****

本科毕业论文(设计)

题 目：柠檬手机销售管理系统

学 院：数据科学与软件工程学院\_\_\_\_\_

专 业：软件工程(服务外包) \_\_

姓 名：刘宁宁 \_

指导教师：王卓 \_

2019年 05 月 24 日

青岛大学本科毕业论文（设计）任务书

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 论文（设计）题 目 | 柠檬手机销售管理系统 | | | |
| 学 院 | 数据科学与软件工程学院 | 专 业 | | 软件工程（服务外包） |
| 学生姓名 | 刘宁宁 | 学 号 | | 201540704138 |
| 选题来源 | 教师科研课题（ ）；生产、工程或社会实践课题（ ）  学生自拟课题（ √ ）；师生共同拟定课题（ ） | | | |
| 大学生创新创业训练项目（ ）； 学科竞赛 （ ） | | | |
| 研究目标或设计构想：  由于智能手机行业的快速发展，传统的手机管理系统已经不能适应该行业的发展的趋势，尤其是那些中小型自营的手机店家，还停留在记账等传统的管理模式上，不能高效，准确的提高销售营业额。该系统可以说是为中小型自营手机店家量身打造的。该系统可以合理划分，每款手机及其每款手机下各种型号的销售情况，每一天系统自动更新日销售情况，每一个月更新本月的销售情况，可以使店家清晰直观的看到每个月的销售情况，通过情况来制定营销计划。 | | | | |
| 研究内容或设计方案：  销售管理模块  该模块用于形成销售订单，该模块提供显示店内所有手机及其型号，在形成订单时，系统会为每一条订单自动形成一个唯一的订单编号，每一次制作订单时，系统会自动查询数据库中该下单的手机型号是否有现货，以及价格。订单形成后，相应的操作员可以撤回删除订单的，处该订单的操作员以及管理员，其他人无权删除。  统计分析模块  该模块用于统计分析每周每月的销售情况。通过订单情况和客户的反馈以视图的形式最直观的展现在店家的面前，系统中的视图会根据系统的设置定期更新数据，让店家以最快的时间获取最新的销售数据。  部门管理模块  该模块用于部门管理。由于中小型手机店家基本都是自体户，所以系统只提供两个本门的管理-销售本门和售后部门。该模块可以对部门进行添加，删除和修改，以及查看该部门下的所有员工。  财务管理模块  统计店家该月的销售额。通过销售订单，分类统计分析每一类手机的销售额以及每一类手机下每种型号手机的销售额。  研究方法或技术路线：   1. 中小型手机店家的销售管理情况， 2. 查阅文献 3. 软件工程   系统开发。按照数据库管理信息系统开发的流程，从需求分析开始，弄清当前系统的工作流程、存在的问题，提出解决方法；对新系统进行逻辑设计。包括数据库流程、数据字典的描述等等；进行系统总体设计，包括功能模块设计、数据库设计等；最后进行详细的代码设计、用户界面设计，并进行一定的测试。 | | | | |
| 进度安排：  3月上旬：动员组员进行中小型手机店家的销售管理的了解。  3月中下旬：开始初步规划系统设置及其功能设置。  4月：进行各自负责功能模块的开发设置。  5月上旬：合并各自开发的功能模块，进行初步测试。  5月中旬：输入大量数据，反复测试，解决bug。  5月下旬：测试，准备相关事项，进行答辩。 | | | | |
| 论文起止时间：  2019年 3月 1日—— 2019年 5月 1日 | | | | |
| 学生（签名）：刘宁宁  指导教师（签名）：  2019年 3月 1日 | | | 教研室意见：  教研室主任（签名）：  年 月 日 | |

注：本表格由学生填写，指导教师审核同意后，由教研室审批留存。

基于支持向量机的压力实时监测方法

郑 重 声 明

本人呈交的学位论文（设计），是在指导教师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。除文中已经注明引用的内容外，本学位论文（设计）的研究成果不包含他人享有著作权的内容。对本论文（设计）所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文（设计）的知识产权归属于青岛大学。

本人签名： 刘宁宁 日期： 2019.5.24

摘要

压力传感器有着广泛的应用，正确测量和控制压力对保证应用过程的安全性和经济性有重要意义，压力识别作为测量过程的关键组成部分，其敏捷度与精度对整个环节有着很大的影响。为了适应应用中更高的要求，需要找到具有更高精度与敏捷度的压力识别方式。

将机器学习方法应用于压力识别过程在近几年得到广泛的关注，其适用于大量数据的分析，具有较好的应用前景。

本文旨在设计开发一种具有高敏捷度，高精度的压力实时监测方法，该方法通过由压力传感器与单片机构成的监测模块实时获取数据，数据经过上位机软件加工处理后产生向量，并通过支持向量机对数据进行拟合产生分类模型，判断实时压力值是否在安全范围，并对不同状况作出即时反应。经过验证，此模型具有较高的灵敏度与准确率，可应用于多个场景。

关键词：压力传感器 ；支持向量机 ；压力实时监测 ；预测模型

Abstract

Pressure sensors have a wide range of applications. Correct measurement and control of pressure are important for ensuring the safety and economy of the application process. Pressure recognition is a key component of the measurement process, and its agility and precision have a great influence on the entire process. In order to adapt to the higher requirements in the application, pressure identification methods with higher precision and agility need to be found.

The application of machine learning methods to the pressure recognition process has received extensive attention in recent years. It is applicable to the analysis of large amounts of data and has a good application prospect.

This paper aims to design and develop a high-accuracy, high-precision real-time pressure monitoring method. This method acquires data in real time through a monitoring module consisting of a pressure sensor and a single-chip microcomputer. The data is processed by the upper computer software to generate vectors, and supports the data. The vector machine fits the data to generate a classification model to determine whether the real-time pressure value is in a safe range and respond to different conditions instantly. After verification, this model has high sensitivity and accuracy and can be applied to multiple scenarios.

Keywords: pressure sensor; support vector machine; real-time monitoring of pressure; prediction model

目录

[第一章 引 言 1](#_Toc514267794)

[1.1研究背景 1](#_Toc514267795)

[1.1.1 压力传感器的使用场景 1](#_Toc514267796)

[1.1.2敏捷度与精度在压力传感器识别中的重要性 1](#_Toc514267797)

[第二章 压力传感器检测模块 3](#_Toc514267798)

[2.1单片机构成 3](#_Toc514267799)

[2.1.1整体结构 3](#_Toc514267800)

[2.1.2压力传感器 3](#_Toc514267801)

[2.1.3 数模转换和信号放大器 4](#_Toc514267802)

[2.1.4 Cortex-M0+ 4](#_Toc514267803)

[2.1.5 EEPROM 5](#_Toc514267804)

[2.2压力传感器监测模块整体参数 5](#_Toc514267805)

[2.3 监测模块的启动停止指令 6](#_Toc514267806)

[2.4 其他功能部件 6](#_Toc514267807)

[2.4.1 单片机电源 6](#_Toc514267808)

[2.4.2 单片机USB连接线 6](#_Toc514267809)

[2.4.3 输出端 6](#_Toc514267810)

[第三章 支持向量机 7](#_Toc514267811)

[3.1简介 7](#_Toc514267812)

[3.2 支持向量机基本理论 7](#_Toc514267813)

[3.2.1 间隔与支持向量 7](#_Toc514267814)

[3.2.2 对偶问题 9](#_Toc514267815)

[3.2.3 核函数 9](#_Toc514267816)

[3.2.4 软间隔与正则化 11](#_Toc514267817)

[3.3 LIBSVM 14](#_Toc514267818)

[3.3.1简介 14](#_Toc514267819)

[3.3.2 LIBSVM工具使用说明 15](#_Toc514267820)

[3.3.3 执行windows文件夹下的exe文件 15](#_Toc514267821)

[3.3.4 在eclipse中建立工程执行 17](#_Toc514267822)

[3.3.5 LIBSVM参数 17](#_Toc514267823)

[3.3.6核函数选择 19](#_Toc514267824)

[3.3.7 核参数的选择 19](#_Toc514267825)

[第四章 上位机软件 21](#_Toc514267826)

[4.1 简介 21](#_Toc514267827)

[4.2 上位机软件实现 21](#_Toc514267828)

[4.2.1 获取端口名的方法 21](#_Toc514267829)

[4.2.2 16进制字符串和byte字符数组之间的相互转换 21](#_Toc514267830)

[4.2.3 主窗体设置，声明变量以及接口 22](#_Toc514267831)

[4.2.4 声明配置文件路径，以及接受数据的字符串 23](#_Toc514267832)

[4.2.5 接受和发送串口数据 23](#_Toc514267833)

[4.2.6 设置串口属性，定义不同的数据位、停止位、奇偶校验位以及波特率等默认数据 24](#_Toc514267834)

[4.2.7设置端口的显示状态，将从用户设置的参数显示在界面上 29](#_Toc514267835)

[4.2.8 应用程序的入口点，开始运行程序，调用多线程进行开发 30](#_Toc514267836)

[第五章 识别模型训练 32](#_Toc514267837)

[5.1通过传感器与单片机采集数据 32](#_Toc514267838)

[5.1.1 设定滑动窗口 32](#_Toc514267839)

[5.1.2候选窗口特征 33](#_Toc514267840)

[5.2 向量设计 34](#_Toc514267841)

[致 谢 35](#_Toc514267842)

[参考文献 36](#_Toc514267843)

# 引 言

## 研究背景

### 压力传感器的使用场景

压力传感器是工业实践、仪器仪表控制常用的传感器，在日常生活及各种工业自控环境中有着广泛的应用。下面简单介绍一些压力传感器的使用场景：

在称重系统中的应用：人们经常需要采集压力信号来实现压力控制，电子称重系统是一种使用压力传感器制作的压力控制装置，电子称重系统可以优化生产，又可以采集物料流动数据，其作为物料流动的控制工具显得越来越重要。

在石油化工中的应用：压力传感器作为测量装置在石油化工行业得到广泛的应用，石化行业对压力传感器的要求集中在可靠性、稳定性和高精度3个方面，主要有压力传感器的稳定性和测量精度保证。

在水处理中的应用：使用压力传感器将液体或气体压力转化为电信号，并将该压力电信号用于测量静态流体的液位。

在智能手机中的应用：压力传感器在智能手机中可用于海拔高度测量，智能手机通过传感器测量大气压计算出海拔高度；此外，压力传感器还可用于辅助GPS导航，通过测量所处的高度，解决有高度引起的GPS误差；最后，压力传感器可以配合加速度计、陀螺仪等技术实现精准的室内定位[1]。

