Семинар 3

Сложность вычисления произведения

Пусть $A \in \mathrm{M}_{m\,k}(\mathbb{R})$ и $B \in \mathrm{M}_{k\,n}(\mathbb{R})$ – некоторые матрицы. Чтобы посчитать матрицу AB нужно найти mn ее коэффициентов. Причем каждый коэффициент требует k умножений и $k-1$ сложение. Обычно сложения игнорируются в таких подсчетах. Итого, получается $O(mkn)$ операций. Если матрицы $A, B \in \mathrm{M}_n(\mathbb{R})$, то получается n^3 операций.

Пусть теперь даны матрицы $A, C \in \mathrm{M}_{1\,n}(\mathbb{R})$ и $B \in \mathrm{M}_{n\,1}(\mathbb{R})$.

$$\left(\begin{pmatrix} a_1 & a_2 & \ldots & a_n \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}\right) \cdot \begin{pmatrix} c_1 & c_2 & \ldots & c_n \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & \ldots & a_n \end{pmatrix} \cdot \left(\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \cdot \begin{pmatrix} c_1 & c_2 & \ldots & c_n \end{pmatrix}\right)$$

Тогда посчитать произведение ABC можно двумя способами: (1) $(AB)C$ и (2) $A(BC)$. Давайте сравним количество операций для вычислений. В пером случае AB считается за n умножений и получаем число, которое умножается на C за n умножений. Итого $2n$ операций. Во втором случае BC считается за n^2 умножений и получается n на n матрица, которая умножается слева на A за n^2 операций. Итого $2n^2$ умножений. Как мы знаем, результат от расстановки скобок не зависит, но скорость вычисления будет сильно отличаться.

LU-разложение

Пусть $A \in \mathrm{M}_n(\mathbb{R})$ – квадратная матрица. Мы хотим понять можно ли представить ее в следующем виде: $A = LU$, где

$$L = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ * & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ * & * & \ldots & 1 \end{pmatrix} \quad \text{и} \quad U = \begin{pmatrix} * & * & \ldots & * \\ 0 & * & \ddots & \vdots \\ \vdots & \vdots & \ddots & * \\ 0 & 0 & \ldots & * \end{pmatrix}$$

и если можно, то найти это представление. Вместо звездочек могут стоять любые числа. Матрица U в этом случае называется верхнетреугольной. А матрица L нижнетреугольной с единицами на диагонали. Есть другая версия LU разложения, в которой наоборот L нижнетреугольная, а U верхнетреугольная с 1 на диагонали.

В начале давайте заметим, что если мы разрешим над матрицей A элементарные преобразования строк первого типа, когда мы вычитаем из нижней строки более высокую с коэффициентом, то это преобразование соответствует матрице $S_{ij}(\lambda)$, где $i > j$. То есть эта матрица будет нижнетреугольной и с единицами на диагонали. Более того $S_{ij}^{-1}(\lambda) = S_{ij}(-\lambda)$. Ясно, что произведение нижнетреугольных матриц с единицами на диагонали дает нижнетреугольную матрицу с единицами на диагонали. Наоборот, если L – какая-то нижнетреугольная матрица с единицами на диагонали, давайте покажем, что она раскладывается в произведение элементарных преобразований первого типа, когда мы прибавляем более верхние строки к более низким. Для этого заметим, что матрицу L можно привести такими преобразованиями к единичной матрице. Это означает, что $U_k \ldots U_1 L = E$, где $U_1, \ldots, U_k$ – последовательность матриц соответствующих элементарным преобразованиям строк. А значит $L = U_1^{-1} \ldots U_k^{-1}$. Это равенство дает искомое разложение. То есть нижнетреугольные матрицы с единицами на диагонали – это в точности матрицы накопленных элементарных преобразований, когда нам разрешено лишь прибавлять более высокие строки с коэффициентом к более низким.

