

Introducción a las dinámicas de dispersión de enfermedades infecciosas en redes de contacto, sus simulaciones y aproximaciones

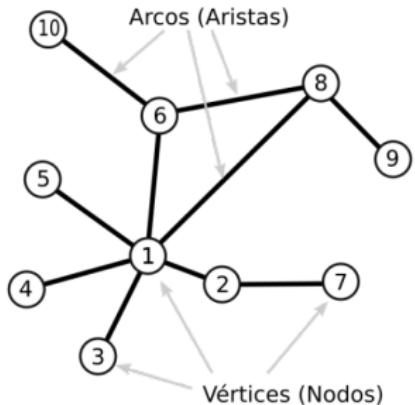
L. Leticia Ramírez Ramírez,
Centro de Investigación en Matemáticas (CIMAT), México

leticia.ramirez@cimat.mx

XXII Escuela de Probabilidad y Estadística, 2024

Introducción a Redes

Grafo (Gráfica)/Red



- ▶ Sociología (Social Network Analysis)
- ▶ Matemáticas (Grafos, Gráficas)
- ▶ Ciencias Computacionales (Grafos, Gráficas)
- ▶ Estadística, Física (Redes complejas)
- ▶ Economía (Redes)
- ▶ Bioinformática (Redes)

Ejemplos

- ▶ Redes sociales:
 - ▶ Redes de amigos, conocidos
 - ▶ Redes de colaboración
 - ▶ Redes de llamadas telefónicas
- ▶ Redes tecnológicas:
 - ▶ Internet
 - ▶ Redes telefónicas
 - ▶ Redes de transporte
- ▶ Redes biológicas
 - ▶ Redes de interacción de proteínas
 - ▶ Redes de regulación genética
 - ▶ Redes de actividad cerebral
 - ▶ Redes sociales animales
- ▶ Economía
 - ▶ Redes de instituciones, gobiernos, regiones

Grafo (Gráfica)/Red

Definición: Grafo

Un grafo (gráfica) se define como el par

$\mathcal{G} = \{V(\mathcal{G}), E(\mathcal{G})\} = \{V, E\}$, donde $V = \{v_1, v_2, \dots, v_n\}$ es el conjunto de vértices (nodos) y $E = \{e_1, e_2, \dots, e_m\} \subset V \times V$ es el conjunto de arcos (aristas).

En este caso consideramos que $n < \infty$ pero también se pueden considerar el caso *infinito*.

- ▶ El **orden** del grafo (denotado como $|V|$) es su número de vértices (n).
- ▶ El **tamaño** del grafo (denotado como $|E|$) es su número de arcos (m)

Grafo (Gráfica)/Red

Decimos que

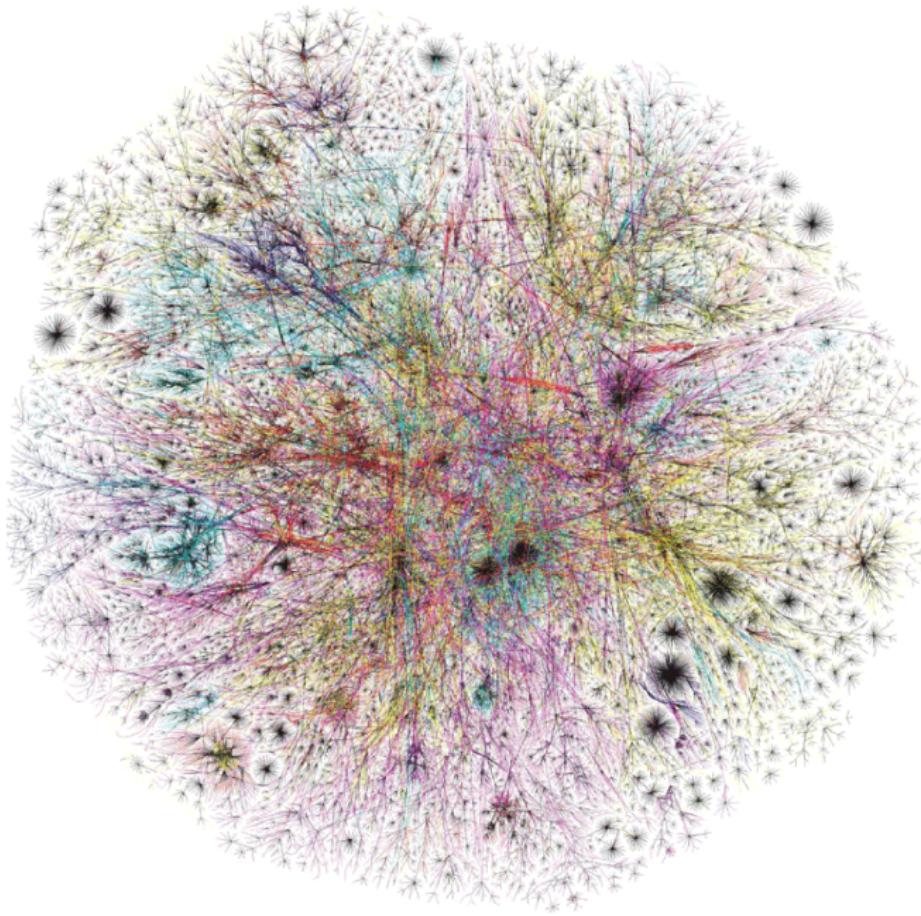
- ▶ Dos vértices u y v son **adyacentes** si existe un arco $e \in E$ tal que $e = (u, v)$. En este caso u y v son **incidentes** a e y e es **incidente** a u y v .
- ▶ Dos arcos e_1 y e_2 son **adyacentes** si ambos son incidentes al menos a un vértice común.

Definición: Grado

El grado de un vértice es el número de arcos que son incidentes a éste.

Grafos y Redes por doquier

Estuctura de Internet (www.opte.org)



Amistades en Facebook

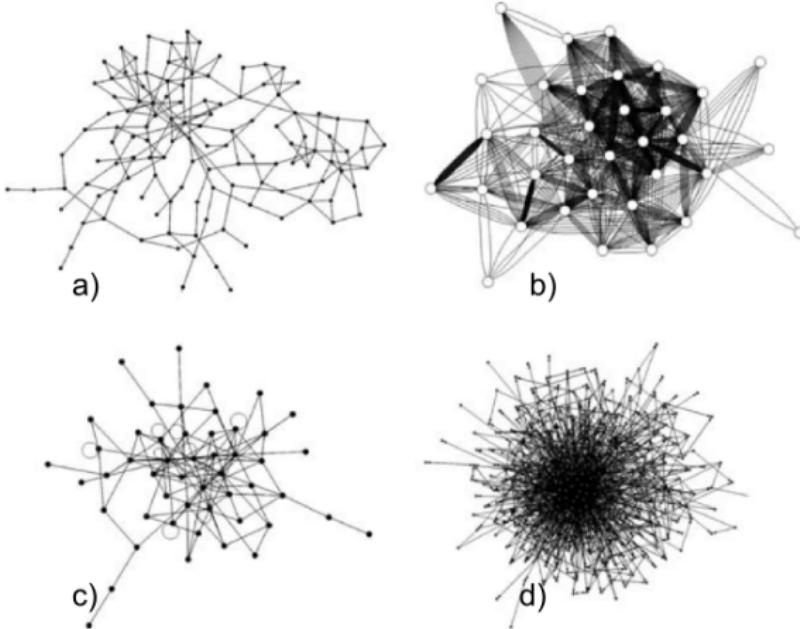


Red de Facebook (Paul Butler,
[https://www.facebook.com/notes/facebook-engineering/
visualizing-friendships/469716398919](https://www.facebook.com/notes/facebook-engineering/visualizing-friendships/469716398919))

Red de tráfico aéreo, 2013



Diversos Grafos



- a) Circuito electronico, b) Frecuencia en interaccion entre grupos de investigación, c) Interacción de proteína-proteína en el virus de herpes, d) Red metabolica de *Helicobacter pylori*. Fuente: Ernesto Estrada, 2011.

Red dirigida con pesos en arcos

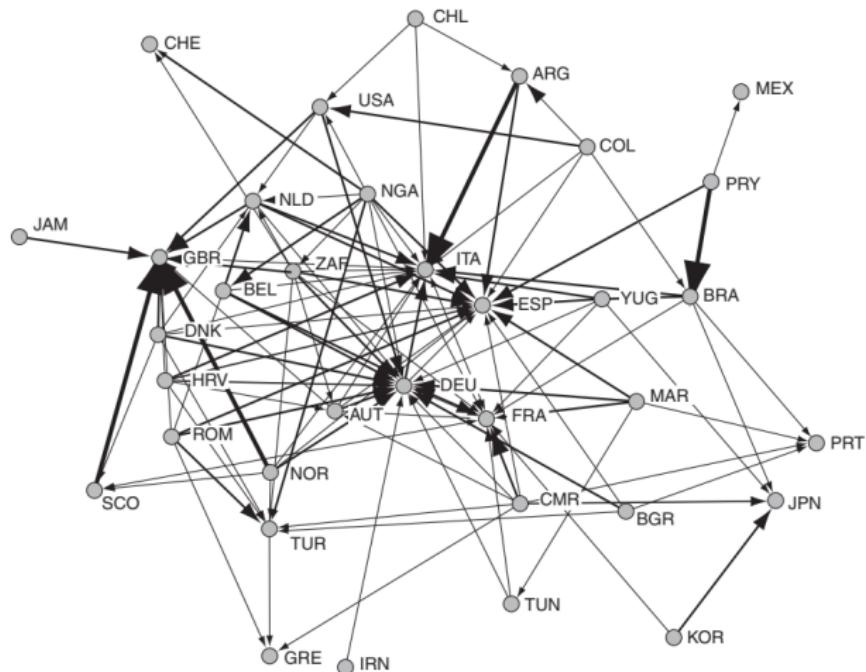


Fig. 1.5

Weighted directed network. Representation of the transfer of football players between countries after the 1998 World Cup in France. Links are drawn, with thickness proportional to the number of players transferred between two countries.

Fuente: Ernesto Estrada, 2011.

Grafo (Gráfica)/Red

- ▶ Si permitimos múltiples arcos entre dos vértices, tenemos un **grafo múltiple**.
- ▶ Si un arco puede tener como vértices incidentes al mismo vértice, este arco se llama **bucle**.
- ▶ Si en \mathcal{G} , $e = (v_i, v_j)$ es equivalente al arco (v_j, v_i) para todo $e \in E$ entonces el grafo es **no direccional**.

Definición: Grafo Simple

Un grafo se denomina simple si es un grafo sin múltiples arcos, sin pesos, sin bucles y es no direccional.

Grado

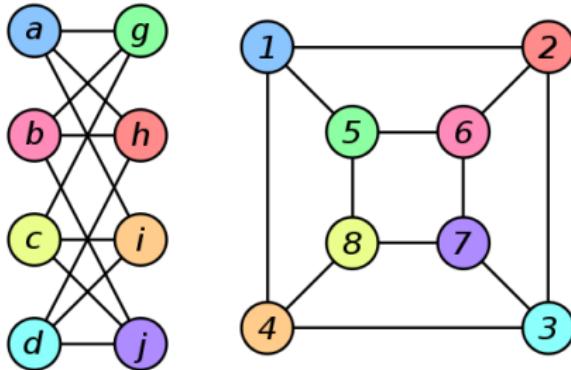
- ▶ Cuando el grafo es direccional, podemos definir el **grado interior** y **grado exterior** de un vértice como el número de arcos que se dirigen y salen del vértice, respectivamente.
- ▶ Un vértice con grado 0 se denomina **aislado**.
- ▶ Si cada vértice en el grafo tiene grado k ($k \in \mathbb{N}$), el grafo se denomina k -regular.
- ▶ La **sucesión de grados** de un grafo es una sucesión no creciente de grados de sus vértices.
- ▶ Un grafo es **completo** cuando cualquier par de sus vértices son adyacentes. En este caso, el grafo de orden n es k -regular con k igual a ...