在医疗行业中的应用：压力传感器在微创导管消融术和体温传感器测量中有着较多的应用，这些应用要求传感器必须外形小巧切能够耐受液体介质；此外，压力传感器还可以充当外科手术使用的一次性低成本导管。这些应用具有很大的潜力。

主要的数据分析方式

对压力传感器采集的数据通过波形分析进行处理：

波形分析是对波谱曲线的形态进行分析，提取表征曲线形态的参量，将这些参量的相似性作为波普相似性的测度。

常用的波形分析方法有中值滤波：对待处理的当前数据，选择一个窗口，该窗口为其邻近的若干个像素组成，对窗口的数据由小到大进行排序，再用窗口的中值来替代原数据的值[2]。优点：抑制效果很好，画面的清析度基本保持；缺点：对高斯噪声的抑制效果不是很好。

### 敏捷度与精度在压力传感器识别中的重要性

压力是日常生活和工业生产中许多应用的重要参数，正确测量和控制压力对保证生产工艺过程的安全性和经济性有重要意义，压力识别作为测量过程的关键组成部分，其敏捷度与精度对整个环节有着很大的影响。

为了适应压力传感器应用中更高的要求，需要找到具有更高精度与敏捷度的压力识别方式，并且对已有的数据采集方式进行优化。

目前，国内外常用的提高压力识别精度与敏捷度的方法有两种[3]，分别为硬件法和软件法。硬件法较为直观，但是存在电路复杂、精度较低、成本较高等缺点；软件法是将特定的算法应用于压力识别过程，克服了硬件法存在的缺点，其中，将支持向量机算法应用于压力识别过程在近几年得到广泛的关注，其适用于大量数据的分析，具有较好的应用前景。

# 第二章 压力传感器检测模块

## 2.1单片机构成

### 2.1.1整体结构

图1压力传感器监测模块的整体架构

### 2.1.2 压力传感器

压力传感器(Pressure Transducer)是能感受压力信号，并能按照一定的规律将压力信号转换成可用的输出的电信号的器件或装置。压力传感器通常由压力敏感元件和信号处理单元组成。按不同的测试压力类型，压力传感器可分为表压传感器、绝压传感器以及压差传感器。



图2 压力/拉力传感器

实验中使用传感器为双向采集，输出分辨率为0.125g/bit。压力配线方式如下图，拉力需要交换白、绿线。

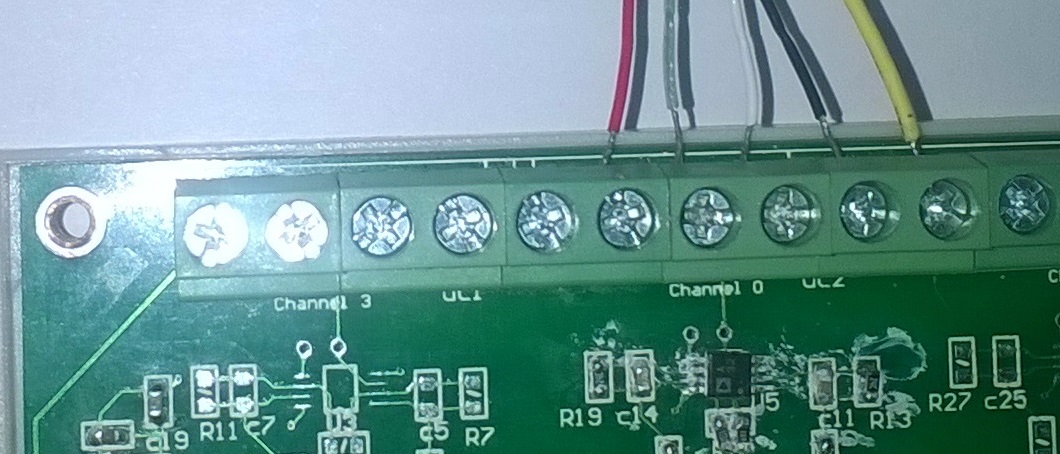


图3 压力传感器配线方式

### 2.1.3 数模转换和信号放大器

数模转换：数模转换是将离散的数字量转换为连续变化的模拟量，在很多数字系统中（例如计算机），信号以数字方式存储和传输，而数字模拟转换器可以将这样的信号转换为模拟信号，从而使得它们能够被外界（人或其他非数字系统）识别[4]。

数字模拟转换器以均匀时间间隔输出模拟值。其输入值以一定时序输入并锁存在转换器中，然后每完成一次转换，转换器的输出值都迅速从上一个输出值更新为当前锁存数值所对应的模拟信号。

信号放大器：信号放大器用来放大信号，又称微型直放站。信号放大器的主要功能有：

放大及滤波：前级仪表放大器的输出经缓冲进入次级放大器，该级为可编程增益放大器PGA，放大后的信号经滤波处理具有最平坦的频响特性

零电平校准：自校准解决方案实现模拟通道的零电平较准，消除不同增益和滤波状态下的零点误差

### 2.1.4 Cortex-M0+

Cortex是ARM公司研发的系列处理器，是一种超低功耗，低成本微控制器，支持ARMv6M 指令集，耗电量仅 9μA/MHz，在不牺牲功耗和面积的情况下，提供了更好的性能。

其主要特点为：

高性能：与高时钟频率相结合的快速处理能力

实时：处理能力在所有场合都符合硬实时限制

安全：具有高容错能力的可靠且可信的系统

经济实惠：可实现最佳性能、功耗和面积的功能。

在Cortex-M0的基础上，Cortex-M0+重新设计加入包括改良的调试和追踪能力、二阶流水线技术、优化闪存访问、单周期输入输出（IO）等多个重要特性，有更快的输入输出速度以及更少的每个指令所需试中周期数。能够二进制兼容已有的Cortex-M0处理器工具和实时系统（RTOS），并延续了其易用性以及C语言编程模型优势。

### 2.1.5 EEPROM

EEPROM (Electrically Erasable Programmable read only memory)，一种掉电后数据不丢失的带电可擦可编程只读存储芯片。EEPROM不需要借助其他设备即可擦除已有信息，重新编程。它通过电子信号来修改其内容，修改时以字节为最小单位，并且支持部分修改。

EEPROM有四种工作模式：读取模式、写入模式、擦除模式、校验模式[5]。

由于EEPROM的优秀性能，以及在在线操作的便利，它被广泛用于需要经常擦除的BIOS芯片以及闪存芯片，并逐步替代部分有断电保留需要的RAM芯片，甚至取代部分的硬盘功能（见固态硬盘）。它与高速RAM成为当前（21世纪00年代）最常用且发展最快的两种存储技术[6]。

## 2.2压力传感器监测模块整体参数

MCU总线频率：48M；

ADC采集速度：1.2M；

ADC采集精度：0.01~0.06%

ADC采集频率：1~30K(ZL-30K-01=6k)；

UART波特率：1.5M；

UART传输频率：1~30K(ZL-30K-01=6k)；

交流相应时间：10~300uS(ZL-30K-01=60uS)；

ADC通道：4通道(ZL-30K-01：adc1有效)；

输出通道：4通道(ZL-30K-01：ac1有效)；

USB数据传输通道：1通道；

输出通道启动阈值：ADC采集量的25~75%，通过USB软件设置；

输出通道停止阈值：ADC采集量的25~75%，通过USB软件设置；

输入输出关系：通过USB软件设置输入对应输出，支持一路为启动阈值，另外一路为停止阈值；

波特率：在信息传输通道中，携带数据信息的信号单元叫码元，每秒钟通过信道传输的码元数称为码元传输速率，简称波特率。波特率是指数据信号对载波的调制速率，它用单位时间内载波调制状态改变的次数来表示(也就是每秒调制的符号数)，其单位是波特（Baud,symbol/s）。波特率是传输通道频宽的指标。

采集精度：样本大小是用每个样本的位数bit/s(即bps)表示的，它反映度量波形幅度的精度。

6K传输频率：更高的传输频率意味着更高的数据采集精度与灵敏度。

## 2.3 监测模块的启动停止指令

启动测试指令：用于启动监测模块，通过传感器采集数据。采用十六进制输入。具体指令为：EF 71 3B 4B 81 01 02 03 04 05 06 07 99 99 6F FE

停止测试指令：用于关闭监测模块，停止数据采集。采用十六进制输入。具体指令为：EF 71 3B 4B D5 01 02 03 04 05 06 07 99 99 3B FE

## 2.4 其他功能部件

### 2.4.1 单片机电源

15v+、15v-接入15~24dcV，产品与电源位轨对轨输出，采集精度对应电源纹波系数。轨对轨指器件的输入输出电压范围可以达到电源电压。它的输入或输出电压幅度即使达到电源电压的上下限，此时放大器也不会像常规运放那样发生饱和与翻转。大大增加了放大器的动态范 围。这在低电源供电的电路中尤其具有实际意义。

### 2.4.2 单片机USB连接线

监测模块采用串口连接。串口连接可以每秒钟接受1.5M数据，满足监测的要求。串口连接时，直接连接单片机和电脑。

### 2.4.3 输出端

输出端采用外置过流保护装置，输出驱动能力220Vac\*200mA。

过流保护：很多电子设备都有个额定电流，不允许超过额定电流，不然会烧坏设备。所以有些设备就做了电流保护模块。当电流超过设定电流时候，设备自动断电，以保护设备。如主板CPU的USB接口一般有USB过流保护，保护主板不被烧坏[7]。

