Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

Cadenas de Markov aplicadas al ordenamiento de páginas web

Tesis

Que para obtener el título de:

Licenciada en Matemáticas Aplicadas

Presenta:

Torres González María Guadalupe

Director de tesis:

Dr. Vázquez Guevara Víctor Hugo

Agosto 2016

Dedicatoria:

A mis padres:

José Francisco
y
María Benita

A mis hermanos:

Abel
y
Juan

A mis sobrinos:

Abel
y
Jesus

Agradecimientos:

Dr. Vázquez Guevara Víctor Hugo

Por confiar en mi, por su tiempo, dedicación y darme la oportunidad de trabajar bajo su tutela.

Dr. Martínez Hernández Mario Iván

Por toda la ayuda que me brindo durante mi carrera.

Lic. Zeleny Vázquez Pablo Rodrigo

Por darme la oportunidad y confianza de trabajar con el.

Introducción

Cuando deseamos encontrar alguna información en internet, solemos usar 'buscadores' de fácil acceso que tienen diversos nombres comerciales: Yahoo, Google, etc. Generalmente sólo atendemos a los primeros resultados que se nos presentan. Por ello, es importante saber cómo nos ordenan las variadas y muchas páginas que tienen algo en común con los temas o palabras consultadas.

Al buscar en cualquier base de datos pedimos simultáneamente que nos responda qué elementos tienen relación con lo que buscamos y cómo los queremos ordenados. En la web, son los buscadores los que eligen el orden. *Google* basa su éxito en un procedimiento que asocia a cada página de la red un número que cuantifica su 'relevancia' (o 'importancia'), y en función de ello ordena los resultados de la búsqueda.

Uno de los principales elementos introducidos por los creadores de *Google* fue el *PageRank*: "Para medir la importancia relativa de las páginas web se propone *PageRank*, un método para calcular un ordenamiento (ranking en inglés) para toda página, basado en el gráfico de la red", explicaron Brin y Page.

Las páginas web varían mucho en el número de vínculos entrantes que poseen. Generalmente, las páginas que son apuntadas desde muchas páginas son más importantes que las páginas a las cuales sólo se llega desde unas pocas. Pero hay muchos casos en los que sólo contar el número de vínculos entrantes no se corresponde con el sentido usual de la importancia de una página web.

Las páginas que aparecen en los primeros lugares de un listado de Google, generalmente, tienen mayor número de visitas que aquellas que aparecen relegadas. En esto radica el interés de los responsables (webmasters) de las páginas comerciales u otras por hacer aparecer sus sitios en los primeros lugares, que intentan aumentar las calificaciones de sus páginas a través de la manipulación de sus enlaces. Los administradores de Google quieren evitar trampas de este tipo, por lo que procuran detectar y penalizar tales intentos.

Pero incluso desde el punto de vista informático, se ha advertido que el PageRank vigente influye en el recorrido mensual realizado por Google: páginas con mayor PageRank son recorridas más rápidamente y 'con mayor profundidad' que otras con menor clasificación.

En este momento, Google no sólo es el buscador más utilizado, sino que vende servicios a portales importantes: Yahoo, AOL, etc. Además, su sistema llamado de 'publicidad direccionada' (junto con los resultados de su búsqueda, Google presenta propaganda relacionada con lo buscado) es la que dirige mayor cantidad de gente hacia sitios comerciales.

¿Cómo funciona búsqueda en Google?

Para explicarlo de forma sencilla, realizar una búsqueda en la Web es como consultar un libro muy extenso en el que un índice exhaustivo nos indica exactamente la ubicación de cada elemento. Cuando se efectúa una búsqueda en Google, nuestros programas consultan el índice para decidir qué resultados de búsqueda son los más relevantes y mostrártelos.

Los tres procesos principales mediante los que se proporcionan los resultados de búsqueda son:

- Rastreo
- Indexación
- Publicación

Los cuales se definen acontinuación.

Rastreo

El rastreo es el proceso mediante el cual el robot de Google descubre páginas nuevas y actualizadas y las añade al índice de Google.

Se utilizan una enorme cantidad de equipos informáticos para obtener (o “rastrear") miles de millones de páginas de la Web. El programa encargado de recuperar este contenido es el robot de Google, también conocido simplemente como robot o araña. El robot de Google utiliza un proceso de rastreo algorítmico: a través de programas informáticos se determinan los sitios que hay que rastrear, la frecuencia y el número de páginas que hay que explorar en cada uno de ellos.

El proceso de rastreo de Google empieza con una lista de direcciones URL de páginas web generada a partir de procesos de rastreo anteriores y se amplía con los datos de los “sitemaps"que ofrecen los webmasters. A medida que el robot de Google visita cada uno de estos sitios web, detecta enlaces en sus páginas y los añade a la lista de páginas para rastrear. Los sitios nuevos, los cambios en los existentes y los enlaces obsoletos se detectan y se utilizan para actualizar el índice de Google.

Indexación

El robot de Google procesa todas las páginas que rastrea para compilar un índice masivo de todas las palabras que ve junto con su ubicación en cada página. Además, también procesa la información incluida en las etiquetas y los atributos de contenido clave, como las etiquetas “title” y los atributos “alt". El robot de Google puede procesar muchos tipos de contenido, pero hay ciertos tipos que no puede procesar, como el contenido de algunos archivos de soportes interactivos y páginas dinámicas.

Publicación de resultados

La relevancia se determina a partir de más de 200 factores, y uno de ellos es la clasificación PageRank de una página en particular. Este parámetro representa la importancia que Google asigna a una página en función de los enlaces procedentes de otras páginas web. Dicho con otras palabras, cada enlace a una página de un sitio incluido en otro sitio añade valor al PageRank del primero. No todos los enlaces son iguales: Google se esfuerza en mejorar el servicio que ofrece al usuario identificando los enlaces fraudulentos y otras prácticas que influyen negativamente en los resultados de búsqueda. Los mejores tipos de enlaces son los que se crean por la calidad del contenido.

Para que un sitio consiga una buena posición en las páginas de resultados, es importante asegurarse de que Google pueda rastrearlo e indexarlo correctamente. Las funciones “Quizás quisiste decir” y “Autocompletar"de Google están diseñadas para permitir a los usuarios ahorrar tiempo y, para ello, se les muestran las palabras relacionadas, los errores ortográficos comunes y las consultas populares. Al igual que los resultados de búsqueda de google.com, las palabras clave que utilizan estas funciones se generan de forma automática a través de los rastreadores web y algoritmos de búsqueda. Sólo se muestran estas predicciones cuando se considera que pueden ahorrar tiempo a los usuarios. Si un sitio está bien clasificado con respecto a una palabra clave, se debe a que se ha determinado mediante algoritmos que su contenido es más relevante para la consulta del usuario.

Cuando necesitamos buscar información lo primero que hacemos es acudir a internet, pero en internet hay mucha información y a veces no encontramos lo que en realidad buscamos, esto puede ser una ventaja y también un inconveniente. Google es actualmente uno de los motores de búsqueda más utilizados, por eso creemos interesante aprender a utilizar el buscador Google con sus múltiples herramientas y sacar el máximo partido a la hora de realizar las búsquedas. Además, tiene un diseño simple y funcional, con unos tiempos de respuesta muy rápidos.

Interfaz

Para acceder a Google tan solo tenemos que ejecutar un explorador de internet y escribir la dirección de Google (www.google.com). La página principal de Google contiene los siguientes campos:

1. **Cuadro de búsqueda:** Lugar en el cual escribimos las palabras que queremos buscar.

2. **Botón de búsqueda en Google:** Una vez que hemos introducido la palabra o palabras que queremos buscar, tan solo tenemos que pulsar sobre “buscar".

3. **Me siento con suerte:** Si pulsamos sobre “me siento con suerte"Google, nos va a mostrar la página que más se ajusta a nuestros criterios de búsqueda.

4. **Búsqueda por voz:** Solo necesitas un micrófono integrado o externo. Haz clic en el micrófono situado en la barra de búsqueda y empieza a hablar.

5. **Barra de estadísticas:** Nos muestra el número de resultados y el tiempo que se ha tardado en completar la búsqueda.

6. **Título de la página:** Nos muestra el título de la página web que ha encontrado. En algunas ocaciones no aparece el título si no la URL, esto quiere decir que la página no tiene título.

7. **URL del resultado:** Dirección web del resultado encontrado.

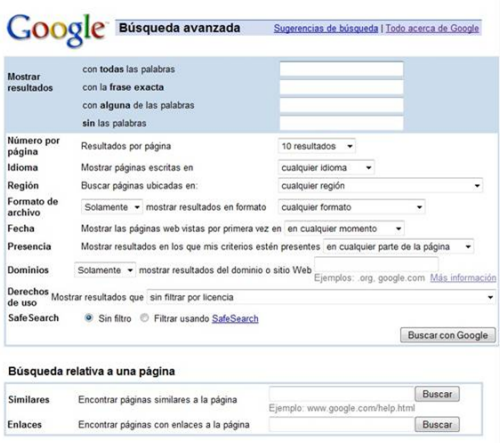
8. **Texto debajo del título:** Nos muestra un resumen de la página con los términos de búsqueda resaltados.

9. **Páginas similares:** Al hacer clic en “páginas similares", Google nos muestra las páginas que están relacionadas con el resultado.

10. **Resultado jerarquizado:** Cuando Google encuentra más de un resultado de nuestra búsqueda, Google muestra en la parte inferior una lista con las páginas más relevantes y a continuación mostrará el resto de los resultados.

Busqueda avanzada

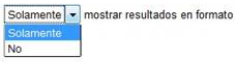
Las búsquedas avanzadas permiten limitar aun más la búsqueda que queramos realizar.



- **Mostrar resultados:** A la hora de mostrar los resultados Google tiene distintas opciones de realizar la búsqueda: con todas las palabras (en la cual introduciendo las palabras en el cuadro de búsqueda, nos mostrara las páginas que

contengan todas las palabras buscadas, pudiendo considerar como la búsqueda que realizamos normalmente, funciona como el operador lógico AND), con la frase exacta (nos va a mostrar páginas que contenga la frase tal y como la hemos escrito en el cuadro de búsqueda. Esta opción equivale a poner el texto entre comillas), con alguna de las palabras (nos va a mostrar páginas que contenga algunas de las palabras que hemos escrito en el cuadro de búsqueda, funciona como el operador lógico OR,y finalmente sin las palabras (nos va a mostrar páginas que no contengan las palabras escritas en el cuadro de búsqueda)

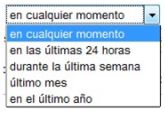
- **Número por página:** Nos permite elegir el número de resultados que queremos obtener por página (10, 20, 30, 50 y 100 resultados).
- **Idioma:** Podemos elegir el idioma en el que queremos que aparezcan nuestras páginas de búsqueda. Por ejemplo, si elegimos el español, las páginas mostradas aparecerán solo en español.
- **Región:** Nos permite elegir el país en el cual queremos que muestre la información buscada. Por ejemplo, si queremos que la información que buscamos nos muestre las páginas del Reino Unido seleccionaremos este país.
- **Formato de archivo:** A partir de dos menús desplegables podemos elegir el tipo de archivo que queremos encontrar. El primer menú desplegable nos permite elegir si solamente queremos ese formato o no.



