Алгоритм быстрого вычисления SVD с помощью QR-алгоритма и бидиагонализации

Шамаев Климент

НИУ ВШЭ

December 8, 2025

Бидиагональные матрицы

Бидиагональные матрицы — это матрицы вида:

1. $A \in M_n$ — Верхняя бидиагональная, если A имеет вид

$$\begin{pmatrix} a_1 & b_1 & 0 & \dots & 0 \\ 0 & a_2 & b_2 & \dots & 0 \\ 0 & 0 & a_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_n \end{pmatrix}$$

2. A — Нижняя бидиагональная, если A^T — верхняя бидиагональная

Оператор Хаусхолдера

Оператором Хаусходера (или оператором отражения) вектора v называется оператор

$$H_v(x) = x - 2(x, v)v.$$

Данный оператор описывает отражение вектора x относительно плоскости $v^{\perp}$.
Матрицей Хаусходера (или матрицей отражения) вектора v называется матрица

$$H_v = E - 2vv^T$$

Вращение Гивенса

Матрицей вращения Гивенса $G(k, l, \phi)$ назовём матрицу

$$
\begin{pmatrix}
1 & \dots & 0 & \dots & 0 & \dots & 0 \\
\vdots & \ddots & \vdots & & \vdots & & \vdots \\
0 & \dots & \cos\phi & \dots & -\sin\phi & \dots & 0 \\
\vdots & & \vdots & \ddots & \vdots & & \vdots \\
0 & \dots & \sin\phi & \dots & \cos\phi & \dots & 0 \\
\vdots & & \vdots & & \vdots & \ddots & \vdots \\
0 & \dots & 0 & \dots & 0 & \dots & 1
\end{pmatrix},
$$

где подматрица $\begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}$ расположена на строках и столбцах с номерами k и l.

Алгоритм Бидиагонализации: формулировка

Задача: $A \in \mathbb{R}^{m \times n}, m \geq n \to U^T A V = \begin{pmatrix} B \\ 0 \end{pmatrix}$, где B — бидиагональная, U, V — ортогональные. То есть, требуется свести матрицу A к бидиагональной посредством применения ортогональных операторов.

Идея: в качестве ортогональных операций будем использовать отражения.

Алгоритм Бидиагонализации: иллюстрация

$$
\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}
\xrightarrow{U_1}
\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}
\xrightarrow{V_1}
$$

$$
\begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}
\xrightarrow{U_2}
\begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}
\xrightarrow{V_2}
$$

$$
\begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}
\xrightarrow{U_3}
\begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & \times \end{bmatrix}
\xrightarrow{U_4}
\begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \end{bmatrix}.
$$

Алгоритм Бидиагонализации: Вычисление

Итеративный метод «Golub-Kahan bidiagonalization» реализует процесс, описанный выше. Сложность составляет $O(m^2 n)$. Метод является численно устойчивым.

Применение бидиагонализации в вычислении SVD

Научимся вычислять *SVD* верхней бидиагональной матрицы. Тогда можно будет вычислять *SVD* произвольной матрицы, сводя её к бидиагональной и вычисляя *SVD* для неё.

QR-алгоритм

Пусть $A_0 = A$ — исходная матрица. Для $k = 0, 1, 2, \dots$ выполним

1. $A_k = Q_k R_k$, где — унитарная (ортогональная) матрица, — верхняя треугольная матрица — QR разложение
2. $A_{k+1} = R_k Q_k$

Алгоритм сходится к верхнетреугольной матрице с собственными значениям исходной матрицы на главной диагонали.

QR-алгоритм

Сингулярное разложение матрицы B можно найти с помощью обычного QR-алгоритма (применяя к $B^T B$). Но это не оптимально: не пользуемся бидиагональностью матрицы.

QR-алгоритм: быстрое вычисление

Будем выполнять каждую итерацию QR-алгоритма неявно. Для ускорения алгоритма будем также применять сдвиг. Псевдокод:

Let μ be the eigenvalue of the trailing 2-by-2 submatrix of $T = B^T B$
that is closer to t_{nn}.

$y = t_{11} - \mu$

$z = t_{12}$

for $k = 1{:}n-1$

Determine $c = \cos(\theta)$ and $s = \sin(\theta)$ such that

$$\begin{bmatrix} y & z \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} * & 0 \end{bmatrix}$$

$B = BG(k, k+1, \theta)$

$y = b_{kk}$; $z = b_{k+1,k}$

Determine $c = \cos(\theta)$ and $s = \sin(\theta)$ such that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix}$$

$B = G(k, k+1, \theta)^T B$

if $k < n-1$

$y = b_{k,k+1}$; $z = b_{k,k+2}$

end

end

QR-алгоритм — корректность и сложность

Доказывается (см. источник), что выполнение итерации неявно корректно.
Выполнение одной итерации таким образом займёт всего $O(n)$: на каждый элемент диагонали приходится по 2 поворота, вычислени одного поворота происходит за $O(1)$. Вычисление итерации таким образом численно устойчиво.

Итоги

Каждый из шагов алгоритма вычисляется устойчиво. Итоговая сложность составит $O(m^2 n)$.

Источники (кликабельно)

1. Алгоритмы бидиагонализации (5.4.2) и итогового вычисления SVD (8.6.1, 8.6.2)