Вычислительная математика. Прямые методы решения систем линейных уравнений.

МФТИ

Содержание

1. LU-разложение и метод Гаусса
2. Метод Холецкого
3. QR-разложение
4. Метод наименьших квадратов

Метод Гаусса

Метод Гаусса

- Исключение неизвестных:

$$+\left\{\begin{array}{l} \times -\frac{a_{21}}{a_{11}} \\ \\ \end{array}\right. \quad \left[\begin{array}{ccc} a_{11} & a_{12} & \dots \\ \hline a_{21} & a_{22} & \dots \end{array}\right]$$

Метод Гаусса

- Исключение неизвестных:

$$+\begin{cases} \times -\frac{a_{21}}{a_{11}} \\ \end{cases} \quad \left[\begin{array}{ccc} a_{11} & a_{12} & \dots \\ \hline a_{21} & a_{22} & \dots \end{array}\right]$$

- Один шаг исключения – умножение слева на нижнетреугольную матрицу L

$$A_1 = \underbrace{\begin{bmatrix} 1 & 0 & \dots & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & 0 & \dots \\ & & \ddots & \end{bmatrix}}_{L} \times A$$

LU- разложение

- Основная идея: с помощью операций со строками привести матрицу к верхнетреугольному виду:

LU- разложение

- Основная идея: с помощью операций со строками привести матрицу к верхнетреугольному виду:

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}}_{A} \xrightarrow{L_1} \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}_{L_1 A} \xrightarrow{L_2}$$

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}_{L_2 L_1 A} \xrightarrow{L_3} \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & \mathbf{0} & \boldsymbol{\times} \end{bmatrix}}_{L_3 L_2 L_1 A}$$

LU- разложение

- Основная идея: с помощью операций со строками привести матрицу к верхнетреугольному виду:

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}}_{A} \xrightarrow{L_1} \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}_{L_1 A} \xrightarrow{L_2}$$

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}_{L_2 L_1 A} \xrightarrow{L_3} \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & \mathbf{0} & \boldsymbol{\times} \end{bmatrix}}_{L_3 L_2 L_1 A}$$

- $\underbrace{L_{n-1} \dots L_2 L_1}_{L^{-1}} A = U,\ L = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1},\ A = LU$

LU- разложение

- Основная идея: с помощью операций со строками привести матрицу к верхнетреугольному виду:

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}} \xrightarrow{L_1} \underset{L_1 A}{\begin{bmatrix} \times & \times & \times & \times \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}} \xrightarrow{L_2}$$

$$\underset{L_2 L_1 A}{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}} \xrightarrow{L_3} \underset{L_3 L_2 L_1 A}{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & \mathbf{0} & \boldsymbol{\times} \end{bmatrix}}$$

- $\underbrace{L_{n-1} \dots L_2 L_1}_{L^{-1}} A = U$, $L = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1}$, $A = LU$
- Могли столкнуться с делением на 0!

Треугольные матрицы

Треугольные матрицы

- $L_1 L_2$ – нижнетреугольная матрица.

Треугольные матрицы

- $L_1 L_2$ – нижнетреугольная матрица.

$$(L_1 L_2)_{ij} = \sum_{k=1}^{n} l^1_{ik} l^2_{k_j} = \sum_{k \le i, k \ge j} l^1_{ik} l^2_{k_j} \overset{i<j}{=} 0$$

Треугольные матрицы

- $L_1 L_2$ – нижнетреугольная матрица.

$$(L_1 L_2)_{ij} = \sum_{k=1}^{n} l^1_{ik} l^2_{k_j} = \sum_{k \le i, k \ge j} l^1_{ik} l^2_{k_j} \overset{i<j}{=} 0$$

- L^{-1} – нижнетреугольная матрица.

Треугольные матрицы

- $L_1 L_2$ – нижнетреугольная матрица.

$$(L_1 L_2)_{ij} = \sum_{k=1}^{n} l^1_{ik} l^2_{k_j} = \sum_{k \le i, k \ge j} l^1_{ik} l^2_{k_j} \stackrel{i<j}{=} 0$$

- L^{-1} – нижнетреугольная матрица.
  По правилу Крамера: $(L^{-1})_{ij} = \frac{\Delta_{ij}}{\Delta}$

Треугольные матрицы

- $L_1 L_2$ – нижнетреугольная матрица.

$$(L_1 L_2)_{ij} = \sum_{k=1}^{n} l^1_{ik} l^2_{k_j} = \sum_{k \le i, k \ge j} l^1_{ik} l^2_{k_j} \overset{i<j}{=} 0$$

- L^{-1} – нижнетреугольная матрица.
По правилу Крамера: $(L^{-1})_{ij} = \frac{\Delta_{ij}}{\Delta}$

$$i < j: \ \Delta_{ij} = \begin{vmatrix} x & & & \\ x & x & & \\ x & x & 0 & \\ x & x & 1 & x \end{vmatrix} = 0$$

Матрицы L_k

Матрицы L_k

$$x_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ x_{k+1,k} \\ \vdots \\ x_{nk} \end{bmatrix},$$

Матрицы L_k

$$x_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ x_{k+1,k} \\ \vdots \\ x_{nk} \end{bmatrix}, L_k x_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Матрицы L_k

$$x_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ x_{k+1,k} \\ \vdots \\ x_{nk} \end{bmatrix}, L_k x_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

нужно вычесть из j-й строки k-ю строку, умноженную на

$$l_{jk} = \frac{x_{jk}}{x_{kk}} \quad k < j \leq n$$

Матрицы L_k

$$x_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ x_{k+1,k} \\ \vdots \\ x_{nk} \end{bmatrix}, L_k x_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

нужно вычесть из j-й строки k-ю строку, умноженную на

$$l_{jk} = \frac{x_{jk}}{x_{kk}} \quad k < j \leq n$$

$$L_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{nk} & & & 1 \end{bmatrix}$$

