

Teoría de Algoritmos II

Parcialito 3

Autor: Alejo Tomás Mariño
Padrón: 105645

Consigna:

1. Se quiere convocar a una elección a la que se presentan 4 candidatos (A, B, C y D). Hay 3 votantes del jurado que tienen sus siguientes rankings individuales:

- Jurado 1: $B \succ C \succ D \succ A$
- Jurado 2: $C \succ D \succ A \succ B$
- Jurado 3: $D \succ A \succ B \succ C$

- ¿Quién ganaría por eliminación iterativa?
- ¿Quién ganaría por Borda rule?
- Suponé que estás a cargo de definir las reglas/formato de la votación, y sos un miembro corrupto que desea que si o si gane la alternativa A (te asegura favores si logra ganar la elección). Definir (si existe) un sistema de votación en el cual A resulte ganador de la elección. En caso de no existir, explicar por qué. ¿Cuál propiedad deseable de los sistemas de votación no se está cumpliendo si, efectivamente, ganara A?

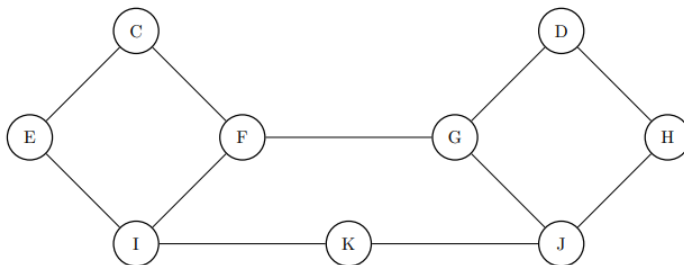
[1 Punto]

2. Considerando el modelo de cascadas de información visto en clase, supongamos que hay una nueva tecnología que los individuos pueden optar por aceptar o rechazar. Supongamos que cada uno que acepta la tecnología recibe una ganancia positiva o negativa (sin conocerla a priori). Estos valores son aleatorios para cada nodo, y si la tecnología es “Buena”, entonces el promedio será positivo, y si la tecnología es “Mala” el promedio será negativo (esta información es conocida por los individuos). Quienes rechacen la tecnología reciben ganancia 0. En este modelo, cuando a un individuo le toca elegir si acepta o rechaza la nueva tecnología, recibe la información de las ganancias de todos los que vinieron antes.

- Supongamos que esta nueva tecnología es, en realidad, “Mala”. ¿Cómo afecta esta nueva información (qué ganancia tuvo cada uno de los que vinieron antes) a la potencial formación de una cascada para que persista la nueva tecnología? (No es necesario dar una demostración, simplemente argumentar)
- Supongamos que esta nueva tecnología es, en realidad, “Buena”. ¿Puede surgir una cascada de rechazo de esta nueva tecnología?

[2 Puntos]

3. Considerá la siguiente red, suponiendo que todos los nodos tienen inicialmente un comportamiento B. Cada nodo puede cambiar al comportamiento A si al menos la mitad de sus vecinos tiene dicho comportamiento.



- Supongamos que los nodos E y F son *early adopters* del comportamiento A. Si los demás nodos siguen la regla del umbral (threshold) para adherir a este nuevo comportamiento, ¿qué nodos implementarían el comportamiento A?
- Explicar a qué se debe que el comportamiento A no se propaga a través de toda la red en el escenario del punto (a). ¿Qué característica de la red lo impide? (responder a esta pregunta no apuntando a nodos particulares sino a presencias de ciertas características) ¿Dónde más tendría que haber otro *early adopter* de A sí o sí para que el comportamiento se propague a través de toda la red?

[2 Puntos]

4. Tenemos dos grafos no dirigidos G_1 y G_2 , con la misma cantidad de vértices y aristas. G_1 es un grafo aleatorio de Erdős-Rényi, mientras G_2 es un grafo que cumple la ley de potencias en la distribución de los grados. Consideremos un virus que comienza en un único vértice aleatorio y se expande según el modelo **SIR**.

- ¿En cuál grafo es más probable que ocurra una epidemia (i.e. se infecte al menos un 30% de la red)? Justificar brevemente la respuesta.
- Supongamos que en vez de comenzar en un vértice aleatorio, la epidemia comenzara en el vértice de mayor grado de G_1 y G_2 , respectivamente. ¿En cuál de los grafos es más probable que ocurra una epidemia? Justificar brevemente la respuesta.
- ¿Cómo afecta la existencia (o no existencia) de comunidades en la expansión de la epidemia?

Para responder estas preguntas, se les recomienda realizar simulaciones. Pueden agregar todo tipo de resultados obtenidos para justificar sus respuestas.

