Метод Жордана нахождения обратной матрицы с выбором главного элемента по строке

Попов Илья

14 февраля 2025 г.

1 Блочный вариант нахождения обратной матрицы методом Жордана с поиском главного элемента по строке, параллельный

Постановка задачи

Находим матрицу обратную к данной

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

Пусть m - размер блока, тогда поделим n - размер матрицы на m с остатком $n = m * k + l$ тогда матрицу можно представить в виде:

$$A = \begin{pmatrix} A_{11}^{m\times m} & A_{12}^{m\times m} & \cdots & A_{1,k}^{m\times m} & A_{1,k+1}^{m\times l} \\ A_{21}^{m\times m} & A_{22}^{m\times m} & \cdots & A_{2,k}^{m\times m} & A_{2,k+1}^{m\times l} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_{k,1}^{m\times m} & A_{k,2}^{m\times m} & \cdots & A_{k,k}^{m\times m} & A_{k,k+1}^{m\times l} \\ A_{k+1,1}^{l\times m} & A_{k+1,2}^{l\times m} & \cdots & A_{k+1,k}^{l\times m} & A_{k+1,k+1}^{l\times l} \end{pmatrix}$$

1.1 Разеделение данных на свои и чужие

Каждый поток получает строки матрицы с номерами $num + p * counter$, где num это номер потока, p это общее число потоков и $num + p * counter <= (k+1)$, $counter$ это натуральные числа и 0
В присоеденненной матрице разделение на потоки происходит в точности таким же образом

1.2 Формулы на поток

Шаг с номером h, где $h <= k$:

1. Поток с номером num находит обратные матрицы среди матриц строки h, а именно среди матриц $A_{h,g}, g = num + p * counter <= k, g >= h$. Для обратных блоков считаются нормы и и выбирается блок с наименьшей нормой обратного

2. Первая точка синхронизации ($reduce\_sum$): потоки находят блоки с наименьшей нормой среди наименьших, но ответ уже для всей строки. Пусть наименьший блок $A_{h,j}$, если во всех потоках не удалось найти обратный, то данный алгоритм не применим

3. Каждый поток в своих строчках меняет местими столбцы с номерами h и j

4. Вторая точка синхронизации (*barrier*)

5. Все потоки находят обратнцю матрицу к блоку $A_{h,h}$

6. Поток с номером num домножает слева на $A_{h,h}^{-1}$ блоки $A_{h,j}, j = num + p * counter <= (k+1), j >= h$ и $B_{h,i}, i = num + p * counter <= h, i >= 1$,где блоки B это блоки присоеденненной матрицы

7. Третья точка синхронизации (*barrier*)

8. Поток с номером num:
   В матрицах A и B вычитаем из строк с номером $i! = h$ строку с номером h домноженную слева на $A_{i,h}$ А именно:
$$A_{i,j} = A_{i,j} - A_{i,q} * A_{q,j}, \quad i = num + p * counter <= (k+1), i >= 1, i! = h \quad j = q+1, .., k+1$$
$$B_{i,j} = B_{i,j} - A_{i,q} * B_{q,j}, \quad i = num + p * counter <= (k+1), i >= 1, i! = h \quad j = 1, ..., h$$

9. Четвертая точка синхронизации (*barrier*) перед следеющим шагом алгоритма

На шаге с номером $k+1$ ищем обратный блок к блоку $A_{k+1,k+1}$ вовсех потоках, если этот блок не обратим, то алгоритм не применим для данного m. В потоке с номером num только в матрице B выполняем следующие:
$$B_{h,i}, i = num + p * counter <= (k+1), i >= 1$$
Точка синхронизации (*barrier*)
$$B_{i,j} = B_{i,j} - A_{i,q} * B_{q,j}, \quad i = num + p * counter <= (k+1), i >= 1, i! = k+1, \quad j = 1, ..., k+1$$
Точка синхронизации (*barrier*)

Так как в матрице A мы меняли мастами стобцы для нахождения главного элемента, то в матрице присоеденённой B мы меняем мастими соответствующие строки.

