桂林電子科技大学

GUILIN UNIVERSITY OF ELECTRONIC TECHNOLOGY

机器学习支持向量机算法综述

课　　程：机器学习

课　　号：2222231

学　　号：2000500927

姓　　名：吴河山

学　　院：计算机与信息安全

专　　业：计算机科学与技术

指导老师：雷晓春

2023 年 5 月

摘 要

支持向量机（SVM）是一种基于统计学习理论的二分类模型，它的目标是寻找一个最大化几何间隔的线性超平面，将不同类别的样本分开。SVM 在计算机视觉、自然语言处理、生物信息学等领域有广泛的应用。本文首先介绍 SVM 的基本理论，通过实验例子展现 SVM 算法的效果，接着介绍了 SVM 在各个领域的应用情况，最后对 SVM 的研究问题和发展趋势进行了展望。

关 键 词：支持向量机，统计学习理论，训练算法

目　录

1 引言

机器学习是一门利用数据中的规律来预测未知或难以观察的数据的学科，它的一个重要理论基础是统计学。统计学习理论 [1] 针对有限样本情况下的机器学习问题，提出了一种新的通用学习方法，叫做支持向量机 (support vector machines,SVM)。它采用结构风险最小化原则，而不是传统统计学的经验风险最小化原则 [2, 3]，在解决小样本、非线性和高维模式识别问题中表现出许多特有的优势，并在很大程度上克服了“维数灾难”和“过学习”等问题，在当时表现出许多优于已有方法的性能，迅速引起各领域的注意和研究兴趣，取得了大量的应用研究成果，推动了各领域的发展。

SVM 是在分类与回归分析中分析数据的监督式学习模型与相关的学习算法，是由 AT&T 贝尔实验室的 Vladimir Vapnik 和他的同事 [Cortes and Vapnik,1995[2]] 开发的，是基于统计学习框架或 Vapnik(1982，1995) 和 Chervonenkis(1974) 提出的 VC 理论 [4] 的最稳健的预测方法之一。

SVM 的目标是寻找一个最大化几何间隔的线性超平面，将不同类别的样本分开。SVM 利用核函数将原始特征空间映射到更高维的特征空间，在此特征空间中构造线性决策面。决策面的特殊性质保证了学习机的高泛化能力从而实现非线性分类 [2]。SVM 的优化问题可以通过拉格朗日乘子法和二次规划求解，也可以采用一些高效的算法，如序列最小优化 (SMO) 算法。SVM 在计算机视觉、自然语言处理、生物信息学等领域 [5] 有广泛的应用 [6]。本文首先对 SVM 的理论进行系统的介绍，通过实验例子展现 SVM 算法的效果，同时阐述 SVM 在各个领域的应用情况，并对未来的研究方向进行展望。

2 研究背景

2.1 支持向量机基本原理

2.1.1 支持向量机的目的

支持向量机的理论最初来自数据分类问题。对于数据分类问题，如果采用传统回归机器学习方法来实现，其原理可以简单地描述为算法随机产生一个平面，直到训练集中属于不同分类的点正好位于平面的不同侧面。这种处理机制决定了进行数据分类最终获得的分割平面将相当靠近训练集中的点，而在绝大多数情况下，并不是一个最优解，同时还增加了训练强度 [6]。为此支持向量机考虑寻找一个满足分类要求的分割平面，并使训练集中的点距离该分割平面尽可能地远，即寻找一个分割平面，使其两侧的空白区域最大，如图2-1所示：