Матрицы L_k

Матрицы L_k

$$L_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{nk} & & & 1 \end{bmatrix}, \quad l_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \vdots \\ l_{n,k} \end{bmatrix}$$

Матрицы L_k

$$L_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{nk} & & & 1 \end{bmatrix}, \quad l_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \vdots \\ l_{n,k} \end{bmatrix}$$

- $L_k = I - l_k e_k^T$

Матрицы L_k

$$L_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{nk} & & & 1 \end{bmatrix}, \quad l_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \vdots \\ l_{n,k} \end{bmatrix}$$

- $L_k = I - l_k e_k^T$
- $(I - l_k e_k^T)(I + l_k e_k^T) = I - l_k \cancel{(e_k^T l_k)} e_k^T = I$

Матрицы L_k

$$L_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{nk} & & & 1 \end{bmatrix}, \quad l_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \vdots \\ l_{n,k} \end{bmatrix}$$

- $L_k = I - l_k e_k^T$
- $(I - l_k e_k^T)(I + l_k e_k^T) = I - l_k\cancel{(e_k^T l_k)}e_k^T = I$
- $L_k^{-1} = I + l_k e_k^T$

Матрицы L_k

$$L_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{nk} & & & 1 \end{bmatrix}, \quad l_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \vdots \\ l_{n,k} \end{bmatrix}$$

- $L_k = I - l_k e_k^T$
- $(I - l_k e_k^T)(I + l_k e_k^T) = I - l_k \cancel{(e_k^T l_k)} e_k^T = I$
- $L_k^{-1} = I + l_k e_k^T$
- $L_k^{-1} L_{k+1}^{-1} = (I + l_k e_k^T)(I + l_{k+1} e_{k+1}^T) =$
  $I + l_k e_k^T + l_{k+1} e_{k+1}^T + l_k \cancel{e_k^T l_{k+1}} e_{k+1}^T = I + l_k e_k^T + l_{k+1} e_{k+1}^T$

Алгоритм LU-разложения без перестановок

Алгоритм LU-разложения без перестановок

$$L = L_1^{-1} \dots L_{n-1}^{-1} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{bmatrix}$$

Алгоритм LU-разложения без перестановок

$$L = L_1^{-1} \dots L_{n-1}^{-1} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{bmatrix}$$

Алгоритм:

```
U = A
L = I
for k = 1, n-1
    for j = k+1, n
        l[j,k] = u[j,k]/u[k,k]
        u[j,k:n] = u[j,k:n] - l[j,k] * u[k,k:n]
```

Число операций

```
U = A
L = I
for k = 1, n-1
    for j = k+1, n
        l[j,k] = u[j,k]/u[k,k]
        u[j,k:n] = u[j,k:n] - l[j,k] * u[k,k:n]
```

Число операций

```
U = A
L = I
for k = 1, n-1
    for j = k+1, n
        l[j,k] = u[j,k]/u[k,k]
        u[j,k:n] = u[j,k:n] - l[j,k] * u[k,k:n]
```

$$\sum_{k=1}^{n-1} \sum_{j=k+1}^{n} 2(n-k+1) =$$

Число операций

```
U = A
L = I
for k = 1, n-1
    for j = k+1, n
        l[j,k] = u[j,k]/u[k,k]
        u[j,k:n] = u[j,k:n] - l[j,k] * u[k,k:n]
```

$$\sum_{k=1}^{n-1} \sum_{j=k+1}^{n} 2(n-k+1) = \sum_{k=1}^{n-1} 2(n-k+1)(n-k) \overset{q=n-k}{=}$$

Число операций

```
U = A
L = I
for k = 1, n-1
    for j = k+1, n
        l[j,k] = u[j,k]/u[k,k]
        u[j,k:n] = u[j,k:n] - l[j,k] * u[k,k:n]
```

$$\sum_{k=1}^{n-1}\sum_{j=k+1}^{n} 2(n-k+1) = \sum_{k=1}^{n-1} 2(n-k+1)(n-k) \overset{q=n-k}{=}$$

$$\sum_{q=1}^{n-1} 2(q+1)q = 2\sum_{q=1}^{n-1} q^2 + 2\sum_{q=1}^{n-1} q =$$

Число операций

```
U = A
L = I
for k = 1, n-1
    for j = k+1, n
        l[j,k] = u[j,k]/u[k,k]
        u[j,k:n] = u[j,k:n] - l[j,k] * u[k,k:n]
```

$$\sum_{k=1}^{n-1} \sum_{j=k+1}^{n} 2(n-k+1) = \sum_{k=1}^{n-1} 2(n-k+1)(n-k) \overset{q=n-k}{=}$$

$$\sum_{q=1}^{n-1} 2(q+1)q = 2\sum_{q=1}^{n-1} q^2 + 2\sum_{q=1}^{n-1} q =$$

$$= 2\frac{(n-1)n(2n-1)}{6} + O(n^2) = \frac{2}{3}n^3 + O(n^2) = \Theta$$

Программа для вычисления LU-разложения без перестановок (демо)

Существование LU-разложения

Существование LU-разложения

- Алгоритм построен в предположении, что $u_{kk} \neq 0$

Существование LU-разложения

- Алгоритм построен в предположении, что $u_{kk} \neq 0$
- $\begin{bmatrix} 0. & 1. \\ 1. & 1. \end{bmatrix}$

Существование LU-разложения

- Алгоритм построен в предположении, что $u_{kk} \neq 0$
- $\begin{bmatrix} 0. & 1. \\ 1. & 1. \end{bmatrix}$

Определение 6.1 Строго регулярная матрица

Матрица называется *строго регулярной*, если все её ведущие подматрицы невырожденные.

Существование LU-разложения

- Алгоритм построен в предположении, что $u_{kk} \neq 0$
- $\begin{bmatrix} 0. & 1. \\ 1. & 1. \end{bmatrix}$

Определение 6.1 Строго регулярная матрица

Матрица называется *строго регулярной*, если все её ведущие подматрицы невырожденные.

