



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 1

Sistemas de ecuaciones lineales

métodos Numericos
Primer Cuatrimestre de 2020

Integrante	LU	Correo electrónico
Manuel Panichelli	72/18	panicmanu@gmail.com
Tomas Tropea	115/18	tomastropeaa@gmail.com
Ignacio Alonso Rehor	195/18	arehor.ignacio@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>

Índice

1. Abstract	2
2. Introducción	2
3. Desarrollo	4
3.1. Implementaciones de Métodos Numericos	4
3.1.1. Algoritmo WP	4
3.1.2. CMM	4
3.1.2.1. Características de Matriz de Colley	4
3.1.2.2. Eliminación Gausseana para CMM	5
3.1.2.3. Estabilidad numerica	6
3.2. Error del método CMM	7
3.3. Comparativa entre métodos	7
3.4. Justeza de CMM	9
3.4.1. Rating	9
3.4.2. Ranking	9
3.5. Análisis de estrategias	9
3.5.1. Simulador de partidos	10
3.5.2. Heurísticas	10
3.5.2.1. Naive	10
3.5.2.2. Hill Climber	10
3.5.2.3. min_cmm	10
4. Resultados y discusión	11
4.1. Error CMM	11
4.2. Comparativa entre métodos	12
4.3. Justeza de CMM	12
4.4. Análisis de estrategias	14
4.4.1. Tolerancia de perdida de ranking	18
5. Conclusión	20

1. Abstract

Se aborda la problemática de la confección de rankings para competencias deportivas. Se comparan los métodos CMM y WP y sus posibles problemas. Finalmente se realiza una comparación de diferentes heurísticas para tener el mayor ranking posible, con la menor cantidad de victorias.

Se concluye que el método CMM es superior a WP, que es justo, y que refleja la habilidad real de los equipos involucrados.

keywords colley matrix method, winning percentage, rankings simulados, errores de calculo

2. Introducción

Las competencias deportivas presentan una problemática compleja al momento de intentar clasificar a los distintos competidores por su desempeño. Este problema es inevitable, ya que el deporte demanda tener una forma de establecer un ranking entre los involucrados en el mismo.

Los distintos métodos que intentan abordar este problema utilizan estrategias variadas. Hay metodos que involucran el porcentaje de victorias de un equipo, o tienen en cuenta contra quienes se enfrentó el equipo para poder ponderar mas sus victorias contra buenos equipos y sus derrotas contra malos equipos. Es importante notar que, a veces son suficientes para lograr resultados reales métodos simples como el

porcentaje de victorias. Este es el caso de las competencias donde los competidores juegan todos contra todos, ya que utilizar este criterio parece ser suficiente para clasificarlos.

Estas ideas, aunque sean realistas para ciertos casos, no siempre funcionan en la practica. Esto se debe a que no capturan toda la información que hubo durante una serie de partidos, como que oponentes se enfrentaron, cuantos partidos hubo entre ellos, etc. Es valido preguntarse si realmente suceden estas cosas, pero hay ejemplos concretos como el de ATP, donde no se juegan la misma cantidad de juegos entre competidores, que presenta este problema.

Para sumarle otro problema a esto, los rankings producidos pueden, en ciertos casos, determinar entre uno y otro equipo que pase una clasificación importante. Esto resulta en que el ranking resultante no sea una cuestión menor, sino que puede determinar cuestiones económicas importantes para un equipo, así también como merito deportivo.

Debido a estas razones nos proponemos investigar y estudiar dos métodos, con el objetivo de poder determinar la eficacia de cada uno en distintos contextos, y como implementaciones diferentes de ellos pueden variar el ranking, y en que se diferencian a la hora de determinarlo.

El metodo **WP (Winning Percentage)** es una forma simple de calcular el ranking de un conjunto de equipos. Propone una formula que resulta natural para definir el rating de un equipo basado en los resultados de sus partidos. Define r_i , el rating del i -esimo equipo como:

$$r_i = \left(\frac{w_i}{w_i + l_i} \right)$$

Donde w_i son los partidos ganados y l_i son los partidos perdidos por el i -esimo equipo. Dado este metodo, podemos plantear un sistema de n ecuaciones equivalente a:

$$\begin{aligned} r_1 &= \left(\frac{w_1}{w_1 + l_1} \right) \\ r_2 &= \left(\frac{w_2}{w_2 + l_2} \right) \\ &\vdots \\ r_n &= \left(\frac{w_n}{w_n + l_n} \right) \end{aligned}$$

Notar que cada ecuación presenta una sola incógnita, r_i , ya que los partidos ganados w_i y perdidos l_i son conocidos para cada equipo. Esto implica que no es necesario plantear un sistema matricial para resolver el sistema, sino que puede resolverse cada ecuación individualmente, haciendo que este método sea muy sencillo de calcular.

Por otro lado, el **Colley Matrix Method (CMM)** es una versión mas avanzada del método WP mencionado anteriormente. Busca determinar el rating de cada uno de los equipos a partir de sus partidos, planteando un sistema de ecuaciones lineales, y resolviendo el sistema:

$$Cr = b$$

Siendo r el vector de ratings de cada equipo, b el vector de longitud igual a la cantidad de equipos, siendo el i -esimo coeficiente correspondiente al i -esimo equipo, equivalente a la expresion:

$$b_i = 1 + \frac{w_i - l_i}{2}$$

Donde w_i son los partidos ganados y l_i los perdidos por el i -esimo equipo.

Por otro lado, C es la *matriz de Colley* definida de la siguiente manera:

$$C_{ij} = \begin{cases} -n_{ij} & i \neq j \\ 2 + n_i & i = j \end{cases}$$

Es decir, el elemento c_{ij} , donde $i \neq j$, es la cantidad de partidos del equipo i contra el oponente j con signo negativo.

3. Desarrollo

3.1. Implementaciones de Métodos Numericos

Para poder evaluar los distintos aspectos de los metodos de CMM y WP, y poder llevar a cabo experimentaciones de los mismos, vamos a necesitar tener implementaciones de los mismos. Por esta razon, decidimos implementar una version de cada metodo en C++. Por un lado, tenemos una version iterativa del metodo WP, y por otro una version que resuelve el sistema planteado por CMM. Hablaremos primero entonces de estas implementaciones y los distintos detalles y consideraciones que tuvimos en cuenta para realizarlos.

3.1.1. Algoritmo WP

El metodo de WP plantea una ecuación para el rating de cada equipo, con una sola incógnita, permitiendo un algoritmo simple para resolverlas. El siguiente pseudocódigo lo ilustra:

Algorithm 1 WP

```

1: procedure WP(games: List[Game])
2:   ratings  $\leftarrow$  vector of size N
3:   for  $i = 1$  to  $N$  do
4:     wins  $\leftarrow 0$ 
5:     total  $\leftarrow 0$ 
6:     for  $j = 1$  to games.size do
7:       if games[j].team1 ==  $i$  or games[j].team2 ==  $i$  then
8:         total  $\leftarrow$  total + 1
9:         if games[i].win ==  $i$  then
10:          wins  $\leftarrow$  wins + 1
11:         end for
12:         ratings[i]  $\leftarrow$  wins/total
13:       end for
14:       return ratings
15:
```

3.1.2. CMM

3.1.2.1 Características de Matriz de Colley

Si recordamos la definición de la matriz de Colley, es evidente que la suma de los módulos de los elementos de la fila i , excluyendo la diagonal, nos da como resultado la cantidad de partidos totales que jugo el equipo i , es decir n_i que es estrictamente menor que $n_i + 2$, que es precisamente la definición del elemento en la diagonal de la fila i , es decir c_{ii} . Esto nos dice que la matriz de Colley resulta una matriz estrictamente diagonal dominante.

