Treball 1: Matrius disperses

Ismael El Habri, Marc Cané, Lluís Trilla

16 d'octubre de 2018

Índex

Capítol 1

Què són les matrius disperses?

Quan parlem de matrius disperses ens referim a matrius de gran tamany en la qual la majoria d'elements son zero. Direm que una matriu és disperrsa, quan hi hagi benefici en aplicar els mètodes propis d'aquestes.

Per identificar si una matriu és dispersa, podem usar el seguent:

> Una matriu $n \times n$ serà dispersa si el número de coeficients no nuls es $n^{\gamma+1}$, on $\gamma < 1$.

En funció del poblema, decidim el valor del paràmetre γ. Aquí hi ha els valors típics de γ:

- $\gamma = 0.2$ per problemes d'anàlisi de sistemes eléctics degeneració i de transpot d'enegía.
- $\gamma = 0.5$ per matrius en bandes associades a problemes d'anàlisi d'estructues.

Podem trobar dos tipus de matrius disperses:

- **Matrius estructurades:** matrius en les quals els elements diferents de zero formen un patró regular. Exemple: Les matrius banda.
- **Matrius no estructurades:** els elements diferents de zero es distribueixen de forma irregular.

Capítol 2

Formes d'emmagatzemar matrius disperses

2.1 Per Coordenades

És la primera aproximació que podríem pensar i és bastant intuïtiva. Per cada element no nul guardem una tupla amb el valor i les seves coordenades: (a_{ij}, i, j).

2.1.1 Exemple

$$\begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 3 & 0 & -2 & 0 \end{pmatrix} = \begin{array}{l|c} \text{índex} & \text{tupla}(a_{ij}, i, j) \\ \hline 0 & (1,0,0) \\ 1 & (2,0,3) \\ 2 & (1,1,1) \\ 3 & (1,3,0) \\ 4 & (-2,3,2) \end{array}$$

Per emmagatzemar això podem usar tres vectors de la mateixa mida (n_z, el nombre d'elements diferents de zero): Un amb els valors, un amb les files i un amb les columnes:

Vector	Coeficients				
valors	1	2	1	1	-2
files	0	0	1	3	3
columnes	0	3	1	0	2

A la realitat però, aquest mètode d'emmagatzemar les dades és poc eficient quan hem de fer operacions amb les matrius.

2.2 Per files

També conegut com a *Compressed Sparse Rows (CSR)*, *Compressed Row Storage (CSR)*, o format *Yale*. És el mètode més estès.

Consisteix en guardar els elements ordenats per files, guardar la columna on es troben, i la posició del primer element de cada fila en el vector de valors. Així ens quedaran tres vectors:

- **valors:** de mida n_z, conté tots els valors diferents.
- **columnes:** també de mida n_z, conté la columna on es troba cada un dels elements anteriors.
- **iniFiles:** de mida $m+1$, conté la posició on comença cada fila en els vectors valors i columnes, sent m el nombre de files de la matriu.

2.2.1 Exemple

$$\begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 3 & 0 & -2 & 0 \end{pmatrix} =$$

Vector	Coeficients				
índex	0	1	2	3	4
valors	1	2	1	1	−2
columnes	0	3	1	0	2
iniFiles	0	2	3		

Si es canvien files per columnes, dona la implementació per columnes, o també anomenada *Compressed Sparse Columns (CSC)*.

2.2.2 Implementació del mètode CSR

Aquí va la implementació i tot el relacionat

2.3 Per perfil

Aquest mètode és una manera eficient de guardar un tipus concret de matrius, les matrius banda.

2.3.1 Matrius banda

Com vam veure a classe, una matriu $n \times n$ és banda si existeixen p i q naturals, tals que $1 < p, q < n$ i $a_{ij} = 0$ sempre que $p \leq j - i$ o $q \leq i - j$.

Anomenem ample de banda de la matriu a $p + q - 1$.

L'**envoltant** de una matriu banda consisteix en tots els elements de cada fila des de el primer no nul fins al ultim no nul, incloent els elements nuls que hi pugui haver entre mig. Aquí la definició formal:

L'envoltant de una matriu banda A, $env(A)$ es defineix com el conjunt $env(A) = \{i, j\} : f_i \leq j \leq l_i, 1 \leq i \leq n$

on f_i és on comencen els valors no nuls en la fila i, i l_i on acaben els valors no nul en la fila i.

2.3.2 El mètode

Consisteix en guardar els elements de l'envoltant de la matriu, la columna on comença l'envoltant en cada fila, i la posició en el vector de valors on comença cada fila.

Doncs, ens quedarien tres vectors:

- **valors:** amb els valors de l'envoltant.
- **columnInici:** amb el valor f_i de cada fila (la columna on comença l'envoltant en cada fila.
- **iniFiles:** amb la posició on comença cada fila en el vector valors.

2.3.3 Exemple

$$\begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & -2 \end{pmatrix} =$$

Vector	Coeficients						
índex	0	1	2	3	4	5	6
valors	1	2	1	2	0	2	-2
columnInici	0	1	1	3			
iniFiles	0	2	3	6			

2.4 Altres mètodes

2.4.1 Diccionari de claus

Conegut també com a *Dictionary of Keys (DOK)*, consisteix en tenir un diccionari que mapeja parelles de (fila, columna) amb el valor de cada element. Els elements que no estan en el diccionari es poden considerar zero. Aquest mètode es bo per construir la matriu de manera incremental en un ordre aleatori, però es dolent al iterar pels elements diferents de zero en un ordre lexicogràfic. El seu ús més habitual es usar aquest format per construir la matriu, per després convertir-la en un format més eficient de processar.

2.4.2 Llista de llistes

Guarda una llista per fila, en la qual cada entrada és una parella de valors (valor, columna). S'acostumen a ordena pel número de columna per motius d'eficiència. Aquest mètode també es bo per construir la matriu de forma incremental.

2.4.3 Esquema DIA

Aquest esquema s'usa quan els valors no nuls estan restringits a un reduït nombre de diagonals. Consisteix en guardar una matriu de dades que conté els valors no nuls i un vector amb els *offsets*, que guarda els *offsets* de cada diagonal respecte la diagonal principal.

La diagonal principal li correspon l'*offset* 0. A les diagonals superiors, les hi assignem valors positius; les inferiors, valors negatius.

Exemple

Donada la matriu:

$$\begin{pmatrix} 1 & 7 & 0 & 0 \\ 0 & 2 & 8 & 0 \\ 5 & 0 & 3 & 9 \\ 0 & 6 & 0 & 4 \end{pmatrix}$$

Necessitaríem guardar el següent:

$$dat = \begin{pmatrix} * & 1 & 7 \\ * & 2 & 8 \\ 5 & 3 & 9 \\ 6 & 4 & * \end{pmatrix}, off = (-2, 0, 1)$$