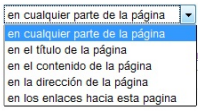
El segundo menú desplegable nos permite elegir el tipo de archivo que queremos que nos muestre.

Así por ejemplo, si seleccionamos “Solamente mostrar resultado en formato Adobe Acrobat PDF (.pdf)”o “No mostrar resultado en formato Miscrosoft Word (.doc)” discriminará los resultados en formato word.

- **Fecha:** A partir de un menú desplegable nos permite seleccionar que páginas queremos que se nos muestren en función de determinadas fechas, a partir de la última en la cual fueron actualizadas.



- **Presencia:** Nos permite seleccionar a partir de un menú desplegable en que parte de la página queremos que se realice la búsqueda.



- **Dominios:** A partir del menú desplegables nos permite el discriminar o bien limitar la búsqueda a un dominio o sitio web.

Si escribimos www.ite.educacion.es y el cuadro de búsqueda de "mostrar resultados" escribimos terremotos, solo nos mostrará las páginas que contengan la palabra terremoto y estén en el sitio web que hemos indicado.

- **Derechos de uso:** Podemos elegir que los resultados que nos muestre se puedan compartir o modificar.
- **Safe Search:** Esta opción nos permite cambiar la configuración del navegador aplicando filtros para evitar contenidos para adultos que aparezcan en los resultados de búsqueda

Búsqueda de imagenes

Si seleccionamos la imagen posicionándonos sobre élla podemos ver el nombre del archivo y su extensión, tamaño de la pantalla, URL donde se encuentra la imagen, otras similares y más tamaños. Si no encontramos una imagen podemos recurrir a la búsqueda avanzada.

Podemos elegir cómo queremos que nos muestren los resultados, si bien "todas las palabras", "con la frase exacta", "algunas de las palabras" o "no relacionadas las

palabras". El tipo de contenido, tamaño, tamaño exacto, formato, tipo de archivo, coloración, dominios, derechos de uso y Safe Search.

Búsqueda de videos

Si queremos la búsqueda más precisa de un video seleccionaremos la búsqueda avanzada, ésta nos va a permitir seleccionar cómo buscar los resultados, el idioma, la duración, el dominio, si queremos que busque libros con subtítulos, cómo queremos que aparezcan ordenados los resultados así como los resultados por página.

Google Académico

Google Académico también dispone de una Búsqueda avanzada que nos va a permitir el poder acotar los resultados.

La búsqueda avanzada de Google Académico permite buscar artículos con todas las palabras, con la frase exacta, con al menos una de las palabras, sin las palabras, donde las palabras aparezcan, de la misma forma que la búsqueda en la web. Podemos hacer uso de las comillas. Autor: con frecuencia para localizar una obra, artículo, etc, recurrimos a la búsqueda mediante el autor, para localizar dicho documento mediante el autor escribiremos el nombre del autor entre comillas incluso también podemos usar iniciales para realizar la búsqueda. Podemos también realizar la búsqueda escribiendo el nombre de donde han sido publicados o bien el año de su publicación. También nos permite configurar el número de resultados que queremos que aparezcan por página.

Índice general

Capítulo 1

Cadenas de Markov

PageRank, el método que utiliza Google para clasificar las páginas web de acuerdo a su importancia, es una de tantas aplicaciones de las cadenas de Markov, las cuales, fueron introducidas por el matemático ruso Andrey Markov alrededor de 1905. Su intención era crear un modelo probabilístico para analizar la frecuencia con la que aparecen las vocales en poemas y textos literarios. El éxito del modelo propuesto por Markov radica en que es lo suficientemente complejo como para describir ciertas características no triviales de algunos sistemas, pero al mismo tiempo es lo suficientemente sencillo para ser analizado matemáticamente. En este capítulo se presenta una pequeña introducción de las Cadenas de Markov que nos servirá para el estudio del PageRank.

Definición 1.1 *Un proceso estocástico es una colección de variables aleatorias* $\{X_t : t \in T\}$ *parametrizada por un conjunto* T*, llamado espacio parametral, en donde las variables toman valores en un conjunto* S *llamado espacio de estados.*

En los casos más sencillos se toma como espacio parametral el conjunto discreto $T = \{0, 1, 2, ...\}$ y a estos números se les conoce como instantes. En este caso se dice que el proceso es a tiempo discreto, y en general este tipo de procesos se denotara por $\{X_n : n \in T\}$, o explicitamente, $X_0, X_1, X_2, ...$.

Este modelo corresponde a un proceso aleatorio de dimensión infinita. Véase Figura 1.1

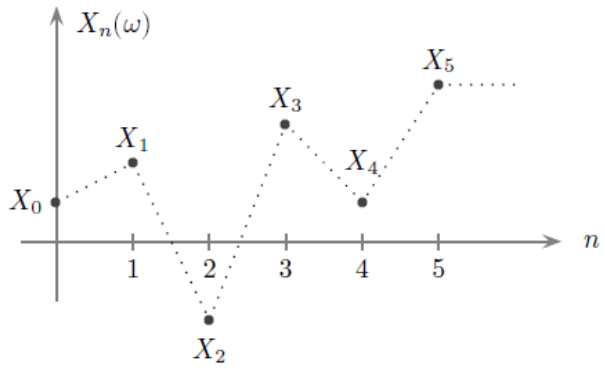


Figura 1.1

El espacio parametral puede también considerarse como el conjunto continuo $T = [0, \infty)$. Se dice entonces que el proceso es a tiempo continuo, y se denota por $\{X_t : t \geq 0\}$.

Por lo tanto, seguiremos la convención de que si el subíndice es n, entonces los tiempos son discretos, y si el subíndice es t, el tiempo se mide de manera continua.

Los espacios de estados que se consideran son subconjuntos de $\mathbb{Z}$, y un poco más generalmente tomaremos como espacio de estados al conjunto de los números reales $\mathbb{R}$. Naturalmente, espacios más generales son posibles, tanto para el espacio parametral como para el espacio de estados. En particular, para poder hablar de variables aleatorias con valores en el espacio de estados S, es necesario asociar a este conjunto una σ-álgebra.

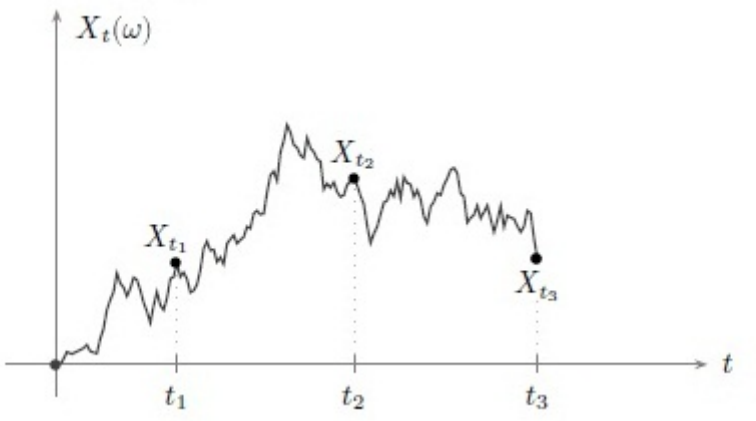


Figura 1.2

Considerando que S es un subconjunto de $\mathbb{R}$, puede considerarse la σ-álgebra de Borel de $\mathbb{R}$ restringida a S, es decir $S \cap \mathscr{B}(\mathbb{R})$.

Un proceso estocástico puede considerarse como una función de dos variables $X : T \text{ x } \Omega \longrightarrow S$ tal que a la pareja (t, ω) se le asocia el estado $X(t, \omega)$, lo cual también puede escribirse como $X_t(\omega)$. Para cada valor de $t \in T$, el mapeo $\omega \mapsto X_t(\omega)$

es una variable aleatoria, mientras que para cada $\omega \in \Omega$ fijo, la función $t \mapsto X_t(\omega)$ es llamada una *trayectoria* o *realización* del *proceso*.

Es por ello que a veces se define un proceso estocástico como una función aleatoria. Una de tales trayectorias típicas que además cuenta con la propiedad de ser continua se muestra en la Figura 1.2, y corresponde a una trayectoria de un movimiento Browniano [5].

1.1. Cadenas de Markov

Muchos sistemas tienen la propiedad de que dado el estado x_n (actual), los estados $x_0, x_1, ...x_{n-1}$ (pasado) no influyen en x_{n+1} futuro. Esta propiedad es llamada, la *propiedad de Markov*, y los sistemas que tienen esta propiedad son llamadas *cadenas de Markov*. La *propiedad de Markov* en el caso discreto es definida por la condición:

$$P(X_{n+1} = x_{n+1}|X_0 = x_0, ..., X_n = x_n) = P(X_{n+1} = x_{n+1}|X_n = x_n)$$

De esta forma la probabilidad del evento futuro $\{X_{n+1} = x_{n+1}\}$ sólo depende del evento $\{X_n = x_n\}$, mientras que la información correspondiente al evento pasado $\{X_0 = x_0, ..., X_{n-1} = x_{n-1}\}$ es irrelevante. Los procesos de *Markov* han sido estudiados extensamente y existe un gran número de sistemas que surgen en muy diversas disciplinas del conocimiento, para los cuales, el modelo de proceso estocástico y la propiedad de *Markov* son razonables.

Definición 1.1 *Una cadena de Markov es un proceso estocástico a tiempo discreto* $\{X_n : n = 0, 1, ...\}$ *con espacio de estados discreto, y que satisface la propiedad de Markov, esto es, para cualquier entero* $n \geq 0$*, y para cualesquiera estados* $x_0, ..., x_{n+1}$*, se cumple*

$$P(X_{n+1} = x_{n+1}|X_0 = x_0, ..., X_n = x_n) = P(X_{n+1} = x_{n+1}|X_n = x_n)$$

en dónde $\{x_{n+1}|x_0, x_1, ..., x_n\}$ *es una forma simplificada del evento*

$$\{X_{n+1} = x_{n+1}|X_0 = x_0, ..., X_n = x_n\}$$

Análogamente con $\{X_{n+1}|X_n\}$.