A partir de esta caracterización de la matriz de Colley podemos ver que se cumplen varias propiedades interesantes. La primera siendo que todas sus submatrices principales también son matrices estrictamente diagonal dominantes, lo que nos dice que estas submatrices principales, son inversibles. Entonces, como todas las submatrices principales de la matriz de Colley son inversibles, podemos garantizar que existe una factorización particular de la matriz de Colley llamada *factorización LU*.

La existencia de esta factorización para la matriz de Colley, nos indica que se puede triangular mediante el método de eliminación de Gauss, sin necesidad de hacer permutaciones entre filas. Es por esta razón que en este caso particular no es necesario hacer pivoteo parcial en la implementación de eliminación gausseana.

Una última propiedad que resulta útil de la matriz de Colley es la de su simetría. Como el elemento en la posición i, j , donde $i \neq j$, representa la cantidad de partidos del equipo i contra el oponente j , con signo negativo, entonces, el elemento en la posición j, i debería representar la cantidad de partidos del

equipo j contra el oponente i , con signo negativo, que lógicamente deben ser iguales. O lo que es igual $c_{ij} = c_{ji}$, que es la definición de una matriz simétrica.

3.1.2.2 Eliminación Gausseana para CMM

A continuación tenemos el algoritmo que hace la eliminación gausseana sobre el sistema $Cr = b$, siendo C y b la matriz y el vector generados por el método de Colley:

Algorithm 2 Gaussian Elimination

```

1: procedure GAUSSIAN ELIMINATION(system)
2:    $A \leftarrow \text{system}.A$ 
3:    $b \leftarrow \text{system}.b$ 
4:   for  $i = 0$  to  $\text{matrix.rows}-1$  do
5:     for  $j = i+1$  to  $\text{matrix.rows}-1$  do
6:        $\text{coeff} \leftarrow A.\text{cells}[i][j]/A.\text{cells}[i][i]$ 
7:       for  $c = 0$  to  $\text{matrix.cols}-1$  do
8:          $A.\text{cells}[j][c] \leftarrow A.\text{cells}[j][c] - A.\text{cells}[i][c]*\text{coeff}$ 
9:       end for
10:       $b[j] \leftarrow b[j] - b[i]*\text{coeff}$ 
11:    end for
12:  end for
13:  return system
14: end procedure

```

Tenemos inicialmente el sistema a resolver, con la matriz A y el vector b . Luego, iteramos cada fila de la matriz, siendo m el numero de la fila. En cada paso, tomamos la fila numero m en la que estamos parados y le restamos a cada una de las filas por abajo de esta un multiplo k de ella. En la linea 6 se puede ver como se calcula este coeficiente k . Si tenemos la fila numero n , y queremos restarle un multiplo de la fila m , para poner un 0 en la posicion n de la fila m , tenemos que calcular como muestra la linea 6:

$$k = a_{nm} - a_{mm}$$

Es decir, el coeficiente en la posición m de la fila n a la que queremos restarle, dividido el coeficiente en la diagonal que se corresponde a la fila n que vamos a usar para restar. Luego tomamos este coeficiente y hacemos lo que muestran la linea 7 y 8:

$$f_n = f_n - k * f_m$$

Esto pone efectivamente un 0 en la posicion m de la fila n , ya que la resta queda:

$$\begin{aligned}
 a_{nm} &= a_{nm} - a_{nm} \cdot \left(\frac{a_{nm}}{a_{mm}} \right) \\
 &= a_{nm} - a_{nm} \cdot \left(\frac{a_{nm}}{a_{mm}} \right) \\
 &= a_{nm} - a_{nm} \cdot 1 = 0
 \end{aligned}$$

también se le resta al coeficiente b que se corresponde con la fila n con el mismo calculo, como se ve en la linea 10:

$$b_n = b_n - k * b_m$$

Por la forma en que se recorre la matriz, este algoritmo genera una matriz triangular superior. Resta aplicar el metodo de backward substitution, como se muestra a continuacion:

Algorithm 3 Backward substitution

```
1: procedure BACKWARD SUBSTITUTION(system)
2:    $A \leftarrow \text{system}.A$ 
3:    $b \leftarrow \text{system}.b$ 
4:    $x \leftarrow \text{vector of size } A.\text{rows}$ 
5:   for  $r = A.\text{rows}-1$  to  $0$  do
6:      $sum \leftarrow 0$ 
7:     for  $c = r+1$  to  $A.\text{cols}-1$  do
8:        $sum \leftarrow sum + x[c] * A.\text{cells}[r][c]$ 
9:     end for
10:     $x[r] \leftarrow (b[r] - sum) / A.\text{cells}[r][r]$ 
11:  end for
12:  return  $x$ 
13: end procedure
```

En este paso lo que hacemos es empezar desde la ultima fila y subir hasta la primera, ya que la matriz es triangular superior. En cada paso utilizamos los ratings que ya despejamos de resolver ecuaciones anteriores, y hacemos la sumatoria de:

$$suma = \sum_{c=r+1}^n a_{rc} * x_c$$

Donde c empieza en $r+1$ y va hasta columnas de A menos 1. Esto produce una sumatoria sin el elemento de la diagonal, porque es el que queremos averiguar. Finalmente tomamos esta sumatoria y despejamos el rating de la diagonal de esa fila, de la siguiente manera, como muestra la línea 10:

$$x_r = \frac{(b_r - suma)}{a_{rr}}$$

3.1.2.3 Estabilidad numerica

Al realizar esta implementación, nos preguntamos los posibles problemas que podia tener la eliminación gausseana en terminos de estabilidad numerica por el uso de aritmetica finita.

A diferencia de los numeros con los que trabajamos normalmente, que pueden tener infinitas cifras, en la computadora se pueden representar con finitas cifras, lo cual genera perdida de precisión. Esta perdida de precisión puede ser minima o no, dependiendo de que paso el los calculos intermedios para encontrar el resultado final. Podemos listar los casos que generan problemas dependiendo de la operación que se realiza:

- Dividir por números pequeños.
- Multiplicar por numeros grandes.
- Restar numeros similares (Cancelacion catastrofica)
- Sumar numeros de magnitudes distintas.

En los primeros dos casos, se genera un error al momento del redondeo proporcional a los numeros que se utilizaron. En el tercer caso se produce una perdida de los digitos significativa y en el ultimo caso, el numero de menor magnitud puede llegar a no afectar a la suma.

Por estas razones es que, al momento de realizar la eliminacion gausseana, se debe tener cuidado con el aspecto de la division por numeros pequeños. Es por esta misma razon que se introduce la tecnica de pivoteo, donde se utiliza la fila que tenga el coeficiente mas grande en cada paso para dividir.

Debido a las características de la matriz de Colley, esto no es necesario. Por ser diagonal dominante, el elemento de la diagonal es el mas grande en valor absoluto de la fila, y por ser simetrica, también

de la columna. Como esta propiedad se mantiene luego de cada paso de eliminación gaussiana para la submatriz que falta triangular, garantiza que el pivoteo parcial nunca hará una permutación. Entonces obtendremos una buena estabilidad numérica sin necesidad de agregar pivoteo al algoritmo descripto anteriormente.

