机器学习实验报告

实验名称：实现支持向量机

学生姓名：陆文韬

学生学号：58122231

完成日期：2024/5/29

任务描述

通过两种方式实现 SVM:

1. 手动实现 SMO 算法，并与直接使用传统二次规划方法进行对比，
2. 通过 scikit-learn 库实现软间隔 SVM。

并在 breast cancer 数据集上进行验证与实验。该数据集是一个二分类问题，属性均为连续属性，并已进行标准化。

教学要求:

1. 掌握硬间隔 SVM 和软间隔 SVM 及其核化方法的原理
2. 掌握 SMO 算法
3. 了解基于 Python 语言的 scikit-learn（sklearn）库，掌握本实验涉及的相关部分
4. 进行参数选择以及参数分析实验，理解正则化常数 C 以及核函数的影响

1. 手动实现 SVM 的 SMO 算法

SVM 的对偶问题实际是一个二次规划问题，除了 SMO 算法外，传统二次规划方法也可以用于求解对偶问题。求得最优拉格朗日乘子后，超平面参数 $\boldsymbol{w}, b$ 可由以下式子得到:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i x_i$$

$$b = \frac{1}{|S|} \sum_{s \in S} \left(\frac{1}{y_s} - \sum_{i \in S} \alpha_i y_i \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_s \right).$$

请完成以下任务:

(1) [10pts] 不考虑软间隔情况，直接使用传统二次规划（QP）方法求解（实现）训练集上的硬间隔 SVM 对偶问题。观察并回答，这样的求解方法是否会出现问题，为什么?

(2) [40pts] 不限定硬间隔或是软间隔 SVM，也不限定是否为核化 SVM，根据需要选择合适的方法，手动实现 SMO 算法[1]求解 SVM 对偶问题。注意第 3 步，KKT 条件验证步骤不能缺少。

(3) [10pts] 对测试数据进行预测，确保预测结果尽可能准确。

2. 使用 sklearn 库简洁实现软间隔 SVM

(1) [20pts] 使用 sklearn 库简洁实现软间隔 SVM。首先实现以下 4 个示例性的 SVM 模

[1] SMO 算法的详细步骤可以参考"王贝伦《机器学习》"4.7.2.2 节 P169。参见实验文档中的相关 pdf 文件。

型：

- 线性 SVM：正则化常数 C=1，核函数为线性核，
- 线性 SVM：正则化常数 C=1000，核函数为线性核，
- 非线性 SVM：正则化常数 C=1，核函数为多项式核，d=2，
- 非线性 SVM：正则化常数 C=1000，核函数为多项式核，d=2，

观察并比较它们在测试集上的性能表现。

(2) [20pts] 参数选择与参数分析

参数的选择对 SVM 的性能有很大的影响。确定正则化常数 C 与核函数及其参数的选择范围（可以在以下常用核函数中选择一种或多种），选用合适的实验评估方法（回顾第 2 章的内容，如 K 折交叉验证法等）进行参数选择，并进行参数分析实验。

- 正则化常数 C 的选择范围可以是以下数量级，如：$10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3$
- 核函数
  - 线性核：$k(x,y) = x^T y$，无参数，退化为 SVM 基本型
  - 多项式核：$k(x,y) = (x^T y + c)^d$，
    - c: 偏移量，通常取值 0 或 1
    - d: 多项式的次数，通常取值 2, 3, 4 等较小的整数，取值为 1 时退化为线性核
  - 多项式核另外一种常用的形式：$k(x,y) = (\gamma * x^T y + c)^d$
    - γ: 用于控制核函数的影响范围，常用值为 1/n_features, n_features 是属性个数，或者也可以尝试 0.001, 0.01, 0.1, 1, 10 等
    - c: 偏移量，通常取值 0 或 1
    - d: 多项式的次数，通常取值 2, 3, 4 等较小的整数，取值为 1 时退化为线性核
  - 高斯核/RBF 核：$k(x,y) = \exp(-\gamma * \|x-y\|^2)$
    - γ：带宽参数，控制核函数的平滑程度。常用取值范围为10^{-3}到10^3。典型值为 1/(n_features* X.var())，其中 X.var()表示数据集 X 的方差，这可以确保 γ 的值在数据特征的数量和数据的方差范围内适当缩放。
  - Laplace 核：$k(x,y) = \exp(-\gamma * \|x-y\|)$
    - γ：带宽参数，控制核函数的平滑程度。常用取值范围参考高斯核的 γ 值
  - Sigmoid 核函数：$k(x,y) = \tanh(\gamma * x^T y + c)$
    - γ：缩放因子，常用取值范围为10^{-3}到10^3
    - c: 偏移量，通常取值 0 或 1

