

Análisis de Redes Sociales basado en las Conquistas de César Borgia

1st Javier Fumanal-Idocin
Estadística, Informática, Matemáticas
Universidad Pública de Navarra
 Pamplona, España
 javier.fumanal@unavarra.es

4th Humberto Bustince
Estadística, Informática, Matemáticas
Universidad Pública de Navarra
 Pamplona, España
 bustince@unavarra.es

2nd Oscar Cordón
Andalucian Research Institute
University of Granada
 Granada, España
 ocordova@decsai.ugr.es

5th Javier Fernández
Estadística, Informática, Matemáticas
Universidad Pública de Navarra
 Pamplona, España
 bustince@unavarra.es

3rd Amparo Alonso-Betanzos
CITIC
University of A Coruña
 A Coruña, Spain
 amparo.alonso.betanzos@udc.es

Resumen—En este trabajo presentamos el modelado de redes sociales y detección de comunidades utilizando como base un evento histórico real, las conquistas de César Borgia en el siglo XV. Para ello, proponemos un nuevo conjunto de funciones, llamadas funciones de afinidad, diseñadas para capturar la naturaleza de las interacciones locales entre cada par de actores en una red. Utilizando estas funciones, desarrollamos un nuevo algoritmo de detección de comunidades, el Borgia Clustering, donde las comunidades surgen naturalmente de un proceso de simulación de interacción de múltiples agentes en la red. También discutimos los efectos del tamaño y la escala de cada comunidad, y cómo pueden ser tomadas en cuenta en el proceso de simulación. Finalmente, comparamos nuestra detección de comunidades con otros algoritmos representativos, encontrando resultados favorables a nuestra propuesta.

I. INTRODUCCIÓN

El análisis de redes se ha convertido en una herramienta importante para estudiar sistemas compuestos por agentes que interactúan, como proteínas o sociedades humanas [1]–[4]. Hay muchos problemas relacionados con las redes sociales, como la difusión de información [5], la detección de círculos sociales [6], la formación de coaliciones [7] o los sistemas de recomendación [8]. Las redes sociales también pueden verse como sistemas multiagente, y es posible estudiar sus propiedades emergentes [9]. Un paso clave para analizar una red es identificar su estructura de comunidad: los grupos de nodos que pueden identificarse como una subpartición funcional del grafo [10], [11]: por ejemplo, un grupo de amigos o un complejo de proteínas [12]. Las comunidades son importantes porque podemos inferir un conocimiento significativo de un nodo o un conjunto de nodos si sabemos si comparten o no la misma comunidad y qué tipo de comunidad es. Existen muchos métodos para realizar la detección de comunidades. Por ejemplo, la propuesta en [13] usa el ratio de conexión internas y externas de grupos de nodos para obtener una rápida división en grupos de la red, o también, los autores de [14] proponen un enfoque de optimización de modularidad

para trabajar en redes grandes, siendo este uno de los métodos de detección de comunidades más utilizados.

En este artículo, nuestro objetivo es resolver algunos de los problemas de análisis de redes sociales que hemos identificado tanto en la interacción entre actores como en la detección de la comunidades. Primero, los algoritmos de detección de comunidades actuales no tienen en cuenta ni la diferente naturaleza de los seres humanos involucrados en estas, ni el impacto de esta naturaleza en la estructura de la red. Tampoco tienen en cuenta cómo la escala en redes sociales puede alterar su dinámica. Además, cuando se trata de detección de comunidades, muchos de los algoritmos existentes tienen problemas en grafos densamente conectados y los algoritmos apropiados a menudo consumen mucho tiempo [15].

Para resolver estos problemas, estudiamos cómo los algoritmos inspirados en humanos pueden conducir a una mejor comprensión de la estructura de una red social y sus comunidades. En contraste con la literatura existente, que utiliza una matriz de adyacencia para modelar una red [16], proponemos un conjunto de funciones con el fin de capturar mejor las relaciones entre cada par de actores en el contexto social de la red. Utilizando como base estas funciones de afinidad y el algoritmo gravitacional de clustering [17], hemos desarrollado un nuevo algoritmo para realizar la detección de comunidades, el Borgia Clustering, que debe su nombre a que su comportamiento busca emular las mismas condiciones que se dieron durante las conquistas de César Borgia.

