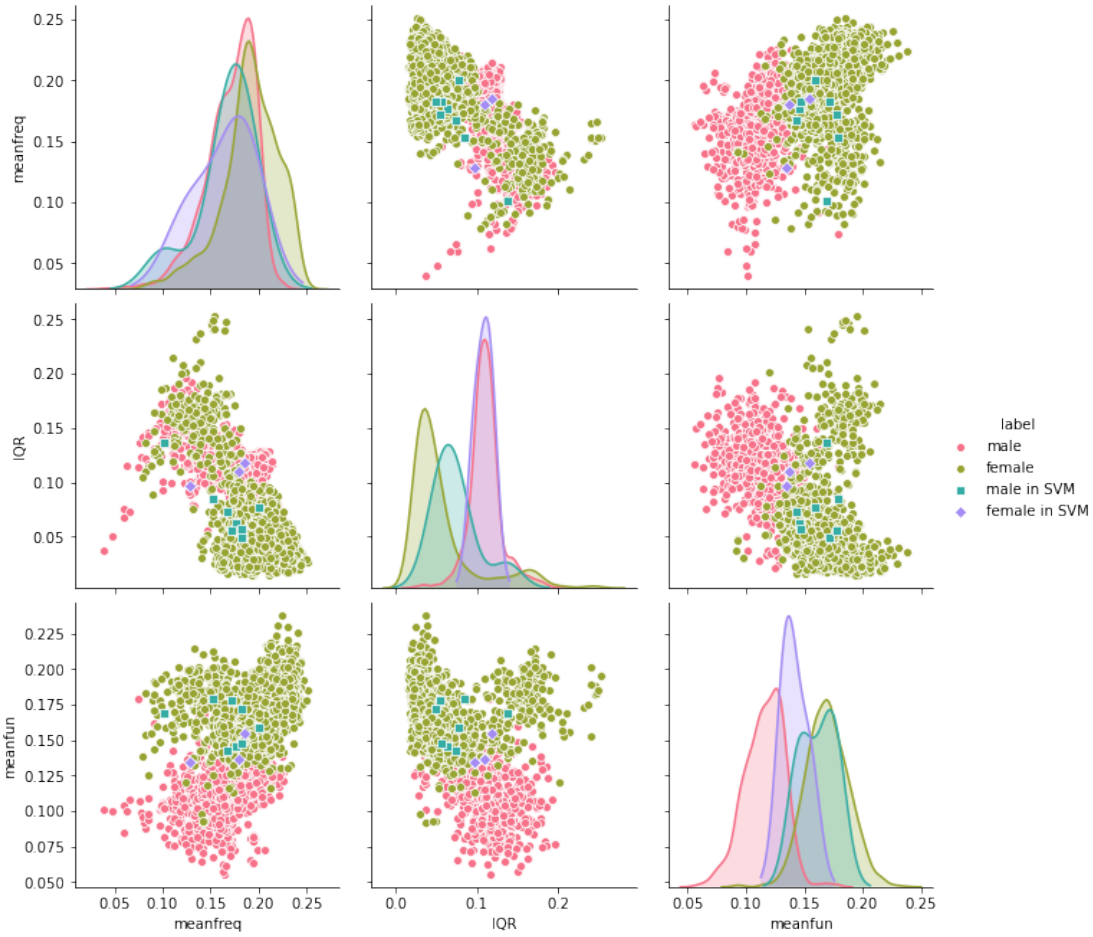

图 13 错误分类样本分析

由图 12 可见，被错误分类的样本分布集中，主要位于两个簇的交界地区，也就是声音特征比较倾向于中性。因此导致模型难以区分这些人的应该归到哪一类。还有少量样本落到异性簇的中间，使得模型进行了错误的分类。

