LU Decomposition и приложения на метода

Боян Дафов, КН, курс 1, поток 1, ФН 82018

30 април 2020 г.

1 Основна идея и дефиниция

Идеята на LU Decomposition е да представяме дадена квадратна матрица като произведение на други две матрици от определен тип. За да опишем метода ще трябва да въведем две прости дефиниции.

Definition 1.1. Долно триъгълна матрица ще наричаме всяка матрица от вида $L = (l_{ij}) \in M_{nxn}(F) : i > j \implies l_{ij} = 0$.

Definition 1.2. Горно триъгълна матрица ще наричаме всяка матрица от вида $U = (u_{ij}) \in M_{nxn}(F) : i < j \implies u_{ij} = 0$.

Нека въз основа на тези две дефиниции да дефинираме LU Decomposition.

Definition 1.3. Нека $A \in M_{nxn}(F)$ тогава казваме, че за матрицата A има LU Decomposition $\Leftrightarrow \exists L(\text{долно триъгълна}), U(\text{горно триъгълна}) : A = LU$.

Въпреки, че на пръв поглед така дефинирана, LU Декомпозицията изглежда ясна и изчерпателна като идея, се оказва че не е особено практична и удобна за използване. Със следващите две твърдения ще илюстрираме основните два проблема на така дефинираната LU Декомпозиция.

Theorem 1. *Не за всички квадратни матрици има така дефинирана LU Декомпозиция.*

Доказателство. Нека $A \in M_{nxn}(R) : a_{11} = 0$, det(A) != 0 и нека $\exists L, U :$ A = LU $\implies$ От дефиниция на L и U : $a_{11} = l_{11} * u_{11} \implies l_{11} = 0 \vee u_{11} = 0 \implies [(l_{11}, \ldots, l_{1n}) = (0, \ldots, 0)] \vee [(u_{11}, \ldots, u_{n1}) = (0, \ldots, 0)] \implies$ det(L)=0$\vee$det(U)=0 $\implies$ det(A) = 0, което е нротиворечие с допускането, че за A има LU Декомпозиция. □

С това твърдение показахме, че не можем да сме сигурни, че аке вземем една квадратна матрица, то ако искаме да намерим LU Декомпозицията, това изобщо е възможно. Това само по себе си е някакъв вид неудобство, но при тази дефиниция на LU Декомпозиция има и още един проблем.

Theorem 2. *Ако за квадратната матрица A, съществува LU Декомпозиция, то тя не е единствена.*

Доказателство. Нека $A \in M_{nxn}(R) : \exists L, U : A = LU$. Нека $D = (d_{ij})_{nxn}$ е диагонална матрица с произволни ненулеви елементи по диагонала. Тогава D е обратима и D^{-1} е диагонална (обратната на диагонална матрица е диагонална). Имаме, че $A = LU = LIU = LDD^{-1}U = (LD)(D^{-1}U)$. D е диагонална, значи е долно триъгълна и от (Lemma 4) $\implies$ LD е долно триъгълна. D^{-1} е диагонална, значи е горно триъгълна и от (Lemma 4) $\implies$ $D^{-1}U$ е горно триъгълна. По този начин по безкраен брой начини можем да избираме матрицата D и да получаваме безкраен брой двойки LD и $D^{-1}U$, които да ни задават сами по себе си LU Декомпозиция. □

Последното може да се види и ако разпишем LU Декомпозицията като система от линейни уравнения. Ако приемем, че това е дефиницията на LU Декомпозиция, с която трябва да работим, то трябва да се съобразяваме със следното твърдение, за да си гарантираме, че изобщо можем намерим такава.

