机器学习实验报告

实验名称：实现 SVM

学生姓名：蒋雨初

学生学号：58121102

完成日期：2023/6/14

任务描述

通过两种方式实现 SVM：

任务一

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) [15pts] 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。

任务二

使用 sklearn 库简洁实现软间隔 SVM

[20pts]（1）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。

[20pts]（2）参数选择与参数分析

确定正则化常数 C 与核函数及其参数的选择范围，选用合适的实验评估方法（回顾第 2 章的内容）进行参数选择，并进行参数分析实验。

实验原理

Breast cancer 数据集

Breast Cancer 数据集是机器学习中常用的经典数据集之一，用于预测乳腺癌的发生。该数据集包含了一系列乳腺肿块的特征信息，以及它们的良性（benign）或恶性（malignant）的标签。

这个数据集最早由威斯康辛州大学的 Wolberg、Street 和 Mangasarian 等人在 1992 年创建，并在机器学习领域广泛应用。它基于乳腺肿块的细胞核图像特征，包括肿块的大小、形状、质地等。具体特征包括以下内容：

1. 半径（radius）：从肿块中心到边界的距离的平均值。
2. 纹理（texture）：灰度级别的标准差。
3. 周长（perimeter）：肿块周长的长度。
4. 面积（area）：肿块的面积。
5. 光滑度（smoothness）：周长的局部变化。
6. 紧凑度（compactness）：周长^2 /面积-1.0。
7. 凹陷（concavity）：轮廓的凹陷部分的严重程度。
8. 凹点（concave points）：凹陷的数量。
9. 对称性（symmetry）：轮廓的对称性。
10. 分形维度（fractal dimension）：海岸线近似-1。

每个样本的特征值都被计算为数字，这些数字用来描述细胞核的形态和特性。而目标变量是一个二元标签，表示肿块是良性（benign）还是恶性（malignant）。

使用这个数据集，研究人员和机器学习从业者可以构建模型来对新的乳腺肿块进行分类预测，从而帮助医生进行初步的诊断。常见的机器学习算法如决策树、支持向量机、逻辑回归等可以应用于这个数据集，并利用已知的特征和标签进行训练，从而使模型能够根据给定的特征预测肿块的良恶性。

Breast Cancer 数据集是一个广泛使用的数据集，对于机器学习算法的评估和比较以及特征选择和降维等任务都非常有用。它提供了一个实际且具有挑战性的问题领域，可以帮助研究人员和从业者更好地理解和解决乳腺癌诊断的问题。

QP 求解硬间隔 SVM

先给出一些基本定义：对于给定训练集 T 和超平面(w,b)，定义该超平面关于样本点(x_i,y_i)的函数间隔为$\hat{\gamma}_i = y_i(\boldsymbol{w}^{\mathrm{T}}x_i + b)$。再定义所有样本点函数间隔的最小值为$\hat{\gamma} = \min\limits_{\mathrm{i=1,2,\cdots N}} \hat{\gamma}_i$。如果对$\hat{\gamma}_i$进行规格化，那么得到几何间隔和最小几何间隔的定义

$$\gamma_i = y_i\left(\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}x_i + \frac{b}{\|\boldsymbol{w}\|}\right), \gamma = \min_{\mathrm{i=1,2,\cdots N}} \frac{\hat{\gamma}_i}{\|\boldsymbol{w}\|}$$

SVM 的优化目标是寻找最大几何间隔的能划分正确数据集的超平面，即

$$\max_{\mathrm{w,b}} \frac{2\hat{\gamma}}{\|\boldsymbol{w}\|}$$
$$s.t.\ y_i(\boldsymbol{w}^{\mathrm{T}}x_i + b) \geq \hat{\gamma}$$

考虑到函数间隔$\hat{\gamma}$的取值不影响最优化问题的解，所以不妨取$\hat{\gamma} = 1$。且最大化$\frac{2}{\|\boldsymbol{w}\|}$和最小化$\frac{1}{2}\|w\|^2$等价，所以就得到硬间隔 SVM 的最优化问题：

$$\min_{\mathrm{w,b}} \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$s.t.\ y_i(\boldsymbol{w}^{\mathrm{T}}x_i + b) - 1 \geq 0$$

