DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Re Entrega Trabajo Práctico 1

“(No) Todo Pasa”

Metodos numericos

Primer Cuatrimestre de 2016

Integrante	LU	Correo electrónico
Ricardo Colombo	156/08	`ricardogcolombo@gmail.com`

Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

http://www.fcen.uba.ar

Resumen

En el mundo de las competencias deportivas existen numerosos rankings por los cuales se otorgan premios económicos o accesos a otro tipo de competencias, beneficiando a ciertos equipos y perjudicando a otros. Por lo que nuestro objetivo será elaborar y analizar algunos métodos para la obtención de rankings con el fin de determinar cuál o cuáles de ellos son más justos a la hora de ubicar a los distintos equipos dentro de los rankings.

Palabras Claves— Sports Analitycs, Rankings, Factorización Matricial, Matriz de Colley.

Índice

1. Introducción teórica

Las Competencias deportivas, de cualquier índole, requieren la comparación de equipos mediante la confección de tablas de Posiciones y ranking en base a los resultados obtenidos durante un cierto periodo de tiempo.
Algunos de los métodos de ordenamiento utilizados normalmente se basan en proporción de victorias sobre partidos jugados, aunque no todos, sin embargo estos no logran mostrar las dificultades presentadas en estas competencias. A su vez este tipo de rankings son utilizados para clasificaciones de otras competencias o el fin de un equipo dentro de las categorías en las cuales algunas de estas se encuentran divididas estas competencias, teniendo impactos económicos en los clubes, demostrando que no son del todo justas estos métodos utilizados.
En este trabajo practico, intentaremos modelar y resolver el problema de generar un ranking de equipos a partir de los resultados, exceptuando en un principio los empates, luego discutiremos como podríamos modelar esto. Para confeccionar dicho ranking presentaremos los siguientes métodos:

- El Porcentaje de Victorias **WP** : Este método básicamente es el que nombramos anteriormente, se basa en armar el ranking en base al porcentaje de victorias sobre partidos totales.
- Colley Matrix Method **CMM** :Este es un método matricial se basa en el armado de un ranking en base a los partidos ganados, perdidos y los partidos totales entre equipos.

Como mencionamos anteriormente el método de Colley, es un método matricial, por lo que para la resolución de este sistema utilizaremos los métodos vistos en la catedra para encontrar los valores del ranking a la solucion Ax=b dichos métodos son:

- La Eliminación Gaussiana **EG**
- La Factorización de Cholesky **CL**

Luego se procederá a un análisis de los distintos métodos de resolución de sistemas lineales, EG y CL, en base a tiempos y tamaños de sistema a resolver, por otro lado estaremos analizando los resultados obtenidos del método de Colley, haremos una comparativa contra el método de porcentajes de victorias para finalizar realizando algún tipo de conclusión respecto a la justicia de dichos métodos frente a los resultados. Todos estos resultados serán representados mediante gráficos comparativos con sus respectivos detalles.

2. Desarrollo

2.1. Porcentaje de Victorias

Primero analizaremos la técnica que se basa a el **Porcentaje de Victorias**, que a lo largo del análisis denominaremos **WP**,la misma analiza la performance de un equipo o participante en base a los partidos ganados sobre el total de sus partidos jugados para el armado del ranking.

La implementación consiste en calcular el ranking r_i de un equipo i, en base a sus partidos ganados, denominemosolo w_i, sobre los partidos totales. Dado a que los empates no se cuentan, los partidos totales los tomamos como la suma de partidos ganados mas los partidos perdidos, estos ultimos denominemoslos l_i.

$$\mathrm{r}_i = \mathrm{w}_i \frac{}{w_i+l_i}.$$

En este caso el puntaje de un equipo no es afectado por los resultados de los demás participante, si afecta su posición final en el ranking. Otra observacion que se puede hacer es que cada partido perdido para el equipo le hace perder valor en el ranking ya que el dividendo es mayor.

Mostremos esto con un ejemplo entre 3 equipos para ver como se ve afectado: Supongamos que tenemos 3 equipos **A**, **B** y **C** al principio, como ninguno disputo ningún partido todos tienen valor 0 en la tabla de posiciones.

Ahora supongamos la siguiente de partidos y veamos cómo queda la tabla de posiciones

- **A** vs **B** => GANA **A**
- **A** vs **C** => GANA **C**

Posición	Equipo	Ranking
1	C	1
2	A	1/2
3	B	0

Como puede observarse en la tabla, el equipo **B** quedo ultimo ya que no gano ningún partido. La diferencia entre el equipo **A** y el equipo **C** es la cantidad de partidos jugados, ya que ambos ganaron la misma cantidad pero como mencionamos anteriormente como el dividendo es más grande hace que el equipo **C** quede mejor posicionado. Veamos que sucede si juegan el equipo **B** vs equipo **C** y gana el equipo **B**.

Posición	Equipo	Ranking
1	C	1/2
2	A	1/2
3	B	1/2

Como podemos ver, todos quedan con igual ranking y esto es ya que todos jugaron la misma cantidad de partidos y ganaron la misma cantidad de partidos.

Con este ejemplo se puede observar lo que mencionamos anteriormente, que si bien los partidos de los demás no afectan a cambiar el valor de un equipo dentro del ranking si puede afectar su posición, a su vez esta técnica no aporta mucha información respecto a la posibilidad de victoria en el siguiente encuentro y tampoco considera el ranking del rival enfrentado, con lo cual la posición del equipo en el ranking depende de los resultados de victorias del equipo. Por otro lado podemos ver que nunca mencionamos los empates con este método, por ende podríamos analizar como podríamos modelarlos y como afectaría el ranking.

2.2. Sistema a resolver

Veamos el otro método para obtener rankings para poder comparar y ver cual nos parece más justo a la hora de utilizarlos para competencias reales. Comencemos armando el sistema de la Matriz de Colley basados en el paper de Colley[1], este método está basado en la Regla de Laplace de sucesos y solo se requiere conocer un historial de partidos. El método de **CMM** propone construir una matriz C $\in R^{nxn}$, un vector b $\in R^n$ tal que el ranking r $\in R^n$ buscado sea la solución al sistema : $Cr = b$ **(1)**
Para cada equipo i llamamos n_i a la cantidad total de partidos jugados por el equipo i, w_i como la cantidad victorias el equipo y, análogamente l_i los partidos perdidos. Por último definimos $n_{i,j}$ la cantidad de partidos jugados entre el equipo i y j. Dadas estas definiciones Colley propone armar la matriz como:

$$C_{i,j} = \begin{cases} -n_{i,j} & \text{si } i \neq j \\ 2 + n_i & \text{si } i = j \end{cases} \quad \textbf{(2)}$$

y para cada posición correspondiente al equipo i, el $b_i = 1 + (w_i$ - $l_i)$ / 2.

Si vemos la definición de la matriz, podemos notar que es estrictamente diagonal dominante ya que la suma de todas las posiciones de la fila, excepto el elemento correspondiente a la diagonal, suman la cantidad de partidos totales n_i, siendo este es el valor de la posición en la diagonal menos 2. Por otro lado, tiene la particularidad de ser simétrica y definida positiva y, y como consecuencia de todas estas propiedades podemos decir que la matriz es no singular, permitiéndonos utilizar la eliminación gaussiana sin pivoteo y obtener la factorización de Cholesky de dicha matriz. Nos centraremos en ambos métodos para la resolución del sistema **(1)** que planteamos arriba, combinándolo con algoritmos de sustitución como son el back substitution y forward substitution.

Volviendo a las características del método de CMM, a simple vista podemos ver que al igual que en WP no se tiene en cuenta el margen de victoria, no modela los empates y tampoco nos da indicio de como impactara en el ranking cada uno de los resultados de los partidos.

En la sección de experimentos, veremos si con este método importa contra quien se gana y contra quien se pierde basándonos en las posiciones y analizaremos que pasara con los empates. Nuestra intuición nos dice que importa más a quien se le gana, pensamos que no es lo mismo ganarle al que esta último que ganarle al que esta primero y en esto nos vamos a basar para realizar nuestras experimentaciones en las secciones de más adelante, para esto utilizaremos un algoritmo greedy tomando al equipo que salga último en el ranking y haciéndolo jugar con otro equipo que este mejor posicionado que él, utilizaremos 2 heurísticas distintas con el fin de obtener una mejor posición, la primera será contra el inmediato siguiente en el ranking y la otra contra el que este primero, siempre tomando el ranking que resulta luego de cada partido.

2.2.1. Eliminación Gaussiana

Sea C es la Matriz de Colley y c_i los elementos de la misma , buscamos obtener el vector r mostrado en la ecuación del sistema a resolver.

$$Cr = b$$

El algoritmo de Eliminación Gaussiana es útil para la resolución de sistemas lineales, con el cual mediante intercambio y combinaciones lineales entre filas podemos llegar a una matriz escalonada por

[1]"**The Colley Matrix Explained**:Wesley N. Colley"

reglones con ceros debajo de los elementos de la diagonal.
Una vez triangulada la matriz, mediante el algoritmo denominado back substitution se realiza el cálculo para obtener el valor de las incógnitas tomando cada elemento como

$$x_i = (b_i - \sum_{j=i+1}^{n} c_{ij} x_i)/c_{ii}$$

La matriz que nos propone armar el sistema de Colley es no singular y diagonal dominante, como mencionamos anteriormente, por lo que sabemos que en la diagonal nunca nos vamos a encontrar con un 0, por lo que en el algoritmo nos vamos a saltear esa validación ya que el intercambio de filas no ocurrirá nunca.

A continuación presentamos un pseudo código del algoritmo de eliminación gaussiana que utilizamos en nuestra implementación, para el mismo tomamos como base el seudocódigo del libro de Burden [2],

TP1 1 vector Gauss(matriz A, vector b)

Para k=1...$n-1$
    Para i=1...$n-1$
        Se toma el elemento $a_{k,k}$ como pivot
        Para $j = i+1, ...n$
            $a_{i,j} = a_{i,j} - a_{i,j} * (a_{i,k}/a_{k,k})$
            $b_i = b_i - a_{i,j} * (a_{i,k}/a_{k,k})$
return backwardSubstitution(A,b)

TP1 2 vector backSubstitution(matriz A, vector b)

$x = vector(Cantidad\_Columnas(A))$
$x_n = a_{n,n+1}/a_{n,n}$
para $i = n-1.,1$
    para $j = i+1..n$
        $sum+ = a_{i,j} * x_j$
    $x_i = \frac{(b_i - sum}{a_{i,i}}$

return x

Este algoritmo tiene complejidad temporal de $O(n^3)$ para la eliminación Gaussiana y $O(n^2)$ para el algoritmo de back substitution dando un total de $O(n^3)$ como consto temporal para obtener una solución a nuestro sistema de ecuaciones lineales.

