# 第9章 支持向量机技术及其在智能传感器系统中的应用

本章内容：

* 统计学习理论
* 支持向量机分类
* 支持向量机回归
* 基于支持向量机实现混合气体的定性识别
* 基于支持向量机实现传感器的非线性校正
* 利用Python实现支持向量机的方法

上一章主要介绍了神经网络技术及其在传感器测量数据拟合中的应用，包括人工神经网络的基础知识、BP神经网络等内容。本章将重点介绍支持向量机技术及其在智能传感器系统中的应用。主要介绍统计学习理论、支持向量机分类、支持向量机回归及其应用等内容，并详细介绍如何利用Python实现支持向量机。通过对这些技术的学习，可以更好地理解和掌握如何利用支持向量机实现智能传感器系统，并在实际应用中更好地运用这些技术。

# §9.1 支持向量机原理

传统统计学研究的是渐进理论，即当样本趋于无穷大时的极限特性。然而，在实际应用中，这一前提往往难以满足，导致一些理论上成熟的方法在实践中表现不佳。为了解决这个问题，统计学习理论应运而生，它是针对有限样本统计问题而建立的一套全新的理论体系。其中最核心的概念是VC维概念，它是描述函数集或学习机复杂性的重要指标。基于这一理论，研究人员得出了一系列关于统计学习的重要结论，包括一致性、收敛速度和泛化性能等方面。在此理论基础上，发展出了一种新的通用学习方法——支持向量机(Support Vector Machine, SVM)。该方法已经初步展现出优于现有方法的性能，并为机器学习理论和技术的进一步发展提供了动力。

## 9.1.1 统计学习理论

机器学习的目的是根据给定的训练样本求出对某系统输入输出之间依赖关系的估计，使它能够对未知输出做出尽可能准确的预测。那么，可以通过风险函数(Risk function，也可称为期望风险)对学习效果进行评估。

对个相互独立并服从同一分布的观测样本，，⋯，，在给定输入为的情况下输出为，且变量与之间存在未知依赖关系，即遵循某一未知的联合概率。因此，机器学习的期望风险可定义为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.1） |

式中：—任何函数集合，又称作预测函数集；

—函数的广义参数；

—用对进行预测而造成的损失，称为损失函数。不同类型的学习问题有不同形式的损失函数。

学习的目的就是为了使期望风险最小。由于可以利用的信息只有样本数据，因此无法直接计算期望风险并进行最小化。因此，实际应用中通常依据大数定理，利用算术平均值的最小值来代替期望风险的最小值，这就是所谓的经验风险最小化原则（Empirical Risk Minimization，ERM)。但是，用ERM准则代替期望风险最小化没有经过充分的理论证明，只是直观上看起来合理的做法。ERM准则不成功的一个例子就是神经网络的“过学习”问题。训练误差小，不一定能够导致好的预测效果。在某些情况下，过小的训练误差可能会降低模型的推广能力，从而导致真实风险增加。

显然，在有限样本下，经验风险最小并不一定意味着期望风险最小。学习机器的复杂性不仅应与所研究的系统有关，也应适应有限的样本数量。

统计学习理论就是研究小样本统计估计和预测的理论，核心内容包括：基于经验风险最小化准则的统计学习一致性条件；统计学习方法推广性的界；在推广界的基础上建立的小样本归纳推理准侧；实现新的准则的实际方法。其中，推广界是最有指导性的理论，其相关的核心概念是VC维（Vapnik-Chervonenkis Dimension）。

VC维的直观定义是：如果一个指示函数集能够将个样本按照所有可能的种形式分开，则称该函数集能够打散这个样本。函数集的VC维就是它能打散的样本数目的最大值。如果存在函数能够打散任意数量的样本，则该函数集的VC维被视为无穷大。通过设定一定的阈值，可以将有界实函数的VC维定义为指示函数。VC维反映了函数集的学习能力，VC维越大，学习机器就越复杂。目前尚没有通用的关于任意函数集VC维计算的理论，只知道一些特殊函数集的 VC 维。

对于两类问题，经验风险和实际风险之间以至少概率满足如下关系：

|  |  |  |
| --- | --- | --- |
|  |  | （9.2） |

式中：—函数集的VC维；

—样本数。

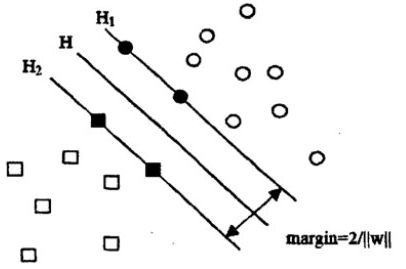
从理论上来说，学习机器的实际风险由两部分构成：一部分是经验风险（也就是训练误差），另一部分被称为置信范围，它与学习机器的VC维和样本数量有关。因此，可以将实际风险简洁地表示为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.3） |

这个公式表明，在有限样本的情况下，学习机器的VC维越高，置信范围越大，这可能会导致真实风险与经验风险之间的差距增大，从而导致过拟合的问题。在机器学习过程中，不仅要尽可能地减小经验风险，还应该尽量缩小置信范围，以获得较小的实际风险。这样的模型才能对未来的样本具有更好的泛化能力。支持向量机实际上就是这种思想的具体实现。接下来，将详细介绍支持向量机。

## 9.1.2 支持向量机分类

线性可分情况下的最优分类面如图9-1所示。

  
图9-1 线性可分情况下的最优分类面

在图9-1中，方形点和圆形点分别代表两类样本。H表示分类超平面，而H1和H2则是过两类样本中离H最近的点的平面，它们与H平行。H1和H2之间的距离被称为分类间隔（Margin）。最优分类面要保证正确地分开两类样本，并使分类间隔最大化。

设线性可分的样本集有个样本，其中，2，⋯，，，。在这里，和分别代表两类的类别标识。在高维空间中，将两类样本无错分开的分类超平面H满足：

|  |  |  |
| --- | --- | --- |
|  |  | （9.4） |

为了确保分类面能正确分类所有样本并具备分类间隔，需要满足以下约束条件：

|  |  |  |
| --- | --- | --- |
|  |  | （9.5） |

要使所有样本被无错误分开，应满足：

|  |  |  |
| --- | --- | --- |
|  |  | （9.6） |

在高维特征空间中，和之间的距离可以计算为。要使分类间隔最大，实际上就是要使

|  |  |  |
| --- | --- | --- |
|  |  | （9.7） |

最小。因此，求解最优分类面就是在满足条件式（9.6）的约束下，求式（9.7）中的极值。过超平面和的样本点就是（9.7）式得到的极值的样本点，它们共同支持了最优分类面，因此称为支持向量（Support Vector）。

最大化分类间隔实际上是为了提高模型的泛化能力，这是SVM的核心思想之一。根据统计学习理论，在维空间中，如果样本分布在一个半径为的超球范围内，那么满足条件的正则超平面构成的指示函数集 （为符号函数）的VC维满足下面的界：

|  |  |  |
| --- | --- | --- |
|  |  | （9.8） |

因此，使最小就是使VC维的上界最小，从而在结构风险最小原理（Structural Risk Minimization，SRM）准则中选择函数的复杂性。

通过使用拉格朗日优化方法，可以将上述最优分类面问题转化为其对偶问题。即，在约束条件：

|  |  |  |
| --- | --- | --- |
|  |  | （9.9） |

下对求解下列函数的最大值：

|  |  |  |
| --- | --- | --- |
|  |  | （9.10） |

为与每个样本对应的拉格朗日乘子。这是一个不等式约束下二次函数寻优的问题，存在唯一解。容易证明，在解中，只有一部分（通常是少部分）不为零，对应的样本就是支持向量。解上述问题后得到的最优分类函数是：

|  |  |  |
| --- | --- | --- |
|  |  | （9.11） |

式中的的求和实际上只对支持向量进行。是分类阈值，可以用任一个支持向量（满足（9.6）式中的等号）求得，或通过两类中任意一对支持向量取中值求得。

另外，可能存在一些样本不能被超平面正确分类，因此引入松弛变量

|  |  |  |
| --- | --- | --- |
|  | ，，，， | （9.12） |

显然，当分类出现错误时，大于零，是分类错误数量的一个上界。为此，引入错误惩罚系数。于是，构造广义最优分类超平面问题就转化为在约束条件

|  |  |  |
| --- | --- | --- |
|  | ，，，， | （9.13） |

下最小化函数

|  |  |  |
| --- | --- | --- |
|  |  | （9.14） |

的问题。

在式（9.14）中，错误惩罚系数为一个正值常数，越大，对错误的惩罚越重。其中第一项是样本到超平面的距离最大，从而提高泛化能力，第二项是使误差尽量小。广义最优分类面的对偶问题与线性可分情况下几乎完全相同，只是式（9.9）中第二个条件变为：

