Il precondizionamento di un sistema lineare

Cristina Rolli

1 Introduzione

Si parla di matrice *sparsa* di ordine $n \times n$ quando il numero di elementi diversi da zero (nz) é dell'ordine di $n \cdot k$, dove k é una costante, generalmente molto piccola rispetto alla dimensione della matrice.

Il nostro obiettivo é risolvere un sistema lineare della forma

$$A\mathbf{x} = \mathbf{b}$$

dove A é una matrice sparsa generalmente di grandi dimensioni. Le matrici sparse, per la loro particolare struttura, sono memorizzate in modo compatto, cioé vengono memorizzati solo gli elementi diversi da zero (*memorizzazione a coordinate*). Un altro fatto che caratterizza tali matrici é che il costo della moltiplicazione matrice-vettore é dell'ordine del numero degli elementi diversi da zero della matrice, cioé $n \cdot k$, mentre di solito é n^2 .

Qual'é quindi il problema? Il primo tentativo di risoluzione di un sistema lineare del tipo $A\mathbf{x} = \mathbf{b}$ é un metodo diretto. I metodi diretti sono tutti accomunati dal tentativo di fattorizzazione della matrice del sistema, del tipo: $A = LU$. Effettuata tale fattorizzazione ci si riduce quindi a risolvere due sistemi lineari piú semplici:

$$L\mathbf{y} = \mathbf{b} \qquad\qquad U\mathbf{x} = \mathbf{y}.$$

Se la matrice A é sparsa non é detto che U e L lo siano. In realtá ci si accorge che in genere il numero di elementi diversi da zero aumenta in modo considerevole implemetando tale fattorizzazione. Di conseguenza i fattori occuperanno molta piú memoria rispetto alla matrice originaria e la risoluzione dei sistemi lineari triangolari sará molto costosa.

Siccome il tentativo con i metodi diretti non sembra essere una buona soluzione per sistemi con matrici sparse, allora si cerca di trovare risposta al problema attraverso metodi iterativi (Jacobi, Gauss-Seidel,SOR, gradiente, gradiente coniugato,...).

2 Il metodo del gradiente coniugato (CG)

É stato ideato per matrici simmetriche e definite positive. L'idea che sta alle spalle del gradiente coniugato é quella di seguire il verso opposto del gardiente della funzione

$$\phi(\mathbf{y}) = \frac{1}{2}\mathbf{y}^T A\mathbf{y} - \mathbf{y}^T\mathbf{b}$$

avendo quindi come formula di ricorrenza

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \nabla\phi(\mathbf{x}^k).$$

Si impone inoltre che ad ogni iterazione il residuo $\mathbf{r}_k$ sia ortogonale all'insieme dei vettori $\mathbf{p}^0, \mathbf{p}^1, ..., \mathbf{p}^{k-1}$.

L'algoritmo del gardiente coniugato é dato da:

Input: $\mathbf{x}_0$, A, $\mathbf{b}$, maxiter, tol

- $\mathbf{r}^0 = \mathbf{p}^0 = \mathbf{b} - A\mathbf{x}_0$, $k = 0$
- WHILE $||\mathbf{r}^k|| > tol$ AND $k < maxiter$ DO
  1. $\mathbf{z}^k = A\mathbf{p}^k$
  2. $\alpha_k = \frac{(\mathbf{p}^k)^T\mathbf{r}^k}{(\mathbf{z}^k)^T\mathbf{p}^k}$
  3. $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k\mathbf{p}^k$
  4. $\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha_k\mathbf{z}^k$
  5. $\beta_k = \frac{(\mathbf{r}^{k+1})^T\mathbf{r}^{k+1}}{(\mathbf{r}^k)^T\mathbf{r}^k}$
  6. $\mathbf{p}^{k+1} = \mathbf{r}^{k+1} + \beta_k\mathbf{p}^k$
  7. $k = k + 1$
- END WHILE

Il metodo del gradiente coniugato converge velocemente a patto che il numero di condizionamento spettrale $\kappa_2(A) = \lambda_n/\lambda_1$ sia sufficientemente piccolo. Esso puó essere molto grande anche per matrici di piccole dimensioni e di conseguenza il numero di iterazioni diventa molto grande.

3 L'idea di precondizionare

L'idea del precondizionamento consiste nel cercare di ridurre il numero di condizionamento della matrice del sistema, premoltiplicandola per una matrice che in genere si indica con M^{-1}. Si ottiene cosí il sistema:

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}.$$

Ovviamente il precondizionamente é efficiente se $\kappa_2(M^{-1}A) << \kappa_2(A)$ e questa proprietá é verificata in particolare quando $M^{-1} \approx A^{-1}$.

