Автор: Кулибаба Данил;

danil.kulibaba@ya.u

Краткое описание

Изначально нам дана матрица $A \in Mat(n, m), n \geq m$. В обратном случае, мы можем транспонировать матрицу и найти SVD для нее, потому что таким образом мы найдём SVD и для изначальной матрицы. Вот доказательство того, что это можно сделать

$$A^T = U'S'V'^T$$

$$A = (A^T)^T = (U'S'V'^T)^T = V'S'^T U'^T = USV^T$$

Существует два варианта метода Якоби – односторонний и двусторонний. Двусторонний подходит только для симметричных квадратных матриц. Цель нашего проекта – алгоритмы получения сингулярного разложения **для матриц общего вида**, поэтому мы его рассматривать не будем, хотя он очень схож с односторонним.

Суть одностороннего метода Якоби состоит в том, чтобы с помощью последовательности поворотов сделать так, чтобы столбцы матрицы стали ортогональными. Некоторые столбцы матрицы могут стать нулевыми, но в этом нет ничего страшного

Поворотом мы называем матрицу поворота Якоби $R \in Mat(m, m)$. Индексом поворота будем называть пару (i, j). Поворот с индексом (i, j) приводит матрицу A к матрице $A' = AR^{(i,j)}$, у которой $a'_{ij} = a'_{ji} = 0$. Формулы, по которым высчитываются элементы матрицы R, будут приведены в псевдокоде. Основная мысль – матрица поворота позволяет занулять элементы исходной матрицы, и мы пользуемся этим свойством.

Очевидно, что зануляя случайные элементы матрицы, мы не приведём её к нужному виду. Нужна стратегия выбора индекса следующего поворота. Изначально было доказано, что алгоритм остается корректным, если использовать циклическую стратегию выбора поворота: поочередно применяются повороты с индексами $(1,2), (1,3), \dots, (1, m), (2,3), (2,4), \dots, (2, m), \dots, (m-1, m)$. Однако такой алгоритм требует очень много времени, поэтому мы будем использовать стратегию «с выбором цели» (Jacobi Target Selection). Она описана в псевдокоде. Такая стратегия среди всех известных обеспечивает наибыстрейшую сходимость алгоритма.

Допустим, мы привели матрицу A к матрице $B = AV_1 \dots V_t$, где V_i – матрица i-ого поворота, у которой все столбцы ортогональны. Найдём матрицы U и S Найдем норму $\|b_i\|_2$ каждого ненулевого столбца b_i. Поменяем столбцы матрицы B так, чтобы их нормы шли в порядке невозрастания, т.е. так, чтобы $\|b_i\|_2 \leq \|b_j\|_2$ при $i > j$. Если мы меняем b_i и b_j местами, то также нужно поменять местами строки v_i, v_j матрицы V. Матрица S формируется следующим образом:

$$S_{ij} = 0 \text{ при } i \neq j, ;$$

$$S_{ii} = \begin{cases} \|b_i\|_2, b_i - \text{ненулевойстолбец} \\ 0, \text{иначе} \end{cases}$$

Нормируем столбцы матрицы B, допишем к ним $n - m$ нулей, чтобы матрица B стала $n \times n$. Теперь заменим нулевые столбцы матрицы B столбцами, ортогональными ненулевым столбцам – первые m компонент этих столбцов должны равняться 0, а одна из оставшихся компонент – 1, остальные также должны быть равными 0. Так мы получаем матрицу U.

Получаем сингулярное разложение U, S, V.

Матрицу будем считать ортогональной, когда $b_i^T b_j < \varepsilon \|A\|_F = \varepsilon \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}^{m}}$, где ε – параметр сходимости, произвольное маленькое число, которое вводит пользователь.

Псевдокод

Вход: матрица $A \in Mat(n,m), n > m, \tau \geq 1, \varepsilon > 0$;

Выход: матрицы $U \in Mat(n,n), S \in Mat(n,m), V \in Mat(m,m)$

$\delta \leftarrow \varepsilon \|A\|_F$

$B \leftarrow A$

$V \leftarrow I_m \text{Под} I_n \text{имеетсяввидуединичнаяматрица} m \times m$

$for\ i \leftarrow 1 \frac{m(m-1)}{2}\ do$

$P' \leftarrow$ пустой массив троек $\left(j, k, b_j^T b_k\right)$

$forj \leftarrow 1m - 1do$

$fork \leftarrow j + 1mdo$

$insert\left(j, k, b_j^T b_k\right) \in P'$;

$P \leftarrow \frac{1}{\tau} \text{частьпоследнихэлементов} P'; \text{немогуадекватноперевестинарусский},$

$top \frac{1}{\tau} fractionofelementsofP'$

Отсортировать элементы P по скалярным произведениям в убывающем порядке

$if \text{наибольшеескалярноепроивезрдениев} P < \delta then$

Вычислить U, S, V как указано в кратком описании;

$returnU, S, V$;

$Q \leftarrow \text{пустаяочередьматрицповоротовЯкоби}$

$foreach(j, k, d) \in Pdo$

$\gamma = \frac{a_k^T a_k - a_j^T a_j}{2 a_j^T a_k}$;

$t = \frac{sgn(\gamma)}{|\gamma| + \sqrt{\gamma^2 + 1}}$;

$c = \frac{1}{\sqrt{1 + t^2}}; s = tc$;

$J - \text{матрицасединицаминадиагонали, укоторой} J_{jj} = J_{kk} = c, J_{jk} = -s, J_{kj} = s$;

$push(Q, J)$

$whileQ - \text{непуста} do$

$J = pop(Q)$

$B \leftarrow BJ$;

$V \leftarrow VJ$;