Matrices sparse

J. J. Cendán Verdes

Departamento de Matemáticas, Universidade da Coruña

jesus.cendan.verdes@udc.es

Matrices sparse

- Generalidades
- Almacenamiento
- Formatos: coordenadas, CSR y CSC. Almacenamiento perfil
- Operaciones
- Representación gráfica
- Reordenamientos

Aplicaciones

Computational fluid dynamics, finite-element methods, statistics, time/frequency domain circuit simulation, dynamic and static modeling of chemical processes, cryptography, electrical power systems, differential equations, quantum mechanics, structural mechanics (buildings, ships, aircraft, human body parts...), heat transfer, MRI reconstructions, vibroacoustics, linear and non-linear optimization, financial portfolio optimization, semiconductor process simulation, economic modeling, astrophysics, crack propagation, Google page rank, 3D computer vision, cell phone tower placement, tomography, multibody simulation, nano-technology, acoustic radiation, elastic properties of crystals, natural language processing, DNA electrophoresis,...

.

Matrices sparse

Son aquellas matrices que tienen solamente unos pocos elementos no nulos. En la práctica, según Wilkinson, aquellas que **pueden tomar ventaja** de la gran cantidad de ceros que tienen. Surgen a menudo en la resolución de EDP,
Varias denominaciones: matrices sparse, huecas o dispersas.

- Estructuradas. Los elementos no nulos siguen un patrón a menudo, a lo largo de un pequeño número de diagonales.
- No estructuradas.

.

Matrices sparse. Ventajas

Ventajas del manejo de matrices sparse frente a matrices llenas

- Almacenamiento.
- Tiempo computacional.
- Menor consumo de energía y de uso del procesador

	sparse	llena
Memoria	$0{,}25Mb.$	$128Mb.$
Calcular SX	$0{,}2s.$	$30s.$
Resolver SX=b	$10s.$	$> 12horas$

Operaciones con un laplaciano discretizado. Gilbert-Moler

.

Matrices sparse. Almacenamiento

Los esquemas de almacenamiento tratan de explotar las características de las matrices dispersas y almacenar únicamente sus elementos no nulos.
La librería SPARSKIT dispone de varios formatos:

- COO: Formato de coordenadas
- CSR: Comprimido por filas (Compressed Sparse Row)
- CSC: Comprimido por columnas
- Almacenamiento perfil
- Otros formatos: DNS formato denso, BND adecuado para matrices tipo banda, ELL Ellpack-Itpack diagonal generalizado, DIA diagonal, BSR matrices por bloques,...

Formato de almacenamiento por coordenadas

$$A = \begin{pmatrix} 1 & 0 & 0 & 2 & 0 \\ 3,1 & 4 & 0 & 5 & 0 \\ -6,2 & 0 & 7 & 8 & -9,1 \\ 0 & 0 & 10,3 & 11 & 0 \\ 0 & 0 & 0 & 0 & 12,3 \end{pmatrix}$$

.

.

$$AA = \begin{pmatrix} 12,3 & \text{-9,1} & 7 & 5 & 1 & 2 & 11 & 3.1 & \text{-6.2} & 4 & 8 & 10.3 \end{pmatrix}$$

.

$$JR = \begin{pmatrix} 5 & 3 & 3 & 2 & 1 & 1 & 4 & 2 & 3 & 2 & 3 & 4 \end{pmatrix}$$

$$JC = \begin{pmatrix} 5 & 5 & 3 & 4 & 1 & 4 & 4 & 1 & 1 & 2 & 4 & 3 \end{pmatrix}$$

- AA: contiene los valores **no nulos** de la matriz A. Su longitud es nnz (número de elementos no nulos de A)
- JR: índices de fila de los elementos de AA
- JC: índices de columna de los elementos de AA

Formato comprimido por filas CSR

$$A = \begin{pmatrix} 1 & 0 & 0 & 2 & 0 \\ 3{,}1 & 4 & 0 & 5 & 0 \\ -6{,}2 & 0 & 7 & 8 & -9{,}1 \\ 0 & 0 & 10{,}3 & 11 & 0 \\ 0 & 0 & 0 & 0 & 12{,}3 \end{pmatrix}$$

.

.

$$AA = \begin{pmatrix} 1 & 2 & 3.1 & 4 & 5 & \text{-6.2} & 7 & 8 & \text{-9,1} & 10.3 & 11 & 12{,}3 \end{pmatrix}$$

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

$$IA = \begin{pmatrix} 1 & 3 & 6 & 10 & 12 & 13 \end{pmatrix}$$

- AA: contiene los valores **no nulos** de A introducidos por filas. Su longitud es nnz (número de elementos no nulos de A)
- JA: índices de columna de los elementos de AA (su longitud es nnz)
- IA: contiene los índices del inicio de cada fila en la tabla AA. Su longitud es n+1. Además $IA(n+1) = IA(1) + nnz$.

