Universidad Nacional Autónoma de México

Facultad de Ciencias

Hallando la página web más relevante;
Algoritmo PageRank

Probabilidad I

Diana Laura Nicolás Pavía

Ciencias de la Computación

Proyecto presentado como parte del curso de **Probabilidad I** impartido por el profesor **Marco Arieli Herrera Valdez** con **Carlos Ignacio Herrera Nolasco** como ayudante.

21 de noviembre del 2019

Link al código fuente: https://github.com/nicolasdi/PageRank

1. Introducción

Nos encontramos en una era en donde la información viaja de forma instantánea, un hecho impensable para la humanidad hace 100 años. En su mayor parte este hecho es gracias al internet ya que a través de él hacemos uso de servicios que nos permiten compartir y consultar información.
Actualmente la cantidad de sitios en internet es superior a los 1.9 billones, esa cantidad de información podría ser un arma de doble filo; es verdad que una cantidad grande podría sugerir diversidad en ella aunque también podría sugerir información repetida o de mala calidad.
Es por eso que son valiosos los algoritmos para filtrar la información de una forma adecuada.

El objetivo de este trabajo es, basados en el algoritmo PageRank, construir un modelo que nos ayude a saber qué página de internet es las más relevante, abriendo paso con lo anterior generar un algoritmo tal que su *output* sea la página más relevante dentro de una base de datos.

En la primera sección del trabajo se encuentra una situación hipotética que dará pie a la construcción del modelo. Luego se da un marco teórico que describe detalles necesarios para la resolución del problema y conceptos nuevos de utilidad además de dos enmarques para entender la complejidad e importancia del ploblema que se intenta atacar a pequeña escala.
Una vez entrados, se construye una solución para nuestra situación hipotética y finalmente se da un algoritmo que soluciona algunos casos particulares del problema inicial.
Se da una pequeña descripción del algoritmo junto con un análisis sobre alcances y limitaciones para dar al fin una conclusión y alguna pequeña reflexión que estuvo presente durante la realización del trabajo.

2. Motivación

2.1. Planteamiento del problema

Supongamos el caso de una estudiante que cuenta con acceso a internet, el costo por acceder a una página de internet es altísimo ¡el precio es su tiempo!, ella quiere seleccionar con cuidado a qué pagina desea entrar para pagar el menor costo posible y obtener la información que necesita para su tarea.
Intenta pensar en cómo funciona internet, sabe varias cosas, una de ellas es que existen páginas que tienen muy buena información mientras que hay otras con información de poca calidad además es común que las páginas de internet se referencien unas a otras. A ella le gustaría saber qué página tiene la información de mejor calidad sin tener que entrar ella misma a hacer una evaluación y así obtener de una forma rápida la información que necesita. A continuación se intenta construir un modelo que nos ayude a averiguar, dentro de un conjunto de sitios, qué página es más importante.

3. Marco Teórico

3.1. Cadenas de Markov

Se supone que se cuenta con conceptos básicos de Teoría de Probabilidad, como espacio muestral (Ω), sigma álgebra ($\mathscr{A}$), medida de probabilidad ($\mathbf{P}$), independencia de eventos, probabilidad condicional y función de distribución.

Un *proceso estocástico* puede ser definido como cualquier colección de variables aleatorias, las denotamos como $(X_0, ..., X_n)$ para $n \in N$.

Definición 1. Sea S un conjunto finito o contable y $(\Omega, \mathscr{A}, P)$ un espacio de probabilidad. Una secuencia $S - valuada$ de variables aleatorias $(\mathrm{X}_0, X_1, ..., X_n)$ n $\in N$, es llamada una *cadena de Markov* $S - valuada$ o *cadena de Markov en* S si para todo $n \in N$ y para todo $s_i \in S$ se cumple que

$$P(X_{n+1} = s_{i_j}) = P(X_{n+1} = s_{i_0} | X_0 = s_{i_1}, X_1 = s_{i_2}, ..., X_n = s_{i_n})$$

y lo anterior es igual a

$$P(X_{n+1} = s_{i_j} | X_n = s_{i_n})$$

Una forma de interpretar lo anterior es que las cadenas de Markov está definidas como una sucesión de estados y que la medida de probabilidad que se usa para los valores que puede tomar la variable aleatoria X_{i+1} dependen del valor que tomó la variable aleatoria en el estado X_i. Dicho de otra forma, basta con saber el estado actual para saber qué probabilidad toma ir a un siguiente estado particular.