实际上，在现实生活中，有一些人的声音确实很特殊，使得我们仅仅通过声音无法判断说话人的性别。所以模型对极少量的样本分类错误也是正常的。

通过以上讨论，最终训练出来的支持向量机在交叉验证中达到了 98.32%的准确率，已经符合了项目最初要求。

本次实验距 Kaggle 所提供的 benchmark 的最优 SVM 模型正确率仅相差 1%左右，因此已经能够保证在绝大多数情况下正确识别出说话人的性别。

7.拓展应用

从应用角度来讲，可以基于性别识别建立性别有关的语音特征参数提取方案，减少特征参数的维数,减少传输带宽，以此提高语音传输效率或识别精度或者可

以基于本实验建立一个完整的声源性别识别方法。

7.1 语音信号处理

语音是一种波。常见的 MP3 格式都是一种压缩格式，必须转化成非压缩的纯波形文件来处理，wav 文件格式，wav 内部存储的除了一个文件头以外，就是声音波形的一个个点。

图 14 MP3 格式转换成 WAV 波形

端点检测：首尾端静音切除

分帧：用窗函数进行分帧，之间存在着交叠（帧移），语音信号短时平稳，长时非平稳。目前采用移动窗函数进行分帧。

7.2 声学特征提取

时域内波形几乎没有表达能力，因此需要将波形做变换(MFCC)

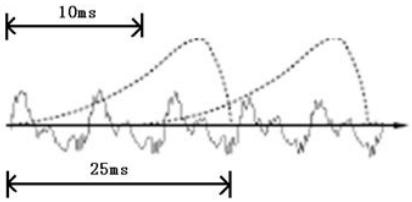

图 15 MFCC 波形变换

在语音识别领域当中，使用对角协方差矩阵的 GMM，将 MFCC 作为声学特征一直是研究的常用手法。

MFCC 声学特征的计算过程如图 16 所示。

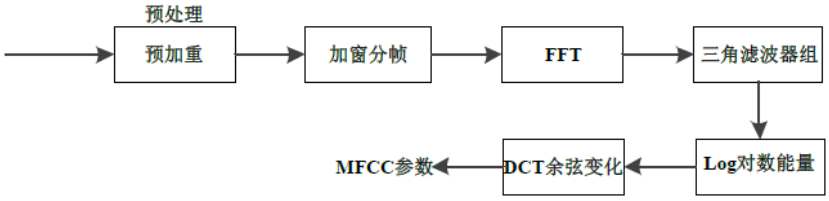

图 16 MFCC 计算流程图

经预处理和快速傅里叶变换(Fast Fourier transformation, FFT)得到语音信号各帧数据的频谱参数，通过一组 N 个三角带滤波器构成的 Mel 频率滤波器作卷积运算，然后对输出的结果作对数运算，依次得到对数能量$S(m), m = 1,2,3,\ldots, N$，最后经离散余弦变换(Discrete cosine transform, DCT)，得到 MFCC 参数

$$C_i(n) = \sum_{m=1}^{M} S(m)\cos\left[\frac{\pi n(m-0.5)}{M}\right], 0 \le m \le M$$

其中， n 代表 MFCC 声学特征的个数， C (n) 是第 i 帧的第 n 个 MFCC 系数，作为 Log 对数能量模块的输 出， M 是 Mel 滤波器的个数。

7.3 实验结果

①语音信号处理结果：

使用任意设备录制一段人声，保存为 MP3 格式，通过 7.1 的相关原理，转换成 WAV 波形，便于后续进行特征处理。

录取一段电影教父的台词，保存为 20200622_090350.mp3

```
Loading 20200622_090350.mp3
Voice signal processing completed.
```