1.3 Формула сложности

Для рассчета формулы сложности считаем, что $n\%m == 0, \frac{n}{m}\%p == 0$
$n = q * m$, p - число потоков

1.3.1 Для исходной матрицы

$$\sum_{k=1}^{q} \lceil \frac{k}{p} \rceil * (2 * m^3 - \frac{m^2}{2} - \frac{m}{2}) + \sum_{k=1}^{q-1} \lceil \frac{q-k}{p} \rceil * (2 * m^3 - m^2) + \sum_{k=1}^{q-1} (q-k) * 2 * m^3 * (\lceil \frac{q-1}{p} \rceil)$$

Так как $\sum\limits_{i=1}^{q} \lceil \frac{i}{p} \rceil = \sum\limits_{i=1}^{p} 1 + \sum\limits_{i=p+1}^{2*p} 2 + ... = p * \sum\limits_{i=1}^{\frac{n}{m*p}} 1 = \frac{q^2}{2*p} + \frac{q}{2}$
То получаем, что
$$(2 * m^3 - \frac{m^2}{2} - \frac{m}{2}) * (\frac{q^2}{2*p} + \frac{q}{2}) + (2 * m^3 - m^2) * (\frac{q^2}{2*p} - \frac{q}{2}) + m^3 * (q-1)^2 * \frac{q}{p} =$$
$$= \frac{n^3}{p} + \frac{n * m^2}{p} + O(n^2 + n * m + m^2)$$

1.3.2 Для присоеденнёной

$$\sum_{k=1}^{q}(\lceil\frac{k-1}{p}\rceil)*(2*m^3-m^2)+\lceil\frac{q-1}{p}\rceil*\sum_{k=1}^{q}k*2*m^3=$$

$$=(2*m^3-m^2)*(\frac{q^2}{2*p}-\frac{q}{2})+2*m^3*(q^2-1)*\frac{q}{p}=\frac{n^3}{p}+\frac{n^2*m}{p}-2*\frac{n*m2}{p}+O(n^2+n*m+m^2)$$

Суммируя получаем итогувую сложность :

$$S_p(n,m,p)=2*\frac{n^3}{p}+\frac{n^2*m}{p}-\frac{n*m^2}{p}+O(n^2+n*m+m^2)$$

$$S_p(n,m,1)=S(n,m)$$

1.4 Оценка числа точек синхронизации

$$4*(\frac{n}{m})+2$$

2 Метод Жордана нахождения обратной матрицы с выбором главного элемента по строке

Постановка задачи

Находим матрицу обратную к данной

$$A=\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n}\\ a_{21} & a_{22} & \dots & a_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ a_{n1} & a_{n2} & \dots & a_{nn}\end{pmatrix}$$

Пусть m - размер блока, тогда поделим n - размер матрицы на m с остатком $n=m*k+l$ тогда матрицу можно представить в виде:

$$A=\begin{pmatrix} A_{11}^{m\times m} & A_{12}^{m\times m} & \cdots & A_{1,k}^{m\times m} & A_{1,k+1}^{m\times l}\\ A_{21}^{m\times m} & A_{22}^{m\times m} & \cdots & A_{2,k}^{m\times m} & A_{2,k+1}^{m\times l}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ A_{k,1}^{m\times m} & A_{k,2}^{m\times m} & \cdots & A_{k,k}^{m\times m} & A_{k,k+1}^{m\times l}\\ A_{k+1,1}^{l\times m} & A_{k+1,2}^{l\times m} & \cdots & A_{k+1,k}^{l\times m} & A_{k+1,k+1}^{l\times l}\end{pmatrix}$$

2.1 Хранение матрицы в памяти

Матрицу будем хранить в памяти следующим образом

$$A=\left(\begin{array}{cccc|cccc|c|ccc|cccc|c} a_{11} & a_{12} & \dots & a_{1,m} & a_{21} & a_{22} & \dots & a_{2,m} & \dots & a_{m,1} & \dots & a_{m,m} & a_{1,m+1} & a_{1,m+2} & \dots & a_{1,m+m} & \dots\end{array}\right)$$

То есть хранение блочное

$$A=\begin{pmatrix} A_{11}^{m\times m} & A_{12}^{m\times m} & \cdots & A_{1,k}^{m\times m} & A_{1,k+1}^{m\times l} & \cdots & A_{k+1,1}^{l\times m} & \cdots & A_{k+1,k}^{l\times m} & A_{k+1,k+1}^{l\times l}\end{pmatrix}$$

2.2 Описание функций getBlock и setBlock

Функции getBlock и setBlock возвращают указатель на начало блока. Адресс блока с номером (i, j) это

$$A + (i-1) * (m*m*k + m*l) + (j-1)*m*m$$

где A - это адресс начала матрицы. Если же $i = k+1$, то адресс будет равняться

$$A + k*(m*m*k + m*l) + (j-1)*l*m$$

2.3 Описание формул

Пусть B - присоеденённая матрица, которая тоже хранится блоками.
Обратная матрица для блоков находится с помощью обычного метода Жордана с выбором главного элемента по строке.
Норма матрицы : $\|A^{m\times m}\| = \max\limits_{i=1,\ldots,m} \sum\limits_{j=1}^{m} |a_{ij}|$
q - шаг меняется от 1 до k включительно

