ACTA UNIVERSITARIA
Multidisciplinary Scientific Journal

ISSN 0188-6266

doi: 10.15174/au.2015.733

Algoritmo computacional de mezcla dinámica de cliqués para medir la resonancia de individuos en redes sociales

Computational algorithm dynamic merge of cliques to measure the resonance of individuals in social networks

Jorge Esteban Zaragoza Salazar*, Adrián Trueba Espinosa*

RESUMEN

Considerando la facilidad que tienen las redes sociales para reprodcir información, la cual puede ser viral en cuestión de horas, provocando un efecto nocivo o favorable en la sociedad, el presente artículo aborda la propagación de información en redes sociales. Para medir la propagación de la información se proponen métricas para medir la resonancia (centralidad) de un individuo en las redes sociales, utilizando algoritmos y modelos matemáticos. Destacaremos el algoritmo de *Dynamic Time Warping* (DTW, por sus siglas en inglés) en los resultados obtenidos, con *random walk*, para simular cómo se difundirá un mensaje en un grafo compuesto por nodos (personas).

ABSTRACT

This article discusses the role social networks have in spreading information. This information can become viral in a matter of hours and could have harmful or beneficial effects on society. In order to measure the spread of information, we propose metrics to measure the resonance (centrality) of an individual in social networks, using algorithms and mathematical models. Highlighting the Dynamic Time Warping (DTW) algorithm on the results obtained with random walk, to simulate a broadcast message as a graph consisting of nodes (people).

Recibido: 19 de enero de 2015
Aceptado: 27 de abril de 2015

Palabras clave:
Cliqué; grafo; red social; centralidad; influencia; fusionar; paseo aleatorio; distorsión temporal dinámica.

Keywords:
Clique; graphs; social network; centrality; influence; merge; random walk; Dynamic Time Warping.

Cómo citar:
Zaragoza Salazar, J. E. & Trueba Espinosa, A. (2015). Algoritmo computacional de mezcla dinámica de cliqués para medir la resonancia de individuos en redes sociales. *Acta Universitaria*, *25*(2), 28-39. doi: 10.15174/au.2015.733

INTRODUCIÓN

El uso de las redes sociales se encuentra en un crecimiento exponencial, por lo tanto, aumenta la propagación de mensajes de forma individual y grupal. La información generada en las redes sociales es asimilada y replicada por una cantidad elevada de personas, donde cada una de ellas tiene influencia sobre otros individuos con quienes se está relacionado. Dicha influencia se conoce como *resonancia*, es decir, cuando un mensaje (información, imágenes, memes o videos) es generado por un individuo con alta capacidad de respuesta, y puede tener un alcance muy elevado y ser asimilado por una gran cantidad de personas (Wei *et al.*, 2012). En la resonancia, el número de enlaces sociales (amistades) de los individuos no afecta la influencia que éstos tienen en las redes sociales. La resonancia e influencia de un nodo (personas) no está totalmente ligada a la cantidad de conexiones que posea, pues está influenciada por otros factores, que pueden ser la capacidad intelectual que el individuo tenga, el liderazgo que posea, el tipo de contenido que genere o incluso el aspecto fisico del individuo, según sea el caso.

* Maestría en Ciencias de la Computación, Campus Texcoco, Universidad Autónoma del Estado de México. Km. 8.5 Carretera Texcoco – Los Reyes La Paz, Av. Jardín Zumpango s/n Fracc. El Tejocote Texcoco – Los Reyes la Paz, Edo. de Méx. Tel.: (01 595) 9-21-04-48; 9-21-12-47; 9-21-03-68. Correos electrónicos: ing.jorge.zaragoza@live.com; atruebae@hotmail.com

La resonancia en las redes sociales brinda a individuos y grupos de personas la capacidad de generar mensajes virales, que pueden provocar diferentes efectos, como pánico colectivo, histeria, descontento social, publicidad, mercadotecnia y propagación de opiniones (Nekovee, Moreno, Bianconi & Marsili, 2007). Las comunidades en las redes sociales pueden alterar diversos aspectos del comportamiento de los individuos, entre los cuales se incluye la resonancia, y de la modularidad de características de éstos.

Se han propuesto varios algoritmos y métricas para calcular la resonancia de individuos, por ejemplo, Zheng, Chen, Zhang & Bu (2010) utilizaron métricas para medir la contribución de integrantes de una red social. Para medir la propagación de información, rastrearon la difusión de una fotografía en *Flickr*, concluyendo que la difusión de la mayoría de información es creada y propagada sólo por una cantidad pequeña de individuos clave. Ellos tienen características específicas, tales como el tipo de contenido que crean, el comportamiento ante las demás personas de la red que hacen que pueden ser considerados como individuos con una alta resonancia. En otro trabajo publicado por Khrabrov & Cybenko (2010) se presenta un conjunto de técnicas para rastrear individuos que tienen influencia en una red social y cómo la obtuvieron; se enfocan en la interacción social sobre *Twitter*, utilizando una escala para medir la influencia de los individuos, calculada a partir de las menciones diarias de los usuarios.

En este trabajo se considera que la resonancia es un factor importante, sin embargo, cuando se mide de forma individual el impacto no se vislumbra claramente; en contraste, cuando se agrupan individuos con características comunes la resonancia es completamente evidente y se puede ver el impacto en las redes sociales con mayor claridad.

Considerando los elementos mencionados, se plantea que es posible analizar y simular la resonancia de comunidades en las redes sociales para generar conjuntos de individuos agrupados de manera dinámica, que pueden reflejar una resonancia más elevada que de forma individual.

Hay algoritmos para detectar comunidades en redes sociales. Algunos consideran redes con características de unipartición, bipartición, ponderadas, no ponderadas, entre otras. También se considera el tipo de estructura que pueden detectar: disjuntas, superpuestas, jerárquica y otras. (Fortunato, 2010). Sin embargo, la mayoría de algoritmos se centran en detectar comunidades disjuntas, las cuales el presente artículo tratará.