，，，，

对于线性不可分的样本，支持向量机的方法是将输入向量映射到一个高维的特征空间，并在该特征空间中构造最优分类面。它能够避免多层前向网络无法克服的一些缺陷。经过证明，我们得到了如下结论：如果选择适当的映射函数，大多数线性不可分的输入空间问题可以在特征空间中转化为线性可分问题。

然而，在将低维输入空间映射到高维特征空间时，由于空间维数急剧增加，在大多数情况下，直接计算特征空间中的最佳分类平面变得困难。为了解决这个问题，支持向量机通过定义核函数（Kernel Function），巧妙的将这一问题转化到输入空间进行计算。其具体机理如下：

在前面介绍的线性可分求解中都只涉及内积运算，因此可以假设有非线性映射：将输入空间的样本映射到高维特征空间中。在构造最优超平面时，训练算法仅使用特征空间中的点积运算。因此，如果能找到一个函数，使得，那么在高维空间中，实际上只需要进行内积运算，甚至不必知道变换的形式。

根据泛函的有关理论，只要一种核函数满足Mercer条件，它就对应某一变换空间中的内积。因此，通过在最优分类面中采用适当的内积函数， 可以实现某种非线性变换后的线性分类，而计算复杂度保持不变。此时目标函数（9.10）变为：

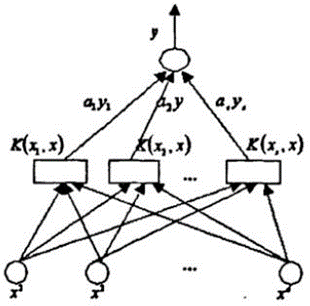
|  |  |  |
| --- | --- | --- |
|  |  | （9.15） |

相应的分类函数也变为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.16） |

算法的其他条件均不变。由于最终的判别函数中实际只包含与支持向量的内积求和，因此识别时的计算复杂度取决于支持向量的个数。

支持向量机的示意图如图9-2所示。

  
图9-2 支持向量机示意图

由图9-2可以看出，支持向量机求得的决策函数形式上类似于一个神经网络，其输出是若干中间层节点的线性组合，而每一个中间层节点对应于输入样本与一个支持向量的内积，因此也被称作是支持向量网络。

在式（9.16）中，满足Mercer条件的内积函数称为核函数。核函数在支持向量机中起到了关键作用，它们可以将数据映射到更高维的特征空间，从而使得原本线性不可分的问题变得线性可分。在支持向量机中，常见的核函数有线性核函数、多项式核函数、径向基函数和Sigmoid核函数。

（1）线性核函数(linear)

线性核函数是最简单的核函数，它直接使用原始特征空间中的内积来进行计算。其数学表达式为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.17） |

这种核函数没有引入额外的参数，计算简单，但只能处理线性可分的问题。

（2）多项式核函数(poly)

多项式核函数通过将数据映射到一个高维空间，引入了多项式的非线性关系。其数学表达式为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.18） |

式中：—核函数的阶数；

—核函数的系数；

—核函数的独立项。

这种核函数适用于一些非线性可分但不过于复杂的问题。

（3）径向基函数（rbf）

径向基函数是一种常用的核函数，它可以将数据映射到无限维的空间。其数学表达式为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.19） |

式中：—核函数的系数。

这种核函数可以处理更复杂的非线性问题

（4）Sigmoid核函数(sigmoid)

Sigmoid核函数将数据映射到一个高维空间，并且其输出是有界的。其数学表达式为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.20） |

式中：—核函数的系数；

—核函数的独立项。

这种核函数适用于一些需要将输出限制在某个范围内的问题。

选择核函数的关键在于对问题的理解和数据的特点。如果问题是线性可分的，线性核函数是一个简单且有效的选择。如果问题是非线性可分的，但不过于复杂，多项式核函数可以考虑。如果问题非常复杂且非线性，径向基函数是一个常用的选择。而如果需要将输出限制在某个范围内，可以考虑使用Sigmoid核函数。

## 9.1.3 支持向量机回归

回归分析又称为函数估计。它的目标是通过有限个样本，建立反映输出和输入之间的函数关系，进而用所得回归函数对新的数据作出计算或预测。这种函数关系通常是连续的。如果输出和输入之间的联系是非线性关系，称为非线性回归分析。

设有回归模型实现映射，使得以求逼近目标函数。所产生的误差为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.21） |

在回归分析中，首先需要考虑的是回归效果的估计，也是学习机最优化准则问题。为了在指定的函数类中选取“拟合最优”的函数，必须明确评优的标准，并对所有训练样本的累计误差最小。

计算累计误差的算法有很多，常用的误差函数如图9-3所示。

|  |  |  |
| --- | --- | --- |
| ：平方误差函数 | ：绝对值误差函数 | ：-不敏感误差函数 |

图9-3 常用误差函数

图9-3中，和为多元统计分析中常用的误差函数。通常只考虑误差值大小，正负是等价的，平方函数和绝对值函数都满足对原点的对称性。不难看出，常用的均方误差和均方根误差函数均属于平方误差函数一类。其特点是对于小误差不敏感，对大误差反映敏感。

在绝对值误差的基础上，Vapnik提出了“-不敏感损失函数”：

|  |  |  |
| --- | --- | --- |
|  |  | （9.22） |

如图9-3所示。当-不敏感损失函数中指定的参数时，等同于绝对值函数；时， 可以看作是把的两条射线分别向左右平移距离。其含义为：它定义了一个“拟合范围”，当估计器输出与期望输出误差绝对值小于时，误差可忽略不计，即为可忽略误差。当误差超过时，误差函数的值为实际误差减去的绝对值。或者说，这种误差函数中间有一个宽度为的不敏感带，称其为带或管道。

鉴于分类与函数逼近的内在联系，前面用于分类的SVM方法，也可类似地应用于回归。在回归中，要求最小化下式：

|  |  |  |
| --- | --- | --- |
|  |  | （9.23） |

其中为上文中介绍的-不敏感损失函数。此时，支持向量机的输出函数为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.24） |

其中，对于任意， ，且 。在约束条件（9.22）下构造拉格朗日函数，对各个变量求导，并令其求导函数等于零，解得：

|  |  |  |
| --- | --- | --- |
|  |  | （9.25） |

将结果代入原始的拉格朗日函数，可得：

|  |  |  |
| --- | --- | --- |
|  |  | （9.26） |

其约束条件为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.27） |

式中：—为惩罚因子。

对于非线性函数，其逼近可以采用支持向量机分类中介绍的每一种核函数，其中采用较多的是径向基核函数。

如前所述，核函数以及 SVM 的参数选择会对 SVM结果产生一定的影响。因此，在实际应用中需要仔细选择SVM模型中的各个参数，主要包括目标函数中的惩罚系数，损失函数中的，以及核函数中的参数。

# §9.2 使用 SVC对两组分混合气体进行定性识别

与两种气体的谱线在其红外吸收区域存在交叉现象，这意味着测量这两种气体的红外传感器存在交叉敏感性。这种现象在实验标定数据中表现为当浓度不变而浓度变化时，不仅红外传感器的输出电压发生变化，红外传感器的输出电压也发生变化。气体传感器的交叉敏感会给气体类型的识别结果带来误差。