请注意，SVM 中的参数通常需要联合选择，特别是对于核函数的参数与正则化参数 C。参数联合优化的策略有：

(1) 网格搜索：网格搜索是一种穷举搜索法，在给定的参数空间中逐一尝试每种可能的

参数组合，并选用合适的评估方法进行评估。

(2) 随机搜索：在参数空间中对参数组合的随机采样进行评估。通常可以在较大的参数空间中找到接近最优的参数组合。步骤如下：
- 定义参数的取值范围
- 随机采样若干参数组合
- 对每个采样的参数组合进行评估
- 选择性能最优的参数组合

(3) 贝叶斯优化：贝叶斯优化是一种更为高级的优化方法，通过构建一个模型来估计参数空间的性能分布，并通过最大化预期改进（Expected Improvement, EI）等准则来选择下一组参数进行评估。其步骤如下：
- 初始采样若干参数组合并进行评估
- 构建模型（如高斯过程回归）
- 基于模型选择下一组参数进行评估
- 更新模型并重复第 3 个步骤，直到达到预定的评估次数或满足停止条件。

本实验中可以选择较为简单的网格搜索策略或随机搜索策略进行调参。

参数分析实验需要报告：

- 评估方法
- 参数取值范围
- 搜索策略
- 比较分析每组参数上的性能情况
- 最终选择的最优参数

实验原理

Breast Cancer Dataset

“Breast Cancer Wisconsin (Diagnostic)”数据集是一种常用于医学图像处理和机器学习研究的公开数据集，它由威斯康星大学医院的 Dr. William H. Wolberg 创建。这个数据集包含了乳腺癌细胞的样本，旨在帮助研究者开发能够识别良性与恶性肿瘤的算法。

数据集的主要特征如下：

1. ID 号：每个样本的唯一识别号。

2. 诊断结果：肿瘤的性质（M = 恶性，B = 良性）。

3. 细胞核特征：包括以下 10 个真实值特征，每个特征计算得出的平均值、标准误差和“最差”或最大（平均的三个最大值）：

- 半径（距离中心到周边点的平均距离）

- 纹理（灰度值的标准偏差）

- 周长
- 面积
- 平滑度（半径长度的局部变异）
- 紧凑度（周长^2 / 面积 - 1.0）
- 凹度（轮廓的凹部分的严重程度）
- 凹点（凹部边缘的点数量）
- 对称性
- 分形维数（海岸线近似 - 1）

数据集通常被用于训练监督学习模型，如支持向量机（SVM）、随机森林、神经网络等，以预测乳腺肿瘤的恶性或良性。这个数据集广泛应用于教育和研究中，因为它允许研究人员开发和测试不同的机器学习技术和方法。

SMO 算法

SMO 算法（Sequential Minimal Optimization，序列最小优化）是一种用于训练支持向量机（SVM）的有效算法，由 John Platt 于 1998 年提出。SMO 算法主要用于解决 SVM 的二次规划（QP）问题，特别是在处理大规模数据集时，相比传统的 QP 求解器，SMO 可以更快地收敛到解。

基本原理

在训练 SVM 时，需要解决的核心问题是一个有约束的优化问题，即最小化目标函数（通常是权重向量的平方和），同时满足一些不等式约束（即数据点必须正确分类）。直接求解这个问题通常需要大量的计算资源，尤其是当数据集很大时。

SMO 算法的核心思想是将大规模的 QP 问题分解为一系列最小化的问题，每次只优化两个变量（拉格朗日乘子），而固定其他变量。这种方法可以显著降低问题的复杂性，因为每次处理的只是二维优化问题。

工作流程

1.选择变量：SMO 算法每次循环选取两个拉格朗日乘子进行优化。选择的标准是这两个乘子能最大化整个系统的优化步骤。通常选择违反 KKT（Karush-Kuhn-Tucker）条件最严重的乘子作为第一个乘子。