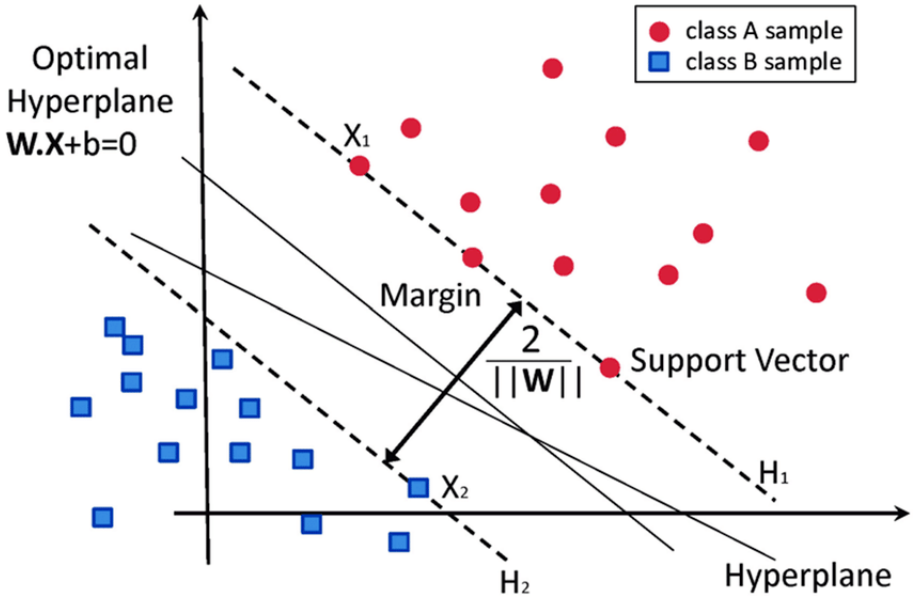

图 2-1 数据点集实现最大化分类间隔

图中红色和蓝色点分别表示两种不同类型的数据，$\omega * x + b = 0$、H_1、H_2 是区分两类数据的分割平面。其中 H_1，H_2 是划分两类数据的边缘分割平面，它们之间的距离 margin 就是两类之间的分割间隔，而图中位于分割平面 H_1，H_2 上的红色和蓝色点即为支持向量 [6]。支持向量机通过先验选择的非线性映射，将输入向量映射到某个高维特征空间 Z 中。在该空间中构造了一个具有特殊性质的线性决策面，保证了数据集的高泛化能力 [2]，其目的就是寻求一个最优的分割平面使两类之间的分割间隔最大。

2.1.2 线性两分类支持向量机

2.1.2.1 线性可分两分类

对于线性可分问题, 支持向量机运用优化算法实现最大化分割间隔, 即只要一个超平面 H 就能正确划分所有训练样本的类别。给定训练样本集 $(x_i, y_i), i = 1, 2, ..., l, x \in R^n, y \in \{\pm 1\}$、, 超平面记作 $(\boldsymbol{\omega} \cdot \boldsymbol{x}) + b = 0$ ，为使分类面对所有样本正确分类并且具备分类间隔，就要求它满足如下约束:

$$y_i[(\boldsymbol{\omega} \cdot x_i) + b] \geq 1 \quad i = 1, 2, ..., l \tag{2-1}$$

可以计算出分类间隔为 $\frac{2}{\|\omega\|}$ ，因此构造最优超平面的问题就转化为在约束式下求:

$$\min \Phi(\boldsymbol{\omega}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 = \frac{1}{2}(\boldsymbol{\omega}' \cdot \boldsymbol{\omega}) \tag{2-2}$$

为了实现最优化问题，引入 Lagrange 函数:

$$\mathrm{L}(\boldsymbol{\omega}, b, \mathrm{a}) = \frac{1}{2}\|\boldsymbol{\omega}\| + a(1 - \boldsymbol{y}((\boldsymbol{\omega} \cdot \boldsymbol{x}) + b)) \tag{2-3}$$

2-3式中，$a_i > 0$ 为 Lagrange 乘数。该 QP 问题转化为相应的对偶问题即:

$$\begin{gathered} \max Q(a) = \sum_{j=1}^{l} a_j - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} a_i a_j y_i y_j (x_i \cdot x_j) \\ \text{s.t. } \sum_{j=1}^{l} a_j y_j = 0 \quad j = 1, 2, \cdots, l, a_j \geqslant 0, j = 1, 2, \cdots, l \end{gathered} \tag{2-4}$$