Sean i y j dos estados de una *cadena* de *Markov*. A la probabilidad

$$P(X_{n+1} = j|X_n = i)$$

se le denota por $p_{ij}(n, n+1)$, y representa la probabilidad de transición del estado i en el instante n al estado j en el tiempo $n+1$. Estas probabilidades se conocen

como las probabilidades de transición en un paso.

Si $\forall k, n \in \mathbb{N} : \mathbb{P}\{X_{n+1} = y | X_n = x\} = \mathbb{P}\{X_{k+1} = y | X_k = x\}$

se escribe P_{xy} ó $P(x, y)$.

Sea A un subconjunto de S. El *tiempo de alcance* T_A de A esta definido por

$$T_A = inf(n > 0 : X_n \in A).$$

Si ocure que $\forall n > 0$, $X_n \notin A$ entonces $T_A = \infty$. De esta manera $T_A : \Omega \mapsto \{1, 2, ..., \infty\}$ es una variable aleatoria. Cuando A es un conjunto unitario, es decir, $A = \{x\}$ denotamos a T_A por T_x

Supongamos que el espacio de estados es finito $S = \{x_0\, x_1 ... \, x_N\}$, entonces las probabilidades de transición se pueden representar por una matriz

$$P = \begin{pmatrix} p(x_0, x_0) & p(x_0, x_1) & ... & p(x_0, x_N) \\ p(x_1, x_0) & p(x_1, x_1) & ... & p(x_1, x_N) \\ \vdots & \vdots & ... & \vdots \\ p(x_N, x_0) & p(x_N, x_1) & ... & p(x_N, x_N) \end{pmatrix}$$

que es llamada *matriz de transición.*

Proposición 1.1 *La matriz de probabilidades de transición de una cadena de Markov $P = p(i, j)$ cumple las siguientes dos propiedades.*

a) $p(i, j) \geq 0$.

b) $\sum_j p(i, j) = 1$

Demostración

La primera condición es evidente pues los números son probabilidades. Para la segunda propiedad, observemos primero que se cumple la descomposición disjunta:

$$\Omega = \bigcup_j (X_1 = j).$$

Por lo tanto, para cualesquiera estados i y j,

$$1 = P(\bigcup_j (X_1 = j) | X_0 = i) = \sum_j P(X_1 = j | X_0 = i) = \sum_j p(i, j).$$

La ecuación de *Chapman-Kolmogorov* es una fórmula sencilla y muy útil que permite descomponer la probabilidad de pasar del estado i al estado j en n pasos, en la suma de probabilidades de las trayectorias que van de i a j, y que atraviesan por un estado cualquiera k en un instante intermedio r. Gráficamente, las trayectorias que van del estado i al estado j en n pasos se descomponen como se muestra en la Figura 1.3.

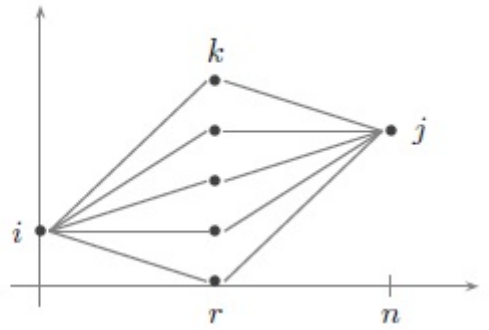


Figura 1.3

Proposición 1.2 *Para cualesquiera números enteros r y n tales que $0 \leq r \leq n$, y para cualesquiera estados i y j se cumple que*

$$p_{ij}(n) = \sum_k p_{ik}(r)p_{kj}(n-r)$$

en donde $p_{ij}(n) = P(X_n = j|X_0 = i)$.

Demostración

Por el Teorema de Probabilidad Total y la propiedad de Markov, se tiene que

$$\begin{aligned} p_{ij}(n) &= P(X_n = j|X_0 = i) \\ &= \textstyle\sum_k P(X_n = j, X_r = k, X_0 = i)/P(X_0 = i) \\ &= \textstyle\sum_k P(X_n = j|X_r = k)P(X_r = k|X_0 = i) \\ &= \textstyle\sum_k p_{kj}(n-r)p_{ik}(r). \end{aligned}$$

Proposición: $p_{ij}(n) = P^n(i,j)$.

En una cadena de *Markov* hay estados recurrentes y transitorios, los cuales, se estudiaran a continuación.

Antes de definir los estados recurrentes, se tiene que:

$$\rho_{xy} = p(x,y) + \sum_{z \neq y} P(x,z)\rho_{zy}.$$

Un estado y se llama *recurrente* si $\rho_{yy} = 1$ y *transitorio* si $\rho_{yy} < 1$, en donde ρ_{xy} se define por

$$\rho_{xy} := P_x(T_y < +\infty) = \mathbb{P}\{T_y < \infty | X_0 = x\},\ x, y \in S$$

representa la probabilidad de visitar al estado y bajo el supuesto de que la cadena inicia en x.

En particular, si $x = y$ entonces ρ_{yy} representa la probabilidad de retorno a y.

1.2. Cadenas Irreducibles

Se dice que una cadena de *Markov* es irreducle si todos los estados se comunican entre sí, es decir, x accede a y para todo $x, y \in C$, donde C es un conjunto cerrado. En seguida definiremos este tipo de conjuntos.

Un conjunto de estados C se dice que es *cerrado* si ningun estado de C accede a algún estado fuera de C, es decir, si

$$\rho_{xy} = 0,\ x \in C,\ y \notin C.$$

El periodo de una cadena es importante al momento de estudiar distribuciones estacionarias como veremos más adelante.

Sea $x \in S$, definimos el periodo de x como:

$$d_x := mcd\{n \geq 1 : P^n(x, x) > 0\}$$

donde mcd denota el máximo común divisor del número de pasos necesarios para volver al estado i, bajo el supuesto de que se inicia en i.

Una cadena es aperiódica si $d_x = 1$, $\forall x \in S$.

Los estados de una cadena de *Markov*, tienen algunas propiedades, las cuales, se veran enseguida.

Sea $N(y)$ el número de veces que la cadena visita el estado y, se tiene entonces que:

$$N(y) = \sum_{k=1}^{\infty} 1_y(X_k) \tag{1.1}$$

Análogamente, $N_n(y) = \sum_{k=1}^{n} 1_y(X_k)$ es el número de veces que la cadena visita el estado y en las primeras n unidades de tiempo.

Se usará la notación $E_x()$ para denotar la esperanza de variables aleatorias en terminos de una cadena de $Markov$ que inicia en x. Por ejemplo.

$$E_x(1_y(X_n)) = P_x(X_n = y) = P^n(x, y) \tag{1.2}$$

Se sigue de (1.1) y (1.2) que

$$\begin{aligned} E_x(N(y)) &= E_x(\textstyle\sum_{n=1}^{\infty} 1_y(X_n)) \\ &= \textstyle\sum_{n=1}^{\infty} E_x(1_y(X_n)) \\ &= \textstyle\sum_{n=1}^{\infty} P^n(x, y). \end{aligned}$$

Sea $G(x, y) := E_x(N(y))$ denota el número promedio de visitas a y dado que la cadena de $Markov$ inicia en x, así

$$G(x, y) = E_x(N(y)) = \sum_{n=1}^{\infty} P^n(x, y)$$

y

$$G_n(x, y) = \sum_{k=1}^{n} P^k(x, y).$$

Representa el número de veces que la cadena se encuentra en el estado y en las primeras n unidades de tiempo.

Teorema 1.1 *Sea y un estado recurrente. Entonces*

$$\lim_{n\to\infty} \tfrac{N_n(y)}{n} = \tfrac{1_{\{T_y<\infty\}}}{m_y} \text{ con probabilidad 1}$$

y

$$\lim_{n\to\infty} \frac{G_n(x, y)}{n} = \frac{\rho_{xy}}{m_y}, x \in S.$$

Definimos el promedio de retorno a y, como

$$m_y = E_y(T_y)$$

para una cadena de $Markov$ comenzando en y si el tiempo de retorno tiene esperanza finita, y como $m_y = \infty$ en otro caso.

1.3. Distribución Estacionaria

Sea $\{X_n, n \geq 0\}$ una *cadena de Markov* con espacio de estados S y matriz de transición P. Si $\pi(x)$, $x \in S$, son números no negativos cuya suma es 1, y además

$$\sum_{x \in S} \pi(x) P(x, y) = \pi(y), \quad y \in S \tag{1.3}$$

entonces, π es llamada una distribución estacionaria. Supongamos que una distribución estacionaria existe y que

$$\lim_{n \to \infty} P^n(x, y) = \pi(y) \tag{1.4}$$

entonces, como pronto veremos, independientemente de la distribución inicial de la cadena, la distribución de X_n converge a π cuando $n \to \infty$. En este caso, π es a veces llamada distribución estacionaria o invariante de estados [5].

Sea π una distribución estacionaria. Entonces

$$\textstyle\sum_x \pi(x) P^2(x, y) = \sum_x \pi(x) \sum_z P(x, z) P(z, y)$$

$$\textstyle\sum_z (\sum_x P(x, z)) P(z, y))$$

$$\textstyle\sum_z \pi(z) P(z, y) = \pi(y).$$

Similarmente; por inducción, basándonos en la fórmula

$$P^{n+1}(x, y) = \sum_z P^n(x, z) P(z, y),$$

concluimos que para toda n

$$\sum_x \pi(x) P^n(x, y) = \pi(y), \quad y \in S. \tag{1.5}$$

Si X_0 tiene la distribución estacionaria π como su distribución inicial, entonces (1.5) implica que, para todo n

$$P(X_n = y) = \pi(y), \quad y \in S, \tag{1.6}$$

y por lo tanto, la distribución de X_n es independiente de n. Supongamos inversamente que la distribución de X_n es independiente de n. Entonces la distribución

inicial π_0 es tal que

$$\pi_0(y) = P(X_0 = y) = P(X_1 = y) = \sum_x \pi_0(x)P(x, y).$$

Por consiguiente, π_0 es una distribución estacionaria. En resumen, la distribución de X_n es independiente de n si y solo si la distribución inicial es una distribución estacionaria.

Supóngase ahora que π es una distribución estacionaria, S es finito y que (1.4) es verdadera. Sea π_0 la distribución inicial. Entonces

$$P(X_n = y) = \sum_x \pi_0(x)P^n(x, y), \quad y \in S. \tag{1.7}$$

Mediante (1.4) y haciendo tender n a infinito en (1.7), se tiene que

$$\lim_{n\to\infty} P(X_n = y) = \sum_x \pi_0(x)P(x, y), \quad y \in S.$$

Dado que $\sum_x \pi_0(x) = 1$, se concluye que

$$\lim_{n\to\infty} P(X_n = y) = \pi(y), \quad y \in S. \tag{1.8}$$

La fórmula (1.8) establece que independientemente de la distribución inicial, para n suficientemente grande la distribución de X_n es aproximadamente igual a la distribución estacionaria π. Esto implica que π es la única distribución estacionaria.

