作业 1：线性模型和支持向量机

清华大学软件学院
机器学习, 2024 年秋季学期

1 介绍

本次作业需要提交说明文档（PDF 形式）和 Python 的源代码。注意事项如下：

- 本次作业有部分附加题，若总得分超过 100 分，则按照 100 分截断。
- 作业按点给分，因此请在说明文档中按点回答，方便助教批改。示例如下：
  2.2.1 $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x_i) - y_i\right)^2 + \cdots$。
- 除非特别说明，本次作业不能直接使用机器学习的开源库，例如 sklearn、pytorch 等。
- 你需要熟练掌握 numpy 及其广播（broadcast）机制，使用 for 循环实现的矩阵运算不会得到编程部分的分数。
- 请通过提供易懂的注释或变量命名确保代码有足够的可读性，若代码的可读性过差，会酌情扣除对应题的分数。
- 在 PDF 中记录你在写程序的哪些模块时与他人有过交流，与他人交流需具体到姓名（网络论坛上的交流可填写用户名），参考网络资料需具体到链接。
- 如使用大模型辅助作业完成，请在 PDF 中声明作业中你使用了大模型的部分和使用的方式。
- 不要使用他人的作业（含代码和文档），也不要向他人公开自己的作业，否则处罚很严厉，会扣至-100（倒扣本次作业的全部分值）。

2 线性模型与梯度下降 (50pt+2pt)

在本题中，你将使用梯度下降法（Gradient Descent）实现岭回归（Ridge Regression）算法。

2.1 特征归一化 (4pt)

在实际应用中，当数据的不同维特征差异很大时，梯度下降的收敛速度会变得很慢。此外，使用正则化时，具有较大绝对值的特征对正则化有更大的影响。因此，我们需要进行特征归一化。一

种常见方法是执行仿射变换，将**训练集**中的所有特征值映射至 $[0,1]$，对测试集上的每个特征也需要使用相同的仿射变换。

1. 补全函数 `split_data`，将数据集划分为训练集和测试集。
2. 补全函数 `feature_normalization`，实现特征的归一化。

2.2 目标函数与梯度 (10pt)

在**线性回归**中，我们考虑线性函数的假设空间 $h_\theta : \mathbf{R}^d \to \mathbf{R}$,

$$h_{\theta,b}(x) = \theta^T x + b,$$

其中 $\theta, x \in \mathbf{R}^d$, 为了书写和编程的方便，我们通常为 x 添加一个额外的维度，该维度始终是一个固定值，例如 1，用于消除 h_θ 的偏移量参数 b。此时，我们可以将待优化的函数写成:

$$h_\theta(x) = \theta^T x,$$

其中 $\theta, x \in \mathbf{R}^{d+1}$。我们需要找到合适的 θ 最小化:

$$J_{\mathrm{MSE}}(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x_i) - y_i\right)^2,$$

其中 $(x_1, y_1), \ldots, (x_m, y_m) \in \mathbf{R}^{d+1} \times \mathbf{R}$ 是训练数据。

岭回归是使用 L_2 正则化的线性回归，其目标函数是

$$J(\theta) = J_{\mathrm{MSE}}(\theta) + \lambda R(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x_i) - y_i\right)^2 + \lambda\theta^T\theta,$$

其中 λ 是正则化参数，它控制正则化程度。

1. 将训练数据的特征记作 $X = (x_1, x_2, \ldots, x_m)^T \in \mathbf{R}^{m\times(d+1)}$ ，即 X 的第 i 行是 x_i^T, 训练数据的输出记作 $y = (y_1, y_2, \ldots, y_m)^T \in \mathbf{R}^m$，请写出 $J(\theta)$ 的矩阵形式。
2. 补全函数 `compute_regularized_square_loss`，给定 θ，计算的 $J(\theta)$。
3. 请写出 $J(\theta)$ 对 θ 梯度的矩阵形式。
4. 补全函数 `compute_regularized_square_loss_gradient`，给定 θ，计算 $J(\theta)$ 的梯度。

   为了检查梯度的实现是否正确，可以用数值法来计算梯度。可导函数的方向导数可以通过下述公式计算，

   $$\lim_{\varepsilon\to 0}\frac{J(\theta+\varepsilon h) - J(\theta-\varepsilon h)}{2\epsilon}$$

