稀疏矩阵特征值

背景介绍：

在实际工程和科学应用中，我们经常会遇到大规模的数据集，其中数据往往以矩阵的形式表示。同时，这些矩阵往往是“稀疏”的，这意味着它们具有大量的零元素，而非零元素只占据了整个矩阵的一小部分。在不同的领域我们都会遇到稀疏矩阵，比如社交网络分析、图像处理、自然语言处理、有限元分析、物理学模拟等。这些领域中的数据通常具有高维度，但由于各种原因，矩阵中的数据元素只有极小一部分是非零的。因此，稀疏矩阵的特性和处理方法变得至关重要。

在这样的情况下，如何高效地计算和理解稀疏矩阵的特征值成为一个具有挑战性的问题。在本次期末大作业中，我们将探讨使用特定的数值计算方法来解决该问题，包括 Power Method、QR 算法和 Arnoldi 迭代算法。我们将比较不同算法在普通矩阵和高维稀疏矩阵上的性能，这将有助于深入理解稀疏矩阵的重要性以及特征值计算方法的应用。

稀疏矩阵的密度

矩阵 $A \in \mathbb{R}^{m\times n}$，其密度定义为非零元素数量与总元素数量的比值：

$$density = \frac{\text{num}(a_{ij} \neq 0)}{mn},$$

其中，a_{ij} 为矩阵 A 第 i 行第 j 列的元素。

QR 算法

QR 算法的核心思路是利用矩阵的 QR 分解，迭代构造一系列矩阵并最终收敛为上三角矩阵，从而计算特征值。具体而言，矩阵 $A \in \mathbb{R}^{n\times n}$ 可以通过 QR 分解得到 $A = QR$，其中 Q 为正交矩阵，R 为上三角阵。通过构建矩阵序列 $A_{k+1} = R_k Q_k = Q_k^\top A_k Q_k$，可以得到矩阵序列 $\{A_k\}$，其中每一个矩阵 A_k 都与矩阵 A 相似，即具有相同的特征值。假设特征值各不相同且降序排列，可以证明矩阵 A_k 中下三角的元素满足 $|a_{ij}^{(k)}| = \mathcal{O}(|\lambda_i / \lambda_j|^k), i > j$。因此，当 k 趋于无穷时，$A_k$ 收敛为上三角矩阵，此时的对角线元素即为特征值。

有三种方法可以用于实现 QR 分解，分别为：Gram–Schmidt 正交化过程，Householder 变换（参考课本[1] Chapter 9.4）和 Givens 变换。其中课本介绍的 Householder 变换可以将任意方阵转换为上 Hessenberg 矩阵 H，其中元素 $h_{ij} = 0, i > j+1$。

Arnoldi 迭代算法

Arnoldi 迭代算法[2]的核心思路是将矩阵投影到低维 Krylov 子空间，从而将矩阵约化为上 Hessenberg 矩阵，从而计算其特征值。通过低维 Krylov 子空间，Arnoldi 迭代并不直接利用矩阵本身，而是利用矩阵向量乘积，从而减小处理高维矩阵时的复杂度。具体而言，对于给定的矩阵 $A \in \mathbb{R}^{n\times n}$ 和向量 $b \in \mathbb{R}^n$，Krylov 序列为向量集合 $\{b, Ab, A^2b, A^3b, \cdots\}$。M 维