Las cadenas recurrentes positivas son de utilidad para el estudio de las distribuciones estacionarias, más adelante veremos que para que la cadena de *Markov* tenga distribución estacionaria única, una de las caracteristicas que debe cumplir es ser recurrente positiva e irreducible.

Ahora definiremos los estados recurrentes positivos

Un estado recurrente y se llama *recurrente positivo* si $m_y < \infty$. Siguiendo el Teorema 1.1, si y es recurrente positivo, entonces

$$\lim_{n\to\infty} \frac{G_n(y, y)}{n} = \frac{1}{m_y} > 0.$$

Una cadena es recurrente positiva si todos sus estados son recurrentes positivos.

Sea π una distribución estacionaria y m un entero positivo. Entonces por (1.3)

$$\sum_z \pi(z)P^m(z,x) = \pi(x).$$

sumando esta ecuación sobre $m = 1, 2, ..., n$ y dividiendo por n, concluimos que

$$\sum_z \pi(z)\frac{G_n(z,x)}{n} = \pi(x), \quad x \in S. \tag{1.9}$$

Teorema 1.2 *Una cadena de Markov irreducible y recurrente positiva tiene una única distribución estacionaria*

$$\pi(x) = \frac{1}{m_x}. \tag{1.10}$$

Demostración:

Siguiendo el Teorema 1.1 y los supuestos de este tenemos que

$$\lim_{n\to\infty} \frac{G_n(z,x)}{n} = \frac{1}{m_x}, \quad x, z \in S. \tag{1.11}$$

Supongamos que π es una distribución estacionaria. Observamos de (1.9), (1.11) y [2] que

$$\pi(x) = \lim_{n\to\infty} \sum_z \pi(z)\frac{G_n(z,x)}{n} = \frac{1}{m_x}, \quad x, z \in S.$$
$$= \tfrac{1}{m_x}\textstyle\sum_z \pi(z) = \tfrac{1}{m_x}.$$

Así, si hay una distribución estacionaria, debe estar dada por (1.10). Para completar la prueba del teorema necesitamos mostrar que la función $\pi(x)$, $x \in S$, definida por (1.10) es una distribución estacionaria. Es claramente no negativa, así debemos mostrar solamente que

$$\sum_x \frac{1}{m_x} = 1, \tag{1.12}$$

y

$$\sum_x \frac{1}{m_x}P(x,y) = \frac{1}{m_y}, \quad y \in S. \tag{1.13}$$

Con este fin, observemos primero que

$$\sum_x P^n(z,x) = 1.$$

Sumando sobre $m = 1, ..., n$ *y dividiendo por* n*, concluimos que*

$$\sum_x \frac{G_n(z,x)}{n} = 1, \quad z \in S. \tag{1.14}$$

Se observa que, por la ecuación de Chapman-Kolmogorov

$$\sum_x P^m(z,x)P(x,y) = P^{m+1}(z,y).$$

De nuevo, sumando sobre $m = 1, ..., n$ *y dividiendo por* n*, concluimos que*

$$\sum_x \frac{G_n(z,x)}{n} P(x,y) = \frac{G_{n+1}(z,y)}{n} = \frac{P(z,y)}{n}. \tag{1.15}$$

Si S *es finito, entonces por (1.11) y (1.14) que*

$$1 = \lim_{n\to\infty} \sum_x \frac{G_n(z,x)}{n} = \sum_x \frac{1}{m_x}$$

es decir, que (1.12) es verdadera. Similarmente, concluimos que (1.13) es válida cuando $n \to \infty$ *en (1.15). Esto completa la prueba del teorema si* S *es finito.*

El argumento para completar la prueba para S *infinito es más complicado, ya que no podemos intercambiar directamente límites y sumandos como hicimos para* S *finito (el teorema de convergencia acotada no es aplicable). Sea* S_1 *un subconjunto finito de* S*. Vemos por (1.14) que*

$$\sum_{x\in S_1} \frac{G_n(z,x)}{n} \leq 1, \quad z \in S_1.$$

Dado que S_1 *es finito, podemos hacer* $n \to \infty$ *y concluimos por (1.11) que*

$$\sum_{x\in S_1} \frac{1}{m_x} \leq 1.$$

La última desigualdad se cumple para cualquier subconjunto finito S_1 *de* S*, y por*

lo tanto

$$\sum_x \frac{1}{m_x} \leq 1. \tag{1.16}$$

Si la suma de $\frac{1}{m_x}$ *sobre* $x \in S$ *supera a 1, la suma sobre algunos subconjuntos finitos de* S *también supera a 1.*

Similarmente, concluimos por (1.15) que si S_1 *es un subconjunto finito de* S*, entonces*

$$\sum_{x \in S_1} \frac{G_n(z,x)}{n} P(x,y) = \frac{G_{n+1}(z,y)}{n} = \frac{P(z,y)}{n}.$$

Cuando $n \to \infty$ *en esta desigualdad y usando (1.11), obtenemos*

$$\sum_{x \in S_1} \frac{1}{m_x} P(x,y) \leq \frac{1}{m_y}$$

concluimos, como en la prueba de (1.16), que

$$\sum_x \frac{1}{m_x} P(x,y) \leq \frac{1}{m_y}, \quad y \in S \tag{1.17}$$

siguiendo con la demostración de la desigualdad (1.17). Se deduce de (1.16) que la suma de y de la parte derecha de (1.17) es finita. Si se mantiene la desiguladad estricta para algunos estados, se seguirá sumando (1.17) sobre y

$$\begin{aligned} \textstyle\sum_y \frac{1}{m_y} &> \textstyle\sum_y (\sum_x \frac{1}{m_x} P(x,y)) \\ &= \textstyle\sum_x \frac{1}{m_x} (\sum_y P(x,y)) \\ &= \textstyle\sum_x \frac{1}{m_x} \end{aligned}$$

lo cual es una contradicción. Esto prueba la desigualdad obtenida en (1.17), es decir, que (1.13) es verdadera.

Sea

$$c = \frac{1}{\sum_x \frac{1}{m_x}}$$

entonces por (1.13)

$$\pi(x) = \frac{c}{m_x}, \quad x \in S$$

define una distribución estacionaria. Por la primera parte de la demostración de este teorema y por lo tanto, $c = 1$. Esto prueba (1.12) y completa la prueba del teorema.

Teorema 1.3 *Sea $\{X_n,\ n \geq 0\}$, una cadena de Markov irreducible, recurrente positiva que tiene una distribución estacionaria π. Si la cadena es aperiódica, entonces*

$$\lim_{n\to\infty} P^n(x,y) = \pi(y), \quad x, y \in S.$$

Consideremos P^n la matriz de transición en $n - pasos$. Por la *ecuación de Chapman-Kolmogorov* para $m = n = 1$

$$P^2(x,y) = \sum_z P(x,z)P(z,y).$$

Recordando la definición de mulltiplicación de matrices ordinaria, observemos que la matriz de transición en dos pasos P^2 es el producto de la matriz con ella misma. Mas aún, para $m = 1$ en la *ecuación de Chapman-Kolmogorov* se observa que

$$P^{n+1}(x,y) = \sum_z P^n(x,z)P(z,y). \tag{1.18}$$

Se sigue de (1.18) por inducción que la matriz de transición P^n en el $n - esimo$ paso es la $n - esima$ potencia de P.

Una distribución inicial π_0 puede ser pensado como un vector fila de $(d+1) - dimensional$

$$\pi_0 = (\pi_0(0), ..., \pi_0(d)).$$

Si π_n denota el vector fila $(d+1) - dimensional$

$$\pi_n = (P(X_n = 0), ..., P(X_n = d)),$$

entonces

$$P(X_n = y) = \sum_x \pi_0(x)P^n(x,y)$$

y

$$P(X_{n+1} = y) = \sum_x \pi_n(x)P(x,y)$$

pueden escribirse respectivamente como

$$\pi_n = \pi_0 P^n$$

y

$$\pi_{n+1} = \pi_n P.$$

Capítulo 2

Las Matemáticas Detrás de Google

En este capítulo se estudiaran conceptos matemáticos que nos ayudaran a la construcción de la matriz de *Google*, además de la obtención del vector *PageRank*. En seguida veremos que, como la matriz de *Google* es de transición entonces, convergerá a un vector propio, que es el *PageRank*.

Teorema 2.1 *Si P es una matriz de transición y $\lim_{k\to\infty} \pi P^k = s$, tal que π es un vector de probabilidad inicial, entonces $sP = s$ y s es un vector propio del valor propio 1.*

Demostración:

Si el $\lim_{k\to\infty} \pi P^k = s$, multiplicando por P por la derecha, entonces se obtiene $\lim_{k\to\infty} \pi P^k P = sP$ o $\lim_{k\to\infty} \pi P^{k+1} = sP$. El límite cuando k tiende a ∞ para P^{k+1} es el mismo que el límite cuando k tiende a ∞ para P^k. Así, $sP = \lim_{k\to\infty} \pi P^{k+1} = \lim_{k\to\infty} \pi P^k = s$.

En muchos trabajos acerca del *PageRank*, se trabaja con matrices que son estocásticas por columnas, en este caso, se trabaja con matrices estocásticas por filas, pero eso no es problema ya que, basta calcular la transpuesta de la matriz, pues ambas tienen el mismo valor propio, como se verá acontinuación.

Lema 2.2 *Para cualquier matriz cuadrada P, λ es un valor propio de P si y solo si λ es un valor propio de P^T.*

Demostración

λ es un valor propio de P si y solo si $det(P - \lambda I) = 0$. Así $0 = det(P - \lambda I)$ si y solo si $0 = det((P - \lambda I)^T) = det(P^T - \lambda I)$. Así, λ es un valor propio de P si λ es un valor propio de P^T.

Ahora veremos que toda matriz estocástica tiene valor propio 1.

Teorema 2.3 *Si P es la matriz de transición de una cadena de Markov, entonces 1 es un valor propio de la matriz P [6].*

Esperamos que, si 1 es un valor propio de una matriz estocástica, (Teorema 2.3) este sea el mayor.

Teorema 2.4 *Si P es la matriz estocástica de una cadena de Markov, entonces cada valor propio (λ) de P satisface que $|\lambda| \leq 1$*

Demostración:

Sea $Pv = \lambda v$, con $v = (v_1 \cdots v_n)^T$ diferente de cero. Sea $1 \leq m \leq n$ tal que $|v_i| \leq |v_m|$ para toda i.