Usando las funciones de afinidad y el Borgia Clustering, nuestro objetivo es tener una mejor comprensión de las interacciones locales entre y cómo pueden afectar la dinámica global y formación de comunidades. De esta manera, podremos generar un dendrograma que refleje fielmente la evolución de la red en el proceso de agrupamiento, y elegir la configuración correcta dentro de él.

El resto del trabajo es la siguiente. En la Sección II explicamos los conceptos básicos del algoritmo gravitacional y la teoría de grafos. En la Sección III explicamos las nuevas

funciones de afinidad y damos un caso práctico de su uso en *República* de Platón. En la Sección IV presentamos el algoritmo Borgia Clustering y el momento histórico que lo inspiró, y en la Sección V comparamos la calidad de nuestra solución con otros algoritmos representativos. Finalmente, en la Sección VI resumimos todo el trabajo y establecemos algunas pautas futuras.

II. PRELIMINARIES

En esta sección explicaremos brevemente algunos de los conceptos ya existentes relacionados con algunas de las nuevas propuestas de este trabajo:

- Algoritmo de clústering gravitacional.
- Teoría de grafos.

II-A. Algoritmo de clústering gravitacional

El algoritmo de clústering gravitacional [17] emplea la ley gravitacional de Newton para modelar la atracción entre distintas observaciones de datos. En este algoritmo, cada observación es una partícula que atrae a las demás según sus distancias y masas. Cuando dos partículas están más cerca que la distancia de colisión, se fusionan en una sola. Su masa es la suma de esas partículas y su posición es su centro de masas. Este proceso se repite hasta que solo existe una partícula. Este algoritmo da como resultado un dendrograma que contiene cada fusión de partículas. Finalmente, la configuración más estable se toma como la resultante. El esquema es el siguiente:

Supongamos que tenemos n partículas p_1, \dots, p_n , con posiciones $s_1, \dots, s_n \in R^n$. También tenemos dos parámetros: ϵ , que establece la distancia de colisión para dos partículas, y δ que determina el movimiento de la partícula más rápida en cada iteración:

1. Inicialmente asignamos una masa de $m_i = 1$ a cada partícula p_i .
2. Decidimos el valor de ϵ y δ
 - Usamos δ para determinar la duración del delta del tiempo, dt , que significa la velocidad a la que la partícula más rápida se mueve.
 - Si en cualquier momento dos partículas se encuentran a una distancia menor de ϵ se unen en una partícula. La masa de la nueva partícula es la suma de las dos, y su posición el centro de masas.
3. El tiempo inicial se pone a $t = 0$.
4. Repetimos los siguientes pasos (i)-(iv) hasta que sólo quede una partícula.
 - (i) En cada intervalo de tiempo $[t, t + dt]$, para cada p_i calculamos la fuerza:

$$g(i, t, dt) = \frac{1}{2} G \sum_{j \neq i} \frac{m_i(t)m_j(t)}{m_i(t)} \frac{s_j(t) - s_i(t)}{|s_j(t) - s_i(t)|^3} dt^2 \quad (1)$$

donde G es una constante positiva.

- (ii) Por cada partícula i , su nueva posición es:

$$s_i(t + dt) = s_i(t) + g(i, t, dt)$$

- (iii) Incrementamos t a $t + dt$.

- (iv) Si dos partículas i y j están más cerca que ϵ , se fusionan en una, siguiendo el procedimiento anteriormente explicado.

Cuando el algoritmo termine, quedará sólo una partícula. La duración del proceso entero es T . Medimos la duración de cada iteración también, con el objetivo de quedarnos con la configuración que más tiempo aguantó, que es el resultado final.