截取 0s-0.8s 生成的 WAV 波形图如图 17 所示

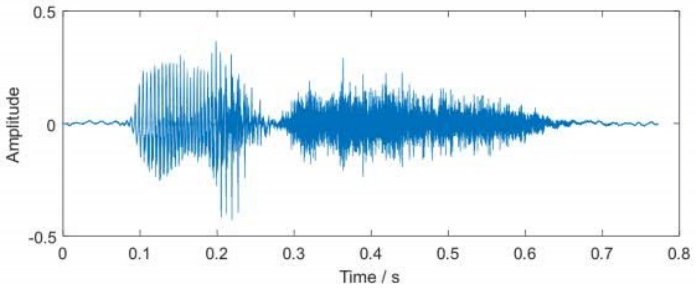

图 17 生成的 WAV 波形图

②声学特征提取

将 WAV 波形图通过原理 7.2 转换成本实验所需的 20 个特征值（基频、标准差、频带中值点等），转换后的结果如下所示。

表 4 声波转换后特征值

feature	meanfreq	sd	median	Q25	Q75
val	0.169353	0.068168	0.142837	0.115805	0.239285
feature	mode	centroid	meanfun	minfun	maxfun
val	0.120477	0.169353	0.11109	0.023256	0.181818
feature	IQR	skew	kurt	sp.ent	sfm
val	0.12348	2.038745	7.546507	0.921624	0.48264
feature	meandom	mindom	maxdom	dfrange	modindx
val	0.917969	0.007813	5.59375	5.585938	0.239254

③结果预测

将录取声源产生的一组特征值作为样本向量，使用上面提到的本实验已经训练好的 SVM 模型进行预测，结果如下：

```
forecast result ： male
```

8.结语

8.1 总结

本文实现了一个根据语音数据识别性别，准确率为 98.32%的支持向量机模型。在数据预处理的步骤中，通过将数据标准化，极大地提高了模型的识别率，验证了标准化数据对支持向量机来说非常重要。然后为了进一步提高模型预测的精准度，找到参数 C 和参数 gamma 的最佳范围，而后使用网格搜索算法找到了参数 C 和参数 gamma 的最佳组合。通过对学习曲线的分析，证明模型没有出现过拟合现象。此外，还观察了被错误分类的样本被错误分类的原因，因为这些人的声音特征比较中性或者和异性的声音特征比较相似，所以被错误地分类。

从整体来看，准确率 98.32%已经接近 Kaggle 所提供的 benchmark 的最优 SVM 模型，同时项目已经完成了准确率 98%左右的目标。

拓展应用实现了 MP3 格式的声源的性别预测，而且实验结果良好，其中部分代码使用了开源项目框架（如 tensorflow 等），极大提升了自己的代码水平。

8.2 改进

从数据集角度，项目使用的数据集包括 3168 条数据，从数量方面来说，3168 个样本难以代表整个人类群体。所以，本项目的一个改进方向是扩大项目的数据集，采集更多的样本用于训练。在这种情况下，训练出来的更具有代表性。

从应用角度，噪声问题一直是语音识别的一个非常困难的问题，在理想的实验室的环境下，识别效果已经非常好了，在给识别做降噪时，经常发现 WER 不降反升，降低了识别率，导致无处下手。由于 DNN 模型本身就有很强的抗噪性，因此可以尝试 DNN 模型进行降噪。

参考文献

[1] 李航．统计学习方法．北京：清华大学出版社，2012：第七章，pp.95-135.

[2] 周志华．机器学习．北京：清华大学出版社，2016：pp.121-139, 298-300.

[3] 刘明. 支持向量机中核函数的研究[D]. 西安电子科技大学, 2009.

[4] 张思懿. 基于核方法的异常检测技术的研究[D]. 江南大学, 2012.

[5] 杨行俊，迟惠生. 语音信号数字处理[M]. 北京: 电子工业出版社,1995.

[6] 胡政权, 曾毓敏, 宗原, 等. 说话人识别中 MFCC 参数提取的改进[J]. 计算机工程与应用, 2014, 50(7): 217-220.

[7] Kovacs G, Toth L, Compernolle D V, et al. Increasing the robustness of CNN acoustic models using autoregressive moving average spectrogram features and channel dropout[J]. Pattern Recognition Letters, 2017, 100(1): 44-50.

[8] kaggle官方文档. URL：https://www.kaggle.com/primaryobjects/voicegender