$|\lambda||v_m| = |\lambda v_m| = |(Pv)_m| = |p_{m1}v_1 + p_{m2}v_2 + \cdots + p_{mn}v_n| \leq |p_{m1}||v_1| + |p_{m2}||v_2| + \cdots + |p_{mn}||v_n| \leq |p_{m1}||v_m| + |p_{m2}||v_m| + \cdots + |p_{mn}||v_m| \leq |v_m|$.

Dado que v es diferente de cero, dividimos ambos lados por $|v_m|$, dando $|\lambda| \leq 1$. Por lo tanto, P tiene valores propios tales que cada uno satisface que $|\lambda| \leq 1$.

Del teorema anterior, surge el siguiente corolario.

Corolario 2.5 *Si P es la matriz de transición de una cadena de Markov entonces $\rho(P) = 1$.*

En donde $\rho(P)$ es el radio espectral de la matriz P que es el mayor de los valores propios de P.

Antes de presentar los siguientes teoremas, veremos algunas observaciones y proposiciones que nos ayudaran a su estudio [4].

Observación 2.6 *Si $P \in MP_n$ (Matriz cuadrada positiva) y un vector $x > 0$, $\beta x < Px \Rightarrow \beta < \rho(P)$ y $Px < \alpha x \Rightarrow \rho(P) < \alpha$ con $\alpha, \beta \in \mathbb{R}$.*

Observación 2.7 *Sean $P \in MEP_n$(Matriz cuadrada de entradas estrictamente positivas), $x \in \mathbb{R}^n$, $x \neq 0$. Notar que, si $x > 0$, entonces tiene que pasar que $Px > 0$.*

Proposición 2.8 *Sean $P \in MEP_n$ y $\lambda \in \sigma(P) = \{\lambda \in \mathbb{C} : ker(P - \lambda I) \neq \{0\}\}$ el espectro de P, un valor propio de módulo máximo, o sea que $|\lambda| = \rho(P)$. Dado un*

vector propio $y \in \mathbb{C}^n$ distinto de cero para λ, es decir, que $Py = \lambda y$, se tiene que

$$|y| > 0 \qquad y \qquad P|y| = \rho(P)|y|.$$

Demostración:

Sea $x = |y|$. Por la desigualdad triangular, se tiene que

$\rho(P)x = |\lambda|x = |\lambda y| = |Py| \leq P|y| = Px.$

Sea $z = Px - \rho(P)x \geq 0$. Queremos mostrar que $z = 0$. Supongamos que eso no pasa. Entonces, por la Observación 2.6 tenemos que $Pz > 0$. Si ahora

$u = Px$, *entonces* $Pz = P(u - \rho(P)x) = Pu - \rho(P)u > 0.$

Por lo tanto, tenemos que $u > 0$ y $Pu > \rho(P)u$. Aplicando la Observación 2.7, se obtiene la contradictoria desigualdad $\rho(P) > \rho(P)$. Dado que esto provino de suponer que z es distinto de cero, $z = 0$ y por ende $Px = \rho(P)x$. Notar que, como $Px > 0$, esto implica que $|y| = x > 0$.

Corolario 2.9 *Si $P \in MEP_n$ entonces, $\rho(P) \in \sigma(P)$ y existe $x \in \mathbb{R}^n$ tal que $x > 0$ y $Px = \rho(P)x$.*

Proposición 2.10 *Sean $P \in MEP_n$ y $\lambda \in \sigma(P)$ tales que $|\lambda| = \rho(P)$. Si $y \in \mathbb{C}^n \setminus \{\overrightarrow{0}\}$, cumple que $Py = \lambda y$, entonces, existe $\theta \in [0, 2\pi)$ tal que $y = e^{i\theta}|y|$, por lo que $\lambda = \rho(P)$.*

Demostración:

Por la Proposición 1 sabemos que $P|y| = \rho(P)|y|$. Además

$$|Py| = |\lambda y| = \rho(P)|y| \Rightarrow P|y| = |Py|.$$

Observando las primeras coordenadas, tenemos que

$$\sum_{i \in \mathbb{I}_n} P_{1j}|y_j| = |\sum_{i \in \mathbb{I}_n} P_{1j}y_j|.$$

Luego vale la igualdad en la desigualdad triangular, y cada y_j apunta hacia el mismo lado. Es decir, que debe de existir $\theta \in [0, 2\pi)$ tal que $y_j = e^{i\theta}|y_j|$ para todo $j \in \mathbb{I}_n$.

Corolario 2.11 *Si $P \in MEP_n$, entonces $\rho(P)$ es el único valor propio de módulo máximo*

Con lo estudiado anteriormente, ahora ya estamos listos para ver los siguientes teoremas.

Teorema 2.12 *Si P es una matriz positiva entonces, $\rho(P)$ es un valor propio simple (es decir, $\rho(P)$ tiene multiplicidad uno) mayor que cero.*

Demostración:

Como $P > 0$, existe $\varepsilon > 0$ tal que $\varepsilon I \leq P$. Así $\rho(P) \geq \rho(\varepsilon I) = \varepsilon > 0$.

Teorema 2.13 *Sea P una matriz de tamaño n x n real o compleja que tiene valores propios $\lambda_1, \lambda_2, ..., \lambda_n$ (no necesariamente distintos). Entonces para algún entero positivo k, los valores propios de P^k son $\lambda_1^k, \lambda_2^k, ..., \lambda_n^k$.*

Una matriz P se llama *regular* cuando para algún entero positivo k, todas las entradas de P^k son positivas.

Usando el Teorema 2.12 y 2.13 probaremos el siguiente teorema:

Teorema 2.14 *Si P es una matriz de transición regular, entonces existe un único vector de probabilidad s que es un vector propio para el valor propio 1, y si λ es un valor propio distinto de 1 entonces $|\lambda| < 1$.*

Demostración:

Sea P una matriz de transición regular. Por lo tanto, por el Corolario 2.5, sabemos que $\rho(P) = 1$. Dado que P es regular, las entradas de P^k son positivas, y por el Teorema 2.12 sabemos lo siguiente:

1) *$\rho(P^k)$ es un valor propio simple de P^k,*
2) *Si μ es un valor propio de P^k y $\mu \neq 1$, entonces $|\mu| < 1$,*
3) *El valor propio 1 de P^k tiene un vector propio positivo, x, y algún vector propio de 1 es un múltiplo de x.*

Por el Teorema 2.13, todos los valores propios de P^k son valores propios de P elevado a la k-ésima potencia, así

1) *1 es un valor propio simple de P,*
2) *Si $\lambda \neq 1$ es un valor propio de P, entonces $|\lambda| < 1$.*

Si x es un vector propio para 1, $Px = 1x = x$. Dado que 1 es un valor propio simple de P^k, al hacer la multiplicación por un escalar hay un único vector propio para 1, así hay un vector propio de P^k que es un vector propio de P y P tiene un vector propio positivo para 1 y si se divide por la suma de sus entradas, obtenemos que el único vector propio para 1 es un vector de probabilidad; el cual llamamos el vector s.

Teorema 2.15 *Si P es una matriz positiva y $\rho(P) < 1$, entonces $\lim_{k\to\infty} P^k = 0$.*

Con ayuda de los Teoremas 2.14 y 2.15, demostraremos lo siguiente,

Teorema 2.16 *Si P es una matriz estocástica regular entonces,*

$$\lim_{k\to\infty} P^k = \begin{pmatrix} s \\ s \\ \vdots \\ s \end{pmatrix}$$

Donde s es el único vector de probabilidad que es un vector propio para el valor propio 1.

Demostración:

Sea P una matriz estocástica regular. Por el Teorema 2.14, P tiene un único vector propio de probabilidad, s, para el valor propio 1 así, $Ps = s$. Sea

$$L = \begin{pmatrix} s \\ s \\ \vdots \\ s \end{pmatrix}$$

$$P \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

dado que las filas de P suman 1.

$$LP = \begin{pmatrix} sP \\ sP \\ \vdots \\ sP \end{pmatrix} = \begin{pmatrix} s \\ s \\ \vdots \\ s \end{pmatrix} = L$$

$$L = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} s$$

$$PL = P \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} s = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} s = L$$

$$L^2 = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} s \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} s = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} s = L$$

dado que

$$s \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = 1.$$

Además, $L(P-L) = LP - L^2 = L - L = 0$*, igualmente,* $(P-L)L = 0$ *y también* $(P-L)^k = P^k - L$ *ya que:*

Sea $k = 1$ $(P-L)^1 = (P-L) = P^1 - L$*. Asumimos que para* k*,* $(P-L)^k = P^k - L$ *es verdadera. Ahora veamos que se cumple para* $k+1$*,* $(P-L)^{k+1} = P^{k+1} - L$

$$\begin{aligned} (P-L)^{k+1} &= (P-L)^k(P-L) = (P^k - L)(P-L) \text{ por la suposición.} \\ &= P^k(P-L) - L(P-L) = P^k(P-L) - 0 = P^kP - P^kL \\ &= P^{k+1} - L. \end{aligned}$$

Ahora probemos que cada valor propio distinto de cero de $(P-L)$ *es un valor propio de* L*.*

Supóngamos que $(P-L)w = \mu w$*,* $\mu \neq 0$ *es valor propio,* $w \neq 0$ *vector propio.*
$\mu Lw = L(P-L)w = 0w = 0$*. Por que* $\mu \neq 0, Lw = 0$*. Así,* $\mu w = (P-L)w = Pw - Lw = Pw$ *y* λ *es valor propio para* P *con vector propio* w*. 1 no es un valor propio de* $(P-L)$ *por que* $(P-L)s = Ps - Ls = s - s = 0$*.*
Así, cualquier valor propio distinto de cero λ *de* $(T-L)$ *satisface* $|\mu| < 1$*. Por lo tanto,* $\rho(P-L) < 1$ *y por el Teorema 2.15,* $\lim_{k\to\infty}(P-L)^k = 0$*. Dado* $(P-L)^k = P^k - L$*,*

$$\lim_{k\to\infty}(P^k - L) = 0.$$

Por lo tanto, cuando k tiende a infinito,

$$\lim_{k\to\infty} P^k = L = \begin{pmatrix} s \\ s \\ \vdots \\ s \end{pmatrix}.$$

En seguida, probaremos que si una matriz estocástica P es regular, existe un único vector de distribución estacionaria para el valor propio 1, para cualquier vector de probabilidad inicial.

