Các phương pháp tìm gần đúng ma trận nghịch đảo

Bùi Tiến Thành - MSSV 20190081
Chu Thị Ngân - MSSV 20195904

CTTN Toán tin K64

Tháng Mười một, 2020

Mục lục

ĐẠI HỌC BÁCH KHOA

Tại sao phải giải gần đúng?

Các phương pháp giải đúng:

- Bắt buộc phải duyệt toàn bộ các phần tử của ma trận và phải lưu toàn bộ ma trận trong bộ nhớ, đặc biệt với các thuật toán đệ quy
- Không thể kiểm soát sai số tính toán do giới hạn tính toán của máy tính

$\Rightarrow$ **Giải pháp?**

Các phương pháp đưa ra

- Phương pháp Newton
- Phương pháp lặp Jacobi
- Phương pháp lặp Gauss-Seidel

Mục lục

Ý tưởng và công thức lặp

Ý tưởng

Từ phương pháp Newton cho $ax = 1$:

$$x_{n+1} = x_n * (2 - ax_n) \text{ với } n \in \mathbb{N}$$

Áp dụng vào ma trận với E là ma trận đơn vị cùng cấp:

Công thức lặp

$$X_{k+1} = X_k(2E - AX_k) \text{ với } k \in \mathbb{N}$$

Điều kiện hội tụ và công thức sai số

Điều kiện hội tụ

$$\left\|A^{-1} - X_k\right\| \leq \left\|A^{-1}\right\| \left\|E - AX_0\right\|^{2^k} \xrightarrow{k\to\infty} 0 \text{ nếu } \left\|E - AX_0\right\| < 1$$

Công thức sai số

Khi $\left\|E - AX_0\right\| \leq q < 1$:

$$\left\|A^{-1} - X_k\right\| \leq \frac{\left\|X_0\right\|}{1-q} q^{2^k}$$

Input, output

- Input: Ma trận A, xấp xỉ đầu X_0 và sai số ε. Giả thiết coi như các điều kiện hội tụ thỏa mãn.
- Output: Ma trận xấp xỉ A^{-1}
- Code:

https://github.com/bu1th4nh/TALENTED-K64MI/blob/master/MI3040/report-code/newton.py

Mã giả

Algorithm 1: Phương pháp Newton tìm ma trận nghịch đảo

Input: Ma trận A, X_0, ε
Output: Ma trận $X^* = A^{-1}$ là ma trận nghịch đảo
begin
  Nhập A, X_0, ε;
  $q \longleftarrow \|E - AX_0\|$;
  $q2k \longleftarrow q$;
  $X \longleftarrow X_0$;
  while $\frac{\|X_0\| * q2k}{1-q} > \varepsilon$ **do**
    $X \longleftarrow X(2E - AX)$;
    $q2k \longleftarrow q2k^2$;
  end
  Đưa ra X chính là ma trận nghịch đảo;
end

Mục lục

Ý tưởng chung

Ý tưởng

Đưa phương trình $AX = E$ về dạng công thức lặp sau:

$$X_{k+1} = BX_k + D$$

Ta cũng chứng minh được nếu $\|B\| < 1$ thì với giá trị xấp xỉ đầu bất kì thì X luôn hội tụ về nghiệm (Phương pháp lặp đơn).
$\longrightarrow$ Vậy làm thế nào để $\|B\| < 1$?

Các phương pháp được lựa chọn

- Phương pháp lặp Jacobi (dựa trên phương pháp lặp đơn)
- Phương pháp lặp Gauss - Seidel (dựa trên phương pháp lặp Siedel)

Mục lục

Công thức lặp và điều kiện hội tụ

Đặt: $T = diag\{\frac{1}{A_{11}}, \frac{1}{A_{22}}, ..., \frac{1}{A_{nn}}\}$ là ma trận đường chéo cấp n, ta có:

Công thức lặp

$$X_{k+1} = (E - TA)X_k + T$$

Điều kiện hội tụ

Để $\|E - TA\| < 1$ thì ma trận A phải chéo trội, nghĩa là:

$$\sum_{j=1, j\neq i}^{n} |A_{ij}| < |A_{ii}| \ \forall i = \overline{1, n} \text{ chéo trội hàng}$$

$$\text{hoặc} \sum_{i=1, i\neq j}^{n} |A_{ij}| < |A_{jj}| \ \forall j = \overline{1, n} \text{ chéo trội cột}$$