[2]"**Análisis Numérico** :Richard L. Burden J. Douglas Faires"

2.2.2. Cholesky

La Matriz del sistema a resolver cuenta, como mencionamos anteriormente, la propiedad de ser simétrica y definida positiva. Estas condiciones nos garantizan que la misma tenga otra factorización del tipo $C = LL^t$, esta se denomina la factorización de Cholesky, compuesta por una matriz Triangular inferior L y la transpuesta de la misma matriz L^t. Donde las posiciones de esta matriz están compuestas de esta manera

$$l_{ji} = \begin{cases} \sqrt{c_{ii} - \sum_{k=1}^{i-1} (L_{ik})^2} & si \quad i = j \\ \\ \frac{c_{ji} - \sum_{k=1}^{i-1} L_{jk} l_{ik}}{l_{ii}} & si \quad i \neq j \end{cases}$$

Una vez obtenido el sistema $LL^t x = b$ se procede a hacer back substituion para resolver $L^t x = y$, luego mediante forward substitution se obtiene la solución de $Ly = b$.

La implementación de la factorización de Cholesky que elegimos, al igual que la eliminación gaussiana, se encuentra en el libro **Burden**. Este seudocódigo representa nuestra implementación sobre la factorización de Cholesky.

TP1 3 vector Cholesky(matriz A, vector b)

```
l_{1,1} = sqrt(a_{1,1})
Para j = 2,...n
    l_{j,1} = a_{j,1}/l_{1,1}
Para i = 2,...n − 1
    l_{i,i} = (a_{i,i} − Σ_{k=1}^{i−1} l_{i,k}^2 ^{1/2})
    Para j = i + 1,...n1
        l_{j,i} = (a_{j,i} − Σ_{k=1}^{i−1} l_{j,k} l_{i,k}/l_{i,i})
l_{n,n} = (a_{n,n} − Σ_{k=1}^{n−1} l_{n,k}^2)
y = backSubstitution(L,b)
x = forwardSubstitution(L,y)
return x
```

Como se detallo en el algoritmo, se utiliza el metodo de backSubstitution al igual que en la eliminacion gaussiana y por otro lado se utiliza el forward subsittution que es similar al metodo mencionado anteriormente para la obtencion de los valores de las incognitas. El algoritmo de este metodo es el siguiente:

La factorizacion de Cholesky tiene complejidad temporal de $O(n^3)$ para Cholesky y $O(n^2)$ para los métodos de forward substitution y back substitution, quedando un total de $O(n^3)$.

TP1 4 vector forwardSubstitution(matriz L, vector b)

$y_1 = \frac{b_1}{l_{1,1}}$
Para i=2..n
  Para j=1..i-1
    sum=$l_{i,j} * y_j$
    $y_i = \frac{b_i - sum}{l_{i,i}}$
return y

3. Resultados y Discusión

Para analizar la efectividad y ecuanimidad de esta nueva forma de calcular el ranking vamos a realizar una serie de test a fin de obtener un análisis cuantitativo y cualitativo que nos permita compararlo con el clásico método de **WP**.
Con los test esperamos encontrar ventajas y desventajas de esta forma de medición, particularmente en escenarios donde no todos los participantes jueguen la misma cantidad de partidos.

Además realizaremos una comparación de los métodos de **Eliminación Gaussiana** y **Cholesky** para ver cuál de los dos computa los rankings de manera más eficiente.

En esta sección solo presentaremos los experimentos realizados y los resultados obtenidos. Las conclusiones de cada experimento las presentaremos en la siguiente sección.

3.1. Ranking

Vamos a comparar la tabla de ranking obtenida a partir de un set de datos de la **ATP 2007**. Es decir calculamos el ranking a partir de la técnica **WP**, considerando partidos ganado partidos jugados, a pesar de que no todos los jugadores hayan participado de la misma cantidad de partidos. Comparándolo con el **CMM** implementado con **Cholesky**.

WP- CMM todos.png

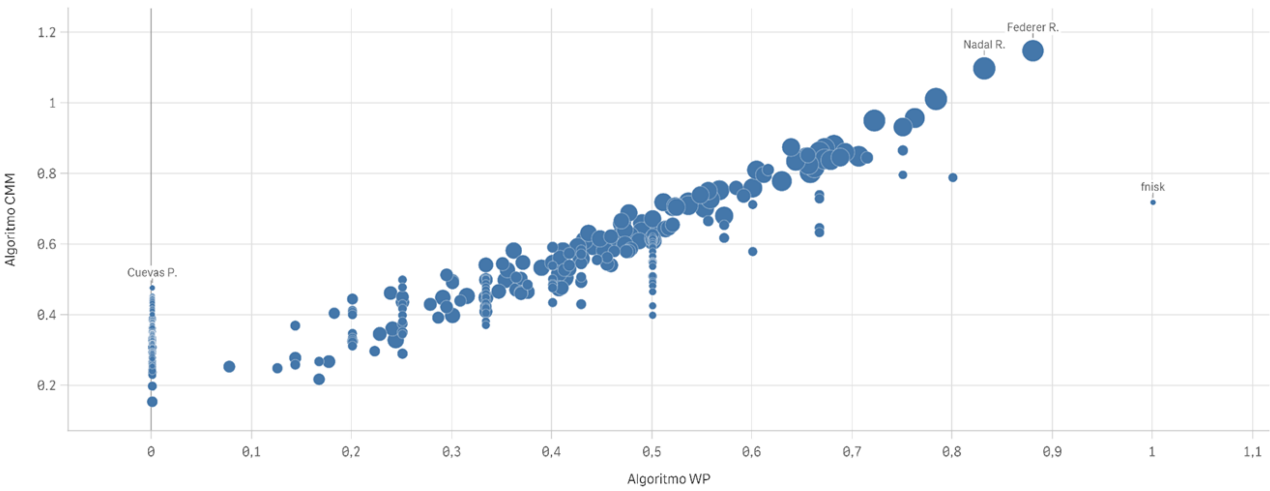


Figura 1: Rankings para comparar WP vs CMM

El grafico elegido para graficar es un gráfico de dispersión, que muestra:

- Eje X el valor obtenido al ejecutar el algoritmo WP.
- Eje Y el valor obtenido al ejecutar CMM implementado con Cholesky.
- El tamaño de la burbuja es la cantidad de partidos jugados.

Lo que se observa en el grafico es que el ranking CMM parece darle relevancia a la cantidad de partidos jugados ya que las burbujas de los top 10 son más grandes. Esto está dado en principio por la característica del deporte (tenis) que permite a los que ganan jugar más partidos en los torneos. Lo que nos llamó la atención es que si observamos el Rank arrojado por WP, se observa que el primero es fnisk. El mismo es un jugador que solo jugo 1 partido y lo gano, por lo cual tiene un 100 % de efectividad y figura como primero.
Además, si observamos los top 10 del Rank WP, observamos algunas burbujas de tamaño pequeño. Esto es porque en algunos casos, este ranking beneficia jugar pocos partidos pero ganarlos.

Hemos realizado un zoom dentro del grafico para corroborar el nombre y posición de ambos rankings.

WP - CMM zoom.png

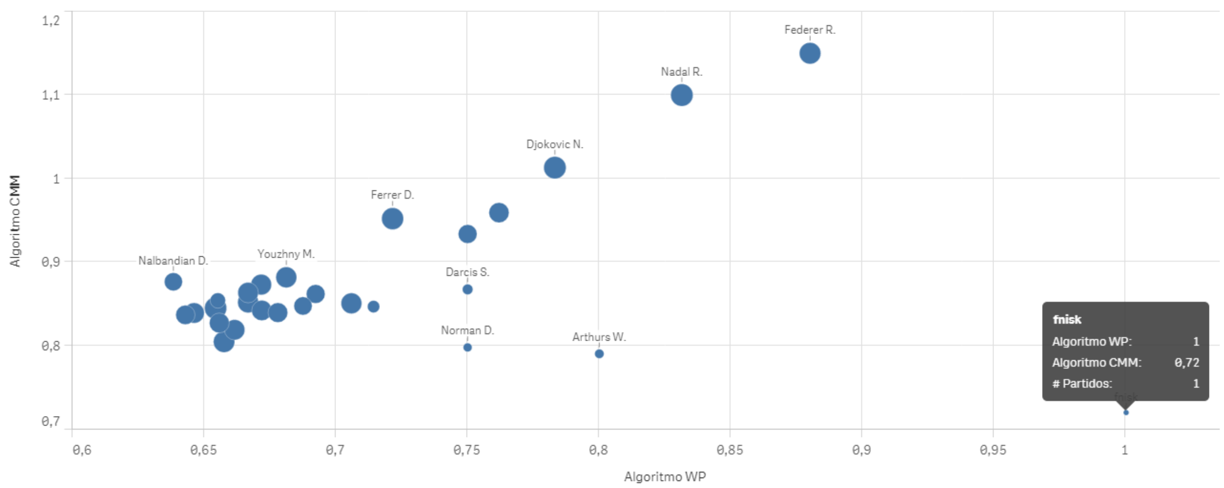


Figura 2: Zoom de las primeras posiciones

En el grafico se resalta el jugador fnisk y se muestra información adicional.

3.2. Justicia

A la hora de hablar de justicia de los rankings primero debemos preguntarnos, justicia respecto a que o cual objetivo. Si observamos las pruebas para el ranking de CMM notamos ciertos movimientos de algunos equipos que no jugaban. Estos movimientos no eran esperados por nosotros por lo que realizamos una prueba para un campeonato pequeño con el fin de replicar este problema.

Supongamos que tenemos un campeonato de 5 equipos si vemos el método de CMM, la matriz C y el vector b que obtendríamos en un momento inicial serian

$$C = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Donde cada posición $c_{i,j}$ representa lo que mencionamos en la sección desarrollo de cómo se construye la matriz.