Definición 2. Una cadena de Markov $S - valuada$ $(X_0, ..., X_n)$ para $n \in N$ es llamada *homogénea temporal* u *homogenea* si para todo $n \in N$ y para todo $i, j \in S$ se cumple que

$$P(X_{n+1} = s_j | X_n = s_i) = P(X_0 = s_j | X_0 = s_i)$$

Lo que significa que cuando se está en un estado, las probabilidades para ir al siguiente estado ya están definidas y no cambian conforme la sucesión de variables aleatorias (cadena de Markov) crece.

Definición 3. Una $\mathbf{P} \in \mathscr{M}_{k \times k}$ es una *matriz estocástica* si cumple

- Para cada $i, j \in \{1, ..., k\}$, $P_{i,j} \geq 0$
- $\forall i \in \{1, ..., k\} \sum_{j=1}^{k} P_{i,j} = 1$

Definición 4. Decimos que una matriz $\mathbf{P} \in \mathscr{M}_{k \times k}$ es una **matriz de transición** para la cadena de Markov homongénea si para todo n, para todo $i, j \in \{1, ..., k\}$ y para todo $i_0, ..., i_{n+1} \in \{1, ..., k\}$ tenemos que

$$P(X_{n+1} = s_{i_{n+1}}) = P(X_{n+1} = s_{i_{n+1}} | X_0 = s_{i_0}, X_1 = s_{i_1}, ..., X_n = s_{i_n}) = \\ \mathrm{P}(\mathrm{X}_{n+1} = s_{i_{n+1}} | X_n = s_{i_n}) = P_{i_n, i_{n+1}}$$

Los elementos de una matriz de transición se llaman probabilidades de transición donde $P_{i_n, i_{n+1}}$ denota la probabilidad de pasar al estado i_{n+1} dado que ya se está en el estado i_n.

Definición 5. Sea $|S| = k$, una *distribución inicial* denotada por μ^0 es un vector tal que
$\mu^0 = (\mu_1^0, ..., \mu_k^0)$
$\mu^0 = (P(X_0 = s_1), P(X_0 = s_2), ..., P(X_0) = s_k)$

Es decir es el vector que nos dice qué probabilidad tiene nuestra cadena de Markov de iniciar en cada estado de su conjunto posible de estados.

Y en general para la cadena de Markov $(X_0, ..., X_n)$ el vector $\mu^i = (\mu_1^i, ..., \mu_k^i)$ denota la distribución de probabilidad de la variable aleatoria X_i de la cadena.

Teorema 1. Para una cadena de Markov homogénea $(X_0, X_1, ..., X_n)$ $n \in N$ con espacio de estados S=$\{s_1, ..., s_k\}$, distribución inicial μ^0 y matriz de transición P, tenemos que para cualquier n la distribución μ^n para el tiempo n satisface que

$$\mu^n = \mu^0 * P^n$$

Definición 6. Decimos que una cadena de Markov $(X_0, X_1, ...)$ con estados $S = \{s_1, ..., s_k\}$ es *irreducible* si cumple que para todo s_i , $s_j \in S$ tenemos que existe una probabilidad estrictamente positiva de ir del estado s_i al estado s_j en alguna cantidad positiva de pasos y viceversa.

Definición 7. Sea una cadena de Markov $(X_0, X_1, ...)$ con estados $S = \{s_1, ..., s_k\}$ diremos que un estado $s_i \in S$ tiene periodo $d(s_i)$ si

$$d(s_i) = mcd\{n \geq 0 | P(X_n = s_i | X_0 = s_i) > 0\}$$

Es decir, cualquier posible forma de regresar al estado s_i debe ocurrir en multiplos de $d(s_i)$ pasos.

