

Métodos Numéricos

TP2

3 de octubre de 2015

Years Later for Guillermo Vilas, He's Still Not the One



Integrante	LU	Correo electrónico
Martin Baigorria	575/14	martinbaigorria@gmail.com
Federico Beuter	827/13	federicobeuter@gmail.com
Mauro Cherubini	835/13	cheru.mf@gmail.com
Rodrigo Kapobel	695/12	rok_35@live.com.ar

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Resumen: TODO
Keywords: TODO

Índice

1. Introduccion	3
2. Desarrollo	5
3. Experimentación	6
4. Conclusiones	7
5. Apéndice A: Enunciado	8
6. Apéndice B: Código	13

1. Introduccion

El 25 de Mayo de 2015 el diario The New York Times publico un articulo titulado "Years Later for Guillermo Vilas, He's Still Not the One", donde se repasa el rendimiento del tenista argentino durante los años 1975/1976 y se discute el calculo del ranking de la ATP en ese momento. Aunque hoy en día Vilas es un icono del tenis argentino, nunca logro estar en la cima del ranking de la ATP.



Figura 1: Guillermo Vilas after winning a tournament in Stockholm in 1975. A journalist has asserted that Vilas deserved to be ranked No. 1 during that year.

En 2016, un grupo de investigadores argentinos decidió analizar el ranking de la ATP en 1975 y 1976 para determinar si Vilas debió haber sido numero 1. Dado que los rankings no se actualizaban constantemente en ese momento, los investigadores mostraron que de haberse actualizado de forma periódica, Vilas hubiese sido numero 1 por durante 7 semanas en 1975 y 1976.

Existen precedentes donde se actualizo un ranking de tenis de forma retroactiva. Este es el caso de la WTA, que determino que Evonne Goolagong Cawley debió haber sido numero 1 por dos semanas en 1976. Por esta razon el grupo de investigación argentino considera que revisar estos rankings no es un esfuerzo en vano. Cuando buscábamos los datos de la ATP en el 1975/1976, uno de los investigadores de este equipo que contactamos nos comento: "Es interesante tu decisión de indagar sobre el tema. Tal vez no estás al tanto del trabajo y lucha que estamos realizando contra la ATP, por el ranking de los 70 en el que perjudicaron a Vilas y muchísimos otros jugadores."

En ese momento, el calculo del ranking de la ATP era bastante rudimentario: "It was a system based on an average of a player's results, and it often rewarded top players who played fewer tournaments. Vilas was a workhorse, which is how he managed not to reach No. 1 in the ATP rankings in 1977, when he won the French Open, the United States Open and 14 other tournaments." [5].

Los métodos para calcular rankings no solo son relevantes para definir las posiciones de equipos y jugadores en eventos deportivos, sino que aparecen constantemente en todo tipo de situaciones donde se debe imponer algun tipo de orden. Este es el caso por ejemplo de los concursos docentes, donde se ponderan los diferentes antecedentes para decidir cual es el candidato *idoneo* para el puesto.

Otro caso sumamente relevante en cuanto algoritmos de ranqueo es el de los motores de búsqueda. Los motores de búsqueda deben encontrar alguna forma de ordenar de forma relevante los sitios web que están relacionados con una consulta. El caso iconico es el de Google con su algoritmo PageRank. Los buscadores antes de 1990 eran sumamente rudimentarios, utilizaban algoritmos de ranqueo vulnerables en el sentido que podían ser manipulados y no se explotaba gran parte de la estructura de la web. Esta fue una de las razones por las cuales una consulta no siempre devolvía resultados relevantes. Este fue el caso por ejemplo de algunos buscadores en ese momento como Yahoo! Search o AltaVista.