气体的定性识别，指的是判断气体是否存在。当气体的浓度小于规定的分辨率时，就认为这种气体不存在；当气体浓度大于规定的分辨率时，就认为这种气体存在。当然，气体传感器的分辨率必须满足所需的分辨率要求。在本节的标定实验中，每种气体都设置了对应的阈值。如果气体浓度低于这个阈值，就认为该气体不存在；如果气体浓度高于这个阈值，就认为该气体存在。

由于所识别的是两种气体的有无，其结果的组合有四种，所以这是一个多类问题。

## 9.2.1 实验标定

在标定实验中，使用红外气体传感器对和的混合气体进行测量，并记录了在不同气体浓度条件下两种传感器的输出电压和。为了确定混合气体中是否存在或，设定了和气体的浓度阈值，分别为30 ppm和5 ppm。如果被测气体的浓度低于阈值，就认为该气体不存在；如果被测气体的浓度高于阈值，就认为该气体存在。标定实验数据如表9-3所示。

表9-3 SO2和NO2两种气体传感器输出电压与气体有无识别的标定数据

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 |  |  |  |  | 序号 |  |  |  |  |
| 1 | 0.00 | 0.00 | 0 | 0 | 39 | 34.31 | 35.55 | 1 | 1 |
| 2 | 0.59 | 4.52 | 0 | 0 | 40 | 34.49 | 36.93 | 1 | 1 |
| 3 | 0.94 | 7.25 | 0 | 1 | 41 | 35.91 | 15.44 | 1 | 0 |
| 4 | 1.57 | 12.09 | 0 | 1 | 42 | 36.50 | 19.97 | 1 | 0 |
| 5 | 2.11 | 16.22 | 0 | 1 | 43 | 36.86 | 22.69 | 1 | 1 |
| 6 | 2.57 | 19.74 | 0 | 1 | 44 | 37.49 | 27.54 | 1 | 1 |
| 7 | 2.86 | 22.03 | 0 | 1 | 45 | 38.02 | 31.66 | 1 | 1 |
| 8 | 3.04 | 23.41 | 0 | 1 | 46 | 38.48 | 35.18 | 1 | 1 |
| 9 | 4.28 | 1.84 | 0 | 0 | 47 | 38.78 | 37.47 | 1 | 1 |
| 10 | 4.87 | 6.36 | 0 | 0 | 48 | 38.96 | 38.85 | 1 | 1 |
| 11 | 5.22 | 9.09 | 0 | 1 | 49 | 37.56 | 16.15 | 1 | 0 |
| 12 | 5.85 | 13.93 | 0 | 1 | 50 | 38.15 | 20.67 | 1 | 0 |
| 13 | 6.39 | 18.06 | 0 | 1 | 51 | 38.50 | 23.40 | 1 | 1 |
| 14 | 6.85 | 21.58 | 0 | 1 | 52 | 39.13 | 28.24 | 1 | 1 |
| 15 | 7.15 | 23.87 | 0 | 1 | 53 | 39.67 | 32.37 | 1 | 1 |
| 16 | 7.33 | 25.25 | 0 | 1 | 54 | 40.13 | 35.89 | 1 | 1 |
| 17 | 8.16 | 3.51 | 0 | 0 | 55 | 40.43 | 38.18 | 1 | 1 |
| 18 | 8.75 | 8.03 | 0 | 0 | 56 | 40.60 | 39.56 | 1 | 1 |
| 19 | 9.10 | 10.76 | 0 | 1 | 57 | 40.01 | 17.21 | 1 | 0 |
| 20 | 9.73 | 15.60 | 0 | 1 | 58 | 40.60 | 21.73 | 1 | 0 |
| 21 | 10.27 | 19.73 | 0 | 1 | 59 | 40.96 | 24.46 | 1 | 1 |
| 22 | 10.72 | 23.24 | 0 | 1 | 60 | 41.59 | 29.30 | 1 | 1 |
| 23 | 11.02 | 25.53 | 0 | 1 | 61 | 42.12 | 33.43 | 1 | 1 |
| 24 | 11.20 | 26.92 | 0 | 1 | 62 | 42.58 | 36.94 | 1 | 1 |
| 25 | 22.65 | 9.74 | 1 | 0 | 63 | 42.88 | 39.23 | 1 | 1 |
| 26 | 23.24 | 14.26 | 1 | 0 | 64 | 43.06 | 40.61 | 1 | 1 |
| 27 | 23.60 | 16.99 | 1 | 1 | 65 | 1.97 | 2.33 | 0 | 0 |
| 28 | 24.23 | 21.83 | 1 | 1 | 66 | 3.21 | 5.65 | 0 | 0 |
| 29 | 24.76 | 25.96 | 1 | 1 | 67 | 2.71 | 8.01 | 0 | 1 |
| 30 | 25.22 | 29.48 | 1 | 1 | 68 | 4.53 | 22.05 | 0 | 1 |
| 31 | 25.52 | 31.77 | 1 | 1 | 69 | 4.49 | 3.41 | 0 | 0 |
| 32 | 25.70 | 33.15 | 1 | 1 | 70 | 4.27 | 4.74 | 0 | 0 |
| 33 | 31.45 | 13.52 | 1 | 0 | 71 | 5.22 | 9.09 | 0 | 1 |
| 34 | 32.03 | 18.05 | 1 | 0 | 72 | 7.95 | 24.21 | 0 | 1 |
| 35 | 32.39 | 20.77 | 1 | 1 | 73 | 37.96 | 19.23 | 1 | 0 |
| 36 | 33.02 | 25.61 | 1 | 1 | 74 | 39.85 | 23.98 | 1 | 1 |
| 37 | 33.55 | 29.74 | 1 | 1 | 75 | 42.44 | 39.71 | 1 | 1 |
| 38 | 34.01 | 33.26 | 1 | 1 |  |  |  |  |  |