3.2. Error del método CMM

Hemos mencionado anteriormente como, al realizar el método de eliminación de Gauss, pueden generarse errores debido al uso de aritmética finita. Por esta razón, quisimos realizar un experimento analizando el grado de error que podía presentar nuestra implementación. Así mismo, quisimos incluir en este análisis a los resultados brindados por la cátedra. Dado que no estamos comparando contra los resultados provistos por la cátedra, el lector podría preguntarse entonces, contra qué resultados es válida la comparación, pues, después de todo, estos también deberían estar sujetos a algún grado de error. La solución que se nos ocurrió fue, dada una muestra, armar el sistema asociado a la matriz de Colley, y resolverlo usando la biblioteca `numpy.linalg`¹. Por este motivo, vale aclarar que este experimento y sus resultados, no son totalmente determinantes, si no, más bien, sirven para darle al lector una idea del grado de error que admiten ambas implementaciones. Con estos supuestos asumidos, la próxima pregunta que podría realizarse el lector es, cómo medir el grado de error que presenta una implementación contra nuestro sujeto de comparación, pues, después de todo, esta comparación debe hacerse entre dos vectores. Lo habitual en estos casos es calcular la distancia vectorial entre estos dos resultados, donde, dados v, w en un espacio vectorial, y $\|\cdot\|$ una norma del mismo espacio vectorial, se define:

$$d(v, w) = \|v - w\|$$

Lo interesante de este enfoque es que nos permite analizar cómo afecta, la elección de distintas normas vectoriales, al resultado final. Para evaluar este punto, decidimos calcular las distancias considerando tres normas vectoriales usuales: la norma euclídea, la norma-1, y por último la norma infinito.

3.3. Comparativa entre métodos

Dependiendo de la competencia con la que se trate, los datos de partidos pueden variar significativamente. El schedule puede ser tal que todos los equipos se enfrenten contra todos equitativamente, como es el caso de la NBA. Pero también puede pasar que hayan equipos que nunca se enfrenten, como es el caso de ATP. Pensamos que esto último puede generar problemas al momento de decidir el ranking, ya que es difícil considerar el factor de que los partidos no estén repartidos equitativamente entre todos los equipos. Por estas razones haremos un análisis sobre la efectividad de los métodos CMM y WP para generar el ranking de competencias con distintos schedules, utilizando datos reales de competencias. Consideraremos la distribución de los partidos para predecir el comportamiento de CMM y WP.

Creemos que con una distribución similar a round-robin de los partidos, como los de NBA, ambos métodos se comportarán de forma similar. Pensamos esto porque, al jugar todos contra todos, el porcentaje de partidos ganados parece un buen estimador del ranking. Entonces como WP representa esta fórmula y el método de CMM también utiliza esta fórmula, esperamos que presenten resultados similares entre sí y razonables.

Por otra parte, si la competencia tiene casos como los de ATP, suponemos que CMM va a lograr generar buenos resultados, pero WP no. Esto último es porque, considerar únicamente los números de partidos ganados y perdidos omite un factor importante, que es contra quien se produjo cada uno de estos resultados. Al no haber una distribución uniforme, puede suceder que un equipo tuvo un gran porcentaje de partidos ganados debido a que se enfrentó contra equipos de bajo rating. Esto resultaría en un rating alto, que hubiera sido mucho más bajo si, en su lugar, se hubiera enfrentado a buenos equipos. Creemos que este posible inconveniente de WP será compensado en CMM al tener en cuenta los ratings de los oponentes, y no permitirá ganar mucho rating si se enfrenta a malos equipos también.

Formalizaremos las nociones descritas anteriormente mediante el uso de dos métricas:

¹<https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.solve.html>

- Como lo mas importante es saber si todos jugaron contra todos, tendremos en cuenta el porcentaje de todos los equipos contra el cual jugó cada uno, el **share**.
- Sin embargo, podría suceder que no hayan jugado la misma cantidad de partidos, con lo cual WP también funcionaría mal, entonces tendremos en cuenta la **participación** de cada equipo, es decir, en cuantos juegos participaron, normalizada por la mayor cantidad de juegos jugada por algún equipo.

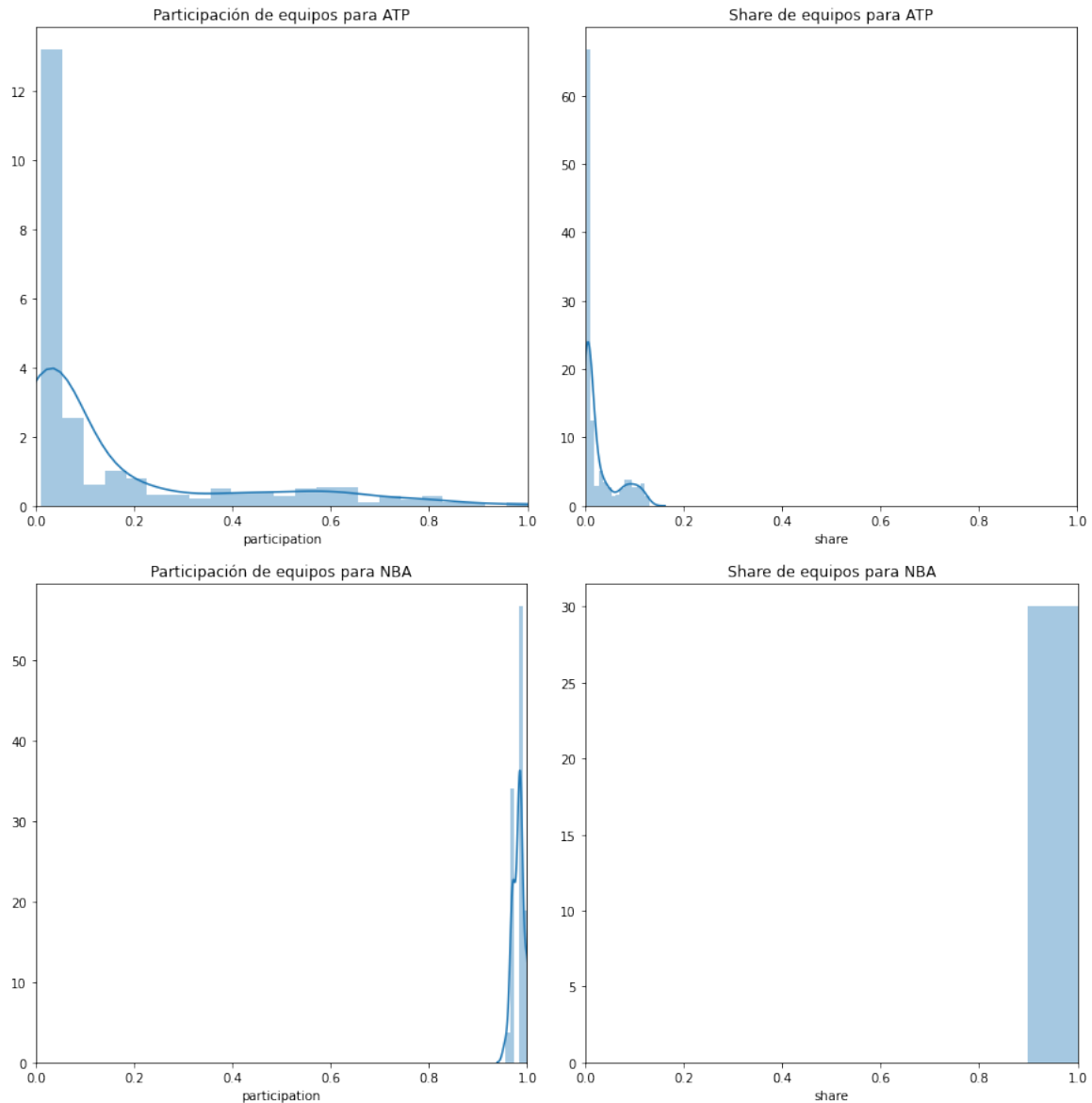


Figura 1: Participacion y share de ATP vs NBA

Como se puede ver en la figura 1,

- NBA sería ideal para WP, ya que al ser round robin, todos jugaron contra todos (el share es 1), y casi todos jugaron la misma cantidad de juegos (la participación de casi todos es 1). Debería dar muy similar los ratings resultantes entre WP y CMM.
- ATP es lo contrario, no juegan todos contra todos, y tampoco juegan la misma cantidad de juegos. Debería dar diferente entre WP y CMM.

