



**DEPARTAMENTO  
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

## Trabajo Práctico 1 - Page Rank

9 de abril de 2018

Métodos Numéricos

**Grupo "Nombre Del Grupo"**

Integrante	LU	Correo electrónico
Facundo, Araujo	321/15	facalj_velez@hotmail.com
Cristian, Kubrak	456/15	Kubrakcristian@gmail.com
Marcela Alejandra, Herrera	1162/84	marcelaalejandraherrera@yahoo.com.ar
Luis Fernando, Greco	150/15	luifergreco@gmail.com



**Facultad de Ciencias Exactas y  
Naturales**

Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta  
Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep.  
Argentina

Tel/Fax: (+54 +11) 4576-3300

<http://www.exactas.uba.ar>

# Índice

1. Introducción	2
2. Demostraciones	2
3. Desarrollo	3
4. Resultados	4
5. Discusión	4
6. Conclusiones	4
7. Apendices	4
8. Referencias	4

# 1. Introducción

## Introducción Teórica

El problema que se nos plantea es el de implementar un Page Rank, es decir, un método para ordenar páginas de acceso público de forma sistematizada y eficiente. Para esto tendremos que definir que parametros vamos a tener en cuenta al momento de armar nuestro orden de páginas. Estos van a ser, la cantidad de links .<sup>entrantes</sup>.<sup>a</sup> una página y la calidad” de cada uno de estos (es decir, que tan relevante es la página de la que proviene ese link).

Encontramos que este problema tiene una gran aplicación, dado que es fundamental para construir un motor de búsqueda y nos permitirá entender mejor como funcionan.

### HABLAR ALGO SOBRE CADENAS DE MARKOV

Para empezar al trabajar con matrices ralas nos vimos en la necesidad de buscar un método implementar este tipo de estructura, teniendo en cuenta un uso eficiente de los recursos del sistema (intentando sacar provecho del conocimiento que tenemos de entrada sobre las matrices). Es en este contexto que decidimos utilizar el formato CSR (Compressed Sparse Row), que vamos a proceder a explicar detalladamente más adelante.

## 2. Demostraciones

### Demostraciones

Justificar que:

$$A = pWD + ez^t$$

Estudiemus como son los elementos de la matriz  $A = pWD + ez^t$ :

$$(pWD + ez^t)_{ij} = p(WD)_{ij} + (ez^t)_{ij} = p\left(\sum_{k=1}^n w_{ik}d_{kj} + 1z^t\right)$$

Como D es una matriz diagonal, los elementos distintos de cero únicamente pueden estar en la diagonal, con lo cual al hacer el producto se anulan todos los términos con  $k \neq j$  y también aquellos términos donde  $w_{ik} = 0$  lo cual ocurre cuando  $c_j = 0$ .

En consecuencia se puede reescribir:

$$(pWD + ez^t)_{ij} = p(w_{ij}d_{jj} + z^t)$$

Vamos a separar en casos para analizar los posibles valores que puede tomar el elemento (notar que por la definición de D, la condición  $d_{jj} = 0$  es equivalente a  $c_j = 0$ ):

caso 1:  $w_{ij} = 0, d_{jj} = 0$  ( $c_j = 0$ )  $\Rightarrow (pWD + ez^t)_{ij} = z_j^t = 1/n$

caso 2:  $w_{ij} = 0, d_{jj} \neq 0$  ( $c_j \neq 0$ )  $\Rightarrow (pWD + ez^t)_{ij} = z_j^t = (1-p)/n$

caso 3:  $w_{ij} = 1, d_{jj} = 0$  ( $c_j = 0$ )  $\Rightarrow (pWD + ez^t)_{ij} = z_j^t = 1/n$

caso 4:  $w_{ij} = 1, d_{jj} \neq 0$  ( $c_j \neq 0$ )  $\Rightarrow (pWD + ez^t)_{ij} = p/c_j + z_j^t = p/c_j + (1-p)/n$

Observando los casos 2 y 4 podemos observar que se pueden unificar en una condición equivalente:  $(pWD + ez^t)_{ij} = w_{ij}p/c_j + z_j^t = p/c_j + (1-p)/n$

Uniando todo lo anterior tenemos que:

$$(pWD + ez^t)_{ij} = \begin{cases} p w_{ij}/c_j + z_j^t = (1-p)/n + p/c_j & \text{si } c_j \neq 0 \\ 1/n & \text{si } c_j = 0 \end{cases} \quad (1)$$

Pero precisamente esto coincide con la definición de los elementos de la matriz A. Luego  $A = pWD + ez^t$ .

