

PRÁCTICA 5**1 Distribución de grado**

La distribución de grado p_k nos da la probabilidad de que tomado un nodo al azar tenga grado k . Si en una red de N nodos, N_k nodos tienen grado k entonces podemos definir p_k :

$$p_k = \frac{N_k}{N}$$

Para la siguiente red de **ejemplo 1** se muestra su distribución de grado (tabla y como histograma):

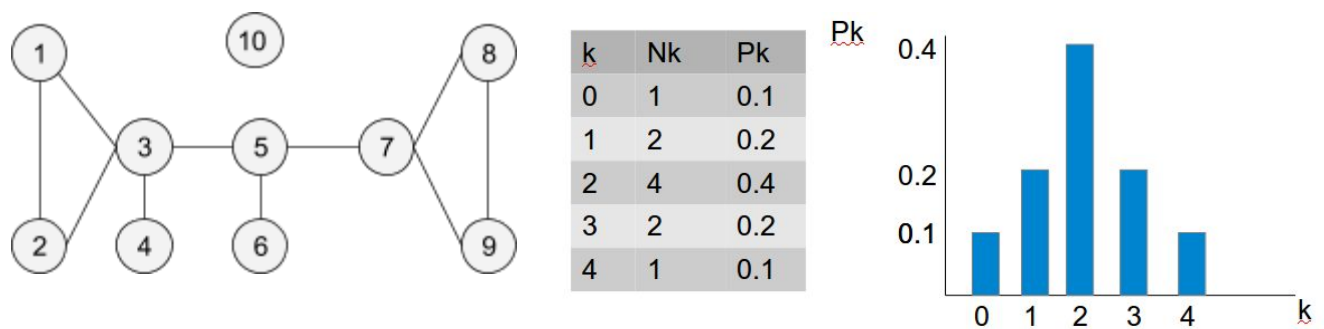


Figura 1. Distribución de grado Ejemplo 1

En el caso de redes dirigidas, se definen dos distribuciones de grado, i.e. in-degree y out-degree.

2 Distribuciones Power Law

El siguiente [documento](#) extraído del libro de Barabasi (2012) introduce el concepto de distribución power law en redes. Lea el documento y trate de contestar a las siguientes preguntas:

2.1 Dadas las redes de la figura, ¿cuál de ellas presenta una propiedad scale-free?

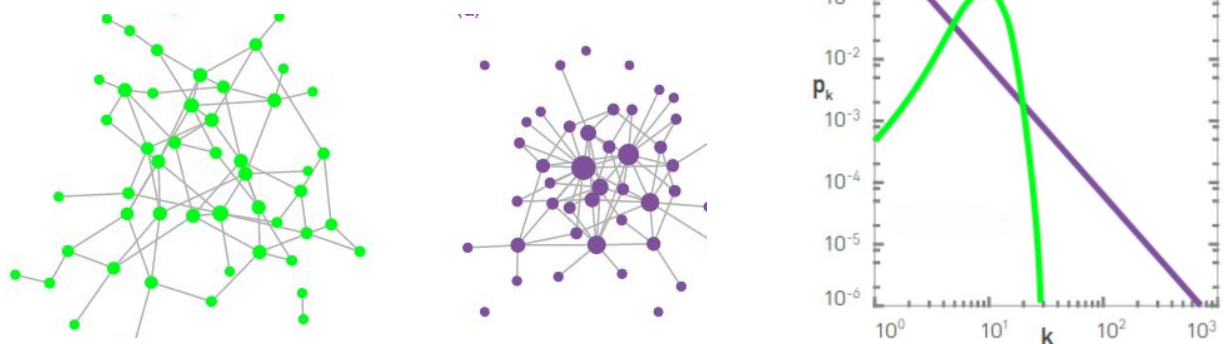


Figura 2. Distribuciones de grado de dos redes

2.2 De los siguientes mecanismos ¿cuál de ellos podría generar propiedad scale-free con más probabilidad?

1. Un usuario crea una página web y añade un enlace a la página de un amigo.
2. Un usuario crea una página web y añade un enlace a la página que le proporciona el primer resultado de una búsqueda en Google

3 Estudio de la distribución de grado de una red

A continuación vamos a estudiar la distribución de grado de una [red de routers de Internet](#) mediante la librería Networkx de python. La librería Networkx no es apropiada para dibujar redes muy grandes como ésta, por lo tanto **no intente visualizar la red completa**.

El objetivo es comprobar que la distribución de grado de la red responde a una power law y tratar de encontrar el parámetro que la define. Veremos dos técnicas distintas.

3.1 Estadísticas

Calcule las siguientes estadísticas de la red:

- Número de nodos
- Número de enlaces
- Grado medio
- Grado máximo
- Grado mínimo

3.2 Histograma

Dibuje un histograma de la distribución de grado de la red. Dada una red, la probabilidad de encontrar un nodo con grado k se define como $p_k = N_k/N$ (siendo N_k el número de nodos con grado k). Le será útil la función:

[`matplotlib.pyplot.hist\(x, bins, normed\)`](#)

Cambie la escala de los ejes de lineal a logarítmica (ver funciones

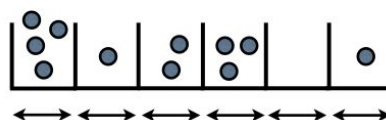
[`matplotlib.pyplot.yscale`](#)

y [`xscale`](#)) y decida con qué escala se visualiza mejor la distribución de grado de esta red.

Si la distribución muestra un significativo comportamiento lineal cuando la representamos en escala logarítmica, es probable que sea una distribución de power law.