Теорема 6.1 LU-разложение

A имеет LU-разложение $\iff$ A строго регулярная.

Свойства LU-разложения

Свойства LU-разложения

- LU-разложение определяется однозначно:
  Допустим $L_1 U_1 = L_2 U_2$, тогда
  $L_2^{-1} L_1 = U_2 U_1^{-1} = D, \Rightarrow$
  D - диагональная, $D = I$, т.е. $L_1 = L_2$, $U_1 = U_2$ $\square$

Свойства LU-разложения

- LU-разложение определяется однозначно:
  Допустим $L_1 U_1 = L_2 U_2$, тогда
  $L_2^{-1} L_1 = U_2 U_1^{-1} = D, \Rightarrow$
  D - диагональная, $D = I$, т.е. $L_1 = L_2$, $U_1 = U_2$ $\square$
- Все ведущие миноры положительны $\Leftrightarrow$ все диагональные элементы $U > 0$

$$A = LU = \begin{bmatrix} 1 & 0 & \dots & \\ x & \ddots & & \\ \vdots & & 1 & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} u_{11} & x & \dots & \\ 0 & \ddots & & \\ \vdots & & u_{kk} & \\ & & & \ddots \end{bmatrix}$$

Свойства LU-разложения

- LU-разложение определяется однозначно:
  Допустим $L_1U_1 = L_2U_2$, тогда
  $L_2^{-1}L_1 = U_2U_1^{-1} = D, \Rightarrow$
  D - диагональная, $D = I$, т.е. $L_1 = L_2$, $U_1 = U_2$ $\square$
- Все ведущие миноры положительны $\Leftrightarrow$ все диагональные элементы $U > 0$

$$A = LU = \begin{bmatrix} \boxed{\begin{matrix} 1 & 0 & \dots \\ x & \ddots & \\ \vdots & & 1 \end{matrix}} & \\ & \ddots \end{bmatrix} \begin{bmatrix} \boxed{\begin{matrix} u_{11} & x & \dots \\ 0 & \ddots & \\ \vdots & & u_{kk} \end{matrix}} & \\ & \ddots \end{bmatrix}$$

$$\det A_{1:k,1:k} = \det L_{1:k,1:k} \det U_{1:k,1:k} = \prod_1^k u_{ii}$$

Свойства LU-разложения

Свойства LU-разложения

- Если $A = A^*$, то

$$A = LU = L \begin{bmatrix} u_{11} & & \\ & \ddots & \\ & & u_{nn} \end{bmatrix} \tilde{U} = LD\tilde{U} = \tilde{U}^*(D^*L^*)$$

из единственности LU: $\tilde{U}^* = L$, $A = LDL^*$, $u_{ii} \in \mathbb{R}$

Свойства LU-разложения

- Если $A = A^*$, то

$$A = LU = L \begin{bmatrix} u_{11} & & \\ & \ddots & \\ & & u_{nn} \end{bmatrix} \tilde{U} = LD\tilde{U} = \tilde{U}^*(D^*L^*)$$

из единственности LU: $\tilde{U}^* = L$, $A = LDL^*$, $u_{ii} \in \mathbb{R}$

- Если $A = A^* > 0$, то существует *разложение Холецкого*:

$$A = LDL^* = (LD^{1/2})(D^{1/2}L^*) = CC^*$$

C - нижнетреугольная, $C_{ii} > 0$.

LU-разложение и ошибки округления

LU-разложение и ошибки округления

- Пусть t - длина мантиссы

$$A = \begin{bmatrix} 2^{-t} & 1. \\ 1. & 1. \end{bmatrix}, \tilde{L} = \begin{bmatrix} 1. & 0 \\ 2^t & 1. \end{bmatrix} \tilde{U} = \begin{bmatrix} 2^{-t} & 1. \\ 0 & \cancel{1} - 2^t \end{bmatrix}$$

$$\tilde{L}\tilde{U} = \begin{bmatrix} 2^{-t} & 1. \\ 1. & 0. \end{bmatrix} = A + \begin{bmatrix} 0 & 0 \\ 0 & -1. \end{bmatrix}$$

LU-разложение и ошибки округления

- Пусть t - длина мантиссы

$$A = \begin{bmatrix} 2^{-t} & 1. \\ 1. & 1. \end{bmatrix}, \tilde{L} = \begin{bmatrix} 1. & 0 \\ 2^t & 1. \end{bmatrix} \tilde{U} = \begin{bmatrix} 2^{-t} & 1. \\ 0 & \cancel{1} - 2^t \end{bmatrix}$$

$$\tilde{L}\tilde{U} = \begin{bmatrix} 2^{-t} & 1. \\ 1. & 0. \end{bmatrix} = A + \begin{bmatrix} 0 & 0 \\ 0 & -1. \end{bmatrix}$$

- Малая величина диагонального элемента, на который мы делим, приводит к росту элементов в L и U.

LU-разложение и ошибки округления

- Пусть t - длина мантиссы

$$A = \begin{bmatrix} 2^{-t} & 1. \\ 1. & 1. \end{bmatrix}, \tilde{L} = \begin{bmatrix} 1. & 0 \\ 2^t & 1. \end{bmatrix} \tilde{U} = \begin{bmatrix} 2^{-t} & 1. \\ 0 & \cancel{1} - 2^t \end{bmatrix}$$

$$\tilde{L}\tilde{U} = \begin{bmatrix} 2^{-t} & 1. \\ 1. & 0. \end{bmatrix} = A + \begin{bmatrix} 0 & 0 \\ 0 & -1. \end{bmatrix}$$

- Малая величина диагонального элемента, на который мы делим, приводит к росту элементов в L и U.
- Решение: на каждом шаге выбирать *ведущий элемент* (pivot) из подматрицы: максимальный по модулю элемент в столбце/строке/подматрице.