Krylov 子空间定义为长度为 m 的 Krylov 序列张成的空间（可以把序列中的向量理解为基向量）：

$$\mathcal{K}_m(A,b) \triangleq \text{span}\{a, Ab, A^2b, \cdots, A^{m-1}b\}.$$

Arnoldi 迭代期望将矩阵分解为 $A = QHQ^\top$，其中 H 为上 Hessenberg 矩阵，Q 为正交矩阵。将矩阵 Q 表示为列向量形式 $Q = \left[q_1 \,|\, q_2 \,|\, \cdots \,|\, q_n\right]$，并矩阵分解形式转换为 $AQ = QH$，其第 m 列的结果可以表示为：

$$Aq_m = h_{1m}q_1 + h_{2m}q_2 + \cdots + h_{m+1,m}q_{m+1},$$

其中，$h_{i,m} = 0, i > m+1$。利用上式可以推导向量 q_{m+1} 的递归形式：

$$q_{m+1} = \frac{Aq_m - \sum_{i=1}^{m} h_{i,m}q_i}{h_{m+1,m}}.$$

不难发现，实际上这与 Gram–Schmidt 求正交基的方法非常类似，q_m 就是 Krylov 子空间的正交基。通过求解的正交矩阵 Q，我们可以计算 Hessenberg 矩阵 $H = Q^\top AQ$。由于 H 与 A 相似，可以利用已有算法计算 H 的特征值从而得到 A 的特征值。

值得注意的是，对于高维矩阵我们往往只关注其一部分的特征值，比如只考虑前 $k \ll n$ 列。此时，可以将矩阵分解简化为矩阵分解为 $AQ_k = Q_{k+1}\tilde{H}_k$，其中 $Q_k = \left[q_1 \,|\, q_2 \,|\, \cdots \,|\, q_k\right]$，

$\tilde{H}_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & & h_{1,k} \\ h_{21} & h_{22} & \cdots & & h_{2,k} \\ 0 & h_{32} & \ddots & h_{k-1,k-1} & \vdots \\ & 0 & \ddots & h_{k,k-1} & h_{k,k} \\ & & & 0 & h_{k+1,k} \end{bmatrix}$。我们给出 Arnoldi 迭代算法的伪代码如下：

Algorithm 1 Arnoldi Iteration.

choose b arbitrarily, then $q_1 = b/\|b\|_2$
for $m = 1, 2, \cdots, k$ **do**
    $v = Aq_m$
    for $j = 1, 2, \cdots, m$ **do**
        $h_{jm} = q_j^\top v$
        $v = v - h_{jm}q_j$
    end for
    $h_{m+1,m} = \|v\|_2$
    $q_{m+1} = v/h_{m+1,m}$
end for

Arnoldi 迭代算法优化

Arnoldi 算法能够高效地从一个 m×m 的低维子空间估计出矩阵的一部分特征值，但是当矩阵的特征值存在分簇现象，子空间的维度 m 会不可避免地变大。此时，Arnoldi 迭代算法会占用较大的存储空间。为解决该问题，一种新的方法被用于改进 Arnoldi 迭代算法[3]，被称为 explicitly restarted Arnoldi method（ERAM）。该方法依旧使用较小维度的 Krylov 子空间，当近似的特征值不满足要求时，算法会在一个新的 Krylov 子空间上重启。新的 Krylov 子空间初始向量b，由上一轮近似的特征值线性组合形成。

当然，ERAM 的性能局限于新的 Krylov 子空间初始向量，不当的线性组合反而可能导致生成一个更差的 Krylov 子空间。为了提升 Arnoldi 算法的性能，一种隐式的方法[4]利用 QR 分解来重启算法，能够实现更高效、更稳定的特征值近似，被称为 implicitly restarted Arnoldi method（IRAM）。当近似的特征值不满足要求时，算法对 Hessenberg 矩阵进行 QR shifted 操作，从而集中所需特征值的信息。QR shifted 之后的矩阵用于重建在 Krylov 子空间的投影，这相当于隐式地使用一个新的初始向量重建 Krylov 子空间，能够有效地减少中间变量的存储空间。

参考文献

[1]. Burden R L. Numerical analysis[M]. Brooks/Cole Cengage Learning, 2011.

[2]. Arnoldi W E. The principle of minimized iterations in the solution of the matrix eigenvalue problem[J]. Quarterly of applied mathematics, 1951, 9(1): 17-29.