Isomorfismo en Grafos

Definición 4

Decimos que dos grafos \mathcal{G}_1 y \mathcal{G}_2 son isomorfos si hay una biyección $f : V(\mathcal{G}_1) \rightarrow V(\mathcal{G}_2)$ tal que $(v_i, v_j) \in E(\mathcal{G}_1)$ ssi $(f(v_i), f(v_j)) \in E(\mathcal{G}_2)$.

Ejemplo:



$$\begin{aligned}f(a) &= 1 \\f(b) &= 6 \\f(c) &= 8 \\f(d) &= 3 \\f(g) &= 5 \\f(h) &= 2 \\f(i) &= 4 \\f(j) &= 7\end{aligned}$$

Observaciones

- ▶ Si dos grafos son isomorfos, su sucesión de grados coincide, pero si la sucesión de grados de dos grafos son iguales, no necesariamente son isomorfos. De un ejemplo de ésto.
- ▶ Dada una sucesión de enteros, ¿Es siempre posible construir un grafo simple?. ¿Existe uno para $2,2,2,2,1$?
- ▶ Dada una sucesión de enteros, ¿Es siempre posible construir un grafo no direccional que le corresponda?. ¿Puedes construir un grafo para la sucesión de grados $6,3,3,2,2$?

Matriz de Adyacencias

Definición: Matriz de adyacencias

Para un grafo simple, la matriz de adyacencia de \mathcal{G} , se define como la matriz cuadrada A de orden n donde su ij elemento es igual a

$$A_{ij} = \begin{cases} 1 & \text{si } (v_i, v_j) \in E(\mathcal{G}) \\ 0 & \text{eoc} \end{cases}$$

Entonces

- ▶ Para una red sin bucles, A ...
- ▶ Para una red no direccional, A ...
- ▶ Para una red múltiple, A puede extenderse a ...
- ▶ Si la red tiene bucles, A ...
- ▶ Si a una red los arcos tienen pesos, A puede extenderse a ...

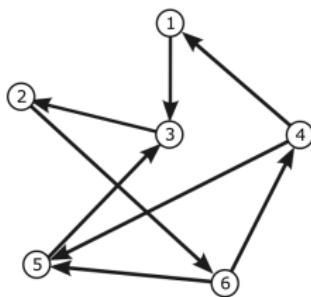
Matriz de Adyacencias: Caso direccional

Definición: Matriz de adyacencias

Para un grafo dirigido y sin múltiples arcos, la matriz de adyacencia de \mathcal{G} , con orden n , se define como la matriz cuadrada A de orden n , donde su ij elemento es igual a

$$A_{ij} = \begin{cases} 1 & \text{si el arco que va de } j \text{ a } i \text{ es elemento de } E(\mathcal{G}) \\ 0 & \text{eoc} \end{cases}$$

Ejemplo:



1	0	0	1	0	0	0
2	0	0	0	0	0	1
3	0	1	0	0	0	0
4	1	0	0	0	1	0
5	0	0	1	0	0	0
6	0	0	0	1	1	0
	1	2	3	4	5	6

Matriz de Adyacencia, Grados y tamaño

Proposición 1

Sea \mathcal{G} un grafo no dirigido, con orden n y tamaño m . Si k_i el grado del vértice i ($i \in \{1, \dots, n\}$) entonces

$$k_i = \sum_{j=1}^n A_{ij} \quad \text{y} \quad 2m = \sum_{i=1}^n k_i = \sum_{j=1}^n \sum_{i=1}^n A_{ij}.$$

Proposición 2

Sea \mathcal{G} un grafo dirigido, con orden n y tamaño m . Si k_i^{in} y k_i^{out} son los grados interior y exterior de $i \in \{1, \dots, n\} = V(\mathcal{G})$, respectivamente, entonces

$$k_i^{in} = \sum_{j=1}^n A_{ij}, \quad k_j^{out} = \sum_{i=1}^n A_{ij} \quad \text{y} \quad m = \sum_{i=1}^n k_i^{in} = \sum_{j=1}^n k_j^{out} = \sum_{j=1}^n \sum_{i=1}^n A_{ij}$$

Camino (Walk)

Definición 6

Un camino (*walk*) en un grafo \mathcal{G} es una sucesión de vértices v_1, v_2, \dots, v_k donde $(v_i, v_{i+1}) \in E(\mathcal{G}), \forall i = 1, 2, \dots, k - 1$.

Definición 6

La longitud de un camino P es el número de arcos que contiene.

Si la gráfica \mathcal{G} es simple, el número de diferentes caminos de longitud 2, $N_{ij}^{(2)}$, es igual a

$$N_{ij}^{(2)} = \sum_{k=1}^n A_{ik} A_{kj} = A^2.$$

Más aun, el número de caminatas de longitud k es igual a

$$N_{ij}^{(k)} = A^k.$$

Camino (Walk)

Definición 7

El *camino* más corto entre dos vértices i y j se define como

$$\min\{k \mid [A^k]_{ij} > 0\}$$

Ejemplo:

- ▶ Cuando los arcos tienen pesos, el camino más corto entre dos vértices se considera como el camino que minimiza la suma de los pesos.
- ▶ El problema de encontrar el camino más corto tiene diversos algoritmos entre los que destacan
 - ▶ Dijkstra
 - ▶ Bellman-Ford
 - ▶ Búsqueda A*
 - ▶ Viterbi

Más sobre conectividad

- ▶ Un **camino** es cerrado si $v_0 = v_k$.
- ▶ Un **sendero** (trail) es un camino en el que todos los arcos son distintos.
- ▶ Un **ruta o camino simple** (path) es un camino en el que todos los vértices y arcos son distintos.
- ▶ Un **ciclo** es un recorrido que es cerrado (ie todos los vértices son distintos exceptuando v_0 y v_k)
- ▶ Dos vértices i y j **están conectados** si existe un camino de i a j .
- ▶ Un **grafo es conectado** si cualquier par de vértices están conectados

Métricas y estructuras topológicas en redes

Métricas y estructuras topológicas en redes

- ▶ Diámetro
- ▶ Coeficiente de Agrupamiento
- ▶ Centralidad
- ▶ Asortatividad

Diámetro

Definición: Diámetro

El diámetro de una red es el más largo de los caminos más cortos entre todos los vértices de la red.

El diámetro mide la conectividad de la red en el mismo sentido que la “Teoría de los seis grados de separación” afirma que en el mundo no hay más de cinco intermediarios de conocidos entre cualesquiera dos personas. La teoría fue inicialmente propuesta en 1930 por el escritor húngaro Frigyes Karinthy en un cuento llamado Chains y puesta a prueba experimentalmente por primera vez por el psicólogo social estadounidense Stanley Milgram en 1967.

Diámetro

Conderamos una de las redes en **NetworkX**: "el club de karate de Zachary"(Zachary, 1977). Esta red representa las amistades entre miembros de un club de karate estudiado entre 1970 y 1972.

Este club de karate en particular ha sido de interés porque finalmente se dividió en dos clubes diferentes después de un desacuerdo entre el instructor y el presidente del club.

En el estudio original, Zachary usó la estructura de la red para predecir qué miembros se unirían a cuál de los dos clubes con una precisión casi perfecta. Específicamente, usó el algoritmo de corte mínimo. En el artículo de Zachary, el nodo con el ID 0 se identificó como el instructor del club.

Coeficiente de agrupamiento

Definición: Coeficiente de Agrupamiento

El coeficiente de agrupamiento (clustering coefficient) del vértice i se define como

$$c_i = \begin{cases} \frac{E_i}{k_i(k_i-1)/2} & \text{si } k_i > 1 \\ 0 & \text{eoc} \end{cases}$$

donde E_i es el número de arcos que conectan a vecinos de i .

- ▶ Este coeficiente es una medida de la propiedad “los amigos de mis amigos son mis amigos”.
- ▶ Como el número máximo de conexiones que pueden existir entre los amigos es $k_i(k_i - 1)/2$, el coeficiente es la razón entre las conexiones existentes entre el máximo número posible.
- ▶ El **coeficiente de agrupamiento de la red** se define como el promedio de los coeficientes de agrupamiento de todos sus nodos.
- ▶ A este coeficiente también se le denomina **Tansitividad de la red**.
- ▶ Un coeficiente de agrupamiento cercano a 1 es una indicador de la propiedad de mundo pequeño.

Centralidad (de grado)

No existe una única medida de centralidad, pero éstas buscan extraer la importancia que los nodos en la red. Buscan contestar a preguntas como “¿Qué vértice en la red debe ser escogido para asegurar que un mensaje o información se disperse al mayor número de vértices posible? En el caso de modelos epidemiológicos estos individuos son candidatos a ser vacunados o aislados.

Definición: Centralidad de grado

La centralidad de grado (degree centrality) del vértice i considera que un nodo es importante si tiene muchos vecinos. Bajo esta idea, la centralidad está definida por su grado.

Si la red es direccionalizada hay dos versiones de este grado: la centralidad hacia dentro y hacia afuera. Si el nodo es más importante por cuantos se ligan a éste, el grado hacia adentro es el que se considera.

Centralidad (Eigencentralidad)

Definición: Eigencentralidad

La eigencentralidad del nodo i se define como la i -ésima entrada del eigenvector asociado con el mayor de los eigenvalores de A .

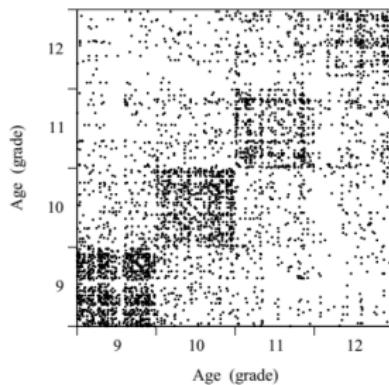
- ▶ La centralidad de eigenvector (eigenvector centrality) es más sofisticada que la de grado ya que depende no sólo de su grado sino de que tan bien estén, a su vez, conectados sus vecinos, y los vecinos de sus vecinos, y así sucesivamente.
- ▶ Una justificación a este índice se puede construir a partir del teorema de Perron-Frobenious aplicado a la matriz $B = A + I$ ya que B es simétrica, con eigenvalores reales y eigenvalores iguales a los de A más uno.
- ▶ La forma en que Google califica a las páginas web según su importancia se basa en esta idea.

Asortatividad

Definición: Asortatividad

La asortatividad es la preferencia de los nodos de una red por unirse a otros que le son similares en alguna característica (En una red poblacional, según raza o religión, por ejemplo).

Ejemplo:

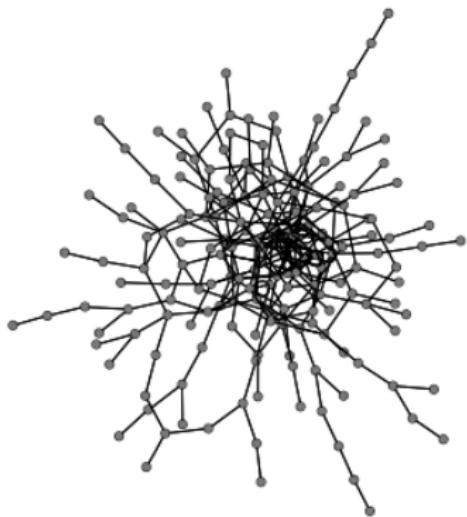


Cuando la asortatividad está en función de los grados de los nodos, ésta puede calcularse como una función de las correlaciones entre los nodos que tienen grado similar.