En este artículo se utiliza el algoritmo de *Dynamic Time Warping* (DTW) para la mezcla dinámica de comunidades (cliqués), basado en la modularidad de características de los nodos de la red. Dicho algoritmo consiste en una técnica originada de la problemática de encontrar una alineación óptima entre dos secuencias de tiempo dadas bajo ciertas restricciones. Dichas secuencias son alineadas de manera no lineal con la finalidad de coincidir entre sí. Originalmente, DTW se ha aplicado para comparar diversos patrones del habla en el reconocimiento de voz automatizado. Sin embargo, en campos como la minería de datos y recuperación de información ha sido aplicado de manera exitosa con el objeto de hacer frente a deformaciones de tiempo y velocidades asociadas con los datos dependientes del tiempo (Müller, 2007). Por ende, dicho algoritmo es utilizado en el presente trabajo para hallar nodos similares. Asimismo, se utiliza el algoritmo de caminatas aleatorias (*Random Walks*) para simular la propagación de información con grafos sociales; dicho algoritmo posee una adopción alta en procesos que requieren trayectorias que resultan de realizar pasos aleatorios sucesivos, tales como la genética de poblaciones, movimiento browniano, movimiento de moléculas de líquidos y gases, procesos de biodifusion y dinámica de poblaciones, en procesamiento de imágenes o para determinar etiquetas asociadas a cada pixel. Además, *Random Walks* es utilizado para representar la propagación irregular de información. Asimismo, se emplea el algoritmo para la maximización de influencia, con métricas.

Trabajos relacionados

Los autores (Zheng *et al.*, 2010) proponen una serie de métricas basadas en características de los usuarios a partir de su comportamiento y sus amistades, mediante un modelo de seguimiento de navegación en las redes sociales. Dicho trabajo muestra que sólo una pequeña parte de los miembros de las redes sociales tienen una contribución relevante en la propagación de información, y que la mayoría de la información que circula es proporcionada por sólo una cantidad pequeña de estos miembros. Por otro lado, los resultados del trabajo presentado por Junquero-Trabado, Trench-Ribes, Aguila-Lorente & Dominguez-Sal (2011) ofrecen una comparación de métricas basadas en información del usuario en diferentes redes sociales, como *Twitter*, *Enron* y *Scopus*, llegando a la misma conclusión que el trabajo anterior, con variaciones muy pequeñas entre

ISSN 0188-6266

las diferentes redes sociales. También el trabajo de investigación de Khrabrov & Cybenko (2010) propone métricas para descubrir la influencia de los usuarios en las redes sociales, utilizando análisis dinámico de grafos, con en el cual monitorearon y recolectaron datos de *Twitter* en un año, obteniendo una cifra de cuatro a cinco millones de *tweets*. A partir de dicha recolección analizaron el número de menciones por usuario, así como la cantidad de réplicas por mensaje diarias de cada usuario, para generar un número entero a partir de éstos (*Drank*) y un histograma de la influencia ejercida por los usuarios con mayor *Drank*. Ello da como resultado mediciones muy certeras sobre el crecimiento de influencia de cada usuario por día; asimismo, detecta los momentos más grandes de influencia y menciones. Relacionado con esto, Durr, Protschky & Linnhoff-Popien (2012) elaboraron un artículo con la finalidad de modelar las interacción en grafos de redes sociales y redes P2P (*Peer to Peer*), utilizando un componente estático para modelar las relaciones y los efectos por usuario, el cual consiste en *datasets* de nodos y aristas para formar un grafo, además de utilizar un componente dinámico para modelar la interacción entre los usuarios, basados en información de retroalimentación, como mensajes generados (publicaciones), menciones y comentarios de fotos. Ello aplicando técnicas de *clustering* de Newman para determinar la probabilidad de selección de nodos y aristas, así como la influencia de la comunidad que obtuvo mejores resultados para simular la actividad de una red social en una red P2P. Por otro lado, Guo (2012) describe una simulación que explica cómo se realiza la propagación de información en las redes sociales, evaluando un modelo clásico utilizado en epidemiología para determinar la influencia de los nodos de una red social. El modelo es empleado para describir la cura de pacientes que no pueden obtener inmunidad a ciertas enfermedades, a través de probabilidad. Los resultados obtenidos fueron eficaces para medir la influencia de un nodo y simular la propagación y viralidad de nodos de una red.

En su trabajo, Belo & Ferreira (2012) explican cómo usan aleatorización para determinar la influencia en redes sociales móviles. Los métodos de aleatorización utilizados consisten en generar pseudo-muestras a partir de los datos originales recolectados, con el objetivo de selectivamente permutar valores de algunos atributos bajo observación y estimar distribuciones empíricas. Los datos originales recolectados son alterados muchas veces permutando algunos atributos, cada uno generando una pseudo-muestra a la cual se aplica un modelo de probabilidad lineal, para determinar su rol de influencia. Los resultados sugieren que al utilizar la aleatorización se contribuye a hacer eficiente determinar las pseudo-muestras con mayor influencia en las redes sociales móviles. En otro trabajo, Liu & Chen (2011) detallan una simulación de propagación de rumores en *Twitter*, describiendo las relaciones de los usuarios como un grafo dirigido y desarrollando un modelo de dos tipos de dichas relaciones basado en el número de seguidores de nodos. Así como un mecanismo de creencia de los nodos basado en aleatorización, que transfiere la atención de nodo a nodo, para determinar los usuarios que serán creyentes de dicho rumor y lo propagarán.

En otro contexto, existen muchos trabajos para generar comunidades de cliqués con grafos a partir de las redes sociales de gran escala, los cuales se basan en diferentes formas para medir la modularidad de los usuarios; uno de esos trabajos es el presentado por Yan & Gregory (2009), donde propone el algoritmo *CliqueMod* para detectar comunidades disjuntas en grafos de redes sociales, empleando algoritmos de escalación de colina para maximizar la modularidad de las particiones de grupos. Dicho trabajo obtiene una modularidad alta y tiempos de ejecución bajos para la generación de comunidades, logrando generar grafos complejos. De igual manera, Ilyas & Radha (2011) realizaron un trabajo de investigación donde explican cómo detectar vecindarios de nodos influenciales en redes sociales masivas, tales como *Orkut* de *Google* y *Facebook*, aplicando componente principal de centralidad (PCC), el cual se determina un valor de centralidad de cada uno de los nodos, utilizando la herramienta *Eigen vector Centrality*, usada para determinar centralidad de nodos en grafos.

