Optimización de Flujo en Redes: Métricas para caracterizar la topología de una red

Alcantar Gómez Alan Arnoldo

23 de abril de 2018

1. Introducción

El presente trabajo tiene la finalidad de introducir algunas de las métricas utilizadas para caracterizar la topología de una red, por mencionar, la ***distancia promedio mínima entre cualquier par de nodos*** y el ***coeficiente de agrupamiento***. Para la generación de los grafos se utilizo *Python* y *Gnuplot* para graficar los resultados.

2. Teoría

La distancia promedio mínima entre cualquier par de nodo se conoce como longitud de camino promedio, es una de las métricas más usadas para cuantificar la topología de una red [1]. Es definida como el numero promedio de aristas que deben ser recorridas en el camino más corto $d_{i,j}$ entre cualquier par de nodos i y j, y se expresa como:

$$L = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} d_{i,j} \tag{1}$$

El coeficiente de agrupamiento describe la probabilidad de que cualquier nodo j en el vecindario, A_i, del nodo i, también sea adyacente a los otros nodos en A_i [1]. Para escribir una expresión para el coeficiente de agrupamiento local, la notación $(e_{j,h})_i$ para denotar el conjunto de todas las aristas existentes en una red, entre los nodos en A_i y k_i es el número de nodos vecinos del nodo i, de modo que la expresión para el coeficiente de agrupamiento del nodos i es:

$$C_i = \frac{2(e_{j,h})_i}{k_i(k_i-1)} \tag{2}$$

De modo que el coeficiente de agrupamiento local es dado por:

$$\bar{C} = \frac{1}{N} \sum_{i=1}^{n} c_i \tag{3}$$

3. Descripción del algoritmos

Para generar un grafo circular es necesario conectar uno de los nodos que formar el grafo circular con sus siguientes k nodos y de forma random se agregar aristas entre cualquier par de nodos

```
for p in range(nodos):
   for j in range(1,k+1):
      if ((p,p+j)) not in self.aristas:
         self.aristas[(p,(p+j)%n)] = 1

for i in range(n-1):
   for j in range(i+1,n):
      if random() >= q:
   ‘     if ((i,j)) not in self.aristas:
            self.aristas[(i,j)]=1
```

Para calcular el coeficiente de agrupamiento global , primero se selecciona un vértice y se obtiene su número de vecinos (t), después cada que un vecino de sus vecinos se encuentre entre sus vecinos se suma 1 a la variable *con*, después de recorrer todos los vecinos del vértice se divide sobre el número máximo de posibles conexiones que pudiera haber entre los vecinos de ese vértice y se suma a la variable *clu* y se repite el proceso con cada uno de los vértices y al final la variable *clu* se divide sobre el número de nodos.

```
def cluster(self,n):
   clu = 0
   for i in self.vertices:
      t=len(self.vecinos[i])
      con=0
      for j in self.vecinos[i]:
         for k in self.vecinos[j]:
            if k in self.vecinos[i]:
               con=con+1
      clu = clu + con/(t*(t-1))
   return clu/n
```

Para obtener la longitud de camino promedio, primero se utiliza el algoritmo de Floyd-Warshall [2] para obtener las distancias mínimas entre cualquier par de nodos del grafo, después se suman todas estas distancias mínimas y se divide sobre el número máximo de pares de nodos que puede haber en el grafo.

```
for (i,j) in d:
   sdm=sdm+d[(i,j)] #sumando las distancias minimas entre todos los
        pares de nodos
adm = sdm/(n*(n-1))
```

4. Experimentación y resultados

Es importante mencionar que las aristas presentes en los grafos generados tienen un peso de uno, esto para que la distancia mínima entre dos nodos sea igual al número de aristas que es necesario cruzar para llegar de un nodo al otro.

En la figura 4 se observa un ejemplo de un grafo generado con 20 nodos, $k = 4$ y una probabilidad de 0.95, en ella se puede ver como las aristas rojas (obligatorias) por el parámetro k hacen que se genere un patrón en forma de circulo, mientras que las aristas azules representan aquellas aristas que se agregaron de forma probabilística y que debido a que la probabilidad para este ejemplo es muy grande solo se pudieron añadir pocas aristas

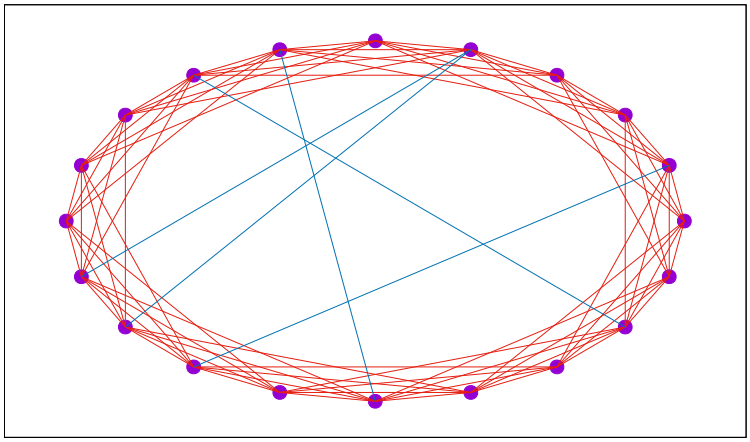

Figura 1: Grafo simple ponderado con 10 vértices.