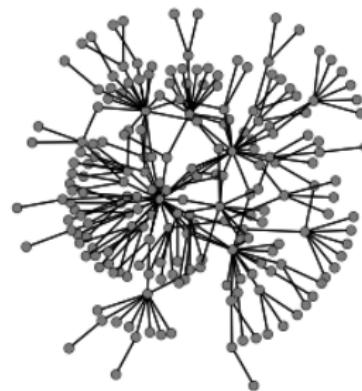
Fuente: Newman, 2010.

Asortatividad

Ejemplos de redes asortativas (a) y antisortativas (b).



(a)



(b)

Fuente: Newman, 2010.

Redes Aleatorias

Propiedades de las redes reales

En la investigación de redes nos interesa poder crear redes con el fin de ..

Las redes que se observan en la realidad y en diversidad áreas, comúnmente comparten algunas las siguientes características

- ▶ La mayoría de los vértices tienen pocos vecinos pero hay algunos cuantos que tienen muchos vecinos (alta heterogeneidad).
- ▶ Si el vértice i está conectado a j y k entonces es más probable que j y k estén conectados (alto grado de agrupamiento).
- ▶ La mayoría de los vértices están a cortos caminos unos de otros (propiedad de mundo pequeño).

Generando redes aleatoriamente

Existen diversas formas (algoritmos) para generar grafos o redes aleatorios. Las primeras propuestas se remontan a 1959. Nosotros consideraremos las propuestas de

- ▶ Erdös-Rényi, 1959 – Gilbert, 1959 (I y II)
- ▶ Prince, 1965 – Barabasi y Albert, 1999
- ▶ Watts y Strogatz, 1998

Generando redes: Erdös-Rényi I

Dado un número entero n y una valor $m \in \{0, 1, \dots, n(n - 1)/2\}$, este grafo se denota como $G(n, m)$ y corresponde al grafo simple de n vértices donde cada uno de los m arcos es aleatoriamente asignado entre cualesquiera dos diferentes vértices.

La única restricción, durante el proceso de construcción, es que un arco no puede conectar a dos vértices que ya estan conectados.

Esto es, los m arcos se seleccionan de los $n(n - 1)/2$ arcos posibles de un grafo completo.

- ▶ El número de arcos m , es un número fijo, pero el grafo resultante es seleccionado aleatoriamente de todos los grafos simples que tienen m arcos.
- ▶ ¿Esto significa que los grafos resultantes son isomorfos?

Generando redes: Erdös-Rényi I

```
In [2]: import matplotlib.pyplot as plt
import networkx as nx

n = 10 # vertices
m = 20 # arcos
seed = 2016 # semilla para generador de números aleatorios

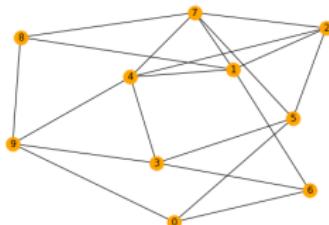
G = nx.gnm_random_graph(n, m, seed=seed)

print("Lista de Adyacencias")
for line in nx.generate_adjlist(G):
    print(line)
```

```
Lista de Adyacencias
0 9 5 6
1 4 7 8 6 2
2 4 5 7
3 5 4 6 9
4 9 7
5 7
6
7 8
8 9
9
```

Generando redes: Erdös-Rényi I

```
In [17]: pos = nx.spring_layout(G, seed=seed) #para mantener la misma posición  
nx.draw(G, with_labels=True, node_color="orange", pos=pos)  
plt.show()
```



Generando redes: Erdös-Rényi I

```
In [18]: A = nx.to_numpy_matrix(G)
A

Out[18]: matrix([[0., 0., 0., 0., 0., 1., 1., 0., 0., 1.],
 [0., 0., 1., 0., 1., 0., 1., 1., 1., 0.],
 [0., 1., 0., 0., 1., 1., 0., 1., 0., 0.1],
 [0., 0., 0., 1., 1., 1., 0., 0., 0., 1.],
 [0., 1., 1., 0., 0., 0., 1., 0., 1., 1.],
 [1., 0., 1., 1., 0., 0., 0., 1., 0., 0.1],
 [1., 1., 0., 1., 0., 0., 0., 0., 0., 0.1],
 [0., 1., 1., 0., 1., 1., 0., 0., 1., 0.1],
 [0., 1., 0., 0., 0., 0., 0., 1., 0., 1.],
 [1., 0., 0., 1., 1., 0., 0., 1., 0., 0.]])
```

Generando redes: Erdös-Rényi II

Dado un número entero n y una valor $p \in (0, 1)$, este grafo se denota como $G(n, p)$ y corresponde al grafo simple de n vértices donde cada arco entre dos vértices está presente con probabilidad p .

Es decir, cada uno de los $n(n - 1)/2$ posibles arcos en la red, “tiramos una moneda” cuya probabilidad de “sol” es p . Si el resultado es “sol” entonces añadimos el arco a la red.

- ▶ El número de arcos es aleatorio, pero ¿Cuál es el número esperado de éstos?
- ▶ Los grados de cada vértice también son aleatorios. ¿Puedes decir cuál es su distribución exacta?
- ▶ ¿Sabes cuál es su distribución aproximada cuando $n \rightarrow \infty$ y $np \rightarrow \lambda > 0$?

Generando redes: Erdös-Rényi II

```
In [28]: n = 10 # vertices
p = 0.5 # probability
seed = 2016 # semilla para generador de números aleatorios

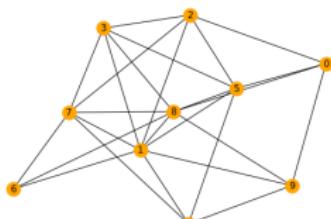
G = nx.gnp_random_graph(n, p, seed=seed)

print("Lista de Adyacencias")
for line in nx.generate_adjlist(G):
    print(line)
```

```
Lista de Adyacencias
0 2 5 8 9
1 2 3 4 5 6 7 8 9
2 3 5 7
3 5 7 8
4 5 7 9
5 8
6 7 8
7 8
8 9
9
```

Generando redes: Erdös-Rényi II

```
In [29]: pos = nx.spring_layout(G, seed=seed) #para mantener la misma posición  
nx.draw(G, with_labels=True, node_color="orange", pos=pos)  
plt.show()
```



Generando redes: Prince, 1965 – Barabasi y Albert, 1999

- ▶ Es un algoritmo de tipo evolutivo.
 - ▶ El proceso se denomina de “conexión preferencial” (preferential attachment) y se conoce como el proceso de “los ricos se vuelven más ricos”.
 - ▶ Genera redes aleatorias libres de escala (scale free). Las redes generadas poseen una distribución de grado de tipo potencial y son muy frecuentes en los sistemas elaborados por el ser humano así como en la naturaleza. Por ejemplos: El internet, redes de citas, y algunas redes sociales, redes eléctricas.
1. **Crecimiento:** comenzando con un número pequeño de vértices (m_0) y cada uno con grado de al menos 1. En cada paso añadimos un vértice con $m \leq m_0$ arcos incidentes que se usan para conectarlo con los vértices ya presentes.
 2. **Conexión Preferencial:** Escoger los vértices a los que el nuevo se conecta, se realiza con un muestreo ponderado donde la probabilidad de conectarse al vértice i , con grado k_i es $k_i / \sum_j k_j$.
 3. **Despues de t pasos**, el agoritmo resulta en una red con $n = t + m_0$ vértices y mt arcos.

Generando redes: Barabasi- Albert

```
In [41]: n = 50 # vertices
m = 3 # probability
seed = 2016 # semilla para generador de números aleatorios

G = nx.barabasi_albert_graph(n, m, seed=seed)

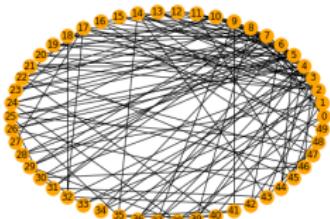
print("Lista de Adyacencias")
for line in nx.generate_adjlist(G):
    print(line)
```

Lista de Adyacencias

0 3 4 7 8 34 38 49
1 3 5 16 28 33 42
2 3 4 5 8 9 13 14 17 20 21 36 37 40 45
3 4 5 6 7 10 14 17 18 19 30 32 33 39 43 46
4 6 7 8 9 13 15 21 25 26 34 35 37 39 44
5 6 10 11 12 13 19 20 21 25 26 27 28 29 36 44
6 16 23 24 29 36 42
7 9 11 15 17 18 19 23 24 28 30 31 34
8 11 15 18 22 29 31 39
9 10 12 14 26 37 41
10 20 22 23
11 12 45
12 16
13
14 35 43 48
15
16 47
17 22 24 27 32 49
18 32
19
20

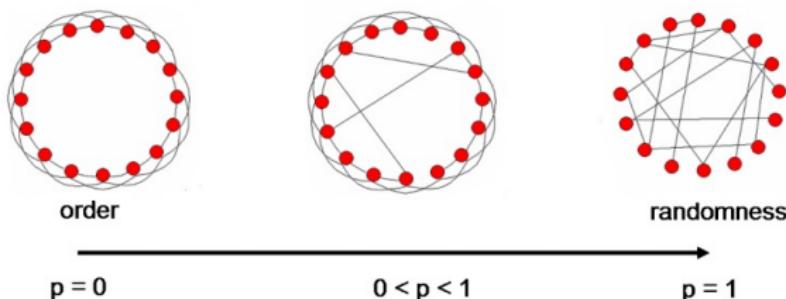
Generando redes: Barabasi- Albert

```
In [34]: pos = nx.circular_layout(G) # posicion circular  
nx.draw(G, with_labels=True,node_color="orange",pos=pos)  
plt.show()
```



Generando redes: Watts y Strogatz, 1998

- ▶ Es un algoritmo de tipo evolutivo.
 - ▶ La distribución de grado de los grafos generados mediante este algoritmo depende del valor del parámetro p .
1. **Comienza** con un anillo, donde cada vértice está conectado a los siguientes z vértices
 2. En cada **paso** se selecciona uno de los vértices sus arcos incidentes se reconectan con el siguiente esquema. Sea (v_i, v_j) uno de los arcos del vértice v_i seleccionado. Con probabilidad p éste se remueve y se substituye por el arco (v_i, v_r) , donde v_r es seleccionado aleatoriamente del conjunto $V \setminus v_i$.