[3]. Saad Y. Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices[J]. Linear algebra and its applications, 1980, 34: 269-295.

[4]. Sorensen D C. Implicit application of polynomial filters in ak-step Arnoldi method[J]. Siam journal on matrix analysis and applications, 1992, 13(1): 357-385.

题目：

1. 生成随机矩阵

（1）生成一个 10x10 维度的随机矩阵。

（2）生成一个 10000x10000 维度且密度为 0.001 的随机稀疏矩阵，并统计矩阵中非零元素数量。

（3）利用库函数计算（1）和（2）中矩阵的特征值。

2. 给出 Power Method 的伪代码并用代码实现，能够输出绝对值最大的特征值。

（1）利用 Power Method 计算题目 1（1）中矩阵的绝对值最大的特征值。

（2）利用 Power Method 计算题目 1（2）中稀疏矩阵的绝对值最大的特征值。

3. 给出 QR 算法的伪代码并用代码实现，并能够实现输出前 k 个绝对值最大的特征值，其中 k 为自定义参数。

（1）利用 QR 算法计算题目 1（1）中矩阵的前 4 个绝对值最大的特征值。

（2）利用 QR 算法计算题目 1（2）中稀疏矩阵的前 5 个绝对值最大的特征值。

4. 用代码实现 Arnoldi 迭代算法，并能够实现输出前 k 个绝对值最大的特征值，其中 k 为自定义参数。

（1）利用 Arnoldi 迭代算法计算题目 1（1）中矩阵的前 6 个绝对值最大的特征值。

（2）利用 Arnoldi 迭代算法计算题目 1（2）中稀疏矩阵的前 7 个绝对值最大的特征值。

5.（bonus，非必做）给出 ERAM 或 IRAM 算法的伪代码并用代码实现，并能够实现输出前 k 个绝对值最大的特征值，其中 k 为自定义参数。

（1）利用 ERAM 或 IRAM 算法计算题目 1（2）中稀疏矩阵的前 8 个绝对值最大的特征值。

（2）利用 ERAM 或 IRAM 算法计算题目 1（2）中稀疏矩阵的前 20 个绝对值最大的特征值。

要求：

1. 项目报告需要包括算法描述、关键步骤的代码分析、计算结果分析和性能分析（计算复杂度、空间复杂度）等；
2. 算法描述清晰易懂，报告最好图文并茂，具有可读性，可以使用中文或英文撰写，格式清晰、表达清楚、语言简练；
3. 报告以 pdf 文件的形式提交，以“学号_姓名_project_report.pdf”的格式命名，可以利用 Latex、Markdown、Word、WPS 等软件撰写；
4. 推荐使用如下编程语言：C、C++、Python、Matlab；
5. 可以调用 Matlab，Numpy 等内置库函数来验证结果准确性，但算法实现不能调用计算特征值相关以及算法相关的内置函数；
6. 上传完整的代码，代码文件夹以“project_code”的格式命名，代码可运行（如使用 Python 或 Matlab 需要有一个单独的主函数文件一次性运行所有题目，如使用 C 或 C++需要提供编译好的可执行文件）；
7. 代码需要有合理的注释，变量以及函数的命名方式规范，并需要提交说明文件（如：README.md），说明主要代码文件的功能以及运行环境；
8. 若有使用额外的库，需要文件说明（如：额外的 Python 库，请标明在 requirement.txt 中，可以利用 pip 安装）。
9. 以 zip 压缩包的形式提交文件，以“学号_姓名_project.zip”的格式命名，需要包括报告以及代码，其中报告和代码的具体要求如下所示。此外，在学在浙大上，除压缩包外需要单独提交一份 pdf 文件以方便查看。

文件结构：

```
├─学号_姓名_project
    ├─学号_姓名_project_report.pdf
    ├─project_code
        ├─README.md
        ├─…
```

截止时间：

2023 年 11 月 19 日 23:59 pm