3.4. Justeza de CMM

Nos interesa estudiar si CMM es un método *justo*, es decir, si los partidos entre dos jugadores podrían afectar a un tercero que no tiene relación con ellos. Además, la *afectación* se puede ver con respecto a dos aspectos: el **rating** (numero que el método le asigna a cada equipo) y **ranking** (ordenando todos los equipos por su rating, en que posición queda).

3.4.1. Rating

Suponiendo que juegan los equipos d con e , ¿Podría el resultado de su partido afectar a un tercero, a ? Nuestra hipótesis es que depende de cual sea la definición de *tercero* empleada. Por como se conforma la matriz de colley, el rating de un equipo depende del rating de sus oponentes. Entonces, si a no jugó contra d ni e , en principio su rating no debería verse afectado, pero si había jugado por ejemplo contra b , que **sí** había jugado contra c y este contra d , el rating de c y d se vería afectado, y luego el de a también.

Podemos ver los dos casos posibles con el siguiente esquema, que representa a cada equipo como un nodo en un grafo, y existe una conexión entre ellos si jugaron al menos un partido:



Figura 2: Posibles relaciones entre equipos

- En 2a, podemos ver que, A no jugó contra D y E, pero sin embargo existe un camino que une a A con D y E, debido a que hay una cadena de relaciones entre equipos que jugaron entre sí. En este caso, el rating se vería afectado por este juego.
- En cambio, en 2b, el equipo A no tiene relación, ni siquiera indirecta con ellos, ya que no existe un camino que los conecte. Luego, su rating no se verá afectado.

3.4.2. Ranking

El ranking **siempre** debería poder cambiar, ya que aunque el rating no se vea afectado, siempre que se vea afectado el de otra persona, podría cambiar el orden de la tabla de posiciones, lo cual cambiaría su ranking.

3.5. Análisis de estrategias

Una pregunta que uno podría hacerse mientras analiza el método CMM, es si existe una estrategia que permita obtener el *mayor ranking posible*, con la *menor* cantidad de victorias. Para poder determinar estas estrategias, tomamos los resultados de una competencia, elegimos un equipo, y aplicamos una de varias heurísticas para así minimizar la cantidad de victorias. Y para hacerlo lo más realista posible, establecemos tres restricciones sobre qué es posible modificar

- Solo se pueden cambiar los resultados de los partidos en los que participó el equipo seleccionado.
- Solo se pueden *perder* partidos que previamente habían sido ganados, y no ganar partidos que habían sido perdidos. Esto es porque nos parece razonable la noción de que uno podría decidir hacer *forfeit* de un juego, pero nunca podría asegurar una victoria.
- El jugador seleccionado conoce las habilidades reales del resto.

3.5.1. Simulador de partidos

Para poder probar estas heurísticas o estrategias en un ambiente lo más controlado posible, haremos uso de un *simulador de partidos* en vez de usar datos reales. Este nos permite, a grandes rasgos, controlar la **habilidad** de cada equipo y como se distribuye, y el **schedule** que siguen los partidos. Luego, para cada partido entre dos equipos, la probabilidad de ganar de cada uno es proporcional a su habilidad.

- La habilidad será distribuida de forma *lineal*, es decir, sea N el máximo poder y T la cantidad de equipos, el poder p_i de cada equipo queda determinado por la siguiente ecuación

$$p_i = i * \frac{N}{T}$$

- El *schedule* a seguir será *round robin*. Es decir, juegan todos contra todos.

3.5.2. Heurísticas

A continuación describimos las diferentes heurísticas que proponemos para lograr el objetivo propuesto: maximizar el ranking, minimizando la cantidad de victorias.

3.5.2.1 Naive

Las heurísticas **lose_better** y **lose_worse** son muy simples, perder todos los partidos contra los que tengan mayor y menor habilidad, respectivamente. Dada su simpleza, esperamos que sean las peores.

Dado el funcionamiento del método CMM, pesan más los partidos ganados contra mejores oponentes. De esto se deriva la última estrategia *naive*, **win_one**. Esta se encarga de perder *todos* los partidos menos uno, el que ganó contra el equipo con mayor rating.

3.5.2.2 Hill Climber

La heurística **hill climber** toma el estado actual de los datos de la competencia, selecciona al azar un juego en el que esté involucrado y haya ganado, lo modifica para perder y recalcula el ranking de colley. Con esto obtiene su nuevo rating, y toma una decisión dependiendo si es mejor o peor que el anterior. Si es igual o mejor, acepta el nuevo estado ya que minimizo sus ganadas sin empeorar su ranking.

Este proceso se repite durante varias iteraciones con el objetivo de poder minimizar lo mas posible sus ganadas. Esta forma de estrategia es también *naive* en cierto sentido, ya que selecciona juegos al azar, y acepta un nuevo estado solo porque se acerco un poco a su objetivo.

Tiene el problema de que puede quedarse atrapado en un mínimo local, cuando tal vez era mejor tomar otra decisión la cual a la larga lograría minimizar aun mas las partidas ganadas.

3.5.2.3 min_cmm

Quisiéramos alguna manera de perder la mayor cantidad de juegos posibles sin perjudicar nuestra posición en el ranking. De esta manera, la estrategia se reduce a averiguar qué oponente es más conveniente para cederle una victoria. Luego, la estrategia consiste en perder contra estos oponentes hasta que otra derrota más implique un descenso en la tabla de posiciones. Si analizamos definición del 'rating' de un jugador en base al sistema asociado a la matriz de Colley, podremos observar que este está determinado, más allá de las victorias y derrotas del propio jugador, por los ratings de sus oponentes, y la cantidad de juegos que el jugador tiene en común.

$$r_i = \frac{1}{2 + n_i} \left(1 + \frac{n_w - n_l}{2} + r_1 n_{i1} + \dots + r_n n_{in} \right)$$

Lo que esto significa es que, en un formato donde juegan todos contra todos, en términos de diferencias absolutas del rating de un equipo, nos es indiferente contra quién perdemos. Perder contra cualquier oponente nos significa la misma caída del rating que contra cualquier otro, y además esta derrota significa el mismo aumento del rating para todos los otros equipos. Con esto, podemos responder cuál es el

oponente más conveniente para cederle una victoria: el último en la tabla de posiciones. La justificación es bastante simple, queremos que el oponente que vaya a incrementar su número de victorias (n_w), sea el que está a mayor distancia de alcanzar nuestra posición. Lo interesante de esta estrategia es que, al ceder una victoria, el estado inicial de la competencia cambia, y no necesariamente se mantiene quién está último en la tabla de posiciones. Este concepto es parte fundamental de esta heurística y nos permite minimizar la cantidad de victorias necesarias para mantener el ranking.