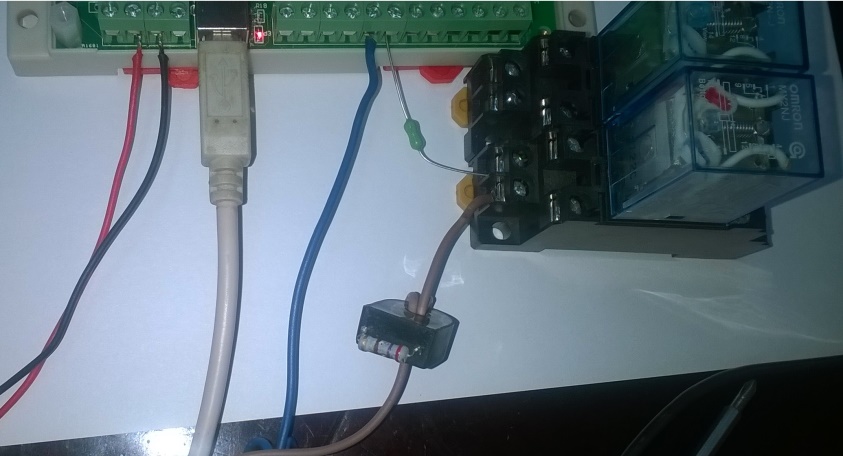


图4 外置过流保护装置

# 

# 第三章 支持向量机

## 3.1简介

支持向量机(Support Vector Machine,SVM)发表于1995年，对于文本分类任务有很好的表现，现已成为机器学习的重要组成部分[8]，并曾经在2000年前后掀起统计学习热潮。但实际上，支持向量的概念早在20世纪60年代就出现了，统计学系理论早在70年代便已经成型。对核函数的研究则更早，但在统计学习兴起后，核技巧才真正成为机器学习的通用基本技术。

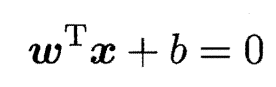
支持向量机基于统计学理论的VC维理论和结构风险最小化原理，被广泛应用于各个领域。通过将向量映射到高维空间，可以有效地解决非线性分类问题。SVM在模式识别、函数逼近、非线性分类等应用中有非常好的表现，近年来被广泛的研究，是机器学习技术发展的重要推动力。

## 3.2 支持向量机基本理论

## 3.2.1 间隔与支持向量

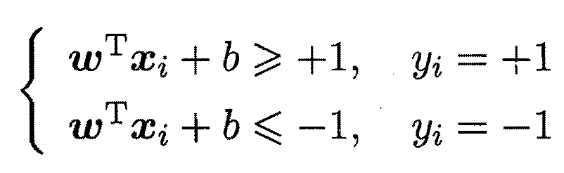
为了将不同类别的样本分开，分类在训练集的基础上寻找一个划分招平面。

在样本空间中，通过如下线性方程来描述划分超平面：

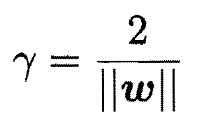
 (1)

其中，超平面方向由决定；超平面与原点间的距离由b决定。通常将超平面标记为。

若超平面可以将训练样本正确的分类，也就是对样本空间中的(xi,yi)，若yi=+1，则;若yi=-1，则[3]。令

 (2)

支持向量(support vector)即为距离超平面最近的几个样本点，它们使上式中的等号成立，异类支持向量到划分超平面的距离和被称为“间隔”(margin)，可用下式表示

 (3)

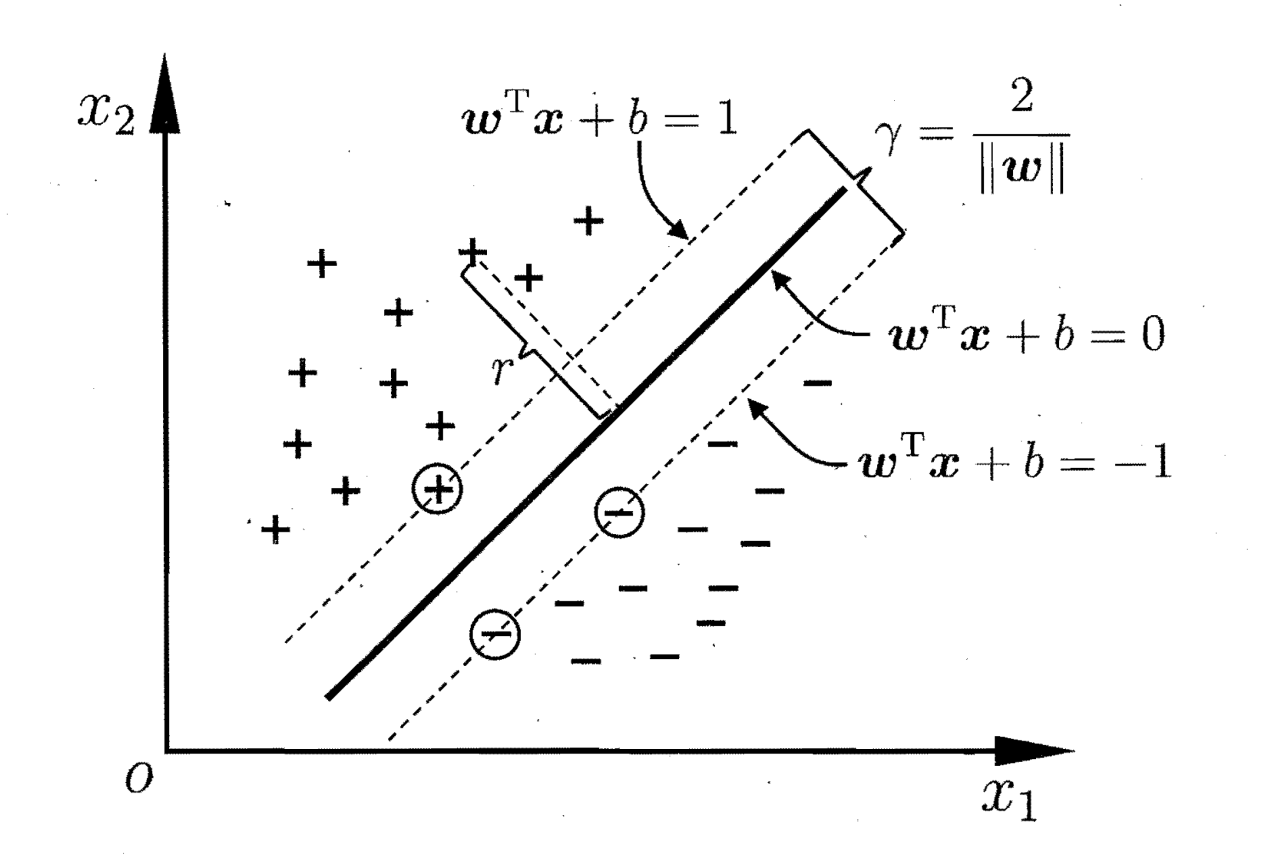
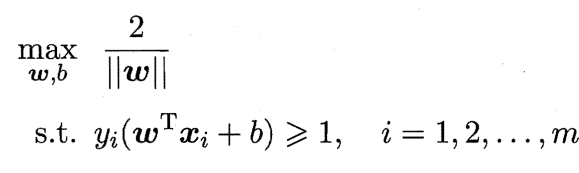
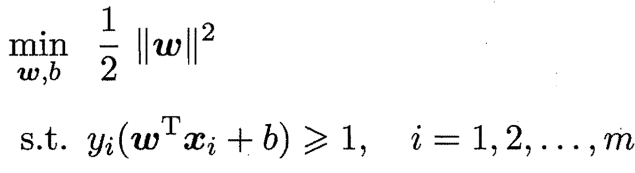


图5 支持向量与间隔

对于参数w和b，如果可以满足(2)式，并且可以(3)最大，那么它们对应的超平面就是具有“最大间隔”(maximum margin)的划分超平面。上述条件可表示为

 (4)

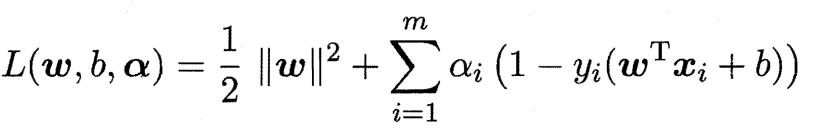
上式亦等价于

 (5)

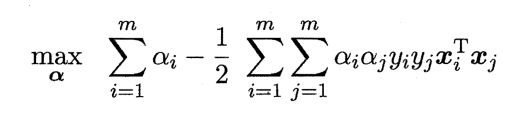
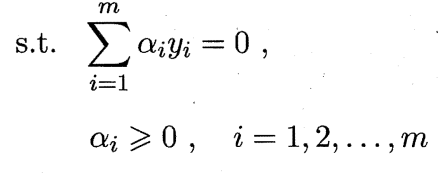
此即为支持向量机的基本型。

### 3.2.2 对偶问题

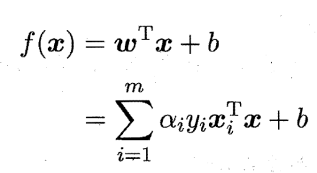
直接求解(5)式这个凸二次规划(convex quadratic programming)问题较为困难，为此，通常使用拉格朗日乘子法得到其“对偶问题”(dual problem)。添加拉格朗日乘子，对应于每一条约束条件，那么该问题的拉格朗日函数可以写成

 (6)

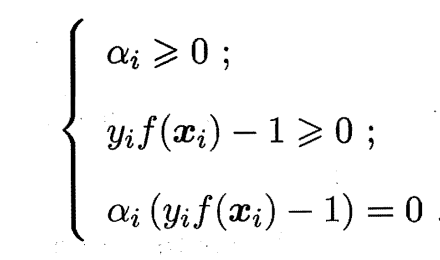
该式对ω和β的偏导数为零，带入该式可得原问题的对偶问题，即

 (7) 

解出后，求出与b即可得到

 (8)

使用SMO(Sequential Minimal Optimization)算法，由对偶问题解出，对应训练样本(),注意上式应满足KKT(Karush-Kuhn-Tucker)条件，即要求

 (9)

即，对任意样本，总有=0或=1，这表明，最终模型仅与支持向量有关。

## 3.2.3 核函数

可能无法在原是样本空间中找到一个划分超平面来正确划分两类样本。如果出现此类情况，可以将样本映射到高维特征空间，使样本在该空间内线性可分。若原始样本空间为有限维，则一定可以映射到一个高维空间使样本可分[10]。

