基于威尔金森位移迭代的奇异值分解数值实验报告

陈远洋

2025 年 1 月 16 日

1 问题背景

1.1 奇异值定义回顾

给定 $m \times n$ 矩阵 A，如果 $\exists$ 非零向量 $\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^m, \sigma \geq 0$，使得 $A\mathbf{x} = \sigma\mathbf{y}$，$A^{\mathrm{T}}\mathbf{y} = \sigma\mathbf{x}$。则称 σ 为 A 的一个奇异值，$\mathbf{x}$ 为 A 的一个右奇异向量，$\mathbf{y}$ 为 A 的一个左奇异向量。

由上面的定义，我们很容易得到的奇异值分解形式如下：

$$A = U\Sigma V^{\mathrm{T}} \tag{1}$$

其中，U 为 $m \times m$ 单位正交矩阵，V 为 $n \times n$ 单位正交矩阵。二者分别保存着 A 的左奇异向量和右奇异向量。Σ 为 $m \times n$ 对角矩阵，其对角线上的元素为奇异值。由 Σ 对角元的非零与否，我们可以进一步得到 A 的秩以及简化奇异值分解形式：

$$A = U_r\Sigma_r V_r^{\mathrm{T}} = \sigma_1\mathbf{u}_1\mathbf{v}_1^{\mathrm{T}} + \sigma_2\mathbf{u}_2\mathbf{v}_2^{\mathrm{T}} + \cdots + \sigma_r\mathbf{u}_r\mathbf{v}_r^{\mathrm{T}} \tag{2}$$

其中，$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$。

1.2 奇异值分解的应用

奇异值分解是矩阵分析中经常使用的一种技术。其应用范围广泛，包括图像处理、信号处理、生物信息学、统计学、机器学习等领域。本实验中，我们主要针对矩阵的 2 范数，广义逆，低秩逼近，主成分分析等问题展开讨论。

1.2.1 2 范数

对于矩阵 A，其 2 范数表示为：

$$\|A\|_2 = \max_{\mathbf{x}\neq 0} \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \max_{\mathbf{x}\neq 0} \sqrt{\frac{\mathbf{x}^T A^T A\mathbf{x}}{\mathbf{x}^T\mathbf{x}}} = \sqrt{\lambda_{\max}(A^T A)} = \sigma_{\max}(A) \tag{3}$$

因此，从理论上来讲直接计算奇异值分解来找到所有奇异值，然后取最大值作为 2 范数是最准确的。虽然计算量较大，特别是对于大型矩阵。但是在小规模的情况下，这种方法是行之有效的。

1.2.2 广义逆

广义逆表示矩阵 A 的逆矩阵，即存在矩阵 X，使得

$$\begin{aligned} AXA &= X \\ XAX &= A \\ (AX)^T &= AX \\ (XA)^T &= XA \end{aligned} \tag{4}$$

广义逆的存在是矩阵分析中一个重要的结果。从理论上来讲，如果我们能够得到形如 (2) 式的奇异值分解。那么我们取 $X = V_r\Sigma_r^{-1}U_r^T$，可以验证 X 就是 A 的广义逆。虽然在实际应用中，奇异值分解计算量较大，但这种路径却不失为一种有效的广义逆矩阵计算方法。

1.2.3 低秩逼近

低秩逼近是指矩阵 A 的秩 k 逼近，即存在矩阵 A_k，使得 A_k 的秩不超过 k，使得 $\|A - A_k\|$ 最小。低秩逼近的目的在于降低矩阵的维度，从而简化矩阵的表示。可以证明在矩阵 2 范数以及 F 范数意义下，最佳秩 k 逼近为：