Definición 8. Si $d(s_i) = 1$ decimos que la cadena es *aperiódica*

Definición 9. Sea una cadena de Markov $(X_0, X_1, ...)$ con estados $S = \{s_1, ..., s_k\}$ diremos que $\pi = (\pi_0, ..., \pi_k)$ es una distribución estacionaria para la cadena si cumple lo siguiente:

- $\pi_i \geq 0$ para todo i
- $\sum_{i=0}^{k} \pi_i = 1$
- $\pi * P = \pi$

Teorema 2. (Existencia y unicidad de distribución estacionaria) Para cualquier cadena de Markov irreducible y aperiódica, existe al menos una distribución estacionaria y ésta es única.

Teorema 3. (Convergencia de una cadena de Markov) Una cadena de Markov irreducible y aperiódica, con una distribución estacionaria π cumple que

$$\lim_{n\to\infty} \mu^n = \lim_{n\to\infty} \mu^0 * P^n = \pi$$

3.2. Teoría de Grafos

También se supondrán conceptos básicos de Teoría de Grafos como digrafica $(V, \mathscr{E})$, ciclo $(\mathscr{C})$, componente conexa.

Definición 10. Sea $(V, \mathscr{E})$ una gráfica decimos que es *fuertemente conexa* si para cada par $u, v \in V$ existe un camino de u a v y viceversa.

Definición 11. Decimos que una digráfica es aperiódica si el máximo común divisor de la longitud de los ciclos es 1.

Teorema 4.Sea $(V, \mathscr{E})$ una digráfica fuertemente conexa, entonces se cumple que $\forall x, y \in V$, $per(x) = per(y)$.
Es decir, el periodo de dicha es igual para todos.

Afirmación 1. Si una gráfica fuertemente conexa (G) define una cadena de Markov, entonces si G es aperiódica, dicha cadena de Markov también.

Algoritmo de Tarjan. Algoritmo para encontrar el número de componentes conexas de una gráfica.

Algoritmo para periodicidad. Algoritmo para hallarla periocidad de una gráfica fuertemente conexa.

3.3. Internet

Internet como lo conocemos hoy en día es un ejemplo de una inversión sostenida y comprometida con la investigación para el desarrollo de la infraestructura de la información, cuando se comenzaba a construir el modelo que lo regiría en la temprana década de los 70's, como fruto de un proyecto gubernamental estadounidense llamado ARPANET (Advanced Research Proyects Agency Network) nadie sabía qué impacto tendría en la humanidad, ya que internet comenzó como una sencilla, pero bien construida filosofía de conexión entre dispositivos.
Con el paso del tiempo dicho proyecto fue liberado y popularizado como una forma de emitir y recibir información, ahora la información estaba en una red distribuida de computadoras y todos podían acceder a ella.

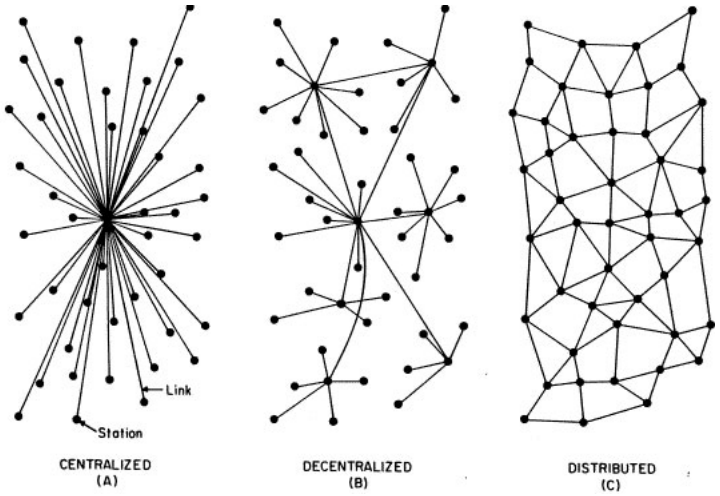


Figura 1: Esquema de diferentes tipos de redes: centralizada, desentralizada y distribuida