En cada iteración, la heurística,

1. Calcula los ratings actuales.
2. Modifica los juegos para perder contra aquel oponente que está último en la tabla de los ratings.
3. Calcula los ratings nuevamente
4. Si su posición en el ranking bajó, entonces volvemos uno atrás y finalizamos. Sino, repetimos el proceso hasta que suceda.

4. Resultados y discusión

4.1. Error CMM

Estos son los resultados de medir el error entre nuestra implementación del **Colley Matrix Method**, y nuestro sujeto de comparación, la solución del sistema asociado a la matriz de Colley, computado por la biblioteca `numpy.linalg`. También medimos el grado de error incluido en los ratings provistos por la cátedra.

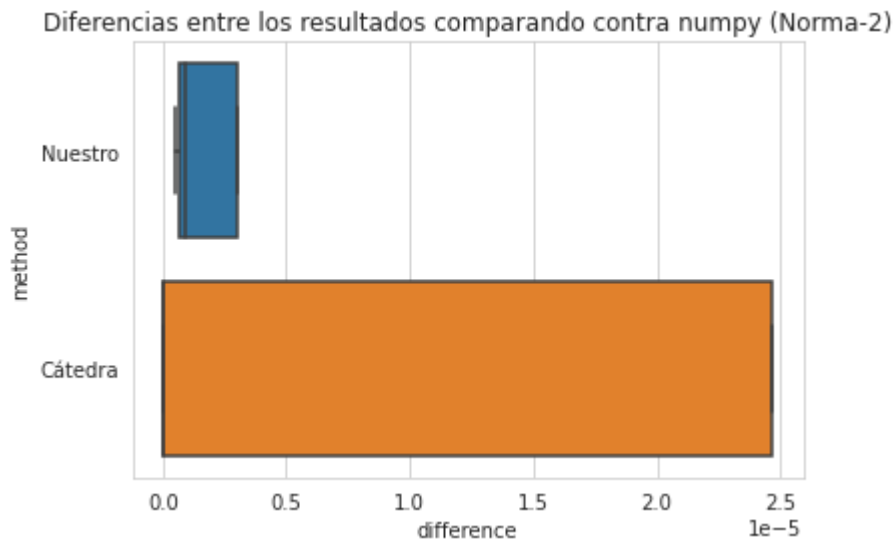
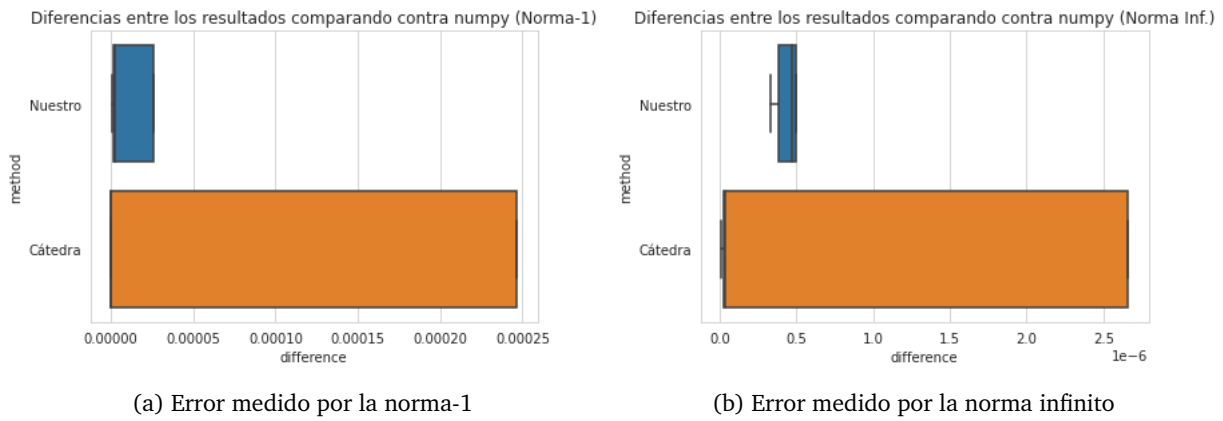


Figura 3: Error medido por la norma euclídea



Como se puede ver en la figura 3, en todos los casos, utilizando todas las normas vectoriales, nuestros resultados resultan más precisos que los provistos por la cátedra. Lo que resulta interesante es que, dentro de todas las normas utilizadas, la que menor varianza tiene es la norma infinito,. Esto tiene sentido dado que si muchos de los componentes tienen algún error proveniente de usar aritmética finita, estos no se suman entre si, cosa que sucede al calcular la distancia utilizando otras normas vectoriales.

4.2. Comparativa entre métodos

Los resultados de correr los rankings son los siguientes

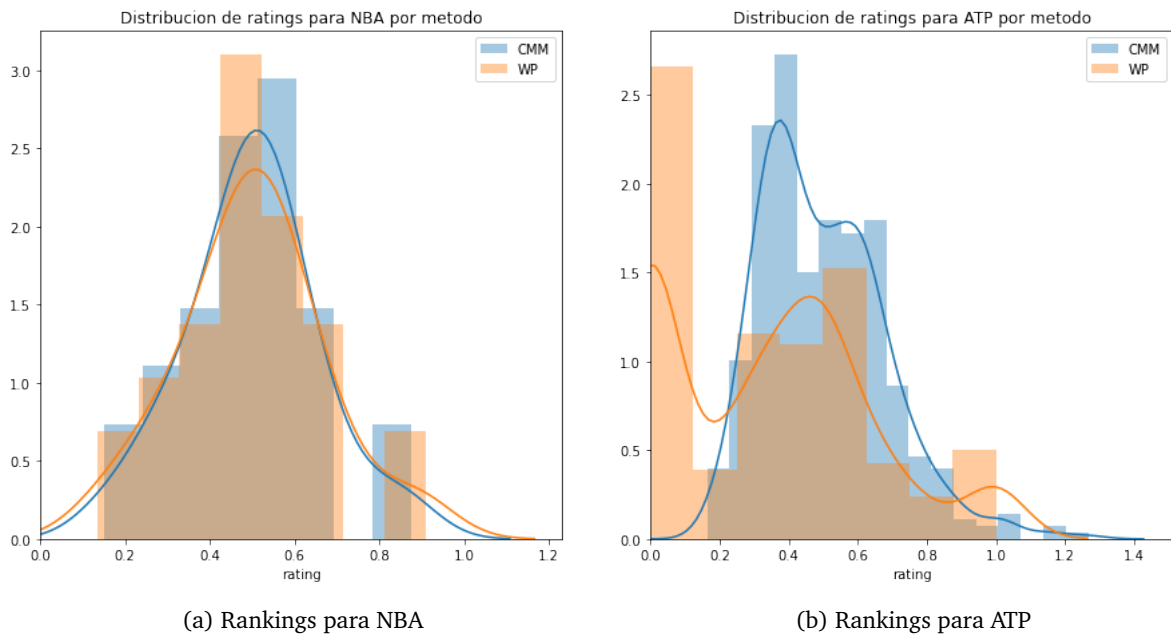


Figura 5: Comparativas de rankings

Como se puede ver en la figura 5a, el ranking coincide para NBA tanto para WP como CMM, pero este no es el caso de ATP como ilustra 5b, donde varían significativamente los rankings resultantes de ambos métodos. Además, como en ATP muchos no jugaron tantos partidos (ya que su participación era baja), su ranking oscila al rededor del inicial de cada método, 0 para WP y 0.5 para CMM.