Generando redes: Watts y Strogatz, 1998

```
In [59]: n = 20 # vértices
k = 4 # número de vecinos más cercanos a los que está conectado
p = 0.2 # probabilidad de reconexión
seed = 2016 # semilla para generador de números aleatorios

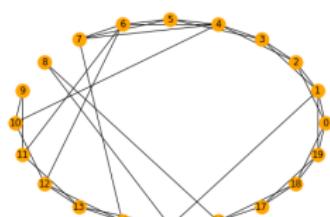
G = nx.watts_strogatz_graph(n, k, p, seed=seed)

print("Lista de Adyacencias")
for line in nx.generate_adjlist(G):
    print(line)
```

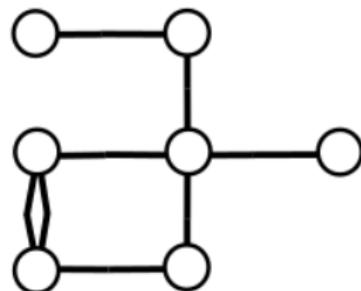
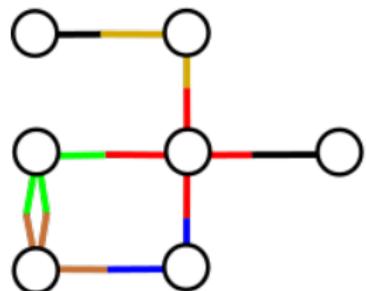
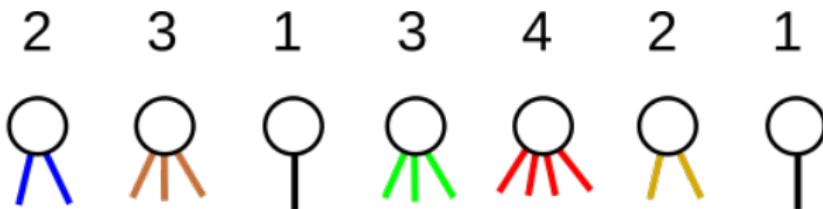
```
Lista de Adyacencias
0 1 19 2 18
1 2 3 19 15
2 3 4
3 4 5
4 5 6 7 10
5 6 7
6 7 11 12
7 14
8 15 16
9 10 11
10 12
11 13
12 13 14
13 14 15
14 15 16
15 16
16 17 18
17 18 19
18 19
19
```

Generando redes: Watts y Strogatz, 1998

```
In [60]: pos = nx.circular_layout(G) # posicion circular  
nx.draw(G, with_labels=True,node_color="orange",pos=pos)  
plt.show()
```

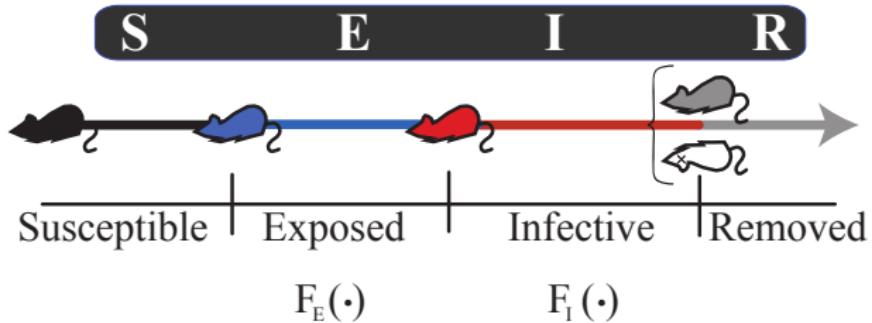
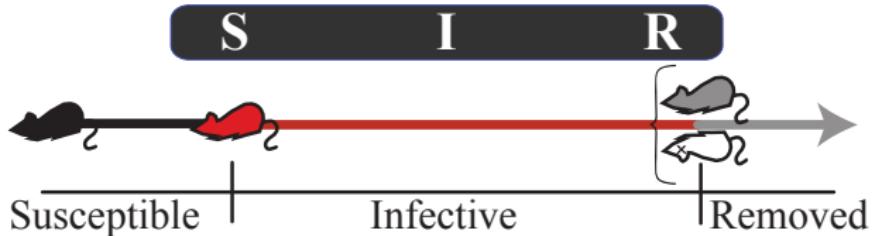


Generando redes: Modelo de Configuración



Introducción a Modelos Epidemiológicos

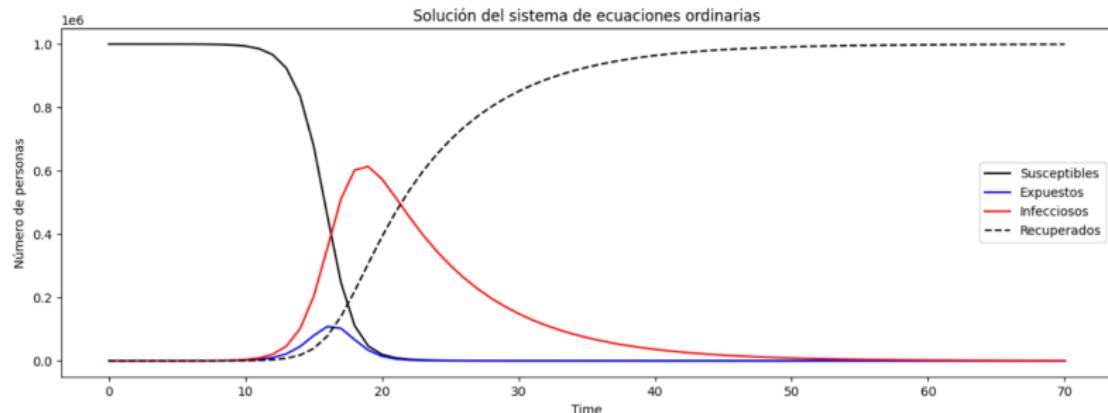
Modelo compartmental



Modelo compartimental: SEIR

Bajo la ley de acción de masas, el modelo es:

$$\begin{aligned}S'(t) &= -\beta SI, \\E'(t) &= \beta SI - \delta E, \\I'(t) &= \delta E - \gamma I, \\R'(t) &= \gamma I.\end{aligned}$$



R_0

En general el parámetro R_0 se conceptualiza como el número de casos secundarios originados por un solo infeccioso cuando se encuentra en una población donde todos los demás son susceptibles.

Se utiliza ampliamente para identificar si un brote evolucionará en una epidemia o si se extinguirá afectando solo a un pequeño número de personas.

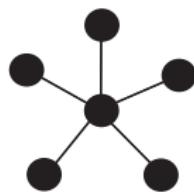
Para los modelos SIR, SEIR:

$$R_0 = \frac{\beta}{\gamma}.$$

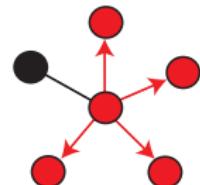
Si $R_0 > 1$ se tendrá un escenario de epidemia. En caso contrario, de un brote que se extingue

Basics

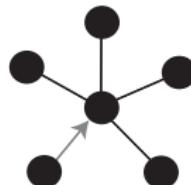
Grados



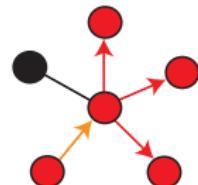
Grados
Ocupados



Grados
Excedentes



Grados
Ocupados
Excedentes



$$K = 5$$

$$K_T = 4$$

$$K_1 = 4$$

$$K_{T1} = 3$$

Brotes en redes simples

GIF

Probability of epidemic and threshold value

Newman (2002) notó que el **número esperado de nuevos infecciosos originados por un infeccioso típico es**

$$\mathcal{R} = \pi E(K_1)$$

donde π es la probabilidad de transmisión de un infeccioso a uno de sus vecinos, define si el brote evoluciona en una epidemia.

En contraste

$$R_0 = \pi E(K).$$

Si $\mathcal{R} > 1$ la probabilidad de que el brote evolucione en una epidemia es mayor a cero.

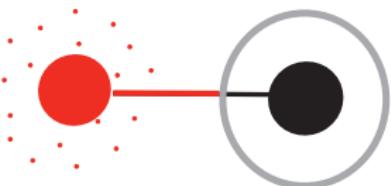
Si $\mathcal{R} < 1$ con total certidumbre todos los brotes se extinguirán pronto.

Medidas de Control

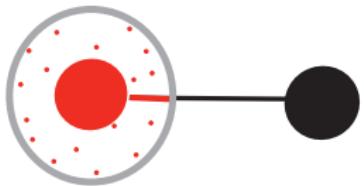
Control Measures



Reduce the number of contacts



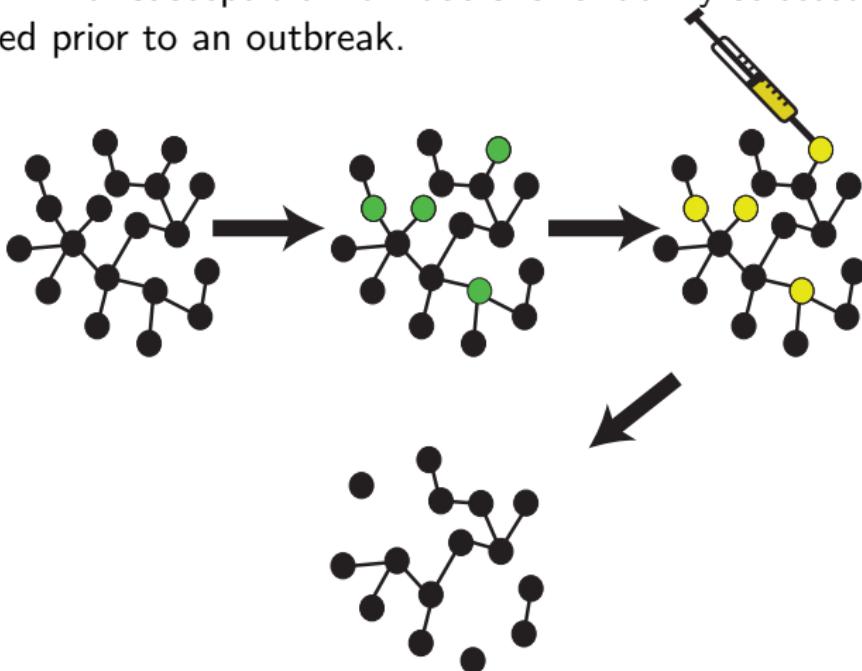
Reduce the susceptibility to the agent



Reduce the infectious agent emissions

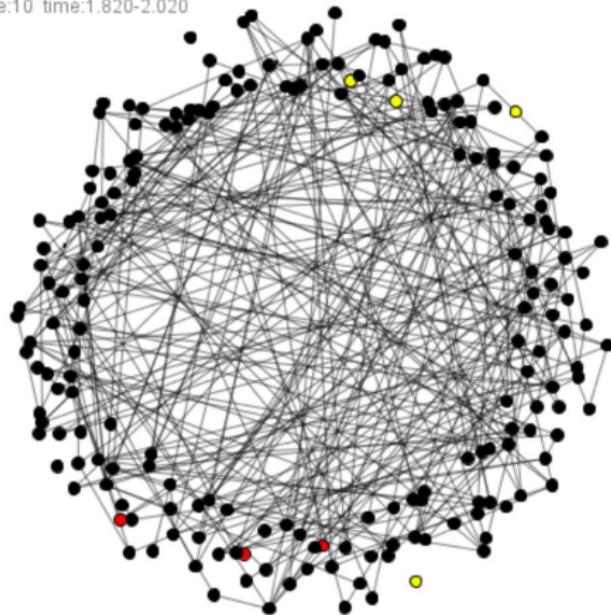
Mass Vaccination

A fraction ν of susceptible individuals is randomly selected and vaccinated prior to an outbreak.



Mass Vaccination

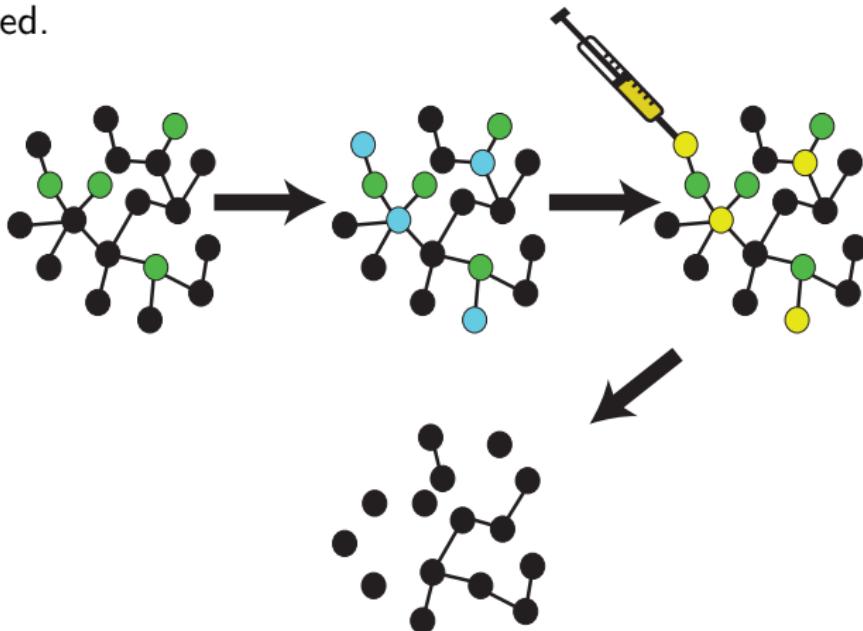
slice:10 time:1.820-2.020



R code visualize with
Social Network Image Animator (Sonia)

Acquaintance Vaccination

First a fraction ω of susceptible nodes is randomly chosen. Then a neighbor of each of those nodes is randomly selected and vaccinated.