3.4. PageRank

Ya que internet es una filosofía de conexión que forma una red, en ésta hay dispositivos que solicitan información de dicha red mientras que hay otros dipositivos que se encargan de proveerla.
En la década de los 90's al empezar el alza en la popularidad del invento se notó la necesidad de una forma eficiente para acceder a los datos dentro de la red, nacieron los motores de búsqueda, donde la tarea de dichos programas era traer información de la red y mostrarla al usuario. Por la forma en que está construida la red, no era viable revisar el contenido de cada una de las páginas hasta encontrar una adecuada y mostrarla, había que poder saber si una página era útil o no sin revisar el contenido de ésta, todavía a finales de los 90's había muy poca investigación sobre el tema.
Hubo varios intentos para la construcción de motores de búsqueda como *AltaVista*, *Yahoo!* entre otros, cuando uno realizaba una sola búsqueda en ellos, desde los resultados arrojados todavía era difícil llegar una página que tuviera la información que se buscaba.

Figura 2: Doodle de Google en su 30° aniversario

No fue sino hasta 1998 que Sergery Brin y Lawrence Page, dos egresados de la Universidad de Stanford, publicaron en su artículo *The Anatomy of a Large-Scale Hypertextual Web Search Engine* la presentación de Google como prototipo de un motor de búsqueda a gran escala el cual hacía uso de un algorítmo que se volvería célebre; *PageRank: Bringing Order to the Web.*

" *PageRank or PR(A) can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web. Also, a PageRank for 26 million web pages can be computed in a few hours on a medium size workstation. There are many other details which are beyond the scope of this paper.* "
- *The Anatomy of a Large-Scale Hypertextual Web Search Engine, 1998*

Este fue el algoritmo que volvió a Google uno de los mejores motores de búsqueda de su época, hoy en día permanece como propiedad intelectual de la Universidad de Stanford y para el 81.5 % de la población de internet sigue siento su motor, por no decir único, principal.
Detrás del algoritmo existen detalles que no siempre pueden ser verificados por el tiempo que tardaría un algoritmo en verificarlo, pero en general la salida del algoritmo es la distribución de probabilidad que representa la probablidad de que una persona, a través de hacer click's aleatoriamente llegue a una página en particular, esta salida nos indica qué página es la más relevante.

4. Análisis del problema

4.1. Caso particular

Para construir el modelo, parece razonable construir uno que funcione a pequeña escala y después ajustarlo a lo que necesitamos.

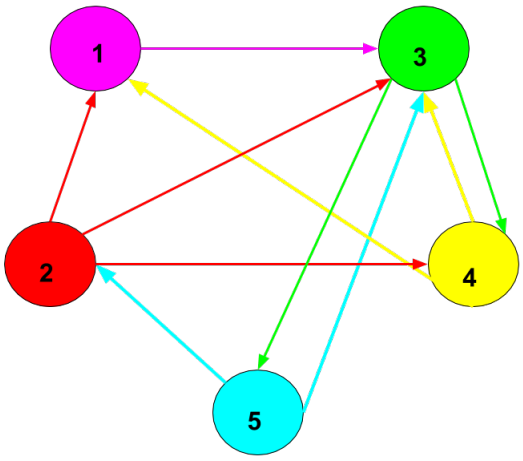
Para el caso de la estudiante, supongamos que se conoce la siguiente información:

- La página 1 cita a la página 3.
- La página 2 cita a la página 1, 3 y 4.
- La página 3 cita a la página 5 y 4.
- La página 4 cita a la página 1 y a la página 3.
- La página 5 cita a la página 2 y 3.

Supongamos que estamos parados en una página cualquiera que enlaza a otras páginas, esa página es importante por contenido que posee, pero sabemos que al enlazar a otras páginas significa que alguna parte del contenido que se encuentra en esta página fue obtenido de la página citada, lo que nos dice que el orígen de la información proviene de la página citada.