Si resolvemos el sistema **(1)** con esta matriz y este vector obtendremos como vector de rankings inicial.

$$\mathrm{r} = \begin{pmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{pmatrix}$$

Si vemos el vector solución son los valores iniciales del ranking, ahora, supongamos que juega el equipo **0** y gana en el partido contra el equipo **1**, Que sucede con el ranking y los valores de cada una de las partes del sistema luego de modificarlo y buscar la solución para el ranking?

$$C = \begin{pmatrix} 3 & -1 & 0 & 0 & 0 \\ -1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix} b = \begin{pmatrix} 1,500 \\ 0,500 \\ 1 \\ 1 \\ 1 \end{pmatrix} r = \begin{pmatrix} 1{,}1667 \\ 0{,}8333 \\ 0{,}5000 \\ 0{,}5000 \\ 0{,}5000 \end{pmatrix}$$

Como se ve en el resultado cambian los valores de los equipos correspondientes y el ranking se modificaría haciendo que el equipo 0 quede en primera posición y el equipo 1 en segunda, los otros no tendrían modificaciones con lo cual hasta el momento no notamos nada que no esperábamos y no podemos replicar lo que suponemos como érrorv́isto en el gráfico. Si repetimos los pasos pero esta vez hacemos jugar al equipo 0 contra el 2,para que gane el 0, luego volvemos a calcular el ranking dándonos lo siguiente.

$$C = \begin{pmatrix} 4 & -1 & -1 & 0 & 0 \\ -1 & 3 & 0 & 0 & 0 \\ -1 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix} b = \begin{pmatrix} 2{,}0000 \\ 0{,}5000 \\ 0{,}5000 \\ 1{,}0000 \\ 1{,}0000 \end{pmatrix} r = \begin{pmatrix} 0{,}7000 \\ 0{,}4000 \\ 0{,}4000 \\ 0{,}5000 \\ 0{,}5000 \end{pmatrix}$$

Si vemos estos resultados, notamos que no solo se modificaron los puntajes de los equipos que perdieron, si no que a su vez se modificaron los puntajes de un equipo que no jugo que en este caso es el puntaje del equipo 1, no solo eso, si no que a su vez notamos que está manteniendo en cada una de las partidas jugadas el valor de 1/2 como promedio del ranking. Si vemos el paper donde se detalla el método todo se deriva de cómo se construye la matriz C en **(2)**, esta misma matriz, puede ser representada como

$$C = 2I + \sum_{i=1}^{total\ partidos} G_k$$

Donde I, es la matriz identidad y G_k tiene 0 en todas las posiciones excepto en las posiciones $G^k_{i,j} = G^k_{j,i} = -1$ y en la posiciones $G^k_{i,i} = G^k_{j,j} = 1$. Luego si a esta matriz G, la multiplicamos por r quedándonos $r' = G_k * r$ obtendríamos que en las posiciones i e j nos queda $r_i - r_j$ y $r_j - r_i$ respectivamente y cero para las demás. Luego

$$\sum_{i=1}^{N} C * r = \sum_{i=1}^{N} (2 * I * r) = 2 * \sum_{i=1}^{N} r_i$$

A su vez, del otro lado del sistema, para b, podemos ver que resulta

$$\sum_{i=1}^{N} b_i = N$$

Finalmente si igualamos ambos resultados obtenemos

$$\sum_{i=1}^{N} C * r = \sum_{i=1}^{N} b_i = N$$
$$2 * \sum_{i=1}^{N} r_i = N$$
$$\implies \frac{\sum_{i=1}^{N} r_i}{N} = \frac{1}{2}$$

Por lo que vemos que este promedio es siempre $\frac{1}{2}$ y esto explica de porque se modifican un tercero en los rankings cuando juegan dos equipos, esto no es lo que esperabamos nosotros en cuanto al metodo y tampoco parece ser muy intuitivo cuando se lo implementa. En caso de querer mas detalles sobre este metodo recomendamos leer el paper de Colley mensionado en la bibliografia.

Por otro lado, que sucede con el método de WP, como marcamos durante el análisis del mismo solo importa la cantidad de partidos totales y la cantidad de partidos que ganaste, pero no el resultado del partido entre otros equipos/jugadores.

Concluyendo que consideramos que los metodos parecen ser justos en cuanto al ranking que arman, en las siguientes secciones mostraremos como podremos escalar en el ranking y asi mismo como se podriamo modelar los empates y si valen la pena para definir si los metodos son justos.

3.3. ¿Importa a quien se le gana?

En el escenario que se utiliza **WP** realmente no importa a que equipo se le gane, ya que todos los partidos tienen la misma importancia y se les asigna el mismo puntaje. Pero en el caso de **CMM** resulta más interesante plantearse esta pregunta.

La hipótesis que tenemos es que tomando un equipo de mitad de tabla, que denominamos **medio** el hecho de que le gane al líder de la tabla va a mejorar mucho más el ranking que derrotando al que ocupe la última posición.

Realizamos un test tomando al equipo **medio**, y agregando un partido victorioso contra el puntero y analizamos como se modifica su ranking. Luego tomamos la tabla inicial, es decir sin ganarle al puntero, y repetimos el experimento esta vez derrotando al último.

Presentamos los resultados obtenidos.

Cuadro 1: Posicion Inicial

Posición	Jugador	Ranking
1	Federer	1,131919
164	Vasallo Arguello	0,474265
334	Vicente F.	0,198425

Cuadro 2: Ganandole al Primero

Posición	Jugador	Ranking
1	Federer	1,117437
141	Vasallo Arguello	0,508447
334	Vicente F.	0,198271

Cuadro 3: Ganandole al Ultimo

Posición	Jugador	Ranking
1	Federer	1,132469
141	Vasallo Arguello	0,480302
334	Vicente F.	0,170821

Se observa una mejoría en el ranking luego de haber ganado contra el primero.

3.4. Racha ganadora

Para estudiar la ecuanimidad del **CMM** realizamos un experimento tomando al participante del **ATP 2007** que se encontraba en el último puesto y le asignamos una racha ganadora contra los primeros diez jugadores del ranking.

Además este test nos permite observar como la racha de un jugador afecta al ranking global y si ganándole a los mejores realmente escala una considerable cantidad de posiciones en el ranking.

A continuación presentamos el ranking calculado construido de la siguiente manera:

- Eje X el valor obtenido al ejecutar CMM implementado con Cholesky.
- Eje Y el valor obtenido al ejecutar CMM implementado con Eliminación Gaussiana.

- El tamaño de la burbuja es la cantidad de partidos jugados.

cmm -cmm foto 0.png

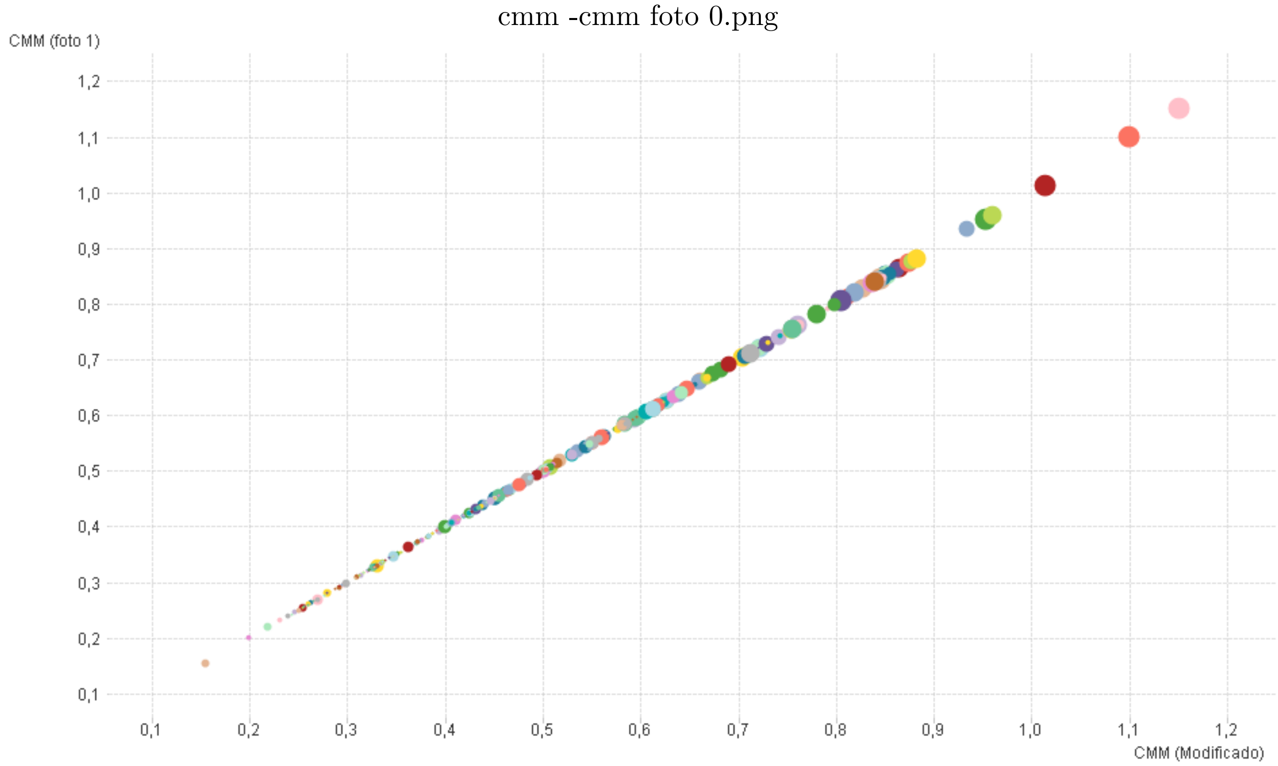


Figura 3: Ranking de jugadores

Es lógico que se observe una diagonal ya que ambos ejes son la misma métrica de CMM.
Luego el experimento realizado fue agregar partidos al historial de partidos, de manera que el ultimo le gane a los 10 primeros y entender qué tipo de reacción tiene el algoritmo CMM.
Para ello se realizaron 10 historiales de partidos distintos de tal forma que en cada uno de esos archivos de entrada exista un partido más que en el caso anterior y el mismo sea una victoria del ultimo jugador del ranking vs alguno de los top 10.