注：（1）与分别为和两种气体传感器的输出电压（mV），作为SVC的样本集的特征；

（2）与分别为判定和两种气体有无的数据，0表示没有该种气体，1表示存在该种气体，作为样本集的标签。

## 9.2.2 数据预处理

在使用支持向量机进行数据处理之前，为了避免数量级的差异，必须对表9-3中的标定数据进行归一化处理。具体来说，使用公式

将样本数据变换到（0，1）的范围之内。归一化后的数据如表9-4所示

表9-4 归一化后的数据

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 |  |  |  |  | 序号 |  |  |  |  |
| 1 | 0.000 | 0.000 | 0 | 0 | 39 | 0.797 | 0.875 | 1 | 1 |
| 2 | 0.014 | 0.111 | 0 | 0 | 40 | 0.801 | 0.909 | 1 | 1 |
| 3 | 0.022 | 0.179 | 0 | 1 | 41 | 0.834 | 0.380 | 1 | 0 |
| 4 | 0.036 | 0.298 | 0 | 1 | 42 | 0.848 | 0.492 | 1 | 0 |
| 5 | 0.049 | 0.399 | 0 | 1 | 43 | 0.856 | 0.559 | 1 | 1 |
| 6 | 0.060 | 0.486 | 0 | 1 | 44 | 0.871 | 0.678 | 1 | 1 |
| 7 | 0.066 | 0.542 | 0 | 1 | 45 | 0.883 | 0.780 | 1 | 1 |
| 8 | 0.071 | 0.576 | 0 | 1 | 46 | 0.894 | 0.866 | 1 | 1 |
| 9 | 0.099 | 0.045 | 0 | 0 | 47 | 0.901 | 0.923 | 1 | 1 |
| 10 | 0.113 | 0.157 | 0 | 0 | 48 | 0.905 | 0.957 | 1 | 1 |
| 11 | 0.121 | 0.224 | 0 | 1 | 49 | 0.872 | 0.398 | 1 | 0 |
| 12 | 0.136 | 0.343 | 0 | 1 | 50 | 0.886 | 0.509 | 1 | 0 |
| 13 | 0.148 | 0.445 | 0 | 1 | 51 | 0.894 | 0.576 | 1 | 1 |
| 14 | 0.159 | 0.531 | 0 | 1 | 52 | 0.909 | 0.695 | 1 | 1 |
| 15 | 0.166 | 0.588 | 0 | 1 | 53 | 0.921 | 0.797 | 1 | 1 |
| 16 | 0.170 | 0.622 | 0 | 1 | 54 | 0.932 | 0.884 | 1 | 1 |
| 17 | 0.190 | 0.086 | 0 | 0 | 55 | 0.939 | 0.940 | 1 | 1 |
| 18 | 0.203 | 0.198 | 0 | 0 | 56 | 0.943 | 0.974 | 1 | 1 |
| 19 | 0.211 | 0.265 | 0 | 1 | 57 | 0.929 | 0.424 | 1 | 0 |
| 20 | 0.226 | 0.384 | 0 | 1 | 58 | 0.943 | 0.535 | 1 | 0 |
| 21 | 0.239 | 0.486 | 0 | 1 | 59 | 0.951 | 0.602 | 1 | 1 |
| 22 | 0.249 | 0.572 | 0 | 1 | 60 | 0.966 | 0.721 | 1 | 1 |
| 23 | 0.256 | 0.629 | 0 | 1 | 61 | 0.978 | 0.823 | 1 | 1 |
| 24 | 0.260 | 0.663 | 0 | 1 | 62 | 0.989 | 0.910 | 1 | 1 |
| 25 | 0.526 | 0.240 | 1 | 0 | 63 | 0.996 | 0.966 | 1 | 1 |
| 26 | 0.540 | 0.351 | 1 | 0 | 64 | 1.000 | 1.000 | 1 | 1 |
| 27 | 0.548 | 0.418 | 1 | 1 | 65 | 0.046 | 0.057 | 0 | 0 |
| 28 | 0.563 | 0.538 | 1 | 1 | 66 | 0.075 | 0.139 | 0 | 0 |
| 29 | 0.575 | 0.639 | 1 | 1 | 67 | 0.063 | 0.197 | 0 | 1 |
| 30 | 0.586 | 0.726 | 1 | 1 | 68 | 0.105 | 0.543 | 0 | 1 |
| 31 | 0.593 | 0.782 | 1 | 1 | 69 | 0.104 | 0.084 | 0 | 0 |
| 32 | 0.597 | 0.816 | 1 | 1 | 70 | 0.099 | 0.117 | 0 | 0 |
| 33 | 0.730 | 0.333 | 1 | 0 | 71 | 0.121 | 0.224 | 0 | 1 |
| 34 | 0.744 | 0.444 | 1 | 0 | 72 | 0.185 | 0.596 | 0 | 1 |
| 35 | 0.752 | 0.511 | 1 | 1 | 73 | 0.882 | 0.474 | 1 | 0 |
| 36 | 0.767 | 0.631 | 1 | 1 | 74 | 0.925 | 0.590 | 1 | 1 |
| 37 | 0.779 | 0.732 | 1 | 1 | 75 | 0.986 | 0.978 | 1 | 1 |
| 38 | 0.790 | 0.819 | 1 | 1 |  |  |  |  |  |

## 9.2.3 标签转换

根据表9-4，这个识别问题实际上是一个多输出的多分类问题。但是，由于支持向量机无法直接处理这种问题，因此需要对标签进行转换。为了简化问题，可以直接将两位二进制转换为十进制，从而实现多输出到单输出的转换。这样，我们就可以使用支持向量机来处理这个问题了。转换后的样本集如表9-5所示。

表9-5 标签转换后的数据

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 |  |  | 类别 | 序号 |  |  | 类别 |
| 1 | 0.000 | 0.000 | 0 | 39 | 0.797 | 0.875 | 3 |
| 2 | 0.014 | 0.111 | 0 | 40 | 0.801 | 0.909 | 3 |
| 3 | 0.022 | 0.179 | 1 | 41 | 0.834 | 0.380 | 2 |
| 4 | 0.036 | 0.298 | 1 | 42 | 0.848 | 0.492 | 2 |
| 5 | 0.049 | 0.399 | 1 | 43 | 0.856 | 0.559 | 3 |
| 6 | 0.060 | 0.486 | 1 | 44 | 0.871 | 0.678 | 3 |
| 7 | 0.066 | 0.542 | 1 | 45 | 0.883 | 0.780 | 3 |
| 8 | 0.071 | 0.576 | 0 | 46 | 0.894 | 0.866 | 3 |
| 9 | 0.099 | 0.045 | 0 | 47 | 0.901 | 0.923 | 3 |
| 10 | 0.113 | 0.157 | 0 | 48 | 0.905 | 0.957 | 3 |
| 11 | 0.121 | 0.224 | 1 | 49 | 0.872 | 0.398 | 2 |
| 12 | 0.136 | 0.343 | 1 | 50 | 0.886 | 0.509 | 2 |
| 13 | 0.148 | 0.445 | 1 | 51 | 0.894 | 0.576 | 3 |
| 14 | 0.159 | 0.531 | 1 | 52 | 0.909 | 0.695 | 3 |
| 15 | 0.166 | 0.588 | 1 | 53 | 0.921 | 0.797 | 3 |
| 16 | 0.170 | 0.622 | 1 | 54 | 0.932 | 0.884 | 3 |
| 17 | 0.190 | 0.086 | 0 | 55 | 0.939 | 0.940 | 3 |
| 18 | 0.203 | 0.198 | 0 | 56 | 0.943 | 0.974 | 3 |
| 19 | 0.211 | 0.265 | 1 | 57 | 0.929 | 0.424 | 2 |
| 20 | 0.226 | 0.384 | 1 | 58 | 0.943 | 0.535 | 2 |
| 21 | 0.239 | 0.486 | 1 | 59 | 0.951 | 0.602 | 3 |
| 22 | 0.249 | 0.572 | 1 | 60 | 0.966 | 0.721 | 3 |
| 23 | 0.256 | 0.629 | 1 | 61 | 0.978 | 0.823 | 3 |
| 24 | 0.260 | 0.663 | 1 | 62 | 0.989 | 0.910 | 3 |
| 25 | 0.526 | 0.240 | 2 | 63 | 0.996 | 0.966 | 3 |
| 26 | 0.540 | 0.351 | 2 | 64 | 1.000 | 1.000 | 3 |
| 27 | 0.548 | 0.418 | 3 | 65 | 0.046 | 0.057 | 0 |
| 28 | 0.563 | 0.538 | 3 | 66 | 0.075 | 0.139 | 0 |
| 29 | 0.575 | 0.639 | 3 | 67 | 0.063 | 0.197 | 1 |
| 30 | 0.586 | 0.726 | 3 | 68 | 0.105 | 0.543 | 1 |
| 31 | 0.593 | 0.782 | 3 | 69 | 0.104 | 0.084 | 0 |
| 32 | 0.597 | 0.816 | 3 | 70 | 0.099 | 0.117 | 0 |
| 33 | 0.730 | 0.333 | 2 | 71 | 0.121 | 0.224 | 1 |
| 34 | 0.744 | 0.444 | 2 | 72 | 0.185 | 0.596 | 1 |
| 35 | 0.752 | 0.511 | 3 | 73 | 0.882 | 0.474 | 2 |
| 36 | 0.767 | 0.631 | 3 | 74 | 0.925 | 0.590 | 3 |
| 37 | 0.779 | 0.732 | 3 | 75 | 0.986 | 0.978 | 3 |
| 38 | 0.790 | 0.819 | 3 |  |  |  |  |