[3 Puntos]

- Aplicar el Algoritmo REV2 al siguiente set de datos de reviews de productos de Amazon, para detectar potenciales usuarios maliciosos y otros ciertamente honestos. Por simplificación (y unificación de criterios), considerar $\gamma_1 = \gamma_2 = 0.5$. Obtener aquellos usuarios cuya *justicia* (*fairness*) es menor o igual a 0.2 (son maliciosos) y tienen al menos 5 reviews, así como la proporción de nodos que son extremadamente justos: aquellos con *justicia* mayor o igual a 0.9, y con al menos 10 reviews (aristas de salida).

[2 Puntos]

1. a. Por eliminación iterativa/sucesiva (en orden A->B->C->D):

Jurado 1: B > C > D > A | A vs B = 2 vs 1 = gana A

Jurado 2: C > D > A > B | A vs C = 1 vs 2 = gana C

Jurado 3: D > A > B > C | C vs D = 2 vs 1 = gana C

.. **gana C.** (si se eligen ciertos órdenes en particular se puede hacer que gane D

[B->C->A->D] o A [D->C->B->A] o B [C->D->A->B])

b. Por borda rule:

Jurado 1: B > C > D > A | A = 0, B = 3, C = 2, D = 1

Jurado 2: C > D > A > B | A = 1, B = 3, C = 5, D = 3

Jurado 3: D > A > B > C | A = 3, B = 4, C = 5, D = 6

.. **gana D.**

c. No se me ocurre (por lo menos con los sistemas de votación vistos en clase) una manera de hacer ganar a A con las decisiones de los jurados actuales, de todas maneras propongo un pequeño cambio para forzar un poco que gane A.

Propongo cambio en Jurado 3: de D > A > B > C a A > D > B > C.

De esta manera se puede ver que A gana por eliminación iterativa en orden

C->D->B->A, por lo que habría que mandarle un paquete de chocolates y flores de parte de A al jurado 3.

(Después me di cuenta que con eliminación iterativa con orden D->C->B->A puedo hacer que gane A)

Si de alguna manera ganase A, no se estaría cumpliendo la clara y lógica propiedad de un sistema de votación que es que si todos (Jurados 1-3) prefieren a D por sobre A, si usamos un sistema de votación basado en una función de bienestar Pareto-eficiente, D DEBE ganar por sobre A.

Si se cumpliera la manera que propuse arriba para que gane A, tampoco se estaría cumpliendo una de las características deseables para una función de un sistema de votación, que es que esta debe ser independiente de alternativas irrelevantes, es decir que no debería poder ser que solo gane A porque elegí el orden que elegí (se ve que si elegía C->A->B->D ganaría D y si elijo C->A->D->B gana B).

2. a. Si yo como consumidor de la tecnología (nodo) recibo la señal de mis adyacentes (pongamos el ejemplo de amigos o conocidos) que esta nueva tecnología es efectivamente mala/buena, yo no teniendo ninguna otra fuente de información, voy a actuar de acuerdo a la señal.

Para resolver este punto voy a basarme en un ejemplo hiper-simplificado y reducido para extrapolar conclusiones y luego expandirse a una conclusión más general.

Digamos que yo soy un early-adopter (por ahora me considero como el único early adopter de la red) de la tecnología y tengo 4 amigos (los cuales tendrán sus amigos recursivamente), digamos entonces que la tecnología es efectivamente 'mala', al ser una tecnología mala podemos decir que para un nodo i (adoptador de la tecnología) la probabilidad que la ganancia sea negativa será por lo menos $>50\%$ (para el ejemplo usamos un 75%). Pero digamos que a mi por algún motivo me retorna una ganancia positiva, yo le enviare la señal positiva a mis 5 amigos y ellos, solo teniendo esa información entre elegir algo que, según mi ganancia, parecer ser bueno o tener una ganancia 0, todos eligen adoptar la tecnología, usando probabilidades, 3 de mis amigos que adoptaron la tecnología van a recibir una ganancia negativa y 1 recibirá una positiva. Para los 3 que recibieron una ganancia negativa, terminará ahí su cascada, ya que sus amigos recibirán la señal negativa y entre una señal negativa y 0 elegirán todos 0 (por más que un promedio de 25% de ellos podrían recibir una ganancia positiva). Para el amigo que recibió una ganancia positiva, le mandará a sus amigos siguientes la señal positiva y todos ellos decidirán adoptar la tecnología (una vez que lo hagan, un promedio de 75% de ellos recibirán ganancia negativa). Ya desde lo dicho se puede ver que las personas (early-adopters) que mejor van a hacer que la cascada se propague por más que la tecnología en sí sea mala sean personas que obtengan una ganancia positiva y tengan un número de amigos muy alto, ya que este número muy alto recibirá la señal positiva y adoptan la tecnología por más que la mayoría termine teniendo una ganancia negativa en un final.