Theorem 3. *Нека $A \in M_{nxn}$, тогава ако A може да се приведе в горно диагонален вид чрез елементарни преобразувания по редове, без да правим размествания на редове, то за A има LU Декомпозиция.*

Доказателство. Нека $A \in M_{nxn}$: $\mathrm{E}_k E_{k-1} \dots E_1 A = U$, където $E_1 \dots E_k$ са матрици на преобразувания по редове. Не сме извършвали размествания на редове, а всички преобразувания са от вида умножаване на ред с число и добавяне към по-долен ред. Тогава лесно може да се провери, че при това положение всяка матрица E_i, i = 1... k е долно триъгълна и има ненулеви елементи по диагонала $\implies$ От (Lemma 5) $E_1, \dots, E_k$ са обратими и $(E_1 \dots E_k)$ е долно триъгълна (доказва се чрез индукция и Lemma 4) $\implies A = LU$, където $L = (E_1 \dots E_k)^{-1}$ е долно триъгилна (Lemma 6). □

Тук накратко съм формулирал лемите, които по-горе съм използвал в доказателствата си. Доказателствата на тези твърдения са относително лесни, но не съм представил всички от тях тук.

Lemma 4. Произведението на две долно (горно) триъгълни матрици е долно (горно) триъгълна матрица.

Доказателство. Нека $A, B \in M_{nxn}$ са долно триъгълни. Трябва да докажем, че $(AB)_{ij} = 0$ когато $j > i$. Това се вижда от

$$(AB)_{ij} = \sum_{k=1}^{n} A_{ik} * B_{kj} = \sum_{k=1}^{i} A_{ik} * B_{kj} + \sum_{k=i+1}^{n} A_{ik} * B_{kj}$$
$$= \sum_{k=1}^{i} A_{ik} * 0 + \sum_{k=i+1}^{n} A_{ik} * B_{kj} = \sum_{k=i+1}^{n} 0 * B_{kj} = 0$$

при $j > i$ което доказва твърдението. □

Lemma 5. Триъгълна матрица (горно или долно) е обратима ⇔ елементите по диагонала са ненулеви.

Доказателство. ⇐/ Нека L е долно триъгълна матрица NxN, която има нулев елемент по главния диагонал на i-ти ред, тоест $L_{ii} = 0$. Нека разгледаме подматрицата A на L, сформирана от първите i-реда на L и всичките колони на L (тоест $A \in M_{in}$). Тогава i-тата колона на A е нулева, защото $L_{ii} = 0$ и L е долно триъгълна. Тогава A има най-много i - 1 ненулеви колони $\implies$ има най-много i - 1 ЛНЗ колони $\implies$ r(colms A) <= (i - 1) $\implies$ r(A) <= (i - 1) $\implies$ r(rows A) <= (i - 1) $\implies$ A има най-много (i - 1) ЛНЗ редове $\implies$ редовете на A са ЛЗ $\implies$ редовете на L са ЛЗ $\implies$ L не е обратима.

⇒/ □

Lemma 6. Ако долно (горно) триъгълна матрица е обратима, то обратната матрица е долно (горно) триъгълна и всеки елемент по главния диагонал на L^{-1} е равен на реципрочното на съответния елемент в L.

Доказателство. Няма да се спираме на строгото доказателство на тази Лема, но интуитивно се вижда, че това е вярно, ако си представим алгоритъма за намиране на обратна матрица, чрез метода на Гаус и единичната матрица. □

2 Оптимална форма на LU Decomposition

Това, което видяхме дотук е, че ако работим с тази дефиниция на LU Decomposition, то нито имаме гарантирано съществуване, нито имаме гарантирана еднозначност. Това разбира се е голям проблем и причинява големи неудобства. Със следващите твърдения ще формулираме и докажем метод, който да ни гарантира точно тези две важни свойства. Първо обаче ще въведем една помощна дефиниция и една Лема.

Definition 2.1. Нека $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ е биекция, която ще интерпретираме като изображение, задаващо пермутация на числата от 1 до n. Тогава нека дефинираме

$$P_\pi(p_{ij})_{nxn} : p_{ij} = \begin{cases} 1, & \text{if } j = \pi(i) \\ 0, & \text{otherwise} \end{cases}$$

матрица на пермутацията π.

Lemma 7. Нека P е марица на пермутация, тогава P е обратима и $P^{-1} = P^T$, тоест е ортогонална.