Mass and Acquaintance Vaccination

The replacement numbers are modified as

$$\mathcal{R}_{\text{mass}} = (1 - \nu)\mathcal{R}$$

and

$$\mathcal{R}_{\text{acquaintance}} = \pi \sum_{k=2}^{\infty} \frac{k(k-1)p_k}{E(K)} \left(E_{K_e}(e^{\omega/k}) \right)^{k-2} e^{-2\omega/k}$$

(Cohen et al. 2003)

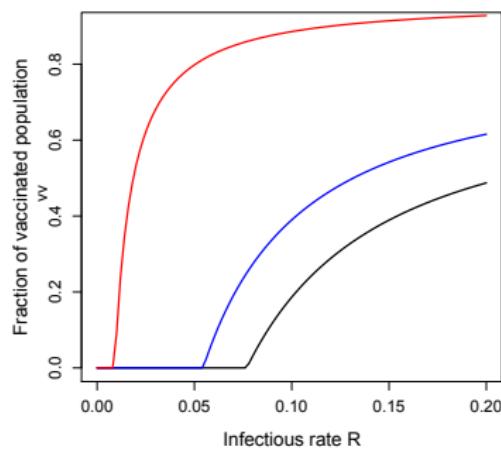
Mass and Acquaintance Vaccination

We consider an infectious period $I \sim N(5.29, 0.625)$, constant infectious rate and three networks with different heterogeneity level but same degree mean ($=4$):

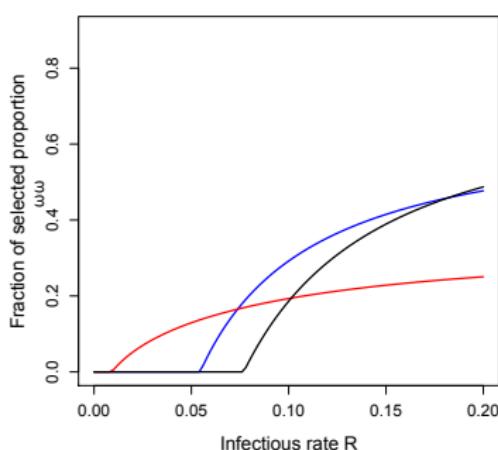
- ▶ Constant($=4$)
- ▶ Poisson (4)
- ▶ Polylogarithmic (1.6599,50)

Proportion of selected individuals to reach the threshold $\mathcal{R} = 1$

Mass vaccination



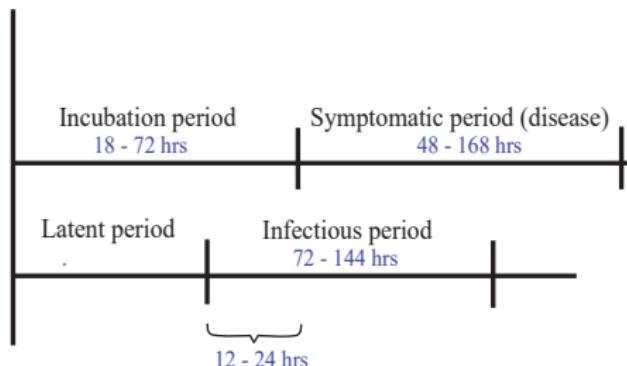
Acquaintance vaccination



Ring vaccination

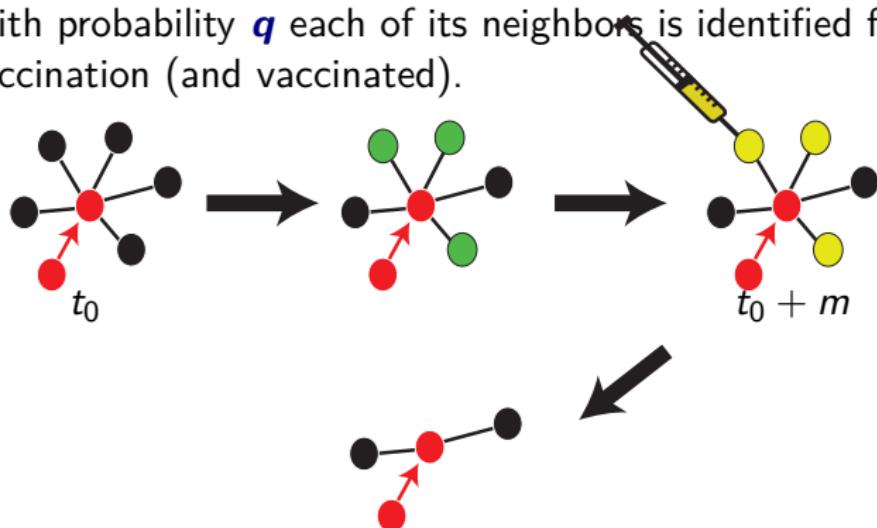
- ▶ With probability γ any infectious node i is detected.
- ▶ This detection is done after m units of time since i is infectious.
- ▶ With probability q each of its neighbors is identified for vaccination (and vaccinated).

Infectious contact



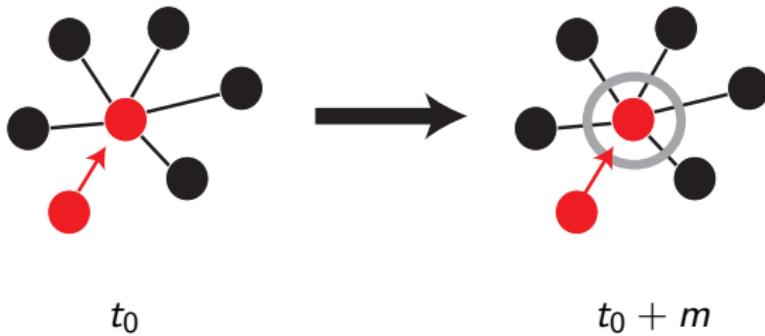
Ring vaccination

- ▶ With probability γ any infectious node i is detected.
- ▶ This detection is done after m units of time since i is infectious.
- ▶ With probability q each of its neighbors is identified for vaccination (and vaccinated).



Quarantine-Isolation

- ▶ With probability γ any infectious node i is detected.
- ▶ This detection is done after m units of time since i is infectious.
- ▶ Once detected i is “isolated”, reducing its transmission rate by $1-\alpha \times 100\%$.



Ring vaccination and isolation

The replacement numbers are modified as

$$\mathcal{R}_{\text{ring}} = E(K_1) [(1 - q\gamma)\pi + q\gamma\pi(m)]$$

and

$$\begin{aligned}\mathcal{R}_{\text{isolation}} &= \\ E(K_1) \left[(1 - q\gamma)\pi + \gamma \left(\pi_m + \int_m^\infty \{1 - E_R(e^{-Rm})E_Y(e^{-Y(I-m)})\} dF_I(I) \right) \right],\end{aligned}$$

where

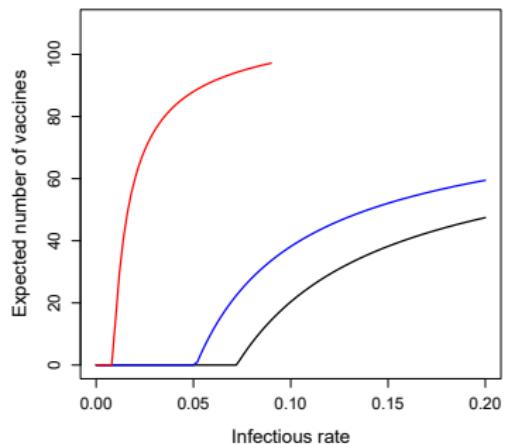
$$\begin{aligned}Y &= \alpha R, \\ \pi_t &= \int_0^t (1 - E_R(e^{-Rl})) dF_I(l), \text{ and} \\ \pi(t) &= \pi_t + E_R(1 - e^{-Rt})P(I > t).\end{aligned}$$

Ring vaccination and isolation

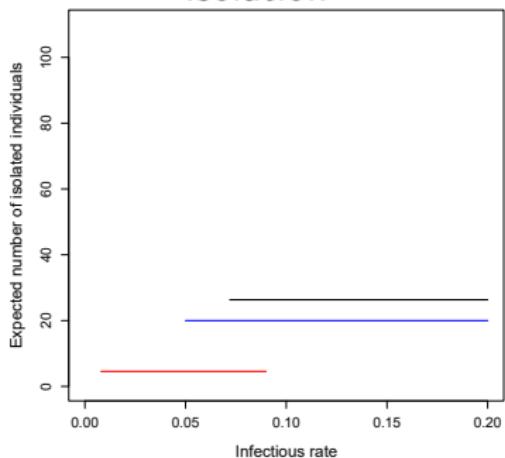
- ▶ Constant(=4)
- ▶ Poisson (4)
- ▶ Polylogarithmic (1.6599,50)

Proportion of selected individuals to reach the value $\mathcal{R} = .95$, with
 $\gamma = 1$ and $m = 0.5$

Ring vaccination



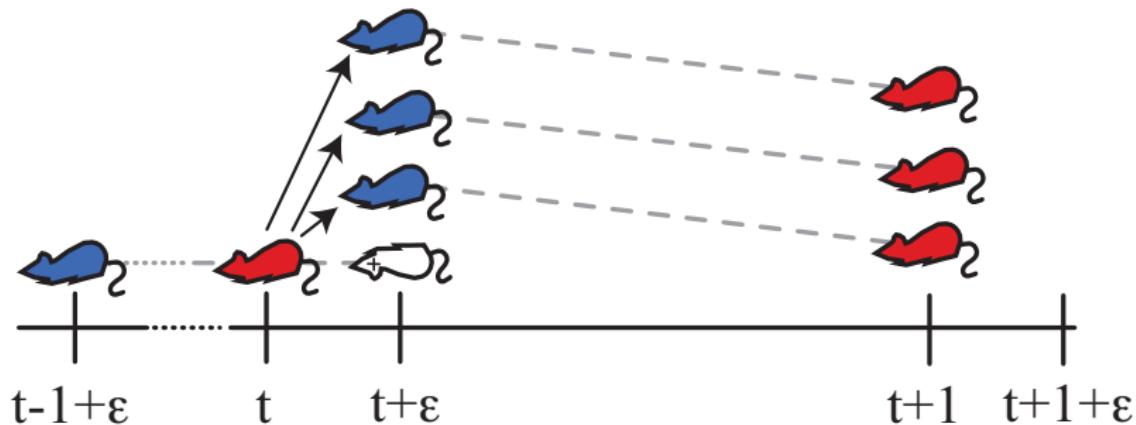
Isolation



Simulaciones del proceso

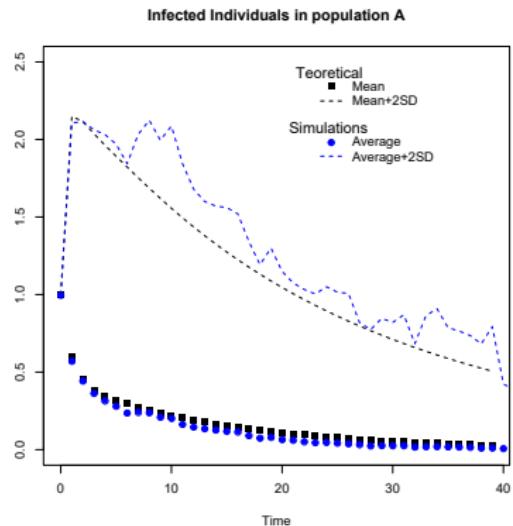
Modelo de Reed-Frost

Después de que un susceptible se infecta, permanece expuesto por una “unidad” de tiempo antes de volverse infeccioso por un periodo corto de tiempo, pero entonces pasar la infección a otros individuos susceptibles.