$$A_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}} \tag{5}$$

这种表示形式也通常被称为截断 SVD。

1.2.4 主成分分析

最后考虑一个统计模型。假设已经测得 d 维空间上的随机变量的 n 个样本 $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n$. 这些样本往往聚集在低维集合上（如直线、平面等）。我们希望找到一个低维空间，使得这些样本的分布尽可能的落在这个低维空间。那么我们就能够近似的得到这个随机变量的分布特征。这个低维空间的基常被称为主成分，这种模型被称为主成分分析。只要能够得到这个低维空间的一组基，我们就得到了整个的信息。下面，我们不妨假设样本的均值通过原点（否则通过平移将其移动到原点）。设低维空间为 **M**，Q 为 **M** 的一组标准正交基，那么样本到空间的距离平方和可以表示为：

$$\sum_{i=1}^{n} \|(I - QQ^T)\mathbf{a}_i\|^2 = \sum_{i=1}^{n} \mathbf{a}_i^T(I - QQ^T)\mathbf{a}_i = \sum_{i=1}^{n} \mathrm{tr}((I - QQ^T)\mathbf{a}_i\mathbf{a}_i^T) = \mathrm{tr}(AA^T) - \mathrm{tr}(Q^TAA^TQ) \tag{6}$$

其中 A 的第 i 列为 $\mathbf{a}_i$. 那么我们要使得逼近效果更好，显然我们需要使得 $\mathrm{tr}(Q^TAA^TQ)$ 尽可能的大。那么，由樊氏迹极大极小原理（Fan's Trace Minimax Principle）我们有：当 Q 取 A 的前 k 个奇异值对应的左奇异向量时，原式恰能取到最小值。

1.3 威尔金森位移迭代法

威尔金森位移迭代法是一种奇异值分解的数值算法。其基本思想是，利用 A 的奇异值与 A^TA 的特征值之间的关系，隐式地构造 A^TA 并利用对称 QR 算法进行迭代求解。首先将 A 双

对角化，得到 $A = UBV^T$，其中 U、V 为正交矩阵，B 为 $m \times n$ 双对角矩阵。用 B 的两个对角线元素隐式表示 A^TA，并隐式地利用对称 QR 算法迭代求解 A^TA 的特征值与奇异值。

2 奇异值分解代码构成与分析

奇异值分解实现与测试部分分为 my_svd.m 与 test_svd.m 两个文件。完整代码详见相应代码文件。my_schur.m 文件为 svd 的实现，该分解由三步构成：

1 第一步：对输入矩阵 A 进行双角化变为 B，使得仅在主对角线与右上方第一条次对角线有非 0。

2 第二步：对 B 的两条对角线元素表示的 B^TB 隐式地进行 Wilkinson 位移 QR 迭代，使仅保留主对角线元素非 0。

3 第三步：对对角线上的元素进行排序，特别的遇到负数需要进行调整。

在上面三步中，第一步对应代码部分的 biodiag 函数。第二步对应代码部分的 svd_iter 函数与 pro_iterate 函数。第三步对应代码部分的 sort_svd 函数。在上面三步中，我们默认对 $A \in \mathbb{R}^{m\times n}$ 有 $m \geq n$. 然而实际情况我们也不可能会全部遇到这样的情况。因此我们将上面三步加上一个封装。对列数大于行数的矩阵，我们先计算其转置矩阵的 SVD，然后将结果的 U、V 矩阵转置，得到的结果即为原矩阵的 SVD。然后再分别调用上面的函数。我们将封装函数命名为 *my_svd* 。而 test_svd.m 文件与 testcases 文件夹为检测部分，包含一些简单的测试样例，以及简单的性能分析。

2.1 奇异值分解代码

1. my_svd 函数。

   输入：待求的初始矩阵 A 以及求解精度（可选，默认为 10^{-12}）。

   输出：奇异值分解得到的奇异值 Σ，以及对应的左右奇异向量组成的矩阵 U、V。

   首先调用 biodiag 函数将输入矩阵 A 双对角化，得到双对角化矩阵 B 以及该过程用到的正交变换矩阵 U、V。然后调用 svd_iter 函数迭代求解隐式表示的 B^TB 的特征值与奇异值，得到奇异值序列。最后调用 sort_svd 函数对奇异值序列进行排序，得到按照递减顺序重新排列的奇异值与奇异向量。

