RNPS No. 2143
ISSN 2072-6260
Versión Digital

REPORTE TÉCNICO
Minería de Datos

SERIE GRIS

Análisis de los métodos para la detección de anomalías en redes sociales

Mario Alfonso Prado-Romero
y Andrés Gago-Alonso

RT_030 **marzo 2015**

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

RNPS No. 2143
ISSN 2072-6260
Versión Digital

SERIE GRIS

REPORTE TÉCNICO
Minería de Datos

Análisis de los métodos para la detección de anomalías en redes sociales

Mario Alfonso Prado-Romero
y Andrés Gago-Alonso

RT_030 **marzo 2015**

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

Siglas y acrónimos

AGM: Minería de grupos anómalos (en inglés *Abnormal Group Mining*)
BBC: Agrupamiento mediante burbujas de Bregman (en inglés *Bregman Bubble Clustering*)
CAD: Detección condicional de anomalías (en inglés *Conditional Anomaly Detection*)
DBSCAN: Agrupación espacial basada en densidad de aplicaciones con ruido (en inglés *Density-Based Spatial Clustering of Applications with Noise*)
DGRADE: Enumeración del gradiente de densidad (en inglés *Density Gradient Enumeration*)
DSEA: Algoritmo llamado detección de subestructuras anómalas (en inglés *Anomalous Substructure Detection*)
DSGA: Algoritmo llamado detección de subgrafos anómalos (en inglés *Anomalous Subgraph Detection*)
FRIMA: Algoritmo para la minería de *itemsets* frecuentes y raros (en inglés *Frequent Rare Itemset Mining Algorithm*)
iForest: Algoritmo de detección de anomalías basado en aislamiento
LOF: Factor de atipicidad local (en inglés *Local Outlier Factor*)
MIS: Soporte mínimo de un ítem (en inglés *Minimum Item Support*)
OutRank: Algoritmo de detección de anomalías
OddBall: Algoritmo para la detección de anomalías colectivas en grafos
RARMA: Algoritmo para la minería de reglas de asociación raras (en inglés *Rare Asociation Rule Mining Algorithm*)
SCiForest: Algoritmo de detección de anomalías basado en iForest

Notaciones

A	Lista de atributos en D
A'	Índices de ψ_A atributos de A
A_j^*	Conjunto de valores del j-ésimo atributo de A en el conjunto D'
a	Un atributo
a^*	Un valor del atributo a
$a_j(x)$	j-ésimo atributo de x
$adj(v)$	Conjunto de vértices adyacentes al vértice v
B	Un conjunto
b	Constante que representa un porciento de los elementos de D
$b(S)$	Función que divide un conjunto S en dos subconjuntos disjuntos
C	Agrupamiento que contiene subconjuntos de D
c_i	i-ésimo grupo de un agrupamiento
$con_i(v)$	Conectividad del vértice v en la iteración i
$conf(.,.)$	Función que determina la confianza de una regla en una base de datos dada.
$cov(.,.)$	Covarianza entre dos elementos
D	Dominio de aplicación, $D \subset U$
D'	Una muestra de elementos de D
D_B	Base de datos transaccional
D_l	Subconjunto de D
D_r	Subconjunto de D
D'_ρ	Un conjunto de valores reales obtenido proyectando D' en un hiperplano ρ
D'^l_ρ	Un subconjunto de D'_ρ
D'^r_ρ	Un subconjunto de D'_ρ
d	Cantidad de dimensiones del universo
$d_\Phi(x,y)$	Divergencia de Bregman
$d(x,y)$	Función de distancia entre dos elementos de D
$d_k(x)$	k-distancia de un elemento x en D
$d_a(x,y)$	Distancia de accesibilidad de un elemento x con respecto a un elemento y
e	Constante de Euler
E	Colección de aristas de un grafo
$E(h(x))$	Promedio de $h(x)$ en un bosque
F	Un bosque
G	Un grafo
G_d	Grafo dinámico
G_{t_i}	Instantánea
$\mathbb{G}$	El espacio de todos los grafos
$\mathbb{G}_D$	El espacio de todos los grafos dinámicos
$\mathbb{G}_L$	El espacio de todos los grafos etiquetados
h_{lim}	Limite de altura
$h(x)$	Altura del elemento x en un árbol
$H(i)$	i-ésimo número armónico
I_S, I_{S_x}, I_{S_y}	*Itemsets*
$I_{S_x} \Rightarrow I_{S_y}$	Regla de asociación
k	Constante entera
l	Longitud de un camino medida hasta el momento
$L(\psi)$	Longitud estimada del camino desde la raíz hasta una hoja en un árbol binario de tamaño ψ

M_{con}	Vector que contiene el valor de conectividad para cada elemento del conjunto
M_{cor}	Matriz de correlación
M_d	Matriz que contiene las distancias entre todos los elementos de D
M_s	Matriz que contiene la similitud entre los elementos de D
M_{trans}	Matriz de transición
M_{vt}	Matriz que almacena el valor de una característica en cada instante de tiempo para cada vértice de un grafo dinámico
M_w	Submatriz de M_{vt} que representa una ventana de tiempo sobre esta
m	dimension del universo
m_p	Cantidad mínima necesaria de elementos en la ϵ-vecindad de un elemento para ser considerado un punto núcleo por el algoritmo DBSCAN
n	Cantidad de elementos de D
n'	Cantidad de elementos de las muestras tomadas de D
n_a	Cantidad de atributos seleccionados de A
n_f	Cantidad de características que conforman el sumario de un grafo
n_t	Cantidad de instantes de tiempo en un grafo dinámico
n_w	Dimensión de la ventana de tiempo
n_ρ	Cantidad de hiperplanos a considerar
p	Punto divisorio de un conjunto
$Q_b(C, R)$	Función de costo utilizada por BBC para medir la calidad de un agrupamiento
R	Conjunto de representantes (centroides) de los grupos de C
r_i	Representante de c_i
$s(x, y)$	Función de similitud entre dos elementos
$s_g(x, c)$	Función de similitud entre un elemento y un grupo
$s_c(c_1, c_2)$	Función de similitud entre dos grupos
$s_h(x, \psi)$	Función que describe la similitud entre el promedio de la altura de un elemento en un bosque de árboles de aislamiento y la altura esperada
$s_{_I}$	Función que mide la similitud interna de un grupo
s_t	Función que extrae los sumarios de dos grafos etiquetados y compara su similitud
$Sup(.,.)$	Función que determina el soporte de una regla de asociación en una base de datos
s_{χ_w}	Función que extrae el sumario del comportamiento de un grafo dinámico en un momento dado y el sumario de su comportamiento anterior a ese momento, para luego compararlos
t	Cantidad de árboles
T_i	i-ésimo *iTree* de F
U	Universo, es un espacio d-dimensional
V	Colección de vértices de un grafo
v	Un vértice de un grafo
$\|v\|$	Grado del nodo v
$\overrightarrow{v}_j$	Vector que representa el comportamiento de un grafo dinámico en el momento j
$\overrightarrow{v}_r$	Vector que representa un resumen del comportamiento de un grafo dinámico antes del momento j
x	Un elemento
x_i	i-ésima componente de x
y	Un elemento
y_i	i-ésima componente de y
α_j	Un coeficiente en el intervalo [-1, 1]
β	Una constante

δ	Constante que representa un umbral de anomalía
ϵ	Constante que pertenece a $\mathbb{R}^+$
$\nu(x)$	Vecindad del elemento x
$\nu_k(x)$	d_k-vecindad de un elemento x en D
$\nu_\epsilon(x)$	ϵ-vecindad de un elemento x en D
ρ	Un hiperplano
ϱ	Factor de suavizado
$\sigma(.)$	Desviación estándar
τ	Cantidad de elementos de un agrupamiento
τ'	Cantidad de elementos en cada grupo de un agrupamiento
χ_w	Grado de atipicidad de un grafo dinámico en un momento de tiempo
$\setminus$	Operador de diferencia de conjuntos
$\cup$	Operador de unión entre conjuntos
$\in$	Operador de pertenencia
$\propto$	Operador de proporcionalidad

Tabla de contenido

Lista de figuras

Análisis de los métodos para la detección de anomalías en redes sociales

Mario Alfonso Prado-Romero y Andrés Gago-Alonso

Equipo de Investigaciones de Minería de Datos, Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV), La Habana, Cuba
{mprado,agago}@cenatav.co.cu

RT_030, Serie Gris, CENATAV
Aceptado: 4 de Febrero de 2015

Resumen. La detección de anomalías es un importante problema que ha sido tratado en diversos dominios de aplicación como la medicina, el procesamiento de imágenes y la detección de intrusos, entre otros. Debido al reciente aumento en el interés por las redes sociales, la detección de anomalías en ellas se ha transformado en un tema de gran interés. En este trabajo se definen los principales tipos de anomalías existentes en las redes sociales y se muestra una taxonomía de los mismos, en la que se incluyen algunos poco estudiados. Además, se brinda un análisis estructurado de varias de las técnicas desarrolladas para su detección, con descripciones detalladas de su funcionamiento. Por último se mencionan varias direcciones que se consideran prometedoras para futuras investigaciones en este dominio.

Palabras clave: anomalías, redes sociales, anomalías puntuales, anomalías agrupadas, anomalías colectivas, anomalías contextuales, reglas raras, técnicas de detección de anomalías.

Abstract. Anomaly detection is an important problem that has been researched within diverse application domains such as medicine, image processing and intrusion detection, among others. Due to the recent increase in interest about social networks, anomaly detection in this field has become a topic of great interest. In this work, the main types of anomalies existing in social networks are defined; moreover a taxonomy about them, containing little studied types of anomalies, is provided. Also, a structured analysis about many of the existing anomaly detection techniques is given, including a detailed description about its behavior. Finally, some promising directions for future research are mentioned.

Keywords: anomalies, social networks, point anomalies, clustered anomalies, collective anomalies, contextual anomalies, rare rules, anomaly detection techniques.

1. Introducción

Las personas en la actualidad se encuentran más interconectadas que nunca antes y las redes sociales forman parte de la vida de muchas de ellas, siendo algunos de los ejemplos más obvios Facebook y Twitter. Sin embargo, estas redes contienen entidades y relaciones entre las mismas, por lo que pueden ser utilizadas para modelar diversos fenómenos como transacciones bancarias, subastas online, llamadas telefónicas y blogs. En la actualidad se ha incrementado el interés en su análisis y resulta de especial importancia la capacidad de detectar anomalías en ellas.

La detección de anomalías trata el problema de identificar patrones en los datos que no se corresponden con una noción establecida de comportamiento normal [1]. Los patrones antes mencionados, son comúnmente llamados anomalías, *outliers*, observaciones discordantes, excepciones, entre otros nombres, en dependencia del dominio de aplicación en que se hallen. La importancia de la detección de anomalías se debe a que estas se traducen en información significativa, incluso crítica en ocasiones, que se puede emplear en gran variedad de aplicaciones. Por ejemplo, se utiliza en la detección de fraude en subastas online [2], en la identificación de transacciones bancarias que podrían enmascarar lavado de dinero [3] y en la detección de intrusiones en redes de telecomunicaciones [4], entre otros dominios de aplicación.

En el presente reporte, se muestra un breve análisis de las principales técnicas existentes para la detección de anomalías en redes sociales. Se tuvieron en cuenta tanto las técnicas diseñadas específicamente para este fin, como otras más generales que pueden ser aplicadas en este dominio. Se ha tratado de identificar las características fundamentales de las distintas técnicas de detección de anomalías y clasificarlas según estas permitiendo una visión estructurada de las mismas.

1.1. ¿Que son las anomalías?

Las anomalías son aquellos elementos pertenecientes a un conjunto de datos, cuyo comportamiento no se corresponden con una noción de comportamiento normal existente en dicho conjunto. Sin embargo en muchos casos no existen modelos que describan qué es lo normal, dificultando la detección de anomalías [5].

Las anomalías pueden ser inducidas en los datos por diversas razones, entre las que se encuentran actividades maliciosas, errores de medición, cambios en el entorno de los datos y errores humanos. Por ello, es común que la detección de anomalías se asocie con la eliminación de ruido y la acomodación de ruido en los datos, aunque las dos, no sean lo mismo. La diferencia fundamental entre ambas radica en que el ruido es un fenómeno en los datos que no es de interés para los analistas, pero que actúa como un obstáculo en el análisis [1], mientras que las anomalías son un fenómeno de gran interés para los analistas [6].

Un tópico relacionado con la detección de anomalías es la detección de novedades [7] que trata el problema de identificar patrones en los datos que no han sido observados con anterioridad. La diferencia entre los patrones novedosos y las anomalías es que los primeros suelen ser incorporados al modelo de comportamiento normal después de su detección.

1.2. Organización de este trabajo

Este reporte está organizado de modo que permita realizar un análisis estructurado de los distintos tipos de anomalías existentes y las técnicas para su detección. En la sección 2, se mencionan las dificultades para definir el concepto de anomalía y se definen los tipos de anomalías existentes. En las secciones 3, 4, 5 y 6, se analizan las técnicas utilizadas para detectar anomalías puntuales, agrupadas, colectivas y contextuales, respectivamente. En la sección 7, se analizan las técnicas de detección de reglas raras. En las secciones 8 y 9 se describen las bases de datos y las medidas de calidad utilizadas por los algoritmos analizados en este reporte durante las experimentaciones. En la sección 10 se muestra una tabla comparativa de los algoritmos analizados y en la sección 11, se concluye el reporte. Además en el anexo 1, se brindan los conceptos fundamentales para comprender las técnicas expuestas en este reporte.

2. El problema de la detección de anomalías

En esta sección se expondrán algunas de las definiciones de anomalía más utilizadas y se explicarán brevemente los principales tipos de anomalías existentes.

2.1. Definiciones de anomalía

En las aplicaciones donde se buscan anomalías, es necesario definir qué es un "comportamiento normal". Este aspecto en ocasiones se dificulta por las siguientes razones:

- La frontera que separa lo normal de lo anómalo, no es clara en muchos casos.
- Cuando las anomalías son el resultado de acciones maliciosas, los individuos maliciosos tratan de hacerlas parecer normales.
- En muchos dominios de aplicación, lo considerado como "comportamiento normal" evoluciona con el tiempo y la noción que se tiene de él, en un momento determinado, puede no ser lo suficientemente representativa en el futuro.
- La noción exacta de anomalía es diferente entre dominios de aplicación.

Las definiciones de anomalía existentes suelen ser poco formales y abstractas, para tratar de abarcar todos los tipos de anomalías existentes, o demasiado específicas restringiéndose solo a un determinado tipo de anomalía en un cierto dominio de aplicación.

Según Hawkins [8], una anomalía es una observación que se desvía tanto de las demás que se vuelve sospechosa de ser generada por un mecanismo diferente. La mayoría de los trabajos en el campo de la estadística utiliza esta definición.

Debido a que en muchos de los dominios de aplicación no se conoce la distribución de los datos y tratar de ajustarlos a una distribución conocida es costoso y complejo, se han buscado otras definiciones de anomalía. La definición de Liu et al. [9], se basa en las características fundamentales de las anomalías, definiéndolas como aquellos elementos que son escasos y diferentes en comparación con los elementos normales.

Existe una gran cantidad de definiciones de anomalías más específicas que las antes mencionadas, en dependencia del modo en que se modele el problema de detección de anomalías, o del dominio de aplicación en el que van a ser utilizadas. Algunas de ellas se podrán ver más adelante cuando se traten las técnicas de detección de anomalías.

2.2. Tipos de anomalías

A continuación se expondrán los principales tipos de anomalías existentes, su definición y algunos ejemplos de dominios donde se aplica su detección. En la figura 1 se propone una taxonomía para los distintos tipos de anomalías. En primer lugar, se propone dividir las anomalías en dos clases: las que se definen a partir de la similitud entre los elementos y las que se definen a partir de la atipicidad de estos. La primera clase requiere que este definida una función de similitud sobre los datos, mientras que la segunda solo necesita conocer si dos elementos son iguales o del mismo tipo. Los tipos de anomalías que pertenecen a cada clase se analizan más adelante.

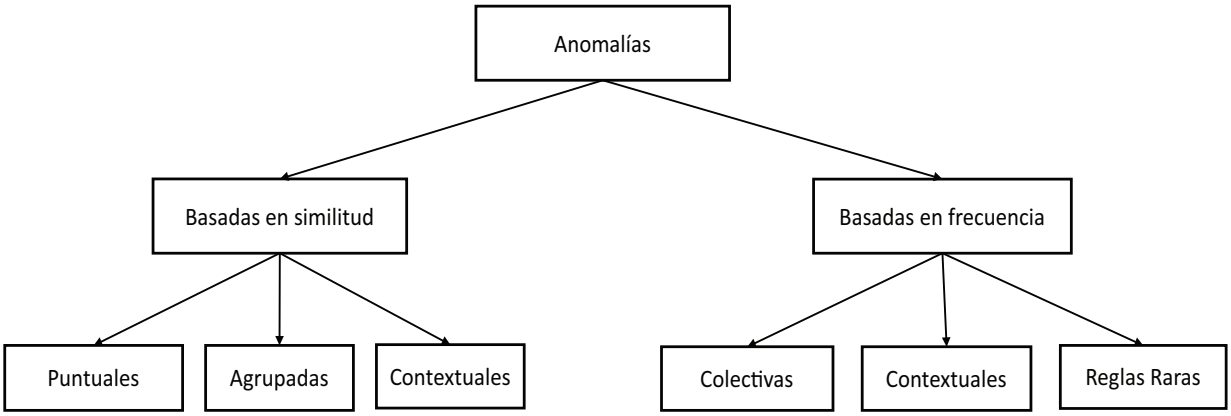

Fig. 1. Taxonomía de los tipos de anomalías presentes en las redes sociales.

2.2.1. Anomalías puntuales

Un elemento x que pertenece a un conjunto de datos D puede considerarse una anomalía puntual si son sus características individuales, y no su relación con otros elementos, las que hacen que sea considerado una anomalía. Este es el tipo más simple de anomalía y el centro de atención de la mayoría de las investigaciones en el área de la detección de elementos anómalos [1,10].

Las anomalías puntuales se pueden definir de modo global o local [11,9]. Se consideran globales cuando están definidas con relación a todo el conjunto de datos. Podemos basarnos en la definición de Liu et al [9], para definir las anomalías puntuales globales del modo siguiente.

Definición 1 (Anomalía puntual global). *Sea D un dominio de aplicación, s una función de similitud entre dos elementos y δ un umbral de similitud. Entonces un elemento $x \in D$ se considera anómalo si $\forall y \in D \setminus \{x\},\ s(x,y) < \delta$.*

Esta definición no abarca las anomalías dispersas que no son anómalas con respecto a todo el conjunto de datos, sino que lo son solo con respecto a la "vecindad" en la que se encuentran. Este tipo de elementos son llamados anomalías puntuales locales. Para definir este tipo de anomalías es necesario, en primer lugar, definir qué es la vecindad de un elemento en un conjunto.

Definición 2 (Función de vecindad de un elemento en un conjunto). *Sea D un dominio de aplicación, $x \in D$ un elemento del mismo. Entonces se le llama función de vecindad a una función $\nu : D \longrightarrow 2^D$, tal que $x \notin \nu(x)$. La vecindad de x en D se denota $\nu(x)$ y a los elementos que pertenecen a ella se les llama vecinos de x.*

El concepto anterior posibilita definir las anomalías puntuales locales. De modo intuitivo estas anomalías son aquellos elementos que no son tan similares a sus vecinos como estos a los miembros de sus respectivas vecindades. En la definición 3 se formaliza este concepto.

Definición 3 (Anomalía puntual local). *Sea D un dominio de aplicación, $x \in D$ uno de sus elementos, ν una función de vecindad, s_g una función de similitud entre un elemento y un conjunto, δ un umbral de similitud, y $s_\nu : D \longrightarrow [0,1]$ una función que determina la similitud entre el valor $s_g(x, \nu(x))$ y los valores $s_g(y, \nu(y))$, para las $y \in \nu(x)$. Entonces, se considera a x una anomalía puntual local si: $s_v(x) < \delta$.*

Una ventaja de la definición anterior es que los elementos que cumplen con la definición de anomalía puntual global (definición 1) también cumplen con la definición local (definición 3). Es importante señalar que en algunos dominios de aplicación resulta complicado definir la vecindad de un elemento, dificultando la aplicación de la definición 3.

En la figura 2, C_1 y C_2 son subconjuntos de un conjunto de puntos en un espacio de dos dimensiones. La similitud entre los elementos de la figura está dada por la inversa de la distancia entre ellos, es decir a menor distancia mayor similitud. El elemento a_g es una anomalía puntual global, pues se encuentra alejado de todos los elementos del conjunto y a_l es una anomalía puntual local, ya que aunque la distancia a sus elementos más cercanos sería normal en C_2, es lejana en comparación con la distancia entre los elementos de C_1.

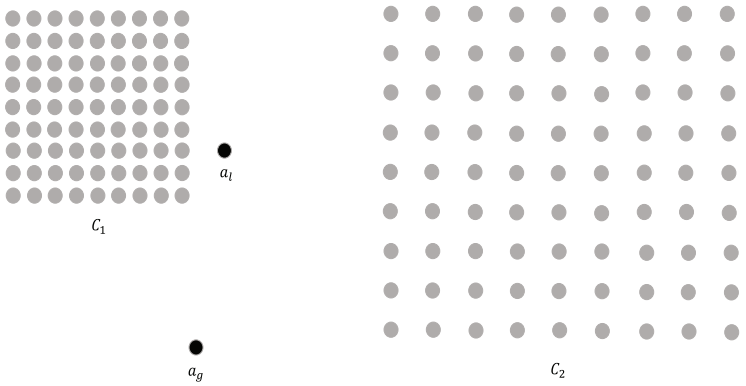

Fig. 2. Ejemplo de anomalías puntuales en un conjunto de datos bidimensionales donde la cercanía de los elementos indica su similitud. El elemento a_g representa una anomalía puntual global y el elemento a_l una anomalía puntual local.

La detección de anomalías puntuales en redes sociales puede ser utilizada para la detección de fraude en redes telefónicas, las cuales son un caso particular de redes sociales. El descubrimiento de anomalías puntuales locales puede ser utilizado para encontrar aquellos usuarios del sector doméstico que reciben muchas llamadas telefónicas y no emiten ninguna, lo cual puede ser indicativo de ciertos tipos de fraude como el llamado *bypass*. Sin embargo, si estos usuarios también emiten llamadas resulta difícil hallarlos utilizando detección de anomalías puntuales. Para resolver el problema anterior es posible utilizar detección de anomalías puntuales locales con el fin de descubrir aquellos usuarios que reciben una cantidad de llamadas muy superior a la de otros usuarios que reciben aproximadamente la misma cantidad de llamadas que ellos.