3.1 Il metodo del gradiente coniugato precondizionato

Si dimostra che il sistema $M^{-1}Ax = M^{-1}b$ é equivalente a un sistema la cui matrice é simmetrica e definita positiva, in particolare si riesce a scrivere

$$B\mathbf{x}' = \mathbf{b}'$$

dove $\mathbf{x}' = X\mathbf{x}$, $\mathbf{b}' = X^{-1}\mathbf{b}$, $B = X^{-1}AX^{-1}$. É possibile scrivere ció se si considera M definita positiva, quindi diagonalizzabile e perció esiste una matrice definita positiva $X = Q^T D^{1/2} Q$ tale che $M = XX$.

Partendo da quest'ultimo sistema si vede che l'algoritmo del gradiente coniugato precondizionato é:

Input: $\mathbf{x}_0$, A, M, $\mathbf{b}$, maxiter, tol

- $\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}_0$, $\mathbf{p}^0 = M^{-1}\mathbf{r}^0$, $k = 0$
- WHILE $||\mathbf{r}^k|| > tol$ AND $k < maxiter$ DO
  1. $\mathbf{z}^k = A\mathbf{p}^k$
  2. $\alpha_k = \frac{(\mathbf{p}^k)^T\mathbf{r}^k}{(\mathbf{z}^k)^T\mathbf{p}^k}$
  3. $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k\mathbf{p}^k$
  4. $\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha_k\mathbf{z}^k$
  5. $\mathbf{g}^{k+1} = M^{-1}\mathbf{r}^{k+1}$
  6. $\beta_k = \frac{(\mathbf{r}^{k+1})^T M^{-1}\mathbf{r}^{k+1}}{(\mathbf{r}^k)^T M^{-1}\mathbf{r}^k}$
  7. $\mathbf{p}^{k+1} = \mathbf{g}^{k+1} + \beta_k\mathbf{p}^k$
  8. $k = k + 1$
- END WHILE

In Matlab si usa il comando:

```
>> [x] = pcg(A,b,tol,maxit,M)
```

E come si puó scegliere quindi il precondizionatore M^{-1}? Se la matrice A é simmetrica e definita positiva allora sono possibili di sicuro queste due scelte

a) precondizionatore di Jacoby: $M = D$, dove D é una matrice diagonale tale che

$$d_{ii} = a_{ii}$$

e questa scelta di solito basta per abbattere in numero di iterazioni del metodo del gradiente coniugato.

b) fattorizzazione incompleta di Cholesky: $M = LL^T$. Si tratta della ben nota fattorizzazione di Cholesky, ma lascia invariati gli elementi che nella matrice A sono uguali a zero, cioé

$$a_{ij} = 0 \Longrightarrow l_{ij} = 0.$$

In Matlab si usa in comando:

```
>> [L] = ichol(A,opt)
```

e poi, per risolvere il sistema usando il metodo del gardiente precondizionato si usa

```
>> [x] = pcg(A,b,tol,maxit,M1,M2)
```

con $M1 = L$ e $M2 = L^T$, infatti questo comando usa come precondizionatore $M = M1 * M2$.

3.2 Cosa si puó fare per le matrici quadrate non simmetriche?

Per le matrici non simmetriche é possibile fare la fattorizzazione LU incompleta, che funziona allo stesso modo della fattorizzazione incompleta di Cholesky. Per le fattorizzazioni incomplete si puó decidere se permettere o meno il *fill in*, cioé il passaggio di elementi della matrice A uguali a zero in elemento diversi da zero nelle matrici di fattorizzazione.

In Matlab si usa il comando:

```
>> [L,U,P] = ilu(A,setup).
```

Per quanto riguarda i metodi che si possono utilizzare ci concentreremo su due in particolare: il GMRES e il BICGSTAB. Per poterne parlare in modo un po' specifico é neccessario concentrarsi prima sui *sottospazi di Krylov*.

4 Metodi di proiezione su spazi di Krylov

Sia $A \in \mathbb{R}^{N\times N}$, $\mathbf{v} \in \mathbb{R}^N$, definiamo lo *spazio di Krylov*

$$\mathcal{K}_m(A, \mathbf{v}) = span\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, ..., A^{m-1}\mathbf{v}\} \subseteq \mathbb{R}^N \qquad m = 1, 2, 3, ...$$

I *metodi di proiezione su sottospazi di Krylov* determinano per ogni m una soluzione approssimata $\mathbf{x}^m$ del sistema lineare $A\mathbf{x} = \mathbf{b}$, che appartiene a $\mathcal{K}_m(A, \mathbf{r}^0)$, dove $\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$ é il vettore residuo iniziale e $\mathbf{x}^0$ é la stima iniziale della nostra soluzione.