## 9.2.4 样本集划分

为了提高SVC模型的泛化能力并减少过拟合的情况，需要将整个样本集划分为训练集和测试集，并分别用于模型的训练与测试。一般来说，通常会从量程范围内随机选取80%的数据作为训练集，剩余20%作为测试集。这样可以确保测试集中有足够数量的样本来评估模型的泛化性能。本节示例中共有75个样本，从中随机选取60个样本作为训练集，剩余15个样本作为测试集。这种划分方式可以在保证预测精度的同时提高模型的泛化能力。

## 9.2.5 SVC模型结构的确定

为了确定SVM的模型结构，需要使用从样本集中划分出来的训练集和验证集。训练集用于训练SVM的参数，包括核函数的类型和参数、参数、惩罚函数的类型等。验证集用于测试SVM的性能，包括分类准确率、误差率等。通过不断地调整训练集上的参数，使得验证集上的性能达到最优。当SVM的性能满足要求时，就可以确定SVM的模型结构，包括SVM的乘子和偏移量，这些参数决定了SVM的分类边界。这样，就可以使用SVM进行模式识别和预测了。

## 9.2.6 两组分混合气体四种模式的识别

将和两种红外气体传感器的实测输出电压代入已经确定的SVC模型，即可获得被测两组分混合气体成分的模式。

## 9.2.7 预测结果评估

在评估分类模型的性能时，通常会使用以下几种评估指标：

### 1. 准确率（Accuracy）

准确率是分类正确的样本数占总样本数的比例。虽然准确率是最直观的评估指标，但在样本不平衡的情况下可能会产生误导。具体的计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.28） |

### 2. 精确率（Precision）

精确率是被模型正确分类为正类的样本数占被模型分类为正类的样本数的比例。具体的计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.29） |

### 3. 召回率（Recall）

召回率是被模型正确分类为正类的样本数占真正的正类样本数的比例。具体的计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.30） |

### 4. F1分数（F1）

F1分数是精确率和召回率的调和平均数，可以同时考虑精确率和召回率。计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | （9.31） |

式中：TP—真正例（True Positive，TP），即被模型正确分类为正类的样本；

TN—真反例（True Negative，TN），即被模型正确分类为反类的样本；

FP—假正例（False Positive，FP），即被模型错误分类为正类的样本；

FN—假反例（False Negative，FN），即被模型错误分类为反类的样本。

### 5. AUC-ROC

AUC-ROC(Area Under the Curve-Receiver Operating Characteristics)是以假正例率为横轴，真正例率为纵轴画出的ROC曲线下的面积。对于其计算，涉及到多个阈值下的TPR（True Positive Rate）和FPR（False Positive Rate），并需要计算曲线下的面积，通常不会直接给出计算公式，而是通过专门的数据分析工具或库来计算。AUC-ROC的值介于0.5（随机分类器）和1（完美分类器）之间。

在本节实例中，采用多项式核的测试样本输出结果为：

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 期望输出 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 3 | 4 | 4 |
| 预测输出 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 3 | 4 | 4 |

采用Gaussian核的测试样本输出结果为：

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 期望输出 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 3 | 4 | 4 |
| 预测输出 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 3 | 4 | 4 |

以上两个表格表明，无论是使用多项式核还是Gaussian核的SVM，都能对两种气体进行有效的辨识。根据表中数据可以看出，所有的预测结果都与期望输出相符，正确分类的样本数等于样本总数，假正例和假反例均等于0，故可计算出准确率为100%，精确率为100%，召回率为100%，F1分数也为100%，AUC-ROC为1，可实现完美分类。需要注意的是，采用Gaussian核时，必须正确调整核函数的带宽σ，否则，如果σ取值不当，可能会影响SVM的分类效果。

# §9.3 基于支持向量机的传感器非线性校正及应用

在传感器的应用中，通常期望被测量与传感器输出信号之间呈线性关系。因此，非线性度是传感器或仪表的一项重要技术指标。它对整个测控系统最终精度的影响很大。无论传感器或仪表的精度有多高，非线性的存在总是无法完全避免的。因此，为了满足生产过程高精度测量的需要，需要对工业过程传感器或仪表的输出输入特性进行非线性校正。这是传感器或者仪表研究的一个重要的研究课题。接下来，将以电容式差压变送器为例，详细介绍如何利用支持向量机进行传感器的非线性校正。

## 9.3.1 一维标定实验

本文采用的电容式差压变送器具有-40kPa∼+40kPa的测量范围，并且其输出为0∼2.5V的模拟电压（经过放大）。通过对批量生产的传感器进行随机采样，抽取了9个样本，样本数据见表9-1。

表9-1 电容式差压变送器标定数据

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P  （） | 传感器输出（V） | | | | | | | | |
| 样本1 | 样本2 | 样本3 | 样本4 | 样本5 | 样本6 | 样本7 | 样本8 | 样本9 |
| 40 | 2.3769 | 2.3419 | 2.3410 | 2.3811 | 2.3550 | 2.3725 | 2.3529 | 2.3405 | 2.3292 |
| 35 | 2.2274 | 2.1960 | 2.1947 | 2.2343 | 2.2118 | 2.2261 | 2.2095 | 2.1955 | 2.1847 |
| 30 | 2.0814 | 2.0534 | 2.0510 | 2.0914 | 2.0695 | 2.0835 | 2.0690 | 2.0529 | 2.0427 |
| 25 | 1.9391 | 1.9139 | 1.9117 | 1.9516 | 1.9289 | 1.9437 | 1.9316 | 1.9116 | 1.9032 |
| 20 | 1.7996 | 1.7770 | 1.7744 | 1.8144 | 1.7903 | 1.8060 | 1.7966 | 1.7724 | 1.7662 |
| 15 | 1.6622 | 1.6421 | 1.6393 | 1.6797 | 1.6538 | 1.6701 | 1.6636 | 1.6354 | 1.6311 |
| 10 | 1.5270 | 1.5089 | 1.5061 | 1.5467 | 1.5193 | 1.5362 | 1.5324 | 1.5002 | 1.4975 |
| 5 | 1.3930 | 1.3767 | 1.3739 | 1.4147 | 1.3858 | 1.4033 | 1.4021 | 1.366 | 1.3649 |
| 0 | 1.2588 | 1.2445 | 1.2417 | 1.2829 | 1.2524 | 1.2702 | 1.2717 | 1.2318 | 1.2324 |
| -5 | 1.1247 | 1.1122 | 1.1095 | 1.1510 | 1.1188 | 1.1372 | 1.1413 | 1.0976 | 1.0995 |
| -10 | 0.9903 | 0.9796 | 0.9772 | 1.0188 | 0.9850 | 1.0039 | 1.0105 | 0.9631 | 0.9665 |
| -15 | 0.8543 | 0.8456 | 0.8434 | 0.8851 | 0.8499 | 0.8690 | 0.8784 | 0.8271 | 0.8318 |
| -20 | 0.7163 | 0.7095 | 0.7077 | 0.7493 | 0.7127 | 0.7322 | 0.7445 | 0.6887 | 0.6951 |
| -25 | 0.5760 | 0.5713 | 0.5697 | 0.6116 | 0.5734 | 0.5931 | 0.6084 | 0.5482 | 0.5560 |
| -30 | 0.4327 | 0.4305 | 0.4288 | 0.4705 | 0.4310 | 0.4510 | 0.4694 | 0.4046 | 0.4138 |
| -35 | 0.2860 | 0.2865 | 0.2843 | 0.3262 | 0.2853 | 0.3057 | 0.3273 | 0.2576 | 0.2681 |
| -40 | 0.1352 | 0.1390 | 0.1362 | 0.1781 | 0.1350 | 0.1561 | 0.1811 | 0.1059 | 0.1189 |

## 9.3.2 数据的预处理

在本节的模型讨论中，一律以传感器输出电压为自变量，以被测差压为因变量，因为只有这样建模才更方便实际使用。

不失一般性，这里以标定数据样本3为样本集进行建模和预测。在使用支持向量机进行数据处理之前，为了避免数量级的差异，必须对表9-1中的标定数据进行归一化处理。具体来说，使用公式