Если для матрицы A существует разложение $A = LU$ как указано выше, то $L^{-1}A = U$. А так как L^{-1} нижнетреугольная матрица с 1 на диагонали, то это равенство означает, что матрицу A можно привести к верхнетреугольному виду с помощью только прибавления более высоких строк к нижним. Наоборот, если мы можем привести A такими преобразованиями к верхнетреугольной матрице, то $U_k \ldots U_1 A = U$. В этом случае положим $L = U_1^{-1} \ldots U_k^{-1}$, она будет нижнетреугольной с 1 на диагонали и $A = LU$. Таким образом мы получили критерий для разложения.

Алгоритм получения такого разложения выглядит следующим образом. Мы можем начать с первой строки матрицы A и занулить с помощью a_{11} все элементы под ним. Потом перейти ко второй строке и занулить все элементы под a_{22} и т.д. Предположим, что привести не удалось, это значит, что у нас случилось что-то вроде ситуации ниже

$$\begin{pmatrix} * & * & * & * & * \\ & * & * & * & * \\ & & 0 & * & * \\ & & * & * & * \\ & & * & * & * \end{pmatrix}$$

В противном случае, мы приводим матрицу A к матрице вида

$$U = \begin{pmatrix} d_1 & * & \dots & * \\ & d_2 & \dots & * \\ & & \ddots & \vdots \\ & & & d_n \end{pmatrix}$$

Если мы хотим восстановить матрицу L, то для этого можно хранить матрицу $(A|E)$ и приводить ее к верхнетреугольному виду указанными преобразованиями строк. Действия со строками равносильны умножению на нижнетреугольную матрицу с единицами на диагонали, пусть это будет матрица D. Тогда в результате алгоритма мы либо встретим проблему с нулем на диагонали и не нулем под ним. Либо мы придем к виду $(U|D)$. В этом случае матрица $L = D^{-1}$. Так как матрица D нижнетреугольная, то ее обратная ищется быстрее, чем в общем случае.

Хочу обратить внимание на важную вещь. В алгоритме выше мы разрешаем любую последовательность элементарных преобразований, когда более высокая строка прибавляется к более низкой. То есть на первом шаге не обязательно занулять элемент a_{21}, мы можем действовать в любом порядке. Платой за такую свободу является дополнительные затраты на восстановление матрицы L (как хранение вспомогательной матрицы D, так и еще ее обращение). Однако, если разрешить делать преобразования в строго определенном порядке, то вычисление матрицы L не потребует дополнительных действий и памяти. Давайте будем всегда сначала занулять коэффициенты первого столбца сверху вниз, потом второго и т.д. Схематично процесс изображен ниже

$$A = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \mapsto \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \mapsto \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ * & * & * & * \end{pmatrix} \mapsto \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{pmatrix} \mapsto$$

$$\mapsto \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & * & * & * \end{pmatrix} \mapsto \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & * & * \end{pmatrix} \mapsto \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{pmatrix} = U$$

Это означает, что мы сделали последовательность преобразований

$$U = S_{34}(\lambda_6)S_{24}(\lambda_5)S_{23}(\lambda_4)S_{14}(\lambda_3)S_{13}(\lambda_2)S_{12}(\lambda_1)A$$

Тогда

$$A = S_{12}^{-1}(\lambda_1)S_{13}^{-1}(\lambda_2)S_{14}^{-1}(\lambda_3)S_{23}^{-1}(\lambda_4)S_{24}^{-1}(\lambda_5)S_{34}^{-1}(\lambda_6)U$$

То есть

$$L = S_{12}^{-1}(\lambda_1)S_{13}^{-1}(\lambda_2)S_{14}^{-1}(\lambda_3)S_{23}^{-1}(\lambda_4)S_{24}^{-1}(\lambda_5)S_{34}^{-1}(\lambda_6) = S_{12}(-\lambda_1)S_{13}(-\lambda_2)S_{14}(-\lambda_3)S_{23}(-\lambda_4)S_{24}(-\lambda_5)S_{34}(-\lambda_6)$$