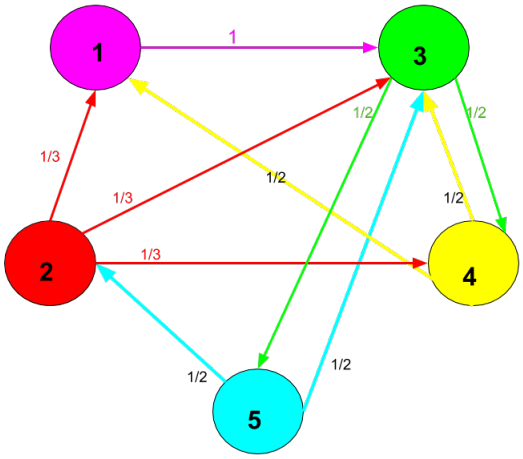
Siguiendo esta linea de pensamiento, quisiéramos saber en qué página se encuentra la mayor densidad de información valiosa a partir de las páginas que la citan (que apuntan a ella).

Una forma de visualizar la información anterior sería la siguiente



Supongamos que al inicio no sabemos qué página es la más importante, por tanto decimos que todas las páginas tienen la misma *importancia*, digamos 1.

Conforme dicha página cita a otras su *importancia* se divide entre esas páginas, por ahora supongamos que la divide de forma equitativa, lo cual se vería de la siguiente forma:



Lo que significa que la página 1 cede toda su importancia a la página 3, la página 2 dá 1/3 de su importancia a las tres páginas a las que cita y así sucesivamente.

Para nuestra situación particular, es válido suponer que cuando una persona comienza a buscar en la red empieza a buscar en una página aleatoria porque no tiene información sobre qué página es importante. Se plantea lo siguiente: si una persona comienza en una página web, elegida de forma aleatoria, y de ahí toma algún link que esté en dicha página y de esa página toma otro link y otro link y otro link... por un periodo muy largo de tiempo. Sabemos que al llevarse acabo esa actividad, el usuario **sólo** se movería entre las páginas a las que enlaza, se movería de un estado a otro, ahora supongamos que pasa x cantidad de tiempo en cada una de las páginas antes de hacer click, al cabo de realizar la actividad por un periodo infinito de tiempo, en promedio ¿en qué página pasó más tiempo el usuario?

La respuesta a la pregunta anterior es importante porque si existe una página en la que en promedio pasó más tiempo significa que es la página en la que se concentra más *importancia* ya que los links hablan sobre la importancia de dicha página y son éstos los que rigen cómo movernos.

Observamos que si nos encontramos parados sobre un vértice de la digráfica, es ese único vértice el de *define* hacia dónde movernos. Dicho de otra forma, el siguiente estado está sólo definido por el estado en el que nos encontramos.
Por tanto a partir de la digráfica anterior es posible construir una cadena de Markov de la siguiente manera:

- Cada página está representada por un vértice de la gráfica y éstos a su vez juegan el papel de estados en la cadena de Markov, pues son los valores a los cuales podemos movernos.
  $S = \{s_1 = 1, ..., s_5 = 5\}$
- Una vez que se está en algún estado de la cadena de Markov, éste tiene asociado una variable aleatoria que nos dice las probabilidades para movernos dentro de la gráfica.
  Definiremos la f.d.p. de la variable aleatoria de la siguiente manera:

  Sea $L(s_i)$ el número de páginas a las que enlaza la página/estado s_i entonces
  $P(X_{n+1} = s_{i_{n+1}} | X_n = s_i) = 1/L(s_i)$

Para la digráfica anterior se puede construir una matriz estocástica de la siguiente manera:

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

Notamos que P es la matriz de transición para nuestra cadena de Markov.

Y nuestra cadena de Markov tiene una distribución inicial de $\mu^o = (1/5, 1/5, 1/5, 1/5, 1/5)$ ya que al inicio todas las páginas tienen la misma probabilidad de ser elegidas.
Vemos también que cada vez que nos encontramos en un estado, la *regla* que nos indica cómo movernos no ha cambiado, es decir nuestra cadena es homogénea.