2. biodiag 函数。

   输入：待求的初始矩阵 A。

   输出：双对角化矩阵 B 以及该过程用到的正交变换矩阵 U、V。

   利用 Householder 变换将输入矩阵 A 变换双对角化矩阵 B。同时，记录下每一步的正交变换，计算出正交变换矩阵 U、V。一共 $4n-2$ 次 Householder 变换，且每一次变换时间复杂度为 $O(mn+m^2)$，故该部分时间复杂度为 $O(mn^2+m^2n)$。

3. svd_iter 函数

   输入：双对角化矩阵 B 的两条对角线 $\delta\gamma$，以及需要累积正交变换的正交矩阵 U、V，以及收敛精度 ϵ。

   输出；迭代将 γ 消为 0 后的主对角线 δ，以及累积正交变换后的 U、V。

   每次迭代，首先将次对角线 ($|\gamma_k| < \epsilon(|\delta_k| + |\delta_{k+1}|)$) 以及主对角线足够小 ($|\delta_k| < \epsilon\|B\|$) 的元素置为 0，然后分理出最大的不可约子块范围 l:m, 将 U、V、δ、γ 的相应部分其送入 pro_iterate 函数迭代一次。在子块分离的过程中需要保证的是其隐式表示的三对角矩阵具有不可约性。因此对于 $\delta_k = 0$ 且 $\gamma_k \neq 0$ 的情况，我们需要利用下方的 δ 利用 Givens 变换将隐式表示的 B 的第 k 行消为 0. 与此同时确保 $n - l \geq 2$，防止迭代过程出现错误。在 $n < 2$ 时，直接返回。对已经收敛的下方（对应 γ 为 0）保持不动。最终让 γ 化为全 0。结合实际情况，迭代次数大概为 $O(n)$(可见图 3)。

4. pro_iterate 函数

   输入：需要迭代的两条对角线 $\delta\gamma$ 的部分，以及需要累积正交变换的 U、V 子阵。

   输出：一次迭代后的对角线 $\delta\gamma$ 以及累积正交变换后的 U、V 子阵。

   在 svd_iter 函数调用中可知，进行操作 $\delta\gamma$ 隐式表示的三对角矩阵的一定不可约且规模大于 2×2，可以放心进行 Wilkinson 位移 QR 迭代。每步 Givens 变换为两行（列）之间的操作，故时间复杂度为 $O(m)$（不考虑正交矩阵 U、V 的时间复杂度的话为 $O(1)$）。则总时间复杂度为 $O(mn)$。

5. sort_svd 函数

   输入：奇异值分解得到的奇异值序列，以及对应的左右奇异向量组成的矩阵 U、V。

   输出：按照递减顺序重新排列奇异值与奇异向量。

   大多数情况下，经过上述迭代过程中得到的奇异值序列已经基本上有序。为了得到想要的奇异值递减的效果，所以我们只需要对结果进行一些“小小”的调整即可。排序的算法有很多种，简便起见，我们采用了插入排序。在不考虑两个正交矩阵 U、V 交换列产生的情况下，排序的最坏的时间复杂度为 $O(n^2)$。相比整个算法，这样的时间复杂度还是比较可接受的。另外由于上述迭代过程并没有刻意保证对角元非负。而奇异值又是绝对非负的，所以我们需要对为负的对角元进调整。考虑到奇异值分解的 (2) 式的形式，我们只需要将 u_i 与 σ_i 同时乘上负一即可。最后，我们得到了按照递减顺序重新排列的奇异值与奇异向量。