b. La manera que una cascada de una tecnología 'buena' no se expanda lo suficiente es justamente la mala elección de early adopters. Para el ejemplo del punto anterior, si elegimos a personas con numeros de amigos inicialmente muy bajos (digamos 1 o 2), y el promedio de ganancia positiva es 60% , habrá un 40% de chance para esos early adopters con un solo amigo que esa cascada muera en ese mismo momento. En cambio si elegimos a un early adopter con 100 amigos, sabemos que en promedio 60

de esos nodos tendrán una ganancia positiva, por lo que esos 60 nodos mandarán una señal positiva a todos sus amigos en el nivel inferior y así sucesivamente.

3. a. Nodos que finalizan con comportamiento A (con E y F como early adopters de A): E, F, C, I, K.

b. Llamemos al conjunto (E, F, C, I, K) como los adopters de A y al conjunto (G, J, H, D) como los adopters de B. Simplificando el porque A no puede expandirse a los adopters de B y cambiarlos, recae en el hecho de que los adopters de B no reciben una señal lo suficientemente fuerte de su conjunto de vecinos para cambiarse a A.

La característica que causa esto es el nivel de clustering que tienen los adopters de B, se ve como desde el conjunto solo tienen un link débil hacia el conjunto de adopters de A mediante (G-F y J-K), al tener un cierto nivel de clustering en B (la verdad que $\frac{1}{2}$ no es tan alto pero lo es para prevenir la cascada de A en este caso) esto hace que no llegue tanta influencia/comportamiento desde el conjunto de adopters de A, debido a que al tener cierto nivel de clustering, estoy en mi “burbuja” de comportamiento gracias a que nos retroalimentamos todos el comportamiento B en la burbuja de early adopters de B.

Si se consigue a H como último early adopter, se puede lograr que A se propague a través de toda la red, finalmente rompiendo este conjunto de adopters de B, haciendo que la red no tenga un cluster tan obvio como el previo lo que permite la propagación de A.

4. Para resolver los tres incisos de este punto inicialmente iba a hacer las simulaciones recomendadas, pero luego de leer bien el enunciado me di cuenta de que las respuestas se puede extrapolar pura y exclusivamente mediante el uso la teoría vista de modelos de epidemias y de ciertas características que tienen los modelos de Erdos-Renyi y de un grafo cuya distribución de grados cumple con la ley de potencias. De todas maneras para no hablar sobre la nada traigo aca unos gráficos del parcialito 1 acerca de la red de viajes, con modelos de Erdos-Renyi y Preferential Attachment (modelo cuya distribución de grados cumple con la ley de potencias).

Erdos-Renyi:

```
-----Erdos-Renyi simulation-----  
  
Average degree of original network: 24.90829694323144  
Average degree of Erdos-Renyi simulation: 25.100436681222707  
Diameter of original network: 5  
Diameter of Erdos-Renyi simulation: 3  
Clustering coefficient of original network: 0.6601565365859736  
Clustering coefficient of Erdos-Renyi simulation: 0.1133974483892974  
Average degree divided by number of nodes (which should be close to the Erdos-Renyi clustering coefficient): 0.10876985564729887
```

De esto extrapolamos que un modelo de Erdos-Renyi de una determinada red:

- Mantiene un grado promedio similar a la red original, aunque la distribución de grados no cumple una ley de potencias, sino que se asemeja más a una campana de Gauss.
- Mantiene distancias/diámetro de red similar a la red original
- Coeficiente de clustering mucho menor que el original
- Para un grado promedio $k > 1$, casi siempre mantiene una componente gigante, característica de redes sociales

Preferential Attachment:

```
Average degree of original network: 24.90829694323144
Average degree of Preferential Attachment simulation: 51.056768558951966
Diameter of original network: 5
Diameter of Preferential Attachment simulation: 2
Clustering coefficient of original network: 0.6601565365859736
Clustering coefficient of Preferential Attachment simulation: 0.22264306019431013
Average path length in original network: 2.247069639163411
Average path length in Preferential Attachment simulation: 1.7760668045660002
Connected components in original network: 1
Connected components in Preferential Attachment simulation: 1
```

De esto extrapolamos que un modelo de Preferential Attachment de una determinada red:

- Tiene un grado promedio mucho mayor al original, y su distribución de grados cumplirá una ley de potencias (“los ricos se vuelven más ricos”).
- Las distancias/diámetro de la red es menor al original
- Coeficiente de clustering menor que el original
- Mantiene componente gigante

a. El modelo SIR habla de 3 grupos: Susceptible, Infected and Recovered, pudiendo pasar desde Susceptible -> Infected y de Infected -> Recovered.