这是一个凸二次规划问题（convex QP）,所以使用拉格朗日乘子法求解。具体来说引入拉格朗日乘子$\alpha_i \geq 0$，写出拉格朗日函数

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{m} \alpha_i\left(1 - y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b)\right)$$

$$= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \sum_{i=1}^{m}\alpha_i - \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{w}^T\boldsymbol{x}_i - \sum_{i=1}^{m}\alpha_i y_i b$$

再分别求 w b 偏导并令为 0,求解

$$\frac{\partial L}{\partial \boldsymbol{w}} = \frac{1}{2}\times 2 \times \boldsymbol{w} + 0 - \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i - 0 = 0 \Longrightarrow \boldsymbol{w} = \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L}{\partial b} = 0 + 0 - 0 - \sum_{i=1}^{m}\alpha_i y_i = 0 \Longrightarrow \sum_{i=1}^{m}\alpha_i y_i = 0$$

把上面的结果代入到$L(\boldsymbol{w}, b, \boldsymbol{\alpha})$中，即可得到原问题的对偶问题

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{\mathrm{i}=1}^{\mathrm{m}}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j$$

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

$$\alpha_i \geq 0 \quad i = 1,2,\ldots,m$$

再把极大问题转为极小问题

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{\mathrm{i}=1}^{\mathrm{m}}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j - \sum_{i=1}^{m}\alpha_i$$

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

$$\alpha_i \geq 0 \quad i = 1,2,\ldots,m$$

解出$\boldsymbol{\alpha}$后，根据 KKT 条件求出 w b

$$\boldsymbol{w} = \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i$$

$$b = y_i - \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$$

即可得到判决函数

$$f(x) = sign(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b)$$

$$= sign\left(\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x} + b\right)$$

SMO 算法求解 SVM

QP 问题具有全局最优解，并且许多最优化算法可以用于这一问题的求解。但是当训练样本容量很大时，这些算法往往变得非常低效。所以，如何高效的实现 SVM 学习就成为一个重要的问题。目前人们已提出许多快速实现算法，例如序列最优化算法（SMO）。

SMO 算法是一种启发式算法，其基本思路是:如果所有变量的解都满足此最优化问题的 KKT 条件(Karush-Kuhn-Tucker conditions)，那么这个最优化问题的解就得到了．因为 KKT 条件是该最优化问题的充分必要条件。否则，选择两个变量，固定其他变量，针对这两个变量构建一个二次规划问题，这个二次规划问题关于这两个变量的解应该更接近原始二次规划问题的解，因为这会使得原始二次规划问题的目标函数值变得更小。重要的是,这时子问题

可以通过解析方法求解，这样就可以大大提高整个算法的计算速度。子问题有两个变量，一个是违反 KKT 条件最严重的那一个，另一个由约束条件自动确定。如此，SMO 第法将原问题不断分解为子问题并对子问题求解，进而达到求解原问题的目的。

使用核技巧，再考虑软间隔，则原优化目标可写为

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{\mathrm{i}=1}^{\mathrm{m}}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \kappa_{ij} - \sum_{i=1}^{m}\alpha_i$$

$$\mathrm{s.t.} \sum_{i=1}^{m}\alpha_i y_i = 0$$

$$0 \le \alpha_i \le C \quad i = 1,2,\dots,m$$

其中，$\kappa_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$。

不失一般性，假设选择的两个变量是α_1, α_2，其他变量α_i $(i = 3,4,\dots,m)$是固定的．于是 SMO 的最优化问题的子问题可以写成:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\kappa_{11}\alpha_1^2 + \frac{1}{2}\kappa_{22}\alpha_2^2 + y_1 y_2 \kappa_{12}\alpha_1\alpha_2 - (\alpha_1 + \alpha_2) + y_1\alpha_1\sum_{i=3}^{m} y_i\alpha_i\kappa_{i1} + y_2\alpha_2\sum_{i=3}^{m} y_i\alpha_i\kappa_{i2}$$

$$\mathrm{s.t.}\ y_1\alpha_1 + y_2\alpha_2 = -\sum_{i=1}^{m}\alpha_i y_i = \varsigma$$