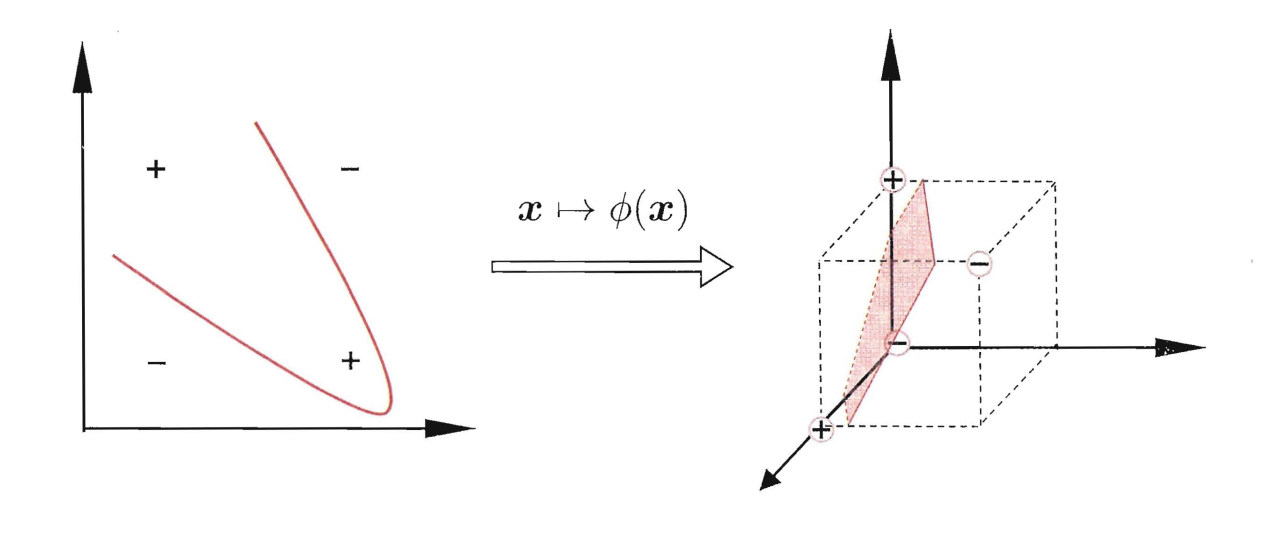
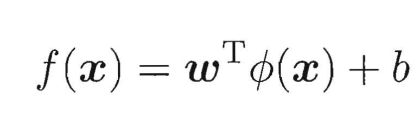
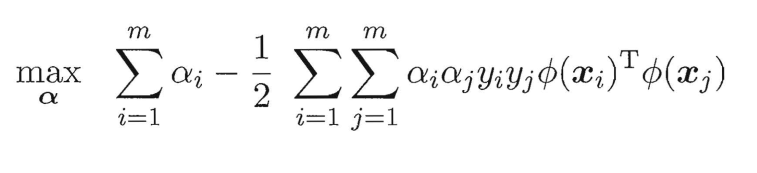
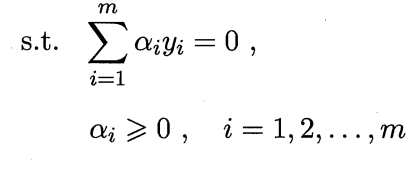


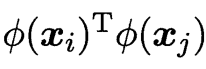
图6 异或问题与非线性映射

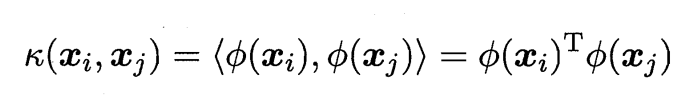
表示将x映射后的特征向量，映射到特征空间后，划分超平面对应的模型为

 (10)

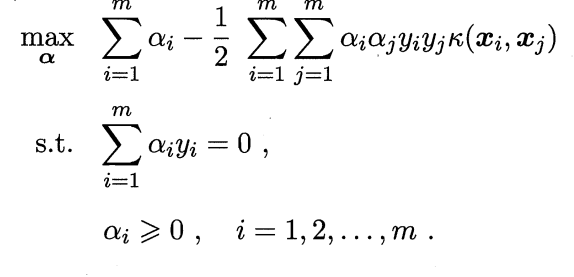
类似的，可得到

 (11) 

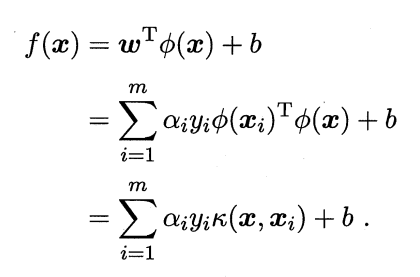
求解上式涉及到计算，此为与映射到特征空间后的内积，直接计算很困难，因此，令

 (12)

有了该函数，就不必直接计算样本映射到高维空间后的內积，通过求解

 (13)

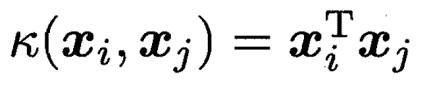
可得到“支持向量展式”

 (14)

由该式可知可通过训练样本的核函数展开模型最优解，式中的即为核函数。特征空间对支持向量机的性能好坏至关重要，因此支持向量机的最大变数在于“核函数选择”，核函数选择合适与否，直接决定了训练出的模型性能。

常用的核函数如下：

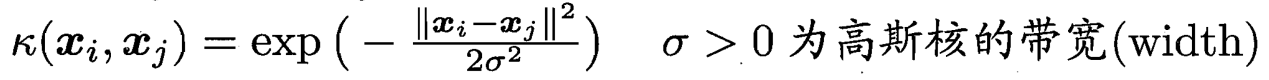
线性核函数：



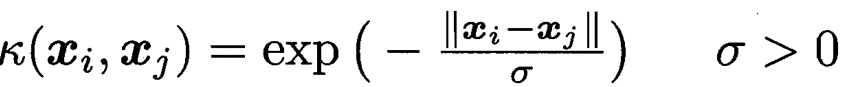
多项式核函数：



高斯核函数：



拉普拉斯核函数：



Sigmoid核函数：



此外，核函数的线性组合，直积仍为核函数。

## 3.2.4 软间隔与正则化

使得样本在特征空间中线性可分的核函数在现实任务中往往无法确定，恰好线性可分的结果也很有可能是由于过度拟合所致。

为此，引入“软间隔”(soft margin)的概念，允许支持向量机在一些样本上出错。

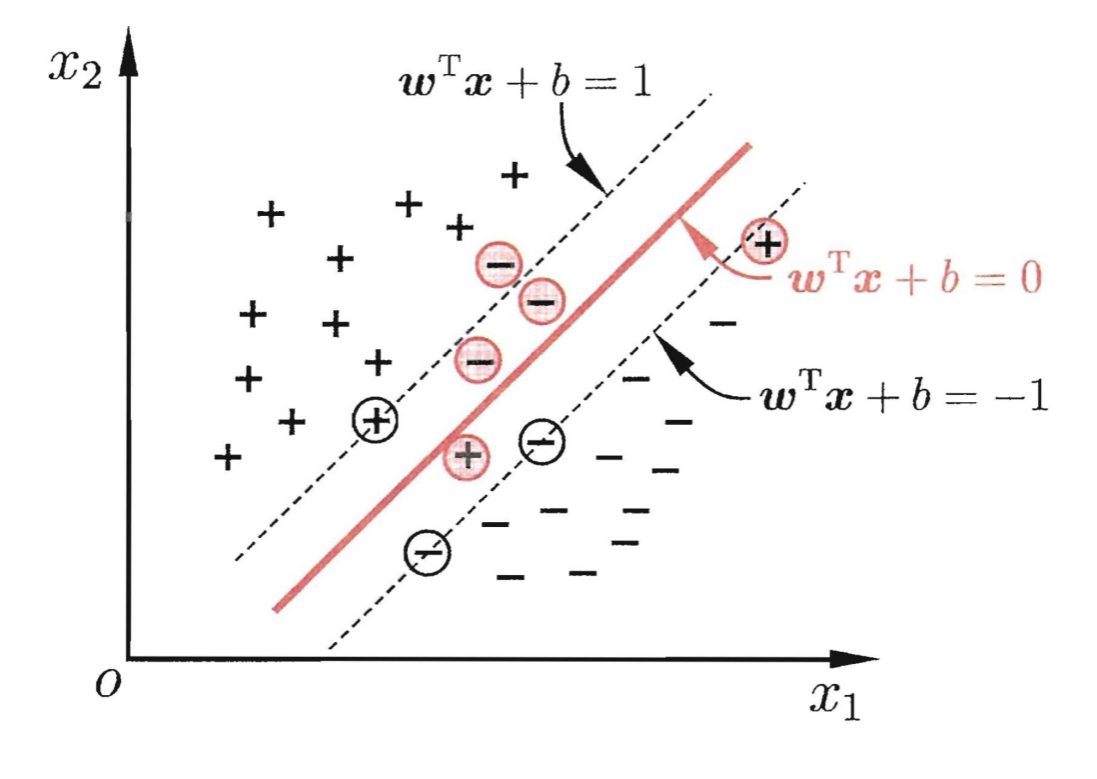
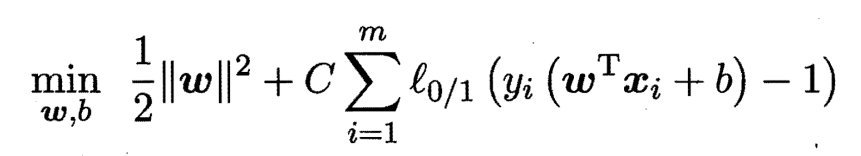
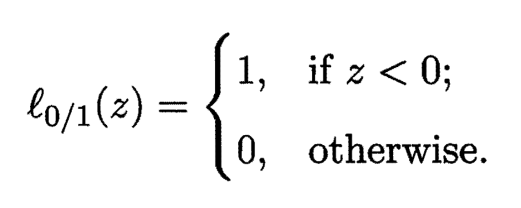


图7 软间隔示意图 红色圈出了不满足约束的样本

前面所说的支持向量机形式中所有的样本都满足约束，即为所有的样本全部划分正确，称为“硬间隔”(hard margin)，某些样本不满足约束也被允许的情况为软间隔，于是，优化目标可以写为

 (15)

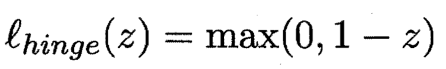
其中，C>0为常数，是“0/1损失函数”

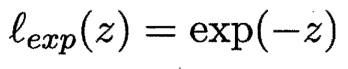
 (16)

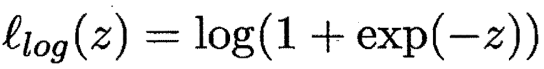
由上式可知，当C为无穷大时，所有样本均应满足约束；而当C为有限值时，允许一些样本不满足约束。