1. На q-ом шаге алгоритма среди блоков $A_{q,j}$ при $j = q,\ldots,k$ выбираем блок, у которой норма обратного наименьшая. Для этого данный блок обращается и у него считается норма, если блок не обратим, то пропускаем его. Если все блоки среди $A_{q,j}$ при $j = q,\ldots,k$ необратимы то данный алгоритм не применим. Пусть у блока $A_{q,g}$ норма обратной наименшая, тога меняем местами столбцы (блочные) с номерами q и g. Матрицу B на данном шаге не трогаем. Запоминаем номерая столбцов которые поменяли местами.

2. Умножаем все блоки $A_{q,j}$ при $j = q+1,\ldots,k,k+1$ слева на к $A_{q,q}^{-1}$. Блок $A_{q,q}$ на $A_{q,q}^{-1}$ не домножаем, просто записываем еденичный блок (этого можно не делать так как обращений к этому блоку больше не будет) В матрице B домножаем слева на $A_{q,q}^{-1}$ блоки $B_{q,j}$ при $j = 1,..,q-1$. На место блока $B_{q,q}$ сразу запишем $A_{q,q}^{-1}$ так ка изначално он еденичный.

$$A_{q,j} = A_{q,q}^{-1} * A_{q,j}, \quad j = q+1,\ldots,k,k+1$$

$$B_{q,j} = A_{q,q}^{-1} * B_{q,j}, \quad j = 1,\ldots,q-1$$

3. В матрицах A и B вычитаем из строк с номером $i! = q$ строку с номером q домноженную слева на $A_{i,q}$ А именно:

$$A_{i,j} = A_{i,j} - A_{i,q} * A_{q,j}, \quad i = 1,\ldots,q-1,q+1,\ldots,k+1, \quad j = q+1,..,k+1$$

$$B_{i,j} = B_{i,j} - A_{i,q} * B_{q,j}, \quad i = 1,\ldots,q-1,q+1,\ldots,k+1, \quad j = 1,\ldots,q$$

На шаге с номером $k+1$ ищем обратный блок к блоку $A_{k+1,k+1}$ если этот блок не обратим, то алгоритм не применим для данного m. Только в матрице B выполняем следующие:

$$B_{k+1,k+1} = A_{k+1,k+1}^{-1}$$

$$B_{i,j} = B_{i,j} - A_{i,q} * B_{q,j}, \quad i = 1,\ldots,k, \quad j = 1,\ldots,k+1$$

Так как в матрице A мы меняли мастами стобцы для нахождения главного элемента, то в матрице присоеденённой B мы меняем мастими соответствующие строки.

2.4 Оценка числа операций (вариант 1)

Умножение двух матриц $A^{n\times m} * A^{m\times k} : n*k(m+m-1) = 2*n*k*m - m*k$
Сложение или вычитание двух матриц $A^{n\times m} + A^{n\times m} : n*m$
Нахождение обратной матрицы $A^{m\times m}$ обычным методом Жордана с посиком главного элемена по строке $2*m^3 + \frac{11}{2}*m^2 - \frac{13}{2}m$
Расчеты для не блочного метода:
Для исходной матрицы:

$$\sum_{k=1}^{m-1}(m-k)+2*(m-1)*\sum_{k=1}^{m-1}(m-k)+3*m^2-3*m = m^2-m+m*(m^2-2*m+1)+3*m^2-3*m = m^3+2*m-3*m$$

Для присоеденённой:

$$\sum_{k=1}^{m} k + 2*(m-1)*\sum_{k=1}^{m} k + 3*m^2 - 3*m = m^3 + \frac{7}{2}*m^2 - \frac{7}{2}*m$$

Складывая получаем:

$$2*m^3 + \frac{11}{2}*m^2 - \frac{13}{2}*m$$

Для блочного варианта: (пусть $q = n/m$ и не будем учитывать не поделившиеся края, так как они не влияют на ассимптотику)
Для присоеденённой матрицы:

$$\sum_{k=1}^{q}(2*m^3-m^2)+\sum_{k=1}^{q}(q-1)*k*(2*m^3-m^2)+\sum_{k=1}^{q}(q-1)*k*m^2 =$$

$$= (2*m^3-m^2)*(\frac{q^2}{2}+\frac{q}{2})+(2*m^3-m^2)*(q^2-1)*\frac{1}{2}+m^2*(q^2-1)*\frac{q}{2} = n^3+n^2*m+O(n^2+n*m+m^2)$$