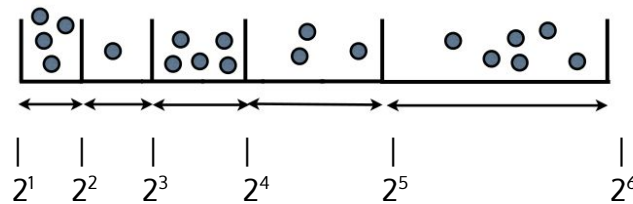
3.3 Histograma con clase variable logarítmica

El histograma anterior supone un tamaño de clase constante:



Pero si la distribución de grado es power law, utilizar clases constantes puede producir ruido puesto que algunas de ellas, por ejemplo las de mayor grado, tienen muy pocos datos.

Para resolver este problema podemos utilizar un tamaño de clase variable. Por ejemplo, una **clase variable logarítmica**:



tal que el tamaño de la n -ésima clase sea:

$$2^{n-1} < k < 2^n$$

Debemos normalizar la frecuencia del número de nodos que caen en cada clase dividiéndola por el tamaño de la clase correspondiente, que para la clase n -ésima sería:

$$2^n - 2^{n-1}$$

Dibuje un nuevo histograma con clase variable logarítmica. En vez de utilizar la función `matplotlib.hist` que calcula y representa gráficamente un histograma, puede utilizar la función:

[`numpy.histogram\(a, bins, normed\)`](#)

que calcula el número de datos $[n_1, n_2, n_3, \dots]$ que caen dentro de las clases definidas en la variable `bins`. Por ejemplo, para `bins=[21, 22, 23, 24, 25, ...]` n_1 define la frecuencia de valores contenida en el rango $[2^1, 2^2)$, n_2 define la frecuencia de valores contenida en el rango $[2^2, 2^3)$, y así sucesivamente. Después se puede dibujar un diagrama de barras para representar el histograma mediante la función:

[`matplotlib.pyplot.bar\(left, height, width\)`](#)

donde podemos utilizar anchos para las barras variables mediante el argumento `width` para representar las clases variables logarítmicas.

3.4 Estimación parámetro de power law

Nos interesa calcular el parámetro de power law. Es decir, suponemos que la distribución de grado responde a la siguiente función:

$$p_k \sim k^{-\alpha}$$

donde el parámetro α define la power law. La forma más sencilla es ajustar un modelo de regresión lineal a los datos (**transformados logarítmicamente**), el parámetro de pendiente corresponderá al parámetro α de la power law:

$$\ln p_k = -\alpha \ln k + c$$

Para ello podemos emplear la función:

[`scipy.stats.linregress\(x, y\)`](#)

que calcula un modelo de regresión lineal, y nos devuelve los parámetros de la recta estimada, así como algunos estadísticos como el R^2 o el p-valor del contraste de

hipótesis de pendiente nula.

Estime el coeficiente α de la distribución de potencia (power law) que mejor se ajusta a la distribución de grado de la red, y muestre el valor de R^2 del ajuste y el p-valor.

3.5 Estimación mediante distribución acumulada

Otro método para calcular el parámetro de power law utiliza la distribución acumulada (Newman, 2010¹). Se define la función de distribución de grado acumulada como:

$$P_k = \sum_{k'=k}^{\infty} p_{k'}$$

Dada una red, P_k se puede calcular como la suma de todos los nodos con grado mayor o igual a k , dividida por el número de nodos N . Una forma muy sencilla de obtener esta distribución es la siguiente, que no requiere calcular el tamaño de las clases de un histograma. Por ejemplo, para la red de ejemplo 1:

1. ordenamos de forma decreciente los grados de todos los nodos,
2. añadimos una columna de ranking del nodo en dicha ordenación,
3. y finalmente dividimos cada valor de ranking por el número total de nodos:

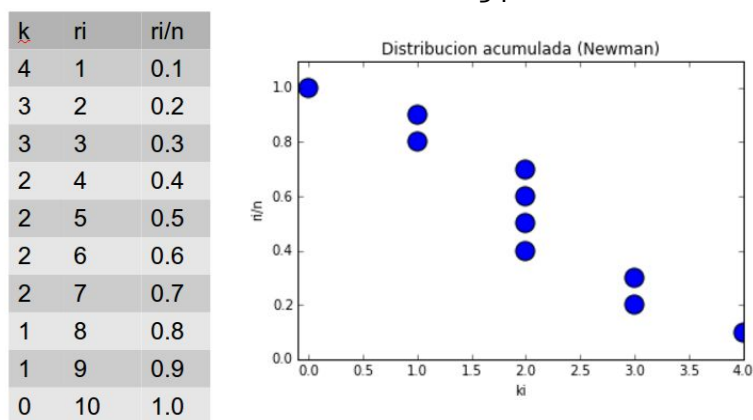


Figura 3. Distribución de grado acumulada Ejemplo 1

Para la red de routers, podemos ajustar un modelo de regresión lineal a los datos (transformados logarítmicamente). En este caso el parámetro de pendiente del modelo está relacionado con el parámetro alfa de power law de la siguiente forma:

$$\text{pendiente} = -\alpha + 1$$

Ya que al ser la función acumulada de una ley de potencia, el exponente responde a la fórmula:

$$P_k \sim k^{-\alpha+1}$$

Estime el coeficiente α de la distribución de potencia (power law) de acuerdo a este método, y muestre el valor de R^2 del ajuste. Compare los resultados con los valores obtenidos anteriormente.

¹ Newman, M. (2010). Networks: An Introduction 1st ed., Oxford University Press (Chapter 8)

3.6 Optativo

Implemente una función python que tenga como argumento un vector con el grado de los nodos de una red, y devuelva el parámetro α de la distribución de potencia que mejor se ajusta a la distribución de grado, el valor de R^2 , y el p-valor del ajuste.

Ayuda

- Repaso ipython y matplotlib [notebook](#)