然而，非凸、非连续，数学性质不好，不易直接求解，通常用其他的“替代损失”(surrogate loss)函数代替。

常用的替代损失函数有：

hinge损失函数：；

指数损失函数(exponential loss)：

对率损失函数(logistic loss)：

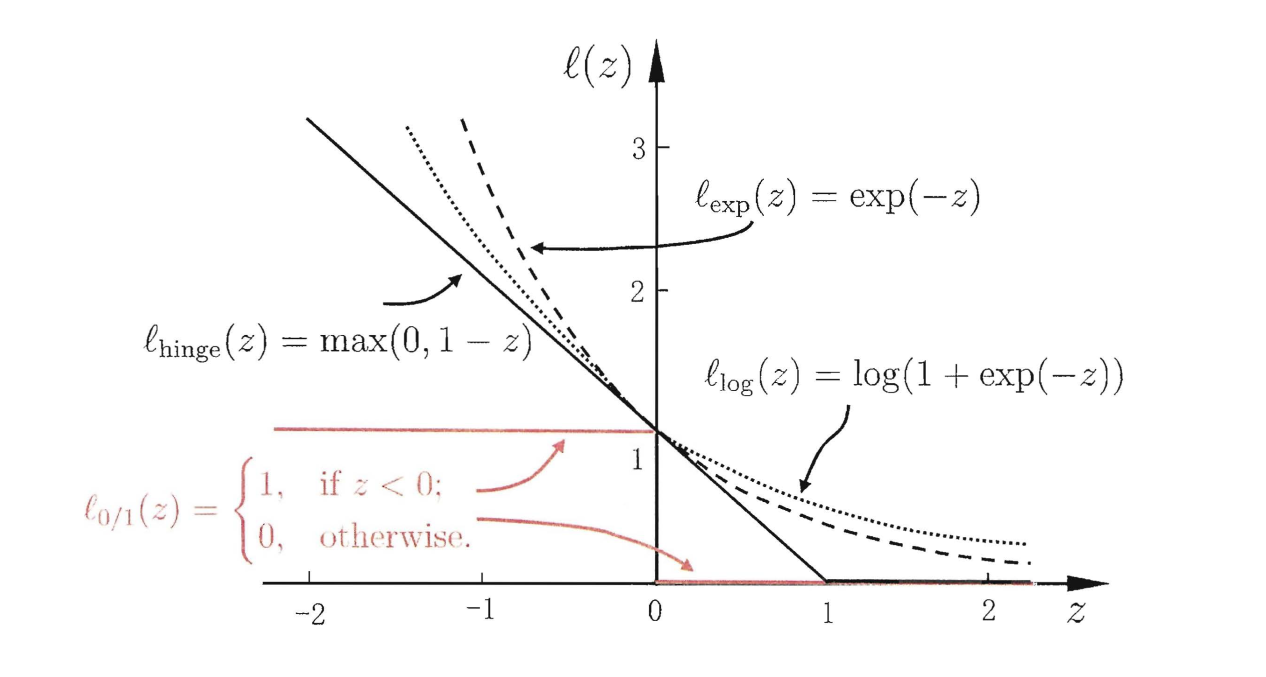
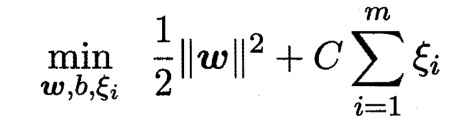
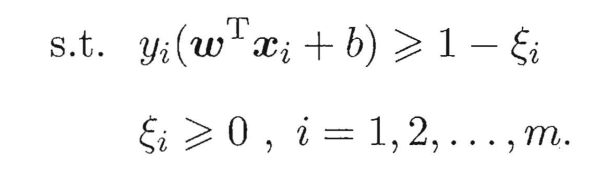


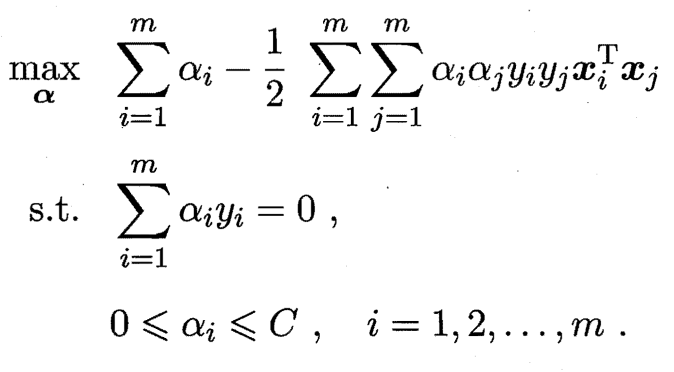
图8 三种常见的替代损失函数:hinge损失、指数损失、对率损失

若采用hinge损失， 并引入“松弛变量”(slack variables) ，可得：

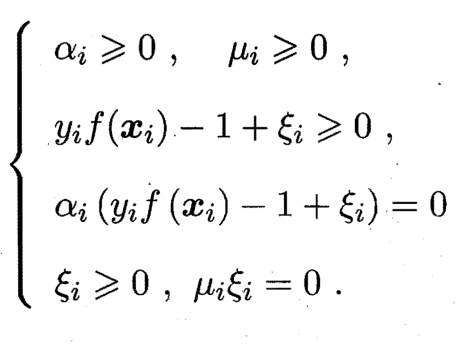
 (17)



此即为常用的“软间隔支持向量机”，这仍然是一个二次规划问题，通过拉格朗日乘子法得到其对应的拉格朗日函数。其对偶问题为

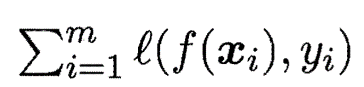
 (18)

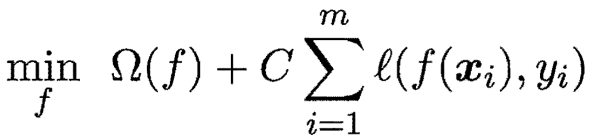
对于软间隔支持向量机，KKT条件要求：

 (19)

于是，对任意样本()，总有或。如果，那么该样本不会对有影响；如果，那么一定有，此时该样本为支持向量。由此可以看出，软间隔支持向量机的最终分类模型仅与支持向量有关，即采用hinge损失函数后仍保持了稀疏性。

如果使用对率损失函数代替hinge损失函数，则得到的模型接近于对率回归模型，这也表明，对率回归的优化目标与支持向量机相近，它们的性能通常情况下也相当。对率回归的优点在于它的输出具有概率意义，也就是在给出预测标记的同事也给出了概率，支持向量机的输出则不具概率意义。为了能用于多分类任务，支持向量机需要进行推广，而对率回归则不需要。此外，对率损失是光滑的单调递减函数，不能支持向量的概念，因此对率回归依赖于更多的训练样本，其预测花费更大，而采用hinge损失的SVM的解是稀疏的，因为hinge损失函数有一段平缓的零区域。

还可以用其他的替代损失函数代替0/1损失函数来得到其他的学习模型，这些模型的性质与使用的替代函数直接相关，它们的共性是：优化目标中的第一项描述的是划分超平面的“间隔”大小，称为“结构风险”(structural risk)，描述模型的性质，另一项描述训练样本上的误差，表述模型与训练样本的切合程度，称为“经验风险”(empirical risk)，结构风险与经验风险的折中依赖于C的选择，优化目标的一般形式可以写为

 (19)

优化目标中的第一项表达了我们希望得到何种性质的模型，这就为引入用户意图和领域知识提供了途径；第二项降低了最小化训练误差带来的过拟合风险，有助于削减假设空间。

## 3.3 LIBSVM

### 3.3.1简介

台湾大学林智仁(Lin Chih-Jen)教授及其团队设计开发的LIBSVM 是一个简单易用的SVM软件包。

LIBSVM是开源的，易于拓展，支持数十种语言版本，具有运用灵活，输入参数少，程序小等特点，在国内被广泛应用。

### 3.3.2 LIBSVM工具使用说明

LIBSVM的软件包主要有6个文件夹和一些c++源码文件。

JAVA——应用于JAVA平台；

MATLAB——应用于MATLAB平台；

Python——用于参数优选的工具；

svm-toy——用来展示训练数据和分类界面的可视化工具，里面是源码，在windows文件夹下有其编译后的程序；

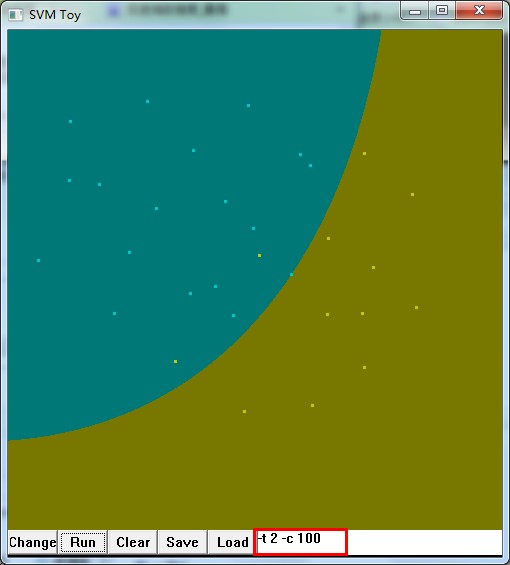
tools——包含进行参数优选（grid），数据检查(checkdata)，集成测试(easy)，数据集抽样(subset)的四个可执行文件；

windows——包含libSVM四个exe程序包。

此外还包括一些.h和.cpp文件,其中，最重要的是svm.h和svm.cpp文件，svm-predict.c、svm-scale.c和svm-train.c都是调用的这个文件中的接口函数。还有可以用记事本打开的样本文件heart\_scale，可以用于测试。

### 3.3.3 执行windows文件夹下的exe文件

**在windows平台下，打开svm-toy.exe文件，点击“Run”按钮，用鼠标随机在黑色区域的部分点下点，此即为第一类的数据分布情况；为了产生第二类数据的分布，点击“Change”，在黑色区域的另一部分点上点；点击“Run”按钮，该工具就能用不同的颜色区域来表示不同的类别，根据这两类数据分出两类。如下图所示：**