LU-разложение с выбором ведущего элемента (pivoting)

Выбор по столбцу (partial pivoting):

LU-разложение с выбором ведущего элемента (pivoting)

Выбор по столбцу (partial pivoting):

- выбираем максимальный по модулю элемент в столбце

LU-разложение с выбором ведущего элемента (pivoting)

Выбор по столбцу (partial pivoting):

- выбираем максимальный по модулю элемент в столбце
- меняем его строку с текущей строкой

LU-разложение с выбором ведущего элемента (pivoting)

Выбор по столбцу (partial pivoting):

- выбираем максимальный по модулю элемент в столбце
- меняем его строку с текущей строкой

$$\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & x_{ik} & \boldsymbol{\times} & \boldsymbol{\times} \\ 0 & \times & \times & \times \end{bmatrix} \xrightarrow{P} \begin{bmatrix} \times & \times & \times & \times \\ 0 & x_{ik} & \boldsymbol{\times} & \boldsymbol{\times} \\ 0 & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ 0 & \times & \times & \times \end{bmatrix}$$

$$\xrightarrow{L} \begin{bmatrix} \times & \times & \times & \times \\ 0 & x_{ik} & \times & \times \\ 0 & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ 0 & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}$$

LU-разложение с выбором ведущего элемента (pivoting)

LU-разложение с выбором ведущего элемента (pivoting)

- $L_{n-1}P_{n-1}\cdots L_2P_2L_1P_1A = U$
  P_k - матрица перестановки, $P_k^{-1} = P_k^T$

LU-разложение с выбором ведущего элемента (pivoting)

- $L_{n-1}P_{n-1}\cdots L_2P_2L_1P_1A = U$
  P_k - матрица перестановки, $P_k^{-1} = P_k^T$
-

$$
\begin{aligned}
&L_3P_3L_2P_2L_1P_1 = \\
&L_3(P_3L_2P_3^{-1})(P_3P_2L_1P_2^{-1}P_3^{-1})P_3P_2P_1 = \\
&L_3'L_2'L_1'P_3P_2P_1
\end{aligned}
$$

где $L_3' = L_3,\; L_2' = P_3L_2P_3^{-1},\; L_1' = P_3P_2L_1P_2^{-1}P_3^{-1}$

Действие матриц перестановок

Действие матриц перестановок

$$P_{k+1} L_k P_{k+1}^{-1} =$$

Действие матриц перестановок

$$P_{k+1} L_k P_{k+1}^{-1} = P_{k+1} \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & x & 1 & & \\ & & \vdots & & \ddots & \\ & & x & & & 1 \end{bmatrix} P_{k+1}^{-1} =$$

$$\begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \boldsymbol{x} & 0 & \dots & 1 \\ & & \vdots & & & \\ & & \boldsymbol{x} & 1 & \dots & 0 \end{bmatrix} P_{k+1}^{-1} =$$

Действие матриц перестановок

$$P_{k+1}L_kP_{k+1}^{-1} = P_{k+1}\begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & x & 1 & & \\ & & \vdots & & \ddots & \\ & & x & & & 1 \end{bmatrix} P_{k+1}^{-1} =$$

$$\begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & \boldsymbol{x} & 0 & \dots & 1 \\ & & \vdots & & & \\ & & \boldsymbol{x} & 1 & \dots & 0 \end{bmatrix} P_{k+1}^{-1} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & x & 1 & & \\ & & \vdots & & \ddots & \\ & & x & & & 1 \end{bmatrix}$$

LU-разложение с выбором ведущего элемента

LU-разложение с выбором ведущего элемента

- $L'_{n-1} \dots L'_1 P_{n-1} \dots P_1 A = U \Rightarrow PA = LU$

LU-разложение с выбором ведущего элемента

- $L'_{n-1} \dots L'_1 P_{n-1} \dots P_1 A = U \Rightarrow PA = LU$
- Выбор главного элемента по столбцу добавляет $O(n^2)$ операций – сложность $\approx \frac{2}{3} n^3$

LU-разложение с выбором ведущего элемента

- $L'_{n-1} \dots L'_1 P_{n-1} \dots P_1 A = U \Rightarrow PA = LU$
- Выбор главного элемента по столбцу добавляет $O(n^2)$ операций – сложность $\approx \frac{2}{3} n^3$

```
U = A, L = I, P = I
for k = 1, n-1
    i = argmax(|u[k:n,k]|)
    u[k,k:n] <-> u[i,k:n]
    l[k,1:k-1] <-> l[i,1:k-1]
    p[k,:] <-> p[i,:]
    for j = k + 1, n
        l[j,k] = u[j,k]/u[k,k]
        u[j,k:n] = u[j,k:n] - l[j,k] * u[k,k:n]
```

LU-разложение с выбором ведущего элемента (демо)

Решение системы с помощью LU разложения

Решение системы с помощью LU разложения

Процесс решения системы $Ax = b$ можно разделить на 3 этапа

Решение системы с помощью LU разложения

Процесс решения системы $Ax = b$ можно разделить на 3 этапа

1. Вычисление LU разложения $PA = LU$ (далее $A = LU$) – $O(n^3)$

Решение системы с помощью LU разложения

Процесс решения системы $Ax = b$ можно разделить на 3 этапа

1. Вычисление LU разложения $PA = LU$ (далее $A = LU$) – $O(n^3)$
2. Решение системы $Ly = b$ (прямая подстановка) – $O(n^2)$

Решение системы с помощью LU разложения

Процесс решения системы $Ax = b$ можно разделить на 3 этапа

1. Вычисление LU разложения $PA = LU$ (далее $A = LU$) – $O(n^3)$
2. Решение системы $Ly = b$ (прямая подстановка) – $O(n^2)$
3. Решение системы $Ux = y$ (обратная подстановка) – $O(n^2)$

Решение системы с помощью LU разложения

Процесс решения системы $Ax = b$ можно разделить на 3 этапа

1. Вычисление LU разложения $PA = LU$ (далее $A = LU$) – $O(n^3)$
2. Решение системы $Ly = b$ (прямая подстановка) – $O(n^2)$
3. Решение системы $Ux = y$ (обратная подстановка) – $O(n^2)$

Для разных b достаточно вычислить LU-разложение только 1 раз!