将样本数据变换到（0，1）的范围之内。归一化后的数据如表9-2所示。

表9-2 归一化后的数据

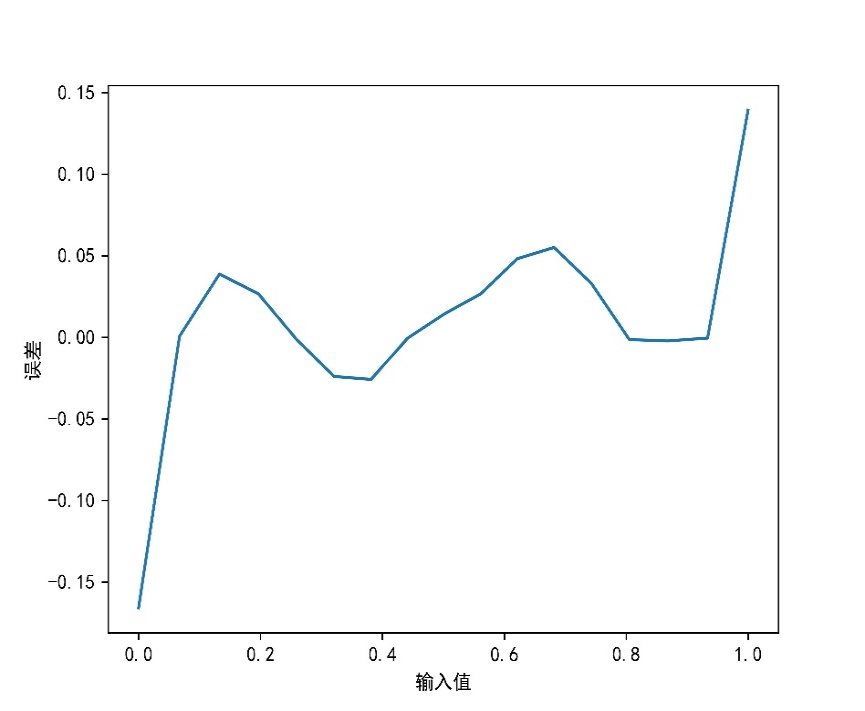
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P  （） | 传感器输出（V） | | | | | | | | |
| 样本1 | 样本2 | 样本3 | 样本4 | 样本5 | 样本6 | 样本7 | 样本8 | 样本9 |
| 40 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 35 | 0.9333 | 0.9338 | 0.9336 | 0.9334 | 0.9355 | 0.9339 | 0.9340 | 0.9351 | 0.9346 |
| 30 | 0.8682 | 0.8690 | 0.8685 | 0.8685 | 0.8714 | 0.8696 | 0.8693 | 0.8713 | 0.8704 |
| 25 | 0.8047 | 0.8057 | 0.8053 | 0.8050 | 0.8081 | 0.8065 | 0.8060 | 0.8081 | 0.8073 |
| 20 | 0.7425 | 0.7436 | 0.7430 | 0.7428 | 0.7456 | 0.7444 | 0.7439 | 0.7458 | 0.7453 |
| 15 | 0.6812 | 0.6823 | 0.6817 | 0.6816 | 0.6841 | 0.6831 | 0.6826 | 0.6845 | 0.6842 |
| 10 | 0.6209 | 0.6219 | 0.6213 | 0.6212 | 0.6236 | 0.6227 | 0.6222 | 0.6240 | 0.6237 |
| 5 | 0.5611 | 0.5619 | 0.5614 | 0.5613 | 0.5634 | 0.5627 | 0.5622 | 0.5639 | 0.5637 |
| 0 | 0.5012 | 0.5018 | 0.5014 | 0.5015 | 0.5033 | 0.5027 | 0.5022 | 0.5038 | 0.5038 |
| -5 | 0.4414 | 0.4418 | 0.4414 | 0.4416 | 0.4432 | 0.4427 | 0.4421 | 0.4438 | 0.4437 |
| -10 | 0.3815 | 0.3816 | 0.3814 | 0.3816 | 0.3829 | 0.3825 | 0.3819 | 0.3836 | 0.3835 |
| -15 | 0.3208 | 0.3208 | 0.3208 | 0.3209 | 0.3220 | 0.3216 | 0.3211 | 0.3227 | 0.3225 |
| -20 | 0.2592 | 0.2590 | 0.2592 | 0.2593 | 0.2602 | 0.2599 | 0.2594 | 0.2608 | 0.2607 |
| -25 | 0.1966 | 0.1962 | 0.1966 | 0.1968 | 0.1975 | 0.1972 | 0.1967 | 0.1979 | 0.1978 |
| -30 | 0.1327 | 0.1323 | 0.1327 | 0.1327 | 0.1333 | 0.1331 | 0.1327 | 0.1337 | 0.1334 |
| -35 | 0.0673 | 0.0670 | 0.0672 | 0.0672 | 0.0677 | 0.0675 | 0.0673 | 0.0679 | 0.0675 |
| -40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

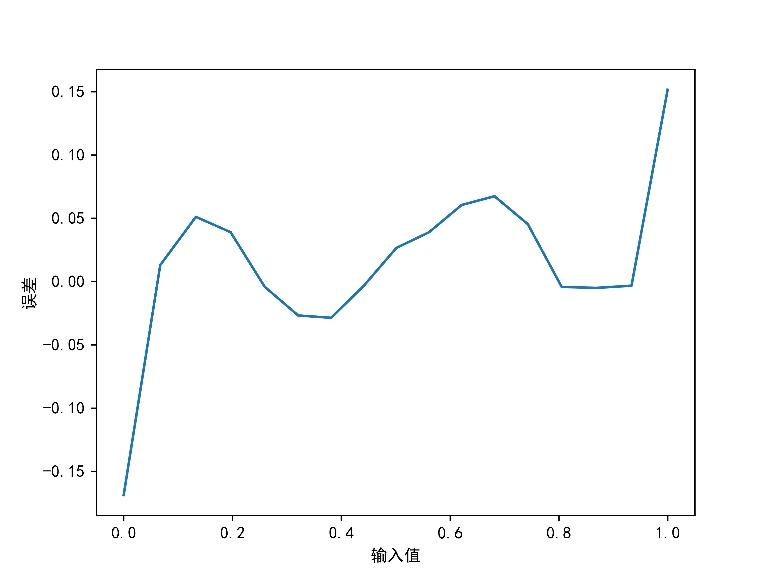
## 9.3.3 样本集划分

如果直接使用整个样本集进行训练和预测，虽然预测的精度可能很高，但模型的泛化能力可能较低。这可能导致得到的拟合曲线是一条振荡曲线，而非期望中的直线。因此，需要将整个样本集划分为训练集和测试集。由于样本3中的标定数据较少，仅有17个，因此采用了较为保守的划分比例，即从量程范围内随机选取80%的数据作为训练集,另外20%作为测试集。这样做可以确保测试集中有足够数量的样本来评估模型的泛化性能。具体来说,在标定数据的量程范围内随机选取了14个样本作为训练集，剩余3个样本作为测试集。这种划分方式在样本量较少的情况下是较为标准和合理的。

## 9.3.4 核函数的选择

目前对于如何选择支持向量机的核函数并没有统一的理论指导。针对本样本数据，这里分别选用常用的径向基核函数和sigmoid核函数进行建模。图9-4和图9-5分别是基于径向基核函数和sigmoid核函数的模型预测后的误差曲线。

