CAPÍTULO I

Aplicación de la teoría de grafos para el tratamiento de complejidades de los datos

A. Guzmán-Ponce, Rosa María Valdovinos Rosas, J. R. Marcial-Romero, J. S. Sánchez y Héctor Miguel Montenegro Monroy

Introducción

En la actualidad, la extracción de conocimiento a partir de conjuntos de datos ha adquirido más valor para las empresas y se está convirtiendo en un activo en la toma de decisiones; no obstante, los conjuntos de datos generados por sensores, sistemas de almacenamiento, interacción en redes sociales u otros medios se encuentran afectados por diversos factores que disminuyen el rendimiento en modelos de aprendizaje [1]. Estos factores adversos comúnmente son denominados complejidades de los datos, entre los principales se encuentran [2], [4]:

- **Desbalance de clases.** Se presenta cuando la distribución de las instancias por clase no es balanceada. Es decir, una o más de las clases tienen un número de instancias notablemente mayor en comparación con el resto de las clases. Un ejemplo de esto se encuentra en contextos financieros, donde el número de transacciones fraudulentas en esquemas de detección de fraude es menos representativo, en comparación con el número total de solicitudes.

- **Solapamiento de clases.** Es cuando existen instancias de diferentes clases que se entrecruzan, es decir, instancias con atributos poco discriminantes. El problema de reconocimiento de personas sanas y enfermas en el área médica es un claro ejemplo ya que si las características que definen una enfermedad no

son plenamente diferenciadas se pueden registrar casos tanto de personas sanas como enfermas, pero que comparten las mismas características.

- **Patrones atípicos o ruido.** Una instancia se denomina atípica cuando, al tener bien definida la clase a la que pertenece, ésta difiere significativamente del resto de instancias que pertenecen a la misma clase. Un ejemplo de ello ocurre cuando una persona joven se diferencia del resto de su grupo de edad al presentar incontinencia urinaria, que suele manifestarse en etapas avanzadas de la vida.

  Por otro lado, una instancia considerada como ruido es una instancia mal etiquetada, que ha sido asignada a una clase equivocada debido a su similitud con instancias de otra clase. Un ejemplo médico es cuando personas diagnosticadas con neumonía atípica son identificadas como COVID-19, debido a la similitud en sus signos y síntomas.

- **Alta dimensionalidad.** Hace referencia al elevado número de características que se requieren para describir un patrón, de tal forma que en algunos casos puede ser mayor que el número de instancias que integran el conjunto de datos. Este problema es común en las micro matrices que miden la expresión genética, donde se pueden tener decenas de cientos de instancias, cada una de las instancias pueden tener docenas de miles de genes (características).

Todos estos factores negativos han sido tema de estudio para el área de minería de datos y reconocimiento de patrones, debido al impacto negativo que tienen en modelos de aprendizaje, logrando que las tasas de precisión disminuyan. La complejidad de mayor impacto y presencia en problemas del mundo real es el desbalance de clases, el cual se puede encontrar en infinidad de ámbitos de la vida cotidiana [1].

Por otro lado, la teoría de grafos ha sido ampliamente estudiada en matemáticas y comúnmente usada en diversas áreas del

conocimiento, tales como la biología, química, redes de comunicación, entre otros [4]. En los últimos años se ha vuelto popular en áreas de inteligencia artificial por la capacidad de representar problemas complejos en términos de vértices y aristas.

La idea de utilizar una estructura de grafos para representar el conjunto de datos o un subconjunto de éste, al mismo tiempo que se tratan algunas de las complejidades existentes en él, es lo que motiva la realización de este estudio. Para este fin se presenta un estudio empírico de propuestas basadas en grafos para solventar algunas de las complejidades existentes en los datos.