Для исходной матрицы:

$$(2*m^3+\frac{11}{2}*m^2-\frac{13}{2}*m)*\frac{1+q}{2}*q+\sum_{k=1}^{q-1}(q-k)*(2*m^3-m^2)*(q-1)+\sum_{k=1}^{q-1}(q-1)(q-k)*m^2 =$$

$$= (2*m^3+\frac{11}{2}*m^2-\frac{13}{2}*m)*(\frac{q^2}{2}+\frac{q}{2})+(2*m^3-m^2)*(\frac{q^2}{2}-\frac{q}{2})+2*m^3*(q-1)^2*\frac{1}{2} =$$
$$= n^3+m^2*n+O(n^2+n*m+m^2)$$

Складывая, получаем:

$$2*n^3+m^2*n+n^2*m+O(n^2+n*m+m^2)$$

2.5 Проверка формулы (вариант 1)

$S(n,1) = 2*n^3 + O(n^2)$
$S(n,n) = 4*n^3 + O(n^2)$

2.6 Оценка числа операций (Вариант 2)

$q = \frac{n}{m}$

2.6.1 Для исходной матрицы:

Сложность нахождения обратных блоков для поиска ведущего среди $A_{k,j}$ при $j=k,...,q$:

$$(2*m^3-\frac{m^2}{2}-\frac{m}{2})*\frac{1+q}{2}*q$$

Сложность умножений на блок, обратный к ведущему $A_{k,j}=A_{k,k}^{-1}*A_{k,j},\quad j=k+1,...,q$:

$$\sum_{k=1}^{q-1}(q-k)*(2*m^3-m^2)$$

Сложность умножений и вычитаний для $A_{i,j}=A_{i,j}-A_{i,k}*A_{k,j},\quad i=1,...,k-1,k+1,...,q,\quad j=k+1,..,q$:

$$\sum_{k=1}^{q-1}(q-k)*2*m^3*(q-1)$$

Тогда, суммируя, получим:

$$2*m^3*(\frac{q^2}{2}+\frac{q}{2})+(2*m^3-m^2)*(\frac{q^2}{2}-\frac{q}{2})+m^3(q-1)^2*q=$$

$$=\frac{1}{2}*(2*m^3)*(\frac{n^2}{m^2}+\frac{n}{m})+\frac{1}{2}*(2*m^3-m^2)*(\frac{n^2}{m^2}-\frac{n}{m})+m^3*(\frac{n^3}{m^3}-2*\frac{n^2}{m^2}+\frac{n}{m})=$$
$$=n^3+n*m^2+O(n^2+m*m+m^2)$$

2.6.2 Для присоеденнёной матрицы:

Сложность умножений на блок, обратный к ведущему $B_{k,j}=A_{k,k}^{-1}*B_{k,j},\quad j=1,...,k-1$:

$$\sum_{k=1}^{q}(k-1)*(2*m^3-m^2)$$

Сложность умножений и вычитаний для $B_{i,j}=B_{i,j}-A_{i,k}*B_{k,j},\quad i=1,...,k-1,k+1,...,q,\quad j=1,...,k$:

$$(q-1)*\sum_{k=1}^{q}*k*2*m^3$$

2.6.3 Тогда, суммируя, получим:

$$(2*m^3-m^2)*(\frac{q^2}{q}-\frac{q}{2})+m^3*(q^3-q)=\frac{1}{2}*(2*n^2*m-2*n*m2)=n^3-n*m^2=$$
$$=n^2*m+n^3-m^2*n+O(n^2+m*m+m^2)$$

2.6.4 Итог:

$$S(n,m)=2*n^3+n^2*m-n*m^2+O(n^2+m*m+m^2)$$

2.7 Проверка формулы (вариант 2)

$S(n,1)=2*n^3+O(n^2)$ - сложность обычного метода Жордана нахождений обратной матрицы
$S(n,n)=2*n^3+O(n^2)$ - сложность обычного метода Жордана нахождения обратной матрицы, но нет ешё одного умножения, так как в присоеденённой матрице единичный блок на $B_{k,k}$ на шаге k на $A_{k,k}^{-1}$ не умножается, а просто заменяется (если все таки умножать, то получится вариант 1, то есть при $m=n$ будет сложность = Жордан + одно умножений = $4*n^3$)