Si possono applicare differenti criteri di ottimalitá sulla soluzione che conducono a differenti algoritmi.

Purtroppo la base di $\mathcal{K}_m(A, \mathbf{v})$ é numericamente instabile in quanto al crescere di m si ha che i vettori tendono a diventare linearmente dipendenti, infatti tendono all'autovettore dominante (metodo delle potenze). É anche per questo che sará necessario estrarre una base ortonormale dalla precedente base di Krylov.

Questi metodi, soprattutto se opportunamente precondizionati, forniscono una soluzione del sistema lineare in un numero di passi molto inferiore rispetto alla dimensione del sistema (N).

Questi metodi ottengono la soluzione approssimata $\mathbf{x}^m$ imponendo m vincoli e tipicamente vengono imposte m condizioni indipendenti di ortogonalitá sul vettore residuo. Queste sono definire implicitamente in un altro sottospazio $\mathcal{L}_m \subseteq \mathbb{R}^N$, generato dagli m vettori indipendenti utilizzati per l'ortogonalitá ($\dim(\mathcal{L}_m) = m$).

In generale si cerca una soluzione approssimante in $\mathcal{K}_m(A, \mathbf{r}^0)$, che ad ogni iterazione aumenta di dimensione, tale che il vettore residuo sia ortogonale a $\mathcal{L}_m$. In altre parole si cerca una soluzione $\mathbf{x}^m \in \mathbb{R}^N$ nel sottospazio affine $\mathbf{x}^0 + \mathcal{K}_m(A, \mathbf{r}^0)$, imponendo che

$$\mathbf{b} - A\mathbf{x}^m \quad \perp \quad \mathcal{L}_m$$

Come $\mathcal{L}_m$ si puó usare $\mathcal{L}_m = \mathcal{K}_m(A, \mathbf{r}^0)$ oppure $\mathcal{L}_m = A\mathcal{K}_m(A, \mathbf{r}^0)$ oppure lo spazio di Krylov associato ad A^T, cioé $\mathcal{L}_m = \mathcal{K}_m(A^T, \mathbf{r}^0)$.

4.1 Il metodo del gradiente coniugato come metodo di Krylov

Sia $A\mathbf{x} = \mathbf{b}$ con A matrice definita positiva, sia $\mathbf{x}^0 = \mathbf{0}$, allora il metodo del gardiente coniugato procede fino a che il residuo $\mathbf{r}^m$ non si annulla. Inoltre si ha che

$$\mathcal{L}_m = span\{\mathbf{p}^0, ..., \mathbf{p}^{m-1}\}$$

e valgono le solite relazioni di ortogonalitá tra i residui e di A-ortogonalitá tra i $\mathbf{p}^i$, $i = 0, ..., m-1$. Si dimostra anche che per quanto riguarda questo metodo si ha che la soluzione $\mathbf{x}^m$ minimizza la A-norma dell'errore, cioé

$$\mathbf{x}^m = \arg \min_{\mathbf{x} \in \mathcal{K}_m(A, \mathbf{r}^0)} ||\hat{\mathbf{x}} - \mathbf{x}||_A$$

dove $||\mathbf{y}||_A = \sqrt{\mathbf{y}^T A \mathbf{y}}$, $\mathbf{y} \in \mathbb{R}^N$.

4.2 Il metodo di Arnoldi

Per poter realizzare un metodo di proiezione é necessario poter imporre le condizioni di ortogonalitá a tutti gli elementi di una base dello spazio di Krylov $\mathcal{K}_m(A, \mathbf{r}^0)$. Non possiamo usare la base che era presente nella definizione perché i vettori tendono alla dipendenza lineare.

Il metodo di Arnoldi costruisce una base, a partire da una base assegnata del sottospazio di Krylov, tramite un processo di ortonormalizzazione (Grand-Schmidt modificato):

$$\begin{array}{rll} \mathbf{v}_1 & = \frac{\mathbf{r}^0}{||\mathbf{r}^0||} & \\ \tilde{\mathbf{v}}_{k+1} & = A\mathbf{v}_k - \sum_{i=1}^{k} h_{ik}\mathbf{v}_i & k = 1, ..., m \\ \mathbf{v}_{k+1} & = \tilde{\mathbf{v}}_{k+1}/||\tilde{\mathbf{v}}_{k+1}|| & k = 1, ..., m \end{array}$$

dove $h_{ik} = \mathbf{v}_i^T A \mathbf{v}_k$, $h_{k+1,k} = ||\tilde{\mathbf{v}}_{k+1}||$.