Teorema 2.17 *Si una cadena de Markov tiene una matriz de transición P regular, entoces, hay un único vector de probabilidad s tal que sP = s. Además, s, es tal que para cualquier vector de probabilidad incial q, la sucesión de vectores q, qP, qP^2,...,qP^k converge a s.*

Demostración:

Por el Teorema 2.16,

$$\lim_{k\to\infty} P^k = \begin{pmatrix} s \\ s \\ \vdots \\ s \end{pmatrix}.$$

Sea q un vector de probabilidad inicial, entonces,

$$\lim_{k\to\infty} qP^k = q \begin{pmatrix} s \\ s \\ \vdots \\ s \end{pmatrix}$$

$$= (q_1, q_2, ..., q_n) \begin{pmatrix} s_1 & s_2 & ... & s_n \\ s_1 & s_2 & ... & s_n \\ \vdots & \vdots & ... & \vdots \\ s_1 & s_2 & ... & s_n \end{pmatrix}$$

$$= (s_1q_1 + s_1q_2 + ... + s_1q_n, s_2q_1 + s_2q_2 + ... + s_2q_n, ..., s_nq_1 + s_nq_2 + ... + s_nq_n)$$

$$= (s_1(q_1 + q_2 + ... + q_n), s_2(q_1 + q_2 + ... + q_n), ..., s_n(q_1 + q_2 + ... + q_n)$$

Por la definición de q

$$\sum_{i=1}^{n} q_i = 1.$$

Por lo tanto,

$$= (s_1(\sum_{i=1}^{n} q_i), s_2(\sum_{i=1}^{n} q_i), ..., s_n(\sum_{i=1}^{n} q_i)$$

$$= (s_1(1), s_2(1), ..., s_n(1))$$

$$= (s_1, s_2, ..., s_n)) = s.$$

De donde se puede concluir que,

$$\lim_{k\to\infty} qP^k = s.$$

Capítulo 3

PageRank y Cadenas de Markov

3.1. Introducción

En mayo de 2005, una consulta en internet usando el motor de búsqueda *Google* informaba que se estaban realizando peticiones sobre un total de 8,085 millones de páginas. Otra búsqueda, esta vez realizada el 26 de octubre de 2006, permite estimar que, al menos, hay unos 24,640 millones de páginas web indexadas por *Google*. Hoy en dia, se estima que hay aproximadamente unos 3 billones de páginas [7]. Estos datos dan una idea del tamaño y la velocidad de crecimiento de internet. Si a esto unimos las posibilidades de negocio mediante anuncios publicitarios que el mismo *Google* promueve, tenemos que es fundamental disponer de un sistema de clasificación de páginas rápido y fiable, para poner orden en toda esta magnitud de datos que ha ido creciendo rápidamente para un análisis sobre la estructura de la World Wide Web.

Google utiliza un programa llamado *PageRank* para priorizar las páginas que se encuentran en una búsqueda; esto es importante por que una búsqueda por lo general regresa ahora más páginas de las que el buscador está dispuesto a mirar. *PageRank* fue desarrollado en 1998 por Larry Page y Serger Brin cuando eran estudiantes de posgrado de ciencias de la computación en la Universidad de Stanford.

El *PageRank*, el método inicial de cálculo que usaron los fundadores de *Google* para clasificar las páginas web según su importancia, es objeto de constantes mejoras. La finalidad del método es la obtención de un vector, también llamado *PageRank*, que da la importancia relativa de las páginas. Dado que el vector *PageRank* se calcula en función de la estructura de las conexiones de la web se dice que es independiente de la petición de la persona que realiza la búsqueda.

Pero, ¿Cómo hacer para obtener el vector *PageRank*?, ¿Habrá un método que nos ayude a calcularlo? y principalmente, ¿Cómo aseguramos que este vector, realmente muestra el orden en que se clasifican las páginas?. Para ello, se estudiará la matriz de

Google, la cual, tiene ciertas propiedades matemáticas, por ejemplo: es una matriz cuadrada, con entradas positivas y cuyas filas suman uno. Eso nos da una idea de que herramienta Matemática podemos utilizar, a saber, las cadenas de *Markov*. ¿Por qué usar cadenas de *Markov*?. Para responder a esto, tomemos en cuenta lo siguiente: Cuando una persona se encuentra en una página, tiene la misma probabilidad de elegir cualquier enlace saliente, además, cada que realizamos una nueva búsqueda *Google*, recuerda solo la búsqueda anterior, sin importar las que hay detrás de élla, además la matriz es estocástica, por lo tanto, mostraremos que cumple con ciertas propiedades que nos ayudaran a obtener un vector de distribución estacionario, que además, demostraremos que es único.

¿Qué es el *PageRank*?

Es un valor numérico que representa la importancia que una página tiene en internet. *Google* "se hace a la idea"de que cuando una página coloca un enlace (link) a otra, es también un "voto"para esta última.

Cuantos más votos tenga una página, será considerada más importante por *Google*. Además, la importancia de la página que emite su voto también determina el peso de éste. De esta manera, *Google* calcula la importancia de una página gracias a todos los votos que recibe, teniendo en cuenta también la importancia de cada página que emite su voto.

¿Cómo ordenar las páginas en la red?

Suponga un pequeño universo de 4 páginas: A, B, C, D. Si todas esas páginas enlazan a A, entonces el PR (*PageRank*) de la página A sería la suma del PR de las páginas B, C y D.

$$PR(A) = PR(B) + PR(C) + PR(D).$$

Pero supóngase que B también tiene un enlace a C y que D tiene enlaces a las otras 3 páginas. Una página no puede votar 2 veces, y por esa razón se considera que B da medio voto a A y medio voto a C. De la misma manera sólo un tercio del voto de D se cuenta para el de A:

$$PR(A) = PR(B)/2 + PR(C) + PR(D)/3.$$

En otras palabras, divídase el PR entre el número total de enlaces que salen de esa página.

$$PR(A) = PR(B)/C(B) + PR(C)/C(C) + PR(D)/C(D).$$

Las páginas web varían mucho en el número de vínculos entrantes que poseen. Generalmente las páginas que tienen muchos vínculos entrantes son más importantes

que las que sólo tienen unos pocos. Sin embargo, hay muchos casos en los cuales sólo el contar el número de vínculos entrantes no se corresponde con el sentido usual de la importancia de una página web. Como escribían Brin y Page: Si una página tiene un vínculo de la página principal de Yahoo, éste puede ser un solo vínculo pero uno muy importante. Dicha página debería estar mejor clasificada que otras páginas con muchos vínculos pero de lugares desconocidos.

El *PageRank* de una página se define como: $PR_j = \sum_{i \in l_j} \frac{PR_i}{|O_i|}$ en donde PR_i es el *PageRank* de la página i y O_i es el número de enlaces salientes de la página i, esto lo podemos escribir en forma matricial de la siguiente manera $\pi = \pi P$, donde π es el vector *PageRank* y P es la matriz de transición de la web. Cada vez que nosotros hacemos una búsqueda en internet, *Google* recuerda solo la búsqueda anterior sin importar las demás, esto quiere decir que, nosotros iniciamos con un valor de *PageRank* π_0, si seguimos navegando en internet, al hacer una nueva búsqueda, el siguente valor de *PageRank* seria $\pi_1 = \pi_0 P$, este proceso lo podemos escribir $\pi_{k+1} = \pi_k P$ [5], en donde π_k es el vector *PageRank*, que nos dice la importancia de cada página.

Una página tiene una clasificación alta si la suma de las clasificaciones de sus vínculos entrantes es alto. Esto cubre ambos casos: muchos vínculos entrantes o pocos con alta clasificación. El algoritmo original del *PageRank* fue descrito en varios trabajos de Brin y Page [3]. Posteriormente presentaron una versión mejorada, que es la que expondremos. El propósito es cuantificar la probabilidad de que un usuario (aleatorio) llegue a la página A utilizando la Red. Se define el *PageRank* por [3]:

$$PR(A) = \frac{(1-\alpha)}{N} + \alpha \left(\frac{PR(T_1)}{C(T_1)} + ... + \frac{PR(T_n)}{C(T_n)} \right)$$

en donde:

N: es el número total de páginas web desde las que salen vínculos.

n: es el número total de páginas web desde las que salen vínculos a la página A.

$PR(T_i)$: es el PR de las páginas T_i que tienen un vínculo hacia la página A.

$C(T_i)$: es el número de vínculos salientes de la página T_i.

α : es un factor de amortiguación que puede ser tomado entre 0 y 1.

Como la suma de los PR de todas las páginas web es uno, es a su vez una distribución de probabilidad (indexada por el parámetro α). Esta normalización (suma = 1) facilita la utilización de resultados generales que no dependen del tamaño del sistema (el número total de páginas).

Analizando con cuidado dicha fórmula se observarán las siguientes características del *PageRank*:

- está definido para cada página y es determinado por los *PageRanks* de las páginas que tienen un vínculo dirigido hacia ella.
- los sitios que enlazan a la página A no influyen uniformemente; depende del número de vínculos salientes que ellas posean: a más vínculos salientes de una página menos beneficiará el *PageRank* de las páginas a las que se una.
- un nuevo vínculo a una página siempre aumenta su valor.
- la definición es recursiva: la clasificación de una página depende de todas las otras que tienen vínculos hacia ella, por ello, la clasificación de cada página depende de todos los sitios de la Red.

En sus explicaciones Brin y Page dan una justificación sencilla para el algoritmo. El *PageRank* modela el comportamiento de un usuario que estando en una página puede:

- elegir al azar entre los vínculos contenidos en la página actual.
- saltar al azar a cualquier página de la red ingresando la dirección; todo ello sin tener en cuenta el contenido de los mismos (esto ha suscitado comentarios y modelos alternativos). Cuantificando esos comportamientos posibles, se supone que seguirá un enlace de la página en que está con probabilidad α, o que salta a cualquier página con probabilidad $1 - \alpha$.

La definición del *PageRank* establece un procedimiento para determinar la probabilidad de que un usuario aleatorio llegue a cierta página web. El navegante aleatorio visita una página web con una probabilidad proporcional al *PageRank* de la página. La probabilidad de elegir un vínculo depende de los vínculos que se pueden elegir en la página en que se está.

El modelo no tiene en cuenta para nada el contenido de las páginas. Se supone que es más probable que siga uno de los enlaces de la página en que está; de hecho, trabajan con un parámetro de 0.85 (85 %). Esta probabilidad la representan con la letra α y la probabilidad de que teclee una dirección sin usar uno de los enlaces disponibles es, por lo tanto, $1 - \alpha$, en este caso, 0.15 (el restante 15 % de las veces) [8]. La probabilidad de elegir uno de los vínculos salientes entre los que figuran en la página se distribuye uniformemente entre la cantidad que allí haya. Ahora veamos; para el 15 % de casos en que el usuario digita la dirección, la probabilidad de que llegue a T_i es de uno sobre el total de páginas web (N).

Ejemplo:

Veamos ahora cómo es el procedimiento recursivo en un ejemplo dado por el siguiente diagrama.