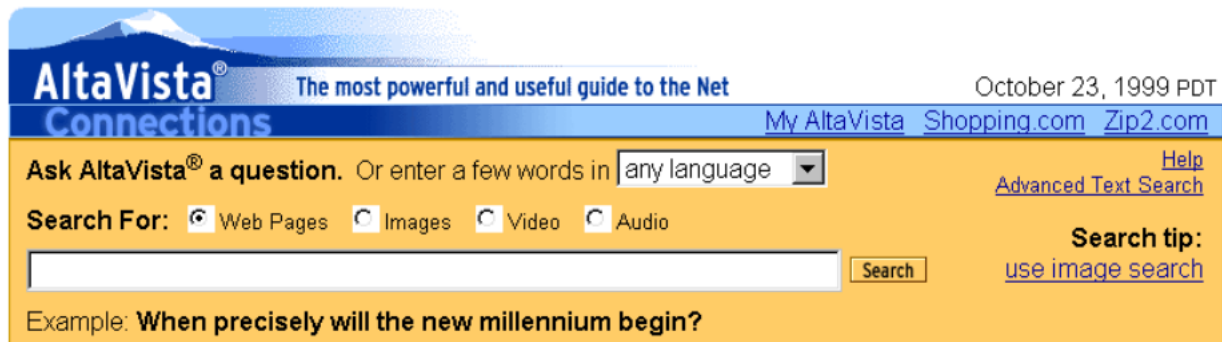


Figura 2: Sitio Web de Altavista, ano 1999.

El clásico paper de Brin y Page, “The anatomy of large-scale hypertextual Web search engine.” [3] explica brevemente el origen del motor de búsqueda de Google y del algoritmo PageRank. La idea es básicamente la siguiente, en primer lugar se implementa un crawler distribuido para poder solicitar y armar el grafo de la web. Las palabras de cada sitio son indexadas y guardadas en una base de datos. Al llegar una consulta al buscador, un programa busca la consulta en los índices de paginas. De esta forma llegamos a un conjunto de paginas que están relacionadas con la consulta. Luego, antes de devolverle al usuario los resultados, estas paginas son ordenadas utilizando el famoso algoritmo PageRank. Este algoritmo se basa en la idea de que para medir la relevancia de un sitio se puede usar como proxy la cantidad de sitios que tienen un link al mismo. Para evitar que un usuario malintencionado manipule los resultados del mismo, la relevancia otorgada por un sitio web que linkea a otro es proporcional a su propia relevancia e inversamente proporcional a la cantidad de links (o grado de salida) del mismo.

El presente trabajo practico tendrá como objetivo implementar el algoritmo PageRank para luego utilizarlo para generar rankings de todo tipo, ya sea para ordenar la relevancia de paginas webs o generar rankings deportivos. PageRank es un algoritmo que basa su ranking en encontrar el autovector de una matriz de transiciones. A priori esto puede sonar complicado, pero luego mostraremos que en realidad es bastante simple y elegante. Dado que ordenar la relevancia de millones de sitios web no es un problema trivial, en la practica este problema se resuelve utilizando álgebra lineal y métodos numéricos. Una muy buena introducción teórica se puede encontrar en el trabajo de Bryan y Leise [4]. Otros autores como Kamvar, Haveliwala, Manning, Christopher y Golub [7] han buscado otros enfoques y métodos para poder acelerar este algoritmo. La idea es encontrar una forma eficiente de poder computar este modelo, calibrando sus diferentes parámetros de modelado y convergencia para lograr un orden relevante.

Una vez planteado el procedimiento, experimentaremos con la complejidad temporal de los métodos implementados y evaluaremos los diferentes parámetros a calibrar. Finalmente concluiremos si según el algoritmo PageRank y nuestra matriz de transición Vilas efectivamente debió haber estado en la punta del ATP en 1975/1976. En caso afirmativo, sin dudas nos comunicaremos con la ATP.



Figura 3: Guillermo Vilas apoya este TP.

2. Desarrollo

3. Experimentación

4. Conclusiones

5. Apéndice A: Enunciado

Métodos Numéricos
Segundo Cuatrimestre 2015
Trabajo Práctico 2



Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ohhh solo tiran π -edras...

Contexto y motivación

A partir de la evolución de Internet durante la década de 1990, el desarrollo de motores de búsqueda se ha convertido en uno de los aspectos centrales para su efectiva utilización. Hoy en día, sitios como Yahoo, Google y Bing ofrecen distintas alternativas para realizar búsquedas complejas dentro de un red que contiene miles de millones de páginas web.

