

PROYECTO FINAL DE FUNDAMENTOS DE ANÁLISIS DE ALGORITMOS

Prof: Ing. Jorge Alberto Correa G.

UNIVALLE – SEDE NORTE DEL CAUCA

Fecha de Entrega: Miércoles 30 de Mayo de 2018

Este proyecto final pueden realizarlo en grupos máximo de dos personas. Deberán entregar, al igual que los trabajos presentados a lo largo del curso, un CD o DVD con la documentación correspondiente al proyecto e igualmente subir al campus virtual un archivo comprimido con el proyecto. Explicar de manera clara y concisa cómo desarrollaron los algoritmos correspondientes e incluir en ellos el código para el cálculo de tiempo de ejecución que deberá mostrarse en nanosegundos. En un archivo pdf deberán mostrar el código fuente respectivo con sus comentarios de cada rutina con el fin de facilitar el entendimiento del mismo.

No olvidar especificar los nombres y códigos de los integrantes del grupo.

Muy importante que, a pesar de que consultando en internet, pueden encontrar los programas que hagan la implementación del proyecto, sean ustedes quienes desarrollen su propio código lo cuál será comprobado.

MATRICES DISPERSAS O ESPARCIDAS

En muchas ocasiones el procesamiento computacional de grandes volúmenes de datos, donde una gran parte de estos datos son ceros o valores nulos, se convierte en una tarea muy compleja y en ocasiones imposible de realizar. Este análisis demanda gran cantidad de recursos computacionales, tanto en recursos de hardware como recursos de cómputo.

Las matrices dispersas son matrices donde una gran parte de sus valores son ceros, por lo cual, estos valores pueden ser omitidos en su almacenamiento tanto en memoria física como en los dispositivos de almacenamiento. Este tipo de matrices es muy empleado en problemas de modelación de grafos, permitiendo la representación de grandes volúmenes de información. En el campo de la inteligencia artificial las matrices dispersas han sido muy empleadas debido a la cantidad de información que en ocasiones procesan los algoritmos de aprendizaje automático.

Uno de los elementos a tener en cuenta cuando se trabaja con matrices dispersas es la estructura de datos que se emplea para su representación, así como la implementación de las operaciones sobre las matrices. Estas operaciones deben ser implementadas de forma específica para la estructura de datos que se emplea.

Las matrices son estructuras de datos que en general son muy eficientes en cuanto al uso de memoria y de procesador. Sin embargo, la representación computacional de las matrices dispersas mediante el enfoque general tiene como consecuencia un uso ineficiente de la memoria y del tiempo de procesamiento. Esto se debe a que se utilizan muchas posiciones de memoria para almacenar ceros y se requiere procesar todas las posiciones de la matriz innecesariamente. En consecuencia, teniendo como objetivo mejorar la complejidad espacial y temporal de las operaciones y de la implementación específica de matrices dispersas, se han propuesto diversas representaciones. De esta manera, las matrices dispersas pueden ser definidas como una estructura de datos. El propósito de dichas representaciones es almacenar únicamente la información relevante de cada fila y columna de la matriz original con el fin de utilizar menos memoria y agilizar las operaciones. En la siguiente sección se describen los principales enfoques de representación de matrices dispersas.

0	2	0	0	0	0	4
0	8	9	0	0	1	0
0	0	0	3	0	0	0
0	0	0	0	0	0	0
5	0	0	0	0	6	0
1	2	0	0	0	0	0
4	0	0	0	0	0	0
0	0	7	0	0	11	0

Cuadro 1: Ejemplo de matriz dispersa

REPRESENTACIONES

En las siguientes secciones se describirán los enfoques de representación más comunes para matrices dispersas. En adelante, se utilizará **n** y **m** para denotar el número de las y columnas de la matriz completa, en tanto que se utilizará **ne** para denotar el número de elementos diferentes de cero. Además, cada una de las representaciones será ilustrada utilizando como referencia la matriz dispersa del cuadro 1.

. FORMATO COORDENADO

La representación de matrices dispersas mediante el formato coordenado almacena los datos distintos de cero de la matriz dispersa junto con su ubicación en la matriz, i.e., el índice de su fila y su columna. Para esto se utilizan 3 vectores de tamaño **ne**. En el primer vector se almacenan los valores distintos de cero de la matriz original mientras que en el segundo y tercer vector se almacenan los índices de la fila y la columna para cada uno de los valores distintos de cero.

Los siguientes serían los valores en los vectores para la representación en formato coordenado de la matriz de ejemplo son:

valores = [2 4 8 9 1 3 5 6 1 2 4 7 11]

filas = [0 0 1 1 1 2 4 4 5 5 6 7 7]

columnas = [1 6 1 2 5 3 0 5 0 1 0 2 5]

. FORMATO COMPRIMIDO POR FILA (CSR)

El formato comprimido por fila (CSR) representa una matriz dispersa por medio de 3 vectores. El primer vector tiene tamaño **ne** y almacena los valores distintos de cero de la matriz original organizados fila por fila. En el segundo vector, que también tiene tamaño **ne**, se almacenan los índices de las columnas en las que están cada uno de los valores del primer vector en la matriz original. En tanto que, el tercer vector tiene tamaño $n + 1$ y almacena la posición donde empiezan los valores de cada fila en el segundo vector (vector de columnas).