2.2.2. Anomalías agrupadas

Las anomalías agrupadas (en inglés, *clustered anomalies* u *outlying clusters*) son pequeños conjuntos $c_i \subset D$ de elementos similares entre sí y distintos en comparación con los datos de $D \setminus c_i$.

Las anomalías agrupadas, al igual que las dispersas, pueden ser definidas de modo global o local. Se puede utilizar la definición de Liu et al. [9], para definir los grupos anómalos globales del siguiente modo:

Definición 4 (Anomalías agrupadas globales). *Sea D un dominio de aplicación, s una función de similitud entre dos elementos, τ' un umbral que determina si la cantidad de elementos de un conjunto es escasa y δ un umbral que determina cuando dos elementos se consideran similares. Entonces un grupo $c \subset D$ se considera anómalo si:*

- $c \subset D$,
- $|c| \leq \tau'$,
- $\forall x \in c, y \in c,\ s(x,y) \geq \delta$,
- $\forall x \in c, y \in D \setminus c,\ s(x,y) < \delta$.

Las dos primeras condiciones de la definición 4 garantizan que los elementos anómalos sean escasos en relación con los elementos normales. La tercera condición exige que los elementos en un mismo grupo anómalo sean similares entre ellos, dándole sentido al agrupamiento. Por último la cuarta condición exige que los elementos del grupo anómalo sean distintos del resto de los elementos. La cuarta condición requiere que x sea una anomalía puntual global en el conjunto $D \setminus c$. Esto puede ser un poco restrictivo en algunos casos, pues obliga a que cada elemento del conjunto sea distinto a todos los elementos fuera de él, por ello es conveniente una definición menos estricta que incluya a los grupos de elementos anómalos con respecto a la vecindad en la que se encuentran.

Definición 5 (Anomalías agrupadas locales). *Sea D un dominio de aplicación, s una función de similitud entre dos elementos, τ' un umbral que determina si la cantidad de elementos de un conjunto es escasa, δ un umbral de similitud y $\chi : D \times 2^D \longrightarrow \{0,1\}$ una función que retorna 1 si un elemento es una anomalía puntual local en un conjunto dado y 0 en caso contrario. Entonces un grupo $c \subset D$ se considera anómalo si:*

- $c \subset D$,
- $|c| \leq \tau'$,
- $\forall x \in c, y \in c,\ s(x,y) \geq \delta$,
- $\forall x \in c, \chi(x, (D \setminus c)) = 1$.

Se puede observar un ejemplo de las dos clases de anomalías agrupadas mencionadas anteriormente, en la figura 3.

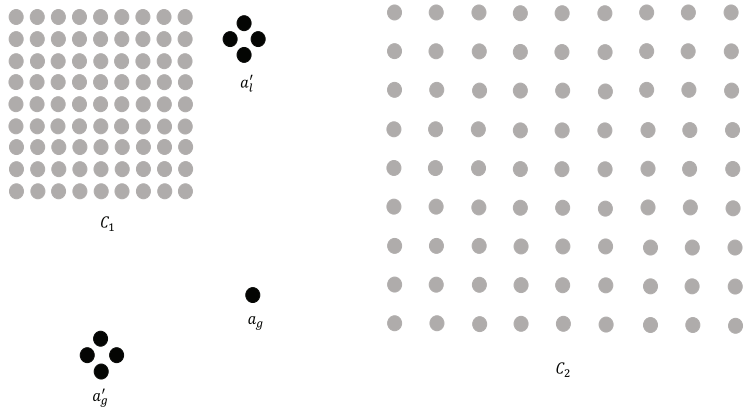

Fig. 3. Ejemplo de anomalías agrupadas y puntuales en un conjunto de datos bidimensionales donde la cercanía de los elementos indica su similitud. El elemento a_g es una anomalía puntual global, mientras que el grupo a'_g conforma una anomalía agrupada global y el grupo a'_l una anomalía agrupada local.

En la figura 3, C_1 y C_2 son subconjuntos de puntos en un espacio de dos dimensiones donde la similitud entre los elementos es la inversa de su distancia. a_g es una anomalía dispersa global.

El grupo a'_g es una anomalía agrupada global pues está conformado por pocos elementos y estos se encuentran alejados de los elementos fuera de él. El grupo a'_l es una anomalía agrupada local debido a que está formado por pocos elementos y estos son anomalías puntuales locales con respecto a los elementos fuera de a'_l.

En muchos dominios de aplicación la ocurrencia de una anomalía ocasional puede ser admisible, sin embargo sería imprudente ignorar la ocurrencia de varias anomalías similares. Un grupo de anomalías parecidas puede ser indicador de un error en las mediciones o de la existencia de una clase de elementos que no se tuvo en cuenta con anterioridad.

Determinar la existencia de este tipo de anomalías resulta de gran importancia en la detección de fraude, donde la identificación de defraudadores frecuentes es mucho más significativa que la de casos aislados y puede prevenir grandes pérdidas. Normalmente un usuario fraudulento cambia ligeramente su modus operandi entre la ejecución de un acto ilegal y otro. Si tenemos en cuenta lo anterior, aunque los defraudadores cambien su identidad, tanto sus perfiles como las actividades que llevan a cabo tendrán muchas similitudes, por lo que pueden ser detectados como un grupo. A casos como este, se les debe prestar gran atención pues se trata de defraudadores reincidentes.

Consideremos la manipulación de precios en la bolsa de valores. Normalmente se asume que el comportamiento del comercio de valores para individuos distintos es mayormente independiente, sin embargo un corredor corrupto puede abusar de múltiples cuentas para manipular los precios del mercado y obtener ganancias ilícitas [12]. Utilizando detección de anomalías agrupadas es posible obtener grupos de cuentas que exhiben comportamientos similares durante un considerable número de días e identificarlas como sospechosas de haber sido manipuladas.

En las redes sociales se pueden encontrar pequeñas comunidades, casi triviales, donde los individuos que las componen poseen una gran similitud entre ellos. A medida que estas comunidades incrementan su tamaño, los individuos comienzan a mezclarse con la red en general siendo menos similares entre ellos y el grupo comienza a comportarse menos como una comunidad [13]. La detección de anomalías agrupadas puede ser utilizada para detectar grupos dentro de la red donde los individuos son muy similares entre sí, permitiendo así, la detección de comunidades en redes sociales.

Puede argumentarse que las anomalías agrupadas no son más que grupos de anomalías puntuales y por tanto pueden detectarse utilizando las técnicas de detección habituales y luego agrupar los elementos detectados. Sin embargo, los elementos que conforman los grupos anómalos son capaces de engañar a la mayoría de las técnicas de detección de anomalías puntuales, debido a la cercanía existente entre ellos.

2.2.3. Anomalías colectivas

Una colección de elementos relacionados entre ellos, se llama anomalía colectiva si es anómala con respecto al conjunto de datos al que pertenece [1]. Los elementos en ella no son necesariamente anómalos por sí mismos, si no que su condición de anomalía está dada por la relación existente entre ellos [9,14].

Tres de los tipos de relaciones que han sido utilizadas en la detección de anomalías colectivas son las secuenciales, las espaciales y las de conexión. Los elementos son seleccionados juntos para formar una anomalía colectiva debido a las relaciones existentes entre ellos y no debido a su similitud [15]. Las anomalías puntuales pueden estar presentes en cualquier conjunto de datos, sin embargo, las anomalías colectivas solo pueden ocurrir en conjuntos de datos donde existen relaciones entre los elementos [1].

La detección de lavado de dinero es uno de los dominios de aplicación de la detección de anomalías colectivas. En la figura 4 se representa la red de transacciones bancarias como un grafo donde los vértices son las cuentas y las aristas representan transacciones entre estas. Se encuentran sombreadas en gris las cuentas sospechosas de participar en el delito de lavado de dinero. Es importante señalar que estas cuentas por sí mismas no tienen nada sospechoso, lo que las hace anómalas es la relación existente entre ellas, donde una cuenta envía dinero a la otra a través de múltiples intermediarios. Las posibles cuentas fraudulentas pueden tener relación con otras cuentas legítimas, como se muestra en la figura 4.

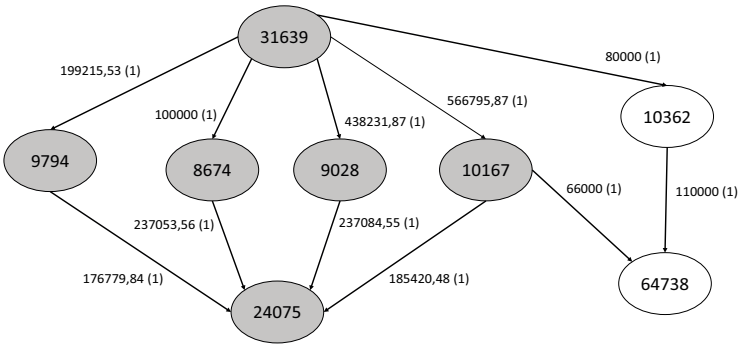

Fig. 4. Ejemplo de anomalía colectiva en una red bancaria. El patrón de diamante formado por los vértices grises y las aristas que los conectan, se considera una anomalía colectiva, pues puede indicar lavado de dinero. Los vértices blancos representan cuentas legítimas.

En la figura 4 se puede observar un grafo de transacciones sospechosas conectado con otras actividades, posiblemente legales. Las etiquetas de los vértices son los identificadores de las cuentas y las de las aristas contienen el número de transacciones entre las dos cuentas y la cantidad total transferida.

2.2.4. Anomalías contextuales

Un elemento es llamado anomalía contextual o condicional, cuando su comportamiento se considera anómalo, solo en un contexto específico [16,1].

La noción de contexto es inducida por la estructura del conjunto de datos y tiene que ser especificada como parte de la formulación del problema. Para determinar si un elemento es una anomalía contextual son necesarias dos clases de atributos [1,14]:

1) **Atributos contextuales.** Se utilizan para determinar el contexto o vecindario de un elemento. Por ejemplo la comunidad de un individuo en una red social representa un atributo contextual.
2) **Atributos de comportamiento.** Definen las características no contextuales de un elemento. El salario de un individuo en una red social es ejemplo de atributos de comportamiento, debido a que no aporta información sobre su contexto en la red.

Un elemento es considerado anómalo si los valores de sus atributos de comportamiento son anómalos en el contexto definido por sus atributos contextuales.

En [17] se puede ver un ejemplo de detección de anomalías en una red de amistades donde cada nodo denota una persona y cada arista una relación de amistad entre dos personas. El salario de una persona se muestra como un número adjunto a cada nodo. Esta red se puede dividir en

dos comunidades, una de altos ingresos y otra de bajos ingresos. En este ejemplo v_6 es anómalo en su comunidad debido a que solo se relaciona con personas de altos ingresos, sin embargo él es de bajos ingresos (ver figura 5). Este usuario podría representar a un emprendedor o a una persona que se ha mudado a un barrio rico.

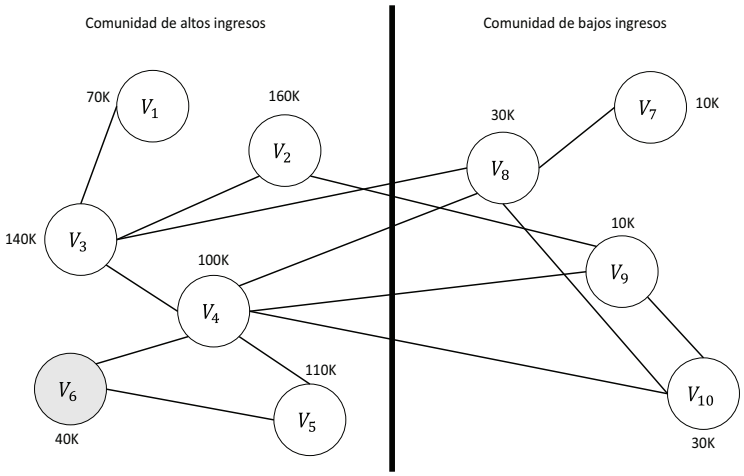

Fig. 5. Ejemplo de anomalía contextual en una red social donde se conocen los ingresos de cada persona. El usuario V_6 se considera anómalo debido a que sus relaciones lo sitúan como miembro de la comunidad de altos ingresos, pese a que sus ingresos son bajos.

La elección de utilizar técnicas de detección de anomalías contextuales está dada por la significación de dichas anomalías en el dominio de aplicación siendo muy importante la disponibilidad de atributos contextuales. En los problemas donde definir un contexto es un proceso sencillo, tiene sentido aplicar la detección de anomalías contextuales, mientras que en los problemas donde resulta muy complejo definir un contexto, se dificulta notablemente la aplicación de estas técnicas [1].

Es importante destacar que tanto las anomalías puntuales, agrupadas o colectivas pueden ser también anomalías contextuales si se analizan en relación a un contexto. El problema de detectar cualquiera de los tipos de anomalías mencionados anteriormente puede ser transformado en un problema de detección de anomalías contextuales si se le incorpora información contextual.

2.2.5. Reglas raras

La mayor parte de los trabajos sobre minería de reglas de asociación se dedican a encontrar reglas formadas por *itemsets* frecuentes. Recientemente se ha comenzado a prestar más atención a las reglas de asociación compuestas por *itemsets* infrecuentes, debido a que estas representan comportamientos poco comunes que pueden ser de mucho interés para los analistas. A este tipo de reglas se les suele denominar reglas raras.

Definición 6 (Regla rara). *Sea D un dominio de aplicación, D_B una base de datos en él, I_{S_x} e I_{S_y} dos transacciones en D_B, $I_{S_x} \Rightarrow I_{S_y}$ una regla de asociación, δ_{sup} y δ_{conf} dos umbrales. Entonces se dice que $I_{S_x} \Rightarrow I_{S_y}$ es una regla rara si:*

i) $Sup(I_{S_x} \Rightarrow I_{S_y}, D_B) < \delta_{sup}$,
ii) $Conf(I_{S_x} \Rightarrow I_{S_y}, D_B) \geq \delta_{conf}$.

Obtener reglas de asociación raras a partir de una base de datos transaccional, resulta beneficioso en muchos dominios de aplicación, pues permite modelar el comportamiento de elementos atípicos en la base de datos. Un ejemplo de aplicación es la detección de fraude en telecomunicaciones, donde este tipo de reglas pueden ser utilizadas para modelar el comportamiento de los defraudadores, puesto que estos son usuarios poco comunes en la red, con características muy particulares en su comportamiento, las cuales los diferencian de los usuarios legítimos. También es posible aplicar la minería de reglas raras en los sistemas de detección de intrusos, como se menciona en [18]. En [19] se propone el uso de esta técnica para extraer de bases de datos educacionales, conocimiento sobre los estudiantes con comportamientos atípicos.

Es importante destacar que las reglas raras no se corresponden con otros criterios de anomalías analizados anteriormente, ya que estas representan asociaciones atípicas entre los elementos de un conjunto de datos y en dichas asociaciones solamente se garantiza que intervienen conjuntos infrecuentes de elementos. La importancia de las reglas raras está dada porque su detección permite determinar comportamientos atípicos en conjuntos de datos en los que existe poca información para aplicar técnicas orientadas a encontrar otros tipos de anomalías.

3. Detección de anomalías puntuales

Las anomalías puntuales son el tipo más simple de anomalías y en ocasiones la detección de otros tipos puede reducirse a la detección de estas. Como consecuencia de lo anterior existen numerosas técnicas para identificar este tipo de anomalías, por lo que en este reporte se mostrarán solo algunas de las existentes. Otros trabajos que analizan las técnicas dedicadas a la detección de anomalías puntuales son [1,6,10].

3.1. Técnicas basadas en distancia

Estas técnicas no asumen ninguna distribución subyacente para los elementos, lo que les permite ser más generales que la mayoría de los métodos estadísticos y ajustarse mejor a conjuntos de datos multidimensionales. Existen varias definiciones de anomalía puntual basadas en distancia, siendo una de las más conocidas la ofrecida por Knorr [20], la cual se muestra a continuación:

Definición 7 (*DB-atípico*). *Sea b un número entre 0 y 1, ϵ un número real positivo y D un dominio de aplicación. Entonces un elemento $x \in D$ se considera $DB(b, \epsilon)$-atípico si al menos una fracción b de los elementos en D se encuentra a una distancia mayor que ϵ de x.*

Esta definición permite detectar anomalías de modo relativamente sencillo, calculando la distancia entre todos los elementos del conjunto, para luego marcar como anómalos aquellos elementos que cumplen con la definición 7. Un algoritmo como el sugerido anteriormente, tendría una complejidad temporal $O(dn)$ donde d es la dimension del dominio de aplicación y n su cantidad de elementos.

Es importante tener en cuenta que los algoritmos basados en la definición 7, no son capaces de detectar anomalías puntuales locales, pues esta trata las anomalías desde un enfoque global, al tener en cuenta la distancia entre todos los elementos del conjunto.

Estas técnicas poseen algunas desventajas, entre ellas están que suelen verse afectadas si hay un aumento significativo en la dimensionalidad de los datos, y que requieren que esté definida una distancia sobre el conjunto de datos.

3.2. Técnicas basadas en densidad

Este tipo de técnicas resultan más robustas que las técnicas basadas en distancia, aunque también resultan computacionalmente más costosas. Comúnmente suelen determinar un grado de atipicidad para cada elemento, basándose en la densidad de su vecindad y en las densidades de las vecindades de sus vecinos.

Una de las técnicas basadas en densidad más conocidas es el factor de atipicidad local, abreviado LOF (por sus siglas en inglés, *Local Outlier Factor*) [11]. Entre las ventajas de esta técnica se encuentra el posibilitar la detección de anomalías puntuales locales. Antes de definir el factor de atipicidad local de un elemento, es necesario definir algunos conceptos, entre los que se encuentra la k-distancia de un elemento.

Definición 8 (k-distancia de un elemento x en D). *Sea D un dominio de aplicación, $x \in D$ uno de sus elementos, y k un entero positivo. La k-distancia de x en D se denota como $d_k(x)$ y se define como la distancia entre x y su k-ésimo elemento más cercano en D.*

Si se analiza la definición anterior se notará que el k-ésimo elemento más cercano a x no es necesariamente único. De existir varios elementos que pudieran considerarse el k-ésimo elemento más cercano a x, entonces la distancia de ellos a x es la misma, por lo tanto el valor de d_k es único.

Definición 9 (d_k-vecindad de un elemento x). *Sea D un dominio de aplicación, $x \in D$ uno de sus elementos y d la función de distancia entre dos elementos del universo. La d_k-vecindad de x se denota $\nu_k(x)$ y se define como $\nu_k(x) = \{y \in D \setminus \{x\} | d(x,y) \leq d_k(x)\}$.*

Otro concepto importante para poder definir el LOF de un elemento, es la llamada distancia de accesibilidad entre dos elementos, la cual se define a continuación:

Definición 10 (Distancia de accesibilidad de x con respecto a y). *Sea D un dominio de aplicación, $x, y \in D$ dos elementos de él y d la función de distancia entre dos elementos del universo. Entonces la distancia de accesibilidad entre x e y se denota $d_a(x,y)$ y se define $d_a(x,y) = max(d_k(y), d(x,y))$.*

Se puede ver intuitivamente a $d_a(x,y)$ como una función de distancia entre dos elementos x e y que resulta insensible a las variaciones de x, siempre que la distancia entre x e y sea menor que la k-distancia de y. La distancia de accesibilidad entre dos elementos, es esencial para definir la densidad de accesibilidad local de un elemento, la cual se define a continuación:

Definición 11 (Densidad de accesibilidad local de un elemento). *Sea D un dominio de aplicación, $x \in D$ uno de sus elementos. La densidad de accesibilidad local de x se denota por $den(x)$ y se define como:*

$$den(x) = \frac{|\nu_k(x)|}{\sum_{y \in \nu_k(x)} d_a(x,y)}.$$

La función $den(x)$ tiene en cuenta la relación existente entre la distancia de x a los elementos $y \in \nu_k(x)$ y las respectivas k-distancias de estos elementos. El valor máximo que puede tomar $den(x)$ es el inverso del promedio de las k-distancias de los elementos que pertenecen a la d_k-vecindad de x, esto ocurre en el caso en que la distancia de x a todo $y \in \nu_k(x)$ es menor que $d_k(y)$.

Visto de modo intuitivo, a mayor valor de $den(x)$ mayor será la cercanía de x a los elementos de su vecindad en relación con la cercanía de dichos elementos a los miembros de sus respectivas vecindades, en otras palabras $\nu_k(x)$ será más densa.

Utilizando las definiciones anteriores se puede definir el factor de atipicidad local de un elemento como se muestra a continuación:

Definición 12 (Factor de Atipicidad Local de un elemento). *Sea D un dominio de aplicación. El factor de atipicidad local de un elemento $x \in D$ se define como:*

$$LOF(x) = \frac{\sum_{y \in \nu_k(x)} \frac{den(y)}{den(x)}}{|\nu_k(x)|}.$$

La definición 12, permite asignarle a un elemento x un valor que tiene en cuenta la densidad de su vecindad y las densidades de las vecindades de los elementos y tales que $y \in \nu_k(x)$. El LOF de un elemento x captura el grado en el que le llamamos anomalía, o sea podemos decir que determina el "grado de atipicidad" de x. Si se desea determinar si un elemento es anómalo o no, utilizando el LOF, solo hay que definir un δ tal que si $LOF_{\bar{k}}(x) > \delta$ entonces x se considere anómalo.

Entre las ventajas de estas técnicas está la capacidad de manejar conjuntos de datos con distintas densidades, lo que evita clasificar incorrectamente a los elementos como anómalos o a las anomalías como elementos normales. Otra ventaja es que permiten detectar anomalías locales y que, en general, la idea de los algoritmos es clara y no demasiado complicada de implementar.

Las principales desventajas de estas técnicas son el aumento del costo computacional en relación con las técnicas basadas en distancia, así como otros problemas heredados de estas, ya que las técnicas basadas en densidad utilizan la distancia en su funcionamiento.