Para poder mostrar una evolución a medida que se van jugando los 10 partidos, hemos dejado el eje Y del gráfico con el ranking inicial, mientras que el eje X paso a ser el ranking recalculado con la agregación de partidos.
En el gráfico se observa mediante una flecha el salto del ranking de un partido a otro y evidencia cual es la ganancia que se tiene al ganarle a los top 10.

cmm -cmm foto 10.png

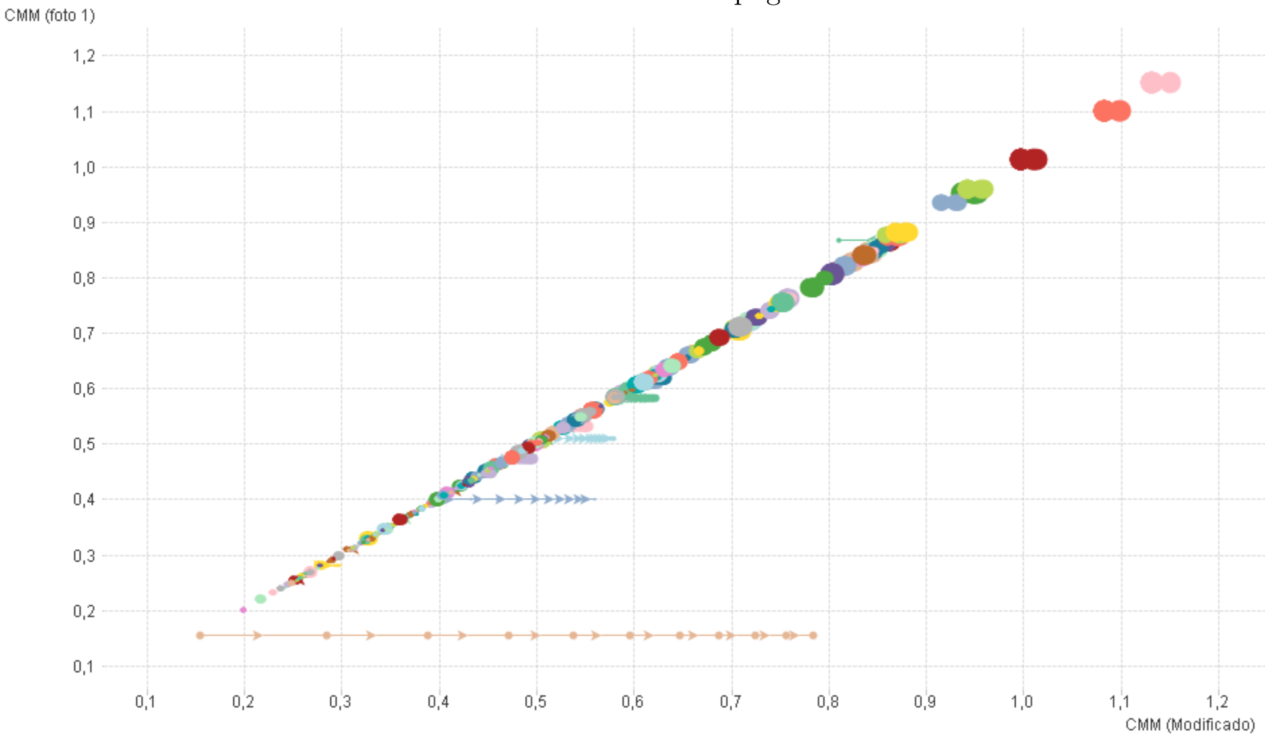


Figura 4: Evolucion de jugadores con el pasar de los partidos

Por último y para entender si el jugador en la última posición podía llegar a ser primero en algún momento, hemos agregado al archivo un total de 100 partidos ganados por el ultimo contra los top 10.

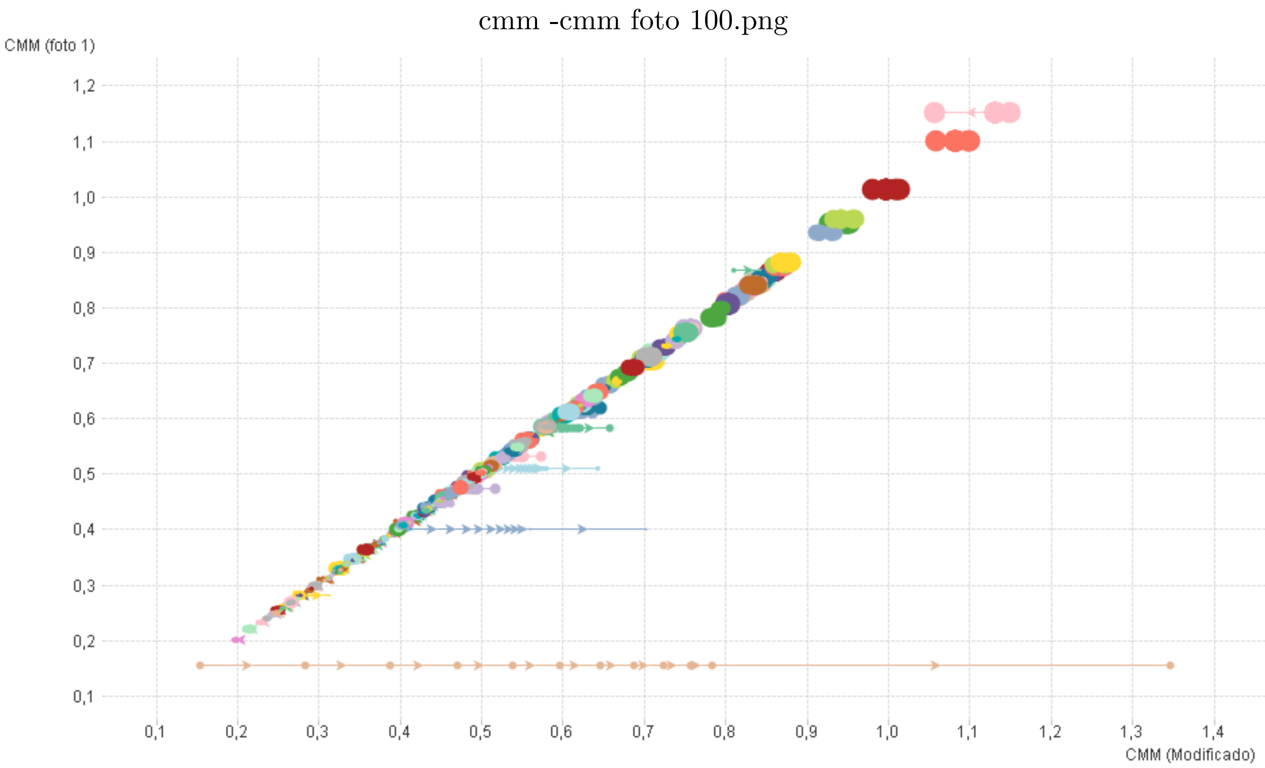


Figura 5: Evolucion de jugadores con el pasar de los partidos

Y tal como vemos, si es posible pero a un costo altísimo (jugar alrededor de 100 partidos).

Lo interesante es que aquellos equipos contra los que el ultimo jugador jugo también sufrieron modificaciones en su ranking.
A continuación se observa un zoom dentro del grafico para observar este comportamiento.

El ultimo jugador del ranking se llama Verkerk M., sus partidos fueron los siguientes:

Cuadro 4: Partidos jugados por Verkerk

Nro Partido	Ganador	Sets Ganados	Perdedor	sets Perdidos
535	de Voest R.	2	Verkerk M.	0
757	Grosjean S.	2	Verkerk M.	0
832	Hanescu V.	2	Verkerk M.	0
902	Chela J.I.	2	Verkerk M.	0
1063	Brands D.	2	Verkerk M.	0
1092	Acasuso J.	2	Verkerk M.	0
1145	Almagro N.	2	Verkerk M.	0
1212	Lapentti N.	2	Verkerk M.	0
1234	Bolelli S.	3	Verkerk M.	0

En el siguiente grafico se observa el movimiento de ranking de los que le ganaron partidos a Verkerk, teniendo en cuenta la escalada al primer puesto de este jugador.

jugados vs verkerk grafico.png

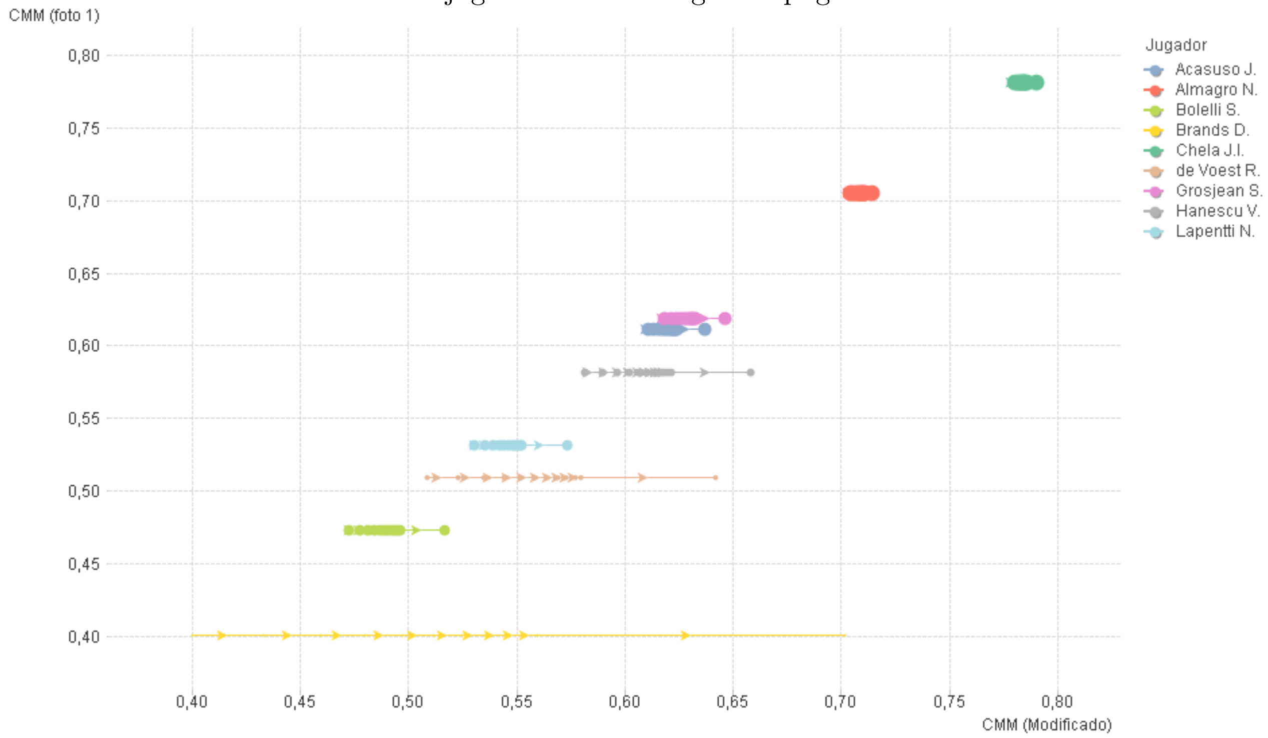


Figura 6: Evolucion de aquellos que le ganaron a Verkerk

3.5. Escalando Posiciones

Una de las consignas del trabajo era encontrar una técnica para hacer escalar en el ranking a un **equipo**, pensamos en 2 técnicas asumiendo que el torneo se encuentra en un punto donde todavía quedan partidos por jugar para saber cuántos partidos debe jugar para llegar al primer puesto y otra para variar los partidos hacia atrás viendo si es posible que llegue al primer puesto.
Las primeras dos se basan en, dado un historial de partidos ya definido donde existe un jugador que esta último en el ranking, hacerlo jugar y ganar, en primer lugar con el que siempre esta primero partido a partido y en otro experimento contra el que tiene inmediatamente arriba en el ranking. Para la otra técnica se realizó un experimento tomando al jugador que este en menor posición con almenos un partido jugado e iremos variando sus partidos para tratar de ver como se modifica el ranking, por ultimo compararemos todas estas etrategias.
Las siguientes estrategias fueron implementadas de manera que los algoritmos terminen la ejecución al llegar al primer puesto, arrojando la cantidad de partidos en total que debió jugar, y ganar, ese jugador para las dos primeras técnicas y para la última cuando se modificaron todos los partidos.