¿Cómo se garantiza la aplicabilidad de la Eliminación Gaussiana? ¿La matriz  $I - pWD$  está bien condicionada? ¿Cómo influye el valor de p?

Sea  $B = WD$ ,

$$B_{ij} = \sum_{k=1}^n w_{ik}d_{kj}$$

Por la definición de D, cuando  $k \neq j$ ,  $d_{kj} = 0$ , por lo tanto se anulan los correspondientes términos de la sumatoria, quedando:

$$B_{ij} = w_{ij}d_{jj}$$

También sabemos por la definición de W que  $w_{jj} = 0$  (porque no se consideran los autolinks), así que podemos deducir que los elementos de la diagonal van a ser iguales a cero, luego:

$$B_{ij} = \begin{cases} 0 & \text{si } i = j \\ 1/c_j & \text{si } i \neq j, w_{ij} \neq 0 \end{cases} \quad (2)$$

Y por lo tanto:

$$(I - pB)_{ij} = \begin{cases} 1 & \text{si } i = j \\ -p/c_j & \text{si } i \neq j \end{cases} \quad (3)$$

La matriz  $B$  tiene la particularidad de que la suma de los elementos de cada columna es igual a 1, ya que en cada columna hay  $c_j$  elementos de valor  $1/c_j$ . Por lo tanto en  $I - pB$  la suma de los elementos de cada columna es  $1 + c_j p/c_j$  con  $|c_j p/c_j| < 1$  porque  $0 < p < 1$ . De esto se desprende que  $(I - pB)^t$  es estrictamente diagonal dominante y como se probó en la teórica esto implica que sea no singular. Además, si una matriz es e.d.d. cada una de sus submatrices principales es e.d.d. y entonces todas son no singulares. Por otra parte se puede probar que si una matriz tiene inversa, su traspuesta tiene inversa. Luego podemos afirmar que  $(I - pB)$  es no singular y que todas sus submatrices principales también son no singulares, luego  $(I - pB)$  tiene descomposición  $LU$  que es equivalente a decir que puede realizarse la Eliminación Gaussiana sin que aparezca un cero en la diagonal en ninguno de los pasos del proceso.

### 3. Desarrollo

#### Desarrollo

Explicar EG, resolución de sistema y cómo funciona nuestra estructura.

En primer lugar comenzamos planteando nuestra estructura como un diccionario DOK (Dictionary of Keys) -explicado en la diapositiva en clase- debido a una relativa facilidad a la hora de implementarla, sin embargo correr los primeros tests (los de la cátedra) nos encontramos con que esta implementación no cumplía nuestras expectativas en cuanto al tiempo de ejecución.

Al comenzar nuestra estructura fue un map, luego lo cambiamos por un unordered map en nuestra búsqueda por bajar los tiempos de ejecución. Y finalmente terminamos volviendo a utilizar un map, solo que esta vez con iteradores, y fue con esta estructura que logramos correr el tp en tiempos que consideramos razonables.

EXPLICAR FUNCIONES BASICAS, MULTIPLICACION, SUMA DE MATRICES.

Luego, para implementar la Eliminación Gaussiana lo llevamos a cabo siguiendo el método explicado en clase, con operaciones de suma y multiplicación entre filas pero sin pivoteo entre las mismas. EXPLICACION MAS DETALLADA DE EG

ACA EXPLICAR RESOLUCION DE SISTEMA

## **4. Resultados**

### **Resultados obtenidos**

## 5. Discusión

### Discusión



## **6. Conclusiones**

**Conclusiones**

## 7. Apendices

Apéndices

## 8. Referencias

### Referencias