   在实际操作中，我们可以选择一个较小的 $\varepsilon > 0$，通过逼近方向导数来逼近梯度。更具体地，可以在每个坐标方向上首先取 $h = (1, 0, 0, \ldots, 0)^T$ 计算第一维的梯度，然后取 $h = (0, 1, 0, \ldots, 0)^T$ 得到第二维的梯度，以此类推。将每一维的梯度合起来，就得到 $J(\theta)$ 在 θ 处的梯度。我们已在代码中实现了函数 `grad_checker`，你可以用它检验梯度计算函数的正确性。

2.3 梯度下降 (12pt)

1. 在最小化 $J(\theta)$ 时，假设取 θ 到 $\theta+\eta h$ 的一步，其中 $h \in \mathbf{R}^{d+1}$ 是前进方向（不一定是单位向量），$\eta \in (0,\infty)$ 是步长。请用梯度写出目标函数值变化的近似表达式 $J(\theta+\eta h)-J(\theta)$，思考 h 为哪一前进方向时目标函数下降最快，并据此写出梯度下降中更新 θ 的表达式。
2. `main` 函数已经加载了数据，将其拆分成了训练集和测试集，并完成了归一化。你现在需要补全函数 `gradient_descent` 实现**梯度下降**算法，使得模型可以在训练集上进行训练。
3. 选择合适的步长：固定 $\lambda = 0$，从 0.1 的步长开始，尝试各种不同的固定步长（至少包括 0.5、0.1、0.05 和 0.01），记录哪个步长收敛最快，哪个步长会导致发散，并绘制目标函数在不同步长下随着训练时间变化的曲线。

2.4 模型选择 (8pt)

1. 我们可以通过**验证集**上的均方误差 $J_{\mathrm{MSE}}(\theta)$ 来选择合适的模型训练超参数，但我们目前还没有验证集。补全函数 `K_fold_split_data`，将训练集分为 K 组（不妨令 K=5）交叉验证的训练集和验证集。
2. 补全函数 `K_fold_cross_validation`，实现 K 折交叉验证。用表格记录不同超参数下模型的验证集均方误差，搜索出最优的模型训练超参数，搜索范围包括步长（至少包括 $0.05, 0.04, 0.03, 0.02, 0.01$）、正则化系数（至少包括 $10^{-7}, 10^{-5}, 10^{-3}, 10^{-1}, 1, 10, 100$），汇报选出的最优超参数与最优模型的测试集均方误差。

2.5 随机梯度下降 (10pt+2pt)

当训练数据集非常大时，梯度下降算法需要遍历整个数据集，因此效率较低。实际中用的往往是**随机梯度下降**（SGD）算法。令 $f_i(\theta) = (h_\theta(x_i) - y_i)^2$ 为第 i 个数据点上的平方误差，则

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} f_i(\theta) + \lambda\theta^T\theta$$

在 SGD 中，每一步会随机采样一个小批量（batch）$\{(x_{i_k}, y_{i_k})\}_{k=1}^{n}$ 来计算目标函数

$$J_{\mathrm{SGD}}(\theta) = \frac{1}{n}\sum_{k=1}^{n} f_{i_k}(\theta) + \lambda\theta^T\theta$$

和梯度方向，其中 $n \ll m$ 并且 i_k 从 $\{1, 2, \ldots, m\}$ 中独立同分布采样。

1. 请写出 $J_{\mathrm{SGD}}(\theta)$ 对应的梯度 $\nabla J_{\mathrm{SGD}}(\theta)$ 的表达式。
2. **（附加题）**证明随机梯度 $\nabla J_{\mathrm{SGD}}(\theta)$ 是 $\nabla J(\theta)$ 的无偏估计，即证明

$$\mathbb{E}_{i_1, i_2, \ldots, i_n}\left[\nabla J_{\mathrm{SGD}}(\theta)\right] = \nabla J(\theta)$$

（提示：可以利用期望的线性性质）

3. 补全函数 `stochastic_grad_descent`，实现**随机梯度下降**算法。
4. 随机梯度下降算法的噪音较大，因此模型较难收敛。你需要选择合适的批大小：固定 $\lambda = 0$，并根据 2.3.3 或 2.4.2 的结果固定选取一个合适的步长，从批大小 1 开始，尝试各种不同的

批大小，并记录随着批大小逐渐增大时，训练曲线发生的变化。注意：在随机梯度下降中，小批量的训练损失函数噪声较大，难以清晰反映模型收敛情况，所以要通过验证集上的全批量损失来判断模型收敛情况。开始训练前，需先将训练集重新划分为训练集和验证集（使用 `split_data` 即可，不需要使用 `K_fold_split_data`）。