En sus comienzos, una de las características que distinguió a Google respecto de los motores de búsqueda de la época fue la calidad de los resultados obtenidos, mostrando al usuario páginas relevantes a la búsqueda realizada. El esquema general de los orígenes de este motor de búsqueda es brevemente explicado en Brin y Page [3], donde se mencionan aspectos técnicos que van desde la etapa de obtención de información de las páginas disponibles en la red, su almacenamiento e indexado y su posterior procesamiento, buscando ordenar cada página de acuerdo a su importancia relativa dentro de la red. El algoritmo utilizado para esta última etapa es denominado PageRank y es uno (no el único) de los criterios utilizados para ponderar la importancia de los resultados de una búsqueda. En este trabajo nos concentraremos en el estudio y desarrollo del algoritmo PageRank.

Por otro lado, las competencias deportivas, en todas sus variantes y disciplinas, requieren casi inevitablemente la comparación entre competidores mediante la confección de *Tablas de Posiciones* y *Rankings* en base a resultados obtenidos en un período de tiempo determinado. Estos ordenamientos de equipos están generalmente (aunque no siempre) basados en reglas relativamente claras y simples, como proporción de victorias sobre partidos jugados o el clásico sistema de puntajes por partidos ganados, empatados y perdidos. Sin embargo, estos métodos simples y conocidos por todos muchas veces no logran capturar la complejidad de la competencia y la comparación. Esto es particularmente evidente en ligas donde, por ejemplo, todos los equipos no juegan la misma cantidad de veces entre sí.

A modo de ejemplo, la NBA y NFL representan dos ligas con fixtures de temporadas regulares con estas características. Recientemente, el Torneo de Primera División de AFA se suma a este tipo de competencias, ya que la incorporación de la *Fecha de Clásicos* parece ser una interesante idea comercial, pero no tanto desde el punto de vista deportivo ya que cada equipo juega contra su *clásico* más veces que el resto. Como contraparte, éstos rankings son utilizados muchas veces como criterio de decisión, como por ejemplo para determinar la participación en alguna competencia de nivel internacional, con lo cual la confección de los mismos constituye un elemento sensible, afectando intereses deportivos y económicos de gran relevancia.

El problema, Parte I: PageRank y páginas web

El algoritmo PageRank se basa en la construcción del siguiente modelo. Supongamos que tenemos una red con n páginas web $Web = \{1, \dots, n\}$ donde el objetivo es asignar a cada una de ellas un puntaje que determine la importancia relativa de la misma respecto de las demás. Para modelar las relaciones entre ellas, definimos la *matriz de conectividad* $W \in \{0, 1\}^{n \times n}$ de forma tal que $w_{ij} = 1$ si la página j tiene un link a la página i , y $w_{ij} = 0$ en caso contrario. Además, ignoramos los *autolinks*, es decir, links de una página a sí misma, definiendo $w_{ii} = 0$. Tomando esta matriz, definimos el grado de la página j , n_j , como la cantidad de links salientes hacia otras páginas de la red, donde $n_j = \sum_{i=1}^n w_{ij}$. Además, notamos con x_j al puntaje asignado a la página $j \in Web$, que es lo que buscamos calcular.

La importancia de una página puede ser modelada de diferentes formas. Un link de la página $u \in Web$ a la página $v \in Web$ puede ser visto como que v es una página importante. Sin embargo, no queremos que una página obtenga mayor importancia simplemente porque es apuntada desde muchas páginas. Una forma de limitar esto es ponderar los links utilizando la importancia de la página de origen. En otras palabras, pocos links de páginas importantes pueden valer más que muchos links de páginas poco importantes. En particular, consideramos que la importancia de la página v obtenida mediante el link de la página u es proporcional a la importancia de la página u e inversamente proporcional al grado de u . Si la página u contiene n_u links, uno de los cuales apunta a la página v , entonces el aporte de ese link a la página v será x_u/n_u . Luego, sea $L_k \subseteq Web$ el conjunto de páginas que tienen un link a la página k . Para cada página pedimos que

$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j}, \quad k = 1, \dots, n. \quad (1)$$

Definimos $P \in \mathbb{R}^{n \times n}$ tal que $p_{ij} = 1/n_j$ si $w_{ij} = 1$, y $p_{ij} = 0$ en caso contrario. Luego, el modelo planteado en (1) es equivalente a encontrar un $x \in \mathbb{R}^n$ tal que $Px = x$, es decir, encontrar (suponiendo que existe) un autovector asociado al autovalor 1 de una matriz cuadrada, tal que $x_i \geq 0$ y $\sum_{i=1}^n x_i = 1$. En Bryan y Leise [4] y Kamvar et al. [7, Sección 1] se analizan ciertas condiciones que debe cumplir la red de páginas para garantizar la existencia de este autovector.