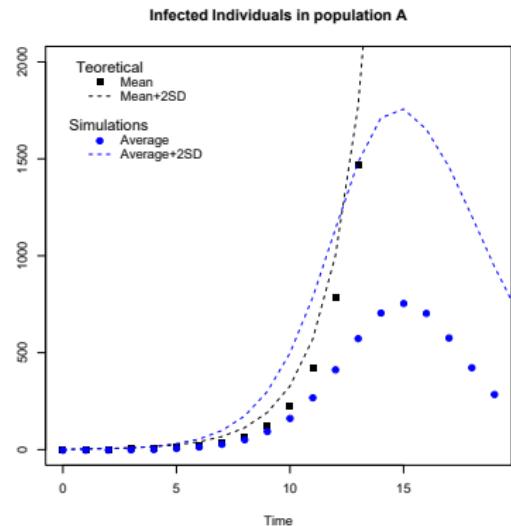


Los nodos infecciosos transmiten la infección a cualquiera de sus vecinos con probabilidad τ .

Ejemplo



$$\tau = 0.3.$$



$$\tau = 0.6.$$

Método directo de Gillespie.

Paso 0 (inicialización).

Tiempo $t = 0$.

$X_1(0), \dots, X_N(0)$ el estado inicial del sistema.

Calculamos $\lambda_1, \dots, \lambda_M$ las tasas de reacción.

Definimos la tasa total $\Lambda = \sum_{j=1}^M \lambda_j$.

Paso 1. De acuerdo a una distribución exponencial de parámetro Λ , obtenemos un tiempo τ .

Paso 2. Generamos u con distribución uniforme en $(0, 1)$ e identificamos i^* tal que

$$\sum_{j=1}^{i^*-1} \lambda_j < u\Lambda \leq \sum_{j=1}^{i^*} \lambda_j.$$

Paso 3. Actualiza el sistema \mathbf{X} según la reacción i^* .

Paso 4. Actualiza el tiempo $t = t + \tau$.

Paso 5. Actualiza las tasa total Λ según el estado actual del sistema.

Paso 6. Vuelve al paso 1.

Gillespie para SIR en redes.

Paso 0 (inicialización).

Tiempo $t = 0$.

$S(0), I(0), R(0)$ el estado inicial del sistema.

Definimos el conjunto de nodos susceptibles $S = S_0(t) \cup S_1(t) \cup \dots \cup S_N(t)$.

Definimos $\lambda_j = \beta \times |S_j(t)| \times j$, para $j = 1, \dots, N$.

Calculamos la tasa total $\Lambda = \left(\sum_{j=1}^N \lambda_j \right) + \gamma I$.

Paso 1. De acuerdo a una distribución exponencial de parámetro Λ , obtenemos un tiempo τ .

Paso 2. Identificamos la reacción *infección* con probabilidad $\sum_{j=1}^N \lambda_j / \Lambda$, o identificamos la reacción *recuperación* con probabilidad $\gamma I / \Lambda$.

Paso 3.

Si la reacción fue *infección*, generamos u con distribución uniforme en $(0, 1)$, e identificamos k^* como el mínimo entero tal que

$$u \left(\sum_{j=1}^N \lambda_j \right) \leq \sum_{j=1}^{k^*} \lambda_j.$$

Gillespie para SIR en redes (cont.)

Seleccionamos aleatoriamente un individuo susceptible i^* que satisface tener k^* contactos infecciosos.

Si la reacción fue *recuperación*, seleccionamos aleatoriamente un individuo infeccioso i^* .

Paso 4. Actualiza el estado del individuo i^* según sea el caso.

Paso 5.

Si el individuo i^* fue actualizado a infeccioso, para cada nodo j conectado a i^* con k_j posibles contactos infecciosos, actualizamos

$$S_{k_j} = S_{k_j} \setminus \{j\}, \quad S_{k_j+1} = S_{k_j+1} \cup \{j\}.$$

Si el individuo i^* fue actualizado a recuperado, para cada nodo j conectado a i^* con k_j posibles contactos infecciosos, actualizamos

$$S_{k_j} = S_{k_j} \setminus \{j\}, \quad S_{k_j-1} = S_{k_j-1} \cup \{j\}.$$

Paso 6. Actualiza el tiempo $t = t + \tau$.

Paso 7. Actualiza las tasas λ_j y Λ según el estado actual del sistema.

Paso 8. Vuelve al paso 1.

Aproximaciones de Campo Medio

Aproximación de Campo Medio

Es una técnica utilizada para simplificar el estudio de sistemas con muchas partes interactuantes. Esta aproximación se emplea para modelar sistemas complejos, como redes neuronales, modelos de espines en física estadística, y en teoría de juegos, entre otros.

El concepto básico detrás es sustituir las interacciones entre todas las partículas del sistema por *una interacción promedio*.

Aproximación de Campo Medio

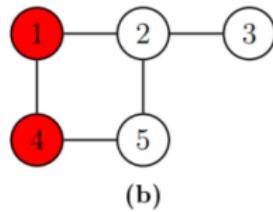
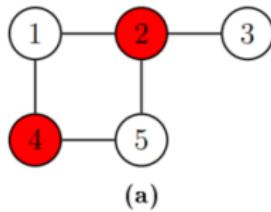


Figura 3.3: Ejemplo de distintas dinámicas de infección con las mismas distribuciones de individuos susceptibles e infecciosos.

Aproximación de Campo Medio: SIS en grafica k reg.

Simon y Kiss (2013) proponen una aproximación muy intuitiva del número de aristas que conectan a un individuo Susceptible con un individuo Infeccioso para gráficas k -regulares, dada por

$$N_{SI}(m) = m \frac{n - m}{n} k$$

donde m es el número de individuos infecciosos en la gráfica.

Se puede interpretar como el número de individuos infecciosos m multiplicado por el grado k de cada uno de esos individuos infecciosos, multiplicado por la proporción de individuos susceptibles en la gráfica.

En este sentido, la aproximación considera que hay m individuos infecciosos distribuidos aleatoriamente en la gráfica.

Aproximación de Campo Medio: SIS en $G(n, p)$

Dado que el número de maneras en la que podemos elegir los extremos de una arista seleccionada de manera aleatoria es $n(n - 1)$, si el número de individuos infecciosos están distribuidos uniformemente sobre la gráfica, entonces la probabilidad de que la arista seleccionada sea Susceptible-Infeccioso es

$$\hat{p} = \frac{2m(n - m)}{n(n - 1)}$$

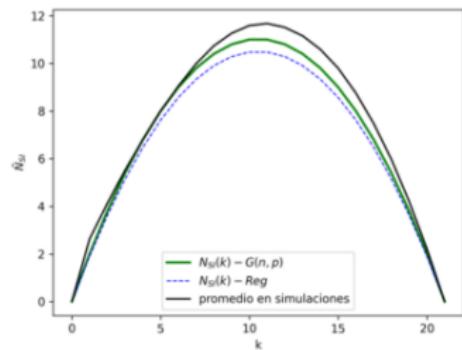
por lo que el valor esperado del número de aristas Susceptible-Infeccioso corresponde a

$$E \left(\sum_{v=1}^n \hat{p} d_v \right) = m \frac{n - m}{n - 1} (n - 1)p = m(n - k)p,$$

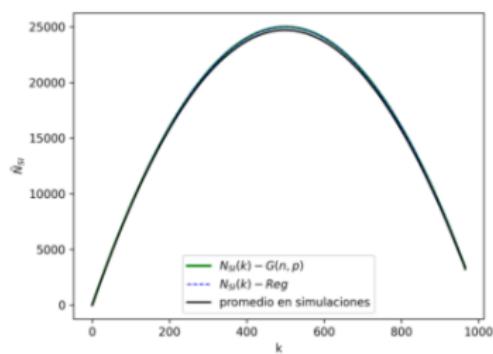
donde $(n - 1)p = E(K)$ del modelo $G(n, p)$ por tener distribución Binomial.

Aproximación de Campo Medio: SIS en $G(n, p)$

1000 simulaciones, cada una en una gráca aleatoria del modelo $G(n, p)$, con $p = 0.1$, que se asocia a gráfica con $E(K) = 2$.



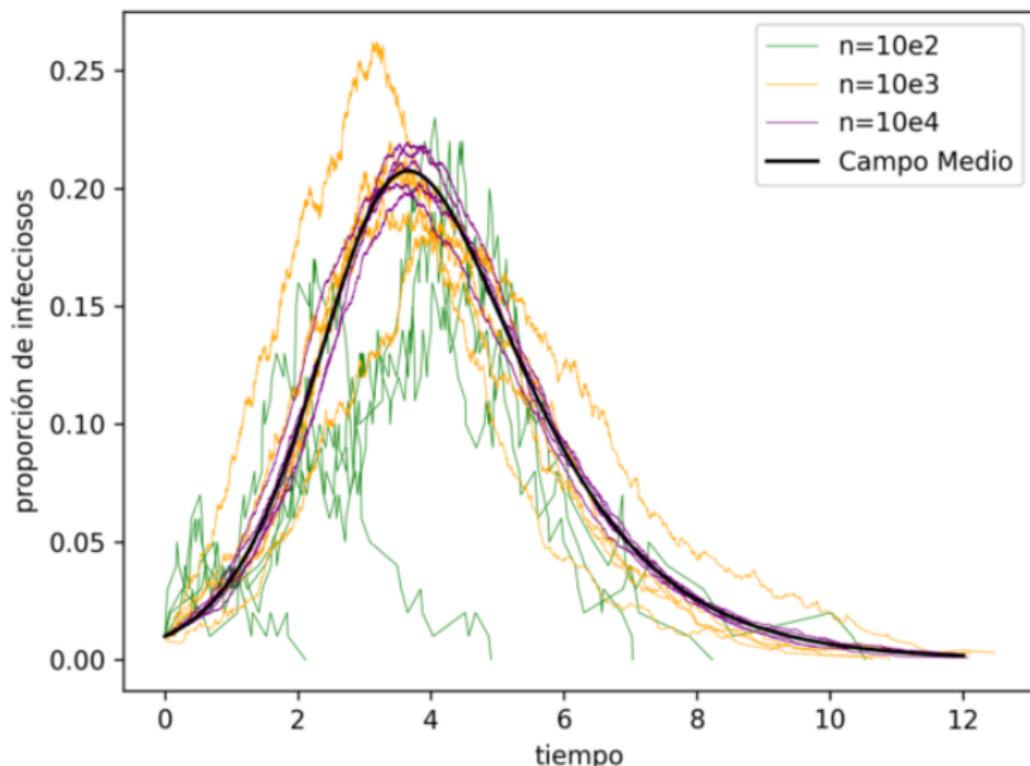
(a) $G(n = 21, p = 0.1)$ con grado promedio 2.



(c) $G(n = 10^3, p = 0.1)$, con grado promedio 100.