II-B. Teoría de Grafos

Un grafo G se representa como $G(V, E)$ donde V es un conjunto de vértices y E es un conjunto de aristas que conectan algunos pares de vértices en el grafo G . Hay diferentes tipos de grafos, dependiendo de la información relacionada con cada arista. En caso de que tengamos alguna información sobre la fuerza de las relaciones, llamamos al grafo un grafo ponderado. Si no tenemos tal información, se llama grafo no ponderado. Las aristas también pueden ser no dirigidas, cuando la arista entre $V_i \leftrightarrow V_j$ representa una relación bidireccional, o dirigida, cuando la arista entre $V_i \rightarrow V_j$ puede ser diferente de la arista $V_j \rightarrow V_i$. Los grafos se pueden caracterizar mediante el uso de muchas estadísticas, como el número promedio de conexiones por nodo, las longitudes de ruta promedio entre nodos, etc.

Los grafos se pueden modelar utilizando diferentes representaciones, las más comunes son la matriz de adyacencia o conectividad y la lista de adyacencia. La matriz de adyacencia A de un grafo G es una matriz $N \times N$ donde N es el número de nodos en G . Cada entrada A_{ij} en esta matriz corresponde al valor asociado con la arista $V_i \rightarrow V_j$. Si el grafo no está ponderado, esos valores serán 0 o 1, mientras que si el grafo está ponderado, esos valores serán los pesos correspondientes para cada arista. La lista de adyacencia es similar a la matriz de adyacencia, pero en lugar de almacenar $N \times N$ elementos, almacenamos una lista para cada arista que contiene el resto de nodos que se comunican con ella.

III. FUNCIONES DE AFINIDAD PARA CARACTERIZAR RELACIONES ENTRE ACTORES

Definimos las funciones de afinidad como funciones para dos actores, $Actor_x$ y $Actor_y$, que establecen una relación mutua usando C :

$$A_C : (Actor_x, Actor_y) \rightarrow [0, 1]$$

Generalmente, esta C es la matriz de adyacencia que cuantifica las relaciones en cada par de actores, aunque C puede ser, por ejemplo, otra matriz construida con funciones de afinidad, o una lista de ellas. La afinidad entre dos actores muestra cuán fuertemente están conectados estos dos. Dado que las afinidades no son necesariamente simétricas, la fuerza de esta interacción depende de quiénes son el emisor y el receptor, como sucede en la interacción humana, por ejemplo amor no correspondido.

Procedemos a enumerar algunas funciones de afinidad:

- **Afinidad de mejor amigo:** la afinidad del actor $Actor_x$ sobre $Actor_y$ se define como el porcentaje de la conectividad total de Y que corresponde a $C_{x,y}$.

$$A_C^{BF}(x, y) = \frac{C_{x,y}}{\sum_{a=1}^N C_{x,a}}$$

- **Afinidad de amigo común:** la afinidad entre dos actores se define como la mayor afinidad común a ambos.

$$A_C^{BCF}(x, y) = \frac{Max\{Min(C_{x,z}, C_{y,z})\}}{\sum_{a=1}^N C_{x,a}}$$

- **Afinidad de Mejores Amigos para Siempre:** la afinidad de dos actores refleja la durabilidad de la relación en el tiempo.

$$A_C^{FF} = \frac{Max\{Min(C_{x,z}, C_{y,z})\}}{\sum_{a=1}^N C_{x,a}}$$

- **Machiavelli Affinity:** la afinidad entre dos actores se basa en la estructura social que se construye alrededor de ellos dos.

$$A_C^{MAC}(x, y) = A_C(x, y) = 1 - \frac{abs(I_x - I_y)}{Max(I_x, I_y)},$$

$$I_a = Sum(Degree(x')) \forall a, C_{a,x'} > 0$$

Existen, principalmente, dos tipos de funciones de afinidad: afinidades personales y afinidades estructurales. Las afinidades personales establecen la fuerza de una conexión interpersonal $Actor_x \rightarrow Actor_y$ usando sus respectivas conexiones y amigos compartidos. Las afinidades estructurales cuantifican la relación de un par de actores en función de las propiedades de sus nodos, como su grado o intermediación. Todas las afinidades aquí propuestas, a excepción de la de Maquiavelo, son afinidades personales.