METODOLOGÍA

Recolección de datos

Se recolectó la información de los nombres, apellidos paternos, maternos, edades, regiones, idiomas, nombre de usuarios de *Twitter* de cada nodo. Los campos *región*, *idioma* y *edad* son recabados para poder realizar posteriormente la creación de cliqués de nodos similares. Mientras que los campos de *nombre*, *apellidos*, *edad* y *usuario* son recabados para la distinción e identificación de los nodos dentro del grafo de nodos no agrupados.

El algoritmo de recolección se implementó en lenguaje *JAVA*. Este algoritmo recabó 8200 nodos, así como 20 100 relaciones. Debido a limitaciones del *Application Programming Interface* (API, por sus siglas en inglés) de *Twitter*, que no permite a las aplicaciones personales realizar más de 350 peticiones por hora, la aplicación cuenta cada una de las peticiones que lleva realizadas, para que una vez alcanzado el límite el hilo del proceso de recolección "duerma" 30 minutos y prosiga con la recolección al pasar este tiempo, iterando

este proceso hasta que termine de obtener todos los datos de los nodos, así como de sus relaciones en el espacio de búsqueda, determinado por una profundidad establecida. Dicha recolección de datos se realiza mediante el siguiente algoritmo:

```
Funcion Recolectar()
NodoRaiz=”cuentaRaiz”
ProfundidadABuscar=3
ContadorPeticiones=0
ListaSeguidoresCapa(0)=Twitter.
ObtenerSeguidores(nodoRaiz)
Incrementar(ContadorPeticiones)
Para i=0 hasta profundidad
ParaCada nodo en ListaSeguidoresCapa(i)
    Si nodo no existe en BaseDeDatos
BaseDeDatos.GuardarNodo(nodo.nombre)
BaseDeDatos.GuardarNodo(nodo.edad)
BaseDeDatos.GuardarNodo(nodo.apellidoPaterno)
BaseDeDatos.GuardarNodo(nodo.apellidoMaterno)
BaseDeDatos.GuardarNodo(nodo.Region)
BaseDeDatos.GuardarNodo(nodo.apellidoPaterno)
BaseDeDatos.GuardarNodo(nodo.usuarioTwitter)
    Termina Si
ListaSeguidoresCapa(i+1).Agregar(Twitter.
ObtenerSeguidores(nodo));
      Incrementar(ContadorPeticiones)
BaseDeDatos.GuardarRelaciones(nodoPadre,
nodoActual)
  Termina ParaCada
Termina Para
Termina FuncionFuncion
Incrementar(ContadorPeticiones)
   Si ContadorPeticiones=300 Entonces
DormirHilo(30 minutos)
ContadorPeticiones=0
   Si no
ContadorPeticiones=ContadorPeticiones+7
   Termina Si
  Regresa ContadorPeticiones
Termina Funcion
```

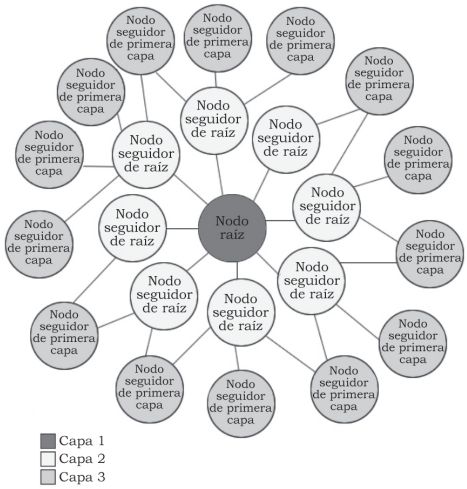

Figura 1. Capas de profundidad.
Fuente: Elaboración propia.

El algoritmo forma capas de profundidad de recolección de nodos, de las cuales la primera está formada únicamente por un nodo raíz, desde el cual la recolección inicia obteniendo una lista de cuentas que son seguidores de dicho nodo raíz. Esta lista obtenida a partir del nodo raíz forma la segunda capa de nodos, a los cuales se les aplica a cada uno el mismo procedimiento, para formar *n* capas de profundidad; con este recorrido se obtiene que todos los nodos son almacenados en la base de datos con su relación de padre (seguidor) e hijo (seguido). Si los nodos ya existen en la base de datos, únicamente se guarda la relación.

La figura 1 muestra las capas de profundidad de recolección, empleadas únicamente para delimitar el espacio de búsqueda; dicho número de capa no tiene relevancia en el almacenamiento ni el procesamiento de datos.

Representación del grafo

Los grafos obtenidos a partir de la recolección son exportados en forma de matriz de adyacencia compacta con nombre, edad, región, idioma y apellidos, para posteriormente ser utilizada en el proceso de simulación. El archivo consta de dos partes: la primera consiste en la información de los nodos y finaliza con un 0 para indicar el término de la sección de datos; la segunda sección pertenece a las relaciones (aristas), y está conformada con un renglón que muestra el número total de aristas y en los renglones posteriores el identificador enumerado del nodo al que apunta la relación, como en el siguiente formato:

ISSN 0188-6266

```
3
214328887,Jorge Esteban Zaragoza Salazar,H,Español,MX,25,2
221036078,Daniel Espinosa Hernández,H,Español,MX,42,1
437804658,Guadalupe Ortiz Mendieta,M,Inglés,US,16,1
0
4
1 221036078
2 437804658
3 437804658
4 214328887
```

Lo anterior representaría un grafo como el mostrado en la figura 2.

La figura 2 muestra de forma visual el grafo generado a partir de la matriz de adyacencia de ejemplo, que podemos observar tres nodos con sus cuatro respectivas relaciones, enumeradas como el archivo descriptor lo declara.