4.3. Justeza de CMM

El experimento que hicimos fue tomar los datos utilizados en Govan et al², que daban como ranking inicial,

²<http://carlmeyer.com/pdfFiles/SASGF08RankingPaper.pdf>

						rating	team
6	10						
1	1	16	4	13		0.668661	3
1	2	38	5	17		0.615978	2
1	2	28	6	23			
1	3	34	1	21		0.539174	6
1	3	23	4	10			
1	4	31	1	6		0.501544	5
1	5	33	6	25			
1	5	38	4	23		0.359707	1
1	6	27	2	6			
1	6	20	5	12		0.314936	4

Figura 6: Data y rankings resultantes con los datos originales de Govan

Y modificarlos para que tengan dos nuevos equipos que hayan jugado entre sí pero contra nadie más, el 7 y 8 (el ultimo juego)

					rating	team
8	11					
1	1	16	4	13	0.668661	3
1	2	38	5	17	0.625000	7
1	2	28	6	23		
1	3	34	1	21	0.615978	2
1	3	23	4	10	0.539174	6
1	4	31	1	6		
1	5	33	6	25	0.501544	5
1	5	38	4	23	0.375000	8
1	6	27	2	6		
1	6	20	5	12	0.359707	1
1	7	20	8	10	0.314936	4

Figura 7: Data y rankings resultantes con los datos modificados de Govan

Como se puede ver en la figura 7, los nuevos equipos, 7 y 8, tienen ranking 2 y 6 respectivamente. Para ver si estos cambian después de juegos de terceros, agregamos a los datos partidos entre 5 y 6, que no están relacionados ni indirectamente con 7 y 8.

[illegible]

Figura 8: Data y rankings resultantes con los datos nuevos de Govan

Como se puede ver en la figura 8, a pesar de que los ratings de 7 y 8 permanecen exactamente iguales, el ranking de 7 se ve afectado, pasando de 2do a 3ro, ya que cambiaron los ratings del resto de los equipos, lo cual confirma nuestras hipótesis.

4.4. Análisis de estrategias

La simulación a realizar tiene 10 equipos, y el elegido para aplicar las heurísticas es el numero 8, (el tercero mejor) ya que tendrá muchas partidas ganadas en las cuales hacer *forfeit*. Para comparar las heurísticas entre sí, tendremos en cuenta dos métricas: *ranking resultante* y *wins*. De esta forma, podremos observar cual es la que minimiza las wins maximizando el ranking.

Comenzamos por determinar una *seed*, para que luego la simulación con cada heurística sea reproducible y obtengamos los mismos resultados. Primero lo simulamos sin aplicar ninguna heurística, y luego sobre el conjunto de partidos resultantes aplicamos cada una. Esto lo iteramos 50 veces, y los resultados son los siguientes:

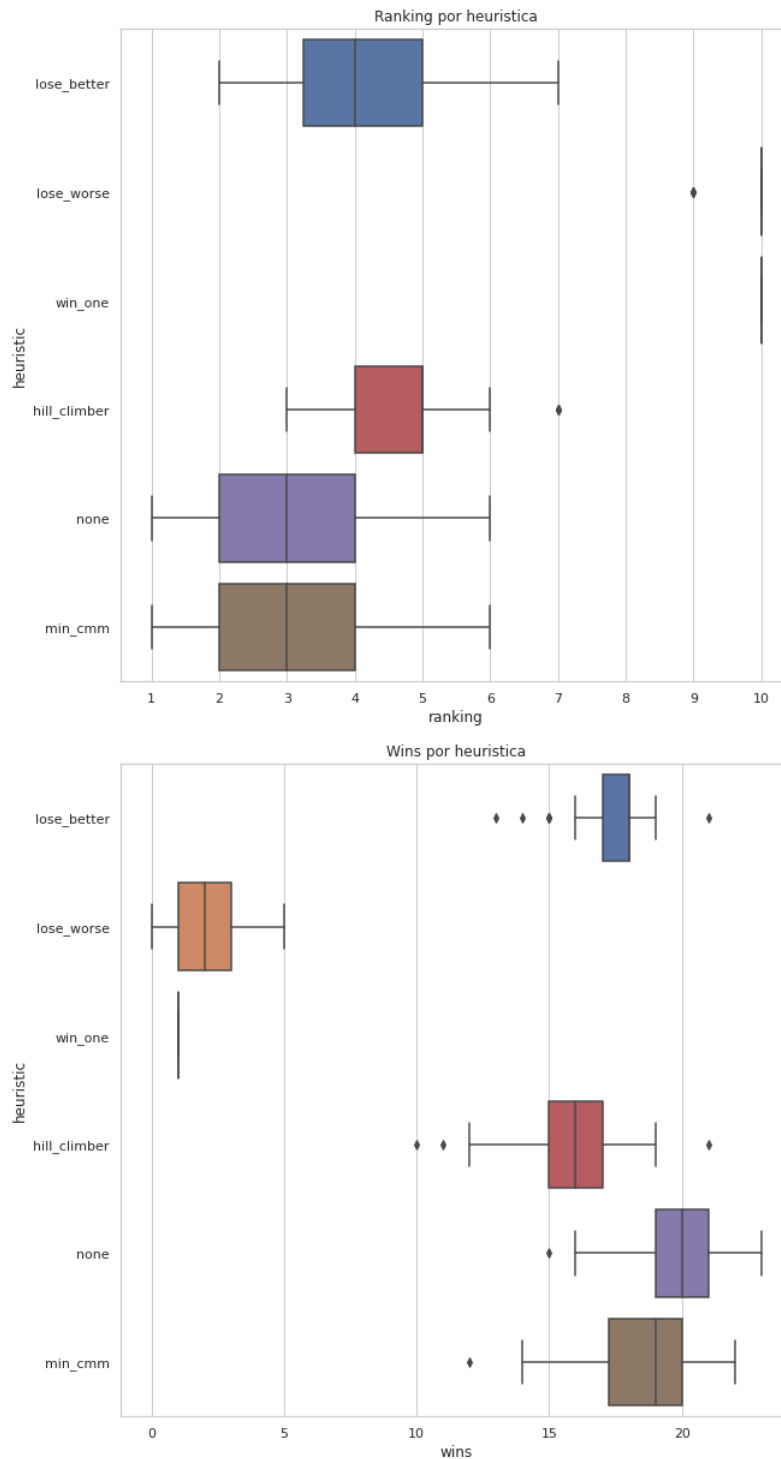


Figura 9: Ranking y wins obtenido por cada heurística luego de haber modificado los juegos

Como se puede ver en la figura 9,

- **none** es el resultado de no aplicar ninguna heurística, un resultado de *control* para comparar con el resto.
- **lose_worse** es el peor, ya que al ser el equipo numero 8, como las habilidades son lineales, casi todos son peores que él, entonces pierde casi todos los juegos, lo cual afecta demasiado a su ranking previo a los cambios.
- **win_one** muestra que al parecer, no basta con ganar solo una para tener un ranking competente,

ya que sale consistentemente último.

- **lose_better** resultó ser eficaz para ser tan simple. Las wins son consistentemente menores que sin heurísticas, pero el ranking disminuyó bastante.
- **hill_climber** es parecido al anterior, con la diferencia de que *siempre* pierde ranking, pero es el que menos victorias tiene hasta el momento.
- **min_cmm** es el mejor y se comporta como era esperado. El ranking es siempre idéntico, y las wins son consistentemente menores, tal como buscábamos.

Luego de haber concluido que **min_cmm** era el mejor, decidimos buscar otra forma de visualizar el comportamiento del mismo. Para ello, consideramos la evolución del rating de cada equipo fecha por fecha.

