Iterativne metode za linearne sustave-metoda bikonjugiranih gradijenata(BCG)

Lea Ćatipović

Kolovoz 2024.

Sadržaj

1 Dvostrani Lanczosov algoritam

Dvostrani Lanczosov algoritam proširenje je simetričnog Lanczosovog algoritma na nesimetrične matrice na način da se dobiva jedna trokoračna rekurzija vezana uz A, a druga vezana uz A^*. Ova metoda konstruira biortogonalne baze za dva Krylovljeva potprostora definiranih za matricu A i matricu A^*:

$$K_k(A, v_1) = \text{span}\{v1, Av_1, \dots, A^{k-1}v_1\} \tag{1}$$

i

$$K_k(A^*, w_1) = \text{span}\{w_1, A^*w_1, \dots, (A^*)^{k-1}w_1\}. \tag{2}$$

Biortogonalnost se očituje u tome da je $\langle v_i, w_j \rangle = 0$, za $i \neq j$.

Algoritam 1 Dvostrani Lanczosov algoritam

Dani su vektori v_1 i w_1 sa $\|v_1\|_2 = 1$ i $\langle v_1, w_1 \rangle = 1$.
Neka su $\beta_0 = \gamma_0 = 0$ i $v_0 = w_0 = 0$.
Za $j = 1, 2, ...$

$$\begin{aligned}
&\text{Izračunaj } Av_j \text{i } A^*w_j,\\
\alpha_j &= \langle Av_j, w_j \rangle,\\
\tilde{v}_{j+1} &= Av_j - \alpha_j v_j - \beta_{j-1}v_{j-1},\\
\tilde{w}_{j+1} &= A^*w_j - \bar{\alpha}_j w_j - \gamma_{j-1}w_{j-1},\\
\gamma_j &= \|\tilde{v}_{j+1}\|_2,\\
v_{j+1} &= \frac{\tilde{v}_{j+1}}{\gamma_j},\\
\beta_j &= \langle v_{j+1}, \tilde{w}_{j+1} \rangle,\\
&\text{ako je } \beta_j = 0 \text{ stani, inače}\\
w_{j+1} &= \frac{\tilde{w}_{j+1}}{\bar{\beta}_j}.
\end{aligned}$$

```
function [V,W,T]=Dvostrani_Lanczos(A,maxiter)

  n=size(A,1);
  v1=zeros(n,1);       %dani vektor v1 t.d. norm(v1,2)=1
  w1=zeros(n,1);         %dani vektor w1 t.d. <v1,w1>=1;
  v1(1)=1;
  w1(1)=1;
  beta=0; gama=0;
  Alpha=[];
  Beta=[];
  Gama=[];
  V=[];
  W=[];
  v0=zeros(n,1);
  w0=zeros(n,1);

  for k=1:maxiter
    V=[V,v1];
    W=[W,w1];
    Av=A*v1;
    Aw=A'*w1;
    alpha=Av'*w1;
    Alpha=[Alpha;alpha];
    vtilda=Av-alpha*v1-beta*v0;
    wtilda=Aw-conj(alpha)*w1-gama*w0;
    gama=norm(vtilda,2);

    if gama==0
      break;
    endif
    Gama=[Gama;gama];
    v1=vtilda/gama;
```

```
    beta=v1'*wtilda;

    if beta==0
      break;
    endif
    Beta=[Beta;beta];
    w1=wtilda/conj(beta);
    v0=V(:,end);
    w0=W(:,end);

  endfor
  vtilda
  T=diag(Alpha)+diag(Beta(1:end-1),1)+diag(Gama(1:end-1),-1);
end
```

Neka je V_k matrica sa stupcima $v_1, .., v_k$ i W_k matrica sa stupcima $w_1, ..., w_k$. Tada rekurzije iz algoritma matrično možemo zapisati na sljedeći način:

$$AV_k = V_k T_k + \gamma_k v_{k+1} \xi_k^T = V_{k+1} T_{k+1,k}, \tag{3}$$

$$A^* W_k = W_k T_k^* + \bar{\beta}_k w_{k+1} \xi_k^T = W_{k+1} \hat{T}_{k+1,k}, \tag{4}$$

gdje je T_k $k \times k$ tridijagonalna matrica koeficijenata rekurzije

$$T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \gamma_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \gamma_{k-1} & \alpha_k \end{bmatrix}$$