La implementación de matriz de adyacencia compacta permite utilizar pesos, empleados como probabilidad de propagación, además de la información de las características de los nodos del grafo. Todos estos datos son usados para el proceso de simulación. La matriz de adyacencia compacta se representa con múltiples listas, de las cuales la primera es donde se encuentra la información de nodos o vértices, representada como *Va*; asimismo, cada uno de los nodos posee el apuntador del punto inicial de su lista de adyacencia en la matriz de aristas, representada como *Aa*, como se puede observar en la figura 3.

Se optó por la representación del grafo como en la figura 3, debido a que un grafo almacenado de esta forma utiliza considerablemente menor espacio en memoria que una matriz de adyacencia no compacta. Del mismo modo, esta representación mejora el procesamiento en paralelo, en razón a que los datos de la matriz de adyacencia pueden ser leídos de forma simultánea por múltiples hilos, abriendo la posibilidad de trabajos futuros en procesamiento paralelo.

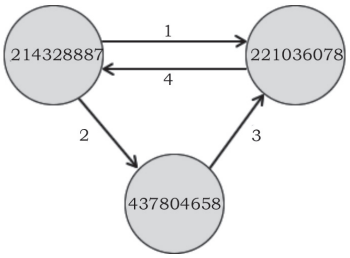

Figura 2. Grafo representado por archivo de matriz de adyacencia.
Fuente: Elaboración propia.

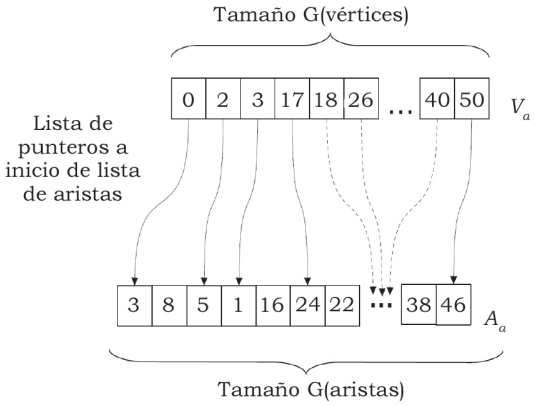

Figura 3. Matriz de adyacencia compacta.
Fuente: Elaboración propia.

Concepción de cliqués

A partir de la representación en memoria del grafo se toma un nodo inicial, así como las relaciones que posee. De estas relaciones, dos nodos con sus características son seleccionados aleatoriamente para generar una matriz numérica en memoria; un ejemplo de esta matriz sería el siguiente:

```
Nodo (SEXO,IDIOMA,REGION,EDAD)=X=[1,12,20,25]
```

Donde las características no numéricas son representadas mediante un índice generado en la base de datos, mediante el agrupamiento por conteo descendiente (*Group by*) de la característica concreta.

Una vez obtenidas las matrices numéricas representativas de las características de los nodos, las matrices son procesadas con el algoritmo de acumulación, el cual almacena el costo y distancia de las matrices enviadas al algoritmo DTW.

El algoritmo de acumulación es el siguiente:

```
Funcion acumularCostoMatrizDTW(x,y,costo)
n=x
m=y
dtw[]=matriz(nxm)
dtw[0,0]=0
para i=1hasta n
  dtw(i,1)=dtw(i-1,1)+c(i,1)
termina para
para j=1hasta m
  dtw(1,m)=dtw(1,j-1)+c(1,j)
termina para
para i=1 hasta n
  para j=1 hasta m
dtw(i,j)=c(i,j)+minimo{dtw(i-1,j);dtw(i,j-1);
dtw(i-1,j-1)}
termina para
termina para
termina función
```

Mientras que el algoritmo para encontrar la distancia (similitud) entre las características de dos nodos es el siguiente:

```
función dtwOptimo(dtw)
ruta[]
i=columnas(dtw)
mientras(i>1) y (j>1)
  si i=1 entonces
     j=j-1
  pero si j=1 entonces
    i=i-1
  si no
     si dtw(i-1,j)=minimo{dtw(i-1,j); dtw(i,j-1);
     dtw(i-1,j-1)} entonces
         i=i-1
     pero si dtw(i,j-1)=minimo{dtw(i-1,j);dtw(i,j-1);
     dtw(i-1,j-1) entonces
         j=j-1
     si no
       i=i-1;j=j-1
      termina si
termina mientras
termina función
```

Donde X es la primera matriz de características, y la segunda matriz de características y costo es un arreglo donde se acumulan los costos de DTW. El objetivo de DTW es comparar dos secuencias dependientes del tiempo $T:=(t1, tx2,..., tN)$, de tamaño N y $R:=(r1, r2,...,rM)$, de tamaño M; estas secuencias pueden ser series de tiempo discretas.

Un alineamiento óptimo entre las secuencias de longitud variable $T: = \{t1,..., tN\}$ y $R: = \{r1,..., rM\}$ y la distorsión total $D(T, R)$ se basa en una suma de distancias locales entre elementos $d(t\,i, r\,j)$.

Una distorsión de alineamiento especial φ, alinea T y R mediante un mapeo de punto a punto.

$\varphi = (\varphi t, \varphi r)$, de longitud $K\varphi$:

$$t\phi_t(k) \Longleftrightarrow r\phi_r(k)\ 1 \le k \le k\phi \qquad (1)$$

El alineamiento óptimo minimiza la distorsión global (Huang, Acero & Hon, 2001).

$$D(\boldsymbol{T}, \boldsymbol{R}) = min\ D\phi(\boldsymbol{T}, \boldsymbol{R})$$

$$D\phi(\boldsymbol{T}, \boldsymbol{R}) = \frac{1}{M\phi}\sum_{k=1}^{k\phi} d(\boldsymbol{t}\phi_t(K), \boldsymbol{r}\phi_r(k))mk \qquad (2)$$

Simulación

Se realizó la implementación del algoritmo de caminatas aleatorias en lenguje C, el cual a partir de la matriz de adyacencia compacta cargada en memoria se encarga de realizar la simulación de propagación de información.