Aproximación de Campo Medio: SIS en $G(n, p)$

1000 simulaciones, cada una en una gráca aleatoria del modelo $G(n, p)$, con $p = 0.1$, que se asocia a gráfica con $E(K) = 2$.



Inferencia Estadística

The Approximate Bayesian computation (ABC)

When the observations $\mathbf{y} \in \mathcal{D}$ can be specified by the model $f(\mathbf{y} | \theta)$ the Bayesian inference on the parameter θ can be obtained from the equation

$$\pi(\theta | \mathbf{y}) \propto \pi(\theta) f(\mathbf{y} | \theta).$$

However the likelihood function $f(\mathbf{y} | \theta)$ can be very complicated or completely intractable. In these cases we want to make inference *without the likelihood function*.

The ABC ([Tavaré et al.,1997](#); [Del Moral et al.,2012](#); [Marin et al.,2012](#)) is a class of methods that provide an alternative to the likelihood computation. It basically requires being able to sample pseudo observations \mathbf{x} from $f(\cdot | \theta)$.

Metropolis - Hastings

The Metropolis–Hastings algorithm can draw samples from any probability distribution with probability density $p(x)$, provided that we know a function $\tilde{p}(x)$ proportional to the density $f(x)$ and the values of $\tilde{p}(x)$ can be calculated.

Algorithm 1 Metropolis-Hastings

1. Either set $X_0 = x_0$, or draw X_0 from some initial distribution
2. For $t = 1, 2, \dots, n - 1$:
3. Assume $X_{t-1} = x_{t-1}$.
4. Simulate $Y_t \sim q(\cdot | x_{t-1})$ and $U_t \sim \text{Unif}[0, 1]$.
5. If

$$U_t \leq \alpha(Y_t | x_{t-1})$$

where

$$\alpha(y | x) = \min \left\{ 1, \frac{\tilde{p}(y)q(x | y)}{\tilde{p}(x)q(y | x)} \right\}$$

set $X_t = Y_t$, otherwise set $X_t = x_{t-1}$.

Metropolis-Hastings algorithm for posterior distribution

Algorithm 2 Metropolis-Hastings

If now at $\theta^{(t)}$,

1. Generate ν_t from the transition distribution $q(\cdot | \theta^{(t)})$
2. Do

$$\theta^{(t+1)} = \begin{cases} \nu_t & \text{with probability } \alpha, \\ \theta^{(t)} & \text{with probability } 1 - \alpha. \end{cases}$$

where

$$\alpha = \min \left\{ \frac{\pi(\nu_t | \mathbf{y})}{\pi(\theta^{(t)} | \mathbf{y})} \frac{q(\theta^{(t)} | \nu_t)}{q(\nu_t | \theta^{(t)})}, 1 \right\}$$

The Approximate Bayesian computation (ABC)

The ABC methods draws inference using the modified posterior density on $\Theta \times \mathcal{D}$

$$\pi_\epsilon(\theta, \mathbf{x}|\mathbf{y}) \propto \pi(\theta) f(\mathbf{x}|\theta) \mathbb{I}_{A_{\epsilon,\mathbf{y}}}$$

where $A_{\epsilon,\mathbf{y}}$ are the pseudo-observations \mathbf{x} that are “closer up to ϵ ” to the true observations \mathbf{y} . Formally

$$A_{\epsilon,\mathbf{y}} = \{\mathbf{z} \in \mathcal{D} : \rho(\eta(\mathbf{z}), \eta(\mathbf{y})) < \epsilon\},$$

where $\eta(\cdot)$ summarizes the observations and ρ is a distance function.

Implemented ABC

A modification to the [Wilkinson\(2008\)](#) method, replaces the indicator by a kernel function K_h evaluated on a joint statistics for \mathbf{x} and \mathbf{y} :

$$\pi_h(\theta, \mathbf{x} | \mathbf{y}) = \frac{\pi(\theta) f(\mathbf{x} | \theta) K_h(t(\mathbf{x}, \mathbf{y}))}{\int \pi(\theta) f(\mathbf{z} | \theta) K_h(t(\mathbf{z}, \mathbf{y})) d\mathbf{z} d\theta},$$

where h is the bandwidth and $t(\mathbf{x}, \mathbf{y})$ in particular can be equal to $\rho(\eta(\mathbf{z}), \eta(\mathbf{y}))$.

Implemented ABC-MCMC

Let $\mathbf{x}_{\theta^{(t)}}$ and \mathbf{x}_ν be the pseudo observations from the model based on the value $\theta^{(t)}$ and ν , respectively.

The acceptance probability $\alpha = \min\{a, 1\}$ associated to the target posterior $\pi_h(\theta, \mathbf{x}|\mathbf{y})$ is that with

$$\begin{aligned} a &= \frac{\pi_h(\nu, \mathbf{x}|\mathbf{y})}{\pi_h(\theta^{(t)}, \mathbf{x}|\mathbf{y})} \frac{g(\theta^{(t)}, \mathbf{x}|\nu, \mathbf{x}_\nu)}{g(\nu, \mathbf{x}|\theta^{(t)}, \mathbf{x}_{\theta^{(t)}})} \\ &= \frac{\pi(\nu)K_h(t(\mathbf{x}_\nu, \mathbf{y}))}{\pi(\theta^{(t)})K_h(t(\mathbf{x}_{\theta^{(t)}}, \mathbf{y}))} \frac{q(\theta^{(t)} | \nu)}{q(\nu | \theta^{(t)})}, \end{aligned} \quad (1)$$

since

$$g(\nu, \mathbf{x}|\theta, \mathbf{x}_\theta) = q(\nu | \theta)f(\mathbf{x} | \nu).$$

Implemented ABC-MCMC

Algorithm 3 ABC-MCMC, (Marjoram et al., 2003 and Wilkinson, 2008)

If now at $\theta^{(t)}$ with $\mathbf{x}_{\theta^{(t)}}$,

1. Generate ν from the transition kernel $q(\cdot | \theta^{(t)})$
2. Simulate a pseudo-observacion \mathbf{x}_ν from $f(\cdot | \nu)$.
3. Do

$$\theta^{(t+1)} = \begin{cases} \nu & \text{with probability } \alpha, \\ \theta^{(t)} & \text{with probability } 1 - \alpha. \end{cases}$$

where $\alpha = \min\{a, 1\}$ with

$$a = \frac{\pi(\nu) K_h(t(\mathbf{x}_\nu, \mathbf{y}))}{\pi(\theta^{(t)}) K_h(t(\mathbf{x}_{\theta^{(t)}}, \mathbf{y}))} \frac{q(\theta^{(t)} | \nu)}{q(\nu | \theta^{(t)})}.$$

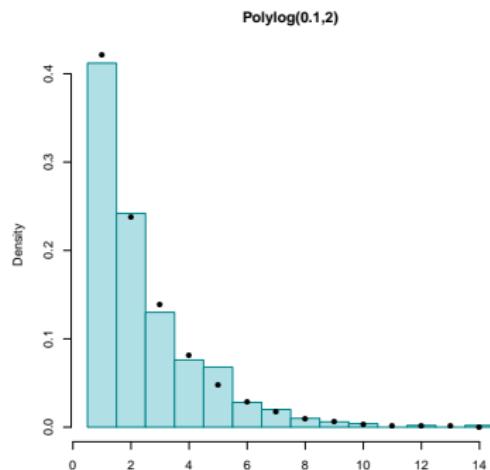
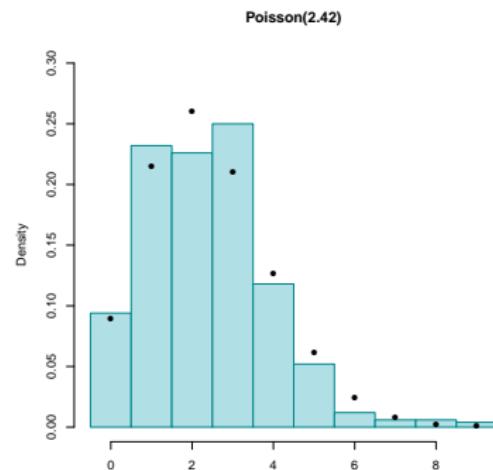
Computational Experiments

We simulate two random networks with order 500 and degree distributions:

- ▶ Poisson(2.42)
- ▶ Polylogarithmic(0.1,2)

These networks are considered the population contact in a hypothetical population.

Degree distributions

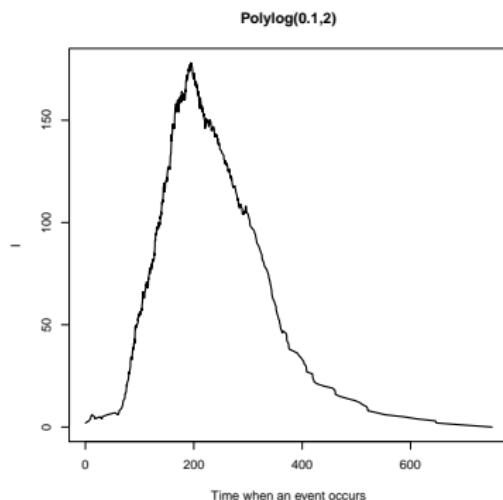
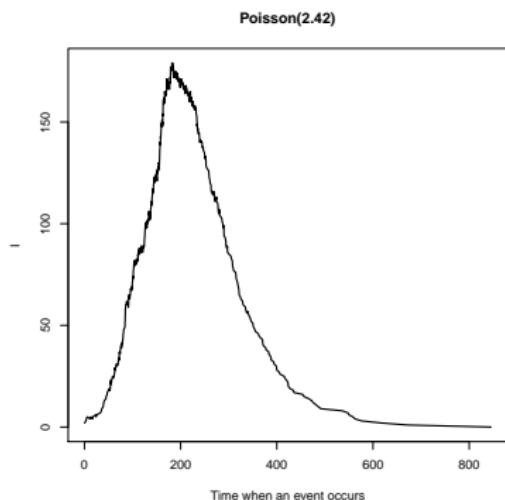


Computational Experiments

Assuming two random individuals infected, we simulate the spread of an infectious agent under an epidemic stochastic SIR model, considering

$$\beta = 0.03 \quad \text{and} \quad \gamma = 0.01$$

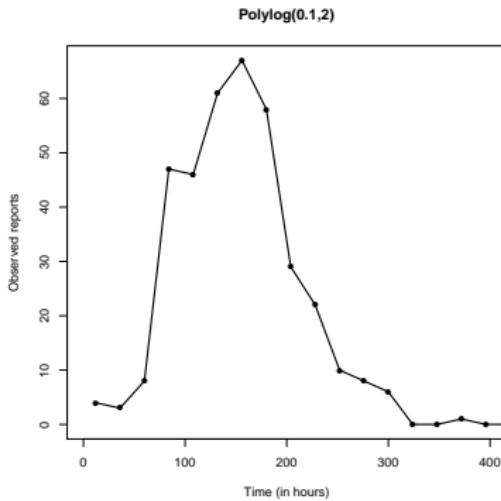
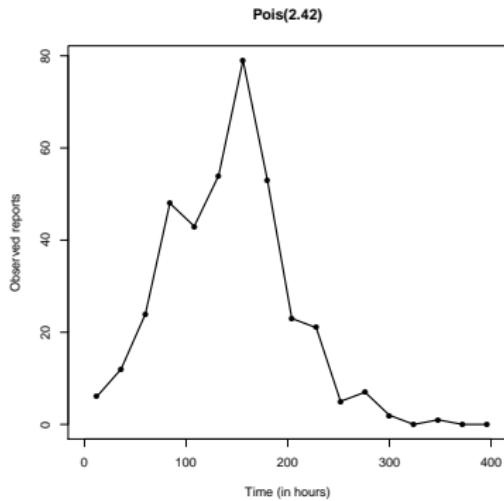
as hourly rates of infection and recovery, respectively.