Un possibile algoritmo per implementarlo é:

Input: A, $\mathbf{r}^0$, $epsi$

- $\mathbf{v}_1 = \frac{\mathbf{r}^0}{||\mathbf{r}^0||}$
- $k = 1$
- WHILE $k \leq N$ AND $h_{k,k-1} < epsi$ DO
  1. $\mathbf{v} = A\mathbf{v}_k$
  2. FOR $i = 1, ..., k$
     (a) $h_{ik} = \mathbf{v}_i^T \mathbf{v}$
     (b) $\mathbf{v} = \mathbf{v} - h_{ik}\mathbf{v}_i$
  3. END FOR
  4. $h_{k+1,k} = ||\mathbf{v}||$
  5. $\mathbf{v}_{k+1} = \frac{\mathbf{v}}{h_{k+1,k}}$
  6. $k = k + 1$
- END WHILE

4.3 Il metodo Generalized Minimal Residual (GMRES)

Tale metodo determina la soluzione del sistema lineare $A\mathbf{x} = \mathbf{b}$ minimizzando la norma del residuo $\mathbf{r}^m = \mathbf{b} - A\mathbf{x}^m$, con

$$\mathbf{x}^0 + K_m = \mathbf{x}^0 + V_m\mathbf{y}$$

dove $\mathbf{x}^0$ é un vettore si stima iniziale , K_m é il spazio di Krylov generato del residuo iniziale $\mathbf{r}^0$, cioé $K_m = \mathcal{K}_m(A, \mathbf{r}^0)$ e $V_m \in \mathbb{R}^{N\times m}$ é la matrice le cui colonne formano una base per K_m.

Il metodo GMRES si preoccupa di fare due cose:

i) ortogonalizzare la base (tramite l'algoritmo di Arnoldi);

ii) minimizzare la norma del residuo.

I vettori ottenuti dal punto i) soddisfano la seguente relazione

$$A\mathbf{v}_s = \sum_{i=1}^{s+1} h_{is}\mathbf{v}_i \qquad s = 1, ..., m$$

che esprime l's-esima colonna del prodotto matriciale

$$A[\ \mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_m\ ] = [\ \mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_{m+1}\ ] \begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & \cdots & h_{1m} \\ h_{21} & h_{22} & h_{23} & \cdots & \cdots & h_{2m} \\ 0 & h_{32} & h_{33} & \cdots & \cdots & h_{3m} \\ 0 & 0 & h_{43} & \ddots & \vdots & h_{4m} \\ \vdots & & \vdots & \ddots & \ddots & \vdots \\ & & & & h_{m,m-1} & h_{mm} \\ 0 & \cdots & & \cdots & 0 & h_{m+1,m} \end{bmatrix}$$

che puó essere riscritto matricialmente cosí

$$\begin{array}{ccccc} A & V_m & = & V_{m+1} & H \\ (N\times N) & (N\times m) & & (N\times m+1) & (m+1\times m) \end{array}$$

inoltre si osservi che

$$V_m^T V_{m+1} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \cdots & 1 & 0 \end{bmatrix}_{m\times m+1} .$$

Il residuo che si vuole quindi minimizzare risulta quindi essere

$$\begin{aligned}\mathbf{r}^m &= \mathbf{b} - A\mathbf{x}^m \\ &= \mathbf{b} - A(\mathbf{x}^0 + V_m\mathbf{y}) \\ &= \mathbf{r}^0 - AV_m\mathbf{y} \\ &= \beta\mathbf{v}_1 - V_{m+1}H\mathbf{y} \\ &= V_{m+1}(\beta\mathbf{e}_1 - H\mathbf{y})\end{aligned}$$

dove $\beta = ||\mathbf{r}_0||$.