2. 优化这两个变量：一旦选定两个乘子，SMO 求解对应的二维优化问题。在这个步骤中，会考虑到这两个乘子的约束条件，即它们的和是固定的，并且它们必须在一个特定的区间内。

3. 更新乘子：求解得到最优解后，更新这两个乘子的值。

4. 重复步骤：重复选择和优化直到所有的乘子满足 KKT 条件，或者整体优化进展非常小。

KKT 条件

KKT 条件是解决优化问题时需满足的一组必要条件，用于检查当前解是否为最优解。对于 SVM，KKT 条件包括：

- 如果乘子为正，那么对应的约束条件应该是活跃的，即对应的样本点正好在边界上。

- 如果样本点被正确分类，并且在边界之外，对应的乘子应该是 0。

优势与限制

优势：

相较于传统的 QP 求解器，SMO 不需要存储整个核矩阵，因此在处理大规模数据集时更为高效。

SMO 的简单性和效率使其成为了训练 SVM 的常用算法之一。

限制：

SMO 的性能严重依赖于合适的核函数和参数选择。

在某些特别设计的数据集或特定类型的核函数上，SMO 的收敛速度可能比较慢。

SMO 算法因其在实际应用中的高效性而广泛应用于支持向量机的训练，特别是在机器学习和数据挖掘领域。

实验设置

核函数

在支持向量机（SVM）中，核函数用来将输入数据映射到一个更高维的空间中，以便找到可以有效分割数据的超平面，特别是在原始数据空间中不可分的情况下。核函数允许我们在高维空间中进行计算，而无需显式地处理这个空间的坐标，这是通过计算原始输入向量的核函数来实现的。以下是一些常用的核函数及其特点：

1. 线性核（Linear Kernel）

线性核是最简单的核函数，直接计算两个向量的内积。它适用于线性可分的数据集。使用线性核的 SVM 相当于没有将数据映射到更高的维度。

2. 多项式核（Polynomial Kernel）

多项式核可以将数据映射到高维空间，适用于非线性可分的数据。参数 d 控制了多项式的阶数，gamma 是缩放数据的比例，r 是核函数中添加的常数项。通过调整这些参数，可以控制高维空间的复杂度。

3. 径向基函数核（Radial Basis Function Kernel，RBF）

RBF 核，也称为高斯核，是一种非常流行的核函数，用于处理不同类别之间有复杂边界的数据。它通过高斯函数的形式考虑两个样本点之间的距离，gamma 控制了函数的宽度。RBF 核可以处理数据在原始空间中的非线性关系。

4. Sigmoid 核

Sigmoid 核将输入向量的内积通过 Sigmoid 函数转换，类似于神经网络中的激活函数。这种核函数曾经被用来模拟两层神经网络的行为。然而，Sigmoid 核可能不总是满足核函数的正定要求，因此在实际使用中要小心选用。

应用选择

选择合适的核函数依赖于数据集的特性：

- 如果数据是线性可分的，线性核是一个好的选择，因为它简单且计算效率高。

- 如果数据是非线性的，多项式核和 RBF 核可以提供更好的模型灵活性。

- Sigmoid 核可能适用于某些特定的数据类型，但通常不推荐作为首选。

总体来说，核函数的选择和参数调整需要通过交叉验证和实验来确定，以确保模型既能很好地拟合数据，也能良好地泛化到新数据上。

惩罚系数 C

在支持向量机（SVM）中，惩罚系数 C 是一个非常重要的参数，它用来控制模型对误分类数据的容忍程度，以及模型整体的复杂性。这个参数在软间隔 SVM 中特别关键，用于处理那些不能被完美线性分割的数据集。

在理想情况下，硬间隔 SVM 寻求一个能够无误差地将不同类别完全分开的决策界面。但在现实中，由于数据可能包含噪声或本身就不是完全线性可分的，硬间隔 SVM 可能不适用。这时，模型可能会过分强调完全匹配训练数据，导致在新数据上表现不佳。