$$0 \le \alpha_i \le C \quad i = 1,2,\dots,m$$

式中忽略了不含α_1, α_1的常数项。

以上最优化问题可以用下面的二维空间中的图形表示。

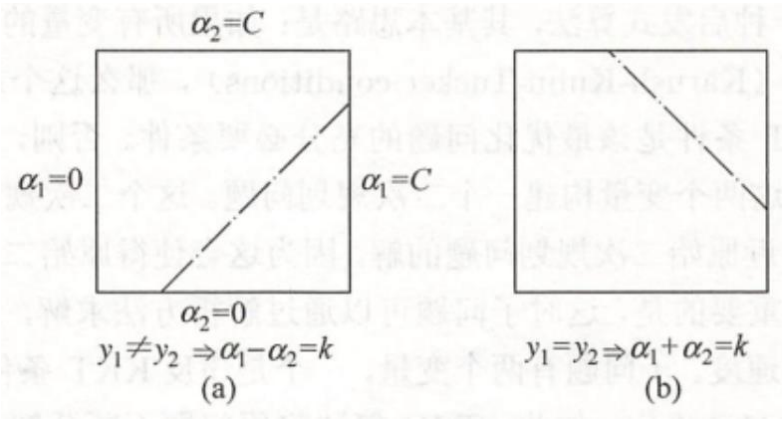


图 1 二变量优化问题图示[1]

不等式约束使得$(\alpha 1, \alpha 2)$在盒子$[0, C] \times [0, C]$内，等式约束使$(\alpha 1, \alpha 2)$在平行于盒子$[0, C] \times [0, C]$的对角线的直线上。这时因为$\alpha 1$和$\alpha 2$所对应的实例的标识可能相同也可能不同，于是需要分为两个情况（即上图中左右两侧），要求的是目标函数在条平行于对角线的线段上的最优值。因为两个变量之间存在关系式，于是二变量的最优化问题实际上是单变量的最优化问题，我们可以将其想成变量α_2的最优化问题。

假设二变量问题的初始可行解为$\alpha_1^{old}, \alpha_2^{old}$，最优解为$\alpha_1^{new}, \alpha_2^{new}$，并且假设在沿着约束方向未经剪辑时$\alpha_2$的最优解为$\alpha_2^{new,unc}$。由于$\alpha_2^{new}$需满足不等式约束$0 \le \alpha_i \le C$，所以最优值$\alpha_2^{new}$的取值范围必须满足条件：

[1] 来自李航《统计学习方法》p122

$$L \leq \alpha_2^{new} \leq H$$

其中，L 与 H 是α_2^{new}所在的对角线段端点的界。

这里的剪辑指，α_2的值可能不满足不等式约束，如果要使其满足不等式约束，需要对其进行剪辑。

如果$y_1 \neq y_2$，则

$$L = \max(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}}), H = \min(C, C + \alpha_2^{\text{old}} - \alpha_1^{\text{old}})$$

如果$y_1 = y_2$，则

$$L = \max(0, \alpha_2^{\text{old}} + \alpha_1^{\text{old}} - C), H = \min(C, \alpha_2^{\text{old}} + \alpha_1^{\text{old}})$$

下面，首先求沿着约束方向未经剪辑即未考虑不等式约束时 α_2的最优解$\alpha_2^{new,unc}$，然后再求经过剪辑后的解α_2^{new}。我们不加证明地给出结论，为了叙述简单，记

$$g(x) = \sum_{i=1}^{m} \alpha_i y_i \kappa(x_i, x) + b$$

令

$$E_i = g(x_i) - y_i = (\sum_{j=1}^{m} \alpha_j y_j \kappa(x_j, x_i) + b) - y_i, i = 1,2$$

当$i = 1,2$时，E_i为函数$g(x)$对输入x_i的预测值与真实输出y_i之差。那么

$$\alpha_2^{new,unc} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{\eta}$$

其中

$$\eta = \kappa_{11} + \kappa_{22} - 2\kappa_{12} = \| \Phi(x_1) - \Phi(x_2) \|^2$$

$$\alpha_2^{\text{new}} = \begin{cases} H, & \alpha_2^{\text{new,unc}} > H \\ \alpha_2^{\text{new,unc}}, & L \leqslant \alpha_2^{\text{new,unc}} \leqslant H \\ L, & \alpha_2^{\text{new,unc}} < L \end{cases}$$