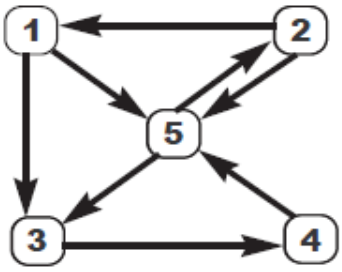


Figura 3.1

Tenemos 5 páginas web e indicamos con una flecha los vínculos. Por ejemplo, de la página 1 salen vínculos a la 3 y 5, y entra un vínculo de la página 2.

Veamos las fórmulas de *PageRank* de una manera más compacta, intentando utilizar la nomenclatura probabilística relacionada con la distribución estacionaria de una cadena de *Markov*. Llamamos $\pi_i = PR(i)$ al *PageRank* de la página i:

$\pi_1 = \frac{1-\alpha}{5} + \alpha(\frac{\pi_2}{2})$

$\pi_2 = \frac{1-\alpha}{5} + \alpha(\frac{\pi_5}{2})$

$\pi_3 = \frac{1-\alpha}{5} + \alpha(\frac{\pi_1}{2} + \frac{\pi_5}{2})$

$\pi_4 = \frac{1-\alpha}{5} + \alpha(\pi_3)$

$\pi_5 = \frac{1-\alpha}{5} + \alpha(\frac{\pi_1}{2} + \frac{\pi_2}{2} + \pi_4)$

Si definimos la matriz:

$$P = \frac{1-\alpha}{5}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} + \alpha \begin{pmatrix} 0 & 0 & 1/2 & 0 & 1/2 \\ 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1/2 & 1/2 & 0 & 0 \end{pmatrix}$$

y $\pi = (\pi_1,\pi_2,\pi_3,\pi_4,\pi_5)$ utilizando que $\sum_1^5 \pi_i = 1$ podemos resumir las 5 ecuaciones en $\pi = \pi P$.

Modelo de navegación

Una de las características del *PageRank* es que si uno navega aleatoriamente por internet y está un tiempo suficientemente grande paseando, entonces tendrá una gran probabilidad de encontrar las páginas con mayor *PageRank*.

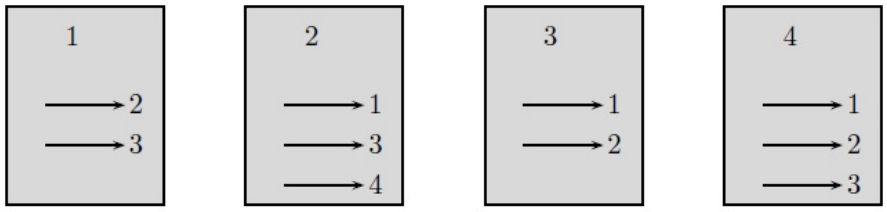


Figura 3.2: Una web con cuatro páginas mostrando sus enlaces salientes.

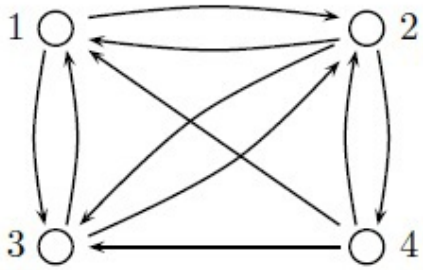


Figura 3.3: Grafo dirigido correspondiente a la web de la Figura 3.2.

Dado un conjunto de n páginas web definimos su matriz de conectividad G como la matriz cuadrada de orden n cuyos elementos denominados $g_{i,j}$, $1 \leq i, j \leq n$, valen 1 si hay enlace de la página i a la página j, con $i \neq j$, y 0 en otro caso.

$$G = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Imaginemos que tenemos una persona que va saltando de manera aleatoria de unas páginas a otras. Queremos construir una matriz de transición P, cuyos elementos den las probabilidades condicionadas de salto. Para ello, vamos a asumir que, cuando se encuentra en una página, tiene la misma probabilidad de elegir cualquier enlace saliente. Esta elección es la base del modelo de Brin y Page.

Desde el punto de vista del cálculo es muy fácil obtener la matriz P a partir de la matriz G: basta dividir cada fila de G por la suma de los elementos de G en dicha fila, siempre que esta cantidad no sea cero, es decir, siempre que esta fila no

corresponda a una página sin salida.

$$p_{i,j} = \begin{cases} \frac{g_{i,j}}{O_i} & \text{si hay un enlace de i a j} \\ 0 & \text{en otro caso} \end{cases}$$

Si $O_i \neq 0$ para todo i, entonces P es una matriz estocástica por filas, es decir, la suma de cada fila vale 1 y cada elemento toma un valor entre 0 y 1. Usando la red de la Figura 3.2, contruimos la siguiente matriz de transición.

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

Nodos colgantes

Analicemos la siguiente matriz

$$P = \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/4 & 1/4 & 0 & 1/4 & 1/4 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 0 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Cuando tenemos una matriz con una fila de ceros, indica que tenemos una página sin vínculos salientes. Este tipo de página (nodo) se llama un *Nodo colgante*. Los cuales, representan un problema cuando se trata de establecer un modelo de *Markov*. Para resolver este problema, se sustituyen dichas filas con $\frac{e}{n}$, donde e es un vector fila de unos y n es el orden de P. Así, creamos una nueva matriz.

$$\bar{P} = \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/4 & 1/4 & 0 & 1/4 & 1/4 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 0 & 1/3 & 0 & 1/3 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix}$$

Pero ser estocástica no es suficiente para garantizar que nuestro modelo de *Markov* convergerá y que existe una distribución estacionaria. El otro problema que enfrenta nuestra matriz de transición P, y cualquier matriz de transición creada para la web, es que la matriz puede no ser regular. La naturaleza de la web es tal que P no sería regular, así que tenemos que hacer más ajustes. Brin y Page obligan a la matriz de transición a ser regular, asegurandose que cada entrada satisface que $0 \leq p_{i,j} \leq 1$ [6]. Esto garantiza la convergencia de π^k a un vector de probabilidad.

Matriz de *Google*

Según Langville y Meyer [1], Brin y Page añaden una matriz de perturbación $E = v.e^T$, para formar lo que generalmente se llama la "*Matriz de Google*".

$$Q = \alpha\bar{P}+(1-\alpha)E, \text{ para algún } 0 \leq \alpha \leq 1. \tag{3.1}$$

Donde v es el llamado vector de personalización o de teleportación y es un vector de distribución de probabilidad que suele tomarse como $v = \frac{e}{n}$. El producto $v.e^T$ es una matriz de orden n. El parámetro α se denomina de amortiguamiento y se suele tomar $\alpha = 0{,}85$, ya que fue el que usaron originalmente Brin y Page. Se dice que en la ecuación 3.1, la matriz Q es una combinación lineal convexa de las matrices $\bar{P}$ y E. El término $(1-\alpha)v.e^T$, con v un vector de distribución de probabilidad, da lugar a que todos los elementos de Q sean no nulos, con lo cual, Q es irreducible.

El efecto estadístico de este término es introducir saltos aleatorios que no dependan de las propiedades de enlace de la página. Valores de α próximos a uno ofrecen comportamientos más realistas pero pueden arruinar la irreducibilidad (en el límite $\alpha = 1$) y aumentar el número de iteraciones del método de la potencia. Nótese que $\alpha = 1$ correspondería a usar la matriz de conectividad real de la web.

Para el ejemplo antes escrito usando $\alpha = 0{,}85$ para la matriz $\bar{P}$ podemos calcular la *Matriz de Google.*

$$Q = 0{,}85\bar{P} + (1 - 0{,}85)E$$

$$= 0{,}85\begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/4 & 1/4 & 0 & 1/4 & 1/4 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 0 & 1/3 & 0 & 1/3 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix} + 0{,}15\begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix}$$

$$= \begin{pmatrix} 1/40 & 37/120 & 37/120 & 37/120 & 1/40 & 1/40 \\ 9/20 & 1/40 & 9/20 & 1/40 & 1/40 & 1/40 \\ 19/80 & 19/80 & 1/40 & 19/80 & 19/80 & 1/40 \\ 37/120 & 1/40 & 1/40 & 1/40 & 37/120 & 37/120 \\ 1/40 & 37/120 & 1/40 & 37/120 & 1/40 & 37/120 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix}$$

¿Cómo funciona *PageRank* usando el vector estacionario?

Hacer el cálculo de potencias de la matriz de transición Q (Teorema 2.16), es una forma en que podemos determinar el vector estacionario. Por ejemplo:

$$Q^{25} = \begin{pmatrix} 0{,}2066 & 0{,}1770 & 0{,}1773 & 0{,}1770 & 0{,}1314 & 0{,}1309 & 0{,}1309 \\ 0{,}2066 & 0{,}1770 & 0{,}1773 & 0{,}1770 & 0{,}1314 & 0{,}1309 & 0{,}1309 \\ 0{,}2066 & 0{,}1770 & 0{,}1773 & 0{,}1770 & 0{,}1314 & 0{,}1309 & 0{,}1309 \\ 0{,}2066 & 0{,}1770 & 0{,}1773 & 0{,}1770 & 0{,}1314 & 0{,}1309 & 0{,}1309 \\ 0{,}2066 & 0{,}1770 & 0{,}1773 & 0{,}1770 & 0{,}1314 & 0{,}1309 & 0{,}1309 \\ 0{,}2066 & 0{,}1770 & 0{,}1773 & 0{,}1770 & 0{,}1314 & 0{,}1309 & 0{,}1309 \end{pmatrix}$$

(A cuatro digitos significativos), podemos deducir que πQ^k está convergiendo a los valores que se muestran en cada fila.

Por lo tanto, el vector estacionario para nuestra matriz es; por el Teorema 2.17

$$s = (0{,}2066, 0{,}1770, 0{,}1773, 0{,}1770, 0{,}1314, 0{,}1309).$$

Supongamos que un usuario introduce una consulta en la ventana de búsqueda de *Google*. Supongamos que dicha consulta tiene dos palabras, entonces queremos saber que páginas están relacionadas con cada una de estás palabras.

$$palabra1 \longrightarrow pag2, pag5, pag6$$
$$palabra2 \longrightarrow pag2, pag3$$

Por lo tanto, las páginas relacionadas con las palabras que introduce el usuario son $\{2, 3, 5, 6\}$. El *PageRank* de estás páginas ahora se compara para determinar el orden de importancia. De acuerdo con nuestro modelo quedan ordenados de la siguiente manera:

$s_2 = 0{,}1770, s_3 = 0{,}1773, s_5 = 0{,}1314, s_6 = 0{,}1309.$

Se considera más importante la página que contenga todas las palabras introducidas, después, ordenamos las demás de acuerdo a su *PageRank*, el cual, estima que la página 3 es más importante, seguido por la 5 y la página 6. Cuando una nueva consulta es introducida, se accede de nuevo y se crea un nuevo conjunto de páginas.