2.2 测试代码

我们将测试代码分为两方面。第一方面为测试 my_svd 函数的正确性。testcases 文件夹中包含一些简单的测试样例。测试样例分为 3 个部分每个部分有 9 个测试样例。第一部分为几个矩阵大小较小的矩阵（1 到 3 行，1 到 3 列），第二部分包含几个中等规模矩阵（50 到 100），第三部分包含几个条件数较大的矩阵（为 5^1 至 5^9）。对奇异值分解的正确性，我们从 Σ 对角元是否是降序排列的正数、U 与 V 是否是正交矩阵、$U\Sigma V^T$ 是否等于 A 的近似等方面进行测

试。第二方面为测试 my_svd 函数的性能。分别从下一节的几个维度展开分析（U、V 正交性，分解结果与 A 的差异，迭代次数，迭代时间）。

3 数值表现

3.1 U、V 的正交性

为了检验左右奇异向量的正交性。下面针对随机生成的 114×514 矩阵进行 SVD 分解。并将 $UU^T - I$ 与 $VV^T - I$ 用 matlab 的 imagesc 函数进行可视化，如图 1 所示。可以看到，每个元素均在 10^{-15} 级别，说明 U、V 的正交性较好。

3.2 分解结果与 A 的差异

简单地，我们可以直接将 SVD 分解得到的结果与原始矩阵 A 进行比较。下面我们对随机生成的 114×514 矩阵进行 SVD 分解，并将分解结果 $U\Sigma V^T$ 与原始矩阵 A 的差值以上面的方式画图比较进行比较。可以看到，两者的差值在 10^{-14} 级别，说明分解结果与原始矩阵 A 的差异较小。（见图 2）

3.3 迭代次数与收敛性

由于我们的奇异值分解求解为一个迭代的过程，所以迭代次数与运行时间复杂度有紧密的关系。所以我们需要对迭代的效率进行分析。一方面，由于我们采取的是 Wilkinson 位移 QR 算法，所以迭代次的渐近收敛速度是三次的。即，迭代次数最多与矩阵的行数（列数）成线性关系。我们可以改变 SVD 函数接口，使其输出我们的迭代步数，将其与矩阵阶数进行比较。对于大小为 2 100 的随机生成方阵，其迭代次数如下图 3 所示。可以看到，迭代次数与矩阵的大小呈线性关系。对一个大小固定的矩阵，我们来分析其收敛的历史。我们以 114×514 矩阵为例，进行 SVD 分解，并将迭代步数与已经奇异值进行画图比较。可以看到，迭代步数与奇异值的关系呈现出线性的关系（如图 4 所示）。且平均一到两次迭代即可收敛出一个奇异值。因此，我们的算法的运行时间复杂度为 $O(m^2n + mn^2)$。

3.4 运行时效分析

为了分析算法的运行时间复杂度，我们可以对算法进行计时。下面我们对 50 到 250 阶方阵进行 SVD 分解，并记录其运行时间。我们将运行时间与矩阵阶数取对数画图，可以看到二者所形成的直线斜率在 2 至 3 之间，说明算法的运行时间复杂度如理论分析是三次。（见图 5）