由α_2^{new}求得α_1^{new}是

$$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new})$$

一些具体的算法实现原理时间受限，此处不予给出。

实验设置

超参数

惩罚系数 C

在支持向量机（Support Vector Machine，SVM）算法中，参数 C 是一个正则化参数，用于控制分类器的错误分类和决策边界之间的权衡。它影响了 SVM 在训练过程中的优化目标。

C 的值越大，SVM 将尽量避免错误分类，可能导致更复杂的决策边界。这意味着模型对训练数据的拟合程度较高，但也容易过拟合。过拟合指的是模型过于关注训练数据的细节和噪声，而忽略了更广泛的数据模式，从而导致在新的未见过的数据上表现不佳。

相反，较小的 C 值允许一些错误分类出现，可能导致更简单的决策边界。这种情况下，模型对训练数据的拟合程度较低，但有更好的泛化能力，能够更好地适应未见过的数据。

选取适当的 C 值通常需要通过交叉验证或网格搜索等方法来确定。常见的方法是尝试一系列的 C 值，然后使用交叉验证来评估每个 C 值对应的模型性能。在交叉验证过程中，将数据集划分为训练集和验证集，并重复多次，每次使用不同的划分。然后根据验证集上的

性能指标（例如准确率或 F1 分数）选择最优的 C 值。

选择 C 值时需要权衡模型的复杂性和泛化能力。如果模型过于复杂（C 值过大），可能会导致过拟合；如果模型过于简单（C 值过小），可能会导致欠拟合。因此，通过调整 C 值，可以在模型的拟合程度和泛化能力之间找到一个平衡点，以获得较好的性能。

核函数

在支持向量机（Support Vector Machine，SVM）算法中，核函数用于将原始特征空间映射到一个高维特征空间，以便更好地分离不同类别的样本。选择合适的核函数对于 SVM 的性能至关重要，不同的核函数适用于不同类型的数据。

以下是一些常见的核函数及其特点：

1. 线性核函数（Linear Kernel）：

线性核函数在原始特征空间中进行线性计算，即不进行任何映射。它适用于特征空间本身是线性可分的情况，例如数据线性可分或特征已经通过其他方式进行了显式的非线性映射。

2. 多项式核函数（Polynomial Kernel）：

多项式核函数通过多项式映射将数据映射到高维特征空间。它引入了多项式项，并可以捕捉到一定程度的非线性关系。多项式核函数的一个重要参数是多项式的次数，可以通过交叉验证来选择合适的次数。

3. 高斯核函数（Gaussian Kernel，也称为径向基函数，Radial Basis Function，RBF）：

高斯核函数通过将数据映射到无穷维特征空间来捕捉非线性关系。它在高维空间中创建了以支持向量为中心的局部决策区域，可以更好地处理复杂的非线性问题。高斯核函数的一个重要参数是带宽（bandwidth）或高斯核函数的标准差，可以通过交叉验证来选择合适的值。

4. sigmoid 核函数：

sigmoid 核函数通过 sigmoid 函数将数据映射到高维特征空间。它可以用于处理非线性问题，但在实践中往往效果不如高斯核函数和多项式核函数好，因此在一般情况下不常使用。

选择核函数需要考虑以下几个因素：

- 数据的性质：核函数的选择应基于对数据的先验知识和理解。如果相信数据具有某种特定的非线性结构，则选择相应的核函数可能会更合适。
- 计算效率：不同的核函数计算复杂度不同。高斯核函数的计算复杂度最高，而线性核函数的计算复杂度最低。因此，在处理大规模数据集时，计算效率是一个重要的考虑因素。
- 参数调节：一些核函数具有参数，如多项式核函数和高斯核函数。适当调节参数可以改善模型的性能，所以交叉验证或网格搜索等方法可以用于选择最佳的参数值。