Formato comprimido por filas CSR

.

- Es el almacenamiento más popular para matrices sparse de estructura irregular
- Su esquema es ventajoso por su simplicidad y flexibilidad
- En muchos paquetes de software es el formato inicial para las matrices sparse
- Variantes: CSC, formato comprimido por columnas y MSR, el cual explota el hecho de que muchas matrices los elementos de la diagonal son no nulos.

Formato comprimido por columnas CSC

$$A = \begin{pmatrix} 1 & 0 & 0 & 2 & 0 \\ 3,1 & 4 & 0 & 5 & 0 \\ -6,2 & 0 & 7 & 8 & -9,1 \\ 0 & 0 & 10,3 & 11 & 0 \\ 0 & 0 & 0 & 0 & 12,3 \end{pmatrix}$$

.

$$AA = \begin{pmatrix} 1 & 3.1 & \text{-}6.2 & 4 & 7 & 10.3 & 2 & 5 & 8 & 11 & \text{-}9.1 & 12.3 \end{pmatrix}$$

.

$$JR = \begin{pmatrix} 1 & 2 & 3 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 3 & 5 \end{pmatrix}$$

$$IC = \begin{pmatrix} 1 & 4 & 5 & 7 & 11 & 13 \end{pmatrix}$$

- AA: contiene los valores no nulos de la matriz A **por columnas**. Su longitud es nnz (número de elementos no nulos de A)
- JR: índices de fila de los elementos de AA (su longitud es nnz)
- IC: contiene los índices del inicio de cada columna en la tabla AA. Su longitud es n+1. Además $IC(n+1) = IC(1) + nnz$.

Matrices sparse. Almacenamiento perfil

Por simplicidad, suponemos A matriz simétrica y describimos el almacenamiento perfil almacenando únicamente la parte triangular inferior (o superior) de la matriz (incluyendo la diagonal).

$$A = \begin{pmatrix} 3 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ -1 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 1 & 0 & 0 & 8 \end{pmatrix}$$

:

$$AA = \begin{pmatrix} 3 & 1 & -1 & 0 & 3 & 5 & 1 & 0 & 0 & 8 \end{pmatrix}$$
$$diag = \begin{pmatrix} 1 & 2 & 5 & 6 & 10 \end{pmatrix}$$

Para A simétrica, denotamos k_i la columna del primer elemento no nulo de la fila i. Así $A(i,j) = 0$ si $1 \leq j < k_i$. Definimos

$$perfil(A) = \{(i,j), 1 \leq i \leq n, k_i \leq j \leq i\}$$

AA: por filas son los coeficientes de A cuyos índices son del perfil.
diag(i): indica la posición de A(i,i) en AA.
Nota. $A = LDL'$ entonces $L + L'$ y A tienen el mismo perfil.

Matrices sparse. Almacenamiento mediante diagonales

En ocasiones, los elementos no nulos se agrupan en la diagonal principal y en paralelas a la misma. Conocida esta estructura almacenamos únicamente los vectores que conforman esas diagonales, donde se encuentran todos los elementos no nulos de la matriz.

Un claro ejemplo es el de una matriz tridiagonal:

$$A = \begin{pmatrix} 3 & -1 & 0 & 0 & 0 \\ 6 & 1 & 7 & 0 & 0 \\ 0 & -5 & 3 & 0 & 0 \\ 0 & 0 & 6 & 5 & 8 \\ 0 & 0 & 0 & -2 & 8 \end{pmatrix}$$

Matrices sparse. Operaciones

Los distintos almacenamientos tipo sparse nos permiten mejorar los tiempos de cálculo. Una de las operaciones más comunes en el manejo de matrices es el producto matriz por vector. Dado que la mayoría de elementos de la matriz son nulos evitamos operar con esos elementos según el formato de almacenamiento elegido.
Por ejemplo si A está almacenada en formato CSR el producto AX lo hacemos siguiendo el esquema siguiente:

Inicio del bucle $i = 1, n$

$$i1 = IA(i)$$

$$i2 = IA(i+1) - 1$$

$$Y(i) = \text{producto escalar } (A(i1:i2), X(JA(i1:i2))$$

Fin del bucle

Nota. Se puede ver una comparación de costes por ops y tiempo de ejecución en: http://jd.jllo.net/works/Cuaderno40.pdf

Matrices sparse. EJERCICIOS

1. Dada la matriz:

$$A = \begin{pmatrix} -1 & 4 & 0 & 1 \\ 0 & 0 & 5 & 0 \\ -4 & 3 & 0 & 0 \\ 2 & 0 & 0 & 30 \end{pmatrix}$$

   - Almacena la matriz en los formatos coordenado, CSR y CSC.
   - Si cambiamos el coeficiente $a_{34} = 6$ calcula de nuevo los almacenamientos.
   - Sea B la matriz de orden 5 que se obtiene al añadir una fila y una columna de ceros en A y haciendo $b_{55} = 9$. Almacena B en los tres formatos.