3.3. Técnicas basadas en agrupamiento

Los algoritmos de agrupamiento pueden ser utilizados para la detección de anomalías, debido a que agrupan los elementos similares entre sí, por tanto, los elementos anómalos al ser distintos a los demás, no serían agrupados con ellos. El problema que intentan resolver los algoritmos de agrupamiento se puede definir del modo siguiente:

Definición 13 (Problema resuelto por los algoritmos de agrupamiento). *Sea D un dominio de aplicación, s una función de similitud entre dos elementos y δ un umbral de similitud. Entonces se desea encontrar un conjunto $C = \{c_1, c_2, \ldots, c_\kappa\}$ tal que:*

i) $\cup_{i=1}^{\kappa} c_i = D$,
ii) $\forall x, y \in c_i \ s(x, y) \geq \delta$,
iii) $\forall x \in c_i, y \in D \setminus c_i \ s(x, y) < \delta$.

Al realizar un agrupamiento según la definición anterior, los elementos que se encuentren en conjuntos unitarios, cumplen con la definición de anomalía puntual global (ver definición 1). Numerosos algoritmos de agrupamiento como ROCK [21], SNN [22] y DBSCAN [23], pueden ser utilizados para detectar anomalías puntuales. Algunos algoritmos realizan una relajación de las condiciones de la definición 13 ya que, en determinados casos, pueden resultar demasiado estrictas.

El algoritmo DBSCAN (por sus siglas en inglés *Density-Based Spatial Clustering of Applications with Noise*) [23] relaja las condiciones de la definición anterior, para poder detectar grupos con formas arbitrarias y además solo agrupa una parte de los elementos de D, considerando al resto como ruido. Desde el punto de vista de la detección de anomalías los elementos considerados ruido por DBSCAN serían considerados anómalos. Para comprender el funcionamiento de este algoritmo es necesario definir algunos conceptos como la ϵ-vecindad de un punto.

Definición 14 (ϵ-vecindad de un punto). *Sea D un dominio de aplicación, x un elemento de él, s una función de similitud entre dos elementos y $\epsilon \in \mathbb{R}^+$ una constante. Entonces se le llama ϵ-vecindad de x al conjunto $\nu_\epsilon(x) = \{y \in D | s(x, y) > \epsilon\}$.*

Asociada a la definición anterior se encuentra la definición de punto núcleo, la cual es muy importante para el funcionamiento de DBSCAN y se muestra a continuación:

Definición 15 (Punto núcleo). *Sea D un dominio de aplicación, x un elemento de él y $m_p \in \mathbb{Z}^+$ un umbral. Entonces se considera que x es un punto núcleo si, $|\nu_\epsilon(x)| \geq m_p$.*

El algoritmo DBSCAN realiza agrupamiento basado en la densidad utilizando los parámetros ϵ y m_p. Con estos parámetros y las definiciones anteriores se define qué se considera que un punto sea densamente alcanzable, de forma directa, desde otro punto.

Definición 16 (Densamente alcanzable de forma directa). *Sea D un dominio de aplicación, x e y dos elementos de él, $\epsilon \in \mathbb{R}^+$ y $m_p \in \mathbb{Z}^+$ dos constantes. Entonces, se dice que x es densamente alcanzable de forma directa desde y, si:*

i) $x \in \nu_\epsilon(y)$,
ii) $|\nu_\epsilon(y)| \geq m_p$ *(condición de pertenencia al núcleo).*

La definición anterior puede ser utilizada para establecer cuándo dos puntos se consideran densamente alcanzables.

Definición 17 (Densamente alcanzable). *Sea D un dominio de aplicación, x e y dos elementos de él, $\epsilon \in \mathbb{R}^+$ y $m_p \in \mathbb{Z}^+$ dos constantes. Entonces x es densamente alcanzable desde y si existe una cadena de elementos $y = y_1, y_2, \ldots, y_j = x$, tal que y_{i+1} es densamente alcanzable de forma directa desde y_i, $\forall 1 \leq i \leq j$.*

Utilizando los conceptos anteriores se define la conectividad basada en densidad, la cual es la base para la formación de los grupos en DBSCAN.

Definición 18 (Conectividad basada en densidad). *Sea D un dominio de aplicación, x e y dos elementos de él, $\epsilon \in \mathbb{R}^+$ y $m_p \in \mathbb{Z}^+$ dos constantes. Entonces x está conectado densamente a y con respecto a ϵ y m_p si existe un elemento $z \in D$ tal que x e y sean densamente alcanzables desde z.*

Una vez definido todo lo anterior, se puede explicar lo que DBSCAN considera como un grupo, que aunque tiene algunas diferencias con lo que otras técnicas consideran un grupo, es una de las mayores fortalezas del algoritmo.

Definición 19 (Grupo según DBSCAN). *Sea D un dominio de aplicación, $\epsilon \in \mathbb{R}^+$ y $m_p \in \mathbb{Z}^+$ dos constantes. Entonces se le llama grupo a un subconjunto no vacío c de D que satisface:*

i) $\forall x, y \in D$ *si* $x \in c$ *y el elemento* y *es densamente alcanzable desde* x *con respecto a* ϵ *y* m_p*, entonces* $y \in c$*,*
ii) $\forall x, y \in c$*,* x *está densamente conectado a* y *con respecto a* ϵ *y* m_p*.*

A las condiciones de la definición anterior se les llama condición de maximalidad y de conectividad, respectivamente. La condición de maximalidad, de cierto modo se asemeja a la tercera condición de la definición general (ver definición 13), ya que garantiza que no existan elementos fuera del conjunto que cumplan las condiciones necesarias para pertenecer a este. La condición de conectividad, por su parte, cumple la función de la segunda condición en la definición 13. La mayor diferencia entre esta definición y la general es que, en lugar de utilizar directamente la similitud entre los elementos como criterio para agruparlos, utiliza el concepto de que estos sean densamente alcanzables entre ellos, relajando las condiciones de la definición general y permitiéndole mayor flexibilidad al algoritmo. Un concepto importante utilizado por DBSCAN es el concepto de ruido, el cual se define formalmente a continuación:

Definición 20 (Ruido según DBSCAN). *Sea* D *un dominio de aplicación,* $\epsilon \in \mathbb{R}^+$ *y* $m_p \in \mathbb{Z}^+$ *dos constantes y* $C = \{c_1, c_2, \ldots, c_\kappa\}$ *el conjunto de los grupos de elementos de* D *que cumplen la definición 19 para los parámetros* ϵ *y* m_p*. Entonces se le llama ruido al conjunto de elementos* $\{x \in D | x \notin \cup_{i=1}^{\kappa} c_i\}$*.*

El ruido definido anteriormente está formado por los elementos considerados anómalos desde el punto de vista de la detección de anomalías, lo que facilita la utilización del algoritmo con este propósito. El enfoque basado en densidad de DBSCAN le permite detectar grupos que posean formas arbitrarias y mejorar de este modo la eficacia de la detección al disminuir los falsos positivos.

La principal desventaja de este algoritmo es la dependencia de los parámetros ϵ y m_p los cuales no son fáciles de determinar a priori. Si los valores de los parámetros antes mencionados son altos, la eficiencia del método se puede ver seriamente afectada mientras que, si son muy bajos, se afecta su eficacia. Una de las mayores desventajas de que DBSCAN reciba estos valores como parámetros es que, al estar determinados de forma global y no en dependencia de la densidad de las zonas analizadas del conjunto de datos, el algoritmo no es capaz de detectar anomalías puntuales locales.

Los algoritmos de agrupamiento, en general suelen estar basados en distancia o en densidad, por lo que heredan algunos de los problemas de las técnicas para la detección de anomalías basadas en estas. Las técnicas basadas en agrupamiento, al detectar anomalías como un subproducto del mismo, hacen que sea necesario interpretar los resultados en cada caso para determinar qué elementos se consideran anómalos y además, pueden ser menos eficientes que otras técnicas, debido a tener que agrupar todo el conjunto de datos para realizar la detección de anomalías.

3.4. Conclusiones parciales

Las técnicas de detección de anomalías puntuales son una parte fundamental de la disciplina de la detección de anomalías. Muchas técnicas diseñadas para detectar otros tipos de anomalías se basan en modelar los problemas, de tal forma que se pueda aplicar en ellos la detección de anomalías puntuales. En esta sección se analizaron varias de estas técnicas y se mencionaron sus ventajas y desventajas.

Las redes sociales son un dominio en el que es común encontrar conjuntos de datos de grandes dimensiones, por ello es importante que, además de eficaces, los algoritmos de detección de anomalías sean eficientes. Las técnicas de detección de anomalías puntuales analizadas en este reporte tienen, de modo general, una complejidad computacional cuadrática o superior. En este sentido, los algoritmos basados en distancia y algunos basados en agrupamiento, se encuentran entre los más eficientes, mientras que los basados en densidad resultan más costosos. No obstante las técnicas que utilizan densidad, suelen detectar mayor cantidad de elementos anómalos, pudiendo, incluso, detectar anomalías puntuales locales.

Existen gran cantidad de técnicas para detectar anomalías puntuales, sin embargo, todas tienen en común que tratan los conjuntos de datos como "nubes de puntos" sin tener en cuenta las relaciones entre los elementos. Esta característica es una limitante para su aplicación en las redes sociales, pues en ellas mucha información sobre los datos proviene de la relación entre los elementos.

4. Detección de anomalías agrupadas

La detección de grupos anómalos puede ser vista como un problema de agrupamiento, porque se desea encontrar conjuntos de elementos similares entre sí que representen solo una porción muy pequeña de los datos. Esta tarea no puede ser abordada de forma eficiente utilizando los algoritmos de agrupamiento clásicos. Agrupar todos los elementos de una red social para luego detectar los grupos que resultan anómalos es poco eficiente como técnica de detección de anomalías e impide su aplicación en redes sociales de grandes dimensiones.

Las particularidades antes mencionadas de la detección de anomalías agrupadas, hacen necesario el desarrollo de técnicas específicas para enfrentar este problema. A continuación se mostrarán algunas de ellas.

4.1. Técnicas basadas en agrupamiento

El uso de algoritmos de agrupamiento para detectar anomalías en redes sociales se basa en la característica de estas expuesta en [13]. Donde se explica que en las redes sociales, a medida que las comunidades van creciendo, la mayoría de sus miembros tienden a parecerse menos entre ellos y más a la mayoría de los miembros de la red. Las comunidades pequeñas formadas por elementos muy similares entre sí, tienden a encontrarse conectadas al cuerpo de la red por pocas aristas y suelen tener no mucho más de 150 miembros. Los elementos que forman parte de las anomalías agrupadas son distintos de la mayoría de los elementos de la red al ser anómalos y al mismo tiempo, se encuentran agrupados con otros elementos similares a ellos. Por tanto se pueden emplear algunos algoritmos de agrupamiento para detectar estas comunidades de elementos anómalas con relación al resto de los elementos de la red. Se puede definir el funcionamiento de los algoritmos basados en agrupamiento como se hace en la definición 21.

Definición 21 (Detección de grupos anómalos utilizando agrupamiento). *Sea D un dominio aplicación, s_c una función de similitud entre dos conjuntos, s_I una función de similitud interna de un conjunto y τ un número entero.*

Se desea obtener un conjunto $C = \{c_1, c_2, \ldots, c_\kappa\}$ de conjuntos $c_i \subset D$ tal que:

- $\cup_{i=1}^{\kappa} c_i \subset D$,

- $|\cup_{i=1}^{\kappa} c_i| \leq \tau$,
- *Se maximiza la diferencia* $s_I(c_i) - s_c(c_i, D \setminus c_i)$.

La primera condición de la definición anterior garantiza que los grupos c_i cumplen con la primera condición de las anomalías agrupadas (ver definición 4). La segunda condición, al hacer que la cantidad de elementos anómalos sea menor que un umbral τ que determina cuándo una cantidad de elementos se puede considerar escasa en el conjunto D, garantiza que los conjuntos c_i cumplan la segunda condición de las anomalías agrupadas. La tercera condición de la definición 21, se centra en maximizar la similitud interna de los conjuntos c_i y en minimizar la similitud entre los elementos de cada conjunto c_i y los elementos fuera de él. Esta condición trata de garantizar que los conjuntos c_i cumplan las condiciones 3 y 4 de las anomalías agrupadas globales, para un umbral de similitud adecuado.

En [15] se utiliza el algoritmo de agrupamiento BBC (por sus siglas en inglés, *Bregman Bubble Clustering*) [24] para la detección de anomalías agrupadas. Dicho algoritmo permite encontrar κ grupos densos, ignorando el resto de los datos. Se puede definir el problema que resuelve BBC del siguiente modo:

Definición 22 (Problema resuelto por el algoritmo BBC). *Sea D un dominio de aplicación, $d_\phi : D \times D \longrightarrow [0, \infty)$ una divergencia de Bregman dada, τ un escalar que representa la cantidad de elementos a agrupar y κ la cantidad de grupos a encontrar, entonces:*

Se desea obtener un conjunto $C = \{c_1, c_2, \ldots, c_\kappa\}$ donde $c_i \subset D$ con un conjunto de κ representantes de los grupos $R = \{r_1, r_2, \ldots, r_\kappa\}$ donde r_i es el centroide del grupo c_i tal que:

- $\cup_{i=1}^{\kappa} c_i \subset D, 1 \leq \kappa < n$,
- *se minimiza el costo* $Q_b(C, R) = \frac{1}{|\cup_{i=1}^{\kappa} c_i|} \sum_{j=1}^{\kappa} \sum_{i=1}^{|c_j|} d_\phi(x_i, r_i)$, *donde* $x_i \in c_j$,
- $|\cup_{i=1}^{\kappa} c_i| = \tau$ *con* $\kappa \leq \tau < n$.

El algoritmo BBC en lugar de distancia euclidiana utiliza divergencias de Bregman que es una clase más general de distorsiones (ver definición 71). Además utiliza una función de costo Q_b para medir la calidad de un agrupamiento. Recibe como parámetros la cantidad de grupos a encontrar κ y los centroides o representantes iniciales de cada grupo. En [24] se exponen dos técnicas capaces de generar automáticamente la constante κ y encontrar los centroides iniciales.

La primera técnica es llamada presurización y es una heurística no determinista. Simula el comportamiento de burbujas de aire al ser sometidas a la presión del agua. Primero se escogen los centroides iniciales aleatoriamente y se ejecuta el algoritmo BBC utilizando un valor de τ mayor que el deseado. En sucesivas iteraciones se utilizan los centroides obtenidos en la iteración anterior, como centroides iniciales y un valor menor para τ, se termina de iterar cuando el valor de τ utilizado sea el deseado. Esta técnica puede resultar poco eficiente, debido al costo de realizar numerosas iteraciones.

La segunda técnica propuesta es el algoritmo DGRADE (por sus siglas en inglés *Density Gradient Enumeration*) pensado para ser utilizado en contextos donde la eficiencia sea indispensable. El algoritmo recibe como parámetros la cantidad de elementos a agrupar τ, una matriz M_d con las distancias entre todos los elementos de D y una constante τ' que representa la cantidad de elementos a agrupar en cada grupo. Aunque DGRADE es eficiente y provee una buena aproximación de los valores iniciales deseados, determinar el parámetro τ' a priori puede ser tan difícil como determinar κ.

La propuesta de [15] elimina la necesidad de conocer la cantidad de grupos a formar y solo requiere la cantidad de elementos a agrupar τ. El algoritmo expuesto, llamado AGM (por sus siglas en inglés *Abnormal Group Mining*), resuelve el siguiente problema:

Definición 23 (Problema resuelto por el algoritmo AGM). *Sea D un dominio de aplicación, s una función de similitud entre dos elementos y τ un número entero. Se desea obtener un conjunto $C = \{c_1, c_2, \ldots, c_\kappa\}$ de conjuntos $c_i \subset D$ tal que:*

- $\cup_{i=1}^{\kappa} c_i \subset D$,
- $\forall i, 1 \leq i \leq \kappa$ *se cumple:*
  - *i)* $\forall x, y \in c_i, s(x, y) \geq \delta$,
  - *ii)* $\forall y \in D$, $y \notin \cup_{i=1}^{\kappa} c_i$, $\forall x \in c_i$ *se tiene* $s(x, y) < \delta$,
  - *iii)* $\nexists y \in D \setminus c_i$ *tal que* $\forall x \in c_i$ *se tenga* $s(x, y) \geq \delta$,
- $|\cup_{i=1}^{\kappa} c_i| = \tau$.

El procedimiento utilizado por los autores para resolver este problema consiste en tomar los $\frac{\tau(\tau-1)}{2}$ pares de elementos más similares de D, luego se toman de entre esos elementos los τ que poseen mayor similitud con otro elemento y luego se agrupan en grupos anómalos. δ corresponde al mayor valor de similitud entre el τ-ésimo elemento seleccionado y el elemento más parecido a él.

Las técnicas de detección basadas en agrupamiento suelen estar diseñadas para agrupar conjuntos de elementos y no para detectar anomalías, realizando esta última función como un subproducto. En la mayoría de los casos su rendimiento es bajo debido a las limitaciones de su algoritmo subyacente. Además, es común que este tipo de técnicas requieran para su funcionamiento que se les proporcionen parámetros muy difíciles de conocer a priori, como la cantidad de elementos por grupo o la cantidad de grupos a formar. Las técnicas basadas en agrupamiento retornan al usuario los grupos anómalos, lo cual es deseable pues se están detectando grupos y no elementos aislados, sin embargo, no dan ninguna noción acerca de qué tan anómalos son dichos grupos.

4.2. Técnicas basadas en aislamiento

Las técnicas basadas en aislamiento utilizan como base la definición de anomalía de Liu et al. [9]. Los algoritmos de este tipo intentan aislar las anomalías realizando divisiones en el conjunto de datos. Parten de la hipótesis de que los elementos anómalos deben aislarse con mayor facilidad que los elementos normales. Estas técnicas utilizan unas estructuras llamadas árboles de aislamiento para representar las distintas divisiones realizadas en el conjunto de datos.

Definición 24 (Árbol de aislamiento asociado a un conjunto). *Se le llama árbol de aislamiento asociado a un conjunto S, a un árbol binario, donde cada vértice v tiene asociado un subconjunto de S y una función $b : 2^S \longrightarrow 2^S \times 2^S$ que determina la relación padre-hijo de los conjuntos asociados a un vértice y a sus hijos al dividir el conjunto asociado al vértice v en dos subconjuntos disjuntos S_l y S_r, donde S_l es el conjunto asociado con el hijo izquierdo de v y S_r el conjunto asociado a su hijo derecho. El subconjunto asociado a la raíz del árbol es el propio conjunto S y los conjuntos asociados a las hojas del árbol son conjuntos unitarios.*

Los árboles de aislamiento son utilizados en una etapa inicial de las técnicas de detección de anomalías. Primero, se seleccionan t muestras de tamaño ψ formadas con elementos tomados aleatoriamente y sin remplazo de D y se construyen los árboles de aislamiento asociados a cada

una de ellas. Luego el bosque resultante es utilizado para determinar qué elementos de D son anómalos (ver definición 25).

Definición 25 (Detección de grupos anómalos mediante aislamiento). *Sea D un dominio de aplicación, F un conjunto de t árboles de aislamiento donde cada árbol esta asociado a una muestra de n' elementos de D, $s_h : D \times \mathbb{N} \longrightarrow [0, 1]$ una función que, dado un elemento $x \in D$ y la cantidad de elementos de la muestra asociada a cada árbol de F, determina la similitud entre el promedio de la profundidad de x en F y la profundidad esperada a la que se debería encontrar un elemento normal en estos árboles, indicando mayor similitud cuando el valor de la función es más cercano a 1. Entonces para todo elemento $x \in D$, x se considera anómalo si $s_h(x, n') < \delta$.*

El algoritmo iForest [25] se divide en dos fases, una de entrenamiento y una de evaluación. En la primera fase recibe dos parámetros que son: la cantidad de árboles de aislamiento a construir t y el tamaño de cada muestra n'. En la fase de evaluación, iForest recibe como parámetro el límite de altura de los árboles durante la evaluación denotado h_{lim}.

En la fase de entrenamiento, se divide el conjunto de datos en t muestras de tamaño n', donde cada una se denota D' y cuyos elementos son seleccionados de D aleatoriamente y sin reemplazo (ver algoritmo 1, líneas 1-3). Luego, se construye para cada muestra un árbol de aislamiento y con los arboles de cada muestra se obtiene finalmente un bosque (ver algoritmo 1, líneas 4-5).

Algoritmo 1: iForest

Entrada: D - conjunto de datos a analizar, t - número de árboles de aislamiento a construir, n' - tamaño de las muestras
Salida: un conjunto de t árboles de aislamiento

```
1 Inicializar Bosque
2 for i = 1 to t do
3 |   D' ← Muestra(D, n')
4 |   Bosque ← Bosque ∪ iTree(D')
5 end
6 return Bosque
```

Cada árbol de aislamiento es construido realizando subdivisiones de una muestra D', hasta que todos los elementos hayan sido aislados (ver algoritmo 2). El caso base es cuando la muestra no se puede dividir (líneas 1-2). Si D' es divisible, se selecciona aleatoriamente un atributo a de los d que posee cada elemento $x \in D$ y luego se selecciona un valor aleatorio entre los valores máximo y mínimo que toma a en el conjunto D', este valor se denotará por a^* (líneas 4-6). Utilizando el valor divisor a^*, se realiza una partición de D' en dos subconjuntos $D_l = \{x \in D' | x_a < a^*\}$ y $D_r = \{x \in D' | x_a \geq a^*\}$ (ver líneas 7 y 8). Finalmente, se almacena en un vértice interior el atributo divisor a y el valor divisorio a^* y se retorna el árbol de aislamiento que tiene a este nodo como raíz y cuyos hijos son el resultado de llamar recursivamente al algoritmo 1 en D_l y D_r respectivamente (ver línea 9).

Después de construir el bosque, este es retornado y comienza la fase de evaluación. Entonces para cada elemento $x \in D$, se calcula la longitud promedio en el bosque, del camino desde la raíz de un árbol de aislamiento hasta la hoja (vértice externo) donde x debería haber sido aislado. Utilizando la longitud promedio del camino y la longitud esperada, se determina el grado de

atipicidad de cada elemento.