评估方法

单一评估指标

对于二分类回归任务，支持向量机（Support Vector Machine，SVM）可以使用一些评估方法来评估其性能。以下是一些常用的评估方法：

1. 准确率（Accuracy）：

准确率是最常见的评估指标之一，表示分类器正确预测的样本数与总样本数之比。它可以通过计算以下公式得到：

准确率 =(真阳性 + 真阴性) / (真阳性 + 真阴性 + 假阳性 + 假阴性)

2. 精确率（Precision）和召回率（Recall）：

精确率和召回率是用于评估不平衡数据集上分类器性能的重要指标。

- 精确率表示分类器预测为正例的样本中，实际为正例的比例。精确率可以通过计算以下公式得到：

精确率 = 真阳性 / (真阳性 + 假阳性)

- 召回率表示实际为正例的样本中，分类器正确预测为正例的比例。召回率可以通过计算以下公式得到：

召回率 = 真阳性 / (真阳性 + 假阴性)

3. F1 分数（F1-Score）：

F1 分数是精确率和召回率的调和平均值，可以综合考虑分类器的精确性和召回率。F1 分数可以通过计算以下公式得到：

F1 分数 = 2 * (精确率 * 召回率) / (精确率 + 召回率)

4. ROC 曲线和 AUC：

ROC 曲线（Receiver Operating Characteristic Curve）和 AUC（Area Under the Curve）用于评估分类器在不同阈值下的性能。ROC 曲线以假阳性率（False Positive Rate，FPR）为横坐标，真阳性率（True Positive Rate，TPR）为纵坐标绘制，AUC 表示 ROC 曲线下的面积。AUC 值越接近 1，表示分类器性能越好。

在评估 SVM 模型性能时，可以根据任务的要求选择合适的评估指标。准确率适用于数据集类别平衡的情况，而精确率、召回率和 F1 分数适用于类别不平衡的情况。ROC 曲线和 AUC 可以提供分类器在不同阈值下的综合性能评估。

交叉验证

除了单一评估指标，还可以结合交叉验证来进行模型评估，以获得更准确的性能估计。

交叉验证是一种常用的模型评估方法，特别适用于数据集有限的情况。它可以提供更可靠的模型性能估计，帮助我们评估模型在未见过的数据上的泛化能力。

以下是几种常见的交叉验证方法：

1. k 折交叉验证（k-fold Cross-Validation）：

k 折交叉验证将数据集分成 k 个近似大小的子集，其中 k-1 个子集用作训练数据，剩下的一个子集作为验证数据。然后，重复 k 次，每次选择不同的验证子集。最后，将 k 次验证结果的平均值作为最终的模型性能评估指标。

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

留一交叉验证是 k 折交叉验证的特殊情况，其中 k 被设置为数据集的样本数量。对于每个样本，它都会成为验证集，而其他样本组成训练集。这种方法的优点是能够使用尽可能多的样本进行训练，但计算成本较高。

3. 分层 k 折交叉验证（Stratified k-fold Cross-Validation）：

分层 k 折交叉验证是在不平衡数据集上应用的一种改进的交叉验证方法。它确保每个子集中的类别分布与整个数据集中的类别分布相似，从而更准确地评估模型在各个类别上的性能。

在任务二中，我使用 sklearn 的分层 k 折交叉验证来进行参数参数探索。

实验结果

任务一

(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？

使用硬间隔 SVM，并以 QP 方法求解，最终在测试集上得到了 61.95%的准确率，并不理想。以下是一些原因分析：

- 硬间隔对异常值敏感：硬间隔 SVM 在求解过程中试图找到一个完全分隔训练样本的超平面。这使得它对异常值（离群点）非常敏感。如果存在离群点，它们可能会对超平面的位置产生较大影响，导致分类器过于关注这些异常值，而忽略了其他样本。
- 不适用于线性不可分问题：硬间隔 SVM 要求训练样本是线性可分的，即存在一个超平面能够完全正确地将正例和反例分开。然而，当数据集线性不可分时，硬间隔 SVM 无法找到满足要求的超平面。在这种情况下，使用硬间隔 SVM 将导致没有可行的解，或者导致过度拟合训练数据。
- 过拟合风险：由于硬间隔 SVM 追求完全分离训练样本的超平面，它可能会过度拟合训练数据。这意味着它对训练数据的噪声或不完全表示过于敏感，可能导致在未见过的数据上表现不佳。但是在本数据集上并没有这个可能，因为经过测试，在训练集上的准确率也仅仅是 62.78%。