III-A. Efecto de diferentes funciones de afinidades en República de Platón

Como caso de estudio, consideramos la red de asociación de palabras en un clásico de la literatura: *República*, de Platón. Hemos visualizado la red resultante para cada afinidad diferente (Figura 1). En estas redes, cada nodo corresponde a una palabra diferente en la obra original, y su tamaño es directamente proporcional a su grado. En la red de co-ocurrencias, cada arista representa la cantidad de veces que dos palabras aparecen juntas en un párrafo y en los grafos de afinidad, cada arista es el valor de afinidad para cada par de nodos. Por motivos de claridad, solo las 130 palabras más frecuentes están presentes en cada red.

Dependiendo de la función de afinidad utilizada, las aristas resultantes y sus pesos pueden ser muy diferentes. Como ejemplo, en la Tabla I hemos calculado las cinco afinidades para la entidad “Man” (Por conveniencia, en este trabajo hemos considerado la traducción inglesa y respetamos el nombre original de las entidades procesadas).

Outgoing	Adj.	BF	BCF	FF	MA
Top 1	Justice	Justice	Justice	Life	Life
Top 2	Life	Life	Injustice	Other	Justice
Top 3	Injustice	Injustice	Soul	One	State
Top 4	Soul	Soul	Life	Soul	Men
Top 5	State	State	State	Evil	Soul
Incoming					
Top 1	Justice	Master	Tyrant	Reward	Soul
Top 2	Life	Desire	Desire	God	Justice
Top 3	Injustice	Action	Spirit	Gold	Life
Top 4	Soul	Word	Journey	Work	State
Top 5	State	Case	Master	Protect	Men

Cuadro I: El efecto de diferentes cálculos de afinidad para el actor “Man” en República de Platón. Cada columna muestra los valores más altos de la matriz de adyacencia y diferentes funciones de afinidad para cada arista del actor “Man”.

IV. BORGIA CLUSTERING

En 1497, César Borgia como Comandante en Jefe del Ejército Papal y al mando de su padre, Alejandro VI, marchó por el centro de Italia, conquistando todos los territorios tradicionalmente vinculados a los Estados Pontificios [19]. Este acontecimiento crucial en la historia del Renacimiento nos dejó no solo momentos de memorable valentía individual y genio táctico, sino también un excelente ejemplo de interacción humana tanto a nivel personal como comunitario.

Nuestro objetivo es generar un algoritmo que parta de una red social cualquiera y simule las mismas dinámicas que generó César Borgia al iniciar su campaña. Concretamente, nos hemos fijados en estas condiciones:

- Todos los países tienen como objetivo crecer. Aunque pueda parecer una cosa trivial, no siempre es así. A veces, los estados prefieren crear y mantener un equilibrio en lugar de romperlo a su favor.
- Algunas partes de Italia son culturalmente más similares a países diferentes. Nápoles, por ejemplo, está mucho más influenciada por la cultura española que Milán, y las futuras uniones dinásticas y conquistas harán evidentes estas diferencias.
- Las diferencias de tamaño y poder hacen que algunas alianzas sean más valiosas que otras. Las repúblicas italianas buscan alianzas no solo en su península sino en tierras de naciones más poderosas. Este comportamiento favorece la creación de bandos enfrentados liderados por grandes potencias, como fue el caso de las numerosas guerras franco-españolas ocurridas en los siglos XV y XVI, lo que puede entrar en conflicto con las afinidades culturales de los distintos países.
- Hubo muchos países de diferente tamaño e importancia involucrados en ese evento histórico. Francia y España fueron los más grandes, seguidos de Milán, Venecia, Florencia y el Reino de Nápoles. También estaban los Estados Pontificios y los “señores independientes” en el