解得最优解:$\boldsymbol{a}^* = (a_1^*, a_2^*, \cdots, a_l^*)^{\mathrm{T}}$。计算最优权值向 ω^* 和最优偏置 b^*，分别为:

$$\begin{gathered} \boldsymbol{\omega}^* = \sum_{j=1}^{l} a_j^* y_j x_j \\ b^* = y_i - \sum_{j=1}^{l} y_j a_j^* (x_j \cdot x_i) \end{gathered} \tag{2-5}$$

2-5式中，下标 $j \in \{j | a_j^*\}$。因此得到最优分类超平面 H：$(\boldsymbol{\omega}^* \cdot \boldsymbol{x}) + b_* = 0$，而最优分类函数为:

$$f(\boldsymbol{x}) = \operatorname{sgn}\{(\boldsymbol{\omega}^* \cdot \boldsymbol{x}) + b^*\} = \operatorname{sgn}\left\{\left(\sum_{j=1}^{l} a_j^* y_j (x_j \cdot x_i)\right) + b^*\right\}, \boldsymbol{x} \in R^n \tag{2-6}$$

2.1.2.2 线性不可分两分类

对于训练样本是线性不可分的情况，存在个别训练样本无法满足式2-1。此时，支持向量机应当允许一定的分类错误。具体的做法是，在式2-1的约束条件中, 引入松弛变量以软化约束条件，同时在目标函数中对松弛变量进行惩罚以避免松弛变量取值过大而引起的大量错分情况 [7]。也就是说，线性不可分两类分类支持向量机求解下列问题:

$$\begin{aligned}&\min_{\omega,b,\xi_t}\Phi(\boldsymbol{\omega})+C\sum_t\xi_t,\\&\text{s.t.}\begin{cases}(\omega)^Tx_i+b\geq 1-\xi_t,\ \text{if } a_t=+1,\\(\omega)^Tx_i+b\leq -1+\xi_t,\ \text{if } a_t=-1,\\\xi_t\geq 0.\end{cases}\end{aligned}\tag{2-7}$$

2.1.3 非线性两分类支持向量机

客观世界大多都是非线性的训练样本, 尤其是不可分问题, 对于非线性不可分问题, 支持向量机通过适当的核函数将输入空间映射到高维空间，将非线性问题转化线性问题，实现高维空间线性可分，然后在新空间中利用二次型寻优算法求取最优线性分类面，从而将两类样本区分开来 [8, 9]。

在上面的对偶问题中, 都只涉及训练样本之间的内积运算, 这样, 在高维空间实际上只需进行内积运算, 而这种内积运算是可以用原空间中的函数实现的, 我们甚至没有必要知道变换的形式 [1]。根据泛函的有关理论, 只要一种核函数 $K(x_i,x_j)$ 满足 $Mercer$ 条件, 它就对应某一变换空间中的内积 [10]。

因此, 在最优分类面中采用适当的内积函数 $K(x_i,x_j)$ 就可以实现某一非线性变换后的线性分类, 而计算复杂度却没有增加 [1], 此时约束条件2-3变为:

$$\Phi(\boldsymbol{\omega})=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_1\alpha_jy_iy_jK\left(x_i,x_j\right)\tag{2-8}$$

支持向量机的核心是选择合适的核函数和参数，它们直接影响了模型的预测能力和泛化能力。为了解决不同的问题，我们需要根据数据特征和目标函数来选择最优的核函数和参数。然而，这是一个具有挑战性的问题，因为我们需要在速度和准确性之间找到最佳的折中。

下面，给出求解非线性两分类问题的支持向量机算法一般框架如下:

1: 准备数据，将数据分为训练集和测试集，确定特征和标签
2: 选择核函数，根据数据的分布和复杂度选择合适的核函数
3: 训练模型，找到最优的超平面，并确定支持向量和分类边界
4: 预测数据，将测试集输入模型，根据超平面的位置判断并计算准确率和误差
5: 评估结果，根据预测的效果和目标，调整参数或核函数，优化模型的性能