  
图9-4 基于径向基核函数拟合后的误差曲线

  
图9-5 基于sigmoid核函数模型拟合后的误差曲线

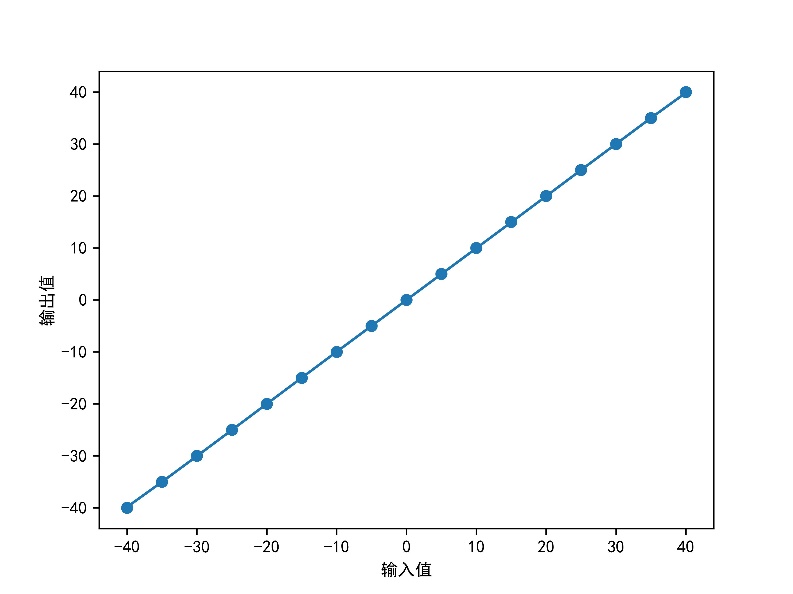
从上面两个图中的效果看，基于sigmoid 核函数训练得到的最大误差为，而基于径向基核函数训练得到的最大误差为。由于径向基核函数的预测精度更高，因此本节选择效果比较理想的径向基核函数作为待训练SVM模型的核函数。

## 9.3.5 SVM参数的优化

如果直接采用默认的参数去训练SVM模型进行预测，其预测精读一般较低，无法满足需要。为此，需要采用合适的优化方法来优化SVM模型的超参数。常用的优化方法主要有粒子群优化方法，蚁群优化方法等，可以自行学习相关的优化方法。SVM模型需要的超参数主要有目标函数中的惩罚系数，损失函数中的，以及核函数中的参数。对于本节选用的径向基核函数，需要优化的主要有惩罚系数C、损失函数中的ε和核参数。可以自行选择合适的优化方法对这些超参数进行优化，在这里直接给出优化结果：，，。

## 9.3.6 预测结果验证

选用优化后的超参数训练SVM模型，并采用训练好的SVM模型对样本3进行预测，预测结果如图9-6所示。

  
图9-6 预测结果图

由图9-6可以看出，校正后，拟合曲线基本上就是一条直线，线性度较高，可以达到非线性校正的目的。

为了进一步说明非线性校正的效果，下面通过

计算线性度来进行定量说明。

校正前：

拟合直线为，将传感器输入代入并与传感器的输出比较，可得到最大拟合误差为，而量程，代入可得：

校正后：

拟合直线为，将传感器输入代入并与传感器的输出比较，可得到最大拟合误差为，而量程，代入可得：

校正后，线性引用误差由降低到，线性度大大提高。

## 9.3.7 模型的移植

在完成了模型的训练和验证之后，需要将模型移植到实际的传感系统中，以实现传感器的非线性校正。传感器在现场实际使用的过程中，其实际测量输出一般需要通过单片机或DSP进行数据处理。然而，这些硬件环境无法安装Python等大型开发环境，因此需要进行移植。移植的步骤如下：

（1）在PC计算机上利用Python开发环境或其它大型数值计算软件训练支持向量机；

（2）提取已训练支持向量回归机SVR的权重（）和偏置项（）。可以使用以下Python代码来提取这些参数：

**dual\_coef = model.dual\_coef\_**  #权重

**intercept = model.intercept\_**  #偏置项

（3）输入待处理的向量，即传感器实际测量的值；

（4）根据核函数求取核函数矩阵的数值；

以RBF核函数为例，其计算公式为：

式中：—待处理的向量；

—支持向量；

—RBF的带宽。

（5）根据支持向量机训练得到的权重系数矩阵、偏置和公式，求取处理结果。

利用C语言编写程序实现上述功能，即可将支持向量机模型移植到单片机或DSP中，以在现场实际环境中使用传感器。

# §9.4 支持向量机的Python实现

## 9.4.1 支持向量机回归（SVR）的实现

### 9.4.1.1 安装需要的第三方库

支持向量机回归（SVR）的实现主要需要三个第三方库，分别是numpy、scikit-learn和pandas。前面已经详细介绍了前两个库的安装方法，下面着重介绍pandas库的安装，其安装命令如下：

pip install pandas

### 9.4.1.2 引用第三方库

在使用第三方库之前，我们需要在程序中导入它们。为了实现SVR，需要添加以下的导入语句：

**import pandas as pd**

**import numpy as np**

**from sklearn.svm import SVR**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.metrics import mean\_squared\_error**

### 9.4.1.3 导入样本数据集

如果样本数据较少，可以直接使用numpy库创建样本数据数组并输入到程序中。但是，当样本数据较多时，通常会先将数据存储为文件，然后再通过pandas库导入到python环境中进行后续处理。在本节案例中，由于样本数据较多，采用了后一种方法。事先已经将数据存储为data.xlsx文件，因此可以使用pandas库中的read\_excel()函数来读取数据。read\_excel()是pandas库中用于读取excel文件并将其转换为DataFrame的函数。其原型为：

pandas.read\_excel(io, sheet\_name=0, header=0, names=None, index\_col=None, usecols=None, dtype=None, skiprows=None, \*)

它具有许多参数，可以根据需求进行调整。下面是一些常用参数的介绍：

**io**：文件路径或类似文件的对象。可以是字符串、路径对象或具有read()方法的对象。

**sheet\_name**：指定要读取的工作表的名称或索引，默认为0，表示读取第一个工作表。

**header**：指定用作列名的行号，默认为’infer’，表示自动推断。

**names**：指定列名，如果不指定，则使用header参数推断列名。

**index\_col**：指定用作行索引的列号或列名。

**usecols**：指定要读取的列。

**dtype**：指定每列的数据类型。

**skiprows**：指定要跳过的行数或要跳过的行号列表。

为了降低编程难度，一般在数据文件中只存放要处理的数据，包括特征和标签，而去掉所有的行号和列名，则读取data.xlsx文件中数据的实例代码为：

data = pd.read\_excel(‘data.xlsx’)

### 9.4.1.4归一化数据

归一化数据是一种数据预处理方法，目的是将数据的数值范围统一到一个较小的区间，通常是[0,1]或[-1,1]，以减少数据的差异性和异常值的影响，提高模型的稳定性和收敛速度。根据最小-最大特征缩放原理，可以采用如下代码实现数据的归一化：

normalized\_data = (data - data.min()) / (data.max() - data.min())

### 9.4.1.5划分训练集和测试集

采用scikit-learn库中model\_selection子模块提供的train\_test\_split()函数划分训练集和测试集，上一章已经介绍，本章不再重复介绍。

### 9.4.1.6 创建并训练SVR模型

scikit-learn库中的svm子模块提供了SVR类，用于实现支持向量机回归预测。该类的语法如下：

model = SVR(kernel='rbf', degree=3, gamma='scale', coef0=0.0, tol=0.001, C=1.0, epsilon=0.1, shrinking=True, cache\_size=200, verbose=False, max\_iter=-1)