Una interpretación equivalente para el problema es considerar al *navegante aleatorio*. Éste empieza en una página cualquiera del conjunto, y luego en cada página j que visita sigue navegando a través de sus links, eligiendo el mismo con probabilidad $1/n_j$. Una situación particular se da cuando la página no tiene links salientes. En ese caso, consideramos que el navegante aleatorio pasa a cualquiera de las página de la red con probabilidad $1/n$. Para representar esta situación, definimos $v \in \mathbb{R}^{n \times n}$, con $v_i = 1/n$ y $d \in \{0, 1\}^n$ donde $d_i = 1$ si $n_i = 0$, y $d_i = 0$ en caso contrario. La nueva matriz de transición es

$$\begin{aligned} D &= vd^t \\ P_1 &= P + D. \end{aligned}$$

Además, consideraremos el caso de que el navegante aleatorio, dado que se encuentra en la página j , decida visitar una página cualquiera del conjunto, independientemente de si esta se encuentra o no referenciada por j (fenómeno conocido como *teletransportación*). Para ello, consideramos que esta decisión se toma con una probabilidad $c \geq 0$, y podemos incluirlo al modelo de la siguiente forma:

$$\begin{aligned} E &= v\bar{1}^t \\ P_2 &= cP_1 + (1 - c)E, \end{aligned}$$

donde $\bar{1} \in \mathbb{R}^n$ es un vector tal que todas sus componentes valen 1. La matriz resultante P_2 corresponde a un enriquecimiento del modelo formulado en (1). Probabilísticamente, la componente x_j del vector solución (normalizado) del sistema $P_2x = x$ representa la proporción del tiempo que, en el largo plazo, el navegante aleatorio pasa en la página $j \in \text{Web}$. Denotaremos con π al vector solución de la ecuación $P_2x = x$, que es comúnmente denominado *estado estacionario*.

En particular, P_2 corresponde a una matriz *estocástica por columnas* que cumple las hipótesis planteadas en Bryan y Leise [4] y Kamvar et al. [7], tal que P_2 tiene un autovector asociado al autovalor 1, los demás autovalores de la matriz cumplen $1 = \lambda_1 > |\lambda_2| \geq \dots \geq |\lambda_n|$ y, además, la dimensión del autoespacio asociado al autovalor λ_1 es 1. Luego, π puede ser calculada de forma estándar utilizando el método de la potencia.

Una vez calculado el ranking, se retorna al usuario las t páginas con mayor puntaje.

El problema, Parte II: PageRank y ligas deportivas

Existen en la literatura distintos enfoques para abordar el problema de determinar el *ranking* de equipos de una competencia en base a los resultados de un conjunto de partidos. En Govan et al. [6] se hace una breve reseña de dos ellos, y los autores proponen un nuevo método basado en el algoritmo PageRank que denominan GeM¹. Conceptualmente, el método GeM representa la temporada como un red (grafo) donde las páginas web representan a los equipos, y existe un link (que tiene un valor, llamado peso, asociado) entre dos equipos que los relaciona modelando los resultados de los posibles enfrentamientos entre ellos. En base a este modelo, Govan et al. [6] proponen calcular el ranking de la misma forma que en el caso de las páginas web.

En su versión básica, que es la que consideraremos en el presente trabajo, el método GeM (ver, e.g., [6, Sección GeM Ranking Method]) es el siguiente²:

1. La temporada se representa mediante un grafo donde cada equipo representa un nodo y existe un link de i a j si el equipo i perdió al menos una vez con el equipo j .
2. Se define la matriz $A^t \in \mathbb{R}^{n \times n}$

$$A_{ji}^t = \begin{cases} w_{ji} & \text{si el equipo } i \text{ perdió con el equipo } j, \\ 0 & \text{en caso contrario,} \end{cases}$$

donde w_{ji} es la diferencia absoluta en el marcador. En caso de que i pierda más de una vez con j , w_{ji} representa la suma acumulada de diferencias. Notar que A^t es una generalización de la matriz de conectividad W definida en la sección anterior.