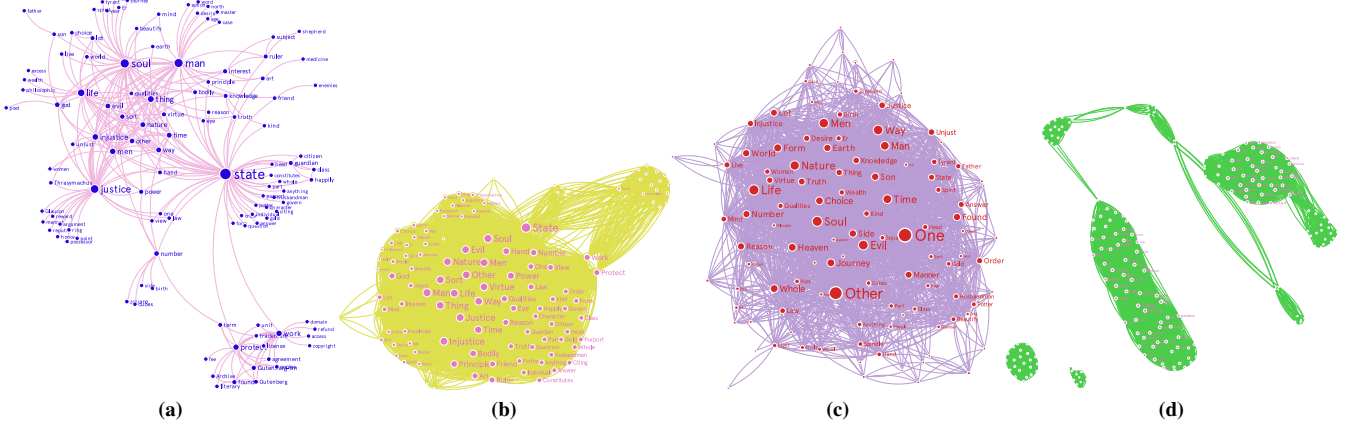


Figura 1: Distintas formas de representación en red de República de Platón. a. Red de coocurrencias/Afinidad de mejor amigo. **b.** Red de afinidad de mejor amigo común. **c.** Red de afinidad de mejores amigos para siempre. **d.** Red de afinidad de Maquiavelo. Las redes han sido dibujadas con el algoritmo Force Atlas 2 [18]

centro de Italia. Contrariamente al pensamiento ingenuo, los condados más pequeños no fueron conquistados por los países más grandes, que son los que tienen ejércitos más grandes, sino por los Estados Pontificios.

El algoritmo se basa en el clásico algoritmo *Gravitational Clustering* [17]: cada actor comienza como una comunidad diferente que se acerca a los demás debido al efecto de una fuerza de atracción. Esta fuerza de atracción hará que cada comunidad quiera crecer siempre, de acuerdo a sus afinidades naturales y al valor de cada posible “alianza”. A continuación, detallamos todas las modificaciones hechas al algoritmo original.

IV-A. Modificación en las partículas

En una red social, cada partícula no es un punto en el espacio, sino un actor. Los actores tienen un conjunto de propiedades adicionales, generalmente relacionadas con la información semántica disponible para cada uno. La más importante es la conectividad, que nos permite calcular la afinidad de mejor amigo y de mejor amigo común.

Para representar a cada actor, utilizamos una combinación de la matriz de estas afinidades, y una matriz de influencia, S , basada en la primera. Al usar la afinidad de mejor amigo, favorecemos interacciones fuertes por pares, y con la afinidad de mejor amigo común, también favorecemos la formación de comunidades cuyos miembros comparten un gran número de amigos.

Los actores tienen un valor social inicial, m , que inicialmente corresponde a su grado, que es similar al concepto de masa de partículas en el algoritmo original y que refleja su popularidad. Esto nos permite mostrar dos comportamientos sociales bien conocidos: (1) las personas populares son más atractivas socialmente que las personas con menos amigos. Este hecho es llamado ventaja acumulativa en la teoría de redes [20] (desde el punto de vista físico de, una mayor masa implica una mayor fuerza de atracción). (2) Las personas con pocos

amigos intentan pasar el rato con gente popular, pero no al revés (una mayor masa también implica menos movimiento).

El concepto de posición es sustituido por el concepto de influencia, denotado como la matriz S , que es construido usando una matriz de afinidad. El valor s_{ij} refleja la influencia que el actor j tiene sobre el actor i . Las autoinfluencias (s_{ii}) se establecen en 1. Los actores interactuarán entre sí a lo largo del tiempo y se acercarán a medida que pase el tiempo debido a esta interacción. Además, las autoinfluencias disminuirán durante la ejecución del algoritmo.