$(k+1) \times k$ matrice $T_{k+1,k}$ i $\hat{T}_{k+1,k}$ imaju T_k tj. T_k^* kao gornji $k \times k$ blok i zadnji red popunjen nulama osim $(k+1, k)$ elementa koji je jednak γ_k tj. $\bar{\beta}_k$. Također se svojstvo biortogonalnosti može matrično prikazati u obliku $V_k^* W_k = I$. Vrijedi sljedeći teorem:

Teorem 1. *Ako dvostrani Lanczosov algoritam ne zakaže do k-tog koraka,a to znači da su definirani svi vektori v_j i w_j iz dvostranog Lanczosovog algoritma, odnosno da je $\langle v_j, w_j \rangle \neq 0$ za $j = 1, ..., k+1$, tada je*

$$\langle v_i, w_j \rangle = 0 \text{ za svake } i, j \leq k+1, i \neq j.$$

Ako Lanczosovi vektori nisu definirani, tj. kada je $\beta_j = 0$ može doći do dvije različite situacije zakazivanja. Prva, koju označavamo kao *regularno zaustavljanje* je kada je dvostrani Lanczosov algoritam pronašao invarijantni potprostor, odnosno ako je $\tilde{v}_{j+1} = 0$ ili $\tilde{w}_{j+1} = 0$. U prvom slučaju vektori $v_1, .., v_j$ razapinju A-invarijantni potprostor, a u drugom vektori $w_1, ..., w_j$ razapinju A^*-invarijantni potprostor. U drugom slučaju, kojeg označavamo kao *ozbiljni slom algoritma* vrijedi $\langle \tilde{v}_{j+1}, \tilde{w}_{j+1} \rangle = 0$, ali ni jedan od vektora $\tilde{v}_{j+1}$ i $\tilde{w}_{j+1}$ nije jednak nuli. Tada netrivijalni vektori $v_{j+1} \in K_{j+1}(A, v_1)$ i $w_{j+1} \in K_{j+1}(A^*, w_1)$ sa svojstvom da $\langle v_{j+1}, w_i \rangle = \langle w_{j+1}, v_i \rangle = 0$ za $i \leq j$ ne postoje.

2 Bikonjugirani gradijenti (BCG)

Računamo aproksimaciju rješenja sustava $Ax = b$ s početnom iteracijom x_0 i rezidualom $r_0 = b - Ax_0$. Neka je $x_k = x_0 + V_k y_k$. Uz izbor $v_1 = \frac{r_0}{\|r_0\|_2}$ i w_1 prozivoljan vektor takav da je $\langle v_1, w_1 \rangle = 1$ jedinstveno su definirani Krylovljevi prostori $K_k(A, v_1)$ i $K_k(A^*, w_1)$ kao i Lanczosovi vektori $v_1, ..., v_k$ i $w_1, ..., w_k$. Jedan od mogućih izbora za y_k je takav da $r_k = r_0 - AV_k y_k$ bude ortogonalan na $K_k(A^*, w_1)$ pa od tu slijedi jednakost

$$W_k^* r_k = W_k^* r_0 - W_k^* A V_k y_k = 0.$$

Zbog biortogonalnosti Lanczosovih vektora vrijedi $W_k^* r_0 = \beta \xi_1$ uz $\beta = \|r_0\|_2$ i $W_k^* A V_k = T_k$ pa slijedi jednadžba

$$T_k y_k = \beta \xi_1. \tag{5}$$

Također iz (3) vrijedi $r_k = -\gamma_k \xi_k^T y_k v_{k+1}$, odnosno rezidual ima smjer vektora v_{k+1} i ortogonalan je na vektore $w_1, ..., w_k$. Analogno rješavamo i sustav $A^* \hat{x} = \hat{b}$. Daljnjim postupkom dolazimo do niza reziduala r_k i $\hat{r}_k$ koji su biortogonalni i niza vektora smjea p_k i $\hat{p}_k$ koji su A-ortogonalni. Odavde iz dvostranog Lanczosovog algoritma slijedi algoritam koji liči na konjugirane gradijente i kojeg nazivamo algoritam *bikonjugiranih gradijenata*.