El algoritmo simula la generación y propagación de un mensaje viral, que parte de un cliqué inicial, del cual se obtiene una lista en memoria que contiene los cliqués seguidores de dicho cliqué inicial. Cada seguidor se recorre mediante el algoritmo *Breadth First Search* (BFS) paralelo, empleado para recorrer el grafo mediante anchura, en conjunto con una caminata aleatoria y una probabilidad de propagación como parámetro de aleatorización del algoritmo; dicho parámetro determina si se realiza la propagación o no de un nodo. Este proceso es realizado iterativamente con todos los cliqués obtenidos en la fase anterior, como se muestra en el siguiente algoritmo:

ISSN 0188-6266

```
Arreglo Va  de todos los vértices en G(V;E),
Arreglo Ea de todas las aristas en G(V;E)
Arreglo de frontera Fa, arreglo de visitas Xa, arreglo
de costos Ca (tamaño v)
inicializar Fa en falso
inicializar Xa al infinito
 Fa[S] true
 Ca [S] 0
 mientras Fa no este vacío haz
   Para cada vertice V de forma paralela haz
   invocar RandomWalkThread (cores,Va;Ea;F a;Xa;Ca)
   termina para
termina mientras
```

Donde *RandomWalkThread* es el proceso encargado de realizar el recorrido y la caminata aleatoria, con la cantidad de hilos *Central Processing Unit* (CPU, por sus siglas en inglés) especificada por el parámetro cores, el cual es la cantidad de hilos de proceso que ejecutará la simulación.

El algoritmo BFS se aplica para recorrer el grafo, a pesar de ser paralelo por diseño, como se explicó anteriormente a partir del parámetro cores que recibe. En la implementación realizada en el presente trabajo es posible determinar la cantidad de núcleos de CPU a utilizar, abriendo posibilidades a futuros trabajos en cuestión de paralelización.

RESULTADOS

En la figura 4 se muestra el grafo parcial de nodos obtenido de la recolección de datos, donde se obtuvieron 8200 nodos, así como 21 000 relaciones entre ellos. Dicho grafo se encuentra reducido a sólo 1200 nodos, debido la difícil visualización de un grafo de estas dimensiones.

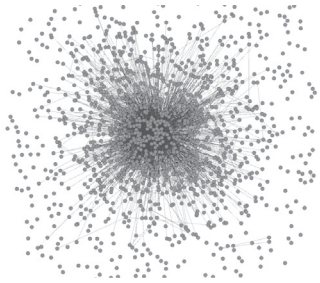

Figura 4. Grafo representado por archivo de matriz de adyacencia.
Fuente: Elaboración propia.

El contenido parcial del archivo para dicho grafo es el siguiente:

```
nodedef>name,mention_count INT,member_count INT
ghettoremedies,6,2
all,2,2
webtrends,1,1
frenchvoc,1,1
euronews,3,2
haitiearthquake,49,40
halffull,1,1
edgedef>node1 VARCHAR,node2 VARCHAR,weight INT
donations,earthquake,2
holiday,yele,2
donate,pray,1
earthquake,facts,1
aid,ict4d,1
ckin,haitian,1
haiti,welheftig,1
```

En la aplicación del algoritmo de mezcla dinámica de cliqués sobre dicho grafo de nodos se obtuvo como resultado el grafo de cliqués de la figura 5.

Como se observa en la figura 5, los nodos del grafo se encuentran agrupados por similitud, a través de las características de cada nodo (sexo, idioma, región, edad), mientras que los nodos no agrupados son nodos que no poseen grupos con características similares. Las tonalidades de los nodos especifican que un nodo coincide en algunas características con los demás de la misma tonalidad, sin embargo, a pesar de esto, pueden no ser del todo similares para DTW. Por esta razón hay nodos con diversas tonalidades, pero pertenecientes a diferentes cliqués, donde éstos especifican la similitud (distancia DTW) de los nodos.

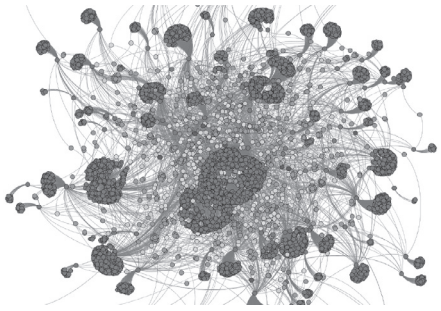

Figura 5. Grafo representado por archivo de matriz de adyacencia.
Fuente: Elaboración propia.

Métricas

Para la evaluación de resultados se usaron las siguientes métricas:

- **Vida media del mensaje:** tiempo total en minutos que tardó en dejar de propagarse el mensaje generado en la simulación.
- **Profundidad:** niveles que logró recorrer el mensaje simulado, desde el nodo inicial hasta la finalización del mismo.
- **Centralidad por cliqués de origen:** centralidad acumulada de nodo inicial e hijos de dicho nodo.
- **Centralidad por cliqués de resonancia:** centralidad por nodo único.
- **Tasa de réplicas (contestaciones):** cantidad de veces que un mensaje fue contestado.
- **Tasa de favoritos:** número de veces que el mensaje simulado fue marcado como favorito.

La ejecución del algoritmo de simulación se realizó en siete secuencias, utilizando como entrada el grafo de 8200 nodos, aplicándole los siguientes tres métodos de mezcla de cliqués (G1, G2, G3), considerando las cuatro características de los nodos.

G-1.
Grafo con cliqués de tamaño pequeño generados a partir de la agrupación no dinámica (una a dos nodos por cliqués).

Ejecución	Profundidad	Cliqués de origen	Resonancia	Tasa de réplicas	Tasa de favoritos
RW1	2	1	1	0	1
RW2	18	10	5	1	0
RW3	34	16	16	1	0
RW4	172	68	22	0	1
RW5	21	1	1	0	0
RW6	10	6	1	0	0
Media	**42.8**	**16.1**	**7.6**	**0.3**	**0.3**

G-2.
Grafo generado con el algoritmo de mezcla de dinámica de cliqués propuesto en el presente trabajo (grafos balanceados y agrupados dinámicamente por similitud).