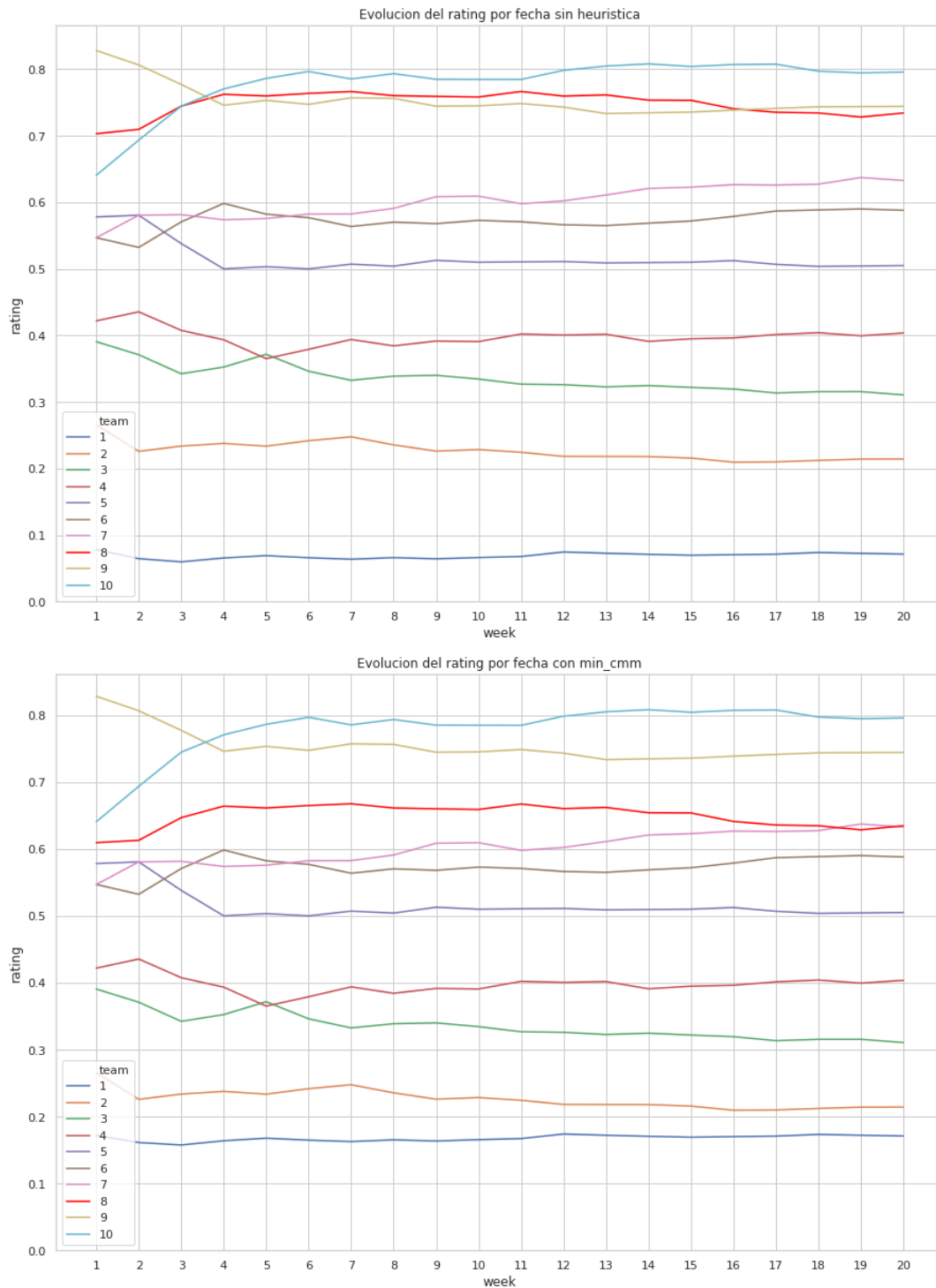


Figura 10: Rating semana a semana sin heurística y aplicando min cmm

Como se puede ver en la figura 10, el equipo elegido (el 8), se deshace de la diferencia de rating innecesaria entre el y el anterior (el 7), hasta *justo* antes de perder un rank. Además, los ratings del último equipo suben, ya que ahora le gana al equipo elegido muchos mas partidos.

Además, el rating de cada equipo se ve correlacionado con su habilidad real. Lo que ilustra que el método CMM produce resultados que reflejan la habilidad real de los equipos.

4.4.1. Tolerancia de pérdida de ranking

Hay casos en donde no es de interés mantener el ranking exacto, como por ejemplo en las selecciones de un mundial de fútbol, donde lo importante es quedar entre los primeros. En esos casos, se podría *tolerar* perder algunos ranks, y así se tener aún menos victorias.

Para modelar esto, introducimos una nueva clase de heurísticas: *min cmm con tolerancia*. Estas *tolera* perder N ranks, en pos de seguir minimizando sus victorias. Con esta consideración, la implementación inicial de min cmm tendría *tolerancia 0*, por ejemplo.

Esto produce que pierda muchos mas partidos, pero sin embargo no llegaba a bajar todos los ranks que le permitía su tolerancia. Esto se debe a que se queda sin partidos que perder contra el último, porque se rinde en todos los posibles. Por esta razón, le agregamos la capacidad de, en el caso de no poder perder mas contra el ultimo, empezar a perder contra el ante último, luego el antepenúltimo, y así. Los resultados fueron los siguientes