A continuación se muestran los análisis.

3.5.1. Ganarle siempre al primero

5.png

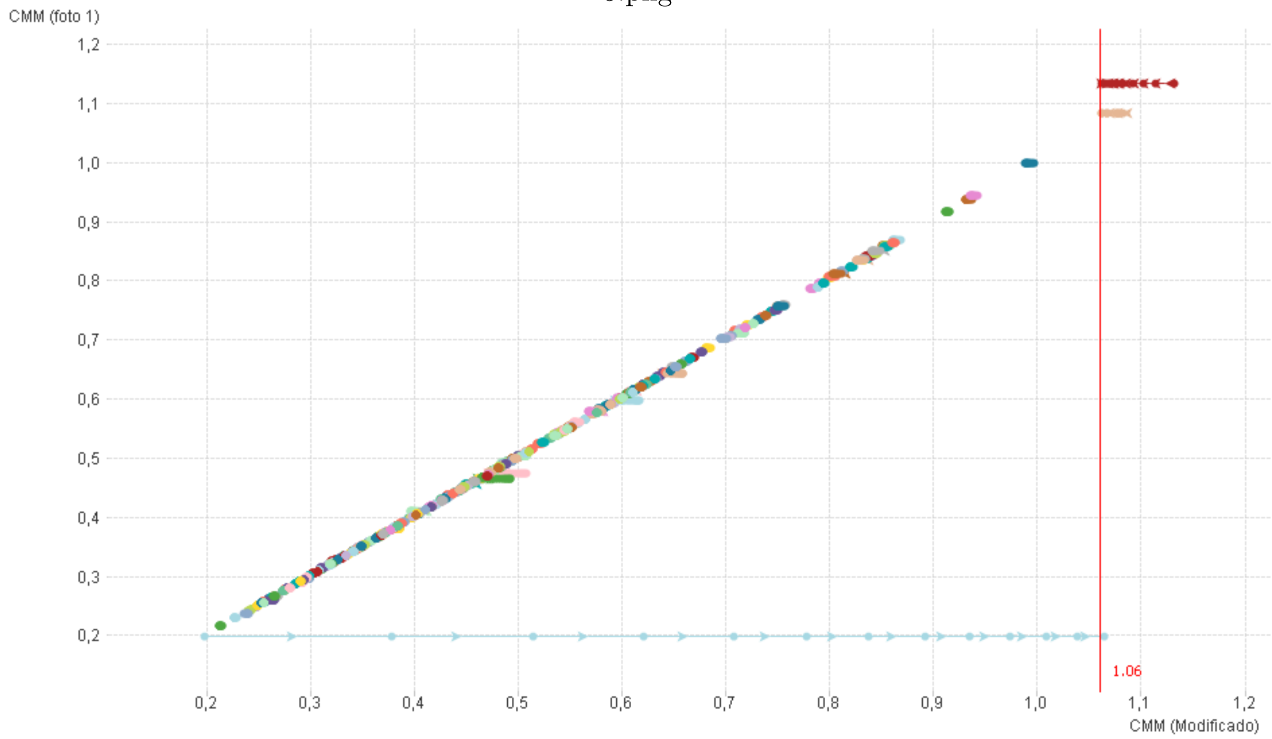


Figura 7: Ganarle siempre al primero

En este grafico se observa lo siguiente.
Cada burbuja es un jugador, el eje de las X es el Rank que arroja CMM para las distintas fechas jugadas, mientras que el eje Y tiene el Rank de CMM base para realizar el análisis y comparar la evolución de los jugadores.

En este grafico se observa que el crecimiento es rápido en un inicio y luego va reduciendo su velocidad de crecimiento. Se observa que el algoritmo necesita de 12 fechas para llegar a la cima del campeonato, validando nuestra teoría.

Se observa también una línea de referencia que indica cuando llega el jugador a superar al primero del ranking.

3.5.2. Ganarle al que esta inmediatamente arriba

4.png

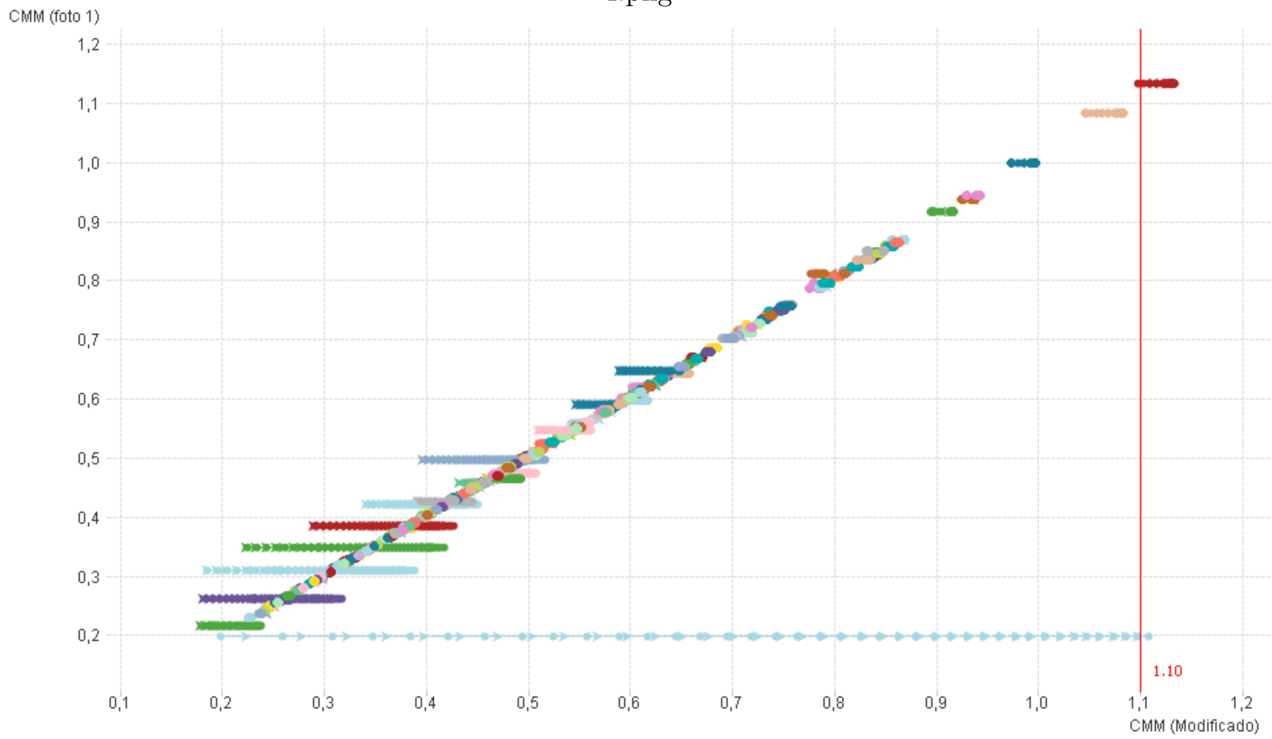


Figura 8: Ganarle siempre al que esta inmediatamente arriba

En este grafico se observa que el crecimiento es lento y que se necesitan aproximadamente 40 fechas para llegar a la cima del campeonato.

Se observa también una línea de referencia que indica cuando llega el jugador a superar al primero del ranking.

3.5.3. Modificando los resultados

Para esta técnica lo que se realizo fue elegir un equipo que se encuentre en la última posición e intentar hacer que suba las posiciones modificando los resultados de sus partidos perdidos.

Porque esto es posible?

Esto es posible porque el resultado del partido, ya sea ganado o perdido, se encuentra en el vector b, con lo cual basta con modificar ese vector para poder ejecutar otra vez el ranking.

Si bien no se utilizó en esta experimentación, podríamos utilizar la ventaja de tener una matriz factorizada para realizar solamente los algoritmos de backgward substitution y forward substitution para obtener los resultados más rápido. De igual manera para implementar esto lo que se hizo fue ejecutar una cantidad de veces igual a la cantidad de partidos ganados del equipo elegido y terminar cuando no tenga más resultados que cambiar o cuando haya llegado a la primera posición. En cada iteración lo que se hizo fue modificar el resultado del partido contra el equipo que perdió y luego volver a calcular el ranking.

Porque elegimos experimentar con el ultimo?

Debido a que con la prueba queremos ver si es posible llegar a la primera posición y empezamos planteando que no es posible para equipos que jugaron pocos partidos. De igual manera estos experimentos podrian realizarse para cualquiera de los equipos, siempre esperando que sucedan cualquiera de los tres escenarios que pensamos que pueden suceder. En primer lugar, si es el equipo que se encuentra primero, va a seguir primero. En segundo lugar, para el equipo que se encuentre en cualquier otra posicion pueden pasar dos cosas, que pase a la primera posicion, o que no peuda llegar nunca que es lo que vamos a ver que puede suceder.

Los resultados fueron los esperados?

Realmente como planteamos anteriormente esperábamos que no llegue a la primera posición, de igual manera el salto que hizo al ganar sus partidos no fue el esperado, llego a mitad de tabla como se puede ver en el siguiente gráfico.