Чтобы понять как будет выглядеть матрица L, то достаточно ее умножить на E справа. А это значит применить к E элементарные преобразования соответствующие множителям слева на право. Получится следующее

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\lambda_6 & 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -\lambda_5 & -\lambda_6 & 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\lambda_4 & 1 & 0 \\ 0 & -\lambda_5 & -\lambda_6 & 1 \end{pmatrix} \mapsto$$

$$\mapsto \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\lambda_4 & 1 & 0 \\ -\lambda_3 & -\lambda_5 & -\lambda_6 & 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\lambda_2 & -\lambda_4 & 1 & 0 \\ -\lambda_3 & -\lambda_5 & -\lambda_6 & 1 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\lambda_1 & 1 & 0 & 0 \\ -\lambda_2 & -\lambda_4 & 1 & 0 \\ -\lambda_3 & -\lambda_5 & -\lambda_6 & 1 \end{pmatrix} = L$$

Кроме того, можно организовать вычисления так, чтобы хранить элементы $-\lambda_i$ вместо нулей в матрице U. А именно, повторим процесс в самом начале

$$A = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \mapsto \begin{pmatrix} * & * & * & * \\ -\lambda_1 & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \mapsto \begin{pmatrix} * & * & * & * \\ -\lambda_1 & * & * & * \\ -\lambda_2 & * & * & * \\ * & * & * & * \end{pmatrix} \mapsto \begin{pmatrix} * & * & * & * \\ -\lambda_1 & * & * & * \\ -\lambda_2 & * & * & * \\ -\lambda_3 & * & * & * \end{pmatrix} \mapsto$$

$$\mapsto \begin{pmatrix} * & * & * & * \\ -\lambda_1 & * & * & * \\ -\lambda_2 & -\lambda_4 & * & * \\ -\lambda_3 & * & * & * \end{pmatrix} \mapsto \begin{pmatrix} * & * & * & * \\ -\lambda_1 & * & * & * \\ -\lambda_2 & -\lambda_4 & * & * \\ -\lambda_3 & -\lambda_5 & * & * \end{pmatrix} \mapsto \begin{pmatrix} * & * & * & * \\ -\lambda_1 & * & * & * \\ -\lambda_2 & -\lambda_4 & * & * \\ -\lambda_3 & -\lambda_5 & -\lambda_6 & * \end{pmatrix}$$

Теперь верхняя часть из звездочек – это матрица U, а нижняя часть дополненная единицами на диагонали дает матрицу L.

Замечание

- Давайте явно проговорим, что LU-разложение НЕ обязательно существует для матрицы. Вот пример

  $$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

  В этом случае уже первый шаг алгоритма не срабатывает, а значит не существует такого разложения.

- Если LU-разложение существует оно не обязано быть единственным. Самый простой пример – нулевая матрица. Тогда можно взять $U = 0$ (она тоже верхнетреугльная, просто все $*$ нули) и любую L – нижнетреугольную с 1 на диагонали. Тогда $LU = 0$. Можно привести примеры и для ненулевых матриц, но уже этого примера достаточно.

LU и невырожденные матрицы В случае если матрица $A \in \mathrm{M}_n(\mathbb{R})$ оказалась невырожденная и для нее существует LU-разложение $A = LU$, то обе матрицы $L, U \in \mathrm{M}_n(\mathbb{R})$ тоже будут невырожденными (мы уже видели, что произведение обратимо тогда и только тогда, когда каждый множитель обратим). В этом случае LU-разложение обязательно единственное. Действительно, если предположить, что мы имеем два разложения

$$A = L_1 U_1 = L_2 U_2$$

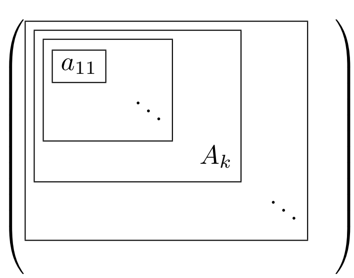
Тогда мы можем умножить на L_2^{-1} слева и U_1^{-1} справа и получим