Algoritmo 2: iTree(D')

Entrada: D' - muestra de datos a analizar
Salida: un árbol de aislamiento
1 **if** *D' no puede dividirse* **then**
2 | **return** $verticeExterior\{cantidadElementos \leftarrow |D'|\}$;
3 **else**
4 | Sea A una lista de atributos en D';
5 | Seleccionar aleatoriamente un atributo $a \in A$;
6 | Seleccionar aleatoriamente un punto de división a^* entre los valores máximo y mínimo del atributo a en D';
7 | $D_l \leftarrow \{x \in D' | x_a < a^*\}$;
8 | $D_r \leftarrow \{x \in D' | x_a \geq a^*\}$;
9 | **return** $verticeInterior\{hijoIzquierdo \leftarrow iTree(D_l), hijoDerecho \leftarrow iTree(D_r), atributoDivisor \leftarrow a, valorDivisor \leftarrow a^*\}$
10 **end**

Para calcular el largo del camino hasta un elemento x se cuentan la cantidad de aristas que es necesario atravesar para llegar de la raíz del árbol a la hoja donde debería haber sido aislado x (ver algoritmo 3, líneas 4-9). Cuando el proceso de medición del camino alcanza una longitud h_{lim} ó llega a un vértice externo, se detiene y se estima la longitud del camino, utilizando la longitud hasta el momento l, más un ajuste $L(cantidadElementos)$ (ver algoritmo 3, líneas 1-3). El ajuste trata de estimar la distancia promedio del camino hasta un elemento en un subárbol aleatorio de tamaño *cantidadElementos*.

Algoritmo 3: PathLength(x, T_i, h_{lim}, l)

Entrada: x - una instancia, T_i - un árbol de aislamiento, h_{lim} - el límite de altura, l - el largo del camino hasta el momento. Se inicializará en 0 la primera vez que se llame
Salida: El largo del camino hasta x
1 **if** *T_i es un vértice externo o* $l \geq h_{lim}$ **then**
2 | **return** $l + L(T_i.cantidadElementos)$
3 **end**
4 $a \leftarrow T_i.atributoDivisor$
5 **if** $x_a < T_i.valorDivisor$ **then**
6 | **return** $PathLength(x, T_i.hijoIzquierdo, h_{lim}, l+1)$
7 **else**
8 | **return** $PathLength(x, T_i.hijoDerecho, h_{lim}, l+1)$
9 **end**

El camino promedio hasta un elemento en un árbol de aislamiento es equivalente al camino promedio de una búsqueda no exitosa en un árbol binario de búsqueda. Dado un conjunto de n' elementos, se puede estimar el largo promedio de las búsquedas no exitosas en un árbol binario de búsqueda que contiene dichos elementos, como se muestra en la ecuación 15.

$$L(n') = \begin{cases} 2H(n'-1) - 2(n'-1)/n & n' > 2, \\ 1 & n' = 2, \\ 0 & n' < 2 \end{cases} . \quad (1)$$

.

En la ecuación 15, $H(i)$ es el número armónico, el cual puede ser estimado como $ln(i) + e$ (Constante de Euler). Debido a que $L(n')$ es el promedio de $h(x)$ dado n', se utiliza para normalizar $h(x)$. Con la información anterior se define el grado de atipicidad de un elemento, como se muestra en la ecuación 2 donde $E(h(x))$ representa el promedio de $h(x)$ en una colección de árboles de aislamiento.

$$s_h(x, n') = 2^{-\frac{E(h(x))}{L(n')}}. \quad (2)$$

Se puede producir un contorno del grado de atipicidad, pasando una muestra reducida a través de iForest, facilitando así un análisis detallado de los resultados de la detección. Al utilizar muestras reducidas de la colección en la construcción de los árboles de aislamiento, en lugar de la colección completa de elementos, se facilita el proceso de aislamiento de las anomalías agrupadas, aminorando el efecto de enmascarado. Al utilizar distintos valores de h_{lim} se puede variar la sensibilidad del algoritmo a los grupos anómalos. Para valores mayores de h_{lim} se suelen detectar como anómalos los elementos esparcidos alrededor de los grupos densos, mientras que a menor valor de h_{lim} se suelen detectar como anómalos los grupos densos de elementos.

Una de las mayores ventajas de iForest es que su tiempo de ejecución es lineal y que tiene un requerimiento espacial bajo. Además puede ser entrenado con un conjunto de entrenamiento que no contenga anomalías y puede detectar anomalías a diferentes grados de granularidad sin que se haga necesario volver a entrenar. El algoritmo no está pensado para trabajar con datos categóricos y no tendría un correcto funcionamiento si se utilizara con tipos de datos mixtos. Los atributos categóricos tendrían menos impacto en el algoritmo que los de valores continuos [25], esto se debe a que los primeros tienen una cantidad de valores posibles, significativamente menor, que los segundos.

En conjuntos de datos donde las anomalías pueden ser detectadas únicamente mediante un análisis de varios atributos al mismo tiempo, dividir el conjunto, utilizando un solo atributo a la vez, no es muy conveniente. El algoritmo SCiForest [9], cuyo nombre proviene de iForest con un criterio de selección de corte (en inglés, *Split Selection Criterion*), ofrece una solución a este problema. SCiForest utiliza hiperplanos seleccionados aleatoriamente para dividir el conjunto de datos, en lugar de utilizar un atributo seleccionado aleatoriamente.

En la fase de entrenamiento, la construcción del bosque se mantiene sin cambios en relación con iForest. La construcción de cada árbol tiene ligeras diferencias, las cuales se pueden apreciar en el algoritmo 4. El algoritmo recibe como parámetros adicionales la cantidad de atributos n_a

utilizados en un hiperplano y el número de hiperplanos a considerar n_ρ.

Algoritmo 4: $iTree(D', n_a, n_\rho)$

Entrada: D' - muestra de datos a analizar, n_a - cantidad de atributos utilizados en un hiperplano, n_ρ - número de hiperplanos considerados en un nodo
Salida: un árbol de aislamiento

```
1 if |D'| ≤ 2 then
2 |  return verticeExterior{cantidadElementos ← |D'|}
3 else
4 |  ρ ← un hiperplano con el mejor punto divisorio p que proporciona el mayor valor
  |  Sd_gain entre n_ρ hiperplanos de n_a atributos seleccionados aleatoriamente
5 |  D_l ← {x ∈ D'|ρ(x) < 0}
6 |  D_r ← {x ∈ D'|ρ(x) ≥ 0}
7 |  v ← max_{x∈D'}(ρ(x)) − min_{x∈D'}(ρ(x))
8 |  return verticeInterior{hijoIzquierdo ← iTree(D_l, n_a, n_ρ), hijoDerecho ←
  |  iTree(D_r, n_a, n_ρ), planoDivisor ← ρ, limiteSuperior ← +v, limiteInferior ← −v}
9 end
```

En cada división, durante la creación de un árbol se construye un hiperplano ρ utilizando el mejor punto de división p y el mejor hiperplano según el criterio Sd_{gain} de entre los n_ρ generados aleatoriamente (ver algoritmo 4, línea 4). El hiperplano es utilizado para dividir el conjunto en dos subconjuntos, sustituyendo al atributo divisor que utiliza iForest (ver algoritmo 4, líneas 5 y 6). Cada nodo interior del árbol de aislamiento, almacena un par de valores $limiteSuperior$ y $limiteInferior$ que se usan en la fase de evaluación (ver algoritmo 4, líneas 7 y 8). En la ecuación 3 se puede ver cómo está formulado ρ.

$$\rho(x) = \sum_{j \in A'} \gamma_j \frac{a_j(x)}{\sigma(A_j^*)} - p. \tag{3}$$

A' tiene n_a índices de atributos, seleccionados aleatoriamente y sin reemplazo del conjunto $\{1, 2, \ldots, |A|\}$. α_j es un coeficiente entre [-1, 1] seleccionado aleatoriamente; A_j^* es el conjunto de valores del j-ésimo atributo de A en el conjunto D'; $\sigma(.)$ es la desviación estándar y $a_j(x)$ es el j-ésimo atributo del elemento x. Luego de construir ρ, se divide el conjunto D' en los conjuntos D_l y D_r tales que $D_l \cup D_r = D'$ y se continua el proceso de construcción del árbol en cada uno de estos conjuntos.

El criterio para seleccionar el corte, es uno de los cambios más importantes que realiza este algoritmo, en relación a su predecesor con el objetivo de mejorar la eficacia en la detección de anomalías agrupadas. Basándose en la definición de anomalía dada por Hawkings, se asume que, bajo ciertas proyecciones, los grupos anómalos deben tener su propia distribución. Por esto se introduce el criterio de selección de corte Sd_{gain} que intenta separar las anomalías agrupadas de los elementos normales, basándose en que poseen distintas distribuciones.

Cuando un corte separa dos distribuciones distintas, sus dispersiones son minimizadas [9]. Utilizando este mecanismo se define el criterio Sd_{gain} como se muestra en la ecuación 4.

$$Sd_{gain}(D'_\rho) = \frac{\sigma(D'_\rho) - avg(\sigma(D'^l_\rho), \sigma(D'^r_\rho))}{\sigma(D'_\rho)}. \tag{4}$$

En la ecuación 4, $D'^{l}_{\rho} \cup D'^{r}_{\rho} = D'_{\rho}$ donde D'_{ρ} es un conjunto de valores reales obtenidos proyectando los elementos de D' en un hiperplano ρ. $\sigma(.)$ es la función de desviación estándar y $avg(a, b)$ retorna $\frac{a+b}{2}$. El objetivo de utilizar este criterio, es encontrar el mejor punto divisorio p que separe a D'_{ρ} en los conjuntos $D'^{l}_{\rho} = \{x \in D'_{\rho} | x < p\}$ y $D'^{r}_{\rho} = \{x \in D'_{\rho} | x \geq p\}$. El criterio es normalizado utilizando $\sigma(D'_{\rho})$, la cual es calculada en una primera pasada por los elementos de D'_{ρ}. En la segunda pasada se busca el mejor punto divisor p con el mayor valor de Sd_{gain} de entre todas las combinaciones posibles de D'^{l}_{ρ} y D'^{r}_{ρ} utilizando la ecuación 4. La desviación estándar mide la dispersión de la distribución de los datos, por lo tanto, cuando un grupo anómalo está presente en D'_{ρ}, es separado, debido a que esto reduce significativamente la dispersión de D'^{l}_{ρ} y D'^{r}_{ρ}.

Aquí es donde SCiForest realiza la mayor parte de los cambios en relación con su predecesor. Se reemplaza la selección de un atributo divisor, por un hiperplano divisor, el cual relaciona varios atributos a la vez. En lugar de seleccionar un punto divisor de entre los valores del atributo divisor en el conjunto de datos, se selecciona el hiperplano con el mejor punto divisor p, utilizando el criterio Sd_{gain}, lo que permite separar elementos con distribuciones distintas. Estos cambios posibilitan un mejor desempeño del algoritmo en la detección de anomalías agrupadas donde los elementos se consideren anómalos como resultado de los valores de varios de sus atributos al mismo tiempo.

En la fase de evaluación el algoritmo se comporta de modo muy similar a iForest, solo que en el cálculo de la longitud del camino hasta un elemento, no utiliza un límite de altura h_{lim} sino que realiza la búsqueda hasta las hojas del árbol (véase algoritmo 5, líneas 1-3). Utiliza los valores $limiteSuperior$ y $limiteInferior$ para definir un rango aceptable para los elementos. Los miembros del conjunto de datos que no han sido vistos anteriormente, se encontrarían fuera del rango aceptable, y al ser más probable que estos sean anómalos, entonces se les penaliza con un no incremento del largo de su camino (véase algoritmo 5, líneas 4-17).

Algoritmo 5: PathLength(x, T_i, l)

Entrada: x - un elemento, T_i - un árbol de aislamiento, l - el largo del camino hasta el momento (se inicializará en 0 la primera vez que se llame)

Salida: El largo del camino hasta x

```
1  if T_i es un nodo externo then
2  |  return l + L(T_i.cantidadElementos)
3  end
4  y ← T_i.planoDivisor(x)
5  if y < 0 then
6  |  if T_i.limiteInferior ≤ y then
7  |  |  return PathLength(x, T_i.hijoIzquierdo, l + 1)
8  |  else
9  |  |  return PathLength(x, T_i.hijoIzquierdo, l)
10 |  end
11 else
12 |  if T_i.limiteSuperior > y then
13 |  |  return PathLength(x, T_i.hijoDerecho, l + 1)
14 |  else
15 |  |  return PathLength(x, T_i.hijoDerecho, l)
16 |  end
17 end
```

El algoritmo mantiene la complejidad temporal de iForest siendo lineal con respecto al conjunto de datos y posee un mejor desempeño en situaciones donde la detección de anomalías depende de múltiples atributos. Además SCiForest es más eficaz en la detección de anomalías agrupadas, debido a que le otorga preferencia a la detección de este tipo de anomalías, mientras que iForest lo hace a la detección de anomalías dispersas.

Las técnicas de detección de anomalías basadas en aislamiento, antes expuestas, poseen una complejidad temporal lineal, sin embargo, en su orden están presentes constantes con un valor relativamente elevado. En la práctica, estos algoritmos pueden presentar peor rendimiento que algunas técnicas con complejidad temporal superlineal en conjuntos de datos con miles de millones de elementos.

Los algoritmos iForest y SCiForest no detectan grupos anómalos, sino que detectan a los elementos que los conforman. La ventaja de estas técnicas es que, al determinar el grado de atipicidad de cada elemento, permiten a los analistas tener una idea de qué tan probable es que estos sean anómalos. La desventaja es que no brindan información acerca de cómo están agrupadas las anomalías. Otro factor a tener en cuenta, es que estos algoritmos utilizan selección de valores de forma aleatoria en ciertos pasos, lo que hace que sean no deterministas.

4.3. Otras técnicas

Existen diversas técnicas para detectar anomalías agrupadas, algunas de ellas más utilizadas que otras. Los algoritmos propuestos en [26] utilizan caminos aleatorios sobre cadenas de Markov para detectar anomalías en un conjunto de datos. Para ello primero se construye una matriz de similitud entre los elementos del conjunto denotada M_s, donde la posición correspondiente a su i-ésima fila y su j-ésima columna, se denota como $M_s[i,j]$ y contiene la similitud entre el i-ésimo y j-ésimo elemento del conjunto. La similitud entre dos elementos x e y se define como el coseno entre los vectores x e y como se puede ver a continuación:

Sean $x = (x_1, x_2, \ldots, x_d)$ y $y = (y_1, y_2, \ldots, y_d)$ dos elementos en U, entonces la similitud entre x y y se define como:

$$s(x,y) = \begin{cases} 0 & x = y, \\ \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i^2}\sqrt{\sum_{i=1}^{d} y_i^2}} & x \neq y \end{cases} . \tag{5}$$

.

La matriz M_s puede ser vista como la matriz de adyacencia de un grafo ponderado. Si el valor de $M_s[i,j]$ es mayor que 0, significa que existe una arista entre el nodo i y el nodo j cuyo valor es la similitud entre dichos elementos.

Utilizando la matriz de similitud y la correspondiente representación en forma de grafo, se modela el problema como una cadena de Markov. El modelo utilizado es un recorrido aleatorio por el grafo definido por los enlaces entre los vértices. La hipótesis utilizada es que, si un elemento tiene baja conectividad con otros elementos del conjunto, entonces es más probable que este sea una anomalía.

La conectividad de un vértice es determinada mediante los votos ponderados que le otorgan los otros vértices del grafo. Los elementos con mayor conectividad traspasan votos con mayor peso que los de menor conectividad. El peso de los votos de cualquier vértice es escalado por la cantidad de elementos adyacentes a él. La conectividad se define del siguiente modo:

$$con(v,i) = \begin{cases} \beta & i = 0 \\ \sum_{v' \in adj(v)} (con(v', i-1)/|v'|) & i \neq 0 \end{cases}. \tag{6}$$

.

La función anterior es una función recursiva, donde β es una constante arbitraria, i representa la iteración, $adj(v)$ el conjunto de vértices adyacentes a v y $|v|$ denota el grado de v. A cada vértice se le asigna un valor arbitrario y luego, en cada iteración, se lleva a cabo un cálculo para refinar el valor de conectividad de cada vértice. El refinamiento de los valores de conectividad se realiza modelando este escenario como una cadena de Markov. La ecuación 6 puede ser planteada de forma matricial del siguiente modo:

$$M_{con_{(i)}} = M_{trans}^T M_{con_{(i-1)}}. \tag{7}$$

En la ecuación 7, la matriz de transición se denota como M_{trans} y $M_{con_{(i)}}$ es el vector con la distribución estacionaria, donde cada componente representa el valor de conectividad para un elemento en el conjunto de datos tras realizar i iteraciones. La matriz de transición M_{trans} se obtiene normalizando la matriz de similaridad M_s. Esta normalización se realiza para asegurar que los elementos de cada fila de la matriz de transición suman uno, lo cual es una importante propiedad de las cadenas de Markov. Como las probabilidades de transición no cambian con el tiempo, esta matriz solo tiene que ser calculada una vez.

$$M_{trans}[i,j] = \frac{M_s[i,j]}{\sum_{k=1}^{n} M_s[i,k]}. \tag{8}$$

Es necesario que la matriz M_{trans} sea irreducible y no periódica, pero la matriz obtenida a partir del conjunto de datos, podría no serlo. Para garantizar la convergencia a una distribución estacionaria única, los autores utilizan una ecuación matricial modificada 9 [27], donde a ϱ se le conoce como factor de suavizado.

$$M_{con_{(i+1)}} = \varrho J_{n\times 1} + (1-\varrho) M_{trans}^T M_{con_{(i)}}. \tag{9}$$

Donde M_{trans} es la matriz de transición de la ecuación 8, $\varrho \in [0,1]$ es una constante y $J_{n\times 1}$ es el vector unitario de n componentes $[1\ 1 \ldots 1]^T$. Esta ecuación utiliza el vector $M_{con_{(i)}}$ obtenido en el paso i para calcular el del paso $i+1$, donde M_{con} es un vector de n componentes y $\forall j, 1 \leq j \leq n, M_{con}[j] = con(x_j)$ donde $con(x_j)$ es el valor de conectividad para el j-ésimo elemento de D.

Utilizando los conceptos antes definidos los autores crean el algoritmo OutRank [27], que recibe como parámetros una matriz de similitud y una constante ϵ. El algoritmo, en primer lugar, crea la matriz de transiciones como se muestra en la ecuación 8 y luego itera utilizando la ecuación 9, hasta obtener un vector $M_{con_{(i)}}$ tal que $\|M_{con_{(i)}} - M_{con_{(i-1)}}\| < \epsilon$ y retorna dicho vector.

Definición 26 (Detección de grupos anómalos utilizando OutRank). *Sea $D \subseteq U$, M_s una matriz de similitud entre los elementos de D, ϵ una constante real, M_{con} el vector resultante de invocar a la función $OutRank(M_s, \epsilon)$ y δ una constante. Entonces se considera que un elemento $x \in D$ es anómalo, si $M_{con}[x] < \delta$.*

Los algoritmos propuestos permiten la detección de anomalías agrupadas, debido a que comparan los elementos de forma global. Este tipo de comparación puede presentar dificultades en la detección de anomalías agrupadas locales. Ambos algoritmos propuestos OutRank-a y OutRank-b

solo se diferencian en el modo de construir la matriz de similitud. La principal dificultad de estas técnicas es el costo de mantener una matriz de adyacencia para los datos así como la gran cantidad de recorridos que se efectúan sobre ella, haciéndolas impracticables en grandes colecciones de datos debido a su alta complejidad espacial y temporal.

Al igual que las técnicas basadas en aislamiento, OutRank detecta los elementos que pertenecen a los grupos anómalos, pero no brinda información sobre cómo están agrupados estos.

4.4. Conclusiones parciales

La cantidad de técnicas desarrolladas para detectar anomalías agrupadas es relativamente pequeña en relación al número de técnicas diseñadas para detectar otros tipos de anomalías. En esta sección se han analizado la mayor parte de las técnicas dedicadas a este fin.

En cuanto a eficiencia, las técnicas basadas en aislamiento resultan las más eficientes, con una complejidad temporal lineal, aunque en la práctica pueden tener un rendimiento cercano a los algoritmos superlineales. Tanto las técnicas basadas en aislamiento como las basadas en caminos aleatorios sobre redes de Markov, tratan el problema de detectar anomalías agrupadas como un problema de detección de anomalías puntuales, por lo que no brindan información sobre la similitud entre los elementos anómalos. Las técnicas basadas en agrupamiento no solo detectan los elementos anómalos, sino que los agrupan teniendo en cuenta su similitud. En redes sociales, esto permite detectar comunidades de elementos muy similares entre sí, información que no es posible obtener utilizando otros métodos, a menos que se realice un procesamiento posterior sobre los datos retornados por el algoritmo.

Las técnicas de detección de anomalías agrupadas tienen la ventaja de que también detectan anomalías puntuales, pues estas son un caso particular de las agrupadas, en las que su grupo solo contiene un elemento. Estas técnicas no utilizan la información sobre las relaciones entre los elementos, presente en las redes sociales, siendo una de sus mayores desventajas en este dominio de aplicación. No obstante, es importante resaltar que es mucho más interesante detectar grupos de elementos anómalos similares entre sí, que elementos aislados y debido al escaso número de trabajos dedicados a la detección de anomalías agrupadas, es un tema que requiere futuras investigaciones.

5. Detección de anomalías colectivas

La detección de anomalías colectivas puede verse a simple vista como un problema de detección de patrones infrecuentes en conjuntos de datos y relaciones entre estos. La diferencia entre un patrón anómalo y uno normal es, en muchas ocasiones, sutil lo que conlleva que las técnicas de detección de patrones infrecuentes comúnmente utilizadas no sean eficaces en la detección de anomalías colectivas.

Las anomalías colectivas pueden encontrarse en cualquier conjunto de datos donde exista alguna relación entre los elementos. Entre estos conjuntos de datos se encuentran los grafos y las secuencias. Este reporte se centra en el caso de los grafos. Un estudio comparativo de las técnicas para detectar anomalías en datos representados como secuencias se puede encontrar en [28].

Detectar anomalías colectivas en datos representados como grafos es de especial interés por su aplicación en redes sociales. Los datos se representan como un grafo $G = \langle V, E \rangle$ donde V representa un conjunto de elementos y E un conjunto de relaciones entre estos. Existen algunos

trabajos dedicados a resumir las técnicas para la detección de anomalías en datos representados como grafos, uno de ellos es [29]. A continuación se mostrarán las clases principales en que se agrupan las técnicas para detectar anomalías en grafos, sean estos dinámicos o estáticos.

5.1. Grafos estáticos

Se denominan grafos estáticos a los grafos que se utilizan para representar un conjunto de datos en un momento determinado, en contraposición a los grafos dinámicos que representan el comportamiento de un conjunto de datos en el tiempo. A continuación se expondrán algunas de las principales técnicas para detectar anomalías en conjuntos de datos representados mediante grafos estáticos.