以下是关于 QP 解法的缺点的说明：

- 维度灾难：当处理高维数据集时，计算二次规划问题的复杂度会随着特征的数量增加而迅速增加。由于二次规划问题的解是由训练样本的内积计算得出的，高维数据集会导致计算量大大增加，使问题变得难以求解。
- 存储需求：对于大规模数据集，需要存储每对训练样本之间的内积，这将占用大量内存。随着数据集的增长，存储需求会显著增加，可能超出计算机的内存限制。
- 计算复杂度：传统的 QP 方法在解决大规模问题时可能面临计算时间过长的问题。二次规划问题的求解时间与样本数量呈二次关系，因此对于大型数据集，求解时间可能会变得非常长。

(2) [15pts] 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 算法求解 SVM 对偶问题。注意第 3 步，KKT 条件验证步骤不能缺少。

使用线性核，并限定 C 为 2，手动实现 SMO 算法，最终得到 95.58%的准确率。接下来再验证 KKT 条件

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{w}} = 0, \dfrac{\partial L}{\partial b} = 0, \dfrac{\partial L}{\partial \alpha} = 0 \\ \alpha_i(1 - y_i(\mathbf{w}^T\mathbf{x} + b)) = 0, i = 1,2,\cdots,N \\ 1 - y_i(\mathbf{w}^T\mathbf{x} + b) \leqslant 0, i = 1,2,\cdots,N \\ \alpha_i \geqslant 0, i = 1,2,\cdots,N \end{cases}$$

事实上，确实存在一些点违反了 KKT 条件，如下所示

```
KKT condition violated for sample 90
    alpha: [1.]
    prediction: [0.68989917]
    true label: 1
KKT condition violated for sample 112
    alpha: [1.]
    prediction: [0.95038797]
    true label: 1
KKT condition violated for sample 157
    alpha: [1.]
    prediction: [-0.04016917]
    true label: 1
KKT condition violated for sample 204
    alpha: [0.29393658]
    prediction: [0.97958219]
    true label: 1
KKT condition violated for sample 208
    alpha: [1.]
    prediction: [0.6314123]
    true label: 1
KKT condition violated for sample 225
    alpha: [1.]
    prediction: [0.80833638]
    true label: 1
KKT condition violated for sample 228
    alpha: [1.]
    prediction: [0.92153408]
    true label: 1
```

但是这是合理的，因为如果训练数据不是线性可分的，即存在一些样本无法被完全正确地分类，那么在软间隔 SVM 中可能会出现一些违反 KKT 条件的 alpha。这是因为在软间隔 SVM 中允许一定程度的误分类，即有一些样本的函数间隔小于 1，但仍然被允许存在。

(3) [10pts] 对测试数据进行预测，确保预测结果尽可能准确。

使用线性核，并限定 C 为 2，最终得到 95.58%的准确率。

任务二

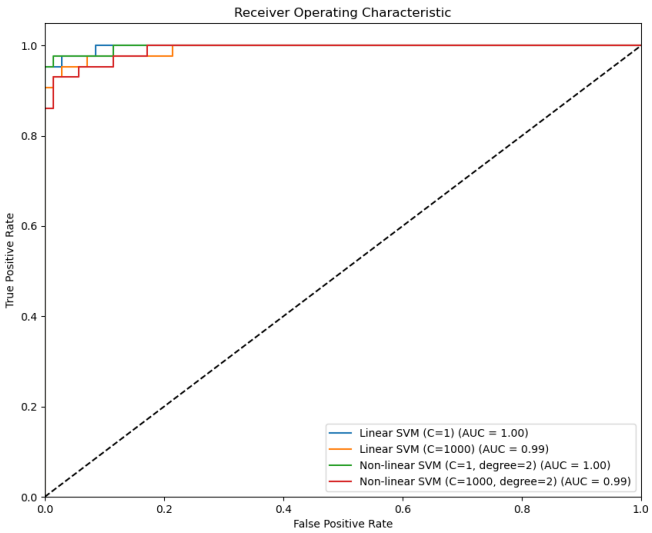
(1) [20pts]使用 sklearn 库简洁实现软间隔 SVM。首先实现以下 4 个示例性的 SVM 模型：