Ejecución	Profundidad	Cliqués de origen	Resonancia	Tasa de réplicas	Tasa de favoritos
RW1	3	0	0	3	0
RW2	3	156	212	0	0
RW3	1	22	13	1	0
RW4	2	26	22	0	0
RW5	2	9	76	0	0
RW6	3	6	1	0	0
RW7	1	44	22	0	0
Media	**1.8**	**37.5**	**49.4**	**0.5**	**0**

G-3.
Grafo con cliqués de gran tamaño agrupados no dinámicamente a través una sola característica de los nodos (de 15 a 200 nodos por cliqués).

Ejecución	Profundidad	Cliqués de origen	Resonancia	Tasa de réplicas	Tasa de favoritos
RW1	0	0	0	0	0
RW2	0	384	44	0	0
RW3	0	12	20	0	0
RW4	0	56	13	0	0
RW5	0	98	22	0	0
RW6	0	11	6	0	0
RW7	0	34	10	0	0
Media	**0**	**85**	**16.4**	**0**	**0**

ISSN 0188-6266

Los resultados de las métricas fueron almacenadas en un archivo de resultados de simulación, el cual posee el grafo completo, las características, las métricas y cómo se propagó el mensaje simulado. El archivo de resultado es generado con el formato del siguiente ejemplo:

```
3
1,Cliqué 1,C,C,C,25,2,80,1,20,50
221036078,Daniel Espinosa Hernández,H,Español,
MX,42,1,30,1,100,10
437804658,Guadalupe Ortiz Mendieta,M,Inglés,
US,16,1,30,1,200,20
0
4
1 221036078
2 437804658
3 437804658
4 214328887
0
7200000
1080
```

Donde:

```
221036078,Daniel Espinosa Hernández,H,Español,
MX,42,1,437804658,30,1,100,10
```

Se refiere:

```
ID,NOMBRE,SEXO,IDIOMA,REGION,Centralidad,
EDAD,NO_RELACIONES,PROPAGADO,PROPAGADO_
PADRE,RESONANCIA,TASA_REPLIES,TASA_FAVORITO
```

Y en la última sección del archivo (siguientes al 0 final), se especifican las métricas de vida media del mensaje y profundidad de origen.

Probabilidad de reenvío de mensajes

Se utilizó el modelo probabilístico *Hidden Variables, Missing Data* para obtener la probabilidad de reenvío de mensajes, disponiendo como entrada los resultados obtenidos de la ejecución del algoritmo de simulación, con cliqués dinámicos q y θ:

θ = *Variables no conocidas*

(*Probabilidad de alta resonacia*)

$$\upsilon = \frac{(Favoritos + Réplicas)}{Profundidad}$$

$f(\theta_s)$ = *Small Cliques*

$f(\theta_d)$ = *Dynamic Cliques*

$f(\theta_b)$ = *Big Cliques*

h = *Corridas RW*1 ... *RW*7(1,2)

$$P(\upsilon \backslash \theta) = \sum_{h} P(\upsilon, h \backslash \theta) \quad (3)$$

Con respecto a θ.

Al realizar el despeje de dicha ecuación se obtiene y desarrolla la ecuación siguiente, donde se usa un valor fijo de las variables visibles de 2.75 para obtener la mayor probabilidad de alta resonancia:

$$P(\upsilon = 2.75|\theta) = \sum_{h=1,2} P(\upsilon = 2.75,\ h|\theta)\ p(h)$$

$$\frac{1}{2\sqrt{\pi}} \left(e^{-(2.75-\theta)2} + e^{-(2.75-2\theta)2}\right)$$

$$q(h|v) = (\log p(v, h|\theta))$$

$$q(h|v) + (\log p(h))\ q(h|\upsilon)$$

$$= -((\upsilon - \theta h)^2) + const$$

$$\log p(\upsilon = 2.75 \mid \theta) \geq L(q(2)\theta)$$

$$\equiv -q(1) \log q(1) - q(2) \log q(2) - \sum_{h=1,2} q(h)(2.75 - \theta h)^2 + const \quad (4)$$

DISCUSIÓN

Al graficar los valores de resultado de las variables visibles y no visibles, en las diferentes ejecuciones (RW1… RW7), se obtienen las figuras 6, 7 y 8. En todas las figuras se mide en el eje *Y* la vida media del mensaje, en el eje *X* las métricas y en el eje *Z* las ejecuciones.

Particularmente en la figura 6 se observa que con el grafo de cliqués de tamaño pequeño se obtiene una profundidad alta en la ejecución RW4, derivado de que el número de nodos del grafo es mayor, y la simulación tiene mayor cantidad de ellos para recorrer hasta finalizar la ejecución.

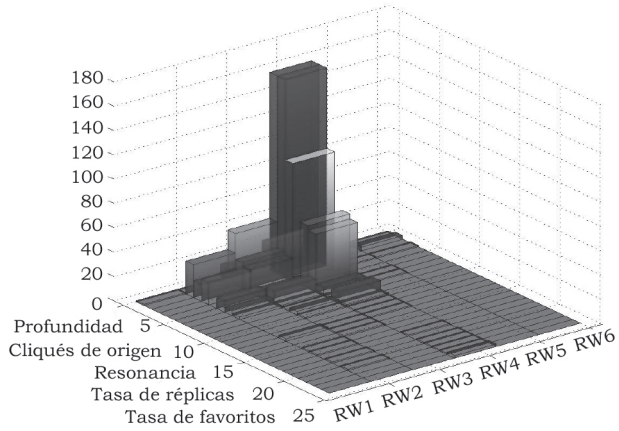

Figura 6. (G1). Ejecución con grafo de cliqués de tamaño pequeño.
Fuente: Elaboración propia.

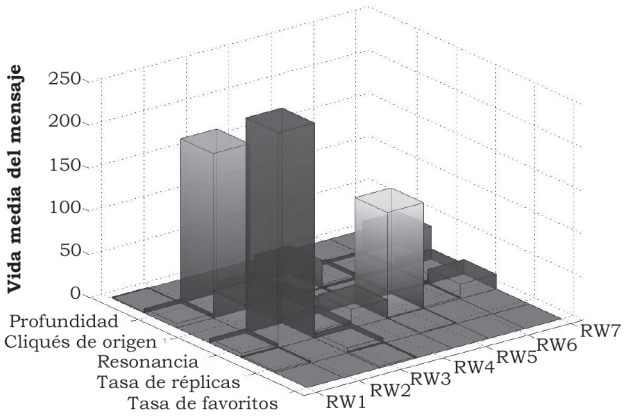

Figura 7. (G2). Ejecución con grafo de cliqués de tamaño pequeño.
Fuente: Elaboración propia.