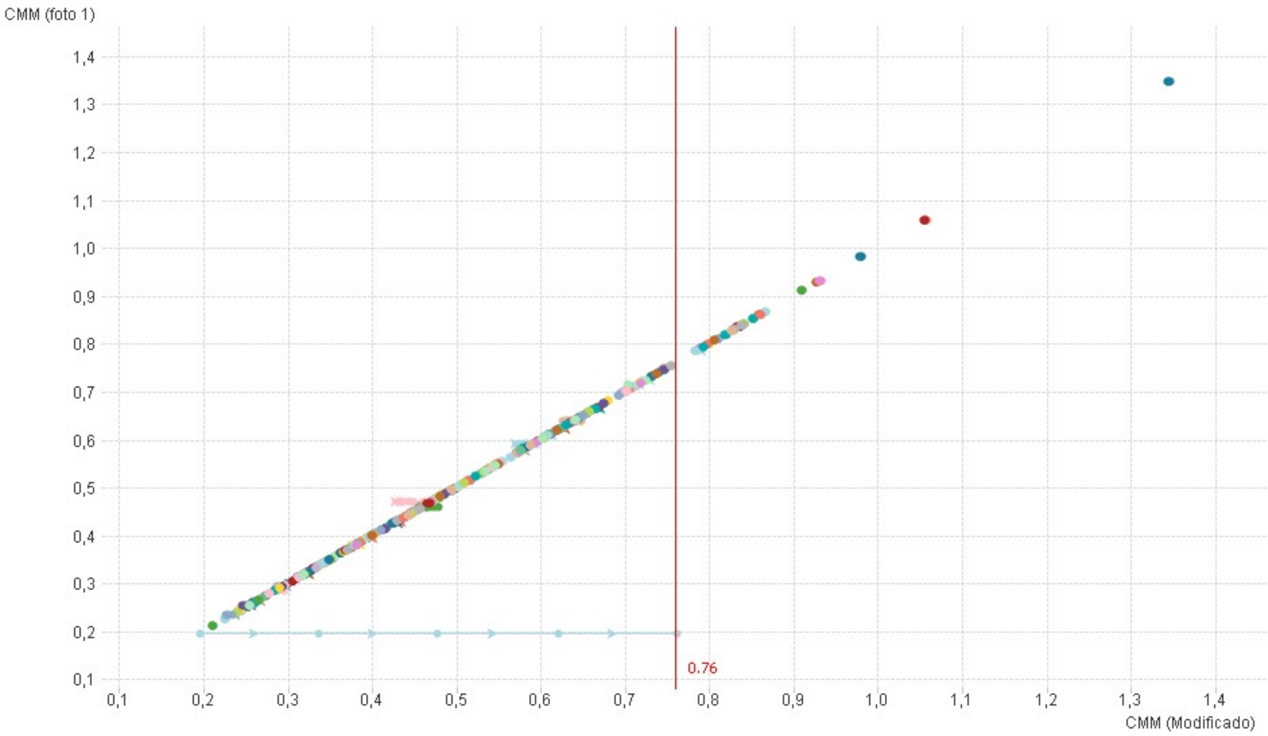


Figura 9: Modificando el vector b

Si analizamos el grafico anterior podemos ver que se modifican 5 partidos,estos son contra equipos que están delante del equipo seleccionado en el ranking y este llega a mitad de tabla.

3.5.4. Comparación de Estrategias

Si nos pusiéramos a analizar las estrategias anteriores con el fin de encontrar similitudes y diferencias, podemos notar que en cuanto a tiempos de ejecución o resolución del sistema tardarían lo mismo, debido a que como mencionaremos en el análisis cuantitativo la cantidad de partidos no afecta el tiempo de resolución del mismo. Por otro lado, la principal diferencia entre una técnica y la otra es que si agregamos partidos a un ranking, siempre llegaríamos a una cantidad de partidos finita para terminar en primera posición, sea la cantidad que sea, diferente es en la técnica de cambiar resultados que podríamos no llegar a estar en primera posición como se ve en el ejemplo anterior, como lo pudimos ver en los gráficos anteriores.

3.5.5. Empate

Dado que ninguno de los metodos modela los empate, decidimos establecer que los empates sean tomados como un partido ganado para cada uno de los equipos, con el fin de ver como se verian afectados los rankings a comparacion del metodo original que no los toma. En ambos metodos se realizaron test con un conjunto de pocos equipos y pocos partidos para ver como resultaban los rankings. Si hacemos el siguiente historial de partidos para 6 equipos.

Equipo1	goles equipo 1	Equipo2	goles equipo 2
1	16	4	13
2	38	5	17
2	0	6	1
3	34	1	21
3	23	4	10
4	31	1	6
5	0	6	1
5	38	4	23
6	1	2	0
6	1	5	1
4	1	6	1
3	1	1	1
3	1	2	1

Para WP obtuvimos los siguientes resultados

Equipo	posicion Sin contar empate	posicion contando empate
1	0.333333	0.500000
2	0.500000	0.666667
3	1.000000	1.000000
4	0.250000	0.400000
5	0.500000	0.666667
6	1.000000	1.000000

Como se puede apreciar en la tabla anterior como se estan contando se ven equipos que tienen 1 en la tabla a y jugaron igual 0 partido perdidos obtuvieron como valor para el ranking un 1. Esto es esperable debido a como es el calculo, en esos casos es total de partidos sobre total de partidos.

Para el caso de CMM, sabiendo que no afecta las propiedades de la matriz y sigue siendo Simetrica definida postiva y diagonal dominante, obtenemos.

1	0.355523	2.478343
2	0.534593	2.601516
3	0.665407	3.009746
4	0.306105	2.420411
5	0.460174	2.499729
6	0.678198	2.988360

Para este caso los cambios son mas significativos, deberiamos de analizar la parte matematica del metodo para saber cual es el riesgo de tomar esta alternativa para tomar los empates.

De igual manera si bien los empates no son tomados en cuenta, para el metodo de WP podria tomarse como un partido perdido para ambos equipos ya que no seria justo para aquellos equipos que ganaron su partido y el que lo perdio contra los que empataron al momento del calculo del ranking, asi mismo podriamos repetir esto dicho para CMM en el tomar empates como partidos perdidos para cada equipo.

3.6. Analisis Cuantitativo

Vamos a estudiar la eficiencia de ambas técnicas incrementando y variando los volúmenes de datos. La idea es repetir el cómputo de los rankings para la misma instancia de datos al azar, y posteriormente ir incrementando la cantidad de datos.

Nuestra hipótesis es el que método de basado en **WP** va a tardar lo mismo para instancias de datos iguales, y se irá incrementando de forma casi lineal a medida que incrementemos los datos. En cambio con **CMM** basando en **Eliminación Gaussiana** y **Cholesky** esperamos que difieran en para las mismas instancias. Nuestra hipótesis sobre esto es que la implementación de **Cholesky** va a demorar menos tiempo.

Para esta prueba se generaron diferentes historiales de partidos variando la cantidad de equipos en 6, 50, 100, 200, 300, 500, 700, 1000 y 2000.

Adicionalmente se varió la cantidad de partidos jugados por cada equipo. Dado el análisis de complejidad de los algoritmos implementados, solo varían el tiempo de calculo en función del tamaño de la matriz definida por la cantidad de equipos, por lo cual al variar la cantidad de partidos no esperamos encontrarnos con variaciones en el tiempo.

Ejecutamos los test para nuestra implementación de Eliminación Gaussiana, a continuación se muestran los resultados de tiempos de ejecución dependiendo la cantidad de equipos. Para evidenciar la complejidad cubica del algoritmo lo hemos encerrado entre 2 funciones cuadráticas que evidencian que no pueden contener la curva de tiempos de nuestro algoritmo.

cuadrativo.png

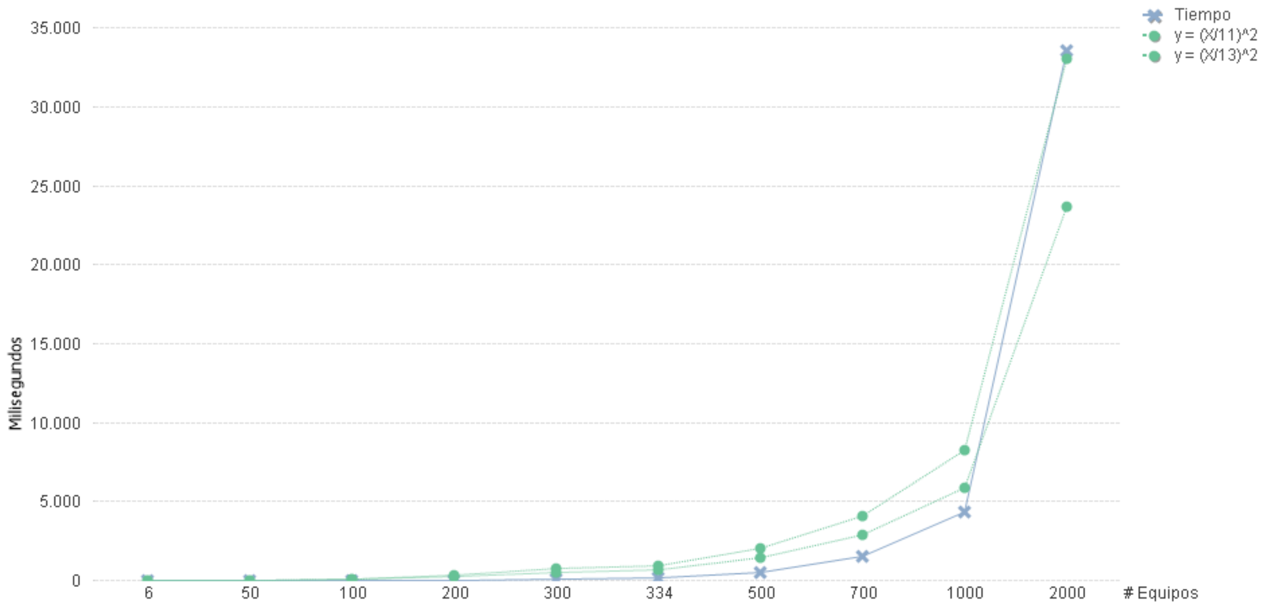


Figura 10: Gauss cuadratico

Luego observamos la misma gráfica pero con líneas de referencia de 2 funciones cubicas.