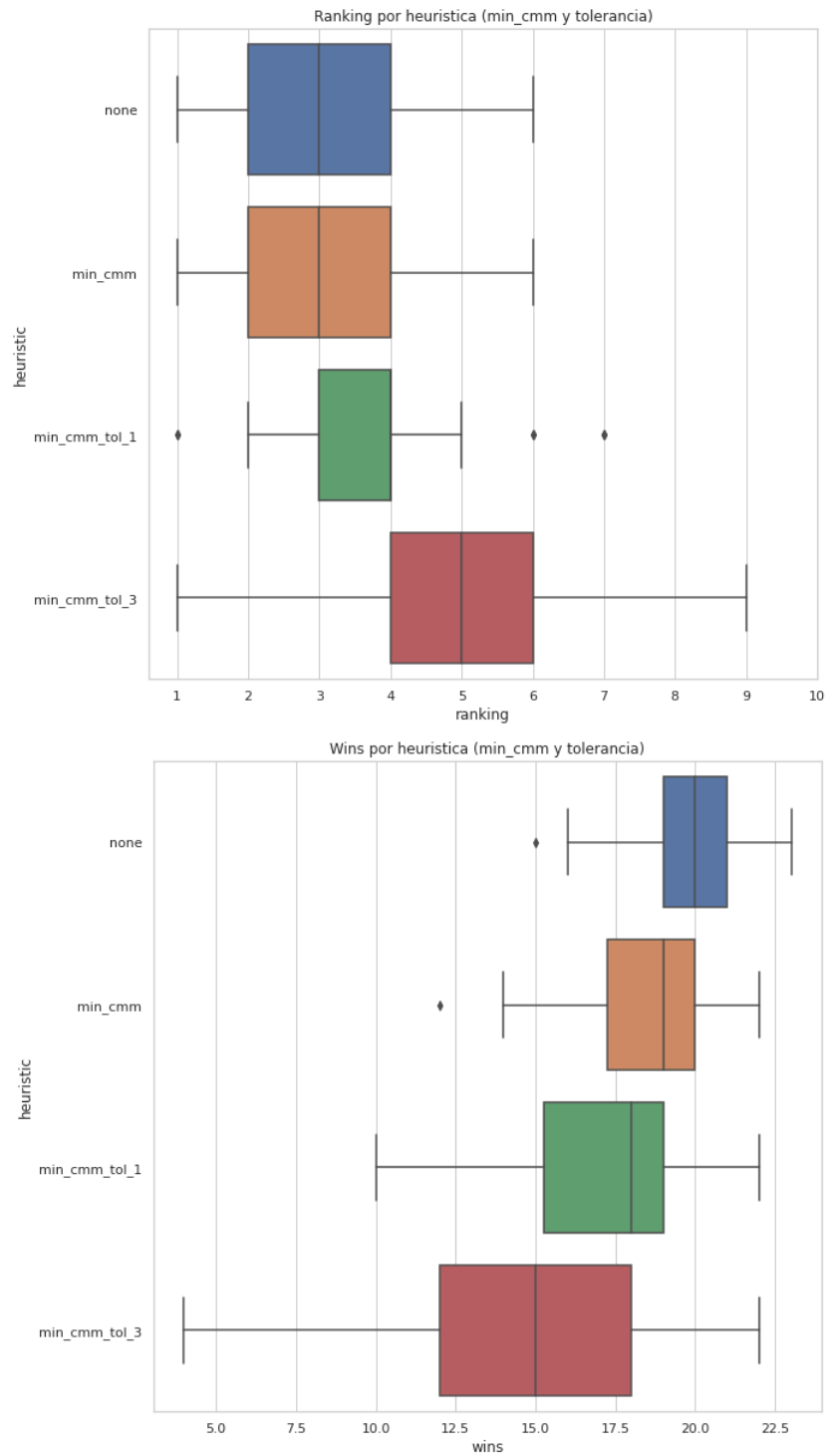


Figura 11: Ranking y wins obtenido por cada heurística de la clase min cmm con tolerancia. Donde *min_cmm_tol_i* tolera perder *i* ranks

En la figura 11 se puede ver que estas nuevas heurísticas se comportan como es esperado, perdiendo la cantidad de ranks que toleran y en consecuencia, las wins bajan sustancialmente. También podemos ver su comportamiento analizando fecha a fecha

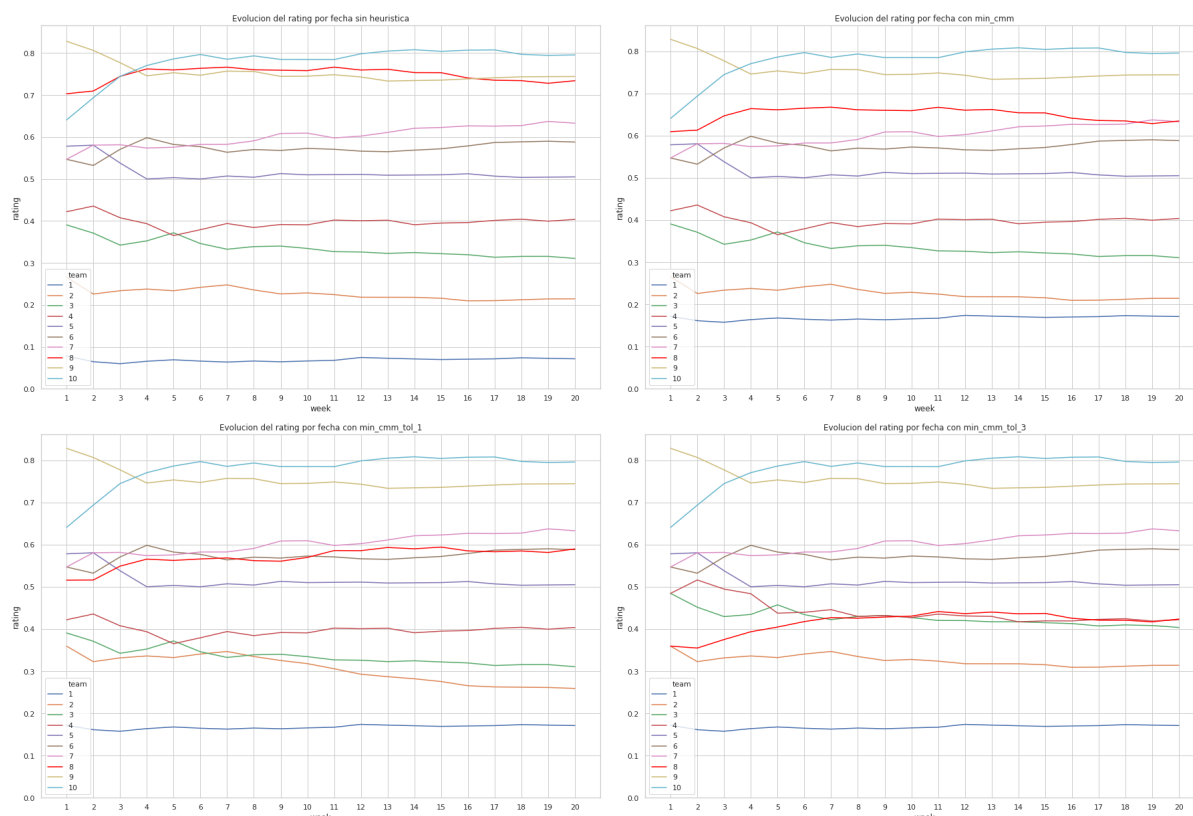


Figura 12: Rating fecha a fecha sin heurística y aplicando min cmm con niveles variables de tolerancia

En la figura 12 Se puede ver como, claramente, al final del torneo el equipo elegido se queda con el rating que le permita mantener *justo* la pérdida de ranking tolerado. También es posible notar que, como pierde contra más jugadores, una mayor proporción de los ratings se ve afectada, ya que mas equipos ganan partidos además del ultimo. Esto ultimo es interesante porque, aunque muchos ratings de equipos se vean afectados, sus rankings finales no se ven afectados, demostrando como CMM mantiene una relación entre los equipos y sus ratings.

5. Conclusión

Como también concluye Colley en su paper original³, WP no es un buen método para confeccionar rankings, dado su mal funcionamiento para ciertos formatos de competencias, i.e diferentes a *round-robin*. Sin embargo, CMM parece producir rankings que reflejan de cierta forma la realidad, ya que para los jugadores simulados, su rating estaba relacionado a su habilidad real.

Al margen de la correctitud del método CMM, consideramos que es un método *justo*. Ya que los ratings de un equipo son afectados por los juegos de otros solo cuando estaban de cierta forma relacionados entre sí.

En cuanto a las heurísticas, si bien concluimos que la mejor es *min_cmm*, esto es solo en el caso en el que formato de la competencia sea *round-robin*. Sino, podría mejorarse la elección del oponente, ya que se debe tener en cuenta la cantidad de partidos que este tuvo contra el resto, y si es baja, tal vez el impacto en el rating de los otros jugadores no sea tanto.

Además, este y el resto de las estrategias empleadas buscan, dado un rank, mantenerlo (o incluso, empeorarlo) minimizando las victorias necesarias. Pero lo que **no** buscan, y podría ser un punto interesante para futuras investigaciones, es mejorar ese ranking inicial. Para ello, tal vez habría que cambiar el enfoque utilizado, y en vez de modificar una lista de partidos ya dada, tomar decisiones sobre la marcha, basándose en la información disponible en el momento.

³<https://www.colleyrankings.com/matrate.pdf>