Доказателство. Че е обратима е очевидно (има ЛНЗ редове). Тогава $PP^{-1} = E$, където E е единичната матрица. Трябва да докажем, че $PP^T = E$. Имаме, че

$$(PP^T)_{ij} = \sum_{k=1}^{n} P_{ik} * P_{jk}.$$

Тогава тъй като на всеки ред и на всеки стълб от P има точно по една единица, действително ако i = j :

$$\sum_{k=1}^{n} P_{ik} * P_{jk} = \sum_{k=1}^{n} P_{ik}^2 = 1$$

и ако $i \neq j$:

$$\sum_{k=1}^{n} P_{ik} * P_{jk} = 0$$

$\implies$ доказахме твърдението. □

Използвайки тази дефиниция, ще намерим решение на проблема, свързан със съществуването на LU Декомпозиция за всяка квадратна матрица.

Theorem 8. *Нека $A \in M_{nxn}$. Тогава $\exists P_\pi \in M_{nxn}$, матрица на пермутация такава, че PA има LU Декомпозиция.*

$$PA = LU$$

Доказателство. Прилагайки метода на Гаус, можем да приведем матрицата A до U (горно триъгълна) и нека операциите, които сме извършили бъдат изразени с умножение отляво на матрицата A със следните матрици : $E_n \dots E_1 A = U$. От друга страна преобразованията, които сме извършили могат да бъдат или от вида разместване на два реда, или от вида умножение на ред с число и добавяне към друг по-долен ред. Нека матриците, използвани за преобразованията от първия вид, бележим с P_i, а от втория L_i. Лесно се проверява, че всички матрици L_i са долно-триъгълни и имат ненулеви елементи по диагонала $\implies$ са обратими.Нека сега първата матрица от вида P_i се среща на позиция j. Тогава

$$E_n \dots E_{j+1} P_j L j - 1 \dots L_1 A = U.$$

Тъй като от предходната Лема, всяка матрица от вида P_i е ортогонална $\implies P_j^T P_j = E$. Тогава

$$E_n \dots E_{j+1} P_j L_{j-1} P_j^T P_j \dots P_j^T P_j L_1 P_j^T P_j A = U.$$

Нека $K_i = P_j L_i P_j^T$ за $i \in \{1 \dots j-1\}$. От друга страна матриците от вида L_i могат да бъдат представени във вида $L_i = E + M$, където M има само един ненулев елемент, който в нашия случай е под главния диагонал (от естеството на преобразованията, които сме извършили). Тогава

$$K_i = P_j L_i P_j^T = P_j (E + M) P_j^T = P_j P_j^T + P_j M P_j^T = E + P_j M P_j^T.$$

Съобразявайки размишленията дотук, сравнително лесно се проверява, че K_i е долно-триъдълна за $i \in \{1 \dots j-1\}$. Освен това са и обратими, защото са произведения на обратими матрици. Прилагайки същата идея всеки път, когато някоя от матриците $E_n \dots E_{j+1}$ е матрица на пермутация, в следното уравнение :

$$E_n \dots E_{j+1} K_{j-1} \dots K_1 P_j A = U$$

Получаваме следното преобразование за изходното уравнение

$$(\prod_{i \in Id_1} K_i)(\prod_{i \in Id_2} P_i) A = U$$

Където Id_1 и Id_2 са съответно множествата от индекси на матрици от вида K_j и матрици на пермутации. Така получхме

$$PA = LU$$

където

$$P = \prod_{i \in Id_1} P_i \text{ и } L = (\prod_{i \in Id_2} K_i)^{-1}$$

По този начин доказахме твърдението, тъй като P е матрица на пермутация (защото е произведение на матрици на пермутации), а L е долно-триъгълна (защото от доказаните по-горе леми, обратната матрица на произведението на долно-триъгълни матрици е долно-триъгълна матрица). □

На много места тази форма на LU Декомпозиция се среща като LUP Декомпозиция, но тук ще продължа да я наричам просто LU Декомпозиция. За да получим завършена и удобна за работа форма на LU Декомпозиция, остава да намерим начим и за уникалност. Ще видим как става това чрез формулирането и доказването на следващото твърдение.