2.6 解析解 (6pt)

1. 对于岭回归模型，我们可以计算出解析解。请写出岭回归模型解析解的表达式，并实现函数 `analytical_solution`。
2. 正则化往往可以有效避免过拟合。请用表格记录不同正则化系数 λ 的岭回归模型解析解在测试集上的均方误差，展现出正则化对于避免过拟合的有效性。
3. 从计算时间、测试集均方误差等角度比较机器学习方法与解析解。机器学习方法相比解析解在该任务中有优势吗？分析原因。

3 Softmax 回归 (10pt+3pt)

线性模型也可用于处理（多）分类任务。假设类别数为 K，输入数据为 $\mathbf{x} \in \mathbb{R}^n$，权重参数为 $\mathbf{W} \in \mathbb{R}^{K\times n}$，偏置项为 $\mathbf{b} \in \mathbb{R}^K$，模型的输出为：

$$\mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}$$

可使用 Softmax 函数将模型输出 $\mathbf{z}$ 转换为类别的概率分布：

$$\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_K]^T, \quad \hat{y}_i = \frac{\exp z_i}{\sum_{j=1}^{K} \exp z_j}$$

其中 $\hat{y}_i$ 表示输入 $\mathbf{x}$ 属于第 i 类的预测概率。

将数据的真实类别表示为独热（one-hot）向量 $\mathbf{y} = [y_1, y_2, \cdots, y_K]^T$ 的形式，其中 $y_k = 1, \forall i \neq k, y_i = 0$（k 是数据的真实类别）。可以得出 Softmax 函数的负对数似然损失函数（交叉熵损失函数）：

$$\mathcal{L} = -\log \hat{y_k} = -\mathbf{y}^T \log \hat{\mathbf{y}}$$

1. 将 $\mathcal{L}$ 视为 $\mathbf{z}$ 的函数，请写出 $\mathcal{L}$ 关于 $\mathbf{z}$ 的梯度。
2. 将 $\mathcal{L}$ 视为 $\mathbf{W}$ 和 $\mathbf{b}$ 的函数，请分别写出 $\mathcal{L}$ 关于 $\mathbf{W}$ 和 $\mathbf{b}$ 的梯度。
3. 假设我们有一个标量函数 $f(\mathbf{x})$，其中 $\mathbf{x} \in \mathbb{R}^n$ 是一个 n 维向量，函数 $f(\mathbf{x})$ 对 $\mathbf{x}$ 中的每个分量有连续的二阶偏导数，那么海森矩阵 $\mathbf{H}$ 是由该函数的所有二阶偏导数组成的矩阵，其形式如下：

   $$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

   请写出 $\mathcal{L}$ 关于 $\mathbf{z}$ 的海森矩阵 $\mathbf{H}$。
4. **(附加题)** 请证明该海森矩阵 $\mathbf{H}$ 是半正定的。

4 支持向量机 (40pt+10pt)

在本题，我们将使用支持向量机（Support Vector Machine）对文本数据进行情绪检测。

4.1 次梯度 (3pt)

对于 $f : \mathbf{R}^d \to \mathbf{R}$，在 x 如果对于所有 z，

$$f(z) \geq f(x) + g^T(z - x),$$

则 $g \in \mathbf{R}^d$ 为 x 处的**次梯度**（subgradients）。次梯度并不唯一，f 在 x 处的次梯度集合记为 $\partial f(x)$，那么 $g \in \partial f(x)$。大部分时候，例如在进行次梯度下降的时候，我们只需要次梯度集合中的一个次梯度即可，这时也可以写作 $\partial f(x) = g$。

基于以上的定义，你需要给出合页损失函数（Hinge Loss）的次梯度。合页损失如下。

$$J(w) = \max\left\{0, 1 - yw^T x\right\}$$

4.2 硬间隔支持向量机 (6pt)

给定线性可分的数据集 $(x_1, y_1), \ldots, (x_m, y_m) \in \mathbf{R}^d \times \{-1, 1\}$，支持向量机模型希望能找到一个超平面将两个类别完全区分开，即找到 $w \in \mathbf{R}^d, b \in \mathbf{R}$ 使得

$$y_i(w^T x_i + b) > 0, \quad 1 \leq i \leq m$$

即所有标签 $y = 1$ 的 x 都在超平面 $\left\{x \mid w^T x + b = 0\right\}$ 的一侧，并且所有标签 $y = -1$ 的 x 都在另一侧。同时，支持向量机希望数据关于该超平面的间隔尽量大。不失一般性，我们可以设距离该超平面最近的点在 $w \cdot x + b = \pm 1$ 上，此时问题可以转化为带约束优化问题：

$$\begin{aligned} \min_{w \in \mathbf{R}^d, b \in \mathbf{R}} \quad & \frac{1}{2}\|w\|_2^2 \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1, \quad 1 \leq i \leq m \end{aligned} \tag{1}$$