En la figura 4 se presentan los diagramas de caja para los tiempos de ejecución de Floyd-Warshall y una curva de ajuste a los promedio de los tiempos de ejecución, de la forma $ax^3 + bx^2 + cx + d$. se puede ver además que la dispersión de los tiempos sugiere que los tiempos de ejecución de Floyd-Warshall están normalmente distribuidos.

En la figura 4 se presenta como varia la longitud de camino promedio y el coeficiente de agrupamiento con la probabilidad de añadir nuevas aristas. Primero analizando el coeficiente de agrupamiento se entiende que cuando la probabilidad es muy pequeña se agregan muchas aristas al grafo por ello cada nodo tiende a tener aristas con casi todos los otros nodos generando un valor cercano a uno para el coeficiente de agrupamiento, por el caso contrario cuando la probabilidad es cercana a uno, casi no se agregan aristas nuevas y el coeficiente de agrupamiento tiende a cero. Analizando la longitud de camino promedio se

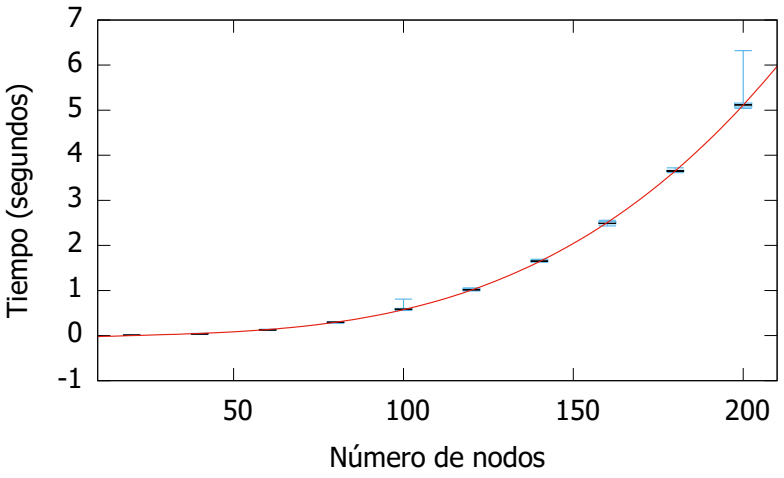

Figura 2: Grafo simple ponderado con 10 vértices.

entiende que cuando se agregan muchas aristas debido a un valor cercano a cero, son necesarias menos aristas para ir de un nodo a cualquier otro, pero cuando se agregan pocas aristas debido a una probabilidad cercana a uno, es necesario recorrer una mayor cantidad de aristas para poder llegar de un nodo a cualquier otro nodo. Entonces tenemos que la longitud de camino promedio aumenta a medida que aumenta la probabilidad.

5. Conclusión

Para poder normalizar la longitud de camino promedio respecto a la probabilidad, es necesario varias cosas, la primera es que a mi juicio es necesario involucrar la probabilidad y la segunda es que el factor de normalización debe ser no lineal y creciente para que en valor pequeños de probabilidad la longitud de camino promedio este cercano a uno y con valores grandes de probabilidad la longitud de camino promedio tienda a cero.

Referencias

[1] Alcantar G. Alan, *implementación de los algoritmos Ford-Fulkerson y Floyd-Warshall*, https://github.com/alan-arnoldo-alcantar/flujos/blob/master/Reporte

[2] J Prettejohn, Brenton and J Berryman, Matthew and Mcdonnell, Mark, 2011, *Methods for Generating Complex Networks with Selected Structural Properties for Simulations: A Review and Tutorial for Neuroscientists*, Frontiers in computational neuroscience

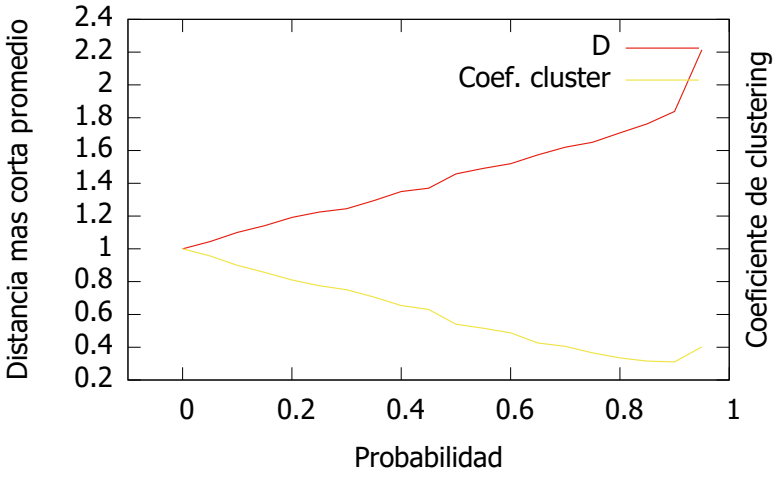

Figura 3: Grafo simple ponderado con 10 vértices.