cubico.png

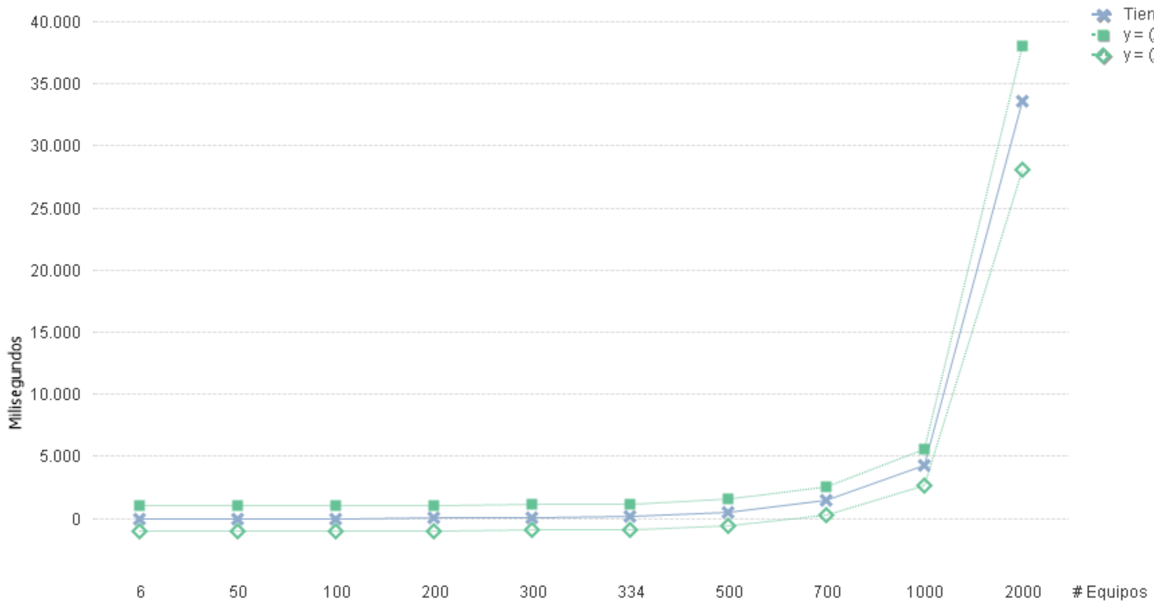


Figura 11: Gauss cubico

Ejecutamos los test para nuestra implementación de Cholesky. A continuación se muestran los resultados de tiempos de ejecución dependiendo la cantidad de equipos. Para mostrar la complejidad cúbica del algoritmo lo hemos encerrado entre 2 funciones cuadráticas que evidencian que no pueden contener la curva de tiempos de nuestro algoritmo, pero si en la cúbica como evidenciamos en los algoritmos cuando presentamos la complejidad de los mismos.

cuadratico.png

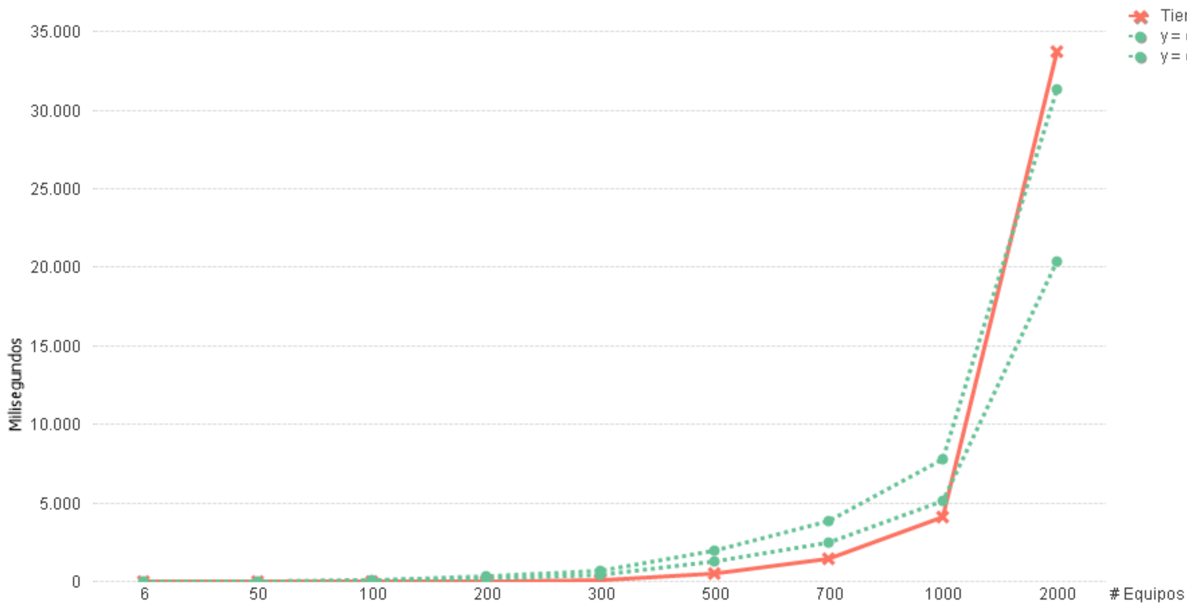


Figura 12: Cholesky cuadratico

Luego observamos la misma grafica pero con líneas de referencia de 2 funciones cúbicas.

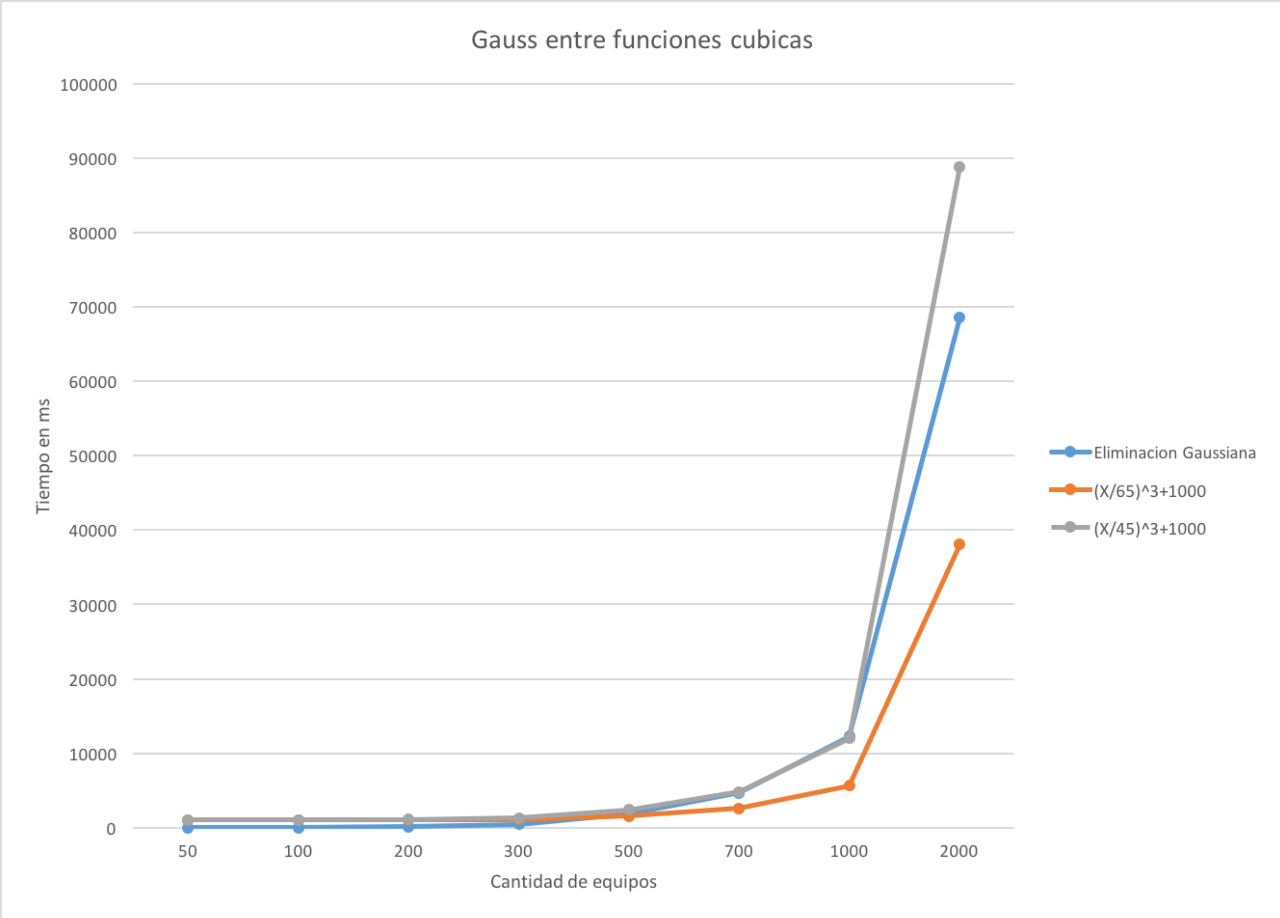
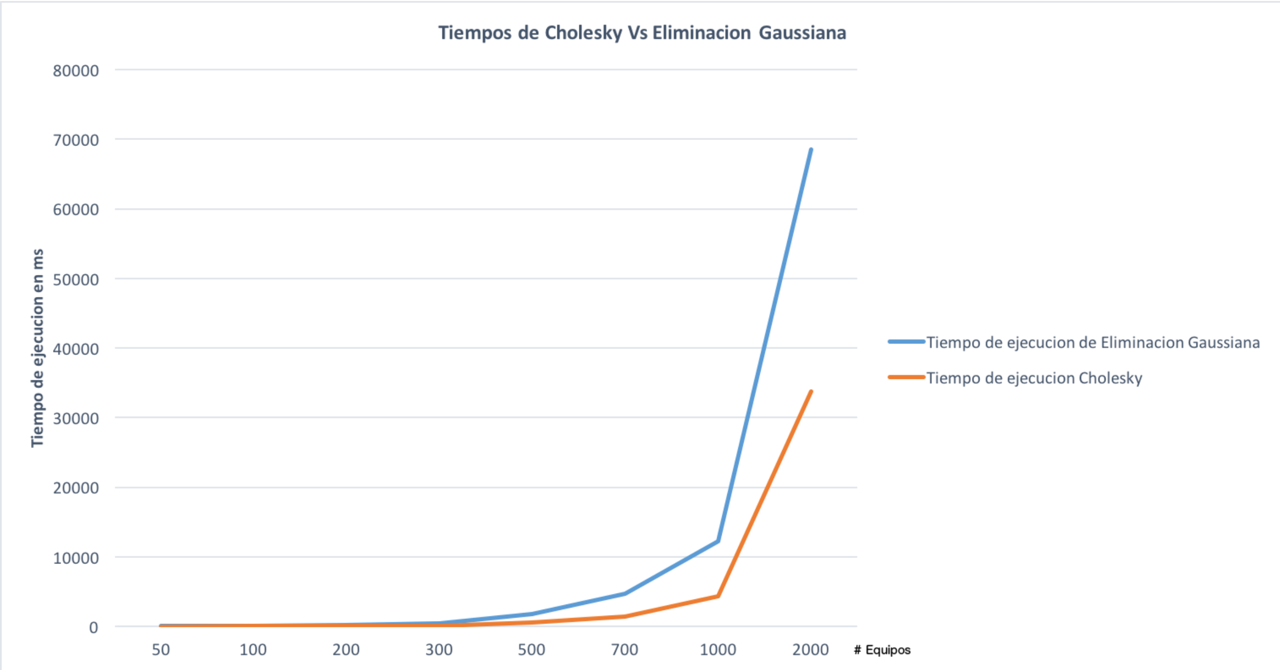


Figura 13: Cholesky cubico

Si comparamos ambos algoritmos, podemos comprobar que nuestra hipótesis de que Cholesky demora menos tiempo es correcta, como se puede ver en la siguiente gráfica.