设拉格朗日变量 $\mu_i \geq 0, \quad 0 \leq i \leq m$ 对应上述问题的 m 个约束，则问题 (1) 对应的拉格朗日函数为：

$$L(w, b, \mu) = \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{m} \mu_i [y_i(w^T x_i + b) - 1] \tag{2}$$

1. 请根据上述内容，写出问题 (1) 的 KKT 条件。
2. 证明满足 KKT 条件的解 w 一定是训练数据 $x_1, x_2, ..., x_m$ 的线性组合。设解 $w = \sum_{i=0}^{m} \alpha_i x_i$，若 $\alpha_i \neq 0$，对应的 x_i 称为支持向量。请说明支持向量均位于分隔平面 $w \cdot x + b = \pm 1$ 上。

4.3 软间隔支持向量机 (10pt+1pt)

线性可分是一种理想的情况，现实数据集经常会有噪声，无法约束所有的数据点都能被正确分类，因此实际中的支持向量机都会引入软间隔，目标是让不满足约束的数据点尽可能少，即

$$\begin{aligned} \min_{w\in\mathbf{R}^d, b\in\mathbf{R}, \xi\in\mathbf{R}^m} \ & \frac{\lambda}{2}\|w\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i^p. \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \quad 1 \leq i \leq m \end{aligned} \tag{3}$$

其中，系数 $p \geq 1$ 用于控制对数据点不满足约束的惩罚力度。

1. 求 $p \geq 1$ 时，该问题的拉格朗日（Lagrange）方程。
2. 求 $p = 1$ 时，该问题的对偶形式（Dual Form）。
3. 在求出对偶问题后，使用 Sequential Minimal Optimization (SMO) 算法，可以实现软间隔支持向量机的实际求解。$p = 1$ 时，上文中的优化问题也可以等价表示为：

$$\min_{w\in\mathbf{R}^d, b\in\mathbf{R}} \frac{\lambda}{2}\|w\|^2 + \frac{1}{m}\sum_{i=1}^{m}\max\left\{0, 1 - y_i(w^T x_i + b)\right\}. \tag{4}$$

   此时，SVM 求解还有另外一种基于**次梯度下降**的算法。证明此时软间隔支持向量机中 $J_i(w) = \frac{\lambda}{2}\|w\|^2 + \max\left\{0, 1 - y_i(w^T x_i + b)\right\}$ 的次梯度由下式[1]给出（提示：参考合页损失函数的次梯度）。

$$\partial J_i|_w = \begin{cases} \lambda w - y_i x_i & \text{for } y_i(w^T x_i + b) < 1 \\ \lambda w & \text{for } y_i(w^T x_i + b) \geq 1. \end{cases}, \quad \partial J_i|_b = \begin{cases} -y_i & \text{for } y_i(w^T x_i + b) < 1 \\ 0 & \text{for } y_i(w^T x_i + b) \geq 1. \end{cases}$$

4. **(附加题)** $p > 1$ 时，请求出 $J_i(w) = \frac{\lambda}{2}\|w\|^2 + \max\left(\left\{0, 1 - y_i(w^T x_i + b)\right\}\right)^2$ 的梯度表达式。

4.4 核方法 (6pt+4pt)

非线性分类问题是指通过利用非线性模型才能很好地进行分类的问题，面对这类问题，我们可以使用**核技巧**（kernel trick）。核技巧将数据 $\boldsymbol{x} = (x_1, x_2, ..., x_d)^T \in \mathcal{X}$ 通过基函数 $\boldsymbol{\Phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), ..., \phi_n(\boldsymbol{x}))^T$ 映射到高维空间，并定义基函数导出的核函数为 $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{\Phi}(\boldsymbol{x}_1) \cdot \boldsymbol{\Phi}(\boldsymbol{x}_2)$。通过对问题的变形，可以用核函数替代高维空间上的点积操作，从而简化问题的计算。