La matriz A_C es inicialmente la misma que la matriz S , pero A_C solo cambia sus valores cuando dos actores chocan y forman una nueva comunidad.

IV-B. Modificaciones en la fusión

La condición para comprobar si dos actores han chocado usando la distancia euclidiana no es buena en este contexto debido a la maldición de la dimensionalidad. Por tanto, hemos utilizado la idea de influencia en lugar de posición para comprobar si dos actores deberían fusionarse o no.

Cada actor comienza siendo completamente influyente sobre sí mismo. Durante la ejecución del algoritmo, los actores se atraen entre sí, provocando una reducción en su propia autoinfluencia y aumentando la influencia que otros actores tienen sobre ellos. Cuando la influencia de un actor sobre otro es mayor que la autoinfluencia de ese actor, esos actores colapsarán en una nueva comunidad.

La fusión de dos actores a, b da como resultado la aparición de un nuevo actor cuyo valor social es la suma de m_a y m_b , cuya posición es su centro de masas de s_a y s_b , y cuyas afinidades son el centro de masas de las afinidades de a y b .

IV-C. Nueva fórmula de atracción

Hemos modificado la fórmula de atracción para tener en cuenta no solo la masa y la distancia, sino todo el conjunto de características presentes en cada actor.

Tenemos en cuenta las relaciones bilaterales utilizando las matrices A_C y S . La matriz A_C devuelve la afinidad por un par de actores/comunidades, y usamos S para calcular las distancias euclidianas entre ellos, de modo que las afinidades más comunes o similares entre dos actores sean, mayor será la fuerza de atracción.

También debemos tener en cuenta el tamaño del actor/comunidad. Para hacerlo, agregamos un parámetro de penalización de “expansión voraz”, v . Este v penaliza el tamaño del actor de una manera no lineal, de modo que cuanto más grande se vuelve, más difícil tiene moverse. La penalización se calcula como $\frac{1}{m_x^v}$ para cada actor x . Esta idea que se usa aquí para afrontar el tamaño es lo que hemos llamado la “política romana inicial”, ya que identificamos este fenómeno como la dificultad de gestionar comunidades grandes, y que necesitan ser gobernadas de formas distintas cuando alcanzan cierta masa crítica, como observó César Augusto al negarse a añadir terrenos nuevos a su imperio tras vencer a los Partos [21].

El producto de las masas y A_C se multiplican. Esto es importante para la velocidad de cálculo porque el producto en el retículo [0,1] es siempre menor o igual que el mínimo. Si $A_C(x, y) = 0$, no necesitamos calcular la fuerza de atracción entre x y y porque sabemos que debe ser 0. Como la densidad de las matrices de afinidad suele ser muy baja, esto supone un gran ahorro computacional.

Como resultado, la fórmula final para la fuerza de atracción es la siguiente:

$$\mathbf{F}_{xy} = \frac{(m_x m_y)^c, A_C(x, y)}{m_x^v}, \frac{\mathbf{s}_x - \mathbf{s}_y}{|\mathbf{s}_x - \mathbf{s}_y|^3} dt \quad (2)$$

donde A_C representa la función de afinidad elegida y \mathbf{s}_x el vector de influencia del actor x .

IV-D. Escogiendo una configuración como resultado final

En el algoritmo gravitacional original, la configuración con la vida más larga en el tiempo simulado se elige como resultado. Este criterio es extremadamente rápido de calcular, pero tiende a devolver un número muy pequeño de comunidades. Esto sucede porque en los últimos pasos del algoritmo solo tenemos un número reducido de partículas que se mueven muy lentamente porque son muy pesadas.

Eso no supone un problema si el número deseado de comunidades es bajo (< 5) pero en caso de querer más, es necesario utilizar otro término para medir la calidad, R , de cada configuración, Z :

$$R(Z) = \text{TiempoSim}(Z) \log(\text{NumCom}(Z)) \quad (3)$$

Al usar esta fórmula, premiamos tanto la estabilidad en el tiempo como un mayor número de comunidades diferentes. También es posible especificar el número exacto de comunidades deseadas.