Решение системы с треугольными матрицами

Решение системы с треугольными матрицами

Алгоритм прямой подстановки $Ly = b$:

```
y[:] = 0
for k = 1, n
    y[k] = b[k]
    for i = 1, k-1
        y[k] = y[k] - y[i] * l[k,i]
    y[k] = y[k] / l[k,k]
```

Решение системы с треугольными матрицами

Алгоритм прямой подстановки $Ly = b$:

```
y[:] = 0
for k = 1, n
    y[k] = b[k]
    for i = 1, k-1
        y[k] = y[k] - y[i] * l[k,i]
    y[k] = y[k] / l[k,k]
```

С использованием скалярного произведения:

```
y[:] = 0
for k = 1, n
    y[k] = (b[k] - <y[1:k-1], l[k,1:k-1]> / l[k,k]
```

Решение системы с треугольными матрицами

Алгоритм прямой подстановки $Ly = b$:

```
y[:] = 0
for k = 1, n
    y[k] = b[k]
    for i = 1, k-1
        y[k] = y[k] - y[i] * l[k,i]
    y[k] = y[k] / l[k,k]
```

С использованием скалярного произведения:

```
y[:] = 0
for k = 1, n
    y[k] = (b[k] - <y[1:k-1], l[k,1:k-1]> / l[k,k]
```

Количество операций: $n^2 + O(n)$

Рост элементов в LU-разложении

Рост элементов в LU-разложении

- При выборе в столбце возможен рост элементов U:

$$\frac{\max_{i,j} |u_{ij}|}{\max_{i,j} |a_{ij}|} \leq 2^{n-1}$$

Оценка достигается на специально подобранных матрицах

Рост элементов в LU-разложении

- При выборе в столбце возможен рост элементов U:

$$\frac{\max_{i,j} |u_{ij}|}{\max_{i,j} |a_{ij}|} \le 2^{n-1}$$

  Оценка достигается на специально подобранных матрицах
- На практике сильный рост не встречается!

Рост элементов в LU-разложении

- При выборе в столбце возможен рост элементов U:

$$\frac{\max_{i,j} |u_{ij}|}{\max_{i,j} |a_{ij}|} \leq 2^{n-1}$$

Оценка достигается на специально подобранных матрицах

- На практике сильный рост не встречается!

In fifty years of computing, no matrix problem that excite an explosive instability are known to have arisen under natural circumstances.
Trefethen L. N., Bau III D. Numerical linear algebra. – Siam, 1997

Разложение Холецкого

Разложение Холецкого

- $A = A^T > 0: \ A = CC^T$, C – нижнетреугольная с положительной диагональю.

Разложение Холецкого

- $A = A^T > 0: \; A = CC^T$, C – нижнетреугольная с положительной диагональю.
- Для $n = 3$:

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} c_{11} & 0 & 0 \\ c_{21} & c_{22} & 0 \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} c_{11} & c_{21} & c_{31} \\ 0 & c_{22} & c_{21} \\ 0 & 0 & c_{33} \end{bmatrix}$$

Разложение Холецкого

- $A = A^T > 0: \ A = CC^T$, C – нижнетреугольная с положительной диагональю.
- Для $n = 3$:

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} c_{11} & 0 & 0 \\ c_{21} & c_{22} & 0 \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} c_{11} & c_{21} & c_{31} \\ 0 & c_{22} & c_{21} \\ 0 & 0 & c_{33} \end{bmatrix}$$

- Последовательно будем вычислять элементы C:

$$c_{11} = \sqrt{a_{11}} \qquad c_{21} = \frac{a_{21}}{c_{11}} \qquad c_{31} = \frac{a_{31}}{c_{11}}$$

$$c_{22} = \sqrt{a_{22} - c_{21}^2} \qquad c_{32} = \frac{a_{32} - c_{31}c_{21}}{c_{22}}$$

$$c_{33} = \sqrt{a_{33} - c_{31}^2 - c_{32}^2}$$

Разложение Холецкого

- $A = A^T > 0: \; A = CC^T$, C – нижнетреугольная с положительной диагональю.
- Для $n = 3$:

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} c_{11} & 0 & 0 \\ c_{21} & c_{22} & 0 \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} c_{11} & c_{21} & c_{31} \\ 0 & c_{22} & c_{21} \\ 0 & 0 & c_{33} \end{bmatrix}$$

- Последовательно будем вычислять элементы C:

$$c_{11} = \sqrt{a_{11}} \qquad c_{21} = \frac{a_{21}}{c_{11}} \qquad c_{31} = \frac{a_{31}}{c_{11}}$$

$$c_{22} = \sqrt{a_{22} - c_{21}^2} \qquad c_{32} = \frac{a_{32} - c_{31}c_{21}}{c_{22}}$$

$$c_{33} = \sqrt{a_{33} - c_{31}^2 - c_{32}^2}$$

- Решение линейной системы: $Ax = b$:
  $A = CC^T$, $CC^Tx = b \to Cy = b \to C^Tx = y$.