Es necesario, dentro de cada uno de los tipos de técnicas, tratar por separado el caso donde el grafo es etiquetado, debido a que este tipo de grafo cuenta con más información que los grafos simples, lo que permite utilizar técnicas más complejas para detectar anomalías en ellos.

5.1.1. Reducción a un problema de detección de anomalías puntuales

La mayoría de las técnicas que utilizan un resumen de las características estructurales del grafo para determinar si un vértice es anómalo, intentan reducir el problema de la detección de anomalías colectivas a un problema de detección de anomalías puntuales. Estas técnicas utilizan las características estructurales del grafo, como las ego-redes, las comunidades y el grado de los vértices, como criterios para determinar qué tan distinto es un elemento de los demás.

Una técnica representativa de las antes mencionadas es la presentada por Akoglu et al. [30,31] consiste en un algoritmo llamado OddBall que detecta anomalías en grafos no etiquetados basándose en características estructurales del grafo. Los autores determinaron estas características mediante el análisis de varias redes sociales reales y la extracción de varias leyes de potencia que se cumplen en estas.

Definición 27 (Ley de potencia). *En estadística, se le llama ley de potencia a una relación funcional entre dos cantidades, donde una cantidad varía como una potencia de la otra.*

La idea subyacente en el algoritmo OddBall es, dada una ley de potencia entre dos características de los elementos del grafo, verificar cuánto se alejan, en el elemento analizado, los valores de dichas características de la relación descrita por la ley de potencia dada.

Un ejemplo de las leyes presentadas es la Ley de Potencia de la Densidad de la Ego-red (en inglés *Egonet Density Power Law*). Si se denota la ego-red de v como $\nu_1(v)$, su cantidad de aristas como $|E(\nu_1(v))|$ y su cantidad de vértices $|V(\nu_1(v))|$, entonces los autores definen esta ley de potencia como:

$$|E(\nu_1(v))| \propto (|V(\nu_1(v))|)^{\vartheta},\ 1 \leq \vartheta \leq 2.$$

El objetivo que se persigue con el uso de las leyes de potencia es tratar de reflejar lo que se considera un comportamiento normal para los elementos de una red social. El funcionamiento de OddBall se basa en penalizar a los elementos que se alejan de este comportamiento.

Definición 28 (Problema resuelto por OddBall). *Sea G un grafo, v un vértice de él, δ un umbral de atipicidad, x_v el valor del atributo x para el vértice v, y_v el valor del atributo y para el*

vértice v. Entonces dada la ecuación de la ley de potencia $y = \beta x^{\vartheta}$ que relaciona los atributos x e y, se dice que el vértice v es anómalo si:

$$\frac{max(y_v, \beta x_v^{\vartheta})}{min(y_v, \beta x_v^{\vartheta})} * \log(|y_v - \beta x_v^{\vartheta}| + 1) > \delta.$$

Este algoritmo tiene problemas para detectar puntos cercanos a la línea de valores descrita por la ecuación de la ley de potencia, aunque estos posean valores muy distantes de los demás elementos del conjunto. También es importante tener en cuenta que las leyes de potencia brindadas por los autores, están basadas en grandes grafos de redes sociales, si se quisiera utilizar el algoritmo en otro dominio de aplicación, seria necesario primero investigar las leyes de potencia que rigen los datos. Los creadores de OddBall proponen que se utilice en conjunto con otras técnicas, como el factor de atipicidad local, para mejorar su precision.

5.1.2. Detección de estructuras infrecuentes en grafos etiquetados

Estas técnicas se basan en detectar subgrafos atípicos en grafos etiquetados, donde la atipicidad suele estar determinada por la infrecuencia de los mismos, aunque es común que se tengan en cuenta otros criterios para refinar la detección. De modo general se podría definir el problema resuelto por estas técnicas como se muestra a continuación.

Definición 29 (Problema resuelto por la detección de estructuras infrecuentes en grafos etiquetados). *Sea $\mathbb{G}_L$ el conjunto de todos los grafos etiquetados, $G \in \mathbb{G}_L$, S un subgrafo de G, $I : \mathbb{G}_L \times \mathbb{G}_L \longrightarrow \mathbb{Z}$ una función que, dados un grafo y un subgrafo de él, retorna la cantidad de subisomorfismos del segundo en el primero, δ_s un umbral para determinar cuando una cantidad de repeticiones de un patrón se considera escasa y δ_v un umbral para la cantidad de vértices de un grafo. Entonces se considera que S es anómalo en G si:*

- $I(G, S) < \delta_s$,
- $|V(S)| < \delta_v$.

No obstante el enfoque general, pueden producirse variaciones en el enfoque utilizado para detectar anomalías en dependencia del dominio de aplicación. Ejemplo de lo anterior es la propuesta de Eberle y Holder [32] que se enfoca en detectar estructuras anómalas muy similares a las normales, pues en dominios de aplicación como el lavado de dinero, los criminales intentan simular un comportamiento normal.

En [33] se presentan dos algoritmos basados en el enfoque general, el primero llamado Detección de Subestructuras Anómalas (en inglés *Anomalous Substructure Detection*) y el segundo llamado Detección de Subgrafos Anómalos (en inglés *Anomalous Subgraph Detection*). En este reporte, para evitar confusiones, se hará referencia a los algoritmos antes mencionadas como DSEA y DSGA respectivamente. Los autores asumen que cada vértice y arista del grafo que representa los datos, posee una etiqueta que identifica su tipo, el cual no tiene que ser único: dos vértices o dos aristas se consideran iguales si son del mismo tipo.

El algoritmo DSEA, asume que las anomalías son subgrafos infrecuentes en el grafo etiquetado G que representa el conjunto de datos, aunque también considera la cantidad de elementos que conforman dichos subgrafos. Es importante tener en cuenta lo anterior, pues si un subgrafo está compuesto por muchos elementos, es más probable que sea infrecuente, aunque no necesariamente anómalo, el ejemplo mas sencillo de esto es el propio G, del cual solo existe una instancia.

Definición 30 (Problema resuelto por el algoritmo DSEA). *Sea G un grafo etiquetado, $|V(G)|$ la cantidad de vértices de G, δ un umbral de atipicidad, $I : \mathbb{G}_L \times \mathbb{G}_L \longrightarrow \mathbb{Z}$ una función que, dados un grafo etiquetado G y un subconjunto S de este, retorna la cantidad de subgrafos de G que son isomorfos con S. Entonces se consideran anómalos los subgrafos S de G, tales que:*

$$\frac{1}{|V(S)| * I(G,S)} > \delta.$$

Las principales ventajas de DSEA son su capacidad de detectar subgrafos atípicos en lugar de solo vértices anómalos y de ofrecer un grado de atipicidad para dichos subgrafos. Sus mayores desventajas son el alto costo computacional de la función I y la sensibilidad del algoritmo al tamaño de los subgrafos, lo que conlleva la necesidad de cambiar el valor de δ ante cambios en las dimensiones del grafo.

El segundo algoritmo propuesto por los autores es DSGA el cual está diseñado para ser aplicado en grafos conformados por subgrafos no conectados entre sí. La idea subyacente es que los subgrafos que poseen pocos patrones en común con los demás, tienen mayor tendencia a ser anómalos que los que comparten mayor cantidad de patrones entre ellos.

Esta técnica forma parte de un sistema llamado Subdue, que realiza una compresión del grafo analizado G, durante el proceso de detección de anomalías. La compresión se realiza seleccionando los subgrafos que se repiten varias veces en G y sustituyendo cada una de sus instancias por un vértice de nuevo tipo que tendrá aristas con todos los vértices que se relacionaban con alguno de los elementos que componían el subgrafo.

El algoritmo que lleva a cabo la compresión mantiene dos listas, una lista de subgrafos descubiertos L_D y una lista con los mejores subgrafos descubiertos hasta el momento L_M. Se inicializa L_D con un vértice de cada clase. Luego se comienza a iterar, repitiendo un proceso donde se remueve cada subgrafo de L_D y se inserta en ella el resultado de adicionarle al subgrafo removido un nuevo vértice y una arista que lo una a él, o una arista entre los vértices que lo conforman. En cada iteración, antes de remover los subgrafos de L_D se seleccionan los mejores y se adicionan a L_M. Al concluir este procedimiento, el mejor subgrafo de L_M es utilizado para comprimir el grafo.

El patrón utilizado para comprimir el grafo se selecciona utilizando una función $g_b : \mathbb{G}_L \times \mathbb{G}_L \longrightarrow \mathbb{Z}$ que determina cuánto se puede comprimir un grafo sustituyendo ese patrón por un vértice. Dicha función utiliza la longitud mínima de descripción (en inglés *minimun description length*) de un grafo, que representa la cantidad de bits necesarios para representar dicho grafo. La función tendría la siguiente forma:

$$g_b(S,G) = mdl(G|S) + mdl(S).$$

Donde G es un grafo, S un subgrafo de él, $mdl(G|S)$ es la longitud mínima de descripción de G después de comprimirlo utilizando a S, y $mdl(S)$ es la longitud mínima de descripción de S.

El algoritmo DSGA determina si un subgrafo S de un grafo G es anómalo, realizando varias iteraciones, en cada una de las cuales se comprime el grafo G. Mientras más porciones de S sean comprimidas, significa que posee más patrones en común con el resto del grafo, por lo que es menos anómalo. La porción del subgrafo que se comprime en cada iteración se calcula como se muestra a continuación:

$$\varpi(S,i) = \frac{mdl_{i-1}(S) - mdl_i(S)}{mdl_0(S)}.$$

Donde $\varpi(S,i)$ es un valor entre 0 y 1 que representa la porción de S que se comprimió en la i-ésima iteración y $mdl_j(S)$ representa la longitud mínima de descripción de S tras j iteraciones.

Utilizando ϖ se puede determinar si un subgrafo es anómalo mediante un procedimiento donde comienza siendo totalmente anómalo y a medida que partes de él son comprimidas, decrece su grado de atipicidad de acuerdo con una relación entre la porción del grafo que se comprimió y la iteración en la que ocurrió. El grado de atipicidad decrece más rápidamente mientras mayor sea la porción del grafo que es comprimida en las primeras iteraciones, ya que los primeros patrones que se comprimen son los más comunes (los menos anómalos). Esta idea se define formalmente a continuación:

Definición 31 (Problema resuelto por el algoritmo DSGA). *Sea G un grafo etiquetado, S un subgrafo de G que se desea determinar si es anómalo, $\bar{k}$ un número entero que representa la cantidad de iteraciones a realizar y δ un umbral de atipicidad. Entonces se considera que S es anómalo en G si:*

$$1 - \frac{1}{\bar{k}} \sum_{i=1}^{\bar{k}} (n - i + 1) * \varpi(S,i) > \delta.$$

La principal desventaja de esta técnica es su alto costo computacional, ya que es necesario realizar compresiones sucesivas del grafo durante el proceso de detección de anomalías.

5.2. Grafos dinámicos

Los grafos estáticos permiten representar una red conformada por elementos y relaciones entre estos en un momento determinado del tiempo; comúnmente a estos grafos se les llama instantánea (del ingles *snapshot*). Los grafos dinámicos, modelan la evolución de una red a través del tiempo, o sea, se podría ver un grafo dinámico como una secuencia de instantáneas, donde cada una, posee una etiqueta temporal (del inglés *timestamp*) que representa el instante de tiempo en el que fue tomada esa instantánea de la red. Veamos a continuación una definición más formal de instantánea y de grafo dinámico:

Definición 32 (Instantánea). *Se le llamará instantánea, a un grafo etiquetado G_{t_i} donde i representa el instante de tiempo en que la instantánea fue tomada y se le llama etiqueta de tiempo.*

Definición 33 (Grafo dinámico). *Se le llama grafo dinámico a una secuencia de instantáneas $G_d = \langle G_{t_1} G_{t_2} \dots G_{t_\kappa} \rangle$.*

En este trabajo se denotará por $\mathbb{G}_D$ al espacio de todos los grafos dinámicos.

La mayoría de los trabajos en el área de la detección de anomalías en grafos están enfocados en la detección de anomalías en grafos estáticos. Lo anterior se debe en gran medida a que los grafos estáticos son útiles para modelar muchos problemas y los algoritmos existentes para detectar anomalías en ellos aún necesitan ser perfeccionados para brindar mayor eficacia y eficiencia. En los últimos años ha habido un aumento notable del interés en los grafos dinámicos debido, entre otros factores, al auge de las redes sociales. Los grafos dinámicos, al permitir modelar la evolución de una red en el tiempo, brindan valiosa información que es utilizada por las técnicas de detección de anomalías para mejorar la eficacia de la detección. Un ejemplo de lo anterior sería un elemento

cuya cantidad de relaciones en un momento determinado fuese significativamente distinta a su cantidad de relaciones históricas, lo cual se consideraría anómalo para dicho elemento.

En la literatura, existen análisis comparativos de las técnicas para la detección de anomalías en grafos dinámicos, entre los que se encuentran [34,29]. A continuación se mostrará una de las clases en las que pueden ser agrupadas dichas técnicas.

5.2.1. Eventos basados en características

La idea tras estas técnicas es que los grafos que representan una misma red en diferentes momentos de tiempo deben poseer características similares. Las características utilizadas para comparar las instantáneas pueden ser varias, como la distribución de los grados de los vértices y el diámetro del grafo, entre otras.

Una parte de las técnicas de detección de eventos anómalos basados en características, se centra en detectar instantáneas anómalas en la evolución de un grafo dinámico, es decir detectan en qué momento de tiempo la red se comportó de manera anómala. El procedimiento que comúnmente realizan estos algoritmos consiste en los siguientes pasos:

- Extraer una suerte de sumario de cada instantánea del grafo dinámico: dicho sumario estaría conformado por características generales del grafo de la instantánea, más que por las características individuales de los vértices o las aristas.
- Elegir una función de similitud entre los sumarios y comparar los sumarios asociados a las instantáneas con etiquetas de tiempo consecutivas.
- Seleccionar un umbral de similitud para determinar cuándo dos sumarios se consideran similares.
- Señalar como anómalas aquellas instantáneas cuya similitud con la instantánea que las precede sea menor que el umbral de similitud.

De un modo más formal se puede definir el problema que resuelven las técnicas de detección de instantáneas anómalas de la siguiente manera:

Definición 34 (Problema resuelto por la detección de instantáneas anómalas). *Sea* $G_d = \langle G_{t_1} G_{t_2} \dots G_{t_\kappa} \rangle$ *un grafo dinámico,* δ *un umbral de similitud y* $s_t : \mathbb{G}_L \times \mathbb{G}_L \longrightarrow [0,1]$ *una función que, dados dos grafos etiquetados, extrae sus sumarios y determina su similitud retornando un valor más cercano a 1, a mayor similitud de estos. Entonces se considera que una instantánea* G_{t_i} *es anómala si* $s_t(G_{t_{i-1}}, G_{t_i}) < \delta$.

Este enfoque es utilizado en varias propuestas, entre las que se encuentran [35,36]. Su principal problema es que solo detecta que el grafo dinámico se comportó de manera anómala en un momento de tiempo dado, sin brindar ningún detalle sobre cuáles nodos y aristas causaron este comportamiento. Este enfoque puede ser utilizado en dominios de aplicación donde lo relevante sea solo el comportamiento de la red en su conjunto a través del tiempo. A diferencia de las técnicas anteriores, existen algunas técnicas de detección de eventos basadas en características que sí son capaces de detectar vértices anómalos en el grafo, entre estas se encuentra la propuesta por Akoglu y Faloutsos [37].

Los autores proponen determinar las características a extraer de los elementos (vértices), entre las cuales se puede encontrar el grado, el peso máximo de las aristas conectadas a él, el promedio de los pesos de dichas aristas, entre otros. Luego para cada una de las n_f características seleccionadas se conforma una matriz M_{vt} de $n_t \times n$ donde n es la cantidad de vértices en el grafo

y n_t la cantidad de instantes de tiempo que han transcurrido. Cada una de las matrices M_{vt_i} correspondiente a la característica a_i de los vértices almacena en la posición $M_{vt_i}[j_1, j_2]$ el valor del atributo a_i en el momento j_1 para el vértice j_2. Mediante n_f matrices M_{vt} se conforma una suerte de sumario de la evolución de la red con la cual trabaja la propuesta de los autores.

El procedimiento empleado para encontrar el momento de tiempo en el que se produjo un cambio significativo en la red, utilizando el sumario antes descrito, consiste en primer lugar, en tomar la matriz M_{vt_i} correspondiente a la característica a_i extraída de los vértices de entre las n_f posibles. Una vez hecho lo anterior, se selecciona una submatriz M_w de M_{vt_i} de dimensiones $n_w \times n$ donde $n_w \leq n_t$, esta submatriz representa los valores del atributo a_i de los vértices en una ventana de tiempo de dimension n_w. Esta ventana de tiempo se va desplazando iterativamente una posición en el tiempo hasta que se alcance el final de la matriz M_{vt_i}. Cada una de las submatrices obtenidas durante el proceso de desplazamiento de la ventana de tiempo, es utilizada para construir una matriz de correlación M_{cor} entre los pares de vectores de la serie temporal de los valores del atributo a_i de los elementos en la ventana de tiempo de dimension n_w, los cuales son los vectores columna de la submatriz M_{w_j}. El valor de $M_{cor_j}[i_2, j_2]$ es el correspondiente a la correlación entre los vectores columna i_2 y j_2 de M_{w_j}, la cual se calcula del modo siguiente:

$$cor(i_2, j_2) = \frac{cov(i_2, j_2)}{\sigma(i_2)\sigma(j_2)}. \tag{10}$$

En la ecuación anterior $cov(.,.)$ representa la covarianza entre dos vectores y $\sigma(.)$ la desviación estándar. Se calcula, luego, el vector propio de las matrices de correlación M_{cor}. El valor de dicho vector, en la posición correspondiente a cada vértice del grafo, describe qué tan relacionado está ese vértice con los demás elementos, encontrándose más relacionado con estos a mayor valor. Los autores llaman a estos vectores propios, "comportamiento propio" de todos los vértices del grafo en conjunto.

Una vez obtenidos los vectores propios de las matrices M_w, es necesario determinar si el vector propio $\overrightarrow{v}_j$, que representa el comportamiento de los vértices en un instante de tiempo j es un punto de cambio en el comportamiento de la red. Para lograr lo anterior, se toma el vector $\overrightarrow{v}_j$ y se le compara con un vector $\overrightarrow{v}_r$, que representa el comportamiento de la red antes del momento j y se construye como el promedio los vectores $\overrightarrow{v}_{j-1}, \overrightarrow{v}_{j-2}, \ldots, \overrightarrow{v}_{j-n_w}$. La comparación entre vectores se realiza utilizando el producto escalar y se determina cuánto ha cambiado el grafo en el momento j en relación a los momentos anteriores, utilizando la siguiente función:

$$\chi_w = 1 - \overrightarrow{v}_j^T \overrightarrow{v}_r. \tag{11}$$

En la ecuación 11, se puede observar que si $\overrightarrow{v}_j$ es perpendicular a $\overrightarrow{v}_r$ entonces χ_w será igual a 1, mientras que si $\overrightarrow{v}_j$ es igual a $\overrightarrow{v}_r$, entonces χ_w será igual a 0. Teniendo en cuenta lo anterior se puede utilizar a χ_w para determinar el grado de atipicidad de la red en un momento de tiempo específico, solo es necesario para ello, determinar un umbral que establezca para qué valores de χ_w se considera anómalo el comportamiento de la red. De un modo más formal, se puede definir el problema resuelto por este algoritmo del modo siguiente.

Definición 35 (Problema resuelto por la detección de eventos anómalos propuesta en [37]). *Sea $G_d = \langle G_{t_1} G_{t_2} \ldots G_{t_\kappa} \rangle$ un grafo dinámico, δ un umbral de similitud y $s_{\chi_w} : \mathbb{G}_D \times \mathbb{Z} \longrightarrow [0, 1]$ una función que dados un grafo dinámico etiquetado y un momento de tiempo i extrae un sumario del grafo en el momento i y un sumario del grafo en el tiempo anterior al momento*

i y determina la similitud entre ambos sumarios, tomando un valor más cercano a 1, a mayor similitud de estos. Entonces se considera que una instantánea G_{t_i} es anómala si $s_{\chi_w}(G_d, i) < \delta$.

La principal ventaja de esta técnica es que el vector de comportamiento propio permite determinar los valores de comportamiento individuales de cada elemento. Por ello, al determinar que la red se comportó de manera atípica en un instante de tiempo, es posible determinar cuáles elementos fueron los de mayor variación en su comportamiento individual y por tanto los que más afectaron el comportamiento general de la red.

Esta técnica tiene como desventaja fundamental la utilización de un gran número de matrices, las cuales pueden ser de dimensiones considerablemente altas cuando se trabaja con grandes conjuntos de datos, lo que hace que este algoritmo sea costoso desde un punto de vista espacial.

5.3. Conclusiones parciales

Las técnicas de detección de anomalías colectivas son capaces de sacar provecho de la información que brindan las redes sociales y utilizarla en la detección de anomalías. En esta sección se analizaron varias de ellas.

Las técnicas que intentan reducir este problema a uno de detección de anomalías puntuales, pueden ser bastante eficientes, ya que extraen las características fundamentales de cada elemento y las utilizan para determinar qué elementos son distintos del resto. Estas técnicas, aunque utilizan información sobre las relaciones entre los elementos, solo manejan información general sobre dichas relaciones, lo que hace que tengan la misma limitación para su aplicación en las redes sociales que las de detección de anomalías puntuales.

El rendimiento de las técnicas basadas en detección de estructuras infrecuentes en grafos etiquetados es menor que el de las mencionadas anteriormente y es una de sus principales limitantes. No obstante, son capaces de detectar grupos de elementos que podrían parecer normales si se analizan sus características individuales, pero que se relacionan de modo anómalo entre ellos. Estas técnicas poseen un gran número de aplicaciones en las redes en general y en las redes sociales en particular. Es importante tener en cuenta que estas técnicas utilizan la estructura de las relaciones entre los elementos, pero no utilizan las características individuales de estos.

La detección de anomalías en grafos dinámicos tiene una aplicación especial en las redes sociales debido a que con ella se puede comparar a un elemento, no solo con los demás elementos de la red, sino con su propio comportamiento en el pasado. El principal reto de las técnicas de detección de anomalías en este tipo de grafos, es lograr procesar gran cantidad de datos con eficiencia.