V. COMPARACIÓN CON OTROS ALGORITMOS DE DETECCIÓN DE COMUNIDADES

Por último, para comparar nuestro algoritmo de detección de la comunidad, hemos escogido datos de redes con etiquetas

reales sobre las comunidades formadas: la famosa red social del club de karate de Zachary [22], Poolboks [23], que contiene la cantidad de compras conjuntas de diferentes libros sobre política estadounidense, Football [10], que representa el número de partidos entre cada par de equipos, y Dolphins [24] es una red que registra la frecuencia con la que cada par de delfines jugó juntos.

Los otros algoritmos de detección de comunidades probados son: Girvan-Newman [25], optimización de modularidad voraz de Newman [13], el algoritmo de Lovaina [14], utilizar los valores propios de la red para detectar comunidades [26] y propagación de etiquetas [27]. Hemos utilizado Normalized Mutual Information (NMI) [28] y el Adjusted Random Index (ARI) [29] para comparar los resultados.

Los resultados se muestran en la Figura 2. En la figura se muestra como el Borgia Clustering superó al resto de algoritmos en la medida conjunta de ARI+NMI, y también cada una individualmente.

VI. CONCLUSIONES Y LÍNEAS FUTURAS

En este trabajo hemos propuesto un nuevo conjunto de funciones basadas en la naturaleza humana para representar a los actores de una red social y hemos mostrado los efectos que diferentes afinidades tienen en *República* de Platón. También hemos discutido la importancia de la escala en las redes y hemos aplicado nuestras ideas para desarrollar un nuevo algoritmo de detección de comunidades basado en un proceso histórico real, el Borgia Clustering, tomando como base las guerras centro-italianas del siglo XV. Utilizando este algoritmo, hemos obtenido buenos resultados en comparación con los algoritmos de detección de comunidades más utilizados hasta la fecha.

Creemos que este algoritmo puede ofrecer mejoras significativas en redes muy densas, cuando la modularidad no es una medida confiable para optimizar. En esos casos la simulación multiagente escala bien al nivel de complejidad necesario para modelar las interacciones de la red. Además, este algoritmo siempre converge al mismo valor, por lo que no es necesario ejecutarlo varias veces para obtener un resultado válido. Además, debido a los principios intuitivos sobre los que se basa el algoritmo, creemos que los resultados obtenidos con él pueden ser fácilmente interpretables.

La investigación futura tendrá como objetivo aplicar Borgia Clustering en redes a gran escala, estudiando las limitaciones y requisitos para realizar el proceso de detección de la comunidad de manera eficiente, en términos de memoria y tiempo de ejecución.

VII. AGRADECIMIENTOS

El trabajo de Javier Fumanal Idocin y Humberto Bustince ha sido financiado por el proyecto PID2019-108392GB-I00 (AEI/10.13039/ 501100011033).

El trabajo de Oscar Cordón ha sido financiado por el gobierno de España, EXASOCO (PGC2018-101216-B-I00), incluyendo fondos de desarrollo regional europeo (ERDF).