Por lo visto anteriormente, la matriz de *Google*, es irreducible y con entradas positivas, además es una matriz estocástica, que tiene a 1 como valor propio, el cual, es el radio espectral y por el Teorema de Perron Frobenius [4], a este valor propio le corresponde un único vector propio que es una distribución de probabilidad. Esto quiere decir que, el vector *PageRank*, será el único vector propio asociado al valor propio 1.

De acuerdo con el sitio web de *Google*, *PageRank* sigue siendo el corazón de *Google*. Tenga en cuenta, sin embargo, que el *PageRank* no es el único criterio que utiliza para clasificar la importancia de una página web [6]. *Google* es un motor de busqueda de texto completo, que utiliza *PageRank* así como otros factores para clasificar resultados de búsqueda.

Capítulo 4

Simulación del *PageRank*

Cuando realizamos una búsqueda en *Google*, aparece una lista enorme de páginas relacionadas con las palabras que introducimos al momento de navegar en internet, las cuales, tienen demasiados vínculos salientes hacia otras páginas, éstas a su vez, también contienen muchos vínculos entrantes y salientes. Por ello, en este trabajo se presentan dos programas que nos ayudarán a hacer simulaciones del funcionamiento del algoritmo. El primer programa, clasifica las páginas web, de acuerdo a los enlaces entrantes y salientes a estas páginas.

Ejemplo:

Para propósitos ilustrativos, consideremos primero una matriz de conectividad de orden 3.

```
A =

   0   1   1
   1   0   0
   1   0   1

T =

   0.00000   0.50000   0.50000
   1.00000   0.00000   0.00000
   0.50000   0.00000   0.50000

la matriz de Google para la web es:B =

   0.050000   0.475000   0.475000
   0.900000   0.050000   0.050000
   0.475000   0.050000   0.475000


M =

   0.39879   0.21949   0.38172
   0.39879   0.21949   0.38172
   0.39879   0.21949   0.38172

el vector PageRank es:

I =

   1   3   2
```

Generamos una matriz de conectividad aleatoria de 3 páginas, donde la página 1 tiene enlaces hacia las páginas 2 y 3, la página 2, va solamente a la página 1 y la página 3, tiene enlaces hacia la página 1 y ella misma.

Hacemos que la matriz de conectividad sea estocástica de manera uniforme por filas.

Se calcula la matriz de Google para la web.

Se realiza el calculo de potencias, para encontrar el vector *PageRank*.

Una vez que hacemos el calculo de potencias, ordenamos nuestros datos de mayor a menor.

Por lo tanto, la página 1 se considera más improtante, seguido de la página 3 y, por último, la página 2.

Si ahora, generamos una web con 100 páginas, los resultados obtenidos son los siguientes:

```
el orden de relevancia es:

I =

 Columns 1 through 13:

    17    22    75    72    79    96    85    62    65    82    57    69    45

 Columns 14 through 26:

    14    87    34    36    30    19    84    11    35    90    38     6    44

 Columns 27 through 39:

    86    60    55    41    63    43    64    89    32    67    56    68    40

 Columns 40 through 52:

    10    54    97     4    93    24    18    71    92    12    51    78    66

 Columns 53 through 65:

    16    25     7    42     3    33    23    88    70    26    15    20     5

 Columns 66 through 78:

    46     8    98    39    91    29    48    81    95    59    61    13   100

 Columns 79 through 91:

    77     1    94    31    28    52    99    27    21    49     2    76    83

 Columns 92 through 100:

    50    37    74    53     9    47    73    80    58
```

Veamos un ejemplo de *nodos colgantes*:

```
A =

   1   0   1
   0   0   0
   0   1   1
```

Generamos una matriz de conectividad aleatoria de 3 páginas, donde la matriz de conectividad tiene un *Nodo Colgante*.

```
T =

   0.50000   0.00000   0.50000
   0.33333   0.33333   0.33333
   0.00000   0.50000   0.50000
```

Hacemos que la matriz de conectividad sea estocástica, lo cual, se logra dividiendo cada fila entre la suma de ésta y reemplazamos el vector nulo por el vector $\frac{e}{N}$.

```
la matriz de Google para la web es:B =

   0.34750   0.30500   0.34750
   0.33333   0.33333   0.33333
   0.30500   0.34750   0.34750
```

Se calcula la matriz de Google para la web.

```
M =

   0.32827   0.32889   0.34284
   0.32827   0.32889   0.34284
   0.32827   0.32889   0.34284
```

Se hace el cálculo de potencias, para hallar el vector *PageRank*.

```
el orden de relevancia es:

I =

   3   2   1
```

Una vez que hacemos el cálculo de potencias, ordenamos nuestros datos de mayor a menor.

El segundo programa, consiste en clasificar a las páginas de acuerdo a las palabras que se relacionen con la búsqueda realizada y después, las ordena de acuerdo a su relevancia.

Ejemplo:

Supongamos que contamos con un universo de 4 palabras, y supongamos una web de 5 páginas.

```
busq =

   1   1   0   1
```

Se establece la búsqueda de 3 palabras.

```
B1 =

   0   1   1   1
   1   0   0   0
   0   1   1   0
   1   1   0   1
   1   0   0   0
```

Se verifica la pertenencia de cada palabra a cada página.

```
C1 =

   2
   1
   1
   3
   1
```

Número de palabras que tiene cada página con respecto a la búsqueda.

```
D1 =

   1
   1
   1
   2
   3
```

Ordenamos de menor a mayor el número de palabras.

```
J1 =

   4
   1
   5
   3
   2
```

Ordenamos las páginas de acuerdo a su *PageRank*

Los programas presentados en este trabajo, sirven para ejemplos más grandes. En cuanto al tiempo de ejecución de los programas después de hacer algunas pruebas, encontramos que para matrices de conectividad de tamaño 2-100 es de 1.5 - 25.43 segundos, para matrices de tamaño 100 - 4000 de 2 - 7.5 segundos y para matrices mayores a 4000, lo recomendable es contar con una memoria RAM mayor o igual a 4GB, ya que después de realizar varias ejecuciones de los programas, puede suceder que la computadora se apague inesperadamente si no se cuenta con el tamaño adecuado de RAM.

PageRank, hasta el momento, es el método empleado por *Google* para clasificar las páginas de acuerdo a su importancia, como vimos, no solo se puede hacer de acuerdo a los enlaces entrantes y salientes, si no que también conforme al número de palabras que contiene la búsqueda realizada. *Google*, es empleado por millones de personas, por ello, se requiere de algoritmos computacionales que se encarguen de ofrecer resultados satisfactorios a los usuarios de *Google*, además, de ser una de tantas aplicaciones de las cadenas de *Markov*.

Conclusiones

Google, es el sitio más visitado por millones de personas, por lo cual, requiere de algoritmos como el *PageRank* para brindar un buen servicio a sus usuarios y proporcionar páginas con la información que requieren.

Por otro lado, *PageRank*, ha sido el centro de *Google* pero, no es el único algoritmo que utiliza para clasificar las páginas de acuerdo a su importancia [6], sin embargo, el más conocido es el *PageRank*. Estos algoritmos son los responsables de que *Google* sea popular entre los sitios de búsqueda, ya que estamos seguros de que las páginas que nos muestra, son páginas importantes y que tienen la información necesaria o suficiente que el usuario solicita.

Sin duda, el éxito del *PageRank* se lo debemos a las matemáticas, en especial a las cadenas de *Markov*, pues la matriz de *Google*, por lo estudiado, tiene propiedades, por ejemplo: ser una matriz estocástica, una cadena irreducible, recurrente positiva y aperiódica, y por lo tanto, converge a una distribución estacionaria (Teorema 1.3), la cual, es un vector propio asociado al valor propio 1, que es el *PageRank*, gracias a que cumple con todas las propiedades ya mencionadas, podemos estar seguros que el *PageRank*, es un buen algoritmo para el ordenamiento de las páginas web, ya que las cadenas de *Markov* nos aseguran que este vector existe y es único.

Es importante construir programas computacionales que nos ayuden a hacer simulaciones del algoritmo *PageRank*, para matrices de cualquier tamaño para entender su funcionamiento y la efectividad de este mismo. Por ello, se presentan dos programas los cuales, nos ayudan al cálculo del *PageRank*.

Apéndice A

Cálculo del vector PageRank

Código en Matlab
N=input('ingrese el tamaño de la matriz');
A= binornd(1,0.5,N,N);
for l=1:N
if(A(l,:)==zeros(1,N))
A(l,:)=1;
end
end
aa=1./sum(A');
T= bsxfun(@times,A,aa')
s=0.85;
E=ones(N,N);
fprintf('la matriz de Google para la web es:');
B=s*T+((1-s)/N)*E
i=100;
$M = B^i$;
fprintf('el vector PageRank es:');
C=M(1,:);
$[D, I]$ = sort(C);
$I = fliplr(I)$

Apéndice B

Cálculo del vector PageRank

Código en Matlab
busq=binornd(1,pp,1,R);
B1=binornd(1,ppp,N,R);
C1=B1*busq';
$[D1, I1] = sort(C1)$;
J1=I1(end:-1:1);
WWW=[ ];
for i=1:R
W=find(C1==R-i+1)
q=size(W);
if q(1)>0
WW=[ ];
for k=1: max(size(W))
WW=[WW find(J==W(k))];
WW=sort(WW);
end
WWW=[WWW WW];
end
end
J(WWW)

Bibliografía

[1] PEDROCHE F., *Métodos de cálculo del vector PageRank*, Univerisdad Politécnica de Valéncia, 2004.

[2] HOEL PAUL G., PORT C. SIDNEY, STONE CHARLES J., *Introduction to Stochastic Processes*, Houghton Mifflin Company, University of California, Los Angeles, 1972.

[3] MARKARIAN ROBERTO, MÖLLER NELSON, *Como cuantificar la importancia individual en una estructura de enlaces: Google-PageRank*, Universidad de la Republica, Uruguay, 2004.

[4] ANTEZANA JORGE, STOJANOFF DEMETRIO, *Análisis Matricial*, Departamento de Matemática, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, 2009.

[5] RINCÓN LUIS, *Introducción a los Procesos Estocásticos*, Departamento de Matemáticas, Facultad de Ciencias UNAM, Enero 2012.

[6] ATHERTON REBECCA, *A Look at Markov Chains and Their Use in Google*, Iowa State University, Summer 2005.

[7] ¿Cuántas búsquedas se hacen en Google?
http://www.seobasico.com/cuantas-busquedas-hacen-google/

[8] MARKARIAN ROBERTO, MÖLLER NELSON, *Cómo ordena el buscador Google sus resultados*, Correo del Maestro No. 105, Febrero 2005.
http://www.correodelmaestro.com/anteriores/2005/febrero/2anteaula105.htm