Existen varios trabajos enfocados en las técnicas de detección de anomalías colectivas, sin embargo la mayoría de las técnicas existentes, no utilizan toda la información disponible en los grafos para realizar la detección o poseen un costo computacional elevado, por lo que futuras investigaciones en este tema son necesarias, para tratar de superar estos problemas.

6. Detección de anomalías contextuales

La detección de anomalías contextuales resulta de gran importancia debido a que muchas anomalías no pueden ser detectadas a menos que sean analizadas dentro de un determinado contexto. La cantidad de trabajos dedicados a la detección de este tipo de anomalías es mucho menor que

la dedicada a las anomalías puntuales y a las colectivas, aunque en la última década ha habido un incremento notable en el interés por la detección de anomalías contextuales.

Existen dos enfoques fundamentales para detectar anomalías contextuales. El primero, está basado en reducir la detección de estas a un problema de detección de anomalías puntuales, mientras que el segundo enfoque, se basa en modelar la estructura de los datos y utilizar este modelo para detectar las anomalías contextuales.

6.1. Reducción a un problema de detección de anomalías puntuales

El problema de detectar anomalías contextuales puede verse como un problema de detección de anomalías puntales en un contexto específico. Siguiendo la lógica anterior se podría detectar anomalías contextuales siguiendo un procedimiento consistente en:

- Determinar para cada elemento el contexto al que pertenece.
- Aplicar alguna técnica conocida de detección de anomalías puntuales en los grupos formados por aquellos elementos que se encuentran en un mismo contexto.

Una técnica perteneciente a esta categoría es la propuesta en [16]. Los autores plantean, que aunque muchas técnicas de detección de anomalías asumen que no se tiene ningún conocimiento acerca de qué atributos pueden conllevar a que un elemento sea anómalo, esto no es cierto en la mayoría de los dominios de aplicación. En muchos problemas es posible dividir los atributos de los elementos en contextuales y de comportamiento, donde sin importar la rareza de los valores de los atributos contextuales de un elemento, estos no conllevan que se le considere anómalo, solo brindan un contexto para el análisis de los atributos de comportamiento. Al diferenciar estos atributos en el análisis y detectar como anómalos solo aquellos elementos que poseen valores atípicos de sus atributos de comportamiento, en un contexto determinado, se reduce la cantidad de falsos positivos arrojados por el algoritmo.

Los autores proponen particionar los atributos de cada elemento x en dos subconjuntos disjuntos, un conjunto $A_c(x)$ correspondiente a los atributos contextuales y otro $A_b(x)$ correspondiente a los atributos de comportamiento, por lo que un elemento x se representaría como un par $x = \langle A_b(x), A_c(x)\rangle$. La distribución de los atributos contextuales del conjunto de datos es modelada mediante un Modelo de Mezcla Gaussiano (del inglés *Gaussian Mixture Model*) denotado por U_c y compuesto por n_{U_c} Gaussianos, donde el i-ésimo Gaussiano se denota U_{c_i}. Los atributos de comportamiento del conjunto de datos también son particionados en n_{U_b} conjuntos, utilizando otro Modelo de Mezcla Gaussiano denotado por U_b. Ademas es aprendida una función de mapeo probabilístico $P(U_{b_j}|U_{c_i})$ que determina la probabilidad de que los atributos de comportamiento $A_b(x)$ de un elemento x hayan sido generados por un Gaussiano U_{b_j}, si sus atributos contextuales $A_c(x)$ son generados por U_{c_i}. Utilizando lo anterior, se define el grado de atipicidad de un elemento como:

$$f_{CAD}(A_b(x)|\Xi, A_c(x)) = \sum_{i=1}^{n_{U_c}} P(A_c(x) \in U_{c_i}) \sum_{j=1}^{n_{U_b}} P(A_b(x) \in U_{b_j}) P(U_{b_j}|U_{c_i}). \tag{12}$$

Donde $P(A_c(x) \in U_{c_i})$ indica la probabilidad del conjunto de atributos contextuales de un punto x de ser generados por la componente de la mezcla U_{c_i} y $P(A_b(x) \in U_{b_j})$ la probabilidad de que los atributos de comportamiento del punto x sean generados por la componente de la

mezcla U_{b_j}. El conjunto de parámetros Ξ controla características específicas de la distribución y es utilizado para ajustar el modelo al conjunto de datos. En [16] se ofrecen varios algoritmos para calcular Ξ. La función f_{CAD} retorna valores entre 0 y 1, siendo más cercanos a 0, a mayor atipicidad del elemento analizado.

Utilizando la función anterior se calcula el grado de atipicidad de cada elemento del conjunto y se identifican como anómalos aquellos cuyo grado de atipicidad sea menor que un determinado umbral. La selección del umbral de atipicidad puede realizarse de forma automática utilizando un porcentaje y un conjunto de entrenamiento. Se puede definir el problema resuelto por el algoritmo CAD (por sus siglas en inglés *Conditional Anomaly Detection*) propuesto en [16], del modo siguiente.

Definición 36 (Problema resuelto por el algoritmo CAD). *Sea D un dominio de aplicación, $x = \langle A_b(x), A_c(x)\rangle$ un elemento de él, con $A_b(x)$ y $A_c(x)$ el conjunto de atributos de comportamiento y contextuales de x respectivamente, f_{CAD} una función que determina el grado de atipicidad de un elemento, definida como en la ecuación 12, Ξ un conjunto de parámetros utilizados para ajustar el modelo al conjunto de datos y δ un umbral de atipicidad. Entonces x se considera anómalo si:*

$$f_{cad}(A_b(x)|\Xi, A_c(x)) < \delta.$$

Este algoritmo tiene como ventaja un menor número de falsos positivos y la utilización de información contextual en la resolución del problema, lo que le permite detectar anomalías que de otro modo pasarían desapercibidas. Su principal desventaja es que asume que los elementos del conjunto de datos pueden ser modelados mediante un Modelo de Mezcla Gaussiano, además de que ajustar el modelo a los datos puede ser costoso computacionalmente. Este algoritmo no tiene en cuenta las relaciones entre los elementos en la detección de anomalías, lo cual es una desventaja para su aplicación en redes sociales. Esta dificultad es común a la mayoría de las técnicas que intentan reducir la detección de anomalías contextuales al problema de detectar anomalías puntuales.

6.2. Utilización de la estructura de los datos

La detección de anomalías contextuales utilizando la estructura de los datos es útil en los dominios de aplicación donde determinar el contexto de cada elemento resulta complicado. Estas técnicas permiten modelar las relaciones entre los elementos con distintos contextos, lo que brinda información importante para la detección de elementos anómalos.

Un ejemplo de la utilidad de modelar la estructura de los datos en la detección de anomalías se puede ver en los problemas donde el conjunto de datos puede ser separado en dos subconjuntos disjuntos, tales que no exista ningún vínculo directo entre los elementos de un mismo conjunto. Los grafos bipartitos resultan un modo natural de representar estos conjuntos de datos y se han desarrollado técnicas de detección de anomalías que utilizan su estructura para brindarle un contexto a los elementos.

En [38] se presenta una técnica para la detección de anomalías en grafos bipartitos, basada en caminos aleatorios sobre cadenas de Markov. La idea de esta técnica es calcular la relevancia entre los elementos y agruparlos por esta, para luego identificar como anómalos a los elementos de un conjunto que estén conectados con elementos de otro conjunto que sean poco relevantes entre ellos.

En primer lugar se construye la matriz de adyacencia M_G del grafo G, que luego se utiliza para construir la matriz de transición de Markov M_{trans}. La matriz de transición se construye del modo siguiente:

$$M_{trans}[i,j] = \frac{M_G[i,j]}{\sum_{k=1}^{|V|} M_G[k,j]}.$$

Para calcular la relevancia de los elementos $y \in V_1$ ($y \in V_2$) para un elemento $x \in V_1$ ($x \in V_2$), se utiliza la matriz de transición M_{trans}, un vector $\overrightarrow{q_G}$ de n componentes que contenga el valor 1 en la posición correspondiente al elemento x y 0 en las demás, además se utiliza una constante $\varrho \in [0,1]$. Entonces el vector $\overrightarrow{q_r}$ con la relevancia de cada elemento se calcula de modo iterativo como se muestra a continuación:

$$\overrightarrow{q_r}_{(i+1)} = \varrho\overrightarrow{q_G} + (1-\varrho)M_{trans}\overrightarrow{q_r}_{(i)}. \tag{13}$$

El vector $\overrightarrow{q_r}_{(i)}$ representa el vector de relevancia en la i-ésima iteración. Utilizando la ecuación anterior se itera hasta que el módulo de la diferencia entre $\overrightarrow{q_r}_{(i)}$ y $\overrightarrow{q_r}_{(i+1)}$ sea menor que un valor prefijado. Cuando se termina de iterar, el vector $\overrightarrow{q_r}$ obtenido, contiene el valor de relevancia para x de cada uno de los elementos de su mismo subconjunto.

Para determinar si un elemento $x \in V_2$ ($x \in V_1$) es anómalo se toma el conjunto de todos los elementos $y \in V_1$ ($y \in V_2$) vinculados a él $S_x = \{y \in V_1 | \{x,y\} \in E\}$. Intuitivamente si x no es anómalo, entonces el grado de relevancia entre cualquier par de elementos de S_x debe ser alto.

El algoritmo de detección de anomalías, determina en primer lugar, la relevancia entre todos los elementos de S_x y conforma con estos valores una matriz de relevancia M_r que en cada posición $M[i,j]$ contiene la relevancia del i-ésimo elemento de S_x para el j-ésimo elemento de S_x. Es importante destacar que M_r no es simétrica, pues la relevancia del elemento i para el elemento j no tiene que ser igual a la relevancia del elemento j para el elemento i. Entonces para detectar anomalías se utiliza una función r_s que calcula el promedio de las relevancias de M_r sin tener en cuenta la diagonal, pues en ella se encuentra la relevancia de cada elemento con si mismo.

Definición 37 (Problema resuelto por el algoritmo AD). *Sea $G = \langle V, E\rangle$ un grafo bipartito, con $V = V_1 \cup V_2$, un elemento $x \in V_1$ que se desea determinar si es anómalo, una matriz de relevancia M_r entre los elementos del conjunto $\{y \in V_2 | \{x,y\} \in E\}$, una función r_s que dada una matriz calcula el promedio de sus valores sin tener en cuenta la diagonal y η un umbral que determina cuando un elemento es considerado normal, dado su grado de normalidad. Entonces x se considera anómalo si $r_s(M_r) \leq \eta$.*

Este algoritmo aprovecha que el grafo sea bipartito para no construir su matriz de adyacencia completa durante el proceso de cálculo de la relevancia entre los elementos, lo que alivia uno de los principales problemas de las técnicas basadas en caminos aleatorios que es su costo espacial. Los autores proponen un método aproximado para el cálculo de la relevancia que consiste en particionar el conjunto de vértices donde se encuentra el nodo x a analizar y luego realizar el cálculo de la relevancia usando solamente el subconjunto donde x se encuentra. Esta mejora intenta filtrar los vértices con relevancia cercana a cero para x, los que comúnmente son una porción significativa de los vértices de G.

No todas las técnicas se aplican en contextos tan específicos como grafos bipartitos, algunas se aplican en contextos más generales, como la propuesta de Gao et al. [17] que brinda una técnica probabilística para la detección de anomalías en comunidades sobre redes de información. Los

autores modelan el problema utilizando variables aleatorias ocultas de Markov, de modo que se tienen en cuenta tanto las características individuales de los elementos como las relaciones entre estos. Una particularidad de las comunidades es que aunque se sabe que son atributos contextuales, no se conoce su valor para cada elemento a priori, sino que tiene que ser determinado.

La técnica antes mencionada toma el grafo que representa la red y a partir de él modela un grafo de dependencia entre los elementos. Esta técnica utiliza $\bar{m}$ componentes para describir las posibles comunidades normales a las que puede pertenecer un elemento y una componente adicional para los elementos atípicos. En cada nodo del grafo de dependencia se mantiene una variable oculta $\bar{z}_i$ que indica la comunidad a la que pertenece.

Si se tiene que $\bar{A} = \{a(x_1), a(x_2), \ldots, a(x_n)\}$ es un conjunto donde $a(x_i)$ representa los datos asociados a x_i y $\bar{X} = \{\bar{x_1}, \bar{x_2}, \ldots, \bar{x_n}\}$ es un conjunto de variables aleatorias donde cada variable aleatoria $\bar{x_i}$ genera los datos $a(x_i)$ asociados al elemento x_i, entonces el modelo propuesto, asume que la naturaleza probabilística de $\bar{x_i}$ esta determinada por la variable oculta de Markov $\bar{z}_i$.

La lógica de esta técnica es que si dos elementos se encuentran unidos por una arista, entonces es más probable que ellos pertenezcan a una misma comunidad. Además se asume que la comunidad de un elemento está fuertemente relacionada con las características de este.

El problema que intentan resolver los autores, en cuanto a detección de anomalías se refiere, es el problema de la detección de anomalías puntuales globales en comunidades, el cual se puede definir del siguiente modo.

Definición 38 (Problema resuelto por la detección de anomalías en comunidades). *Sea $G = \langle V, E\rangle$ un grafo etiquetado, s una función de similitud entre vértices, δ un umbral de similitud y $cm : V \longrightarrow \mathbb{Z}$ una función que dado un vértice de G retorna la comunidad a la que pertenece, entonces se consideran anómalos los elementos $x \in V$ tales que $\forall y \in \{y \in V | cm(y) = cm(x)\}$ se cumple $s(x, y) < \delta$.*

La propuesta de Gao et al. tiene como principal ventaja, que resuelve dos problemas, en primer lugar agrupa los elementos en comunidades, atendiendo tanto a sus características individuales como a las relaciones entre estos, y en segundo lugar detecta los elementos que son anómalos en su comunidad.

Esta técnica posee varias dificultades para su aplicación que deben ser mencionadas. En primer lugar es necesario que se pueda definir una similitud entre las etiquetas de los nodos para así detectar los elementos que se diferencian más de los miembros de su comunidad, pero para algunos tipos de datos podría no tener sentido establecer tal función de similitud. En segundo lugar hay que tener en cuenta que esta técnica asume que la distribución de los elementos normales es Normal (Multinomial si los datos son texto) y que la distribución de las anomalías es Uniforme, pero la distribución de los datos es algo que no se conoce en la mayoría de los dominios de aplicación e intentar ajustar los datos a una distribución puede resultar costoso computacionalmente. Por último, es necesario añadir, que para el funcionamiento de este algoritmo deben proveerse ciertos parámetros que pueden ser difíciles de estimar y que, ante cambios en el conjunto de datos, podría ser necesario volverlos a estimar.

6.3. Conclusiones parciales

Las anomalías contextuales son un tipo interesante de anomalía, pues cualquiera de las anomalías antes mencionadas pueden ser también contextuales, siempre que se incluya en la modelación del problema información del contexto. A diferencia de otros tipos de anomalías se asume que

se conoce información sobre ciertos atributos que no indican que un elemento sea anómalo y se utiliza esta información durante el proceso de detección.

Las técnicas que reducen el problema a detección de anomalías puntuales, son recomendables para dominios de aplicación donde el contexto de los elementos se pueda obtener de modo directo y donde no sea necesario utilizar información sobre las relaciones entre ellos, por lo que no son capaces de aprovechar toda la información que brindan las redes sociales. Algunas de estas técnicas basadas en modelos estadísticos, pueden necesitar varios parámetros que resultan difíciles de seleccionar, los cuales son necesarios para ajustar el modelo.

Las técnicas que emplean la estructura de los datos tratan de adaptarse a un dominio de aplicación específico y a lo que se considera anómalo en él. La complejidad de estas técnicas depende en gran medida de la estructura de los datos, por lo general se intenta aprovechar en ellas toda la información que brindan los datos. Debido a la naturaleza de estas técnicas, este es un tema en el que se pueden realizar futuras investigaciones, ya que es posible desarrollar tantas técnicas como dominios de aplicación existan.

7. Detección de reglas raras

Dentro de la minería de reglas de asociación, el problema de la minería de reglas raras ha sido uno de los menos estudiados, mientras que otros, como la generación eficiente de reglas frecuentes, han sido bastante tratados. En teoría, muchos de los algoritmos para detectar reglas de asociación frecuentes están preparados para encontrar reglas raras, pero en la practica se vuelven intratables si se reduce el umbral mínimo del soporte lo necesario para detectar este tipo de reglas.

Los algoritmos de detección de reglas raras suelen dividirse en dos etapas. La primera fase, busca todos los conjuntos de *itemsets* cuyo valor de soporte se encuentre en un intervalo determinado. La segunda fase consiste en generar reglas de asociación para esos *itemsets* teniendo como restricción que el valor de confianza de estas supere un determinado umbral.

El método utilizado por las técnicas de detección de reglas raras para generar las reglas de asociación consiste normalmente en, para cada *itemset* infrecuente, obtener todos los pares correspondientes a dividir dicho *itemset* en dos subconjuntos disjuntos, luego para cada par se crea una regla de asociación que utiliza los subconjuntos de este como antecedente y consecuente respectivamente. Por último se utiliza alguna medida de calidad (comúnmente la confianza) para seleccionar las reglas más relevantes. Las variaciones en este procedimiento y en la detección de *itemsets* infrecuentes, son los factores que diferencian una técnica de detección de reglas raras de otra.

En [39] se propone una técnica para la detección de reglas raras. Los autores toman el algoritmo Apriori [40] y lo modifican para poder encontrar *itemsets* infrecuentes. La modificación propuesta al algoritmo Apriori es llamada Apriori-Rare [41] y permite encontrar *itemsets* mínimos raros, los cuales se definen a continuación:

Definición 39 (*Itemset* mínimo raro). *Sea D un dominio de aplicación, I_{S_x} un* itemset *en él, D_B una base de datos en D, δ_{sup} un umbral. Entonces I_{S_x} es un* itemset *mínimo raro si:*

i) $Sup(I_{S_x}, D_B) < \delta_{sup}$,
ii) $\forall I_{S_y} \subset I_{S_x}$ *se tiene que* $Sup(I_{S_y}, D_B) > \delta_{sup}$.

Los autores proponen en [41] que todos los *itemsets* mínimos raros son generadores, por lo que se pueden utilizar para generar todos los *itemsets* infrecuentes del dominio de aplicación. La

generación de los *itemsets* infrecuentes a partir de los *itemsets* mínimos raros puede ser llevada a cabo de modo combinatorio. Por ultimo se utilizan estos *itemsets* para obtener un conjunto de reglas raras, no redundantes, tales que permitan representar todas las asociaciones entre los *itemsets* raros que posean una alta confianza.

La ventaja de esta técnica es que brinda la posibilidad de encontrar un conjunto mínimo de reglas raras, las cuales permiten minimizar el espacio necesario para representar las asociaciones de alta confianza entre los *itemsets* mínimos raros. No obstante las ventajas de esta técnica, el algoritmo Apriori-Rare falla en encontrar todos los *itemsets* raros mínimos.

Los algoritmos de minería de *itemsets* infrecuentes, comúnmente son modificaciones de algoritmos de minería de *itemsets* frecuentes y cuando se utilizan, detectan tanto los primeros como los segundos. El problema con lo anterior es la explosión en el número de *itemsets* detectados cuando el soporte mínimo se disminuye lo suficiente como para detectar *itemsets* infrecuentes. Esto conlleva que al ser utilizados en la generación de reglas, no solo se requiera más tiempo para generar las reglas, sino que se obtengan un gran número de reglas sin ningún tipo de interés para los analistas. El problema antes mencionado se suele denominar problema de los *items* raros [42].

La técnica presentada en [43] intenta resolver el problema de los *items* raros mediante la utilización de múltiples soportes mínimos. Esta técnica utiliza el concepto de soporte mínimo de un ítem o MIS (por sus siglas en inglés *Minimum Item Support*).

Definición 40 (Soporte mínimo de un ítem o MIS). *Sea D un dominio de aplicación, x un ítem de él, D_B una base de datos en D, β_{sup} y l_{sup} dos constantes entre 0 y 1. Entonces el MIS de x se define como:*

$$MIS(x, D_B, \beta_{sup}, l_{sup}) = \begin{cases} \beta_{sup} Sup(\{x\}, D_B) & \beta_{sup} Sup(\{x\}, D_B) > l_{sup}, \\ Sup(\{x\}, D_B) & \beta_{sup} Sup(\{x\}, D_B) \leq l_{sup} \end{cases} . \tag{14}$$

.

En la definición anterior, β_{sup} y l_{sup} son parámetros definidos por el usuario, el primero representa un porcentaje y el segundo, un valor mínimo para el soporte. El valor de este soporte es utilizado en la generación de los *itemsets*, pues no se generan aquellos *itemsets* tales que su soporte sea menor que el soporte del ítem de menor soporte que pertenece a él. Además se utiliza el MIS durante la extracción de reglas, donde para que una regla sea válida, tiene que cumplir que su soporte sea mayor o igual que el menor MIS de los *items* que la componen.

El principal beneficio de esta técnica es que permite detectar reglas raras, pero al reducir el número de *itemsets* generados y el número de reglas extraídas de estos, aumenta su eficiencia, al mismo tiempo que reduce la cantidad de reglas que resultan de poco o ningún interés para los analistas. La mayor desventaja de esta técnica es que el resultado depende en gran medida del valor β_{sup} definido por el usuario, lo que limita su flexibilidad.

Obtener reglas que resulten de verdadera utilidad es uno de los mayores desafíos de las técnicas de detección de reglas raras. En [18] se propone enfrentar este problema con un método multiobjetivo de generación de reglas que permite filtrar las reglas que van a ser generadas, basándose en más características además del soporte y la confianza.

Los autores proponen un algoritmo llamado FRIMA (por sus siglas en inglés *Frequent Rare Itemset Mining Algorithm*) para detectar *itemsets* tanto raros como frecuentes. Este algoritmo tiene un funcionamiento similar al algoritmo Apriori y hace uso de la propiedad de la clausura descendente. En general FRIMA suele generar una cantidad de *itemsets* mayor o igual que el algoritmo Apriori-Rare.

Una vez que se han generado los *itemsets* de interés utilizando FRIMA, los autores proponen aplicar un método multi-objetivo de generación de reglas raras al cual llaman RARMA (por sus siglas en inglés *Rare Association Rule Mining Algorithm*). RARMA resuelve un problema de optimización multi-objetivo mediante el uso de un algoritmo genético, permitiendo generar reglas con los mejores valores para varios parámetros entre los que se pueden encontrar confianza, interés y comprensibilidad, entre otros.