Ejecutamos los test para nuestra implementación de WP. A continuación se muestran los resultados de tiempos de ejecución dependiendo la cantidad de equipos:

lineal.png

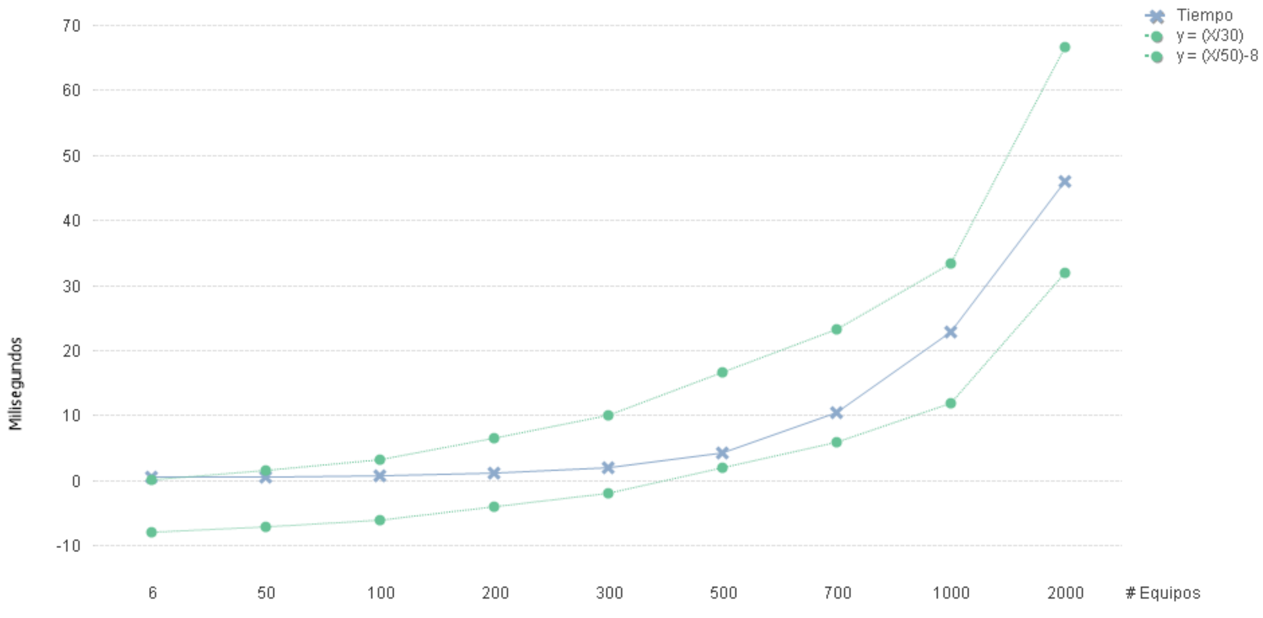


Figura 14: WP lineal

Cuando leímos el enunciado encontramos una frase que nos llamó la atención y era la siguiente:

Se pide comparar, para distintos tamaños de matrices, el tiempo de computo requerido para cada método en el contexto donde la información de la matriz del sistema C se mantiene invariante, pero varia el termino independiente b

Como marcamos anteriormente, para Cholesky no aprovechamos la ventaja de tener factorizada la matriz con el fin de acelerar los cálculos cuando modificamos el termino independiente, dado a que solamente hariamos uso del back substitution y forward substitution, ya que el cálculo de la factorización no requiere del termino independiente como si lo utiliza la eliminación gaussiana. Esto se puede apreciar mirando los algoritmos, donde Eliminación Gaussiana si hace uso del término independiente para el calculo de las filas de la matriz factorizada, a diferencia de Cholesky que lo utiliza solo para los algoritmos de reemplazo. De haber realizado esto notaríamos cambios significativos en los tiempos de Cholesky para estas pruebas.

4. Conclusiones

Consideramos que ambos criterios se acercan a la realidad, a pesar de que no modelan los empates, de igual manera, nos inclinamos hacia el uso de WP por sobre el método de Colley debido a la observación de que el encuentro de dos equipos modifica a un tercero.

Respecto a que implementación, a pesar de no haber usado la matriz factorizada, es más performante el método de Cholesky y nos inclinaríamos hacia el uso de este algoritmo en caso de ser necesario por sobre la Eliminación Gaussiana, siempre y cuando se cumplan las condiciones necesarias para la solución del sistema.

En contados deportes el resultado puede predecirse de antemano, debido a las características de los rivales, como en el caso del Polo. Caso Contrario, no puede aplicarse a deportes como el Fútbol, donde en innumerables ocasiones el equipo menos favorito termina llevándose el partido.
Hay casos en donde Sports Analitycs parece haber dado buenos resultados, sin ir más lejos en el último mundial de futbol donde gano Alemania, se utilizó Sports Analistycs con una herramienta llamada SAP Sports One con la cual se pueden saber todas las estadísticas de los jugadores y se podrían tener datos de los rivales para realizar cierto análisis previo a cada partido, para este dejamos una nota de un diario canadiense en la bibliografía (item 3 bibliografía).

Por último sobre la frase **La utilización de técnicas avanzadas de análisis de datos son imprescindibles para mejorar cualquier deporte**, y en base a las películas vistas , consideramos que el Análisis de datos es una herramienta importante para mejorar el rendimiento deportivo de un equipo, siempre y cuando se utilicen correctamente. Las estadísticas son indicadores de que tan bien rinden los jugadores pero hay factores externos como el ánimo, el nivel de los competidores que no son considerados en las estadísticas ya que no son medibles y pueden modificar bastante el rendimiento de los jugadores.Afortunadamente la frialdad de los números no es aplicable a la pasión de todos los deportes.

5. Apéndice

5.1. Entrada y salida de los algoritmos

Dados los requerimientos de la catedra el programa toma como parámetros 3 argumentos, el primero es el archivo de entrada, luego el archivo de salida y por último el modo. La catedra solicitaba 3 modos el modo 0,1 y 2 para los 3 métodos solicitados para ejecutar sobre la matriz Colley, luego agregamos 3 modos más utilizados durante la etapa de experimentación.

1. Eliminación Gaussiana(EG)
2. Factorización de Cholesky(CL)
3. WP
4. Cholesky con modificación de partidos jugados
5. Cholesky haciendo ganar al último con el siguiente reiteradas veces hasta quedar primero
6. Cholesky haciendo ganar al último con el primero del ranking hasta quedar primero

En todos los modos como paso previo a la realización de algunos de los métodos arma la matriz de Colley, explicada en la sección siguiente , para luego utilizando los métodos 0-1 en el programa (1 o 2 en la lista) resolver el sistema pedido.

Se agregó un parámetro más que puede ser enviado de manera opcional en caso de probar los resultados con empates y sin empates, 1 y 0 respectivamente a los parámetros anteriores.

El modo 3 corre Cholesky, como paso siguiente busca 2 equipos que hayan jugado previamente para cambiar su resultado y luego volver a ejecutar Cholesky, Este modo contiene un ciclo para repetir esta operación 100 veces.
Para el modo 4 corre Cholesky y luego ejecuta un ciclo donde el objetivo es lograr que el que haya salido último luego de obtener el ranking de Cholesky llegue al primer puesto ganándole al que tiene por arriba inmediato en el ranking. En cada paso agrega un partido más y vuelve a calcular Cholesky para la nueva matriz, así obtenemos un nuevo ranking y continuamos iterando hasta quedar en la primera posición, siempre utilizando al que salió último en la primera utilización del método de Cholesky.

Una vez finalizado por stdout devuelve la cantidad de partidos jugados, además de que en el archivo rankingSTEPS_4.out dentro de la carpeta test se encuentran los rankings en cada iteración y como es la evolucion. El modo 5 es similar al anterior con la sutil diferencia que el participante que salió ultimo la primera vez que corrió cholesky llegue al primer puesto jugándole al que se encuentra en el primer puesto en cada iteración.

Tanto el formato de entrada y de salida del programa son los solicitados por la catedra. Para todos los métodos el archivo de entrada es el mismo, que contiene el siguiente formato:

$$\begin{pmatrix} (n) & (k) & & & \\ (x1) & (e1) & (r1) & (t1) & (s1) \\ (x2) & (e2) & (r2) & (t2) & (s1) \\ ... & & & & \\ (xk) & (ek) & (rk) & (tk) & (s1) \end{pmatrix}$$

La primer línea tiene 2 valores n representa la cantidad de equipos y k representa la cantidad de partidos, seguido k líneas que representa cada partido, donde $x1$ representa una fecha que en nuestro

caso no utilizamos, luego *ei* y *ti* representan los números de los equipos, y por último *ri* y *si* representan las anotaciones de cada equipo respectivamente.
en nuestra experimentación con fines de no complejizar aún más el problema no utilizamos en campo que representa la fecha si no que asumimos que dado el orden que venían los resultados era el orden de los partidos.

Luego una vez se ejecutan los métodos 0-4 (1 a 4 en la lista) devuelven un archivo de n líneas donde en la línea se obtiene el ranking del equipo i.

Para los métodos 4 y 5 se devuelve el ranking para cada iteración antes de jugar el partido, estas se repiten hasta que el que comenzó ultimo termine primero utilizando dos métodos descriptos anteriormente, en la última línea se obtiene la cantidad total de partidos, estos dos métodos además devuelven por stdout la cantidad de partidos jugados hasta que termino en la primer posición el que comenzó último.

5.2. Archivos de test usados

Dentro de la carpeta /src/testsPropios se encuentran los siguientes archivos usados en la experimentación

- ATP2007.in este archivo es usado en la experimentación de escalar posiciónes gandandole al siguiente o al primero del ranking
- ATP2007_100.in este archivo se utilizo en el analisis de salto de posiciónes en cuanto a 100 partidos jugados para obtener una cota
- carpeta random test tiene todos los archivos de test que se utilizaron para la medición de tiempos
- carpeta empate con resultados de los tests utilizados para la seccion de empates

Dentro de la carpeta /src/test se encuentran los provistos por la materia para verificar funcionalidad.

- test1.in archivo provisto por la materia
- test2.ina archivo de prueba provisto por la materia

6. Bibliografía

6.1. Bibliografía

- Numerical Analysis, Richard L. Burden & J. Douglas Faires, Chapter 6: Direct Methods for Solving Linear Systems.
- Paper The Colley Matrix Explained http://www.colleyrankings.com/matrate.pdf.
- SAP Sports Analistics http://goo.gl/Q0dnH0

7. Codigo