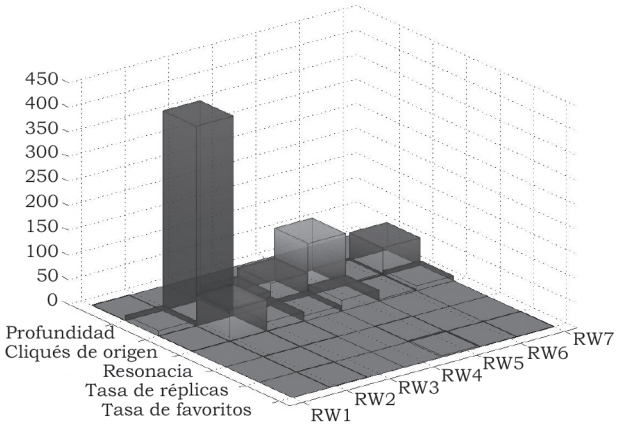

Figura 8. (G3). Ejecución con grafo de cliqués de tamaño balanceado y generado a través del algoritmo de mezcla dinámica.
Fuente: Elaboración propia.

Asimismo, la centralidad por cliqués de origen es alta, lo cual muestra que se replicó una centralidad acumulada elevada. Este efecto es debido a la cantidad de nodos del grafo, puesto que la centralidad por una gran cantidad de nodos acumulados aumenta. Sin embargo, la centralidad por cliqués de resonancia es baja, mostrando que los grupos generados con cliqués de tamaño pequeño no obtienen una resonancia alta, lo cual indica que dicha forma de generar los cliqués obtiene resultados no favorables si se quiere medir la resonancia para conocer el grado de respuesta de los participantes en la red.

La tasa de réplicas y de favoritos fueron pequeñas, debido de nuevo al gran tamaño de población de nodos en el grafo.

La figura 7 denota que la simulación con el grafo de cliqués balanceados dinámicos, obtenidos con el algoritmo de mezcla dinámica de cliqués, obtuvo una profundidad baja en todas las ejecuciones, debido a que al algoritmo de simulación no le toma tantas réplicas recorrer cada cliqué, teniendo así una menor cantidad de nodos para propagar.

Por otro lado, la centralidad por cliqués de origen es baja. Ello prueba que se replicó una centralidad acumulada baja, debido, igualmente, a la cantidad de nodos del grafo, puesto que la centralidad por una cantidad media de nodos acumulados no se eleva. Sin embargo, la centralidad por cliqués de resonancia es muy alta en tres de las ejecuciones, demostrando que los grupos generados con cliqués dinámicos de tamaño balanceado permiten agrupar los nodos de forma que los grupos generados se hagan más resonantes, mostrando así que el algoritmo presentado en el este trabajo mejora la centralidad en la agrupación de nodos. Lo relevante al usar este algoritmo es que se puede percibir con mayor claridad la resonancia de un mensaje en las redes sociales, además de considerar mayor cantidad de nodos que en el algoritmo G1. Esto lo hace más relevante.

La tasa de réplicas y de favoritos es baja por la formación de cliqués que se formaron. De acuerdo con los datos obtenidos en G1, sugiere que estas dos métricas no son relevantes.

Con la figura 8 se explica la simulación con el grafo de cliqués de gran tamaño. Se obtuvo una profundidad promedio alta, en razón a que al algoritmo de simulación le toma muchas réplicas recorrer cada cliqué de gran tamaño.

ISSN 0188-6266

Por otro lado, la centralidad por cliqués de origen que se obtuvo es mediana, evidenciando que se replicó una centralidad acumulada media. Esto es debido a la cantidad de recorridos que debe realizar el algoritmo en cada cliqué por su tamaño en el grafo. No obstante, la centralidad por cliqués de resonancia es baja en todas las ejecuciones, asunto que denota que los grupos generados con cliqués de gran tamaño no obtienen buen resultados al buscar centralidad de grupos.

Si se consideran los valores medios de todas las secuencias de cada uno de los experimentos (figura 9), se infiere que los mejores resultados favorecen al algoritmo propuesto en este trabajo, por lo que es recomendable utilizar la mezcla dinámica de cliqués con un alto índice de resonancia.

Las métricas utilizadas en el presente artículo obtienen una distribución de resonancia (contribución) similar al del trabajo presentado por Zheng *et al.* (2010), donde la resonancia es pequeña, y sólo por algunos miembros de la red que crean grandes cantidades de información. Dichos resultados también muestran una distribución similar al algoritmo basado en *pagerank*, propuesto por Khrabrov & Cybenko (2010), en el cual se emplean métricas similares a las del presente artículo para evaluar la resonancia de individuos.

Por otro lado, resultados obtenidos muestran que los cliqués generados a partir del algoritmo de mezcla dinámica de cliqués logran una alta modularidad, muy cercana al algoritmo conocido como *CliqueMod-BK* presentado por Yan & Gregory (2009), donde, a partir de un grafo de entradas con pocos nodos, mediante su algoritmo de mezcla de cliqués, disminuye la complejidad del grafo casi entre ocho y diez veces, generando resultados acordes con los del presente trabajo y dando posibilidad a futuros trabajos en cuestión de tiempos de ejecución y paralización del algoritmo, utilizando técnicas como mapReduce, como las propuestas en el trabajo de Svendsen, Mukherjee & Tirthapura (2014).

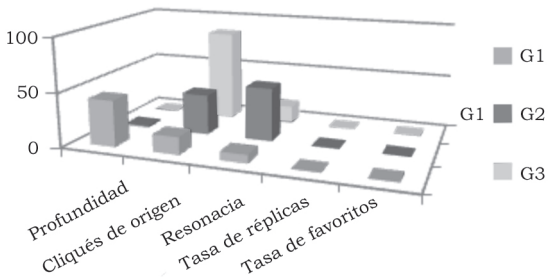

Figura 9. (G2). Ejecución con grafo de cliqués de tamaño pequeño.
Fuente: Elaboración propia.