软间隔 SVM 通过引入所谓的松弛变量来解决这个问题。这些松弛变量允许某些数据点在一定程度上违反分割界面的完美分类要求，从而为模型提供一定的灵活性。

在软间隔 SVM 的目标中，我们不仅需要最小化与间隔大小相关的一项（通常与模

型的复杂度相关），还需要控制松弛变量的总和，以限制数据点违背完美分类的程度。惩罚系数 C 就是用来平衡这两个目标的参数：

- 高 C 值：对于分类错误的数据点施加高惩罚，这会使模型更加努力地匹配所有训练数据（包括异常值和噪声），可能导致过拟合。

- 低 C 值：减少对分类错误的惩罚，允许更多的误分类，这有助于提高模型对新数据的适应能力，从而可能减少过拟合的风险。

选择合适的 C 值通常通过交叉验证来进行。在交叉验证中，通过尝试不同的 C 值来寻找在训练数据上表现良好且在未见数据上也能保持合理表现的最佳平衡点。

惩罚系数 C 在软间隔 SVM 中扮演着决定性角色，影响模型对数据异常和噪声的敏感性。选择正确的 C 是确保模型有良好泛化能力的关键步骤。

交叉验证

交叉验证是一种统计方法，用于评估机器学习模型的泛化能力，即模型对新数据的处理能力。这种方法通过在不同的子集上重复地训练和验证模型来实现，目的是确保模型不仅在训练数据上表现良好，也能在未见过的数据上保持稳定的表现。这样可以避免模型过拟合或欠拟合的问题。

主要类型的交叉验证：

1. K-折交叉验证（K-Fold Cross-Validation）

在这种方法中，数据集被分割成 K 个大小相等的子集。每次，其中的一个子集被用作验证集（测试模型的性能），而其余的 K-1 个子集合并后用作训练集。这个过程重复 K 次，每次选择不同的子集作为验证集。

通过这种方式，每个数据点都有一次机会作为验证集中的一部分，从而使得模型评估更加全面和准确。

2. 留一交叉验证（Leave-One-Out Cross-Validation, LOOCV）

这是 K-折交叉验证的一个特殊情况，其中 K 等于样本总数。这意味着每次留下一个样本作为验证集，其余的所有样本用作训练集。虽然这种方法在小数据集上可以非常精确，但计算成本高，特别是在大数据集上。

3. 随机划分交叉验证（Shuffle-Split Cross-Validation）

在这种方法中，数据集被随机划分成训练集和验证集。与 K-折交叉验证不同，这种划分可以重复多次，但每次的划分可以完全不同。这提供了更多的随机性，并可以更灵活地控制训练集和验证集的比例。

优点：

降低偏差：由于模型需要在多个独立的数据子集上表现良好，交叉验证可以帮助降低模型评估的偏差。

减少过拟合：通过在多个训练集上训练并在多个独立的验证集上评估，模型的泛化能力通常会增强，从而减少过拟合。

更加稳健：交叉验证提供了对模型性能的全面评估，这有助于理解模型在不同类型的数据上的表现。

缺点：

时间和计算成本：尤其是在数据集较大或模型较为复杂时，执行多轮训练和验证可能非常耗时。

数据分布敏感性：如果数据集中的数据分布不均匀，交叉验证可能会给出误导性的结果。因此，有时需要先进行数据的洗牌和分层处理。

交叉验证是机器学习实践中一个非常有用的工具，它帮助研究人员和工程师验证他们的模型在实际应用中的可靠性和有效性。通过适当的交叉验证，可以显著提升模型的鲁棒性和信度。

实验结果

一、(1)

在实际应用的情况中，用 QP 算法直接求解 SVM 会遇到许多的问题:

1. 计算复杂度

规模问题：在硬间隔 SVM 的对偶形式中，问题的规模与训练样本的数量直接相关。每增加一个数据点，就相应增加一个拉格朗日乘子作为优化变量。因此，对于大规模数据集，这将导致 QP 问题规模大幅增加，处理成百上千甚至更多变量。

内存需求：传统的 QP 求解器通常需要构建和存储一个大型的 Hessian 矩阵（即所有样本间的内积或核函数计算结果的矩阵）。对于大数据集，这种需求迅速增加，可能需要巨大的内存和存储空间。

2. 数值稳定性和精确性

数值问题：处理大规模优化问题时，数值精度可能成为关键问题。这是因为在优化过程中，求解器可能需要处理非常小的数值，或者由于累积的计算误差导致数值不稳定。

收敛问题：对于条件数差或者非常复杂的 QP 问题，找到全局最优解的过程可能非常缓慢，求解器甚至可能无法保证收敛。

3. 求解器限制

通用性与特殊性：传统的 QP 求解器虽然设计得很通用，但它们可能不会针对 SVM 问题的特定结构进行优化。相比之下，专门为 SVM 设计的算法（如 SMO 算法）能够利用问题的特定结构（如只需处理拉格朗日乘子的优化），从而在计算效率和资源使用上更加高效。