算法 2-1 (支持向量机算法)

3 实验测试

3.1 实验背景

Python 的 sklearn 扩展包中包含了“威斯康星州乳腺癌”数据集，其中详细记录了威斯康星大学附属医院的乳腺癌测量数据。数据集包括 569 行和 31 个特征。可以使用这个数据集训练一个支持向量机模型，以判断一个患者的肿瘤是良性还是恶性。

3.2 实验步骤

- 加载 data 文件夹里的数据集：威斯康星乳腺肿瘤数据集
- 进行数据清洗（如删除无用列，将诊断结果的字符标识 B、M 替换为数值 0、1 等）
- 进行特征选取（方便后续的模型训练）。
- 进行数据集的划分（训练集和测试集），抽取特征选择的数值作为训练和测试数据。
- 配置模型，创建 SVM 分类器，选取线性核函数与高斯核函数进行对比。
- 训练测试与评估模型。

3.3 实验结果

$$Accuracy\ of\ Linear\ kernel:\ 95.32\%$$

$$Accuracy\ of\ RBF\ kernel:\ 94.74\%$$

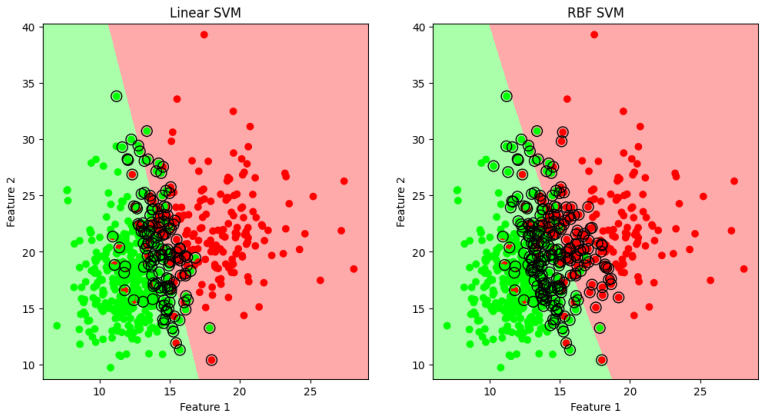

图 3-1 线性核函数与高斯核函数相对比

实验结果表明，高斯核函数在本数据集上的表现优于线性核函数，具有更高的准确率，这说明高斯核函数能够更好地捕捉数据的非线性特征，提高分类性能。

4 支持向量机算法的应用

支持向量机算法可以有效地处理高维数据和非线性数据，具有很高的准确性和泛化能力，在模式识别、生物信息学、金融预测、医学检测、计算机视觉、文本分类等领域都有广泛应用。

4.1 模式识别

支持向量机在模式识别领域的应用最广泛，已成功地解决了诸如手写体、图像处理、语音识别等许多识别和分类问题。

在手写字体识别方面，当采用 5 层神经网络算法时，其识别的错误率为 5.1%；贝尔实验室 [2] 最先将 SVM 应用于手写字体识别研究，选取三种不同的核函数时，得到的误识率分别为 4.0%，4.1% 和 4.2%，可看出支持向量机方法比神经网络算法具有更好的分类准确性。

在人脸识别方面，使用 C-SVC 和 nu-SVC 支持向量机模型, 对人脸进行识别，在叶晓波等 [11] 的实验验结果显示,nu-SVC 模型的人脸识别正确率高, 且波动性小, 更适合用于人脸识别,SVM 分类器用于人脸识别效果较好。

在语音识别方面，由于背景环境中存在不同程度的噪杂声，熊卫华 [12] 等设计了一种基于集总经验模态分解 (EEMD) 和最小二乘支持向量机 (LSSVM) 的语音识别算法，该方法能够快速有效地识别各种语音信号, 与 EEMD 结合 BP 神经网络方法相比, 识别准确率更高, 抗干扰能力更强。