Para el caso del ejemplo se tiene:

valores = [2 4 8 9 1 3 5 6 1 2 4 7 11]

cols = [1 6 1 2 5 3 0 5 0 1 0 2 5]

cfilas = [0 2 5 6 6 8 10 11 13]

Observe que el tercer vector (vector **cfilas**) se puede interpretar asociando el valor **ci** y el valor $c_{i+1} - 1$ como el índice inicial y el índice final en el vector de columnas asociados a la fila *i*. De esta manera, para el ejemplo anterior se tiene:

Fila	Límites	Columnas asociadas
0	cfilas[0] a cfilas[1]-1	cols[0], cols[1]
1	cfilas[1] a cfilas[2]-1	cols[2], cols[3], cols[4]
2	cfilas[2] a cfilas[3]-1	cols[5]
3	cfilas[3] a cfilas[4]-1	-
4	cfilas[4] a cfilas[5]-1	cols[6], cols[7]
5	cfilas[5] a cfilas[6]-1	cols[8], cols[9]
6	cfilas[6] a cfilas[7]-1	cols[10]
7	cfilas[7] a cfilas[8]-1	cols[11], cols[12]

Es importante resaltar que en esta representación cuando una fila no contiene ningún valor distinto a cero entonces el valor asociado a dicha fila en el tercer vector (vector **cfilas**) es el mismo

de la fila anterior. Además, el ultimo valor en el tercer vector corresponde al último índice valido en el segundo vector más 1.

. FORMATO COMPRIMIDO POR COLUMNA (CSC)

El formato comprimido por columna (CSC) también representa una matriz dispersa por medio de 3 vectores. Este formato es análogo al formato CSR y utiliza también 3 vectores. El primer vector tiene tamaño **ne** y almacena los valores no nulos en la matriz original organizados por columna. El segundo vector tiene tamaño **ne** y almacena los índices de las filas en las que están cada uno de los valores del primer vector. Mientras que el tercer vector tiene tamaño **m+1** y almacena la posición donde empiezan los valores de cada columna en el segundo vector (vector de filas). Para el caso del ejemplo la representación es:

`valores = [5 1 4 2 8 2 9 7 3 1 6 11 4]`

`filas = [4 5 6 0 1 5 1 7 2 1 4 7 0]`

`ccolumnas = [0 3 6 8 9 9 12 13]`

Fila	Límites	Columnas asociadas
0	ccolumnas[0] a ccolumnas[1]-1	filas[0], filas[1] filas[2]
1	ccolumnas[1] a ccolumnas[2]-1	filas[3], filas[4], filas[5]
2	ccolumnas[2] a ccolumnas[3]-1	filas[6], filas[7]
3	ccolumnas[3] a ccolumnas[4]-1	filas[8]
4	ccolumnas[4] a ccolumnas[5]-1	-
5	ccolumnas[5] a ccolumnas[6]-1	filas[9], filas[10] filas[11]
6	ccolumnas[6] a ccolumnas[7]-1	filas[12]

Como en la representación CSR, cuando una columna no contiene ningún valor distinto a cero entonces el valor asociado a dicha columna en el tercer vector (vector **ccolumnas**) es el mismo de la columna anterior. Además, el ultimo valor en el tercer vector corresponde al último índice valido en el segundo vector más 1.

OPERACIONES

Las representaciones de matrices dispersas deben permitir:

1. Crear una matriz completa. Pasar de la representación a la forma matricial
2. Obtener representación. Pasar de la forma matricial a la representación
3. Obtener elemento: Dada una representación y una posición i,j debe retornarse el valor asociado.

4. Obtener fila: Dada una representación y una fila i , debe retornarse la fila asociada
5. Obtener columna: Dada una representación y una columna j , debe retornarse la columna asociada.
6. Modificar posición: Dada una representación, un posición i,j y un elemento, debe modificarse la representación para ingresar este elemento.
7. Suma de matrices: Dadas dos representaciones (del mismo tipo) permitir la suma de las dos matrices (Ambas deben ser del mismo tamaño)
8. Matriz transpuesta: Dada una representación, retornar la representación de la matriz transpuesta

PROYECTO

Este proyecto puede ser realizado en grupos de hasta 2 personas.

1. (30 %) Implementación de las funciones de creación de las representaciones. Estas reciben una matriz y retorna la representación.

a) Función para generar representación en formato coordenadas

b) Función para generar representación en formato CSC

c) Función para generar representación en formato CSR

2. (15 %) Implementación de las operaciones 2, 3, 4, 5 y 6

3. (15 %) Implementación de la suma de matrices. Esta función recibe una representación y retorna la representación de la suma. Debe funcionar para las 3 implementaciones.

4. (15 %) Implementación de la matriz transpuesta. Esta función recibe una representación y retorna la representación de la transpuesta. Debe funcionar para las 3 implementaciones.

5. (25 %) Realizar una comparación entre las tres implementaciones, teniendo en cuenta la complejidad de las operaciones y aspectos generales de ellas. Realice las conclusiones de los ejercicios en base a este análisis. Recuerde que las conclusiones son producto del ejercicio realizado.

Como parte de la nota, debe realizar un informe en formato PDF que cumpla:

1. El informe debe tener los nombres y códigos de los integrantes del grupo
2. Explique el proceso que realiza para generar la representación dada una matriz cualquiera. Realice un ejemplo adicional al mostrado en este enunciado de las 3 representaciones. Esto es parte de la nota del ítem 1.

Explique los aspectos más relevantes de su implementación, tales como estrategias y estructuras de datos utilizadas.

IMPORTANTE: Evite copiar código. La implementación debe ser propia de ustedes.

3. Explicar cada una de las funciones implementadas, las estrategias utilizadas y estructuras de datos.

4. Entregar en un CD debidamente marcado.

GRACIAS