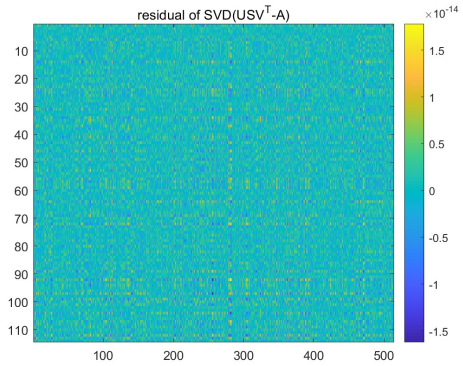

图 2: 随机 114×514 矩阵 $U\Sigma V^T - A$

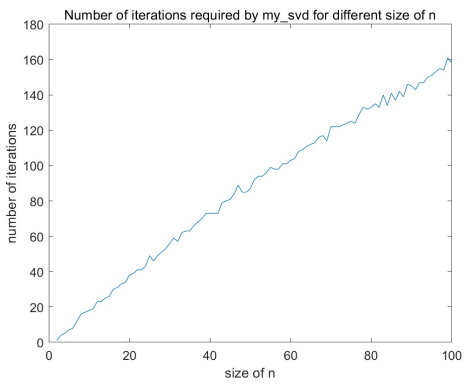

图 3: 迭代次数与方阵阶数关系图示

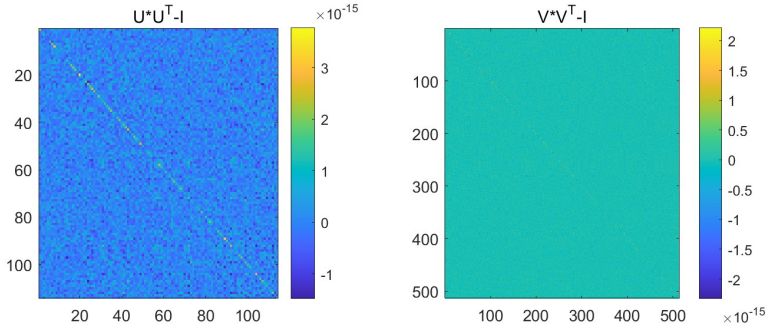

图 1: 随机生成的 114×514 矩阵的左右奇异向量正交性图示

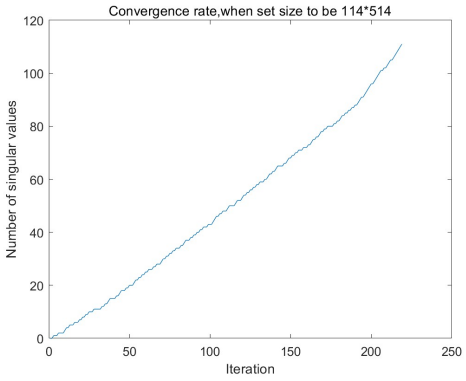

图 4: 随机 114×514 矩阵收敛奇异值个数与迭代次数关系图示

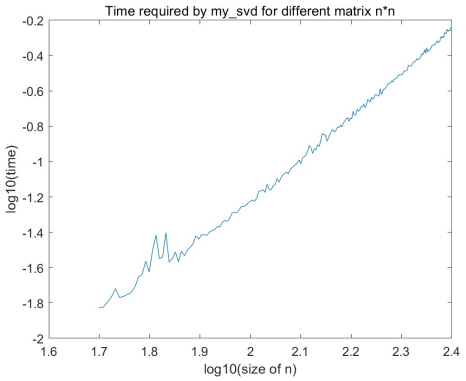

图 5: 50 250 阶方阵 SVD 分解运行时间（log-log）图示

4 实际应用探讨

4.1 2 范数求解

正如第一部分中所探讨的那样，求解 2 范数最精确的做法（尽管不是那样的高效），是求解其最大的奇异值。因此我们利用奇异值分解求解 2 范数的时候，我们只关心奇异值。并不关心

奇异向量。因此，我们可以省去 SVD 迭代步骤中对 U、V 的累积。并且最终，我们也只需要求出对角线上绝对值最大元素即可。这样，我们就可以得到 2 范数。因此，从最后的结果来讲，求解速度应当是显著快于 SVD 的（见图 7）。进一步地，我们可以将结果与 MATLAB 内置的 norm 函数进行比较。可以看到，我们求解的准度还是很高的。结果来看，我们将求解的函数封装为 my_SSnorm2.m。可供其他文件调用。我们在 test_norm.m 文件中对其进行测试与调用。