Theorem 9. *Нека $A \in M_{nxn}$ е обратима и $P \in M_{nxn}$ матрица на пермутация такава, че PA има LU Декомпозиция. Тогава съществуват единствени долно-триъгълна матрица L с единици по главния диагонал и горно-триъгълна матрица U такива, че*

$$PA = LU$$

Доказателство. Нека $M = PA$, тогава нека допуснем, че съществуват две двойки матрици (L_1, U_1) и (L_2, U_2) такива, че удовлетворяват условието в твърдението. Тогава

$$L_1 U_1 = M = L_2 U_2$$

Нека за пригленост

$$M = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} L_1 = \begin{bmatrix} 1 & \cdots & 0 \\ l'_{21} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l'_{n1} & \cdots & 1 \end{bmatrix} L_2 = \begin{bmatrix} 1 & \cdots & 0 \\ l''_{21} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l''_{n1} & \cdots & 1 \end{bmatrix}$$

$$U_1 = \begin{bmatrix} u'_{11} & \cdots & u'_{1n} \\ 0 & \cdots & u'_{2n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & u'_{nn} \end{bmatrix} U_2 = \begin{bmatrix} u''_{11} & \cdots & u''_{1n} \\ 0 & \cdots & u''_{2n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & u''_{nn} \end{bmatrix}$$

Оттук реализирайки умножението на L_1, U_1 и умножението на L_2, U_2 и имайи предвид, че и двете дават резултат M, получаваме система линейни уравнения, в които има равен брой неизвестни и уравнения $\implies$ има единствено решение $\implies$ $L_1 = L_2$ и $U_1 = U_2$ $\implies$ доказахме твърдението. □

С формулирането и доказването на последните две твърдения получихме удобна за работа LU Декомпозиция. Важно е да се отбележи, че има и други разновидности на LU Декомпозиция, които не се различават съществено от това, което показахме тук. Доказателствата на тези твърдения

ни дават и добра насока за това, как да намираме така дефинираната LU Декомпозиция за конкретна матрица. Тук можете да видите бърз пример за това, как се намира LU Декопозицията с матрица на пермутация за конкретна 3x3 матрица.

3 Приложения на LU Декомпозицията

На пръв поглед LU Декомпозицията изглежда доста тромава и дори безсмислена техника. В нея обаче се крие нещо, което придава смисъл на цялата тази идея и ни дава някои приложения. Това, което прави метода смислен, е факта, че намирайки LU Декомпозиция на една матрица ние правим едно своеобразно кодиране на метода на Гаус за дадената матрица. Най-лесно ще можем да обясним това с няколко примера за това.

3.1 Решаване на системи линейни уравнения

Нека имаме стандартно матрично уравнение, на което да търсим решение. Тогава ако имаме LU Декомпозиция от вида PA = LU имаме

$$AX = B \implies LUX = PB$$

Тогава първо решаваме

$$LY = PB, \text{ където } Y = UX$$

после решаваме

$$UX = Y$$

Много е важно да се отбележи, че тук намирането на PA = LU не зависи от матрцата B. Освен това L и U са матрици в триъгълен вид, което прави пресмятането на Y и X много бързо. Това ни дава и предимство при решаване на линейни уравнения по този начин, тъй като ако веднъж сме пресметнали PA = LU, можем много бързо да решаваме уравнения от този вид за различни матрици B. Именно тук идеята, че "кодираме"метода на Гаус, придобива смисъл.

3.2 Намиране на обратна матрица

Директно се вижда ако резултатната матрица по-горе е единичната матрица, тоест ако PA = LU и LUX = PE. Тогава решавайки това уравнение (което при наличие на PA = LU видяхме, че е много бързо), получаваме $X = A^{-1}$.

3.3 Намиране на детерминанта

Нека имаме PA = LU $\Leftrightarrow A = P^{-1}LU \implies$ det(A) = det(P^{-1}) det(L) det(U), където отново трябва да имаме предвид, че детерминантите на P^{-1}, L и U, се намират много бързо.

Всички тези примери ни дават основание да направим следното заключение. Ако трябва еднократно или да намерим детерминатна, или да намерим обратна матрица, или да решим система линейни уравнения, може би LU Декомпозицията не е най-практичния избор. Ако обаче многократно ще извършваме операции върху една и съща матрица, този метод ни дава начин, на цената на малко повече еднократна работа в началото да оптимизираме операциите след това.

4 Алгоритъм за криптиране и декриптиране на текст, чрез LU Декомпозиция

Алгоритъма, който ще предтавим се състои от три основни стъпки - подготовка на изходния текст за алгоритъма, алгоритъм за криптиране и алгоритъм за декриптиране.