Nuestra cadena de Makov con su matriz de transición cumple las hipótesis del ***Teorema 1***, por tanto podemos afirmar que su conclusión es cierta para nuestro situación hipotética, es decir la distribución para μ^n de nuestra cadena está definido por

$$\mu^n = \mu_0 * P^n$$

Es decir, si nosotros nos mantenemos moviéndonos sobre las páginas, por tiempo indefinido, el vector μ^n nos dice la probabilidad de terminar en el estado s_i y está representada por μ_i^n

Como a nosotros nos interesaría saber en cuál página es en la que en general se pasaría más tiempo si nos mantenemos moviéndonos un intervalo de tiempo infinito, la pregunta se transforma en si y hacia dónde converge el siguiente límite

$$\lim_{n\to\infty} \mu^0 * P^n$$

Habría entonces que ver si el límite anterior existe para nuestra cadena de Markov, por el ***Teorema 3*** habría que verificar si nuestra cadena es irreducible y aperiódica.

Es fácil ver que verificar que nuestra cadena sea irreducible es equivalente a verificar si nuestra gráfica es *fuertemente conexa*. Para nuestro ejemplo, tras una revisión exhaustiva podemos verificar que lo cumple.

Para verificar si nuestra cadena es aperiódica, basta verificar si nuestra gráfica es aperiódica. Utilizando el *Algoritmo para periocidad* vemos que nuestra gráfica es aperiódica.

Una vez que se ha verificado que nuestra cadena cumple los requisitos, podemos concluir que nuestra cadena de Markov converge.

Y a través de elevar a P a su $n-esima$ potencia para un n *suficientemente grande* (se encontró que n = 100 es *suficientemente grande* para nuestros propósitos [1]) y mutiplicar por la distribución inicial podemos obtener una aproximación de nuestro vector π.

El resultado de la multiplicación anterior nos sugeriría qué página es la más relevante, es aquella etiquetada con el índice que posee el valor más alto del vector.

Para nuestro caso particular se realizó un programa que nos ayudara a obtener la solución que queremos, ya que se intentó realizar a través de programas como Sybolab, pero no permitía una exponenciación tan grande, calculando una aproximación a su distribución estacionaria tenemos:

$\mu^0 * P^{100} =$

$$\mu^0 * \begin{vmatrix} 0 & 0 & 1 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 \end{vmatrix} = [0.13846154, 0.09230769, 0.36923077, 0.21538462, 0.18461538] \approx \pi$$

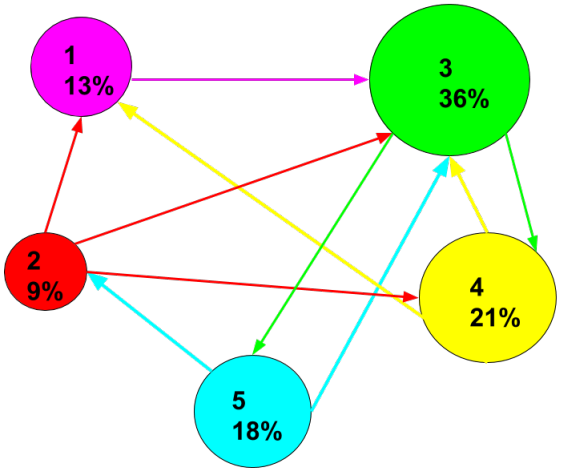
[1] Google Patents; *Method for node ranking in a linked database*

Observamos que el valor máximo de nuestra aproximación se encuentra en el índice 3 de *pi*.

$$\pi \approx [0.13846154,\ 0.09230769,\ \mathbf{0.36923077},\ 0.21538462,\ 0.18461538]$$

Es decir en el estado 3 de nuestra distribución se encuentra se encuentra la página más relevante, que a su vez es la página 3.
Vemos también que la suma de todas las entradas de π efectivamente da 1. Es decir **sí** es una medida de probabilidad.