该类的参数说明如下：

**kernel**：指定算法中使用的核函数类型，可以是字符串类型，默认为`'rbf'`，表示使用径向基函数，也可以取其他值，如`'linear'`表示使用线性核函数；

**degree**：指定多项式核函数的次数，可以是整数类型，默认为3，表示多项式核函数的次数为3；

**gamma**：指定核函数的系数，可以是字符串类型或浮点数类型，默认为`'scale'`，表示使用`1 / (n\_features \* X.var())`作为gamma的值；

**coef0**：指定核函数中的独立项，可以是浮点数类型，默认为0.0；

**tol**：指定停止准则的容忍度，可以是浮点数类型，默认为0.001；

**C**：指定正则化参数，可以是浮点数类型，默认为1.0，表示正则化强度与C成反比；

**epsilon**：指定epsilon-SVR模型中的epsilon值，可以是浮点数类型，默认为0.1，表示在训练损失函数中与实际值距离epsilon以内的点不会受到惩罚；

**shrinking**：指定是否使用收缩启发式方法，可以是布尔类型，默认为True；

**cache\_size**：指定内核缓存的大小（以MB为单位），可以是浮点数类型，默认为200；

**verbose**：指定是否启用详细输出，可以是布尔类型，默认为False；

**max\_iter**：指定求解器内部迭代次数的硬限制，可以是整数类型，默认为-1，表示没有限制。

该类的主要方法说明如下：

**fit(X, y)**：根据输入数据X和标签数据y训练支持向量机回归模型；

**predict(X)**：根据输入数据X预测输出数据y；

**score(X, y)**：根据输入数据X和标签数据y评估支持向量机回归模型的性能；

使用该类的一个示例代码如下：

**from sklearn.svm import SVR**

**model = SVR(kernel='rbf', C=1.0, epsilon=0.1)**

**model.fit(X\_train, y\_train)**

### 9.4.1.7 预测并评估模型性能

预测并评估模型性能是机器学习中的一个重要步骤，它有助于了解模型的泛化能力和优化方向。具体的方法，在上一章已经介绍，本章不再赘述。

**例9-1** 编写Python程序，实现9-3节的涡流传感器的非线性校正。

代码如下：

# 导入必要的库

import pandas as pd

import numpy as np

from sklearn.svm import SVR

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_squared\_error

# 假设您的数据存储在一个名为"data.csv"的文件中

data = pd.read\_csv("data.csv", header=None, index\_col=None)

# 按列归一化

normalized\_data = data.copy()

normalized\_data.iloc[:, 1:] = (data.iloc[:, 1:] - data.iloc[:, 1:].min()) / (data.iloc[:, 1:].max() - data.iloc[:, 1:].min())

# 样本3 传感器的输出

x = normalized\_data.iloc[:, 3].values.reshape(-1, 1)

# 样本3 传感器的输入

y = normalized\_data.iloc[:, 0].values

# 划分训练集和测试集

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2)

# 创建SVR模型

svr = SVR(kernel='rbf', C=1000, gamma=1, epsilon=0.001, max\_iter=10000)

# 训练模型

svr.fit(x\_train, y\_train)

# 计算拟合后的输出值

y\_fit = svr.predict(x\_test)

# 计算拟合误差

mse = mean\_squared\_error(y\_test, y\_fit)

## 9.4.2 支持向量机分类（SVC）的python实现

### 9.4.2.1 安装需要的第三方库

支持向量机分类（SVC）的实现需要三个第三方库，分别是numpy、scikit-learn和pandas。这些库的安装方法已经在前面详细介绍过了，这里不再赘述。可以根据前面的介绍安装这些库，以便在程序中使用它们。

### 9.4.2.2 引用第三方库

在使用第三方库之前，需要在程序中导入它们。为了实现支持向量机分类（SVC），需要添加以下的导入语句：

import numpy as np

import pandas as pd

from sklearn.svm import SVC

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

这些语句将导入numpy库、pandas库、scikit-learn库中的SVC类、train\_test\_split函数和accuracy\_score函数，以便在程序中使用它们。您可以根据需要添加其他的导入语句。

### 9.4.2.3 导入样本数据集

与SVR类似，本节仍然pandas库导入样本数据。但是，与上一次介绍的导入语句相比，本节增加了一个lambda匿名函数，以实现将多输出转换为单输出，具体的示例代码如下：

**data = pd.read\_csv('data1.csv')**

**X = data.iloc[:, :2]**

**y = data.iloc[:, 2:].apply(lambda x: x[0]\*2 + x[1], axis=1)**

### 9.4.2.4划分训练集和测试集

采用scikit-learn库中model\_selection子模块提供的train\_test\_split()函数划分训练集和测试集，前面已经介绍，本节不再重复介绍。

### 9.4.2.5归一化数据

SVC的归一化与SVR的完全一致，本节不再赘述，可以自行参阅9.4.1.4。

### 9.4.2.6 创建并训练SVC模型

scikit-learn库中的svm子模块提供了SVC类，用于实现支持向量机分类预测。该类的语法如下：

model = SVC(\*, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache\_size=200, class\_weight=None, verbose=False, max\_iter=-1, decision\_function\_shape='ovr', break\_ties=False, random\_state=None)

该类的参数说明如下：

**C**：正则化参数，浮点数，默认为1.0。正则化强度与C成反比。必须严格为正。惩罚是平方l2惩罚；

**kernel**：指定算法中使用的核函数类型，可以是字符串类型或可调用类型，默认为'rbf'，表示使用径向基函数，也可以取其他值，如'linear'表示使用线性核函数；

**degree**：指定多项式核函数的次数，可以是整数类型，默认为3，表示多项式核函数的次数为3；

**gamma**：指定核函数的系数，可以是字符串类型或浮点数类型，默认为'scale'，表示使用1 / (n\_features \* X.var())作为gamma的值；

**coef0**：指定核函数中的独立项，可以是浮点数类型，默认为0.0；

**shrinking**：指定是否使用收缩启发式方法，可以是布尔类型，默认为True；

**probability**：指定是否启用概率估计，可以是布尔类型，默认为False；

**tol**：指定停止准则的容忍度，可以是浮点数类型，默认为0.001；

**cache\_size**：指定内核缓存的大小（以MB为单位），可以是浮点数类型，默认为200；

**class\_weight**：设置类i的参数C为class\_weight [i]\*C for SVC。如果未给出，则所有类都应具有权重1。 “平衡”模式使用y的值自动调整权重与输入数据中类频率成反比，即n\_samples / (n\_classes \* np.bincount(y))。

**verbose**：指定是否启用详细输出，可以是布尔类型，默认为False；

**max\_iter**：指定求解器内部迭代次数的硬限制，可以是整数类型，默认为-1，表示没有限制。

该类的主要方法说明如下：

**fit(X, y)**：根据输入数据X和标签数据y训练支持向量机分类模型；

**predict(X)**：根据输入数据X预测输出数据y；

**score(X, y)**：根据输入数据X和标签数据y评估支持向量机分类模型的性能；

使用该类的一个示例代码如下：

**from sklearn.svm import SVC**

**model = SVC(kernel='rbf', C=1.0)**

**model.fit(X\_train, y\_train)**

### 9.4.2.7 预测并评估模型性能

预测并评估模型性能是机器学习中的一个重要步骤，它有助于了解模型的泛化能力和优化方向。在SVC模型中，可以使用训练好的模型在测试集上进行预测，并计算各种评价指标来评估模型性能。下面是一个简单的示例代码：

# 在测试集上进行预测

y\_pred = clf.predict(X\_test\_scaled)

# 计算评价指标

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred, average='macro')

recall = recall\_score(y\_test, y\_pred, average='macro')

f1 = f1\_score(y\_test, y\_pred, average='macro')

在这段代码中，首先使用训练好的模型clf对测试数据X\_test\_scaled进行分类预测得的预测输出数据y\_pred。然后，使用accuracy\_score、precision\_score、recall\_score和f1\_score函数分别计算准确率、精确度、召回率和F1值。这些评价指标有助于了解模型在测试集上的性能，并为后续的模型优化提供参考。

# 习题9

1. 常见的核函数有哪些？它们各自适用于什么样的问题？

2. 什么是惩罚系数？它有什么影响？

3. 什么是松弛变量？它有什么作用？

4. 常用的误差函数有哪些？它们的特点是什么？

5. 什么是气体的定性识别？

6. 如何对标签进行转换？

7. 如何确定SVC模型的结构？

8. 什么是准确率、精确率、召回率、F1分数和AUC-ROC？

9. 什么是传感器非线性校正？为什么需要进行非线性校正？

10. 如何评价SVM模型的非线性校正效果？需要使用哪些指标？

11. 请简述将SVM模型移植到实际的传感系统中的步骤？