$$L_2^{-1} L_1 = U_2 U_1^{-1}$$

Но справа стоит верхнетреугольная матрица, а слева нижнетреугольная с 1 на диагонали. А значит такое возможно лишь когда и левая и правая часть равны единичной матрице, то есть

$$L_2^{-1} L_1 = E = U_2 U_1^{-1} \;\Rightarrow\; L_1 = L_2 \;\text{ и }\; U_1 = U_2$$

Кроме того, в этом случае мы можем сформулировать критерий существования LU-разложения (пусть и не очень хорошо проверяемый на практике). Давайте в матрице A выделим диагональные подматрицы натянутые на несколько первых строк

$A =$ 

Заметим, что если мы приводим матрицу A к матрице U с помощью преобразований первого типа лишь прибавляя более высокие строки к более низким, то для каждой подматрицы A_k мы либо ее не меняем, либо выполняем точно такое же преобразование. В частности это означает, что над каждой подматрицей

A_k выполняется последовательность элементарных преобразований. А значит если блок A_k был невырожден в матрице A, то соответствующий блок размера k на k будет невырожден и в матрице U. И аналогично наоборот.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \mapsto \ldots \mapsto \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix}$$

Но так как U верхнетреугольная и обязательно обратимая, то ее угловые блоки – это верхнетреугольные матрицы с ненулевыми числами на диагонали. Значит мы можем сформулировать критерий существования LU разложения для невырожденной матрицы так: оно существует тогда и только тогда, когда все блоки A_k невырождены.

PLU-разложение Как мы видели не всякая квадратная матрица $A \in \mathrm{M}_n(\mathbb{R})$ обладает LU-разложением. Однако, если предварительно переставить в матрице A строки, то всегда можно достичь этой перестановкой существования LU-разложения. Матрица перестановки есть перемножение матриц вида U_{ij} и представляет из себя матрицу, в которой в каждом столбце и в каждой строке только один не нулевой элемент 1. Такую матрицу можно получить из E применением перестановки к ее строкам. Если мы обозначим матрицу перестановки за P, то получится $PA = LU$. Обратите внимание, что такое разложение означает, что мы должны сначала заранее угадать как надо переставить все строки матрицы A и лишь потом использовать алгоритм нахождения LU-разложения. Мы не можем просто так перемешивать эти шаги ибо если сделать перестановку где-то между преобразованиями прибавления верхней строки к нижней, то это действие может превратиться в итоге в прибавление нижней строки к верхней и сломается нижняя треугольность матрицы L. Потому для аккуратного рассказа как именно выполняется выбор P нужно выполнять преобразования еще более аккуратно, но я позволю себе об этом умолчать.

Зачем нужны такие разложения Если мы хотим решать систему $Ax = b$, где $A \in \mathrm{M}_n(\mathbb{R})$, $b \in \mathbb{R}^n$ и $x \in \mathbb{R}^n$, то мы можем это сделать Гауссом за $O(n^3)$. Но если нам надо решить k систем $Ax = b_i$, то суммарно нам надо будет сделать $O(kn^3)$ операций. С другой стороны, мы можем сначала за $O(n^3)$ найти LU разложение. Но тогда система $LUx = b$ решается за $O(n^2)$. Потому что треугольный вид матриц L и U означает, что сложность решения будет как в обратном ходе гаусса. Тогда решение k систем займет $O(n^3 + kn^2)$ операций, что существенно ниже если k порядка n или более. Интуитивно это можно понимать так: LU-разложение позволяет вместо прямого и обратного метода гаусса сделать два обратных в противоположные стороны и это дешевле. Но за это придется один раз в начале заплатить полную цену прямого хода гаусса. Потому бонус может быть лишь при массовом решении систем с одинаковой левой частью.