4.2 金融预测

随着经济水平的提高, 越来越多的人投入到了股票预测的研究之中, 在众多的股票预测方法中, 支持向量机对于回归预测也有很强大的效果。张磊 [13] 通过主成分分析降维, 让支持向量机模型在略微的预测正确率损失情况下其运行速度得到巨大的提升, 同时发现支持向量机对于预测股票的最佳时间滑窗为 3, 支持向量分类和回归的结合能提升模型 3%-4% 的预测准确率。

4.3 文本分类

利用计算处理系统处理文本信息, 能够有效提升文本分类的质量与效率, 提升数据信息的利用率, 从而促进信息化技术的普及 [14]。何铠等 [15] 在传统卷积神经网络模型的基础上提出了一种基于卷积神经网络和支持向量机结合的文本分类模型 CNNSVM, 使用基于支持向量机的分类器替代传统模型中的 softmax 层帮助实现文本的分类。该模型提升了特征词语的提取效果，有效解决了 softmax 层泛化能力较弱的问题。

5 结论与展望

SVM 以统计学习理论为基础，存在全局优化、泛化性能好等优点，同时也存在诸多缺陷，有很多问题需深入研究：

1. SVM 算法对大规模训练样本难以实施，这是因为支持向量算法借助二次规划求解支持向量，这其中会设计 m 阶矩阵的计算，所以矩阵阶数很大时将耗费大量的机器内存和运算时间。这也限制了 SVM 在大数据领域的应用。
2. SVM 对非线性问题没有通用解决方案，有时候很难找到一个合适的核函数。核函数的选择需要根据数据的特点和经验进行调整，没有一个统一的标准。核函数的选择也会影响 SVM 的性能和效果。
3. SVM 对缺失数据敏感，对参数和核函数的选择敏感。SVM 需要对数据进行预处理，如归一化、去噪等，否则会影响分类结果。SVM 的参数如正则化参数 C、核函数参数 gamma 等也需要通过交叉验证等方法进行调优，否则会导致过拟合或欠拟合。
4. SVM 的模型解释性不强，难以理解为什么会做出某种预测。SVM 的模型涉及到高维空间的映射和超平面的构造，不容易直观地展示给用户。SVM 也没有提供特征选择或权重分配的方法，不容易分析特征对分类结果的影响。

目前，SVM 仍然存在很多问题需进一步的研究，可将 SVM 与离散余弦变换、小波包分解、主元分析、独立分量分析、聚类、粗糙集理论、深度神经网络等方法结合 [6]，提高应用效果并不断探索 SVM 新的应用领域。

参考文献

[1] 张学工. 关于统计学习理论与支持向量机 [J]. 自动化学报. 2000, 26(1):32–42.

[2] Cortes C, Vapnik V. Support-vector networks[J]. Machine learning. 1995, 20:273–297.

[3] 祁亨年. 支持向量机及其应用研究综述 [J]. 计算机工程. 2004, 30(10):4.

[4] Blumer A, Ehrenfeucht A, Haussler D, et al. Learnability and the vapnik-chervonenkis dimension[J]. Journal of the ACM (JACM). 1989, 36(4):929–965.

[5] 顾亚祥, 丁世飞. 支持向量机研究进展 [J]. 计算机科学. 2011, 38(002):14–17.

[6] 张松兰. 支持向量机的算法及应用综述 [J]. 江苏理工学院学报. 2016, (2):14–17.

[7] 胡春, 胡文, 李圣华. 支持向量机研究综述 [J]. 知识窗. 2018, (24):1.

[8] Sebald D J, Bucklew J A. Support vector machine techniques for nonlinear equalization[J]. IEEE transactions on signal processing. 2000, 48(11):3217–3226.

[9] 萧嵘, 王继成. 支持向量机理论综述 [J]. 计算机科学. 2000, 27(3):1–3.