CONCLUSIONES

Es posible simular la propagación de mensajes en redes sociales, a trabajos de modelos matemáticos como las caminatas aleatorias, las cuales en el presente trabajo son utilizadas para representar de forma estocástica la propagación de dichos mensajes.

Mediante la evaluación de métricas es posible simular la resonancia de los nodos de un grafo social.

A través de algoritmos de búsqueda de similitud, como DTW, es posible crear grupos (cliqués) dinámicos de individuos que tengan características similares.

Los grafos con cliqués de muy pocos nodos (de pequeño tamaño) logran conseguir una profundidad y una resonancia acumulada por cliqués de origen muy alta, por la gran cantidad de nodos que un mensaje debe recorrer, pero una baja resonancia individual por cliqué/nodo cuestión que no lo favorece a ser el mejor.

La tasa de respuestas y los mensajes favoritos son irrelevantes para la generación de cliqués dinámicos con alta resonancia.

El algoritmo de mezcla dinámica de cliqués mediante DTW obtiene buenos resultados, similares a trabajos de otros investigadores al generar cliqués.

Los grafos con cliqués dinámicos balanceados, agrupados por la mezcla dinámica de cliqués mediante DTW, obtienen una baja profundidad y resonancia acumulada, sin embargo, dichos cliqués generados poseen una alta resonancia individual que los hacer ser favorecidos para ser el mejor en el presente trabajo.

Los grafos con cliqués muy grandes (con una gran cantidad de nodos) tienen una alta resonancia acumulada y una baja profundidad, debido a que los grupos son de un tamaño muy alto y la cantidad de nodos que el mensaje debe recorrer no es tan elevada. Sin embargo, poseen una resonancia individual por cliqué muy baja, cuestión que lo hace ser no muy deseable para los fines de este trabajo.

El presente trabajo propone la metodología de un algoritmo para generar grupos de nodos con similitud y que posean una alta resonancia (mezcla dinámica de cliqués) y tiene resultados equiparables a los de otros investigadores con trabajos similares.

REFERENCIAS

Belo, R. & Ferreira, P. (September, 2012). *Using Randomization to Identify Social Influence in Mobile Networks*. Trabajo presentado en la 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)Privacy, Security, Risk and Trust (PASSAT), Amsterdam, Países Bajos. doi: 10.1109/SocialCom-PASSAT.2012.62

Durr, M., Protschky, V. & Linnhoff-Popien, C. (August, 2012). *Modeling Social Network Interaction Graphs*. Trabajo presentado en la 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Istanbul, Turquía. doi: 10.1109/ASONAM.2012.110

Fortunato, S. (2010). Community detection in graphs. *Physics Reports 486*, 75-174. doi: 10.1016/j.physrep.2009.11.002

Guo, R. (December, 2012). *Research on Information Spreading Model of Social Network*. Trabajo presentado en la 2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC), Harbin, China. doi: 10.1109/IMCCC.2012.220

Huang X., Acero A., & Hon H. W. (2001). *Spoken Language Processing*. USA: Prentice-Hall.

Ilyas, M. U. & Radha, H. (June, 2011). *Identifying Influential Nodes in Online Social Networks Using Principal Component Centrality*. Trabajo presentado en la 2011 IEEE International Conference on Communications (ICC), Kyoto, Japón. doi: 10.1109/icc.2011.5963147

Junquero-Trabado, V., Trench-Ribes, N., Aguila-Lorente, M. A. & Dominguez-Sal, D. (October, 2011). *Comparison of influence metrics in information diffusion networks*. Trabajo presentado en la 2011 International Conference on Computational Aspects of Social Networks (CASoN). Salamanca, España. doi: 10.1109/CASON.2011.6085914

Khrabrov, A. & Cybenko, G. (August, 2010). *Discovering Influence in Communication Networks using Dynamic Graph Analysis*. Trabajo presentado en la 2010 IEEE Second International Conference on Social Computing (SocialCom), Minneapolis, EU. doi: 10.1109/SocialCom.2010.48

Liu, D. & Chen, X. (November, 2011). *Rumor Propagation in Online Social Networks Like Twitter —A Simulation Study*. Trabajo presentado en la 2011 Third International Conference on Multimedia Information Networking and Security (MINES), Shanghai, China. doi: 10.1109/MINES.2011.109.

Müller, M. (2007). Dynamic Time Warping. In *Information Retrieval for Music and Motion*. Berlin Heidelberg: Springer-Verlag.

Nekovee, M., Moreno, Y., Bianconi, G. & Marsili, M. (2007). Theory of rumour spreading in complex social. *Physica A*, *374*, 457-470. doi: 10.1016/j.physa.2006.07.017

Svendsen, M., Mukherjee, A. P. & Tirthapura, S. (2014). Mining Maximal Cliques from a Large Graph using MapReduce: Tackling Highly Uneven Subproblem Sizes. *Journal of Parallel and Distributed Computing* (Available online). doi:10.1016/j.jpdc.2014.08.011

Wei, X., Valler, N., Prakash, B. A., Neamtiu, I., Faloutsos, M. & Faloutsos, C. (2012). Competing memes propagation on networks: a case study of composite networks. *ACM SIGCOMM Computer Communication Review*, *42*(5), 6-11. doi: 10.1145/2378956.2378958

Yan, B. & Gregory, S. (November, 2009). *Detecting Communities in Networks by Merging Cliques*. Trabajo presentado en la ICIS 2009. IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009, Shanghai, China. doi: 10.1109/ICICISYS.2009.5358036

Zheng, J., Chen. W., Zhang, L. & Bu, J. (April, 2010). *A Metric for Measuring Members' Contribution to Information Propagation in Social Network Sites*. Trabajo presnetad en la 2010 12th International Asia-Pacific Web Conference (APWEB), Busan, Corea del Sur. doi: 10.1109/APWeb.2010.50