不适用于线性不可分的问题：采用硬间隔的方法求解问题的过程中，目标是找到一个可以完全分割不同类样本点的超平面，但是对于有噪声的数据集和一些线性不可分的数据集来说，这样的超平面是不存在的。同时硬间隔 SVM 对于异常值的存在极为敏感，可能会导致超平面的位置和期望值发生比较大的偏移。

4. 算法效率

算法特化：SVM 特化算法，如 SMO，通过分解原问题为更小的子问题来显著提高求解速度。它们通过优化一小部分变量的策略，可以更快地找到解决方案，而不是一次性解决所有变量的 QP 问题。

优化策略：传统 QP 求解器缺乏这种分解大问题为小问题的策略，它们通常在全局层面上处理所有变量，这在大数据集上可能非常低效。

```
 -1.26478335e-10 -2.63498709e-11  3.21172613e+03  4.32202748e-11
 -1.79639965e-10 -1.63919937e-11 -2.15719890e-11 -9.67538099e-11
 -1.30083778e-11 -1.46597776e-10 -6.48984658e-11 -3.72013964e-11
  1.84973954e+04 -6.54643944e-11 -6.60176379e-12 -7.73503951e-11
  9.12383634e-11 -1.29481155e-10 -6.74565356e-11 -1.10558061e-10
 -1.17194396e-10 -1.27515329e-10 -1.44574639e-10  6.91119040e-12
 -2.07014539e-10 -4.45073744e-11 -6.13189256e-11 -5.13641184e-11
  1.96728987e-10 -1.69318212e-10  1.11644979e+04  4.41045808e-11
 -7.86787404e-11 -8.20549129e-11 -6.15908712e-11 -4.35781108e-13
  4.06331259e+03 -1.08567060e-10 -1.13589499e-10  1.62542260e-10
 -7.11197624e-11  1.32727141e-11  1.49556849e+03  7.34601505e-10
 -2.24060682e-11 -9.38146181e-11 -3.83504849e-11 -1.18476526e-10
 -3.32326205e-11 -6.46775808e-11 -2.75136302e-11  5.11947548e-11
 -8.07866922e-11 -5.24316070e-11 -6.81785814e-11 -1.81761718e-10
 -1.37193830e-10  1.19577501e-10 -1.40665604e-10 -1.12193663e-10
 -7.94630238e-11  1.06859928e+04 -6.94933546e-11 -4.04073624e-11
 -1.27486464e-10 -1.13751631e-10 -1.49199547e-10  1.59150125e-10
 -6.86467057e-11 -1.69693518e-10  5.26566336e-11 -4.04198594e-11
  5.46698777e-11 -1.03675707e-10 -4.91893076e-11 -1.04300322e-10
 -1.29644670e-10 -1.47488044e-10 -9.83888067e-11 -9.85004113e-11
  1.12004426e-10 -9.57239571e-11  5.07838859e-11  1.20085219e-12
  8.14754862e+03  7.70682673e-10 -1.53319385e-10 -4.39022761e-11
 -5.16155686e-11 -2.66233288e-10 -1.40864967e-11 -1.33951307e-10
 -2.27797482e-11 -3.67164417e-11 -1.03131733e-10  5.24970268e+03
 -7.60776610e-11 -7.63562015e-11 -2.05595969e-10  1.42863623e-10
 -6.79868821e-11  1.28922588e-11 -3.38018300e-11]
accuracy:  0.9385964912280702
```

一、(2)

在这个程序中，我们采用线性核和软间隔来手动实现 smo 算法，最终的预测的准确率为 98.23%，同时打印出了不符合 KKT 条件的样本。出现个别不符合 KKT 条件的样本是完全合理的，因为如果数据集不是完全线性可分的，在采用软间隔 SVM 的时候，就会出现个别的点被误分类，这些点的数量只要在设定的 tolerant 阈值之内，就是合理的。

```
KKT条件违反的样本及其情况:
样本 155: 1 - toler <= yi*fxi <= 1 + toler not met, 0 < alpha < C
样本 246: 1 - toler <= yi*fxi <= 1 + toler not met, 0 < alpha < C
样本 394: 1 - toler <= yi*fxi <= 1 + toler not met, 0 < alpha < C
准确率:  0.9823008849557522
```