3. Definir la matriz $H_{ji}^t \in \mathbb{R}^{n \times n}$ como

$$H_{ji}^t = \begin{cases} A_{ji}^t / \sum_{k=1}^n A_{ki}^t & \text{si hay un link } i \text{ a } j, \\ 0 & \text{en caso contrario.} \end{cases}$$

¹Aunque no se especifica, asumimos que el nombre se debe a las iniciales de los autores.

²Notar que en artículo, Govan et al. [6] lo definen sobre la traspuesta. La definición y las cuentas son equivalentes, simplemente se modifica para mantener la consistencia a lo largo del enunciado.

4. Tomar $P = H^t$, y aplicar el método PageRank como fue definido previamente, siendo π la solución a la ecuación $P_2x = x$. Notar que los páginas sin links salientes, en este contexto se corresponden con aquellos equipos que se encuentran invictos.
5. Utilizar los puntajes obtenidos en π para ordenar los equipos.

En función del contexto planteado previamente, el método GeM define una estructura que relaciona equipos dependiendo de los resultados parciales y obtener un ranking utilizando solamente esta información.

Enunciado

El objetivo del trabajo es experimentar en el contexto planteado utilizando el algoritmo PageRank con las variantes propuestas. A su vez, se busca comparar los resultados obtenidos cualitativa y cuantitativamente con los algoritmos tradicionales utilizados en cada uno de los contextos planteados. Los métodos a implementar (como mínimo) en ambos contextos planteados por el trabajo son los siguientes:

1. *Búsqueda de páginas web*: PageRank e IN-DEG, éste último consiste en definir el ranking de las páginas utilizando solamente la cantidad de ejes entrantes a cada una de ellas, ordenándolos en forma decreciente.
2. *Rankings en competencias deportivas*: GeM y al menos un método estándar propuesto por el grupo (ordenar por victorias/derrotas, puntaje por ganado/empatado/perdido, etc.) en función del deporte(s) considerado(s).

El contexto considerado en 1., en la búsqueda de páginas web, representa un desafío no sólo desde el modelado, si no también desde el punto de vista computacional considerando la dimensión de la información y los datos a procesar. Luego, dentro de nuestras posibilidades, consideramos un entorno que simule el contexto real de aplicación donde se abordan instancias de gran escala (es decir, n , el número total de páginas, es grande). Para el desarrollo de PageRank, se pide entonces considerar el trabajo de Bryan y Leise [4] donde se explica la intuición y algunos detalles técnicos respecto a PageRank. Además, en Kamvar et al. [7] se propone una mejora del mismo. Si bien esta mejora queda fuera de los alcances del trabajo, en la Sección 1 se presenta una buena formulación del algoritmo. En base a su definición, P_2 no es una matriz esparsa. Sin embargo, en Kamvar et al. [7, Algoritmo 1] se propone una forma alternativa para computar $x^{(k+1)} = P_2x^{(k)}$. Este resultado debe ser utilizado para mejorar el almacenamiento de los datos.

En la práctica, el grafo que representa la red de páginas suele ser esparso, es decir, una página posee relativamente pocos links de salida comparada con el número total de páginas. A su vez, dado que n tiende a ser un número muy grande, es importante tener en cuenta este hecho a la hora de definir las estructuras de datos a utilizar. Luego, desde el punto de vista de implementación se pide utilizar alguna de las siguientes estructuras de datos para la representación de las matrices esparsas: *Dictionary of Keys* (dok), *Compressed Sparse Row* (CSR) o *Compressed Sparse Column* (CSC). Se deberá incluir una justificación respecto a la elección que considere el contexto de aplicación. Además, para PageRank se debe implementar el método de la potencia para calcular el autovector principal. Esta implementación debe ser realizada íntegramente en C++.