[10] Vapnik V. The nature of statistical learning theory[M]. [S.l.]: Springer science & business media, 1999.

[11] 叶晓波, 秦海菲, 吕永林. 基于支持向量机的人脸识别应用研究 [J]. 楚雄师范学院学报. 2019, 3.

[12] 熊卫华, 梁坤. 基于最小二乘支持向量机的含噪语音识别算法 [J]. 工业控制计算机. 2018, 31(10):86–88.

[13] 张磊. 基于支持向量机的股票市场趋势分析及预测研究 [D]. 南京邮电大学: 南京邮电大学, 2020.

[14] 何焱. 文本分类中支持向量机研究 [J]. 河南科技. 2019, 29.

[15] 何铠, 管有庆, 龚锐. 基于深度学习和支持向量机的文本分类模型 [J]. 计算机技术与发展. 2022.

附录 A 算法

A.1 代码

代码 A-1 实验代码

language

```
from sklearn.svm import SVC
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap

# 导入肺癌数据集
data = load_breast_cancer()
X = data['data']
y = data['target']
print(X.shape)

x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

# 建立支持向量机模型
clf1 = SVC(kernel='linear')    #线性核函数
clf2 = SVC(kernel='rbf', C=10, gamma=0.0001) #采用高斯核函数
clf1.fit(x_train, y_train)
clf2.fit(x_train, y_train)
from sklearn.metrics import accuracy_score

# 对线性核函数模型进行预测并计算准确率
y_pred_linear = clf1.predict(x_test)
accuracy_linear = accuracy_score(y_test, y_pred_linear)
print("Accuracy of linear kernel: {:.2f}%".format(accuracy_linear * 100))

# 对高斯核函数模型进行预测并计算准确率
y_pred_rbf = clf2.predict(x_test)
accuracy_rbf = accuracy_score(y_test, y_pred_rbf)
print("Accuracy of RBF kernel: {:.2f}%".format(accuracy_rbf * 100))

# 导入numpy模块
import numpy as np

```

```
35 # 定义一个函数，用于绘制支持向量机的决策边界
36 def plot_svm_decision_boundary(clf, X, y, title):
37     # 获取数据集的最小值和最大值
38     x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
39     y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
40     # 生成网格点
41     xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.02),
42                          np.arange(y_min, y_max, 0.02))
43     # 预测网格点的类别
44     Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
45     # 将预测结果reshape为网格的形状
46     Z = Z.reshape(xx.shape)
47     # 定义颜色映射
48     cmap_light = ListedColormap(['#FFAAAA', '#AAFFAA'])
49     cmap_bold = ListedColormap(['#FF0000', '#00FF00'])
50     # 绘制背景颜色
51     plt.pcolormesh(xx, yy, Z, cmap=cmap_light)
52     # 绘制数据点
53     plt.scatter(X[:, 0], X[:, 1], c=y, cmap=cmap_bold)
54     # 绘制支持向量
55     plt.scatter(clf.support_vectors_[:,0], clf.support_vectors_[:,1], s=100,
56                 facecolors='none', edgecolors='k')
57     # 设置标题和坐标轴标签
58     plt.title(title)
59     plt.xlabel('Feature 1')
60     plt.ylabel('Feature 2')
61
62 # 只选择数据集中的前两个特征，方便可视化
63 X2 = X[:, :2]
64
65 # 建立两种支持向量机模型，使用前两个特征
66 clf3 = SVC(kernel='linear')
67 clf4 = SVC(kernel='rbf', C=10, gamma=0.0001)
68 clf3.fit(X2, y)
69 clf4.fit(X2, y)
70
71 # 绘制两种支持向量机模型的决策边界
72 plt.figure(figsize=(12,6))
73 plt.subplot(121)
74 plot_svm_decision_boundary(clf3, X2, y, 'Linear SVM')
75 plt.subplot(122)
```

```
76 plot_svm_decision_boundary(clf4, X2, y, 'RBF SVM')
77 plt.show()
```