Algoritam 2 BCG

Dana je početna iteracija x_0,
$r_0 = b - Ax_0$.
Izaberi $\hat{r}_0$ takav da je $\langle r_0, \hat{r}_0 \rangle \neq 0$.
$p_0 = r_0$.
$\hat{p}_0 = \hat{r}_0$.
Za $k = 1, 2, ...$

$$\begin{aligned} &\text{izračunaj } Ap_{k-1}, \\ &\text{izračunaj } A^*\hat{p}_{k-1}, \\ \alpha_{k-1} &= \frac{\langle r_{k-1}, \hat{r}_{k-1} \rangle}{\langle Ap_{k-1}, \hat{p}_{k-1} \rangle}, \\ x_k &= x_{k-1} + \alpha_{k-1} p_{k-1}, \\ r_k &= r_{k-1} - \alpha_{k-1} A p_{k-1}, \\ \hat{r}_k &= \hat{r}_{k-1} - \bar{\alpha}_{k-1} A^* \hat{p}_{k-1}, \\ \beta_{k-1} &= \frac{\langle r_k, \hat{r}_k \rangle}{\langle r_{k-1}, \hat{r}_{k-1} \rangle}, \\ p_k &= r_k + \beta_{k-1} p_{k-1}, \\ \hat{p}_k &= \hat{r}_k + \bar{\beta}_{k-1} \hat{p}_{k-1}. \end{aligned}$$

```
function [x,k,rez,rezh]=BCG(A,b,tol,maxiter)
  n=size(A,1);
  x=zeros(n,1);
  r=b-A*x;
  rh=r;

  if r'*rh==0
    error;
  endif

  p=r;
  ph=rh;

  for k=1:maxiter
    Ap=A*p;
    Aph=A'*ph;
    alpha=(r'*rh)/(Ap'*ph);
    x=x+alpha*p;
    r1=r-alpha*Ap;
    if norm(r1)<tol
      break;
    endif
    r1h=rh-conj(alpha)*Aph;
    if norm(r1h)<tol
      break;
    endif
    beta=(r1'*r1h)/(r'*rh);
    p1=r1+beta*p;
    p1h=r1h+conj(beta)*ph;

    r=r1;
    rh=r1h;
```

```
    p=p1;
    ph=p1h;
  endfor

  rez=norm(r);
  rezh=norm(rh);
end
```

Ako je u (5) T_k singularna onda ta jednakost ne mora imati rješenje, tj. tada ne postoji aproksimacija $x_k = x_0 + V_k y_k$ za koju je $W_k^* r_k = 0$ i tada algoritam zakazuje. Kod rješavanja linearnih sustava sa nehermitskom matricom GMRES metoda zadovoljava neko svojstvo minimalnosti na Krylovljevim potprostorima koje generira matrica A, ali ne može se izračunati s malim brojem operacija i uz malo memorije po iteraciji. Za BCG metodu pak vrijedi obratno. Uz to ova metoda može zakazati na još dva načina, slomom dvostranog Lanczosovog algoritma tj. kada bi došlo do dijeljenja s nulom zbog $\beta_k = 0$, a drugi je slučaj singularne matrice T_k.

