1 Вступление

Multiple Relatively Robust Representations (MR^3) - алгоритм, решающий Tridiagonal Symmetric Eigenvalue Problem ($TSEP$). Подробней о том, как именно он работает и вообще вся данная статья в деталях будет находиться в отдельном pdf-файле. Сейчас нам лишь важно рассмотреть, какие существуют способы использовать данный алгоритм для решения Bidiagonal Singular Value Decomposition ($BSVD$), и как мы их имплементировали. За основу взята [данная диссертация](https://d-nb.info/1002687853/34), 3-я глава.

2 Black Box Approaches

Вообще, как вы могли заметить, сам MR^3 не решает $BSVD$; стало быть, нам необходимо каким-либо способом преобразовать исходную матрицу A_{nxn} к бидиагональному виду, обозначим его B_{nxn}.

$$\mathbf{B} = diag(a_1, \dots, a_n) + diag_{+1}(b_1, \dots, b_{n-1}).$$

$$\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^*$$

$$\mathbf{U}^*\mathbf{U} = \mathbf{V}^*\mathbf{V} = \mathbf{I}\ , Sigma = \mathrm{diag}(\sigma_1, \dots, \sigma_n) sigma_1 \le \dots \le \sigma_n \Gamma$$

Далее встаёт вопрос: а как привести свести решение $BSVD$ К ($TSEP$), чтобы его обработал MR^3? Существуют 2 простых способа, однако доказано (см. диссертацию), что они не годятся для BSVD. Как раз один из них мы и имплементируем, доказав его непригодность.

3 Нормальные уравнения

Зададим 2 уравнения:

$$\mathbf{B}\mathbf{B}^* = \mathbf{U}\Sigma^2\mathbf{U}^* \quad , \quad \mathbf{B}^*\mathbf{B} = \mathbf{V}\Sigma^2\mathbf{V}^*$$

Обе получившиеся матрицы являются тридиагональными.

$$\mathbf{B}\mathbf{B}^* = diag(a_1^2 + b_1^2, \dots, a_{n-1}^2 + b_{n-1}^2, a_n^2) + diag_{\pm 1}(a_2 b_1, \dots, a_n b_{n-1})$$

$$\mathbf{B}^*\mathbf{B} = diag(a_1^2, a_2^2 + b_1^2, \dots, a_n^2 + b_{n-1}^2) + diag_{\pm 1}(a_1 b_1, \dots, a_{n-1} b_{n-1})$$

4 Матрица Голуба-Кахана

Имея бидиагональную матрицу $\mathbf{B}$ можно рассмотреть задачу нахождения нахождения сингулярных векторов на матрице

$$\begin{bmatrix} 0 & \mathbf{B} \\ \mathbf{B}^* & 0 \end{bmatrix}$$

Если $\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^*$,то $\mathbf{B}^* = \mathbf{V}\Sigma\mathbf{U}^*$ Эти два равенства позволяют сделать следующее разложение:

$$\begin{bmatrix} 0 & \mathbf{B} \\ \mathbf{B}^* & 0 \end{bmatrix} = \mathbf{J} \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma \end{bmatrix} \Sigma^* \qquad \mathbf{J} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{U} & \mathbf{U} \\ -\mathbf{V} & \mathbf{V} \end{bmatrix}.$$

Пермутацией этой матрицы можно получить матрицу Голуба-Кахана

$$T_{GK}(B) = diag_{+1}(a_1, b_1, a_2, b_2, \dots, a_{n-1}, b_{n-1}, a_n)$$

$$\text{Сингулярная тройка } (\sigma, u, v) \text{ матрицы } \mathbf{B} \quad ||u|| = ||v|| = 1$$

Взаимно однозначна с собственной парой

$$(\pm\sigma, q) \text{ матрицы } T_{GK}(B), \text{ при } ||q|| = 1, \sqrt{2}q^* = [v_1, u_1, v_2, u_2, \dots, v_n, u_n]$$

5 Имплементация

Оба способа позволяют свести BSVD к TSEP. Но подход с нормальными уравнениями очевидно неудачен заранее - мы буквально перемножаем элементы матрицы, что влечёт за собой потерю точности. А вот с подходом через матрицу Голуба-Кахана всё менее однозначно, и поэтому мы выбрали данный подход для имплементации. Реализацией MR^3 в этом случае будет алгоритм библиотеки линейной алгебры LAPACKE (оболочка LAPACK) - [dstemr](https://netlib.org/lapack/explore-3.2-html/dstemr.f.html), а в целом для совместимости с основным фреймворком - Eigen. К сожалению, в связи с внутренними проблемами LAPACKE и Eigen не удалось рассмотреть работу алгоритма для матриц размера больше, чем 3x3. Все ошибки и комментарии по этому поводу находятся в mrrr.h