- **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。

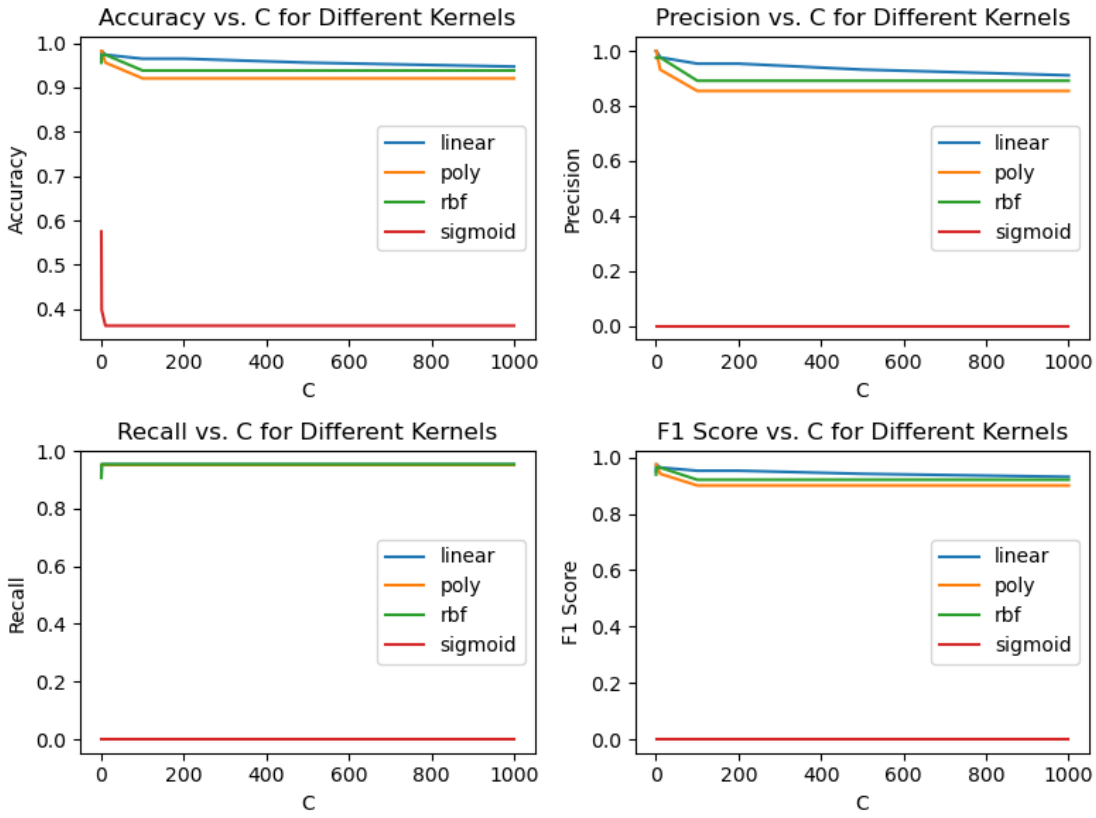
分别实现四种 SVM 并从准确率、精确率、召回率和 f1 score 进行比较，并绘制 ROC 图。由下面两图可见，C=1 的 SVM 好于 C=1000 的 SVM；在 C 较大时，使用线性核的效果好于使用多项式核的效果。总体来说，选择 C=1，核函数为线性核或多项式核是最优的。

```
Linear SVM (C=1)
Accuracy: 0.9823008849557522
F1 Score: 0.9761904761904763
Recall: 0.9534883720930233
Precision: 1.0

Linear SVM (C=1000)
Accuracy: 0.9469026548672567
F1 Score: 0.9318181818181819
Recall: 0.9534883720930233
Precision: 0.9111111111111111

Non-linear SVM (C=1, degree=2)
Accuracy: 0.9823008849557522
F1 Score: 0.9761904761904763
Recall: 0.9534883720930233
Precision: 1.0

Non-linear SVM (C=1000, degree=2)
Accuracy: 0.9380530973451328
F1 Score: 0.9213483146067417
Recall: 0.9534883720930233
Precision: 0.8913043478260869
```