Công thức sai số

Cho $B = E - TA, B_1 = E - AT$

Trường hợp chéo trội hàng: Với $\|B\|_\infty \leq q < 1$

$$\|X_k - X^*\|_\infty \leq \frac{q}{1-q} \|X_k - X_{k-1}\|_\infty$$

$$\|X_k - X^*\|_\infty \leq \frac{q^k}{1-q} \|X_1 - X_0\|_\infty$$

Trường hợp chéo trội cột: Cho $\lambda = \frac{max|A_{ii}|}{min|A_{ii}|}, \|B_1\|_1 \leq q < 1$

$$\|X_k - X^*\|_1 \leq \lambda \frac{q}{1-q} \|X_k - X_{k-1}\|_1$$

$$\|X_k - X^*\|_1 \leq \lambda \frac{q^k}{1-q} \|X_1 - X_0\|_1$$

Thuật toán chính

Algorithm 2: Phương pháp Jacobi tìm ma trận nghịch đảo

Input: Ma trận A chéo trội, sai số ε
Output: Ma trận $X^* = A^{-1}$ là ma trận nghịch đảo
begin

Nhập A, ε;

$p \longleftarrow$ checkDomination(A);
$T \longleftarrow diag\{\frac{1}{A_{11}}, \frac{1}{A_{22}}, \dots, \frac{1}{A_{nn}}\}$;
$B \longleftarrow E - TA, B_1 \longleftarrow E - AT$;
$q \longleftarrow$ getNorm(B, B_1, p);
$\lambda \longleftarrow$ getLambda(A, p);

$X^* \longleftarrow$ iterate($X_0 \leftarrow A, B, T, q, \lambda, p, \varepsilon$);
Trả về X^* là ma trận nghịch đảo;

end

Chuyển sang phần code

Function checkDomination(A)

Input: Ma trận A

Output: Giá trị kiểm tra p bằng 1 nếu A chéo trội hàng, -1 nếu A chéo trội cột, 0 khi A không chéo trội

begin

    $row\_dom \longleftarrow 1$, $col\_dom \longleftarrow 1$;

    for $i = 1$ **to** n **do**

        if $\sum_{j=1, j\neq i}^{n} |A_{ij}| >= |A_{ii}|$ **then** $row\_dom \longleftarrow 0$;

        if $\sum_{j=1, j\neq i}^{n} |A_{ji}| >= |A_{ii}|$ **then** $col\_dom \longleftarrow 0$;

    end

    if $row\_dom = 1$ **then return** *1*;

    if $col\_dom = 1$ **then return** *-1*;

    return *0* ;

end

Trở về thuật toán chính

Function getNorm(A, A_1, p)

Input: Ma trận A, giá trị kiểm tra p
Output: $\|A\|_\infty$ nếu $p = 1$, $\|A_1\|_1$ nếu $p = -1$
begin
  if $p = 1$ **then return** $\|A\|_\infty$;
  if $p = -1$ **then return** $\|A_1\|_1$;
end

Function getLambda(A, p)

Input: Ma trận A, giá trị kiểm tra p
Output: $\lambda = 1$ nếu $p = 1$, $\lambda = \frac{max|A_{ii}|}{min|A_{ii}|}$ nếu $p = -1$
begin
  if $p = 1$ **then return** *1*;
  $max\_A \longleftarrow |A_{11}|$;
  $min\_A \longleftarrow |A_{11}|$;
  for $i = 1$ **to** n **do**
    $max\_A = max(max\_A, |A_{ii}|)$;
    $min\_A = min(min\_A, |A_{ii}|)$;

Lặp - Đánh giá tiên nghiệm

Function iterate($X_0, B, T, q, \lambda, p, \varepsilon$)

Input: Ma trận xấp xỉ đầu X_0, B, T, hệ số q, λ, giá trị kiểm tra p và sai số ε

Output: X^* là ma trận nghịch đảo theo đánh giá tiên nghiệm

begin

$qk \longleftarrow 1, X \longleftarrow X_0$;

$predecessor\_norm \longleftarrow$ getNorm$((BX_0 + T) - X_0, p)$;

while $\frac{\lambda * qk * predecessor\_norm}{1-q} > \varepsilon$ **do**

$X \longleftarrow BX + T$;

$qk \longleftarrow qk * q$;

end

return X

end

Trở về thuật toán chính

Lặp - Đánh giá hậu nghiệm

Function iterate($X_0, B, T, q, \lambda, p, \varepsilon$)

Input: Ma trận xấp xỉ đầu X_0, B, T, hệ số q, λ, giá trị kiểm tra p và sai số ε

Output: X^* là ma trận nghịch đảo theo đánh giá hậu nghiệm

begin

$old\_X \longleftarrow X_0$;

$new\_X \longleftarrow BX_0 + T$;

while $\frac{\lambda * q * getNorm(new\_X - old\_X, p)}{1-q} > \varepsilon$ **do**

$old\_X \longleftarrow new\_X$;

$new\_X \longleftarrow B * old\_X + T$;

end

return new_X

end

Trở về thuật toán chính

Chương trình

https://github.com/bu1th4nh/TALENTED-K64MI/blob/master/MI3040/report-code/jacobian.py

