

Fondamenti di Informatica, A.A. 2011-2012

13/02/2013

Prova Pratica

È data una matrice quadrata di dimensione arbitraria. La matrice si assume *sparsa*, ossia contiene molti elementi nulli. Si richiede di convertire la matrice nel formato ELLPACK, costituito compattando gli elementi diversi da zero:

- Gli elementi non nulli vengono memorizzati in una matrice rettangolare;
- La matrice rettangolare ha un numero di colonne sufficiente a contenere le righe che abbiano il maggior numero di elementi non zero;
- Le righe che contengono un numero di nonzeri inferiore al massimo vengono riempite nelle posizioni rimanenti con degli zeri;
- Parallelamente alla matrice dei coefficienti si memorizza una matrice di interi di uguale dimensione che contiene in corrispondenza della posizione di un certo elemento non zero, l'indice di colonna a cui quell'elemento si trovava nella matrice originaria. Le righe vengono poi riempite con valori 1.

Ad esempio la matrice

$$A = \begin{bmatrix} 1.5 & 2.5 & 0 & 0 & 3 \\ 2.5 & 1.5 & 2.5 & 0 & 0 \\ 0 & 3.5 & 4 & 0 & 0 \\ 0 & 0.5 & 0 & 3 & 2 \\ 0 & 5.5 & 0 & 0 & 0 \end{bmatrix}$$

viene rappresentata con la coppia

$$KA = \begin{bmatrix} 1.5 & 2.5 & 3 \\ 2.5 & 1.5 & 2.5 \\ 3.5 & 4 & 0 \\ 0.5 & 3 & 2 \\ 5.5 & 0 & 0 \end{bmatrix}$$

$$JA = \begin{bmatrix} 1 & 2 & 5 \\ 1 & 2 & 3 \\ 2 & 3 & 1 \\ 2 & 4 & 5 \\ 2 & 1 & 1 \end{bmatrix}$$

Si scriva una funzione Matlab che data una matrice A produca la sua rappresentazione KA, JA .

Soluzione

La soluzione del problema assegnato dovrà passare attraverso le fasi:

1. Identificazione degli elementi non-zero della matrice di input;
2. Calcolo del massimo numero di elementi non-zero per riga;
3. Allocazione e inizializzazione delle matrici di output;
4. Copia dei coefficienti dalla matrice di input nelle matrici di output, e copia delle relative posizioni

Una possibile soluzione, molto semplice, è la seguente

```
function [ka , ja]=dns2ells(a)
    % Convert a dense matrix to ELLPACK format
    [nr , nc] = size(a);

    % Compute max row occupancy
    nce = 0;
    for i=1:nr
        k=0;
        for j=1:nc
            if (a(i , j) ~= 0)
                k = k + 1;
            end
        end
        if (k>nce)
            % Maximum so far
            nce=k;
        end
    end

    % Initialize output
    ka = zeros(nr , nce);
    ja = ones(nr , nce);
    % Copy
    for i=1:nr
        k = 0;
        for j=1:nc
            if (a(i , j) ~= 0)
                k = k + 1;
                ka(i , k) = a(i , j);
                ja(i , k) = j;
            end
        end
    end
end
```

Una soluzione che fa uso estensivo degli operatori su array e quindi risulta più compatta ma raggiunge lo stesso risultato è:

```

function [ka,ja]=dns2ell(a)
    % Convert a dense matrix to ELLPACK format
    [nr,nc] = size(a);

    % Prepare index array
    ix = 1:nc;
    % Locate nonzeros
    xa = (a~=0);
    % Compute row occupancies
    nrv = sum(xa,2);
    % Max nonzero per row
    nce = max(nrv);
    ka = zeros(nr,nce);
    ja = ones(nr,nce);
    for i=1:nr
        nci = nrv(i);
        ka(i,1:nci) = a(i,xa(i,:));
        ja(i,1:nci) = ix(xa(i,:));
    end

```