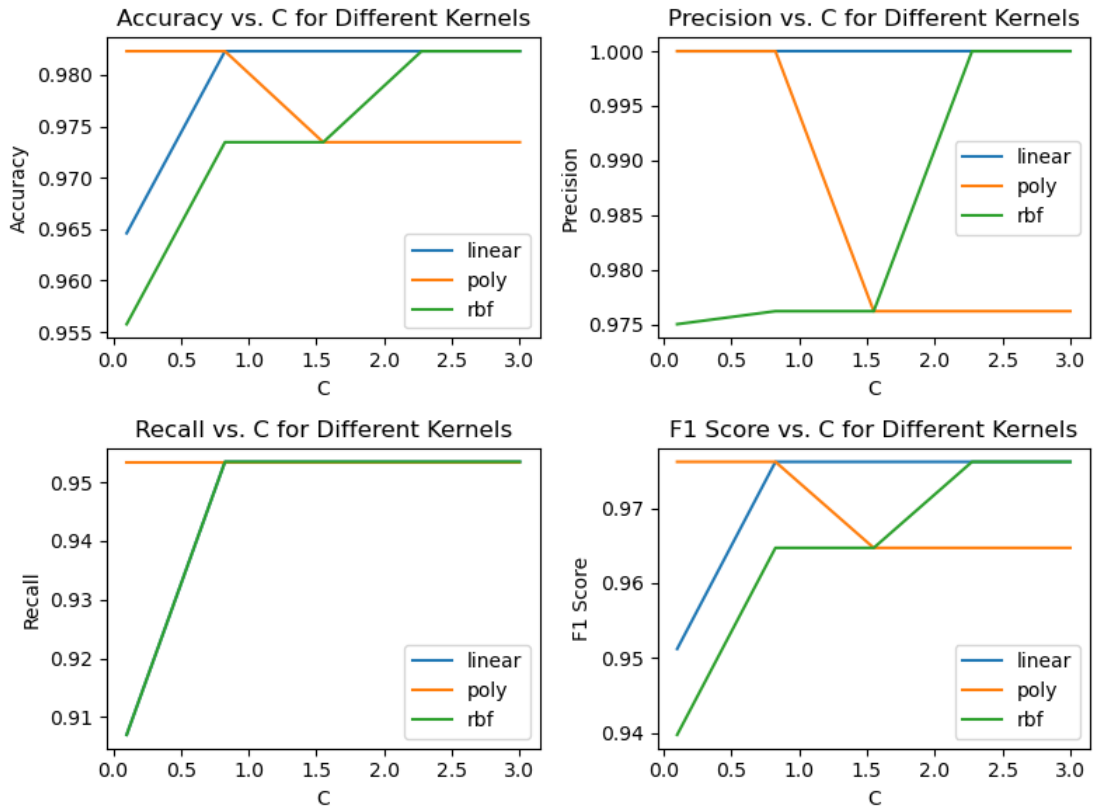


(2) [20pts]参数选择与参数分析，确定正则化常数 C 与核函数及其参数的选择范围，选用合适的实验评估方法（回顾第 2 章的内容）进行参数选择，并进行参数分析实验。

先在 C={0.1, 1, 10, 100, 200, 500, 1000}中分别使用线性核、多项式核、高斯核和 sigmoid 验证。如下所示，sigmoid 的效果非常差，而其余的几种核都有随 C 增大而性能指标减小的趋势。因此，接下来去掉 sigmoid 核，在剩下三种核当中进一步缩小参数确定的范围。



在 C={0.1,0.42 0.74, 1.06 1.38, 1.71, 2.03 2.35 2.67 3.0}中分别使用线性核、多项式核和高斯核。可以看到线性核在 C=1 附近的表现是最佳的。



这个实验使用的数据集是乳腺癌数据集。事实上，对于乳腺癌，假阴性的后果要比假阳

性的后果大得多，所以在选择衡量模型的指标的时候，我们应该多考虑召回率。