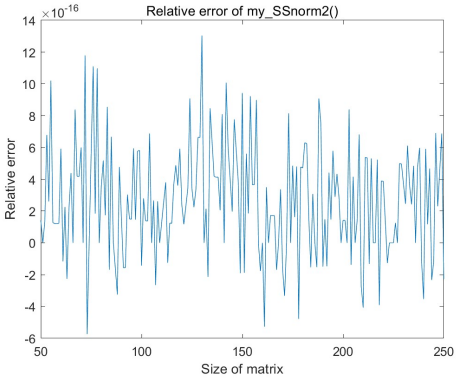

图 6: 相对误差与矩阵（方阵）大小比较

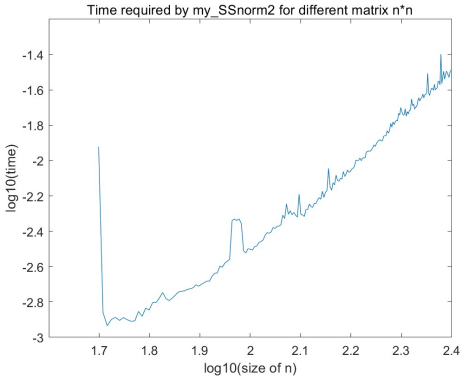

图 7: 运行时间与矩阵（方阵）大小关系（log-log）

4.2 低秩逼近与图像压缩

计算机在显示图像时，往往是向屏幕上打上均匀的正方形网格。因此在存储时，我们以每个像素点为一个元素来说，是一个大的 2 维矩阵。比如黑白图片，每个像素点只需存储一个介于 0 到 255 的灰度值。RGB 格式下的彩色图片，每个像素点需要存储三个颜色分量，因此需要三个元素。来源于生活中的图像，其产生的矩阵往往是低秩的，因为图像往往是平滑的，这意味着我们有很多行（列）都是线性相关的。并且，理想情况下来源于现实生活中的图像可以划分为若干个区域，每个区域内部的灰度、颜色都比较单纯或者平滑，而区域之间有很大的不同，从而形成一些比较明显的分界线。这可以推断出图像对应的矩阵奇异值差异是比较大的。每个区域是低秩的，所以整个图像也是低秩的。

用低秩逼近的想法来看。原来一幅 $m \times n$ 的矩阵 A，需要存储 mn 个元素。如果对 A 做秩 k 逼近。由 (2) 式的截断 SVD，我们只需存储 k 个奇异值，以及对应的 k 个左右奇异向量，一共 $mk + nk + k = k(m + n + 1)$ 个元素。当 m、n 较大时，这种方法可以大大减少存储空间。可以计算，压缩比为

$$\text{压缩比} = \frac{k(m+n+1)}{mn} \tag{7}$$

以代码文件 pro_pic.m 为例。我们的附例为一个 315×399 像素的彩色图片（zby.jpg）。针对 RGB 三个通道分别进行低秩逼近后重组，分别取 $k = 18, 44, 88, 131, 175$ 计算可知压缩比分别为 $10.24\%, 25.03\%, 50.06\%, 74.52\%, 99.55\%$，然后对生成结果分别输出。并且，我们发现实际上 50% 左右的压缩比时，图像的信息已经较为完整。另外需要注意的是，这部分功能的视线依赖于 my_svd.m 文件，因此请务必保证运行该文件时，my_svd.m 文件在当前目录下。具体效果可以参

图 8: 未压缩前原图片效果

图 9: 秩 88 压缩后效果（压缩比 50.06%）

考下图。

5 总结

综上，本实验实现了基于威尔金森位移迭代法的奇异值分解算法，并对其进行了验证。并提供了多个测试样例，用以分解结果是否正确。进一步从左右奇异向量的正交性、奇异值的准确性、迭代次数与收敛性、运行时效分析等方面进行了分析。最后针对 2 个比较有趣的实际应用，（较低维情况矩阵 2 范数求解、低秩逼近与图像压缩），进行了一定的探讨并给出了自己的处理方法或者想法。最终实现效果以及理论分析来看奇异值分解算法运算量约为 $4mn^2 - \frac{4n^3}{3}$（不计算 U、V，否则为 $4m^2n + 8mn^2 + 9n^3$）。

6 参考文献

[1] 徐树方，高立，张文平等. 数值线性代数（第 2 版）. 北京：北京大学出版社，2013.1.