Siccome la matrice V_{m+1} é ortonormale si ha

$$||\mathbf{r}^m|| = ||\mathbf{b} - A(\mathbf{x}^0 + V_m\mathbf{y})|| = ||\beta\mathbf{e}_1 - H\mathbf{y}||$$

L'algoritmo GMRES vuole minimizzare tale norma, cioé cerca una soluzione approssimante $x^m = \mathbf{x}^0 + V_m\mathbf{y}^m$, dove

$$\mathbf{y}^m = \arg\min_{\mathbf{y}} ||\beta\mathbf{e}_1 - H\mathbf{y}||.$$

Un possibile algoritmo per implementarlo é:

Input: $\mathbf{x}^0$, A, $\mathbf{b}$, maxiter, tol

- $\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$, $\rho_0 = ||\mathbf{r}^0||$, $\mathbf{v}_1 = \frac{\mathbf{r}^0}{\beta}$
- WHILE $\rho_k > tol$ AND $k < maxiter$ DO
  1. calcolo V_m con il metodo di Arnoldi
  2. $\mathbf{y}^k = \arg\min_{\mathbf{y}} ||\beta\mathbf{e}_1 - H_k\mathbf{y}||$
  3. $\rho_k = ||\beta\mathbf{e}_1 - H_k\mathbf{y}^k||$
- END WHILE
- $\mathbf{x}^k = \mathbf{x}^0 + V_k\mathbf{y}^k$ (aggiornamento di $\mathbf{x}^k$)

In Matlab si usa il comando:

```
>> [x] = GMRES(A,b,restart,tol,maxit)
```

Il metodo GMRES é ottimo nel senso che minimizza il residuo su un sottospazio di dimensione crescente (proprietá di terminazione finita), ma

1. é molto costoso: é necessario mantenere in memoria tutti i vettori della base di Krylov e tutta la matrice H_m;
2. il costo della ortogonalizzazione di Gram-Schmidt cresce all'aumentare del numero di iterazioni.

Nelle applicazioni si definisce a priori un numero massimo p di vettori (comando 'restart') che possono essere memorizzati. Dopo p iterazioni si calcola un'approssimazione $\mathbf{x}^p$ della soluzione, si cancellano tutti i vettori dello spazio di Krylov e si riparte con un nuovo spazio di Krylov avente come vettore generatore $\mathbf{r}^p = \mathbf{b} - A\mathbf{x}^p$ (si perde la proprietá di ottimo).

Si puó precondizionare anche il metodo GMRES in Matlab:

>> [x] = GMRES(A,b,restart,tol,maxit,M1,M2)

4.4 Il metodo del gradiente biconiugato stabilizzato (BICGSTAB)

É un altro metodo implementato per matrici asimmetriche che si basa sul metodo del gradiente biconiugato (BICG), la cui convergenza é stabilizzata.

>> [x] = BICGSTAB(A,b,tol,maxit)

>> [x] = BICGSTAB(A,b,tol,maxit,M1,M2)

Il metodo BICG é basato sul metodo CG, ma é formulato per matrici non simmetriche. Esso costruisce due set di vettori, partendo da un certo $\hat{\mathbf{r}}^0$ non ortogonale a $\mathbf{r}^0$, in questo modo

$$\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m \in \mathcal{K}_m(A, \mathbf{r}^{(0)})$$
$$\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m \in \mathcal{L}_m(A^T, \hat{\mathbf{r}}^{(0)})$$

tali che

$$\mathbf{w}_i^T \mathbf{v}_j = \delta_{ij}$$

dove δ_{ij} é la delta di Kronecker.

Quindi si usano due sequenze di residui basate sulla matrice A e sulla matrice A^T

$$\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha_{k+1} A\mathbf{p}^{k+1}, \quad \hat{\mathbf{r}}^{k+1} = \hat{\mathbf{r}}^k - \alpha_{k+1} A^T \hat{\mathbf{p}}^{k+1}$$

le cui direzioni di discesa sono

$$\mathbf{p}^{k+1} = \mathbf{r}^k + \beta_k A\mathbf{p}^k, \quad \hat{\mathbf{p}}^{k+1} = \hat{\mathbf{r}}^k + \beta_k A^T \hat{\mathbf{p}}^k$$

con

$$\alpha_{k+1} = \frac{(\hat{\mathbf{r}}^k)^T \mathbf{r}^k}{(\hat{\mathbf{p}}^{k+1})^T A\mathbf{p}^{k+1}},$$
$$\beta_k = \frac{(\hat{\mathbf{r}}^k)^T \mathbf{r}^k}{(\hat{\mathbf{r}}^{k-1})^T \mathbf{r}^{k-1}}.$$

Inoltre sono verificate le seguenti relazioni di ortogonalitá

$$(\hat{\mathbf{r}}^k)^T \mathbf{r}^j = (\hat{\mathbf{p}}^k)^T A\mathbf{p}^j = 0, \quad k \neq j.$$