4.1 Подготовка на изхдния текст

Нека S е множеството от всички значими символи (такива, които имат значение в контекста на едно съобщение). Нека $input = a_1 \dots a_k$ е входния текст, където a_i са някакви символи (без значение дали $a_i \in S$) за $i = 1 \dots k$. Нека ' * ' е символа, който ще съпоставяме на недефинираните символи и нека $\Sigma = S \cup \{*\}$, $N = \{0, \dots, |\Sigma|-1\}$. Тогава нека дефинираме $encode : \Sigma \to N$, биекция. Нека $A = (a_{ij})_{mxm}$ е обратима матрица с положителни елементи, ненадвишаващи $|\Sigma - 1|$, с която ще работи нашия алгоритъм, може да се разгледа и като матрица на линейно изображение (може да бъде генерирана от ключ, който се задава чрез текст или по друг начин). Въз основа на това ще дефинираме

$$C = (c_{ij})_{mxn} : c_{ij} = \begin{cases} \text{encode}(a_{(i-1)n+j}), & \text{if } (a_{(i-1)n+j} \in S) \wedge ((i-1)n + j <= k) \\ \text{encode}(*), & \text{otherwise} \end{cases}$$

матрица на изходния текст в числов вид. Ясно е, че така дефинирана, обработката на текста има лимит за изходен текст с дължина mxn символа. Лесно се вижда, че винаги можем да избираме n въз основа на дължината на текста. Въпрос на решение относно имплементация, каква е оптималната стойност за m, съобразявайки памет и оптимално действие.

4.2 Алгоритъм за криптиране

Имаме C - матрица на трансформирания тескт (mxn), A - константна матрица, инициализирана преди стартиране на алгоритъма. Сега изпълняваме следните стъпки

4.2.1 Генерираме матрица $B = (b_{ij})_{mxn}$, решавайки следното матрично уравнение $B = AC$ (mod p), където $p = |\Sigma|$

4.2.2 Генерираме ключ за криптиране L_{mxm} и матрица на пермутация P_{mxm} по следния начин $PA = LU$, използвайки метода на LU Декомпозиция

4.2.3 Получаваме матрица на криптираното съобщение E_{mxn} (encrypted) по следния начин

$$E = L^{-1}PB$$
$$E = E(\text{mod p})$$

4.2.4 Генерираме криптирано съобщение, използвайки $encode^{-1}$ върху елементите на матрицата E

4.3 Алгоритъм за декриптиране

4.3.1 Възстановяваме матрицата на криптирания текст - E_{mxn}, използвайки $encode$ върху символите от криптирания текст

4.3.2 Генерираме ключ за декриптиране U_{mxm} по следния начин $PA = LU$, използвайки метода на LU Декомпозиция

4.3.3 Получаваме матрица на декриптирането D_{mxn} (decrypt) по следния начин

$$D = U^{-1}E$$
$$D = D(\text{mod p})$$

4.3.4 Възстановяваме изходния текст, прилагайки $encode^{-1}$ върху елементите на D, обхождайки ги по редове

Последното нещо, което трябва да докажем е, че алгоритъма е коректен. По точно ще покажем, че $D = C$. Това в случая се вижда от следната серия от равенства

$$\begin{aligned} D &= U^{-1}E \\ &= U^{-1}L^{-1}PB \\ &= U^{-1}L^{-1}PAC \\ &= U^{-1}L^{-1}LUC \\ &= C \end{aligned}$$

Което наистина показва, че след криптиране и декриптиране наистина получаваме изходния текст.

5 Използвани източници

Основна конструкция на теорията :
$https://en.wikipedia.org/wiki/LU_decomposition$
За конкретни проблеми, възникнали по време на реализация :
$https://math.stackexchange.com/$
Допълнителни източници:
$https://www.math.purdue.edu/\ arapura/preprints/gaussian.pdf$
$https://www.umbc.edu/photonics/Menyuk/ENEE605/menyuk_ENEE605_lecture4_130908n.pdf$
$https://www.geeksforgeeks.org/l-u-decomposition-system-linear-equations/$
$https://www.statlect.com/matrix-algebra/triangular-matrix$