**图9 使用svm-toy.exe分类**

**第一类模式所在的区域是图中左上方绿色的区域，选择的第二类模式所在的的区域是右下方的黄色区域， SVM的最优分类面是两者的分界面。SVM在高维空间进行线性分类是通过核函数将原始数据映射到高维空间。这两类数据在高维空间是线性可分的，上图中所呈现的分类结果是将高维空间分类的结果又映射回原始空间后产生的，即：非线性的分类面。在上述界面的右下角，有一个编辑框，写着“-t 2 -c 100”，这是libSVM的参数，为了达到最优的分类效果，可以更改这些参数，来选择不同的核函数、不同的SVM类型等。**

**libSVM中包含以下可执行程序文件：**

**（1）svm-scale.exe：一个简易工具，用于对输入数据进行归一化的**

**（2）svm-toy.exe：一个带有图形界面的交互式小工具，用于SVM二分类功能演示；**

**（3）svm-train.exe：在用户输入的数据的基础上进行SVM训练。按照以下格式输入训练数据：**

**<类别号> <索引1>：<特征值1> <索引2>：<特征值2>...**

**（4）svm-predict.exe：使用已有的模型对输入数据进行分类。测试数据的格式与训练数据格式相同。**

### 3.3.4 在eclipse中建立工程执行

（1）准备训练样本和测试样本，本实验通过压力传感器采集数据，并使用上位机软件对数据进行处理。

（2）建立Java工程后导入libsvm.jar包，之后还需要导入在LibSVM的基础上进一步封装的java文件下的svm\_train.java和svm\_predict.java，为了方便API调用，它们把命令行参数转换成String[]类型的函数参数。

（3）在工程文件下或自定义数据存放目录存放训练数据和测试数据。

（4）调用LibSVM API分类代码需要编写如下JAVA代码：

****

（5）测试SVM分类性能时采用10折交叉验证法，运行程序得到分类准确率。

### 3.3.5 LIBSVM参数

（1）-s 默认值为0：设置SVM 类型，可选类型有：

0 -- C- SVC

1 -- nu - SVC

2 -- one-class-SVM

3 -- e - SVR

4 -- nu-SVR

（2）-t 默认值为2：设置核函数类型，可选类型有：

0 -- 线性核：u'\*v

1 -- 多项式核：(g\*u'\*v+ coef0)degree

2 -- RBF 核:exp(-||u-v||\*||u-v||/g\*g)

3 -- sigmoid 核：tanh(g\*u'\*v+ coef 0)

（3）-d默认值为3：核函数中的degree设置；

（4）-g r(gama)：核函数中的函数设置(默认1/ k);

（5）-r 默认值为0：设置核函数中的coef0；

（6）-c默认值为1：设置C- SVC、e - SVR、n - SVR中从惩罚系数C；[11]

（7）-n默认值0.5：设置nu - SVC、one-class-SVM 与nu - SVR 中参数nu ；

（8）-p默认值为0.1：核宽,设置e - SVR的损失函数中的e；

（9）-m 默认40：设置cache内存大小，以MB为单位：

（10）-e默认值为0.001：设置终止准则中的可容忍偏差；

（11）-h默认值为1：是否使用启发式，可选值为0 或1[12]；

（12）-b默认0：是否计算SVC或SVR的概率估计，可选值0 或1；

（13）-wi 默认值为1：对各类样本的惩罚系数C加权；

### 3.3.6核函数选择

通常而言，RBF核是四个核函数中的首选， RBF核能够处理属性和分类标注之间的非线性关系，因为它能将样本非线性地映射到一个更高维的空间。实际上，线性核是RBF核的一个特例[11]，某些情况下，RBF核与使用一个参数C的线性核具有相同的性能。同时，一定参数的RBF核表现与Sigmod核一致。多项式核比RBF核有更多的超参数（hyperparameter），而模型选择的复杂度受到超参数的数量影响，因此RBF核有更少的数值复杂度（numerical difficulties）。多项式核是相对于RBF核是高阶运算。此外，存在一些情况Sigmod核无法使用，当然，也存在一些情形RBF核是不适用的。而线性核在特征维数非常大时更加适用。

### 3.3.7 核参数的选择

RBF核有两个参数：C和γ。对于给定的问题，我们无法事先知道哪个C和γ是最佳的；为了分类器能够精确地预测未知数据，我们需要通过一定的方法确定合适的参数。如上讨论，通常的做法是将数据集合划分成两部分，即训练集和测试集[13]。分类器在未知数据集合上的性能效果可以通过在测试集上的效果体现。交叉验证(cross-validation)是这种做法的改进版本。为了预测整个样本集中的每个实例，在v折交叉验证(v-fold cross-validation)中，样本集合被划分成相同大小的v个子集。然后将其中v-1个子集作为训练集训练分类器，剩下的一个子集作为测试集[14]。通过交叉验证的表现选择合适的参数C和γ。

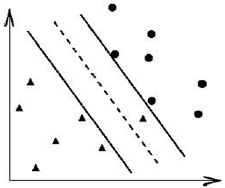
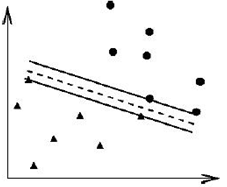


图10 一个过拟合分类器 图11 不存在过拟合现象的分类器

过拟合(overfitting)问题可以通过交叉验证法避免。为了说明这个问题，列举如图1所示的二分类问题。实心圆和三角是训练数据，而空心圆和三角是测试数据。在图10中的分类器存在对训练数据过拟合现象。考虑将图10中的训练和测试数据分别作为交叉验证的训练和校验集合，显然这个准确率是不好的。另一方面，在图11中的分类器则没有对训练数据过拟合，从而能够得到更好的测试准确率和交叉验证的准确率。

为了更加快速地确定参数C和γ时，我们使用一种“网格搜索”(grid-search)的方法，选择一个能够得到最高的交叉验证准确率的参数值对(C,γ)。虽然有其他更高级的方法可以节省计算成本，例如近似逼近交叉验证率（approximating the cross-validation rate），然而，我们更倾向于使用这个简单的网格搜索方法。原因有两个，一方面，没有详细的参数搜索的方法不安全，这些高级方法往往是近似法或者启发式；另一方面，这些高级方法在只有两个参数需要确定的情况下并不会比网格搜索节省很多时间。进一步而言，并行化在网格搜索中更容易实现（因为每个(C,γ)是独立的），而不容易在许多高级方法中实现，因为这些方法往往是一个迭代过程（例如walking along a path）。为了进一步提高网格搜索的速度，我们首先使用一个粗糙的网格。在确定网格中一个“更好”区域后，可以在这个区域中执行更好的网格搜索。在对数据集缩放之后，我们先使用一个粗糙的网格，找到当前最好的(C,γ)。然后，在当前最好的(C,γ)临近区域进行了一次更好的网格搜索，获得了一个更好的交叉验证的正确率。在最好的(C,γ)找到以后，可以在整个训练集上再次训练产出最终的分类器。

# 第四章 上位机软件

## 4.1 简介

发出操作控制命令的计算机称为上位机，实验中采用pc作为上位机；可以控制设备获取设备状况的计算机称为下位机，实验中采用单片机(single chip)作为下位机[16]。下位机接收到来自上位机的命令后，为了控制相应的设备，根据上位机发出的命令形成相应的时序信号。此外，设备状态数据由下位机读取，并将其转换为数字信号，由上位机接受该信号。可以将上位机看作控制者和服务提供者，被控制者和被服务者则是下位机。上位机软件用来与下位机进行通信，安装于上位机中，可以控制下位机或者读取下位机的数据。

## 4.2 上位机软件实现

### 4.2.1 获取端口名的方法

//获取端口名

public static string[] ActivePorts()

{

ArrayList activePorts = new ArrayList();

foreach (string pname in SerialPort.GetPortNames())

{

activePorts.Add(Convert.ToInt32(pname.Substring(3)));

}

activePorts.Sort();

string[] mystr = new string[activePorts.Count];

int i = 0;

foreach (int num in activePorts)

{

mystr[i++] = "COM" + num.ToString();

}

return mystr;

}

### 4.2.2 16进制字符串和byte字符数组之间的相互转换

//16进制字符串转换为byte字符数组

public static Byte[] \_16strToHex(string strValues)

{

string[] hexValuesSplit = strValues.Split(' ');

Byte[] hexValues = new Byte[hexValuesSplit.Length];

Console.WriteLine(hexValuesSplit.Length);

for (int i = 0; i < hexValuesSplit.Length; i++)

{

hexValues[i] = Convert.ToByte(hexValuesSplit[i], 16);

}

return hexValues;

}

//byte数组以16进制形式转字符串

public static string ByteTo16Str(byte[] bytes)

{

string recData = null;

//创建接收数据的字符串

foreach (byte outByte in bytes)//将字节数组以16进制形式遍历到一个字符串内

{

recData += outByte.ToString("X2") + " ";

}

return recData;

}

//16进制字符串转换字符串

public static string \_16strToStr(string \_16str)

{

string outStr = null;

byte[]streamByte = \_16strToHex(\_16str);

outStr = Encoding.Default.GetString(streamByte);

return outStr;

}

### 4.2.3 主窗体设置，声明变量以及接口

//声明变量

SerialPort sp = new SerialPort();

bool isSetProperty = false;//串口属性设置标志位

private enum PortState//声明接口显示状态，枚举型

{

打开,

关闭

}

### 4.2.4 声明配置文件路径，以及接受数据的字符串

//声明配置文件路径

string path = AppDomain.CurrentDomain.BaseDirectory + "confing.ini";

string tbSendDataStr = "";//发送窗口字符串存储

string tbSendData16 = "";//发送窗口16进制存储

List<byte> receivedDatas = new List<byte>();//接收数据泛型数组

### 4.2.5 接受和发送串口数据

//接收串口数据，代码中，使用到了byte数组，并且单独设计一个函数进行缓冲区数据的清理。

private void sp\_DataReceived(object sender, SerialDataReceivedEventArgs e)

{

byte[] ReceivedData = new byte[sp.BytesToRead];

//创建接收字节数组

sp.Read(ReceivedData, 0, ReceivedData.Length);

//读取所接收到的数据

receivedDatas.AddRange(ReceivedData);

tbxRecv.Text = (Convert.ToInt32(tbxRecv.Text) + ReceivedData.Length).ToString();

if (cb16Display.Checked)

tbReceivedData.Text = Methods.ByteTo16Str(receivedDatas.ToArray());

else

tbReceivedData.Text = Encoding.Default.GetString(receivedDatas.ToArray());

sp.DiscardInBuffer();//丢弃接收缓冲区数据

}

//发送串口数据，在发送数据的时候需要检查是否按照16进制进行发送，判定发送的格式，进行不同的处理。

private void DataSend()