二、(1)

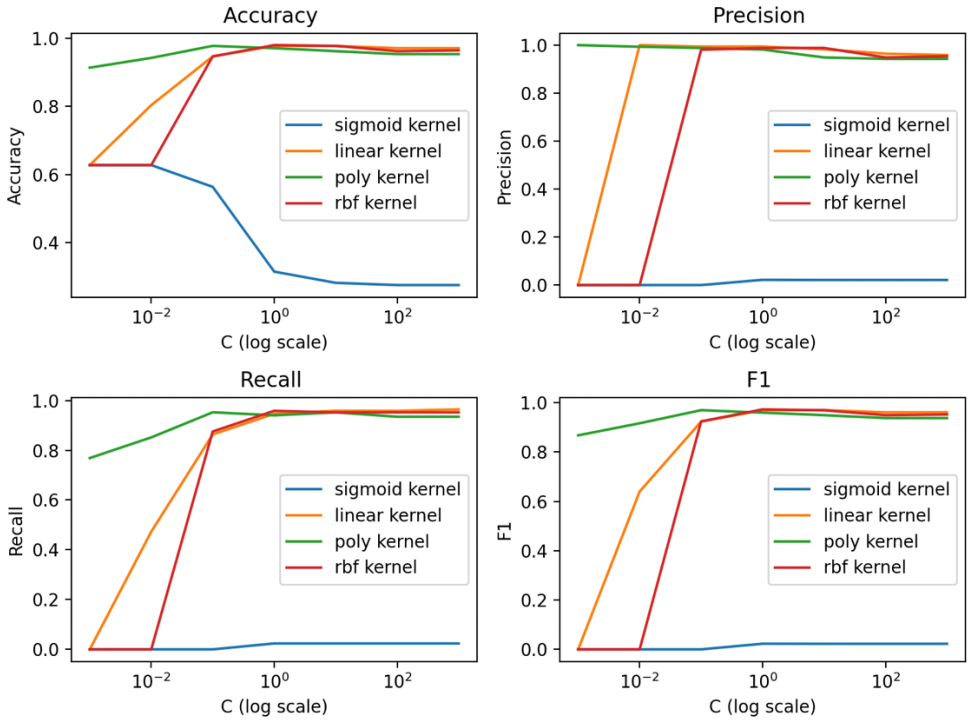
在本程序中，使用了四种不同的 SVM 模型:

- 线性 SVM：正则化常数 C=1，核函数为线性核，
- 线性 SVM：正则化常数 C=1000，核函数为线性核，
- 非线性 SVM：正则化常数 C=1，核函数为多项式核，d=2，
- 非线性 SVM：正则化常数 C=1000，核函数为多项式核，d=2