Al ver a π ya podemos darle un orden de importancia a las páginas y asociarle la probabilida de al caminar aleatoriamente por la gráfica terminar en algúno de sus estados:



Finalmente, para nuestro caso particular, la página más *relevante* y por consultar es la 3.

4.2. Solución Algorítimica

4.2.1. Algoritmo

En un intento de automatizar la solución a algunos casos particulares de este problema propone el siguiente algoritmo

Result: Página más relevante de una base de datos
Construcción de matriz a partir de la base de datos
G ← base de datos
x ← $i \in$ range (0, G-1) $x_i = 1/|V(G)|$
if *! (aTarjan(G) == 1)* **then**
  Termina
else
  if *! aAperiodica(G)* **then**
    Termina
  else
    P ← expBin(G, 100)
  end
end
phi ← $matMul(\text{x}, P)$
maxValue ← $max(phi)$
mostImportantPage ← $index(maxValue)$
return G(mostImportantPage)

4.2.2. Glosa

Para el algoritmo anterior, lo primero que se hace es construir a la matriz estocástica (que a su vez representa a una gráfica G) de la base de datos a partir de las refencias de cada página.

Como inicialemente cada página tiene la misma probabilidad de ser elegida, se construye el vector de la distribución inicial con una probabilidad uniforme. Luego usando el algoritmo de Tarjan, se verifica que la gráfica sea fuertemente conexa, si no lo es terminamos y lanzamos un error, si lo es se verifica a través del algoritmo de periodicidad si la gráfica es aperiódica, si no lo es terminamos, si lo es a través del algoritmo de exponenciación binaria se calcula P^n.
Finalmente se realiza la multiplicación de la distribución inicial con P^n y se encuentra el valor máximo del vector resultante.
Se devuelve la página de la gráfica G que fue representada con dicho índice.

5. Análisis de Resultados

Vemos que el algoritmo anterior funciona para los casos en que se puede constuir una cadena de Markov irreducible y aperiódica, se decidió utilizar el algoritmo de Tarjan y de periodicidad de una gráfica pues existen demostraciones que nos aseguran que los algoritmos *funcionan* es decir, de vuelven lo que prometen. Se evitará hacer un análisis más detallado de la *complejidad* del algoritmo pues está fuera de los objetivos de este trabajo.

Es importante mencionar que se está trabajando bajo ciertas suposiciones como que las páginas distribuyen su importancia de forma equitativa a las páginas a las que cita, que al inicio todas las páginas tienen la misma probabilidad de ser elegidas, que dentro de la base de datos se encuentra la página más relevante, entre otros.

Este trabajo presenta un algoritmo **muy** simplificado del algoritmo original, en la realidad se dan varias situaciones; las páginas no dividen de forma equitativa su importancia (se tienen métodos para saber cómo hacerlo), si la gráfica original de la red no es fuertemente conexa existen métodos para volverla fuertemente conexa y/o tratar con cada componente de la gráfica, además la distribución inicial no es uniforme es decir, no todas las páginas tienen la misma probabilidad de ser elegidas,

a través de la información colectada de cada usuario de Google se decide qué página es más relevante para **esa** persona (usando sus intereses como criterio de ponderación).
Mientras que nos detuvimos de devolver una respuesta si la cadena de Markov no converge, en la realidad es que estas verificaciones no se hacen para *PageRank*, en el documento original donde se presenta a *Google* como un motor de búsqueda en la web, dice textualmente *suponemos que la cadena converge.*

Este modelo se construyó bajo ciertas suposiciones para volverlo manejable.

6. Conclusiones

Fue interesante y agradable conocer sobre Cadenas de Markov y sus aplicaciones dentro del campo de la computación, la tarea de recomendar páginas de buena calidad no es sencillo.
Vemos que en la realidad no siempre nos podemos apegar a la formalidad y rigor de las matemáticas porque podría detenernos de convertir un pedacito de teoría en algo útil para muchas personas.