Алгоритм разложения Холецкого

Алгоритм разложения Холецкого

```
C = 0
for k = 1, n
    c[k,k] = (a[k,k] - sum(c[k,1:k-1]^2))^1/2
    for i = k+1, n
        c[i,k] = a[i,k] - sum(c[i,1:k-1] * c[k,1:k-1])
        c[i,k] = c[i,k] / c[k,k]
```

Алгоритм разложения Холецкого

```
C = 0
for k = 1, n
    c[k,k] = (a[k,k] - sum(c[k,1:k-1]^2))^1/2
    for i = k+1, n
        c[i,k] = a[i,k] - sum(c[i,1:k-1] * c[k,1:k-1])
        c[i,k] = c[i,k] / c[k,k]
```

- Количество операций

$$\sum_{k=1}^{n}\left(2k + \sum_{i=k+1}^{n} 2k\right) = \sum_{k=1}^{n}(2k + 2k(n-k)) =$$

$$(n+1)n + n^2(n+1) - 2\frac{n(n+1)(2n+1)}{6} = \frac{1}{3}n^3 + O(n^2)$$

Алгоритм разложения Холецкого

```
C = 0
for k = 1, n
    c[k,k] = (a[k,k] - sum(c[k,1:k-1]^2))^1/2
    for i = k+1, n
        c[i,k] = a[i,k] - sum(c[i,1:k-1] * c[k,1:k-1])
        c[i,k] = c[i,k] / c[k,k]
```

- Количество операций

$$\sum_{k=1}^{n}\left(2k + \sum_{i=k+1}^{n} 2k\right) = \sum_{k=1}^{n}(2k + 2k(n-k)) =$$

$$(n+1)n + n^2(n+1) - 2\frac{n(n+1)(2n+1)}{6} = \frac{1}{3}n^3 + O(n^2)$$

- В методе Холецкого нет роста элементов!

QR-разложение

QR-разложение

- Пусть $A \in \mathbb{C}^{m \times n}$, $m \geq n$, $A = [a_1 | a_2 | \ldots | a_n]$

QR-разложение

- Пусть $A \in \mathbb{C}^{m \times n}$, $m \geq n$, $A = [a_1 | a_2 | \dots | a_n]$
- Применим ортогонализацию Грамма-Шмидта:

QR-разложение

- Пусть $A \in \mathbb{C}^{m \times n}$, $m \geq n$, $A = [a_1 | a_2 | \ldots | a_n]$
- Применим ортогонализацию Грамма-Шмидта:
    1. $q_1 \equiv a_1 / \|a_1\|_2$

QR-разложение

- Пусть $A \in \mathbb{C}^{m \times n}$, $m \geq n$, $A = [a_1 | a_2 | \dots | a_n]$
- Применим ортогонализацию Грамма-Шмидта:
  1. $q_1 \equiv a_1 / \|a_1\|_2$
  2. $p_2 = a_2 - (a_2, q_1) q_1$, $q_2 = p_2 / \|p_2\|_2$ (нормировка)

QR-разложение

- Пусть $A \in \mathbb{C}^{m \times n}$, $m \geq n$, $A = [a_1 | a_2 | \dots | a_n]$
- Применим ортогонализацию Грамма-Шмидта:
  1. $q_1 \equiv a_1 / \|a_1\|_2$
  2. $p_2 = a_2 - (a_2, q_1) q_1$, $q_2 = p_2 / \|p_2\|_2$ (нормировка)
  3. $p_3 = a_3 - (a_3, q_1) q_1 - (a_3, q_2) q_2, \dots$

QR-разложение

- Пусть $A \in \mathbb{C}^{m \times n}$, $m \geq n$, $A = [a_1|a_2|\dots|a_n]$
- Применим ортогонализацию Грамма-Шмидта:
  1. $q_1 \equiv a_1/\|a_1\|_2$
  2. $p_2 = a_2 - (a_2, q_1)q_1$, $q_2 = p_2/\|p_2\|_2$ (нормировка)
  3. $p_3 = a_3 - (a_3, q_1)q_1 - (a_3, q_2)q_2, \dots$
- k-й столбец A есть линейная комбинация $q_1, \dots, q_k$. В матричном виде:

QR-разложение

- Пусть $A \in \mathbb{C}^{m \times n}$, $m \geq n$, $A = [a_1 | a_2 | \dots | a_n]$
- Применим ортогонализацию Грамма-Шмидта:
  1. $q_1 \equiv a_1 / \|a_1\|_2$
  2. $p_2 = a_2 - (a_2, q_1) q_1$, $q_2 = p_2 / \|p_2\|_2$ (нормировка)
  3. $p_3 = a_3 - (a_3, q_1) q_1 - (a_3, q_2) q_2, \dots$
- k-й столбец A есть линейная комбинация $q_1, \dots, q_k$. В матричном виде:

$$\left[\begin{array}{c|c|c} a_1 & \dots & a_n \end{array}\right] = \underbrace{\left[\begin{array}{c|c|c} q_1 & \dots & q_n \end{array}\right]}_{Q} \underbrace{\begin{bmatrix} r_{11} & \dots & \times \\ 0 & \ddots & \times \\ 0 & \dots & r_{nn} \end{bmatrix}}_{R} = QR$$

Алгоритм Грамма-Шмидта

Алгоритм Грамма-Шмидта

```
for j = 1, k
    p[:,j] = a[:,j]
    for i = 1, j - 1
        p[:,j] = p[:,j] - q[:,i] * <a[:,j], q[:,i]>
    q[:,j] = p[:,j] / ||p[:,j]||_2
```

Алгоритм Грамма-Шмидта

```
for j = 1, k
    p[:,j] = a[:,j]
    for i = 1, j - 1
        p[:,j] = p[:,j] - q[:,i] * <a[:,j], q[:,i]>
    q[:,j] = p[:,j] / ||p[:,j]||_2
```

- В стандартном алгоритме происходит *потеря ортогональности*

Алгоритм Грамма-Шмидта

```
for j = 1, k
    p[:,j] = a[:,j]
    for i = 1, j - 1
        p[:,j] = p[:,j] - q[:,i] * <a[:,j], q[:,i]>
    q[:,j] = p[:,j] / ||p[:,j]||_2
```

- В стандартном алгоритме происходит *потеря ортогональности*
- На практике используют модифицированный алгоритм Грамма-Шмидта:

Алгоритм Грамма-Шмидта

```
for j = 1, k
    p[:,j] = a[:,j]
    for i = 1, j - 1
        p[:,j] = p[:,j] - q[:,i] * <a[:,j], q[:,i]>
    q[:,j] = p[:,j] / ||p[:,j]||_2
```

- В стандартном алгоритме происходит *потеря ортогональности*
- На практике используют модифицированный алгоритм Грамма-Шмидта:

```
for j = 1, k
    p[:,j] = a[:,j]
    for i = 1, j - 1
        p[:,j] = p[:,j] - q[:,i] * <p[:,j], q[:,i]>
    q[:,j] = p[:,j] / ||p[:,j]||_2
```

Решение линейной системы с помощью QR-разложения

Решение линейной системы с помощью QR-разложения

- Можно решить систему $Ax = b$ в 3 шага

$$Q^*(QR)x = Q^*b \Rightarrow Rx = Q^*b = y$$

Решение линейной системы с помощью QR-разложения

- Можно решить систему $Ax = b$ в 3 шага

$$Q^*(QR)x = Q^*b \Rightarrow Rx = Q^*b = y$$

1. $A = QR$ - вычисляем разложение $(O(n^3))$

Решение линейной системы с помощью QR-разложения

- Можно решить систему $Ax = b$ в 3 шага

$$Q^*(QR)x = Q^*b \Rightarrow Rx = Q^*b = y$$

1. $A = QR$ - вычисляем разложение $(O(n^3))$
2. вычисляем $y = Q^*b$ $(O(n^2))$

Решение линейной системы с помощью QR-разложения

- Можно решить систему $Ax = b$ в 3 шага

$$Q^*(QR)x = Q^*b \Rightarrow Rx = Q^*b = y$$

1. $A = QR$ - вычисляем разложение $(O(n^3))$
2. вычисляем $y = Q^*b$ $(O(n^2))$
3. $x = R^{-1}y$ - решаем систему с треугольной матрицей за $O(n^2)$

Решение линейной системы с помощью QR-разложения

- Можно решить систему $Ax = b$ в 3 шага

$$Q^*(QR)x = Q^*b \Rightarrow Rx = Q^*b = y$$

1. $A = QR$ - вычисляем разложение ($O(n^3)$)
2. вычисляем $y = Q^*b$ ($O(n^2)$)
3. $x = R^{-1}y$ - решаем систему с треугольной матрицей за $O(n^2)$

- В 2 раза больше вычислений, чем LU разложение

Решение линейной системы с помощью QR-разложения

- Можно решить систему $Ax = b$ в 3 шага

$$Q^*(QR)x = Q^*b \Rightarrow Rx = Q^*b = y$$

1. $A = QR$ - вычисляем разложение $(O(n^3))$
2. вычисляем $y = Q^*b$ $(O(n^2))$
3. $x = R^{-1}y$ - решаем систему с треугольной матрицей за $O(n^2)$

- В 2 раза больше вычислений, чем LU разложение
- В QR разложении нет проблем, связанных с ростом элементов

Решение линейной системы с помощью QR-разложения

- Можно решить систему $Ax = b$ в 3 шага

$$Q^*(QR)x = Q^*b \Rightarrow Rx = Q^*b = y$$

  1. $A = QR$ - вычисляем разложение ($O(n^3)$)
  2. вычисляем $y = Q^*b$ ($O(n^2)$)
  3. $x = R^{-1}y$ - решаем систему с треугольной матрицей за $O(n^2)$
- В 2 раза больше вычислений, чем LU разложение
- В QR разложении нет проблем, связанных с ростом элементов
- Матрицы Q, R содержат дополнительную информацию

Переопределенные системы, метод наименьших квадратов

Переопределенные системы, метод наименьших квадратов

- $Ax = b$, $A \in \mathbb{C}^{m \times n}$, $m > n$, $rank(A) = n$

Переопределенные системы, метод наименьших квадратов

- $Ax = b$, $A \in \mathbb{C}^{m \times n}$, $m > n$, $rank(A) = n$
- В общем случае система не имеет решения:
  $Ax = b \iff b \in Lin\{a_1, \ldots, a_n\}$

Переопределенные системы, метод наименьших квадратов

- $Ax = b$, $A \in \mathbb{C}^{m \times n}$, $m > n$, $rank(A) = n$
- В общем случае система не имеет решения: $Ax = b \iff b \in Lin\{a_1, \dots, a_n\}$
- Естественное обобщение: $\|r\| = \|Ax - b\| \to \min$

Переопределенные системы, метод наименьших квадратов

- $Ax = b$, $A \in \mathbb{C}^{m\times n}$, $m > n$, $rank(A) = n$
- В общем случае система не имеет решения:
  $Ax = b \iff b \in Lin\{a_1, \dots, a_n\}$
- Естественное обобщение: $\|r\| = \|Ax - b\| \to \min$
- Введем квадратичный функционал:

$$F(x) = \|Ax - b\|_2^2 = (Ax - b, Ax - b) =$$

$$\sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j - b_i \right)^2$$

$$\nabla F(x) = 2(A^* Ax - A^* b) = 0 \iff A^* Ax = A^* b$$

Переопределенные системы, метод наименьших квадратов

- $Ax = b$, $A \in \mathbb{C}^{m \times n}$, $m > n$, $rank(A) = n$
- В общем случае система не имеет решения: $Ax = b \iff b \in Lin\{a_1, \dots, a_n\}$
- Естественное обобщение: $\|r\| = \|Ax - b\| \to \min$
- Введем квадратичный функционал:

$$F(x) = \|Ax - b\|_2^2 = (Ax - b, Ax - b) =$$
$$\sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j - b_i \right)^2$$
$$\nabla F(x) = 2(A^* Ax - A^* b) = 0 \iff A^* Ax = A^* b$$

- Систему $(A^* A)x = A^* b$ называют системой *нормальных уравнений*

МНК, геометрическая интерпретация

МНК, геометрическая интерпретация

- Ax - вектор из линейной оболочки столбцов A

МНК, геометрическая интерпретация

- Ax - вектор из линейной оболочки столбцов A
- Хотим найти в $Lin\{a_1, ..., a_n\}$ вектор, ближайший к вектору b из $\mathbb{R}^m, m > n$.