En función de la experimentación, se deberá realizar un estudio particular para cada algoritmo (tanto en términos de comportamiento del mismo, como una evaluación de los resultados obtenidos) y luego se procederá a comparar cualitativamente los rankings generados. La experimentación deberá incluir como mínimo los siguientes experimentos:

1. Estudiar la convergencia de PageRank, analizando la evolución de la norma Manhattan (norma L_1) entre dos iteraciones sucesivas. Comparar los resultados obtenidos para al menos dos instancias de tamaño mediano-grande, variando el valor de c .
2. Estudiar el tiempo de cómputo requerido por PageRank.
3. Para cada algoritmo, proponer ejemplos de tamaño pequeño que ilustren el comportamiento esperado (puede ser utilizando las herramientas provistas por la cátedra o bien generadas por el grupo).

Puntos opcionales:

1. Demostrar que los pasos del Algoritmo 1 propuesto en Kamvar et al. [7] son correctos y computan P_2x .
2. Establecer una relación con la proporción entre $\lambda_1 = 1$ y $|\lambda_2|$ para la convergencia de PageRank.

El segundo contexto de aplicación no presenta mayores desafíos desde la perspectiva computacional, ya que en el peor de los casos una liga no suele tener mas que unas pocas decenas de equipos. Más aún, es de esperar que en general la matriz que se obtiene no sea esparsa, ya que probablemente un equipo juegue contra un número significativo de contrincantes. Sin embargo, la popularidad y sensibilidad del problema planteado requieren de un estudio detallado y pormenorizado de la calidad de los resultados obtenidos. El objetivo en este segundo caso de estudio es puramente experimental.

En función de la implementación, aún cuando no represente la mejor opción, es posible reutilizar y adaptar el desarrollo realizado para páginas web. También es posible realizar una nueva implementación desde cero, simplificando la operatoria y las estructuras, en C++, MATLAB o PYTHON.

La experimentación debe ser realizada con cuidado, analizando (y, eventualmente, modificando) el modelo de GeM:

1. Considerar al menos un conjunto de datos reales, con los resultados de cada fecha para alguna liga de algún deporte.
2. Notar que el método GeM asume que no se producen empates entre los equipos (o que si se producen, son poco frecuentes). En caso de considerar un deporte donde el empate se da con cierta frecuencia no despreciable (por ejemplo, fútbol), es fundamental aclarar como se refleja esto en el modelo y analizar su eventual impacto.
3. Realizar experimentos variando el parámetro c , indicando como impacta en los resultados. Analizar la evolución del ranking de los equipos a través del tiempo, evaluando también la evolución de los rankings e identificar características/hechos particulares que puedan ser determinantes para el modelo, si es que existe alguno.
4. Comparar los resultados obtenidos con los reales de la liga utilizando el sistema estándar para la misma.

Puntos opcionales:

1. Proponer (al menos) dos formas alternativas de modelar el empate entre equipos en GeM.

Parámetros y formato de archivos

El programa deberá tomar por línea de comandos dos parámetros. El primero de ellos contendrá la información del experimento, incluyendo el método a ejecutar (**alg**, 0 para PageRank, 1 para el método alternativo), la probabilidad de teletransportación c , el tipo de instancia (0 páginas web, 1 deportes), el *path* al archivo/directorio conteniendo la definición de la red (que debe ser relativa al ejecutable, o el path absoluto al archivo) y el valor de tolerancia utilizado en el criterio de parada del método de la potencia.

El siguiente ejemplo muestra un caso donde se pide ejecutar PageRank, con una probabilidad de teletransportación de 0.85, sobre la red descrita en **test1.txt** (que se encuentra en el directorio **tests/**), correspondiente a una instancia de ranking aplicado a deportes y con una tolerancia de corte de 0,0001.

```
0 0.85 1 tests/red-1.txt 0.0001
```

Para la definición del grafo que representa la red, se consideran dos bases de datos de instancias con sus correspondientes formatos. La primera de ellas es el conjunto provisto en SNAP [2] (el tipo de instancia es 0), con redes de tamaño grande obtenidos a partir de datos reales. Además, se consideran las instancias que se forman a partir de resultados de partidos entre equipos, para algún deporte elegido por el grupo.