Por otro lado es indescriptible el asombro de ver cómo con el enfoque correcto ideas tan profundas como los procesos estocásticos pueden ser de gran utilidad para nuestros problemas y que ésta nazca a partir de escribir un algoritmo *sencillo* que es capaz de ocultar la complejidad de las ideas detrás de él, este hecho me dejó fascinada.

Quizá hay cosas que fueron tratadas con poco detalle, se trató de ahondar sólo lo suficiente para el desarrollo de este trabajo.

7. Meditación

Algo que me parece importante resaltar es que hoy en día las prácticas de *Google* son cuestionadas por muchos. El algorítmo es buenísimo y entre otras cosas realiza un análisis sintáctico de las palabras, datos como ubicación, edad, idioma, datos de usuario y bastantes otras ¡para una sola búsqueda!.
La trampa de todo esto es que podemos caer en una burbuja falsa de realidad con la información sesgada que ofrece el buscador, pues el algoritmo sólo nos muestra *lo que nos interesa* decidiendo esto a partir de los datos que posee de cada uno.

Encima el algoritmo tiende de ponderar con calificaciones más altas a páginas con *autoridad* más no verifica *veracidad*, provocando en algunos casos desinformación sobre las pesonas que utilizan el servicio.
Tomm Scott dijo en algún momento que *Google* es el motor de radicalización más poderoso que existe, haciendo referencia a lo fácil que es caer en ideas cada vez más radicales cada vez con las búsquedas de dicho buscador.
Me parece que sería bueno mantener un grado saludable de escepticismo cada que se haga uso de dicho servicio y tratar de recordar estos hechos.

8. Bibliografía

- Brzezniak, Z. y Zaztawniak, T. (2002) *Londres: Springer Undergraduate Mathematics Serieis, Basic Stochastic Processes - A Course Through Exercises*, Springer-Verlag Londres Ltd.
- Haggstrom, O. (2002) *Finite Markov chains and algorithmic applications*, London, UK: London Mathematical Society Students Text.
- Wikipedia,(2018) *Markov Chain*. Recuperado de: https://en.wikipedia.org/wiki/Markov$_c hain$

- Kazdan, Jerry. (2012), University of Pennsylvania. *Markov chain, Google's PageRank Algorithm.* Recuperado de: https://www.math.upenn.edu/ kazdan/312F12/JJ/MarkovChains/markov$_g oogle.pdf$
- Wikipedia.(2019) *PageRank*. Recuperado de: https://en.wikipedia.org/wiki/PageRank
- Google's Patent.(2012) *Method for node ranking at linked Database*. Recuperado de: https://patents.google.com/patent/US6285999
- Roberts, E.(2016) *The Google PageRank Algorithm*. Recuperado de: https://web.stanford.edu/class/cs54n/handouts/24-GooglePageRankAlgorithm.pdf/
- Wikipedia, (2019) *Aperiodic Graph*. Recuperado de: https://en.wikipedia.org/wiki/Aperiodic$_g raph$
- Chardtrand G., Zang P.,(2005), Western Michigan University. *A First Course in Graph Theory*, New York, Dover Publications.
- Geeks for Geeks, (2015). *Tarjan's Algorithm to find Strongly Connected Components*. Recuperado de: https://www.geeksforgeeks.org/tarjan-algorithm-find-strongly-connected-components/
- Musib, A.(2017)*PageRank Algorithm and Implementation*. Recuperado de: https://www.geeksforgeeks.org/page-rank-algorithm-implementation/
- Liedke, L. (2019) *+100 internet statistics and facts*. Recuperado de: https://www.websitehostingrating.com/internet-statistics-facts/
- NetworkXDevelopers. (2015) *Creating a graph*. Recuperado de: https://networkx.github.io/documentation/networkx-1.10/tutorial/tutorial.html
- NetworkXDevelopers.(2015)*Digraph - Directed Graph with self loops*. Recuperado de: https://networkx.github.io/documentation/networkx-1.10/reference/classes.digraph.html