{

try

{

if(cb16Send.Checked)

{

byte[] hexBytes = Methods.\_16strToHex(tbSendData16);

sp.Write(hexBytes, 0, hexBytes.Length);

tbxSend.Text = (Convert.ToInt32(tbxSend.Text) + hexBytes.Length).ToString();

}

else

{

sp.WriteLine(tbSendDataStr);

tbxSend.Text = (Convert.ToInt32(tbxSend.Text) + tbSendDataStr.Length).ToString();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

return;

}

}

### 4.2.6 设置串口属性，定义不同的数据位、停止位、奇偶校验位以及波特率等默认数据

//设置串口属性

private void SetPortProperty()

{

sp.PortName = cbxCOMPort.Text.Trim(); //设置串口名

sp.BaudRate = Convert.ToInt32(cbxBaudRate.Text.Trim());//设置波特率

switch (cbxStopBits.Text.Trim())

{

case "1": sp.StopBits = StopBits.One; break;

case "2": sp.StopBits = StopBits.Two; break;

case "1.5": sp.StopBits = StopBits.OnePointFive; break;

default: sp.StopBits = StopBits.None; break;

}

sp.DataBits = Convert.ToInt32(cbxDataBits.Text.Trim());

switch (cbxParity.Text.Trim())

{

case "无": sp.Parity = Parity.None; break;

case "奇校验": sp.Parity = Parity.Odd; break;

case "偶校验": sp.Parity = Parity.Even; break;

default: sp.Parity = Parity.None; break;

}

sp.ReadTimeout = 5000;

Control.CheckForIllegalCrossThreadCalls = false;

sp.DataReceived += new SerialDataReceivedEventHandler(sp\_DataReceived);

}

//对端口数据进行赋值，给定初值

//加载配置文件

Hashtable ht = new Hashtable();

if (File.Exists(path))

{

try

{

string myline = "";

string[] str = new string[2];

using (StreamReader sr = new StreamReader(path))

{

myline = sr.ReadLine();

while (myline != null)

{

str = myline.Split('=');

ht.Add(str[0], str[1]);

myline = sr.ReadLine();

}

}

}

catch (Exception ex) { MessageBox.Show(ex.Message.ToString()); }

}

string[] ports = SerialPort.GetPortNames();

cbxCOMPort.Items.AddRange(ports);

cbxCOMPort.SelectedItem = cbxCOMPort.Items[0];

this.MaximumSize = this.Size;

this.MinimumSize = this.Size;

this.MaximizeBox = false;

#region 列出常用的波特率

cbxBaudRate.Items.Add("1500000");

cbxBaudRate.Items.Add("2400");

cbxBaudRate.Items.Add("4800");

cbxBaudRate.Items.Add("9600");

cbxBaudRate.Items.Add("19200");

cbxBaudRate.Items.Add("38400");

cbxBaudRate.Items.Add("43000");

cbxBaudRate.Items.Add("56000");

cbxBaudRate.Items.Add("57600");

cbxBaudRate.Items.Add("115200");

if (ht.ContainsKey("cbxBaudRate"))

cbxBaudRate.SelectedIndex = cbxBaudRate.Items.IndexOf(ht["cbxBaudRate"].ToString());

else

cbxBaudRate.SelectedIndex = 0;

cbxBaudRate.DropDownStyle = ComboBoxStyle.DropDownList;

#endregion

#region 列出停止位

cbxStopBits.Items.Add("1");

cbxStopBits.Items.Add("1.5");

cbxStopBits.Items.Add("2");

if (ht.ContainsKey("cbxStopBits"))

cbxStopBits.SelectedIndex = cbxStopBits.Items.IndexOf(ht["cbxStopBits"].ToString());

else

cbxStopBits.SelectedIndex = 0;

cbxStopBits.DropDownStyle = ComboBoxStyle.DropDownList;

#endregion

#region 列出数据位

cbxDataBits.Items.Add("8");

cbxDataBits.Items.Add("7");

cbxDataBits.Items.Add("6");

cbxDataBits.Items.Add("5");

if (ht.ContainsKey("cbxDataBits"))

cbxDataBits.SelectedIndex = cbxDataBits.Items.IndexOf(ht["cbxDataBits"].ToString());

else

cbxDataBits.SelectedIndex = 0;

cbxDataBits.DropDownStyle = ComboBoxStyle.DropDownList;

#endregion

#region 列出奇偶校验位

cbxParity.Items.Add("无");

cbxParity.Items.Add("奇校验");

cbxParity.Items.Add("偶校验");

if (ht.ContainsKey("cbxParity"))

cbxParity.SelectedIndex = cbxParity.Items.IndexOf(ht["cbxParity"].ToString());

else

cbxParity.SelectedIndex = 0;

cbxParity.DropDownStyle = ComboBoxStyle.DropDownList;

#endregion

#region COM口重新加载

/\*

cbxCOMPort.Items.Clear();//清除当前串口号中的所有串口名称

cbxCOMPort.Items.AddRange(Methods.ActivePorts());

if (ht.ContainsKey("cbxCOMPort") && cbxCOMPort.Items.Contains(ht["cbxCOMPort"].ToString()))

cbxCOMPort.SelectedIndex = cbxCOMPort.Items.IndexOf(ht["cbxCOMPort"].ToString());

else

cbxCOMPort.SelectedIndex = 0; cbxCOMPort.DropDownStyle = ComboBoxStyle.DropDownList;

\*/

#endregion

#region 初始化计数器

tbxSend.Text = "0";

tbxSend.ReadOnly = true;

tbxRecv.Text = "0";

tbxRecv.ReadOnly = true;

#endregion

# region 初始化当前时间

toolStripStatusLabel3.Text = DateTime.Now.ToString();

#endregion

#region 初始化串口状态

toolStripStatusLabel1.ForeColor = Color.Blue;

if (!isSetProperty)//串口未设置则设置串口属性

{

SetPortProperty();

isSetProperty = true;

}

try

{

sp.Open();

btnOpenCOM.Text = "关闭串口";

DisplayPortState(PortState.打开);

}

catch (Exception)

{ //串口打开失败后，串口属性设置标志位设为false

isSetProperty = false;

MessageBox.Show("串口无效或已被占用!", "错误提示");

}

#endregion

### 4.2.7设置端口的显示状态，将从用户设置的参数显示在界面上

//设置端口显示状态

private void DisplayPortState(PortState portstate)

{

toolStripStatusLabel1.Text = cbxCOMPort.Text + "端口处于" +

portstate + "状态，波特率：" + cbxBaudRate.Text + "数据位：" + cbxDataBits.Text +

"停止位：" + cbxStopBits.Text + "校验位：" + cbxParity.Text;

}

重新打开串口，读取数据。

//重新打开串口

private void ReOpenPort()

{

if (sp.IsOpen)

{

try

{

sp.Close();

SetPortProperty();

isSetProperty = true;

sp.Open();

}

catch (Exception)

{

isSetProperty = false;

btnOpenCOM.Text = "打开串口";

DisplayPortState(PortState.关闭);

MessageBox.Show("串口无效或已被占用", "错误提示");

return;

}

DisplayPortState(PortState.打开);

}

else

DisplayPortState(PortState.关闭);

}

### 4.2.8 应用程序的入口点，开始运行程序，调用多线程进行开发

static class Program

{

/// <summary>

/// 应用程序的主入口点。

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

程序运行界面：



图12 程序运行界面

一般常见的单片机，波特率一般设定为9600，如果波特率设定的不准确，无法得到相应的返回数据，如果波特率设定太高，对单片机和整个系统都会带来危害。该压力传感器检测模块，在设计中，实现了6k/s的传输速度，同时波特率的设计规格达到了1.5mBd，所以在设定初始波特率的时候设定为1500000Bd，设定方式如下图所示：

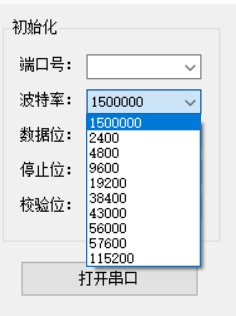


图13 设置波特率

# 

# 第五章 结论

# 5.1通过传感器与单片机采集数据

我们的关注区域应该具有如下特征:压力值在段时间内大幅度变化；最大压力超过150N；在压力值回落后有较长时间波动。

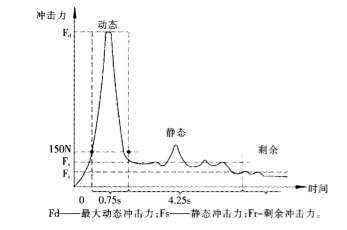


图14 关注的压力波形区域

## 5.1.1 设定滑动窗口

滑动窗口由两个关键变量构成：窗口大小（window size）和滑动步长（step）

假设传感器的采样频率是f HZ，那么一个窗口大小通常为2f，步长为f，

通常窗口大小设定为

## 5.1.2候选窗口特征

一个候选窗口中需要采集一个采集点上的数据在时域、频域以及在形状上的共14个特征值。其中，波形数据在时域上的特征有：

a.均值

b.标准差（体现数据的离散程度）

c.众数（体现数据高频重复的次数和位置）

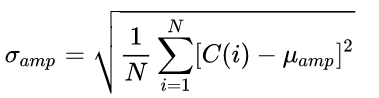
d.最大值max

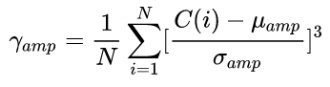
e.最小值min

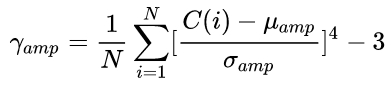
f.范围：

采集到的波形数据在频域上的特征有：

a.均值

b.标准差

c.偏度

d.峰度

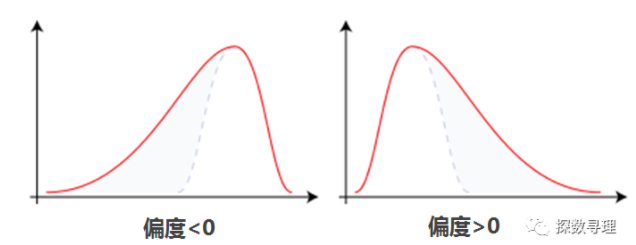
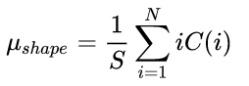
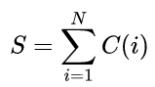
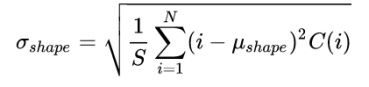
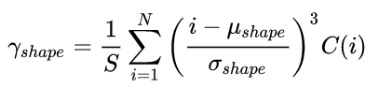