МНК, геометрическая интерпретация

- Ax - вектор из линейной оболочки столбцов A
- Хотим найти в $Lin\{a_1, ..., a_n\}$ вектор, ближайший к вектору b из $\mathbb{R}^m, m > n$.
- Решение получается *ортогональной проекцией* b на $Lin\{a_1, ..., a_n\}$

МНК, геометрическая интерпретация

- Ax - вектор из линейной оболочки столбцов A
- Хотим найти в $Lin\{a_1, ..., a_n\}$ вектор, ближайший к вектору b из $\mathbb{R}^m, m > n$.
- Решение получается *ортогональной проекцией* b на $Lin\{a_1, ..., a_n\}$
- $$b = Ax + v, v \perp Lin\{a_1, \ldots, a_n\} \Rightarrow A^* b = A^* Ax + 0$$

Решение переопределенной системы

Решение переопределенной системы

- Метод Холецкого:

Решение переопределенной системы

- Метод Холецкого:
  1. Вычисляем матрицу A^*A и вектор A^*b

Решение переопределенной системы

- Метод Холецкого:
  1. Вычисляем матрицу A^*A и вектор A^*b
  2. Вычисляем разложение Холецкого $A^*A = CC^*$

Решение переопределенной системы

- Метод Холецкого:
  1. Вычисляем матрицу A^*A и вектор A^*b
  2. Вычисляем разложение Холецкого $A^*A = CC^*$
  3. Решаем 2 системы $Cy = A^*b$, $C^*x = y$

Решение переопределенной системы

- Метод Холецкого:
  1. Вычисляем матрицу A^*A и вектор A^*b
  2. Вычисляем разложение Холецкого $A^*A = CC^*$
  3. Решаем 2 системы $Cy = A^*b$, $C^*x = y$
- QR-разложение

$$x = (A^*A)^{-1}A^*b = (R^*Q^*QR)^{-1}R^*Q^*b =$$
$$(R^*R)^{-1}R^*Q^*b = R^{-1}(R^*)^{-1}R^*Q^*b = R^{-1}Q^*b$$

Решение переопределенной системы

- Метод Холецкого:
  1. Вычисляем матрицу A^*A и вектор A^*b
  2. Вычисляем разложение Холецкого $A^*A = CC^*$
  3. Решаем 2 системы $Cy = A^*b$, $C^*x = y$
- QR-разложение

$$x = (A^*A)^{-1}A^*b = (R^*Q^*QR)^{-1}R^*Q^*b =$$
$$(R^*R)^{-1}R^*Q^*b = R^{-1}(R^*)^{-1}R^*Q^*b = R^{-1}Q^*b$$

1. Вычисляем $A = QR$

Решение переопределенной системы

- Метод Холецкого:
  1. Вычисляем матрицу A^*A и вектор A^*b
  2. Вычисляем разложение Холецкого $A^*A = CC^*$
  3. Решаем 2 системы $Cy = A^*b$, $C^*x = y$
- QR-разложение

$$x = (A^*A)^{-1}A^*b = (R^*Q^*QR)^{-1}R^*Q^*b =$$
$$(R^*R)^{-1}R^*Q^*b = R^{-1}(R^*)^{-1}R^*Q^*b = R^{-1}Q^*b$$

  1. Вычисляем $A = QR$
  2. Вычисляем Q^*b

Решение переопределенной системы

- Метод Холецкого:
  1. Вычисляем матрицу A^*A и вектор A^*b
  2. Вычисляем разложение Холецкого $A^*A = CC^*$
  3. Решаем 2 системы $Cy = A^*b$, $C^*x = y$
- QR-разложение

$$x = (A^*A)^{-1}A^*b = (R^*Q^*QR)^{-1}R^*Q^*b =$$
$$(R^*R)^{-1}R^*Q^*b = R^{-1}(R^*)^{-1}R^*Q^*b = R^{-1}Q^*b$$

  1. Вычисляем $A = QR$
  2. Вычисляем Q^*b
  3. Решаем систему $Rx = Q^*b$

Пример применения МНК, регрессия

Пример применения МНК, регрессия

- Даны значения функции $y_1, \dots, y_m$ в точках $x_1, \dots, x_m$

Пример применения МНК, регрессия

- Даны значения функции $y_1, \ldots, y_m$ в точках $x_1, \ldots, x_m$
- Хотим приблизить функцию многочленом степени $n - 1$, $n \leq m$

Пример применения МНК, регрессия

- Даны значения функции $y_1, \dots, y_m$ в точках $x_1, \dots, x_m$
- Хотим приблизить функцию многочленом степени $n-1$, $n \leq m$

$$
\begin{bmatrix} 1 & x_1^1 & \cdots & x_1^{n-1} \\ 1 & x_2^1 & \cdots & x_2^{n-1} \\ & \vdots & & \vdots \\ 1 & x_m^1 & \cdots & x_m^{n-1} \end{bmatrix}
\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix}
=
\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}
$$

Пример применения МНК, регрессия

- Даны значения функции $y_1, \dots, y_m$ в точках $x_1, \dots, x_m$
- Хотим приблизить функцию многочленом степени $n-1$, $n \le m$

$$\begin{bmatrix} 1 & x_1^1 & \cdots & x_1^{n-1} \\ 1 & x_2^1 & \cdots & x_2^{n-1} \\ & \vdots & & \vdots \\ 1 & x_m^1 & \cdots & x_m^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

- При $n = m$ существует единственное решение

Пример применения МНК, регрессия