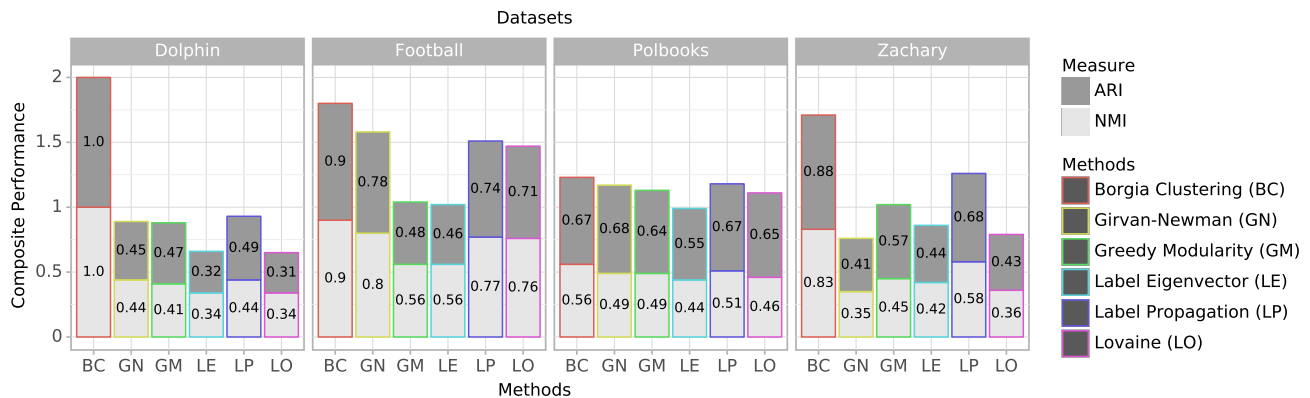


Figura 2: Comparación de diferentes algoritmos de detección de comunidades. Hemos comparado el algoritmo Borgia Clustering con tres métodos de optimización de modularidad: Girvan-Newman, optimización de modularidad voraz de Newman y el algoritmo Lovaina. También lo hemos comparado con el uso de valores propios de matrices para detectar comunidades y propagación de etiquetas.

La investigación de Amparo Alonso Betanzos ha sido parcialmente financiado por el Ministerio de Economía y Competitividad de España (TIN2015-65069-C2-1-R), por los fondos europeos FEDER y por la Consellería de Industria de la Xunta de Galicia (GRC2014 /035).

El trabajo de María Minárová ha sido financiado por los proyectos APVV-17-0066, and APVV-18-0052.

REFERENCIAS

- [1] John Scott. Social network analysis. *Sociology*, 22(1):109–127, 1988.
- [2] Stanley Wasserman, Katherine Faust, et al. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.
- [3] Stephen P Borgatti, Ajay Mehra, Daniel J Brass, and Giuseppe Labianca. Network analysis in the social sciences. *Science*, 323(5916):892–895, 2009.
- [4] Steve Horvath. *Weighted network analysis: applications in genomics and systems biology*. Springer Science & Business Media, 2011.
- [5] Ilan Lobel and Evan Sadler. Information diffusion in networks through social learning. *Theoretical Economics*, 10(3):807–851, 2015.
- [6] Julian McAuley and Jure Leskovec. Discovering social circles in ego networks. *ACM Transactions on Knowledge Discovery from Data*, 8(1):4, 2014.
- [7] Liat Sless, Noam Hazon, Sarit Kraus, and Michael Wooldridge. Forming k coalitions and facilitating relationships in social networks. *Artificial Intelligence*, 259:217 – 245, 2018.
- [8] Wei Yu and Shijun Li. Recommender systems based on multiple social networks correlation. *Future Generation Computer Systems*, 87:312 – 327, 2018.
- [9] Jordi Delgado. Emergence of social conventions in complex networks. *Artificial Intelligence*, 141(1):171 – 185, 2002.
- [10] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, Jun 2002.
- [11] Haijun Zhou. Distance, dissimilarity index, and network community structure. *Physical review E*, 67(6):061901, 2003.
- [12] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814, 2005.
- [13] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.
- [14] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), oct 2008.
- [15] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.
- [16] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.
- [17] W.E. Wright. Gravitational clustering. *Pattern Recognition*, 9(3):151 – 166, 1977.
- [18] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PLOS ONE*, 9(6):1–12, 06 2014.
- [19] José Catalàn Deus. El príncipe del renacimiento. *Debate*, 26:90, 2008.
- [20] Robert K Merton. The Matthew effect in science: The reward and communication systems of science are considered. *Science*, 159(3810):56–63, 1968.
- [21] Alejandro Bancalari. *Orbe romano e imperio global: la romanización desde Augusto a Caracalla*. Editorial Universitaria de Chile, 2007.
- [22] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.
- [23] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.
- [24] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.
- [25] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.
- [26] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, Sep 2006.
- [27] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.
- [28] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(Dec):583–617, 2002.
- [29] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.