Mục lục

Công thức lặp và điều kiện hội tụ

Đặt $B = E - TA$ với $T = diag\{\frac{1}{A_{11}}, \frac{1}{A_{22}}, \ldots, \frac{1}{A_{nn}}\}$ tương tự phương pháp Jacobi

Công thức lặp - Phương pháp Gauss-Seidel nguyên bản [4]

$$X_i^{(k+1)} = \sum_{j=1}^{i-1} B_{ij} X_j^{(k+1)} + \sum_{j=i+1}^{n} B_{ij} X_j^{(k)} + T_i$$

với mọi $i = \overline{1, n}$ và A_i là dòng i của ma trận A

Mở rộng: Hệ số điều chỉnh cho phương pháp Gauss-Seidel [1]

$$X_i^{k+1} = (1-\omega) X_i^k + \omega \left[\sum_{j=1}^{i-1} B_{ij} X_j^{(k+1)} + \sum_{j=i+1}^{n} B_{ij} X_j^{(k)} + T_i \right]$$

với ω là hệ số điều chỉnh (relaxation factor)

Điều kiện hội tụ: Tương tự phương pháp **Jacobi**, tức là A phải chéo trội

Trường hợp chéo trội hàng

Cho:

$$q = \max_{1 \le i \le n} \frac{\sum_{j=i}^{n} |B_{ij}|}{1 - \sum_{j=1}^{i-1} |B_{ij}|} \le \|B\|_\infty < 1$$

Công thức sai số [4]

$$\|X_k - X^*\|_\infty \le \frac{q}{1-q} \|X_k - X_{k-1}\|_\infty$$

$$\|X_k - X^*\|_\infty \le \frac{q^k}{1-q} \|X_1 - X_0\|_\infty$$

Trường hợp chéo trội cột

Cho:

$$q = \max_{1 \le i \le n} \frac{\sum_{j=1}^{i} |B_{ji}|}{1 - \sum_{j=i+1}^{n} |B_{ji}|} \le \|B\|_1 < 1$$

$$S = \max_{1 \le i \le n} \sum_{j=i+1}^{n} |B_{ji}|$$

Công thức sai số[4]

$$\|X_k - X^*\|_1 \le \frac{q}{(1-S)(1-q)} \|X_k - X_{k-1}\|_1$$

$$\|X_k - X^*\|_1 \le \frac{q^k}{(1-S)(1-q)} \|X_1 - X_0\|_1$$

Thuật toán chính

Algorithm 3: Phương pháp Gauss-Seidel tìm ma trận nghịch đảo

Input: Ma trận A chéo trội, sai số ε, hệ số điều chỉnh ω
Output: Ma trận $X^* = A^{-1}$ là ma trận nghịch đảo
begin

Nhập A, ε;

$p \longleftarrow$ checkDomination(A);
$T \longleftarrow diag\{\frac{1}{A_{11}}, \frac{1}{A_{22}}, \ldots, \frac{1}{A_{nn}}\}$;
$B \longleftarrow E - TA$;
$S \longleftarrow$ getSCoeff(B, p);
$q \longleftarrow$ getqCoeff(B, p);

$X^* \longleftarrow$ iterate$(X_0 \leftarrow A, B, T, S, q, p, \omega, \varepsilon)$;
Trả về X^* là ma trận nghịch đảo;

end

Chuyển sang phần code

Function getSCoeff(A, p)

Input: Ma trận A, giá trị kiểm tra p

Output: $S = 0$ nếu $p = 1$, $\max\limits_{1\le i\le n} \sum\limits_{j=i+1}^{n} |A_{ji}|$ nếu $p = -1$

begin

  if $p = 1$ **then return** *0*;

  $S \longleftarrow 0$;

  for $i = 1$ **to** n **do** $S \longleftarrow max(S, \sum\limits_{j=i+1}^{n} |A_{ji}|)$;

  return S

end

Trở về thuật toán chính

Function getqCoeff(A, p)

Input: Ma trận A, giá trị kiểm tra p

Output: $q = \max\limits_{1\le i\le n} \dfrac{\sum\limits_{j=i}^{n}|A_{ij}|}{1-\sum\limits_{j=1}^{i-1}|A_{ij}|}$ nếu $p=1$, $\max\limits_{1\le i\le n} \dfrac{\sum\limits_{j=1}^{i}|A_{ji}|}{1-\sum\limits_{j=i+1}^{n}|A_{ji}|}$ nếu $p=-1$

begin

$q \longleftarrow 0;$

for $i = 1$ **to** n **do**

$Q1 \longleftarrow 0, Q2 \longleftarrow 0;$

if $p = 1$ **then** $Q1 \longleftarrow \sum\limits_{j=i}^{n} |A_{ij}|, Q2 \longleftarrow \sum\limits_{j=1}^{i-1} |A_{ij}|;$

else $Q1 \longleftarrow \sum\limits_{j=1}^{i} |A_{ji}|, Q2 \longleftarrow \sum\limits_{j=i+1}^{n} |A_{ji}|;$