Si tanto en G1 (Erdos-Renyi) y en G2 (Ley de potencias) toda la red comienza como Susceptible excepto por un único nodo, es mucho más probable que suceda una epidemia (30% de los nodos totales infectados) en G1, ya que al agarrar un nodo random de G1, al ser una red de Erdos-Renyi es muy probable que desde el primer infectado se infecten más personas que en G2 (obviamente también depende del valor Beta que es la posibilidad de que un nodo vecino se infecte). Esto se debe claramente a que G1 va a mantener una distribución de grados que sigue una campana de Gauss (la gran mayoría de los nodos tienen un grado promedio y a medida que aumentamos o disminuimos el grado, la cantidad de nodos que poseen dicho grado se reduce drásticamente), como la consigna habla de que el virus comienza en un nodo aleatorio, en G1, en promedio, ese nodo random tendrá un grado de 25 digamos. Ahora el G2, al

cumplir una ley de potencias, tenemos que tener en cuenta que en el grafo habrá un pequeño conjunto de vértices que tendrá un grado increíblemente alto (digamos 100), pero un nodo aleatorio, en promedio, no estará en ese conjunto de nodos con grado tan alto, sino que es más probable que sea un nodo con un grado mucho más bajo (1-5). Al empezar (en promedio, ya que por probabilidad el nodo random a elegir tendrá grado bajo) en G2 en un nodo con grado bajo, le costara muchísimo más al virus comenzar su expansión, mientras el nodo aleatorio este más y más cerca a estos nodos centrales (por lo que su grado será mayor), más rápido se expandirá el virus.

El modelo de Erdos-Renyi, al ser más aleatorio pero más consistente con la distribución de grados nos asegura que si el nodo es aleatorio, al menos su grado será cercano al promedio (por probabilidad).

b. Sacando esa restricción que tenía el punto a de la aleatoriedad ahora es 100% seguro y sin ningún tipo de dudas que se expandirá más rápido en G2, por el mero hecho de que la distribución de grados en G2 cumple con una ley de potencias, por lo que se cumple el concepto de “los ricos se vuelven más ricos” y debido a esto, estos nodos con los grados mayores en G2 tienen un grado **extremadamente** alto comparado contra los demás en la red por lo que el virus tendrá muchas más chances de expandirse rápidamente desde estos nodos.

c. Las comunidades, al ser conjuntos de nodos que están bien conectados entre sí (pero no necesariamente con el resto de la red, es decir tienen un coeficiente de clustering elevado entre ellos pero tienen un par de links débiles que los conectan con el resto de la red), hacen que si el virus llegase a entrar a la comunidad, las chances de que todos se infecten son increíblemente altas, un ejemplo sería un k-clique con $k=11$ (uso un ejemplo extremo para mostrar más fácilmente la característica de las comunidades que quiero mostrar) al cual le llega el virus mediante un único weak link hacia el resto de la red, al infectarse este primer nodo, digamos que el virus tiene una chance de infección (Beta) de 30%, tras un timeslice t de los otros 10 nodos vecinos en el clique se infectan 3 más y el original entra a estar recovered, esos 3 nodos ahora dan un 30% de infección a los 7 nodos del clique, se infectan otros 3 y los 3 anteriores pasan a recovered, ahora los 3 infected dan un 30% de infección a los 4 susceptibles, and so on and so forth.

Conclusión de esto es que en una comunidad si no te infectaste apenas llegó el virus a una persona de la comunidad, esa persona seguro se lo paso a otro de tu comunidad en otro timeslice distinto, y ahora en ese otro timeslice también sos susceptible a que ese otro te infecte, y así timeslice tras timeslice hasta quedar toda la comunidad infectada.

Una característica que cabe destacar de todas maneras es que en general las comunidades tienen pocos links con el resto de la red (en comparación con los que

tienen entre los de la comunidad) por lo que tambien sera dificil que los virus entren a la comunidad en sí (obviamente también depende del factor de infectividad pero estoy hablando en promedio), pero una vez que entran es muy probable que la gran mayoría por no decir toda la comunidad termine infectada en algún momento.

5. (El .py del código estará adjuntado dentro del .zip. No lo estoy pudiendo lograr hacer funcionar completamente pero está todo el código comentado paso a paso explicando el algoritmo. Nota: como no se especificaron en la consigna y en el paper de Stanford los describen como “parámetros de ajuste”, no use α_1 , α_2 , β_1 , β_2 y γ_3).

(No adjunte el .csv en el .zip ya que son >300mb y no le vi el sentido.)