La ventaja más evidente de esta técnica es que permite generar reglas que cumplan varios parámetros a la vez. No obstante es importante mencionar que el algoritmo utilizado para generar los *itemsets* raros y frecuentes, muestra menor rendimiento en algunos conjuntos de datos que otros algoritmos existentes para esta tarea. En cuanto a RARMA, es importante notar que al hacer uso de un algoritmo genético, el cual es no determinista, hace que el proceso de generación de reglas funcione como una "caja negra" para el analista que va a utilizar las reglas.

7.1. Conclusiones parciales

Las reglas raras nos brindan información sobre las asociaciones entre los elementos atípicos de un conjunto de datos. Aunque un elemento atípico no necesariamente es anómalo, una de las características fundamentales de las anomalías es que son atípicas. Las técnicas de minería de reglas raras pueden ser utilizadas para extraer asociaciones entre los elementos atípicos de un conjunto de datos de entrenamiento y luego utilizar las reglas extraídas para detectar estas asociaciones atípicas en el dominio de aplicación.

La mayor ventaja de las reglas raras es que, a pesar de que es costoso generarlas, una vez obtenidas, se puede verificar su cumplimiento de forma eficiente lo cual es conveniente para su aplicación en grandes conjuntos de datos como las redes sociales. También es importante mencionar que las reglas raras generadas por estas técnicas permiten a los analistas comprender como se relacionan los elementos atípicos, a diferencia de otros algoritmos que funcionan como "cajas negras". La mayor desventaja de utilizar reglas con estos fines, es que en conjuntos de datos en constante cambio, como las redes sociales, es necesario volver a generar las reglas cada cierto intervalo de tiempo, para que verdaderamente reflejen las relaciones entre los datos.

Todas las técnicas analizadas, sufren del problema de los *items* raros, lo que conlleva que de modo general se dificulte la extracción de reglas de interés para los analistas. Se utilizan distintos enfoques para evitar este problema, desde extraer un conjunto reducido no redundante de reglas, hasta algoritmos genéticos para extraer las mejores reglas de acuerdo a varios parámetros. Aun así es necesario realizar futuras investigaciones en este tema para desarrollar técnicas que generen todos los *itemsets* raros y eviten el problema de los *items* raros, permitiendo en una fase posterior la extracción de reglas verdaderamente significativas para los analistas.

8. Bases de datos usadas en las experimentaciones

Varios de los trabajos analizados en este reporte brindan información sobre las bases de datos que utilizan para evaluar sus propuestas. En la tabla 1 se presenta un resumen de las características fundamentales de estas bases de datos. Las características que se tuvieron en cuenta para cada base de datos fueron: el nombre, una breve descripción, la cantidad de instancias que contiene, si existen relaciones explícitas entre los elementos y su disponibilidad.

Tabla 1. Bases de datos reportadas en las técnicas analizadas.

	Nombre	**Descripción**	**# de instancias**	**Relaciones explícitas entre los datos**	**Disponibilidad**
	DBLP [44]	Información bibliográfica de las principales publicaciones y eventos sobre ciencia de la computación.	2850187 publicaciones, 1510732 autores, 3706 conferencias, 1385 revistas	Si	Pública.
Repositorio UCI [45]	El Nino	Datos de lecturas oceanográficas y meteorológicas de la superficie, tomados mediante boyas colocadas a lo largo del Pacifico ecuatorial.	178080	No	
	KDD Cup 1999	Registros de conexiones generadas en una simulación de una red militar .	4000000	No	
	Iris	Clasificación de flores.	150	No	
	Wine	Análisis químico de vinos italianos.	178	No	
	Twenty Newsgroups	Mensajes recolectados de 20 cadenas de noticias.	20000	No	

Muchos de los trabajos analizados en este reporte utilizan en su experimentación bases de datos sintéticas a las cuales no brindan acceso. Todas las bases de datos mostradas en la tabla 1 se pueden descargar gratuitamente desde sus respectivos sitios web.

Es importante señalar que, de las bases de datos mostradas, ninguna es específica para la detección de anomalías, lo que dificulta la validación de los resultados. La base de datos KDD Cup 1999 es la más apropiada para aplicar detección de anomalías, debido a que fue creada para detectar ataques en una red militar y esto es una aplicación de la detección de anomalías, sin embargo no contiene relaciones explícitas entre los elementos, lo que impide la detección de ciertos tipos de anomalías.

La mayoría de las técnicas de detección de anomalías colectivas validan su eficacia en conjuntos de datos sintéticos y luego para probar su efectividad en redes reales utilizan la base de datos DBLP. Esta base de datos tiene una estructura muy particular, propia de un grafo bipartito, pues almacena información sobre autores, eventos y publicaciones. Ninguno de los trabajos analizados utiliza una base de datos pública con relaciones explícitas entre sus elementos y que sea específica para la detección de anomalías. La carencia de bases de datos de este tipo es una dificultad que hay que tener en cuenta si se desea desarrollar una técnica de detección de anomalías en redes sociales.

9. Medidas de calidad utilizadas en las experimentaciones

Las técnicas de detección de anomalías se utilizan en muchos dominios de aplicación debido a su capacidad de detectar elementos atípicos sin necesidad de definir a priori qué características se consideran anómalas en un elemento, sin embargo, la tendencia de ellas a la detección de falsos positivos es una de sus mayores desventajas. Con el objetivo de evaluar el desempeño de las distintas técnicas de detección de anomalías se utilizan numerosas medidas de calidad, en esta sección se enumeran algunas de las más utilizadas. Antes de hablar de las medidas de calidad es importante mencionar los siguientes conceptos:

Definición 41 (Verdadero positivo). *Sea D un dominio de aplicación y χ_D un algoritmo de detección de anomalías, se le llama verdadero positivo a aquel elemento de D que es anómalo y fue identificado como tal por χ_D.*

Definición 42 (Falso positivo). *Sea D un dominio de aplicación y χ_D un algoritmo de detección de anomalías, se le llama falso positivo a aquel elemento de D que es normal y fue identificado como anómalo por χ_D.*

Definición 43 (Verdadero negativo). *Sea D un dominio de aplicación y χ_D un algoritmo de detección de anomalías, se le llama verdadero negativo a aquel elemento de D que es normal y fue identificado como tal por χ_D.*

Definición 44 (Falso negativo). *Sea D un dominio de aplicación y χ_D un algoritmo de detección de anomalías, se le llama falso negativo a aquel elemento de D que es anómalo y fue identificado como normal por χ_D.*

Utilizando los conceptos anteriores se definen varias medidas de calidad. Una de ellas es la llamada precisión (en inglés, *Precision*) que tiene en cuenta la cantidad de anomalías detectadas que son anomalías reales. Esta medida se define como se muestra a continuación:

Definición 45 (Precisión). *Sea D un dominio de aplicación, χ_D un algoritmo de detección de anomalías, TP la cantidad de verdaderos positivos de χ_D en D y FP la cantidad de falsos positivos de χ_D en D. Entonces se define la precision del algoritmo como:*

$$\frac{TP}{TP+FP}.$$

En ocasiones se quiere tener una noción acerca de qué proporción de los elementos no anómalos fueron detectados como tal. La medida de calidad conocida como tasa de falsas alarmas (del inglés, *False Alarm rate*) se utiliza con este fin y se define como se muestra a continuación:

Definición 46 (Tasa de falsas alarmas). *Sea D un dominio de aplicación, χ_D un algoritmo de detección de anomalías, FP la cantidad de falsos positivos de χ_D en D y TN la cantidad de verdaderos negativos de χ_D en D. Entonces se define la tasa de falsas alarmas como:*

$$\frac{FP}{FP+TN}.$$

Uno de los aspectos más importantes a tener en cuenta cuando se analiza la eficacia de un algoritmo de detección de anomalías es la cantidad de elementos anómalos que fueron detectados como tal. Una de las medidas de calidad utilizada para eso es la llamada *Recall* definida a continuación:

Definición 47 (*Recall*). *Sea D un dominio de aplicación, χ_D un algoritmo de detección de anomalías, TP la cantidad de verdaderos positivos de χ_D en D y FN la cantidad de falsos negativos de χ_D en D. Entonces se define la medida de calidad Recall como:*

$$\frac{TP}{TP+FN}.$$

Es deseable que las medidas de calidad tomen en cuenta tanto los falsos positivos como los falsos negativos para buscar un balance entre ambos. La F-medida (en inglés, *F-Measure*) hace esto, a la vez que le otorga más peso a los verdaderos positivos. A continuación se muestra su definición:

Definición 48 (F-Medida). *Sea D un dominio de aplicación, χ_D un algoritmo de detección de anomalías, TP la cantidad de verdaderos positivos de χ_D en D, FP la cantidad de falsos positivos de χ_D en D y FN la cantidad de falsos negativos de χ_D en D. Entonces se define la F-medida como:*

$$\frac{2TP}{2TP+FP+FN}.$$

Una medida de calidad que se utiliza con mucha frecuencia es el AUC (por sus siglas en inglés, *Area Under receiver operating characteristic Curve*). Esta medida se utiliza para determinar el rendimiento general de un clasificador sin tener en cuenta la diferencia existente entre la cantidad de verdaderos positivos y verdaderos negativos. El AUC se define como se muestra a continuación:

Definición 49 (AUC). *Sea D un dominio de aplicación, χ_D un algoritmo de detección de anomalías, RA la cantidad de verdaderas anomalías en D, RN la cantidad de elementos normales en D, D_{R_k} una lista ordenada de mayor a menor que contiene el grado de atipicidad asignado por*

χ_D a cada elemento de D y $R_{k_s} = \sum_{i=1}^{RA} R_{k_i}$ donde R_{k_i} es el i-ésimo elemento en D_{R_k}. Entonces el AUC se define como:

$$\frac{R_{k_s} - (RA^2 + RN)/2}{RA * RN}.$$

La selección de una medida de calidad para evaluar un algoritmo debe realizarse teniendo en cuenta el dominio de aplicación específico en el que este va a ser utilizado. En dominios como la detección de fraude en redes de telecomunicaciones, los falsos positivos se consideran muy costosos para la compañía, mientras que en el dominio de la detección de intrusiones en redes empresariales es preferible un falso positivo a un falso negativo. Cuando se desea evaluar la eficacia de un algoritmo de modo general es recomendable el uso de varias medidas de calidad, de este modo se puede tener una idea de en qué dominios de aplicación sería más apropiado utilizar dicho algoritmo.

10. Discusión

En esta sección se muestran algunas características importantes para los algoritmos de detección de anomalías en redes sociales y luego se realiza una comparación cualitativa de los algoritmos tratados en este atendiendo a dichas características.

Las principales características a tener en cuenta cuando se analiza un algoritmo para la detección de anomalías en el dominio de las redes sociales son las siguientes:

- **Complejidad computacional:** El aumento en la cantidad de usuarios de las redes sociales en los últimos años ha sido increíble por lo que una característica fundamental de las redes sociales es la gran cantidad de elementos que las componen. Como consecuencia de lo anterior, los algoritmos de detección de anomalías en redes sociales deben tratar de tener la menor complejidad computacional posible para poder analizar la red en un tiempo aceptable.
- **Cantidad de parámetros:** La cantidad de parámetros que recibe un algoritmo de detección de anomalías, así como la dificultad para determinar los mismos a priori, es un tema a tener en cuenta durante la selección de un algoritmo. La dificultad para determinar los parámetros es subjetiva, y depende, en gran medida, del dominio de aplicación, por lo que no se va a tener en cuenta al comparar los algoritmos. Los parámetros de un algoritmo, si bien permiten ajustar el desempeño de este, pueden ser el origen de problemas de eficiencia o eficacia si son mal seleccionados. Como resultado de esto, se considera que es deseable que un algoritmo de detección de anomalías en redes sociales infiera más información de los datos y requiera una menor cantidad de parámetros para su funcionamiento.
- **Relaciones entre los datos:** Una de las características más importante de las redes sociales es que existen relaciones entre sus datos, esta información permite la existencia de anomalías colectivas en ellas y resulta de gran importancia que se tenga en cuenta por los algoritmos de detección de anomalías.
- **Información contextual:** El principal problema de los algoritmos de detección de anomalías son los falsos positivos. El uso de información contextual permite reducir el número de falsos positivos de los algoritmos al analizar cada elemento en relación con su contexto, también permite detectar elementos que parecen normales al analizarse en relación con la totalidad del conjunto de datos al que pertenecen, pero que son anómalos en el contexto en que se

encuentran. Las redes sociales son un buen dominio para aplicar estas técnicas, pues su riqueza expresiva no solo permite utilizar atributos contextuales, si no también inferirlos.

- **Tipo de anomalía detectado:** En las redes sociales existen varios tipos de anomalías y ningún algoritmo es eficaz en la detección de todas, por lo que es importante tomar en consideración el tipo de anomalías que detectan antes de efectuar una comparación entre ellos. Un ejemplo de lo antes mencionado son los algoritmos de detección de anomalías puntuales y los de colectivas, los segundos suelen tener una complejidad computacional mayor que los primeros, pero esto se debe a la complejidad de las anomalías que detectan.

En la tabla 2 se muestra una comparación cualitativa de los algoritmos para la detección de anomalías en redes sociales analizados en este reporte.

Tabla 2. Comparación de los algoritmos para detección de anomalías en redes sociales.

Algoritmo	Complejidad computacional	Cantidad de parámetros	Relaciones entre los elementos	Utiliza información contextual	Tipo de anomalía detectada
Knorr [20]	Media	2	-	-	Anomalías puntuales
LOF [11]	Alta	1	-	-	
DBSCAN [23]	Alta	2	-	-	
BBC [24]	Media	2*	-	-	Anomalías agrupadas
AGM [15]	Media	1	-	-	
iForest [25]	Baja	3	-	-	
SCiForest [9]	Baja	4	-	-	
OutRank [27]	Alta	2	-	-	
OddBall [30,31]	Baja	1	√*	-	Anomalías colectivas
DSEA [33]	Alta	0	√	-	
DSGA [33]	Alta	0	√	-	
Akoglu y Faloutsos [37]	Alta	1	√*	-	
CAD [16]	Baja*	1*	-	√	Anomalías contextuales
Sun et al. [38]	Alta	0	√*	√*	
Gao et al. [17]	Muy alta	3	√	√	
Szathmary et al. [41]	Baja*	1	√*	-	Reglas raras
Bansal et al. [43]	Baja*	2	√*	-	
RARMA [18]	Baja*	1*	√*	-	

El asterisco en la columna referente a la cantidad de parámetros del algoritmo BBC indica que este algoritmo puede recibir menos parámetros si se utiliza la técnica llamada presurización, pero se afectaría la eficiencia del algoritmo.

El asterisco del algoritmo OddBall [30,31] y del de Akoglu y Faloutsos [37] en la columna referente al uso de información sobre las relaciones de los elementos indica que estos utilizan esta información de un modo limitado. Lo anterior es el resultado de que estos algoritmos reducen la detección de anomalías colectivas a detección de anomalías puntuales.

El algoritmo CAD [16] tiene un asterisco en la columna de complejidad computacional debido a que es eficiente en la fase de evaluación, pero los autores no tienen en cuenta el costo de tener que ajustar el Modelo de Mezcla Gaussiano a los datos. En la columna parámetros aparece un asterisco porque el parámetro que recibe el algoritmo es un conjunto de parámetros para ajustar el modelo, los cuales no son detallados por los autores, pero cuya selección es una de las mayores dificultades para aplicar este algoritmo en la práctica.

La técnica propuesta por Sun et al. [38], aunque tiene en cuenta las relaciones y el contexto de los elementos, lo hace asumiendo que el grafo que modela la red es un grafo bipartito, por lo que no se puede aplicar a las redes sociales en general.

El algoritmo RARMA [18] recibe un conjunto de parámetros que tratará de optimizar en las reglas generadas.

Las técnicas de minería de reglas raras resultan costosas en la fase de extracción de las reglas, sin embargo, una vez extraídas pueden ser utilizadas para detectar la existencia de relaciones entre elementos atípicos en el conjunto de datos de forma eficiente. Las reglas raras pueden ser utilizadas para detectar anomalías colectivas en redes sociales, aunque no es una tarea sencilla ya que estas técnicas están diseñadas para ser utilizadas en bases de datos transaccionales.

La información presentada en la tabla 2 permite realizar varias observaciones. Se puede ver que los algoritmos de detección de anomalías agrupadas pueden tener una complejidad computacional menor o igual que los de detección de anomalías puntuales, por lo tanto como las anomalías puntuales son un caso particular de las agrupadas, resulta más interesante descubrir las segundas. Las tecnicas de deteccion de reglas raras resultan muy eficientes en la detección y pueden incluso detectar algunas anomalías colectivas, aunque utilizarlas con este fin en las redes sociales puede resultar complicado. Por ultimo es importante notar que de todos los algoritmos analizados solo dos utilizan toda la información disponible en las redes sociales durante la detección y de ellos solo el propuesto por Gao et al. [17], el cual detecta anomalías colectivas contextuales, puede ser utilizado en una red social cualquiera, pero posee una complejidad computacional muy alta.

Las observaciones realizadas muestran que es necesario realizar futuras investigaciones para desarrollar algoritmos de detección de anomalías en redes sociales que utilicen toda la información que brindan estas redes, obtengan un bajo número de falsos positivos y tengan una complejidad computacional apropiada para ser aplicados a redes reales.

11. Conclusiones generales

La detección de anomalías es un campo de la minería de datos de gran importancia, sin embargo debido a la propia naturaleza de estas resulta difícil definir este problema de modo general. En el presente reporte se han analizado los principales tipos de anomalías existentes y las técnicas más importantes para su detección, explicando algunas de ellas detenidamente para que se pueda comprender su funcionamiento. Además, se ha mostrado que uno de los mayores retos que enfrentan estas técnicas es definir la frontera que separa lo normal de lo anómalo, la cual en la mayoría de los casos no es clara e incluso puede variar con el tiempo.

En dependencia de las características del dominio de aplicación es posible la existencia en él, de ciertos tipos de anomalías. Desarrollar una técnica que detecte todos los tipos de anomalías con calidad y eficiencia, resulta imposible, por lo que la mayoría de ellas se centran en detectar un tipo específico de anomalía en un dominio de aplicación determinado. Es especialmente importante limitar la cantidad de falsos positivos en los resultados obtenidos tras aplicar estas técnicas, ya que estos son, en algunos dominios de aplicación, más perjudiciales que los falsos negativos.

Las redes sociales son un dominio de aplicación con características muy interesantes, ya que brindan gran cantidad de información y es posible encontrar en ellas todos los tipos de anomalías analizados en este trabajo. La gran cantidad de elementos que poseen las redes sociales, así como la evolución de su estructura en el tiempo, son los principales retos que enfrentan las técnicas de detección de anomalías.

Existen muchas direcciones prometedoras para futuras líneas de investigación en la detección de anomalías en redes sociales. Las técnicas de minería de reglas raras existentes en la actualidad son afectadas por el problema de los *items* raros que provoca que las reglas obtenidas no siempre resulten de interés para los usuarios, por lo que es un gran reto desarrollar nuevas técnicas capaces de sobreponerse a este problema. Las técnicas de detección de anomalías agrupadas han sido poco estudiadas a pesar de que también son capaces de detectar anomalías puntuales. En relación con las anomalías agrupadas, sería interesante tomar la idea de las técnicas que reducen los problemas de anomalías colectivas y contextuales a detección de anomalías puntuales y en su lugar reducirlos a detección de anomalías agrupadas. Las técnicas orientadas a detectar anomalías colectivas y contextuales han aumentado su popularidad en los últimos tiempos, debido entre otros factores al auge de las redes sociales. La detección de anomalías contextuales es una línea de investigación especialmente prometedora, debido a que estas técnicas han sido relativamente poco estudiadas y porque pueden combinarse con otras para obtener información contextual no explícita en los datos, como puede ser la comunidad a la que pertenece un elemento en una red social.

En general con este trabajo se muestra una visión general de un problema muy vasto y de vital importancia para la sociedad actual. La intención de este reporte es servir de punto de partida para futuras investigaciones sobre detección de anomalías en redes sociales.