2. Producto matriz vector. Sea $v = (1,2,3,4)$. Realiza el producto Av con los tres almacenamientos de A.
3. Almacena en formato perfil la matriz simétrica $C = AA'$.
4. Explica el almacenamiento ELL. Pon 2 ejemplos

Matrices sparse. Representación gráfica

La teoría de grafos es una herramienta útil para representar la estructura de una matriz sparse, ayudando en la búsqueda de precondicionadores y algoritmos paralelos. Un grafo está definido por dos conjuntos, un conjunto de vértices:

$$V = \{v_1, v_2, ..., v_n\}$$

y un conjunto de lados E formado por pares (v_i, v_j), donde v_i, v_j son elementos de V. El grafo $G = (V, E)$ se representa por un conjunto de puntos del plano unidos por una línea si los puntos están conectados por un lado. El grafo puede verse como una relación binaria entre objetos de un conjunto V.
Por ejemplo si V es el conjunto de ciudades de Europa podemos dibujar un lado uniendo dos ciudades si existe un vuelo directo entre ambas.

.

Nota. [Saad] www.stanford.edu/class/cme324/saad.pdf

Matrices sparse. Representación gráfica

El grafo $G = (V, E)$ asociado a una matriz sparse está formado por las n incógnitas como vértices. Los lados representan las relaciones binarias establecidas por las ecuaciones del modo siguiente:
Hay un lado desde el nodo i al nodo j si $a_{ij} \neq 0$.
El lado nos indica que en la ecuación i interviene la incógnita j.
Notemos que el patrón que representa el grafo es simétrico si verifica:

$$a_{ij} \neq 0 \Longleftrightarrow a_{ji} \neq 0$$

En el manejo se las matrices huecas, como por ejemplo en la resolución de algunos sistemas de ecuaciones asociados EDP, cada incógnita se corresponde con un nodo de la malla empleada en la discretización. La velocidad de convergencia de un algoritmo iterativo puede depender del orden empleado. Además si empleamos métodos directos se pueden llenar posiciones que son nulas en la matriz inicial.

Matrices sparse. Representación gráfica

Ejemplo. Matrices estrella. [Saad ej. 3.3]

$$\begin{pmatrix} x & x & x & x & x \\ x & x & & & \\ x & & x & & \\ x & & & x & \\ x & & & & x \end{pmatrix}$$

La matriz reordenando tiene un significativo impacto en algoritmos como el de Gauss frente a:

$$\begin{pmatrix} x & & & & x \\ & x & & & x \\ & & x & & x \\ & & & x & x \\ x & x & x & x & x \end{pmatrix}$$

Matrices sparse. Reordenamientos

Permutar filas y columnas es una de las operaciones más comunes con matrices sparse, empleado por ejemplo en computación paralela. Los grafos de las matrices nos permiten observar el efecto de los algoritmos de reordenación.

- Cuthill-McKee
  Se inicia en un punto continuando por sus vecinos y así progresivamente. Presenta diversas variantes [Saad]
- Colores
  Basado en el conocido problema de pintar un mapa de modo que dos países fronterizos no tengan el mismo color.
- Mínimo grado
  Ordena las filas de la matriz según el número de elementos no nulos. Es el más empleado para evitar el llenado en el proceso de eliminación de Gauss, especialmente en el caso de matrices simétricas y definidas positivas.

Reordenamiento de Cuthill-McKee

Para simplificar suponemos que la matriz es simétrica. Construimos la denominada matriz de adyacencia que pensando en grafos calcula para cada nodo el grado de incidencia. Construimos una n-tupla ordenada R de vértices que supone un reordenamiento de los mismos. Para ello, ordenamos según el grado de incidencia los nodos, partimos de uno de los nodos de menor grado x y llamamos $R = x$. Iterando:

- Para i=1,2,...
  Construimos el conjunto de adyacencia A_i excluyendo los nodos contemplados en pasos anteriores
- Ordenamos A_i siguiendo una ordenación ascendente de vértices
- Insertamos A_i al conjunto resultado R.

En resumen, numeramos los vértices de acuerdo a una particular búsqueda en anchura transversal, donde los vértices vecinos se visitan de menor a mayor.

Colecciones de matrices sparse

Existen numerosas colecciones de matrices sparse (dispersas) que nos permiten obtener ejemplos de estas matrices atendiendo a su construcción y a sus propiedades entre otras.
Además exportan la matriz en diferentes formatos como fortran, C y matlab.
The University of Florida Sparse Matrix Collection
http://www.cise.ufl.edu/research/sparse/matrices/index.html
http://math.nist.gov/MatrixMarket/

Ejercicio 5. Analiza colecciones de matrices sparse (formatos, tipos...) y elabora una tabla comparando las diferentes opciones que presentan.