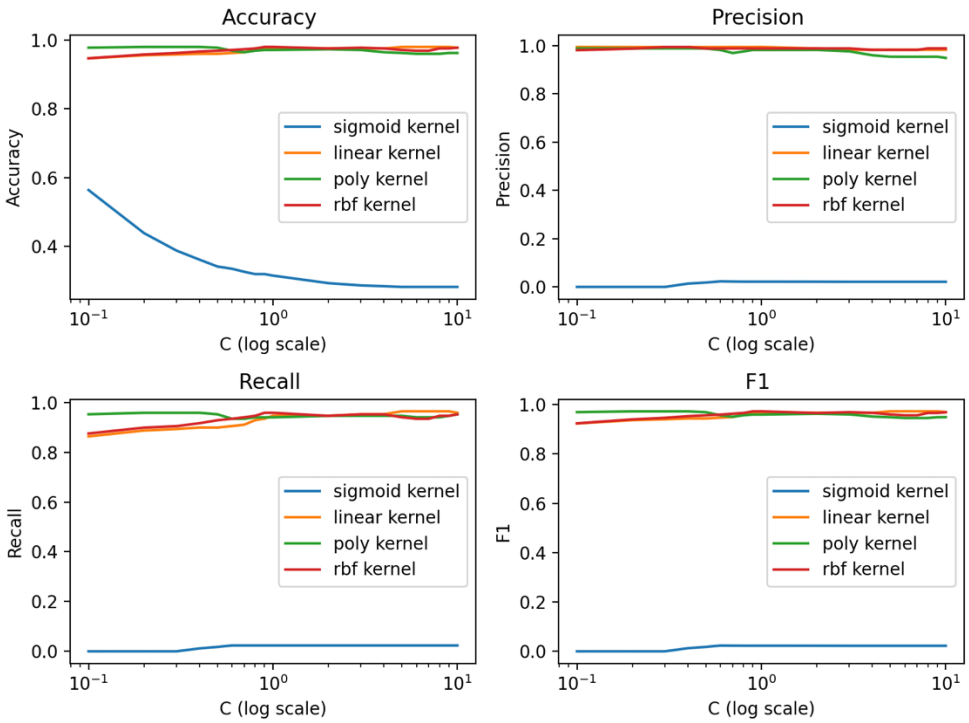
```
Linear SVM C=1:
  Accuracy = 0.98
  Precision = 1.00
  Recall = 0.95
  F1 Score = 0.98
Linear SVM C=1000:
  Accuracy = 0.95
  Precision = 0.91
  Recall = 0.95
  F1 Score = 0.93
Polynomial SVM C=1 d=2:
  Accuracy = 0.98
  Precision = 1.00
  Recall = 0.95
  F1 Score = 0.98
Polynomial SVM C=1000 d=2:
  Accuracy = 0.94
  Precision = 0.89
  Recall = 0.95
  F1 Score = 0.92
```

在该程序中，我们采用了多种指标，全面详细地分析了不同 SVM 模型的性能，从数据中我们不难发现，惩罚系数 C=1 的模型明显优于惩罚系数 C=1000 的模型。在 C=1 的时候，线性核和多项式核的性能非常接近，但是在 C=1000 的时候可以看到，线性核的性能要稍微优于多项式核。

二、(2)



在 C=0.001 到 1000 的尺度上观察，我们发现 sigmoid 核的性能在各项指标上都远远不如其他三个核心，因此在进一步的比较中，我们不再考虑 sigmoid 核。



通过缩小尺度进一步观察我们发现，三个核心在 C=1 的附近具有更好的表现。

在处理乳腺癌数据集时，选择更重要的指标取决于具体的应用场景和业务需求。对于乳腺癌检测，通常关注以下几个指标：

1. **Recall (Sensitivity or True Positive Rate)**：对于癌症检测，Recall 是一个非常重要的指标，因为它反映了模型能正确识别出患病个体的能力。高 Recall 意味着较少的假阴性（即实际患病但被预测为健康的病例）。假阴性在癌症检测中可能导致严重的后果，因此通常优先关注 Recall。
2. **Precision**: Precision 反映了模型预测为阳性（患病）的个体中实际为阳性个体的比例。高 Precision 意味着较少的假阳性（即实际健康但被预测为患病的病例）。在某些情况下，如资源有限的医疗环境中，高 Precision 可以减少不必要的进一步检查和治疗。
3. **F1 Score**: F1 Score 是 Precision 和 Recall 的调和平均，它在 Precision 和 Recall 之间寻找平衡。如果没有明确的偏好，F1 Score 是一个很好的综合指标，尤其是在类别不平衡的情况下。
4. **Accuracy**：虽然 Accuracy 是一个常用的指标，但在类别不平衡的情况下（如癌症检测中健康人群远多于患病人群），高 Accuracy 可能是由于正确预测了大多数健康个体，而忽略了对患病个体的检测。因此，Accuracy 在这种情况下可能不如 Recall 或 Precision 重要。

在乳腺癌检测中，**Recall** 通常是最重要的指标，其次是 Precision 和 F1 Score。具体的优先级应根据应用场景进行调整。因为作为癌症检测的目标是尽可能减少漏诊（假阴性），所以 Recall 是关键指标，我们可以选择 C=1，核函数为 rbf 核进行计算。

在确定采用 rbf 核函数后，我们采用网格搜索（GridSearchCV）来找到最佳的 RBF 核函数的带宽参数 gamma，得到最佳的带宽参数为 1.0。

```
Best gamma: 1.0
Best model performance on test set:
  Accuracy = 0.97
  Precision = 0.98
  Recall = 0.95
  F1 Score = 0.96
```