Referencias bibliográficas

1. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys (CSUR) **41**(3) (2009) 15
2. Pandit, S., Chau, D.H., Wang, S., Faloutsos, C.: Netprobe: a fast and scalable system for fraud detection in online auction networks. In Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J., eds.: WWW, ACM (2007) 201–210
3. Michalak, K., Korczak, J.J.: Graph mining approach to suspicious transaction detection. In Ganzha, M., Maciaszek, L.A., Paprzycki, M., eds.: FedCSIS. (2011) 69–75
4. Becker, R.A., Volinsky, C., Wilks, A.R.: Fraud detection in telecommunications: History and lessons learned. Technometrics **52**(1) (2010)
5. Thottan, M., Liu, G., Ji, C.: Anomaly detection approaches for communication networks. In Cormode, G., Thottan, M., eds.: Algorithms for Next Generation Networks. Computer Communications and Networks. Springer London (2010) 239–261
6. Chandola, V., Banerjee, A., Kumar, V.: Outlier detection: A survey. ACM Computing Surveys (2007)
7. Markou, M., Singh, S.: Novelty detection: a review-part 1: statistical approaches. Signal processing **83**(12) (2003) 2481–2497
8. Hawkins, D.M.: Identification of outliers. Volume 11. Springer (1980)
9. Liu, F.T., Ting, K.M., Zhou, Z.H.: On detecting clustered anomalies using sciforest. In Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M., eds.: ECML/PKDD (2). Volume 6322 of Lecture Notes in Computer Science., Springer (2010) 274–290
10. Zhang, J.: Advancements of outlier detection: a survey. ICST Transactions on Scalable Information Systems **13**(1) (2013) 1–26
11. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: ACM Sigmod Record. Volume 29., ACM (2000) 93–104
12. Xiong, Y., Zhu, Y.: Mining peculiarity groups in day-by-day behavioral datasets. In 0010, W.W., Kargupta, H., Ranka, S., Yu, P.S., Wu, X., eds.: ICDM, IEEE Computer Society (2009) 578–587
13. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: WWW '08: Proceeding of the 17th international conference on World Wide Web, New York, NY, USA, ACM (2008) 695–704
14. Gogoi, P., Bhattacharyya, D., Borah, B., Kalita, J.K.: A survey of outlier detection methods in network anomaly identification. The Computer Journal **54**(4) (2011) 570–588
15. Xiong, Y., Zhu, Y., Yu, P.S., Pei, J.: Towards cohesive anomaly mining. In desJardins, M., Littman, M.L., eds.: AAAI, AAAI Press (2013)

16. Song, X., Wu, M., Jermaine, C., Ranka, S.: Conditional anomaly detection. Knowledge and Data Engineering, IEEE Transactions on **19**(5) (2007) 631–645
17. Gao, J., Liang, F., Fan, W., Wang, C., Sun, Y., Han, J.: On community outliers and their efficient detection in information networks. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2010) 813–822
18. Hoque, N., Nath, B., Bhattacharyya, D.: An efficient approach on rare association rule mining. In Bansal, J.C., Singh, P.K., Deep, K., Pant, M., Nagar, A.K., eds.: Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012). Volume 201 of Advances in Intelligent Systems and Computing. Springer India (2013) 193–203
19. Romero, C., Romero, J.R., Luna, J.M., Ventura, S.: Mining rare association rules from e-learning data. In: EDM, ERIC (2010) 171–180
20. Knorr, E.M.: Outliers and Data Mining: Finding Exceptions in Data. PhD thesis, The University of British Columbia (2002) AAINQ73191.
21. Guha, S., Rastogi, R., Shim, K.: Rock: A robust clustering algorithm for categorical attributes. In: Data Engineering, 1999. Proceedings., 15th International Conference on, IEEE (1999) 512–521
22. Ertöz, L., Steinbach, M., Kumar, V.: Finding topics in collections of documents: A shared nearest neighbor approach. Springer (2004)
23. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. Volume 96. (1996) 226–231
24. Gupta, G., Ghosh, J.: Bregman bubble clustering: A robust framework for mining dense clusters. ACM Trans. Knowl. Discov. Data **2**(2) (July 2008) 8:1–8:49
25. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD) **6**(1) (2012) 3
26. Moonesinghe, H.D.K., Tan, P.N.: Outlier detection using random walks. In: ICTAI, IEEE Computer Society (2006) 532–539
27. Moonesinghe, H., Tan, P.N.: Outrank: a graph-based outlier detection framework using random walk. International Journal on Artificial Intelligence Tools **17**(01) (2008) 19–36
28. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection for discrete sequences: A survey. Knowledge and Data Engineering, IEEE Transactions on **24**(5) (2012) 823–839
29. Akoglu, L., Tong, H., Koutra, D.: Graph-based anomaly detection and description: A survey. CoRR **abs/1404.4679** (2014)
30. Akoglu, L., McGlohon, M., Faloutsos, C.: Anomaly detection in large graphs. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 (November 2009)
31. Akoglu, L., McGlohon, M., Faloutsos, C.: Oddball: Spotting anomalies in weighted graphs. In: Advances in Knowledge Discovery and Data Mining. Springer (2010) 410–421
32. Eberle, W., Holder, L.: Discovering structural anomalies in graph-based data. In: Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on, IEEE (2007) 393–398
33. Noble, C.C., Cook, D.J.: Graph-based anomaly detection. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2003) 631–636
34. Bilgin, C.C., Yener, B.: Dynamic network evolution: Models, clustering, anomaly detection. IEEE Networks (2006)
35. Gaston, M.E., Kraetzl, M., Wallis, W.D.: Using graph diameter for change detection in dynamic networks. Australasian Journal of Combinatorics **35** (2006) 299
36. Berlingerio, M., Koutra, D., Eliassi-Rad, T., Faloutsos, C.: Netsimile: a scalable approach to size-independent network similarity. arXiv preprint arXiv:1209.2684 (2012)
37. Akoglu, L., Faloutsos, C.: Event detection in time series of mobile communication graphs. In: Army Science Conference. (2010) 77–79
38. Sun, J., Qu, H., Chakrabarti, D., Faloutsos, C.: Neighborhood formation and anomaly detection in bipartite graphs. In: Data Mining, Fifth IEEE International Conference on, IEEE (2005) 8–pp
39. Szathmary, L., Valtchev, P., Napoli, A.: Finding minimal rare itemsets and rare association rules. In: Knowledge Science, Engineering and Management. Springer (2010) 16–27
40. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD Record. Volume 22., ACM (1993) 207–216
41. Szathmary, L., Napoli, A., Valtchev, P.: Towards rare itemset mining. In: Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on. Volume 1., IEEE (2007) 305–312
42. Liu, B., Hsu, W., Ma, Y.: Mining association rules with multiple minimum supports. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (1999) 337–341

43. Bansal, A., Baghel, N., Tiwari, S.: An novel approach to mine rare association rules based on multiple minimum support approach. International Journal of Advanced Electrical and Electronics Engineering, (IJAEEE) 10.1214/IJAEEE/145 **2**(4) (2013) 75–80
44. Universität Trier: Dblp computer science bibliography. [citado 15 de enero de 2015]. Disponible en Internet: http://dblp.org/db/
45. UCI Repository: UC Irvine machine learning repository. [citado 15 de enero de 2015]. Disponible en Internet: http://archive.ics.uci.edu/ml/

Anexo 1

En este anexo se introducen las definiciones básicas necesarias para la comprensión de las técnicas de detección de anomalías expuestas en el presente reporte.

1.1. Elementos de la teoría de conjuntos

A continuación se definen algunos conceptos necesarios para definir el dominio sobre el cual se van a aplicar las técnicas de detección de anomalías.

Definición 50 (Universo). *Sea m un número entero positivo, $m \geq 2$. La mayor parte de los métodos analizados en este reporte procesan objetos que se encuentran en un espacio m-dimensional conocido como universo y denotado por U.*

El universo antes definido es el utilizado por las técnicas expuestas en este trabajo, a menos que se especifique lo contrario.

Definición 51 (Dominio de aplicación). *El dominio de aplicación, denotado por D, es un conjunto $D \subset U$ sobre el cual resulta de interés aplicar las técnicas de detección de anomalías.*

Definición 52 (Conjunto potencia). *El conjunto potencia de un conjunto B es el conjunto que contiene todos los subconjuntos de B y se denota por 2^B.*

Algunas de las técnicas expuestas en este reporte requieren que el conjunto de datos sobre el que se apliquen sea convexo. A continuación se muestra la definición de este tipo de conjuntos:

Definición 53 (Conjunto convexo). *Sea B un espacio vectorial, B' un subconjunto de él, entonces se considera que B' es convexo si para todo $x, y \in B'$ y para todo $\lambda \in [0, 1]$ se tiene que el punto $(1 - \lambda)x + \lambda y$ pertenece a B'.*

Una clase especial de conjuntos en $\mathbb{R}^n$ son los hiperplanos, los cuales se definen a continuación:

Definición 54 (Hiperplano). *Sean $\alpha \in \mathbb{R}^n$ con $\alpha \neq 0$ y $\beta \in \mathbb{R}$, entonces un hiperplano en $\mathbb{R}^n$ es un conjunto de la forma:*

$$h_p = \{x | \alpha^T x = \beta\}.$$

1.2. Teoría de grafos

En esta sección se muestran los conceptos relacionados con la teoría de grafos necesarios para comprender algunos de los temas tratados en este trabajo.

Definición 55 (Grafo). *Un grafo es una tupla $\langle V, E \rangle$, donde:*

- *V es un conjunto cuyos elementos son llamados vértices,*
- *$E \subset \{\{u, v\} | u, v \in V, u \neq v\}$ es un conjunto cuyos elementos son llamados aristas.*

El conjunto de vértices de un grafo G se suele denotar como $V(G)$ y el de aristas como $E(G)$. El espacio de todos los grafos se denotará por $\mathbb{G}$ en este reporte.

Definición 56 (Subgrafo). *Sea G un grafo, entonces se le llama subgrafo de G a un grafo S tal que:*

- $V(S) \subseteq V(G)$,
- $E(S) \subseteq \{\{v_1, v_2\} | \{v_1, v_2\} \in E(G) \wedge v_1, v_2 \in V(S)\}$.

Un tipo interesante de subgrafo, que merece la pena tratar debido a su uso en diversas aplicaciones, son las llamadas ego-redes [1]. Una ego-red se define para un vértice específico. A continuación se muestra su definición:

Definición 57 (Ego-red de un vértice). *Sea G un grafo, $v \in V(G)$ un vértice de él, se denomina ego-red de v en G, al subgrafo S tal que $V(S) = \{u \in V(G) | \{v, u\} \in E(G)\}$ y $E(S) = \{\{u, w\} \in E(G) | u, w \in V(s)\}$.*

En algunos grafos las aristas poseen sentido, este tipo de grafos se suele llamar grafos dirigidos.

Definición 58 (Grafo dirigido). *Un grafo dirigido es una tupla $\langle V, E \rangle$, donde:*

- *V es un conjunto cuyos elementos son llamados vértices,*
- *$E \subset \{\langle u, v \rangle | u, v \in V, u \neq v\}$ es un conjunto de tuplas llamadas aristas.*

Existen grafos que poseen una estructura particular que los hace interesantes para determinadas aplicaciones. Un ejemplo de esto son los grafos bipartitos.

Definición 59 (Grafo bipartito). *Un grafo $G = \langle V, E \rangle$ se denomina grafo bipartito, si se puede dividir a V en dos subconjuntos disjuntos V_1 y V_2 donde $V_1 \cup V_2 = V$ y $V_1 \cap V_2 = \varnothing$, tales que para toda arista $\langle v_1, v_2 \rangle \in E$ se tiene que $v_1 \in V_1$ y $v_2 \in V_2$ o $v_2 \in V_1$ y $v_1 \in V_2$.*

Es común que se utilicen grafos cuyos vértices y aristas posean etiquetas.

Definición 60 (Grafo etiquetado). *Un grafo etiquetado y no dirigido es una tétrada $\langle V, E, L, l \rangle$ donde:*

- *$\langle V, E \rangle$ es un grafo,*
- *L es un conjunto de etiquetas,*
- *$l : V \cup E \longrightarrow L$ es una función que asigna etiquetas a vértices y aristas del grafo.*

Esta definición puede generalizarse a grafos parcialmente etiquetados si se incluye la etiqueta vacía ϵ. El conjunto de todos los grafos etiquetados será denotado por $\mathbb{G}_L$ en este reporte.

Un grafo dirigido y etiquetado se define del mismo modo que un grafo etiquetado, la única diferencia es que las aristas son tuplas en lugar de conjuntos, pues el orden de los vértices en la arista de un grafo dirigido es lo que define la dirección de la arista.

Un tipo de grafo etiquetado que se utiliza con mucha frecuencia en distintas aplicaciones son los llamados grafos ponderados.

Definición 61 (Grafo ponderado). *Un grafo etiquetado $G = \langle V, E, L, l \rangle$ es llamado grafo ponderado si $L \subset \mathbb{R}$.*

Definición 62 (Isomorfismo y sub-isomorfismo). *Se dice que f es un isomorfismo entre dos grafos etiquetados $G_1 = \langle V_1, E_1, L_1, l_1 \rangle$ y $G_2 = \langle V_2, E_2, L_2, l_2 \rangle$ si $f : V_1 \longrightarrow V_2$ es una función biyectiva tal que:*

- $\forall v \in V_1, l_1(v) = l_2(f(v))$,
- $\forall \{u, v\} \in E_1, \{f(u), f(v)\} \in E_2 \wedge l_1(\{u, v\}) = l_2(\{f(u), f(v)\})$.

Definición 63 (Camino y ciclo). *Se dice que $P = (v_1, v_2, \ldots, v_k)$ es un camino en un grafo G si todo vértice de P está en $V(G)$ y para cada par de vértices v_i y v_{i+1} (consecutivos en P) se cumple que $\{v_i, v_{i+1}\} \in E(G)$. En tal caso se dice que v_1 y v_k están conectados por P. Si $v_1 = v_k$ entonces se dice que P es un ciclo.*

Definición 64 (Grafo conexo). *Un grafo G es conexo, si todo par de vértices en $V(G)$ están conectados por algún camino.*

Definición 65 (Árbol libre). *Un árbol libre, es un grafo conexo y sin ciclos.*

Cuando en un árbol libre se selecciona un vértice como raíz entonces se tiene un árbol enraizado. En adelante se utilizará el término árbol para referirse a los árboles enraizados.

Definición 66 (Relación padre-hijo en árboles enraizados). *Sea T un árbol enraizado con $v_0 \in V(T)$ seleccionado como raíz y $v, u \in V(T)$ dos vértices cualesquiera de T. Se dice que v es padre de u si $\{v, u\} \in E(T)$ y el camino que une a v con v_0 está completamente contenido en el camino que une a u con v_0. En ese caso se dice que u es un hijo de v.*

Definición 67 (Árbol binario). *Un árbol binario, es un árbol enraizado, donde cada vértice puede tener a lo sumo dos hijos.*

Una de las formas de representar un grafo es mediante una matriz de adyacencia [2]. En la definición 68 se muestra en qué consiste.

Definición 68 (Matriz de adyacencia de un grafo). *Una matriz M se considera de adyacencia de un grafo G = (V, E) si:*

$$m_{ij} = \begin{cases} 1 & (v_i, v_j) \in E, \\ 0 & (v_i, v_j) \notin E, \end{cases} . \tag{15}$$

.

1.3. Distancia y densidad

En esta sección se tratarán las definiciones relacionadas con la distancia entre los elementos de un conjunto y la densidad de los grupos formados por estos. A continuación se define la distancia entre dos elementos, concepto fundamental para el funcionamiento de algunas de las técnicas de detección de anomalías, que se muestran en la sección 3:

Definición 69 (Distancia entre dos elementos). *Sea U un conjunto m-dimensional conocido como universo. Se le llama distancia a una función $d : U \times U \longrightarrow \mathbb{R}$ tal que $\forall x, y, z \in U$ se cumple:*

- $d(x, y) \geq 0$,
- $d(x, y) = 0 \Leftrightarrow x = y$,
- $d(x, y) = d(y, x)$,
- $d(x, z) \leq d(x, y) + d(y, z)$.

En algunos casos es conveniente utilizar funciones más generales que las funciones de distancia, pero que al mismo tiempo mantengan ciertas propiedades de las mismas. Las divergencias de Bregman [3] son un tipo de función con las características antes mencionadas, pero para su comprensión, es necesario definir qué se entiende por función convexa.

Definición 70 (Función convexa). *Una función $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ definida sobre un conjunto convexo, es convexa, si para todo par de puntos x e y que pertenecen a su dominio y para todo $\lambda \in [0, 1]$ se cumple:*

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \tag{16}$$

Si la desigualdad anterior es estricta para todo par de puntos x, y tales que $x \neq y$ donde se tiene además $0 < \lambda < 1$, entonces se dice que f es estrictamente convexa.

Definición 71 (Divergencia de Bregman). *Sea $S \subseteq \mathbb{R}^d$ un conjunto convexo, que se asume no vacío y $\phi : S \longrightarrow \mathbb{R}$ una función estrictamente convexa, definida en S, tal que ϕ es diferenciable en el interior relativo de S(denotado por $ri(S)$). La divergencia de Bregman $d_\phi : S \times ri(S) \longrightarrow [0, \infty)$ se define como:*

$$d_\phi(x, y) = \phi(x) - \phi(y) - \langle x - y, \nabla\phi(y)\rangle,$$

donde $\langle x, y\rangle$ representa el producto interno de dos vectores y $\nabla\phi(y)$ representa el vector gradiente de ϕ evaluada en y.

1.4. Funciones de similitud

Los conceptos que se exponen a continuación, resultan esenciales para la definición de algunos de los tipos de anomalías expuestos en este reporte.

Definición 72 (Función de similitud). *Se le llamará función de similitud, a una función $s : D \times D \longrightarrow [0, 1]$ que determina qué tan similares son dos elementos de D, indicando mayor similitud cuando el valor de la función es más cercano a 1.*

En algunas ocasiones se quiere saber la similitud entre un elemento y los miembros de un conjunto, para ello se define la similitud entre un elemento y un conjunto.

Definición 73 (Función de similitud entre un elemento y un conjunto). *Se le llamará función de similitud entre un elemento y un conjunto de elementos, a una función $s_g : D \times 2^D \longrightarrow [0, 1]$ que determina qué tan similar es un elemento a un conjunto, indicando mayor similitud cuando el valor de la función es más cercano a 1.*

Para determinar cuando los elementos de un conjunto son más similares entre ellos que con los elementos fuera del conjunto resultan convenientes las siguientes definiciones:

Definición 74 (Función de similitud entre conjuntos). *Se le llamará función de similitud entre dos conjuntos, a una función $s_c : 2^D \times 2^D \longrightarrow [0, 1]$ que determina qué tan similares son dos subconjuntos de D, indicando mayor similitud cuando el valor de la función es más cercano a 1.*

Definición 75 (Función de similitud interna de un conjunto). *Se le llamara función de similitud interna de un grupo, a una función* $s_I : 2^D \longrightarrow [0,1]$ *que determina qué tan similares son los elementos de un conjunto, indicando mayor similitud cuando el valor de la función es más cercano a 1.*

1.5. Reglas de asociación

A continuación se expondrán los conceptos esenciales para comprender la detección de anomalías utilizando reglas de asociación. Este tema será tratado en la sección 7 de este trabajo. En primer lugar resulta necesario definir lo que se entiende por ítem.

Definición 76 (Ítem). *Sea D un dominio de aplicación, se le llama ítem a un elemento* $x \in D$.

Los *items* son los elementos que conforman los *itemsets*, los cuales son utilizados para definir las reglas de asociación. A continuación se define qué se entiende por *itemset*:

Definición 77 (Conjunto de *items* o *itemset*). *Sea D un dominio de aplicación, se le llama conjunto de* items *o* itemset*, a un subconjunto* $I_S \subseteq D$.

Utilizando como base los conceptos anteriores se puede definir qué es una regla de asociación.

Definición 78 (Regla de asociación). *Sea D un dominio de aplicación,* I_{S_x} *e* I_{S_y} *dos* itemsets *de él. Entonces se le llama regla de asociación (RA), a una implicación de la forma* $I_{S_x} \Rightarrow I_{S_y}$ *donde* $I_{S_x} \cap I_{S_y} = \varnothing$.

Normalmente, en la minería de reglas de asociación se extraen, a partir de una base de datos, reglas que representan implicaciones entre transacciones de esta. A continuación se definen los conceptos de base de datos y transacción:

Definición 79 (Base de datos y transacción). *Sea D un dominio de aplicación, se le llama base de datos a un conjunto de transacciones* D_B*, donde una transacción es un* itemset *de D.*

Es útil conocer en qué transacciones de una base de datos se encuentra contenido un *itemset*, para ello se define el cubrimiento transaccional de un *itemset* en una base de datos.

Definición 80 (Cubrimiento transaccional de un *itemset* en una base de datos). *Sea D un dominio de aplicación,* I_{S_x} *un* itemset *de él y* D_B *una base de datos de D. Entonces se le llama cubrimiento transaccional de* I_{S_x} *en* D_B *al conjunto:*

$$C_T(I_{S_x}, D_B) = \{T_S \in D_B | I_{S_x} \subseteq T_S\}. \tag{17}$$

Dos conceptos fundamentales para el trabajo con reglas de asociación, son los conceptos de soporte y confianza de una regla. A continuación se muestra la definición de soporte de un *itemset*, necesaria para poder definir el soporte de una regla:

Definición 81 (Soporte de un *itemset*). *Sea D un dominio de aplicación,* I_{S_x} *un* itemset *en él y* D_B *una base de datos en D. Entonces el soporte de* I_{S_x} *en* D_B *está dado por la siguiente expresión:*

$$Sup(I_{S_x}, D_B) = \frac{|C_T(I_{S_x}, D_B)|}{|D_B|}. \tag{18}$$

El concepto de soporte de una regla de asociación es fundamental para la comprensión de las técnicas mostradas en la sección 7. De modo intuitivo, el soporte se puede entender como el porcentaje de las transacciones de una base de datos en las que se encuentran presentes los *itemsets* que conforman una regla.

Definición 82 (Soporte de una regla de asociación en una base de datos). *Sea D un dominio de aplicación y D_B una base de datos en él, $I_{S_x} \Rightarrow I_{S_y}$ una regla de asociación en D_B. Entonces el soporte de $I_{S_x} \Rightarrow I_{S_y}$ en D_B está dado por la expresión:*

$$Sup(I_{S_x} \Rightarrow I_{S_y}, D_B) = Sup(I_{S_x} \cup I_{S_y}, D_B) = \frac{|C_T(I_{S_x} \cup I_{S_y}, D_B)|}{|D_B|}. \tag{19}$$

La confianza de una regla sirve como medida de la veracidad de la asociación entre dos *itemsets* que la misma propone. A continuación se muestra su definición:

Definición 83 (Confianza de una regla de asociación en una base de datos). *Sea D un dominio de aplicación, B_D una base de datos en él, $I_{S_x} \Rightarrow I_{S_y}$ una regla de asociación en D_B. Entonces la confianza de $I_{S_x} \Rightarrow I_{S_y}$ en D_B se define como:*

$$Conf(I_{S_x} \Rightarrow I_{S_y}, D_B) = \frac{Sup(I_{S_x} \Rightarrow I_{S_y}, D_B)}{Sup(I_{S_x}, D_B)}. \tag{20}$$

Una propiedad importante que utilizan algunos algoritmos para generar *itemsets*, es la propiedad de clausura descendente del soporte de los *itemsets*, la cual se define a continuación:

Propiedad 1 (Clausura descendente del soporte de los itemsets*).* El soporte de cualquier subconjunto de un *itemset* es mayor o igual que el soporte de ese *itemset*. Además, todo subconjunto de un *itemset* frecuente es frecuente, mientras que cualquier supraconjunto de un *itemset* infrecuente también es infrecuente.

Referencias bibliográficas del anexo

1. Akoglu, L., McGlohon, M., Faloutsos, C.: Oddball: Spotting anomalies in weighted graphs. In: Advances in Knowledge Discovery and Data Mining. Springer (2010) 410–421
2. Diestel, R.: Graph Theory {Graduate Texts in Mathematics; 173}. Springer-Verlag Berlin and Heidelberg GmbH & Company KG (2000)
3. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with bregman divergences. The Journal of Machine Learning Research **6** (2005) 1705–1749

RT_030, marzo 2015
Aprobado por el Consejo Científico CENATAV

Editor: Lic. Lucía González Bayona
Diseño de Portada: Di. Alejandro Pérez Abraham
RNPS No. 2143
ISSN 2072-6260
Indicaciones para los Autores:
Seguir la plantilla que aparece en www.cenatav.co.cu
C E N A T A V
7ma. A No. 21406 e/214 y 216, Rpto. Siboney, Playa;
La Habana. Cuba. C.P. 12200
Impreso en Cuba