$q \longleftarrow max(q, \frac{Q1}{1-Q2});$

end

return q

end

Lặp - Đánh giá tiên nghiệm

Function iterate(X_0, B, T, S, q, p, ω, ε)

Input: Ma trận xấp xỉ đầu X_0, B, T, hệ số S, q, giá trị kiểm tra p, hệ số điều chỉnh ω và sai số ε

Output: X^* là ma trận nghịch đảo theo đánh giá tiên nghiệm

begin

$qk \longleftarrow 1$;

$X \longleftarrow X_0$;

$X_1 \longleftarrow \text{nextIteration}(X_0, B, T, \omega)$;

$predecessor\_norm \longleftarrow getNorm(X_1 - X_0, p)$;

while $\frac{qk*predecessor\_norm}{(1-q)*(1-S)} > \varepsilon$ **do**

$X \longleftarrow \text{nextIteration}(X, B, T, \omega)$;

$qk \longleftarrow qk * q$;

end

return X

end

Lặp - Đánh giá hậu nghiệm

Function iterate(X_0, B, T, S, q, p, ω, ε)

Input: Ma trận xấp xỉ đầu X_0, B, T, hệ số S, q, giá trị kiểm tra p, hệ số điều chỉnh ω và sai số ε

Output: X^* là ma trận nghịch đảo theo đánh giá hậu nghiệm

begin

  $old\_X \longleftarrow X_0$;

  $new\_X \longleftarrow \text{nextIteration}(X_0, B, T, \omega)$;

  while $\frac{\lambda * q * getNorm(new\_X - old\_X, p)}{1-q} > \varepsilon$ **do**

    $old\_X \longleftarrow new\_X$;

    $new\_X \longleftarrow \text{nextIteration}(X, B, T, \omega)$;

  end

  return new_X

end

Trở về thuật toán chính

Function nextIteration(old_X, B, T, ω)

Input: Ma trận old_X, B, T, và hệ số ω

Output: Đưa ra ma trận tiếp theo thu được từ ma trận ban đầu old_X theo công thức lặp Gauss-Seidel

begin

$new\_X \longleftarrow$ Ma trận không cấp n;

for $i = 1$ **to** n **do**

$$new\_X_i = \sum_{j=1}^{i-1} B_{ij} * new\_X_j + \sum_{j=i+1}^{n} B_{ij} * old\_X_j + T_i;$$

end

return $(1-\omega) * old\_X + \omega * new\_X$

end

Trở về thuật toán chính
Trở về tiến trình lặp

Chương trình

https://github.com/bu1th4nh/TALENTED-K64MI/blob/master/MI3040/report-code/gauss_seidel.py

Mục lục

Ưu, nhược điểm

Phương pháp	Newton	Lặp Jacobi	Lặp Gauss-Seidel
Ưu điểm	Kiểm soát được sai số tính toán, sai số được cải thiện sau m Tốc độ hội tụ nhanh trong một số trường hợp		
Nhược điểm	Khó tìm giá trị xấp xỉ đầu X_0	Yêu cầu ma trận phải chéo trộ	

$\longrightarrow$ Tìm giá trị xấp xỉ đầu cho X_0 trong phương pháp Newton?

Ưu, nhược điểm

Phương pháp	Newton	Lặp Jacobi	Lặp Gauss-Seidel
Ưu điểm	Kiểm soát được sai số tính toán, sai số được cải thiện sau n Tốc độ hội tụ nhanh trong một số trường hợp		
Nhược điểm	Khó tìm giá trị xấp xỉ đầu X_0	Yêu cầu ma trận phải chéo trộ	

$\longrightarrow$ Tìm giá trị xấp xỉ đầu cho X_0 trong phương pháp Newton?

- Đặt $X_0 = \dfrac{A}{\|A\|_1 \|A\|_\infty}$ [3]
- Sử dụng kết quả của các phương pháp tính trực tiếp ma trận nghịch đảo làm xấp xỉ đầu

Câu hỏi?

Tài liệu tham khảo

- Richard L. Burden J. Douglas Faires. *Numerical Methods, 4th Edition*. Brooks / Cole, Cengage Learning, 2013, 2003, 1998.
- The Pennsylvania State University Jaan Kiusalaas. *Numerical Methods in Engineering With MATLAB*. Cambridge University Press, 2005.
- Thomas E. Phipps Jr. "The inversion of large matrices: The Pan and Reif algorithm provides a solution". In: *Byte Magazine* 11.04 (4-1986).
- Lê Trọng Vinh. *Giáo trình Giải tích số*. NXB Khoa học và kỹ thuật, 2007.


thanks for attending