3 Primjeri

3.1 Primjer 1

```
>> A =

   2  -1   0

  -1   2  -1

   0  -1   2


>> b =

   1

   0

   1

>> [x,k,rez,rezh]=BCG(A,b,1e-6,100)

x =

   1

   1

   1


k = 2

rez = 1

rezh = 1


>> [V,W,T]=Dvostrani_Lanczos(A,2)

vtilda =

   1

   1

   1


V =

   1   0

   0  -1

   0   0
```

```
W =
   1   0
   0  -1
   0   0

T =
   2   1
   1   2

>> (W'*A)*V
ans =
   2   1
   1   2
```

U ovom primjeru rezidual dosegne nulu i BCG program završava u m=2 iteracije. Za vrijednosti V_m, W_m i T_m dobivene pomoću dvostranog Lanczosovog algoritma zaista vrijedi jednakost $(W_m)^T A V_m = T_m$.

3.2 Primjer 2

```
    >> A =
   4   1  -2
   1   4   1
   2  -1   3

>>b =
   1
   2
   3

>> [x,k,rez,rezh]=BCG(A,b,1e-6,100)
x =
   0.5507
```

```
   0.1884
   0.6957

k = 3
rez = 2.6511
rezh = 4.8240
>> [V,W,T]=Dvostrani_Lanczos(A,3)
vtilda =   0
  -2.2204e-16
            0

V =
   1.0000         0         0
        0    0.4472    0.8944
        0    0.8944    0.4472

W =
   1.0000         0         0
        0   -0.7454    1.4907
        0    1.4907   -0.7454

T =
   4.0000   -1.3416         0
   2.2361    1.3333   -2.3333
        0    2.3333    5.6667

>> (W'*A)*V
ans =
   4.0000   -1.3416         0
   2.2361    1.3333   -2.3333
        0    2.3333    5.6667
```

3.3 Primjer 3

```
 >> A =
   1.0000e+00   2.0000e+00   3.0000e+00   4.0000e+00   5.0000e+00
   1.0000e-03   1.0000e+00   1.0000e-03   1.0000e-03   1.0000e-03
   5.0000e+00   4.0000e+00   3.0000e+00   2.0000e+00   1.0000e+00
   1.0000e+00   1.0000e+00   1.0000e+00   1.0000e+00   1.0000e+00
   2.0000e+00   2.0000e+00   2.0000e+00   2.0000e+00   2.0000e+00

>> b =
   1.5000e+01
   5.0000e-03
   1.5000e+01
   5.0000e+00
   1.0000e+01

>> gmres(A,b)
gmres converged at iteration 3 to a solution with relative residual 4.07508e-16
ans =
   1.3889e+00
   5.2584e-17
   1.9444e+00
   5.5556e-01
   1.1111e+00

>> [x,k,rez,rezh]=BCG(A,b,1e-6,100)
x =
   1.1960e+00
  -6.0494e-06
   2.3210e+00
   5.7260e-01
   9.1049e-01
```

```
k = 100
rez = 2.2429e-03
rezh = 9.8358
```

Gornji primjer u kojem je matrica A nehermitska i slabo uvjetovana pokazuje da je GMRES metoda u ovom slučaju bolji izbor od BCG metode.

3.4 Primjer 4

```
 >>A =
   4   1   0   0   0
   1   4   1   0   0
   0   1   4   1   0
   0   0   1   4   1
   0   0   0   1   4

>> b =
   1
   2
   3
   4
   5

>> gmres(A,b)
gmres converged at iteration 5 to a solution with relative residual 4.99e-16
ans =
   0.1679
   0.3282
   0.5192
   0.5949
   1.1013

>> [x,k,rez,rezh]=BCG(A,b,1e-6,100)
x =
```

```
   0.1679
   0.3282
   0.5192
   0.5949
   1.1013
k = 5
rez = 0.010449
rezh = 0.010449
>> tic;gmres(A,b);toc;
Elapsed time is 0.00255108 seconds.
>> tic;[x,k,rez,rezh]=BCG(A,b,1e-6,100);toc;
Elapsed time is 0.000477791 seconds.
```

Ovo je primjer koji pokazuje da postoje slučajevi u kojima je konvergencija BCG metode dobra kao i kod GMRES metode.

Literatura

[1] Nela Bosner, *Iterativne metode za rješavanje linearnih sustava*, magistarski rad