要验证任意的函数 $k(\boldsymbol{x}_1, \boldsymbol{x}_2)$ 是否是核函数，除了直接找出其对应的基函数 $\boldsymbol{\Phi}(\boldsymbol{x})$，还可以通过核矩阵是否对称半正定进行判断。Mercer 条件指出，对于紧集 $\mathcal{X} \subset \mathbf{R}^d$ 和对称函数 $k : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$，$k$ 是核函数当且仅当对于任意 $(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m) \subseteq \mathcal{X}$，矩阵 $\boldsymbol{K} = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j} \in \mathbf{R}^{m\times m}$ 对称半正定，即矩阵 $\boldsymbol{K}$ 对称且对于任意列向量 $\boldsymbol{a} = (a_1, a_2, ..., a_m)^T \in \mathbf{R}^{m\times 1}$，

$$\boldsymbol{a}^T \boldsymbol{K} \boldsymbol{a} = \sum_{i,j=1}^{m} a_i a_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$$

[1] $\partial J_i|_w$ 表示次梯度 ∂J_i 在 w 对应的维度上的分量，$\partial J_i|_b$ 同理。

1. 对于定义在 $\mathbf{R}^n \times \mathbf{R}^n$ 上的对称函数 $k(\boldsymbol{x}, \boldsymbol{x}') = cos\angle(\boldsymbol{x}, \boldsymbol{x}')$，证明矩阵 $\boldsymbol{K} = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j} \in \mathbf{R}^{m \times m}$ 对称半正定，并求出其对应的基函数 $\boldsymbol{\Phi}(\boldsymbol{x})$ 使 $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{\Phi}(\boldsymbol{x}_1) \cdot \boldsymbol{\Phi}(\boldsymbol{x}_2)$。（$\angle(\boldsymbol{x}, \boldsymbol{y})$ 表示向量 $\boldsymbol{x}$, $\boldsymbol{y}$ 的夹角）
2. **(附加题)** 证明对任意两个核函数 $k_1, k_2 : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$，它们的和 $k_s(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') + k_2(\boldsymbol{x}, \boldsymbol{x}')$ 仍是核函数，乘积 $k_p(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') \cdot k_2(\boldsymbol{x}, \boldsymbol{x}')$ 也是核函数。（提示：对任意对称半正定矩阵 $\boldsymbol{K}$，存在矩阵 $\boldsymbol{M}$ 使得 $\boldsymbol{K} = \boldsymbol{M}\boldsymbol{M}^T$）
3. 已知非线性映射 $\boldsymbol{\Phi}$ 和其对应的核函数 $k(\cdot, \cdot)$，若在 4.3 节中的支持向量机 (取 p=1) 中使用该核函数，请写出对偶问题形式，并指出如何用使用核函数计算测试样本的分类。

4.5 情绪检测 (15pt+5pt)

我们将使用**线性软间隔支持向量机**进行情绪检测，类别包括：开心（joy）和伤心（sadness）。我们在 `start_code.py` 中已经实现了数据集的加载与预处理，你也可以自己实现。你需要完成

1. 在函数 `linear_svm_subgrad_descent` 中实现 SVM 的随机次梯度下降算法，并在情绪检测数据集上进行训练（提示：可参考 2.5 节随机梯度下降的代码）。
2. 调整超参数，例如批大小、正则化参数 λ、步长、步长衰减策略等，观察训练完成时训练集上准确率与验证集上准确率随着超参数发生的变化，绘制曲线或者表格来记录你的超参数调优过程和结果。（提示：仅需要如实地记录自己的超参数调优过程即可，**不要求完备地或有计划地搜索超参数空间**）
3. 在函数 `kernel_svm_subgrad_descent` 中实现基于核函数的非线性 SVM（例如基于线性核或者高斯核），调整相关的超参数，记录它在测试集上的准确率。核函数的引入能提高当前模型的准确吗？试解释原因。(6pt)
4. 计算并汇报最终的 SVM 模型在验证集上的准确率，F1-Score 以及混淆矩阵（Confusion Matrix）。
5. **(附加题)** 写出**逻辑斯特回归**（Logistic Regression）的目标函数和梯度的矩阵形式[2]，并实现逻辑斯特回归的随机梯度下降算法，汇报结果（提示：可以复用 SVM 中的大部分代码甚至超参数）。

[2]为了表达的简洁，在逻辑斯特回归中我们可以用 $y = 0$ 和 $y = 1$ 来表示两个类，而非 $y = -1$ 和 $y = +1$。