

Mini Matrix 矩阵运算库设计与实现技术报告

课程：计算导论 (CS1602/CS124)

组长：邹佳晨 (525030910080)

组员：李谨硕 (525030910098)

2026 年 1 月 9 日

摘要

本报告详细阐述了基于纯 Python 语法实现的简易矩阵运算库 `minimatrix` 的设计思路与技术细节。本项目旨在模仿 `numpy` 库的核心功能，实现了矩阵的基础运算、维度变换、行列式计算、矩阵求逆以及基于最小二乘法的线性回归应用。通过对 `minimatrix.py` 核心模块的封装与 `main.py` 的测试验证，证明了该库具备处理基础科学计算的能力。

1 项目背景与目标

本项目依照要求在不依赖 `numpy` 等第三方科学计算库的前提下，仅使用 Python 标准库（如 `random`, `fractions`）构建一个二维矩阵运算库。

主要目标包括：

1. 实现矩阵类的定义与基础属性（如形状、维度）。
2. 实现矩阵间的加、减、乘（Hadamard 积与点积）、转置等运算。
3. 实现高级线性代数功能：行列式、逆矩阵、秩。
4. 利用该库解决最小二乘法问题，验证其实用性。

2 系统设计思路

2.1 数据结构设计

为了表示二维矩阵，本库在 `Matrix` 类内部采用 `** 嵌套列表 (Nested List)**` 作为底层存储结构。

- `self.data`: 存储矩阵元素的二维列表，例如 `[[1, 2], [3, 4]]`。
- `self.dim`: 一个元组 `(rows, cols)`，用于快速访问矩阵形状，避免重复计算长度。

这种设计利用了 Python 列表的灵活性，能够方便地进行行索引和元素访问，同时符合题目对于“纯 Python 语法”的要求。

2.2 核心类架构

`Matrix` 类封装了所有对数据的操作。初始化方法 `__init__` 支持两种模式：

1. `** 数据驱动 **`: 传入现有的嵌套列表进行封装，并进行维度一致性检查。
2. `** 维度驱动 **`: 传入维度 `dim` 和初始值 `init_value`，生成指定大小的矩阵。

此外，为了提供类似 `numpy` 的便捷性，我们在类外部实现了一系列辅助生成函数，如 `arange`, `zeros`, `ones`, `nrandom` 等。

3 关键算法实现细节

3.1 矩阵乘法 (Dot Product)

我们在 `dot` 方法中实现了标准的三重循环算法：

$$C_{ij} = \sum_{k=1}^n A_{ik} \times B_{kj}$$

为了保证运算的合法性，在运算前首先检查左矩阵的列数是否等于右矩阵的行数。

3.2 高斯消元与数值精度

在实现行列式 (`det`)、逆矩阵 (`inverse`) 和秩 (`rank`) 时，核心算法均为 ** 高斯消元法 (Gaussian Elimination)**。

为了解决浮点数运算可能带来的精度丢失问题（特别是当矩阵作为分母或进行大量累加时），本项目引入了 Python 标准库 `fractions.Fraction`。

```
1 # 代码片段示意：使用 Fraction 保证高斯消元精度
2 mat = [[Fraction(x) for x in row] for row in self.data]
```

在计算逆矩阵时，构建增广矩阵 $[A|I]$ ，通过行变换将 A 变为单位矩阵 I ，此时右侧即为 A^{-1} 。使用分数类确保了在中间步骤不会出现截断误差，从而在计算 100×100 规模矩阵时仍能保持较高的理论精度。

3.3 运算符重载

为了使代码更加 Pythonic，我们重载了 Python 的内置运算符：

- `__add__` (+), `__sub__` (-): 对应矩阵加减。
- `__mul__` (*): 对应矩阵的元素级乘法 (Hadamard Product)。
- `__pow__` (**): 实现了矩阵的快速幂算法。
- `__getitem__` / `__setitem__`: 支持切片操作，如 `mat[0:2, 0:2]`。

4 测试与应用验证

4.1 基础功能测试

在 `main.py` 中，我们首先构建了一个 3×3 的矩阵进行基础测试 [?]. 测试内容涵盖：

- 形状变换：`reshape` 成功将 3×3 矩阵转换为 1×9 等形状 [?].
- 统计函数：`sum(axis=0)` 正确计算了列和。
- 生成器：`arange`, `zeros_like`, `nrandom_like` 均按预期输出了对应维度的矩阵。

4.2 最小二乘法求解线性回归

作为本项目的综合应用测试，我们模拟了 $m = 1000, n = 100$ 的线性回归问题 [?]:

$$Y = Xw + e$$

其中 X 为特征矩阵， w 为权重向量， e 为随机噪声。

我们利用 `minimatrix` 实现了正规方程的闭式解（Closed-form solution）:

$$\hat{w} = (X^T X)^{-1} X^T Y$$

代码实现如下：

```
1 # 最小二乘法核心实现
2 X_T = X.T()
3 X_T_X = X_T.dot(X)
4 X_T_X_inv = X_T_X.inverse() # 求逆
5 X_T_Y = X_T.dot(Y)
6 w_hat = X_T_X_inv.dot(X_T_Y) # 得到估计值
```

测试结果分析：程序输出了估计值 \hat{w} 与真实值 w 的相对误差。结果显示误差在可接受范围内，证明了矩阵乘法、转置以及求逆算法的正确性。