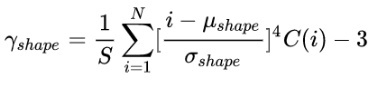
图15 不同的偏度值对应的波形图像

形状上的特征统计项有：

a.均值，

b.标准差

c.偏度

d.峰度

## 5.2 向量设计

将提前采集好的数据分组，每个滑动窗口包含七个数据采集点，以时间作为横坐标，压力值作为纵坐标。

每个采集点的数据记为，并且将每个滑动窗口的中的数据值表示为,,,,,,，以为中心，计算关于对称的两点压力值随时间的变化率，分别记为，，。

每个向量有12个属性，分别为连续七个数据点的偏度值，连续七个点的峰度，以及，，，，，，，，，。

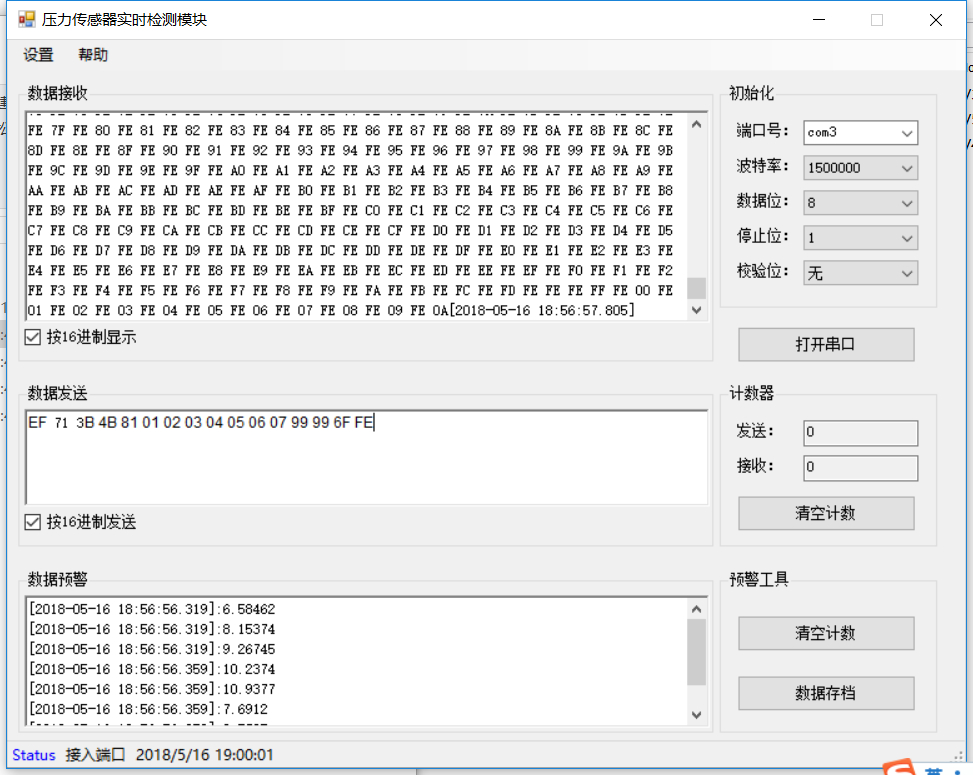
5.3 Training Set与Testing Set数据集

在实验中，为了得到训练SVM预测模型的training set集合，采用标准砝码进行测试。具体方法是，将4KG的标准砝码放在压力传感器上，时长5秒，采集到约30k个数据。对这5秒内的数据加上标签“+1”，表示该向量为真值。然后，取下砝码，进行空载测试，时长5秒，采集到约30k个数据。对这5秒空载得到的数据加上标签“-1”，表示该向量为假值。“+1”和“-1”标签的向量按照1:1的比例进行组合，得到training set集合。Training set的示例如表X所示。

表X：Training Set集合示例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 标签 | 时间 | 4KG砝码称重值 | 标签 | 时间 | 空载称重值 |
| +1 | 18:56:25.000 | 4.66536 | -1 | 18:56:25.018 | 0.056766 |
| +1 | 18:56:25.001 | 3.733625 | -1 | 18:56:25.019 | 0.08994 |
| +1 | 18:56:25.002 | 4.11265 | -1 | 18:56:25.020 | 0.069668 |
| +1 | 18:56:25.003 | 3.745965 | -1 | 18:56:25.021 | 0.11779 |
| +1 | 18:56:25.004 | 3.635855 | -1 | 18:56:25.022 | 0.08092 |
| +1 | 18:56:25.005 | 3.906375 | -1 | 18:56:25.023 | 0.079083 |
| +1 | 18:56:25.006 | 3.462975 | -1 | 18:56:25.024 | 0.08669 |
| +1 | 18:56:25.007 | 4.328855 | -1 | 18:56:25.025 | 0.027983 |
| +1 | 18:56:25.008 | 3.331805 | -1 | 18:56:25.026 | 0.05981 |
| +1 | 18:56:25.009 | 3.521045 | -1 | 18:56:25.027 | 0.127152 |
| +1 | 18:56:25.010 | 3.57362 | -1 | 18:56:25.028 | 0.023906 |
| +1 | 18:56:25.011 | 3.60018 | -1 | 18:56:25.029 | 0.08503 |
| +1 | 18:56:25.012 | 3.519705 | -1 | 18:56:25.030 | 0.021304 |
| +1 | 18:56:25.013 | 3.63961 | -1 | 18:56:25.031 | 0.073182 |
| +1 | 18:56:25.014 | 3.61453 | -1 | 18:56:25.032 | 0.072404 |
| +1 | 18:56:25.015 | 3.97424 | -1 | 18:56:25.033 | 0.00464 |
| +1 | 18:56:25.016 | 3.858225 | -1 | 18:56:25.034 | 0.088901 |
| +1 | 18:56:25.017 | 3.860505 | -1 | 18:56:25.035 | 0.144292 |

Testing set集合中的数据是采用10kg砝码随机放置在压力传感器上进行采集，时长不超过10秒。得到约30k个数据。利用svm在training set中生成的预测模型来对testing set进行实时检测，如发现有波形的快速上升，即预测到波形由空载向负载的快速拉升，则进行数据报警，显示在软件的界面中，如图X所示。



图X：打开串口接收数据的同时，对数据进行实时监测，并在必要时发出数据预警。

# 

# 致 谢

感谢陈宇老师在毕业设计阶段给我的指导，从定题，到资料收集，论文写作、修改到定稿，老师多次询问我的研究进程，为我拓展思路，指点迷津。陈老师的精心点拨、热忱鼓励是我毕业设计过程中的最大动力。在毕业设计过程中，从老师那里学到的不仅仅是知识，更是踏踏实实、严谨求实的作风，这些都是使我受益无穷的宝贵财富。

白驹过隙一般，四年的大学生活转眼间已经接近尾声。四年的求学生涯，我获益良多，走的辛苦却也收获满满。

感谢所有给予我指导的恩师，你们的教导带领我走进知识殿堂，让我能够站在更高的高度上去看待、思考问题，采用更有效的方法去解决问题；是你们用博大的胸怀，给予我无私的奉献，让我能够用全新的角度发现生活的美好。

感谢我的朋友、同学，你们是我的益友，亦是我的良师，你们陪我度过了大学生活中最美好的时光，也陪我走过了许多艰难的时刻。

感谢我的父母，从小到大的每一个脚印，都浸满这你们的无私关爱与谆谆诲，

大学四年，无论我面对怎样的困难，你们总是最坚定的站在我的身边，支持我坚定信念，不断向前。

在论文的写作过程中，我深感自己知识的不足。整个过程，虽然不易，但在思考和努力过后，却能让我戒除浮躁，提高自己。长风破浪会有时，直挂云帆济沧海，将此诗作为结尾，鼓励自己，坚持最初的梦想，永不言弃！

# 参考文献

[1]姜洋. 空气呼吸器与压力传感器的多种应用[J].《黑龙江科学》，2017年，08期

[2]解岩,安振庄.多功能数字信号发生器[J].电子测量技术,2004(05):64-65.

[3]于德鳌, 李慧, 李超,等. 基于单片机的音乐彩灯控制器设计[J]. 新校园旬刊, 2013(10):26-26.

[4]张秀再, 陈彭鑫, 吴华娟. 基于MEMS重力检测的数字水平仪[J]. 实验室研究与探索, 2016, 35(3):51-55.

[5]吴雷. 在线监测式EPS应急电源系统的研制[D]. 上海交通大学, 2007.

[6]赵宏石.基于SVM模型的工业企业技术创新能力成熟度评价初探[J].商,2016(22):146-147.

[7]刘磊,段俊奇,刘轶铭,翁丽娜.基于听觉特征的舰船辐射噪声分类研究[J].中国电子科学研究院学报,2016,11(04):401-406.

[8]凤志民,田丽,吴道林,李从飞.IGSA优化LSSVM的短期风电功率预测研究[J].可再生能源,2017,35(11):1699-1705.

[9]CSDN.LibSVM学习详细说明https://blog.csdn.net/github\_39261590/article/details/73648335

[10]彭俊伟. 基于静态和动态算法结合的公交到站时间预测[D].杭州电子科技大学,2016.

[11]席敏, 朱国魂, 姜茜. 基于小波变换与SVM的调制信号识别方法[J]. 桂林电子科技大学学报, 2014(4):274-278.

[12]黄丹,王志海,刘海洋.一种局部协同过滤的排名推荐算法[J].山东大学学报(工学版),2016,46(05):29-36.

[13]翁硕,薛松.基于BI-RADS描述的乳腺肿瘤判别[J].河南科技大学学报(医学版),2014,32(02):99-101.

[14]田子建,彭霞,苏波.基于机器视觉的煤岩界面识别研究[J].工矿自动化,2013,39(05):49-52.

[15]陈志文,王玮.基于Pt100铂热电阻的温度变送器设计与实现[J].现代电子技术,2010,33(08):197-199.