En el caso de la base de SNAP, los archivos contiene primero cuatro líneas con información sobre la instancia (entre ellas, n y la cantidad total de links, m) y luego m líneas con los pares i, j indicando que i apunta a j . A modo de ejemplo, a continuación se muestra el archivo de entrada correspondiente a la red propuesta en Bryan y Leise [4, Figura 1]:

```
# Directed graph (each unordered pair of nodes is saved once):
# Example shown in Bryan and Leise.
# Nodes: 4 Edges: 8
# FromNodeId    ToNodeId
1    2
1    3
1    4
2    3
2    4
3    1
4    1
4    3
```

Para el caso de rankings en ligas deportivas, el archivo contiene primero una línea con información sobre la cantidad de equipos (n), y la cantidad de partidos totales a considerar (k). Luego, siguen k líneas donde cada una de ellas representa un partido y contiene la siguiente información: número de fecha (es un dato opcional al problema, pero que puede ayudar a la hora de experimentar), equipo i , goles equipo i , equipo j , goles equipo j . A continuación se muestra el archivo de entrada con la información del ejemplo utilizado en Govan et al. [6]:

```
6 10
1 1 16 4 13
1 2 38 5 17
1 2 28 6 23
1 3 34 1 21
1 3 23 4 10
1 4 31 1 6
1 5 33 6 25
1 5 38 4 23
1 6 27 2 6
1 6 20 5 12
```

Es importante destacar que, en este último caso, los equipos son identificados mediante un número. Opcionalmente podrá considerarse un archivo que contenga, para cada equipo, cuál es el código con el que se lo identifica.

Una vez ejecutado el algoritmo, el programa deberá generar un archivo de salida que contenga una línea por cada página (n líneas en total), acompañada del puntaje obtenido por el algoritmo PageRank/IN-DEG/método alternativo.

Para generar instancias de páginas web, es posible utilizar el código Python provisto por la cátedra. La utilización del mismo se encuentra descripta en el archivo README. Es importante mencionar que, para que el mismo funcione, es necesario tener acceso a Internet. En caso de encontrar un bug en el mismo, por favor contactar a los docentes de la materia a través de la lista. Desde ya, el código puede ser modificado por los respectivos grupos agregando todas aquellas funcionalidades que consideren necesarias.

Para instancias correspondientes a resultados entre equipos, la cátedra provee un conjunto de archivos con los resultados del Torneo de Primera División del Fútbol Argentino hasta la Fecha 23. Es importante aclarar que los dos partidos suspendidos, River - Defensa y Justicia y Racing - Godoy Cruz han sido arbitrariamente completados con un resultado inventado, para simplificar la instancia. En función de datos reales, una alternativa es considerar el repositorio DataHub [1], que contiene información estadística y resultados para distintas ligas y deportes de todo el mundo.

Fechas de entrega

- *Formato Electrónico*: Martes 6 de Octubre de 2015, hasta las 23:59 hs, enviando el trabajo (informe + código) a la dirección `metnum.lab@gmail.com`. El subject del email debe comenzar con el texto [TP2] seguido de la lista de apellidos de los integrantes del grupo.
- *Formato físico*: Miércoles 7 de Octubre de 2015, a las 18 hs. en la clase práctica.

Importante: El horario es estricto. Los correos recibidos después de la hora indicada serán considerados re-entrega.

6. Apéndice B: Código

Referencias

- [1] Datahub. <http://datahub.io>.
- [2] Stanford large network dataset collection. <http://snap.stanford.edu/data/#web>.
- [3] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, April 1998.
- [4] Kurt Bryan and Tanya Leise. The linear algebra behind google. *SIAM Review*, 48(3):569–581, 2006.
- [5] Christopher Clarey. Years later for guillermo vilas, he’s still not the one. *The New York Times*.
- [6] Angela Y. Govan, Carl D. Meyer, and Rusell Albright. Generalizing google’s pagerank to rank national football league teams. In *Proceedings of SAS Global Forum 2008*, 2008.
- [7] Sepandar D. Kamvar, Taher H. Haveliwala, Christopher D. Manning, and Gene H. Golub. Extrapolation methods for accelerating pagerank computations. In *Proceedings of the 12th international conference on World Wide Web*, WWW ’03, pages 261–270, New York, NY, USA, 2003. ACM.