Computational Experiments

For a real-life scenario, we consider to have daily reports of new infected individuals. It means, we don't have the exact number of infected people in each time, instead we have the aggregated number of new cases in daily (24 hours) intervals.

Then, from the simulated process, we calculate $\mathbf{y} = (y_1, \dots, y_n)$, which we assume to be the observed reports of new infected individuals.



Computational Experiments

To measure the difference between the observed data \mathbf{y} and a simulated vector \mathbf{x} , we consider the following statistic

$$t(\mathbf{x}, \mathbf{y}) = \sum_i \sqrt{\frac{(x_i - y_i)^2}{y_i \mathbb{I}(y_i > 0) + \varepsilon \mathbb{I}(y_i = 0)}}, \quad (2)$$

where $\varepsilon > 0$ is a small constant introduced to define $t(\mathbf{x}, \mathbf{y})$ beyond the observed outbreak span.

Computational Experiments

To implement the ABC-MCMC algorithm, the proposal distribution considered was a Normal Mixture:

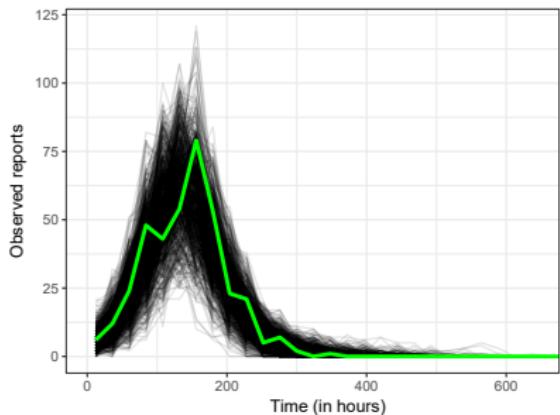
$$q(\theta' | \theta) = r N(\theta, \Sigma_1) + (1 - r) N(\theta, \Sigma_2).$$

Where $\Sigma_1 = 0.05I_d$, $\Sigma_2 = 0.1I_d$ and $0 < r < 1$.

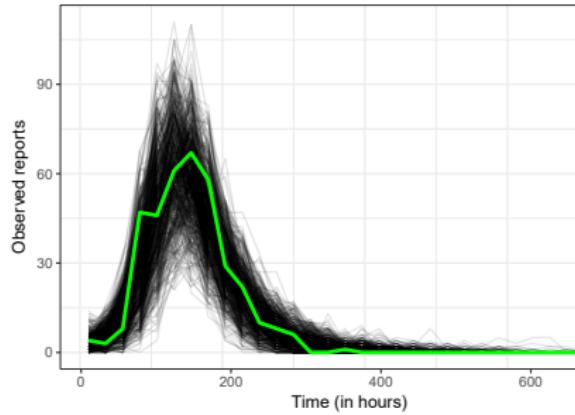
The prior distributions for both parameters were $\text{Gamma}(1,10)$.

Results

Curves simulated with accepted parameters.



Network A: Poisson.



Network B: Polylogarithmic.

Simulated reports from outbreaks with sampled ABC-MCMC parameter values. The green lines correspond to the simulated data reports y .

Results

Point and interval estimation of SIR parameters.

Quantile	A: Poisson		B: Polylogarithmic	
	β	γ	β	γ
2.5 %	0.0250	0.0015	0.0176	0.0008
50 %	0.0333	0.0127	0.0249	0.0085
97.5 %	0.0622	0.0341	0.0411	0.0269

Statistics of posterior ABC samples.

Remember that the real values where $\beta = 0.03$ and $\gamma = 0.01$

Use of a surrogate model

When we have a SIR model with fixed transmission rate β and exponentially(β) distributed infectious period, we propose a surrogate model based on a ODE classical model with modified transmission and removal rates.

Similarly to R_0 , \mathcal{R} describes the expected number of directly new cases originated by a infective case a, but in contrast to R_0 , this assumes a as a secondary case, itself. For network epidemic model it is

$$\mathcal{R} = \pi E(K_1);$$

where π is the probability of transmission between an infective and one of its susceptible neighbors. That is

$$\pi = \int_0^\infty \int_0^\infty (1 - e^{-\beta\ell}) dF_R(\beta) dF_I(\ell) = 1 - \frac{\gamma}{\beta + \gamma} = \frac{\beta}{\beta + \gamma}$$

Use of a surrogate model

The term $E(K_1)$ corresponds to the mean excess degree, that is equal to

$$E(K_1) = \frac{E(K^2)}{E(K)} - 1 = \left(\frac{\text{Var}(K)}{E(K)} + E(K) - 1 \right),$$

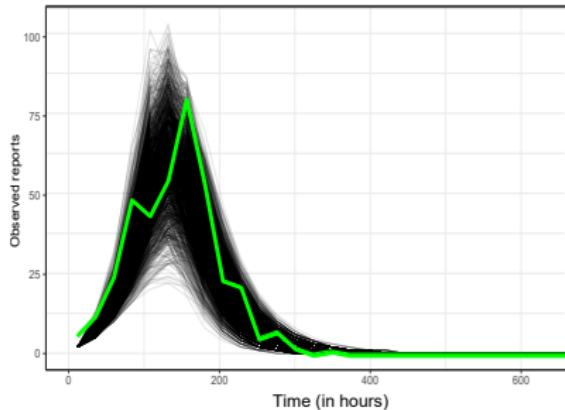
where K is the network degree distribution.

Since the deterministic version of model has basic reproductive number $R_0 = \beta/\gamma$, then with parameter values

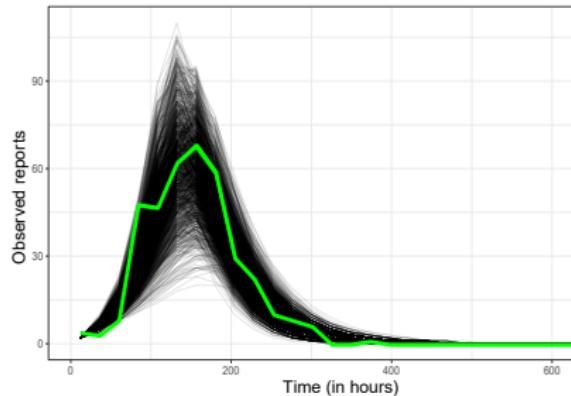
$$\beta_1 = \beta E(K_1) \quad \text{and} \quad \gamma_1 = \beta + \gamma$$

we have $R_0 = \pi E(K_1)$.

Use of a surrogate model



Network A: Poisson.



Network B: Polylogarithmic.

Simulated reports from outbreaks with sampled approximated ABC-MCMC parameter values, using only the surrogate model.

Surrogate: ARIMAX

Let $\mathbf{y} = y_1, \dots, y_m$ be the observed surveillance reports, and $\mathbf{z} = (z_1, \dots, z_r)$ be significant covariates. The considered ARIMAX model is

$$y_t^* = \mathbf{z}\boldsymbol{\beta} + \sum_{i=1}^p \phi_i y_{t-i}^* + \sum_{j=1}^q \psi_j v_{t-j} + v_t, \quad (3)$$

where $y_t^* = (1 - B)^d y_t$, $p, d, q \in \mathbb{Z}^+$, $t = 1, \dots, m$;
 $\phi(\lambda) = 1 - \phi_1\lambda - \dots - \phi_p\lambda^p$; $\psi(\lambda) = 1 + \psi_1\lambda + \dots + \psi_q\lambda^q$, and
 $\phi(\lambda), \psi(\lambda) \neq 0$, $\forall |\lambda| \leq 1$.

The selection of p , d and q was based on Akaike information criterion corrected for small sample sizes (AICc). The remaining coefficients were estimated with least squares.

RNN

RNN.pdf

RNNx

RNNx.pdf

Training Data for ARIMAx and RNNx

We generate 180 networks

- ▶ With orders
 $1 \times 10^3, 5 \times 10^3, 1 \times 10^4, 2.5 \times 10^4, 5 \times 10^4, 1 \times 10^5$
- ▶ As Watts-Strogatz, Barabassi-Albert and with Degree Distributions: Poisson, Polylogarithmic, and Power Law.

For each network we simulate 4 outbreaks (2 with $I \sim \exp$ and 2 $I \sim \text{LogNorm}$). Each outbreak had two initial infectious cases and the parameters were simulated to jointly comply with the influenza transmission rate, infectious period and replacement number, reported in [Cori et al. 2012](#)

Then for training we have a total of **720 outbreaks**.

Covariates

1. Order
2. Size
3. Density
4. Eigen Centrality Variance (VarEigen)
5. Transitivity
6. Assortativity (Degree assortativity)
7. Mean Degree
8. Degree Variance
9. Skewness

Covariates

Results: Incidence observations y

Parameteres:

$$\tau = 0.011, \\ I \sim Exp(\delta = 0.011)$$

Network:

10,000 vertices
Polylogaritmic (0.2, 10)
 $E(K) = 8.80, SD(K) = 8.90,$
 $E(K_1) = 16.73$

Parameteres:

$$\tau = 0.06, \\ I \sim LN(\mu = 4.50, \sigma = 0.15)$$

Network:

50,000 vertices
Watts-Strogatz(1,0.5)
 $E(K) = 2.00, SD(K) = 1.74,$
 $E(K_1) = 1.87$

Results: Accepted curves

Exact

ARIMAx

RNNx

Results

Exponential infectious period: Real parameteres: $\tau = 0.011$,
 $I \sim Exp(\delta = 0.011)$

	Exact		ODE		ARIMAx		RNNx	
	τy	δy						
2.5%	0.0097	0.0005	0.0102	0.0184	0.0098	0.0005	0.0124	0.0105
50%	0.0129	0.0103	0.0126	0.0537	0.0131	0.0111	0.0160	0.0195
97.5%	0.0188	0.0320	0.0158	0.1098	0.0184	0.0383	0.0212	0.0419

LogNormal infectious period: Real parameteres: $\tau = 0.06$,
 $I \sim LN(\mu = 4.50, \sigma = 0.15)$

	Exact			ARIMAx			RNNx		
	τy	μy	σy	τy	μy	σy	τy	μy	σy
2.5%	0.0462	3.3242	0.0133	0.0494	3.5405	0.1609	0.0559	3.0398	0.2019
50%	0.0538	5.4687	0.1028	0.0591	4.5653	0.1940	0.0687	3.7244	0.2903
97.5%	0.0681	8.6026	0.1853	0.0724	6.0432	0.2573	0.0866	4.4340	0.4050

Results: Coverage and Times

References

- ▶ Andersson, H (1998) Limit theorems for a random graph epidemic model. *Ann Appl Probab*, 8(4):1331-1349, 1998.
- ▶ Cohen, R; Havlin, S and Ben-Abraham, D (2003) Efficient immunization strategies for computer networks and populations. *Phys Rev Lett*, 91(24).
- ▶ Del Moral P; Doucet, A and Jasra, A (2012) An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Stat and Comput*, 22(5):1009
- ▶ Marin JM; Pudlo, P; Robert CP and Ryder, RJ (2012). Approximate Bayesian computational methods. *Stat Comput*, 22(6):1167-1180.
- ▶ Newman, MEJ (2002). Spread of epidemic diseases on networks. *Physical Review, E*.
- ▶ Ramirez-Ramirez, LL and Thompson, ME (2014) Probability intervals based on mean and variance of final outbreak size for diseases spreading in networks. *Methodol Comput Appl Probab*, 16(4): 839-862.
- ▶ Wilkinson, RD (2008) Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. arXiv:08113355.