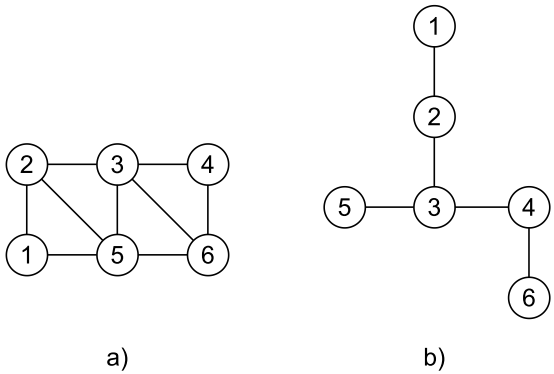
Teoría de grafos

Un *grafo* es una estructura formada por elementos denominados *vértices* y por una relación entre ellos, denominada *aristas*[1] [5]. Un *grafo completo* es un grafo en el cual cualquier par de vértices está conectado por una arista. El *vecindario* de un vértice v en un grafo es el conjunto de vértices adyacentes a v, es decir comparten arista con v. En este sentido, *subgrafo* de un grafo G es un grafo cuyos conjuntos de vértices y aristas son subconjuntos de los de G; se dice que es *inducido* cuando contiene un subconjunto de vértices Y del grafo original y cuyas aristas constan de todas las aristas del grafo original, que tienen ambos extremos en Y.

Un tipo de estructura que se utiliza comúnmente en grafos es el árbol, que se define como un grafo sin ciclos, lo que significa que hay un único camino que une cualquier par de vértices. Por otro lado, un árbol de expansión es aquel que incluye todos los vértices del grafo original, pero no incluye aristas que formen ciclos (Fig. 1b muestra un posible árbol de expansión del grafo del grafo a su izquierda) [5].

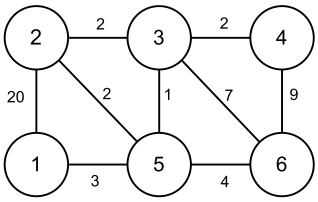
[1] Los vértices son nodos en una estructura de grafo, mientras que las aristas son las líneas que conectan los vértices, éstas representan una relación entre ellos.

Fig. 1. Ejemplo de grafos. a) Grafo simple; b) Árbol.



Un *grafo ponderado* G_w es un grafo donde cada arista tiene asociado un valor real, denominado *peso*. Una manera computacional de representar un grafo es por medio de una *matriz de adyacencia* (matriz bidimensional), donde cada una de las filas y columnas representa un vértice en el grafo: el valor que se almacena en la fila u y columna v indican si hay una arista entre los vértices u y v. Para el caso de un grafo ponderado, el valor asociado en la matriz será el peso de la arista, la Fig. 2 ilustra la matriz adyacencia del grafo ponderado de su izquierda.

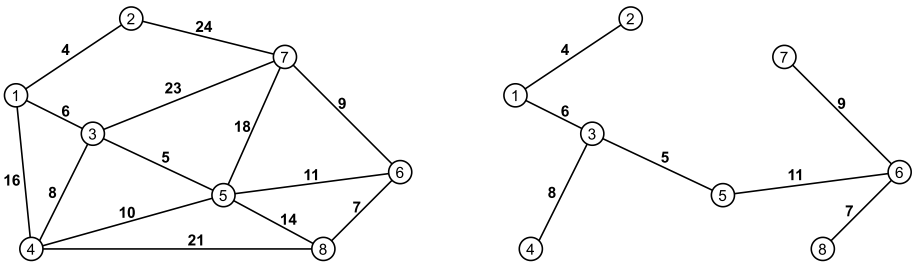
Fig. 2. Ejemplo de matriz de adyacencia.



	1	2	3	4	5	6
1	0	20	0	0	3	0
2	20	0	2	0	2	0
3	0	2	0	2	1	7
4	0	0	2	0	0	9
5	3	2	1	0	0	4
6	0	0	4	9	4	0

Por último, un *Árbol Mínimo de Expansión* de un grafo G_w es un subgrafo formado por un subconjunto de aristas de G_w, que conecta a todos los vértices, sin ciclos con la condición de tener el mínimo peso total de aristas (Fig. 3).

Fig. 3. Grafos ponderados.
a) Grafo ponderado; b) Árbol de Expansión Mínimo.



Tratamiento de complejidades de los datos

Existen dos categorías principales en el preprocesado de datos: preparación y reducción [6]. En la preparación, el conjunto de datos es ajustado para que algún modelo de aprendizaje lo use, en específico, los algoritmos de limpieza identifican datos redundantes, como ruido o solapamiento y se busca repararlos. En tanto que, en la reducción se busca obtener una representación reducida de los mismos sin comprometer la integridad del conjunto original. En específico, para el tratamiento de desbalance de clases, se utilizan técnicas de remuestreo clasificadas en tres categorías [2], [6]:

- **Bajo-muestreo:** Consiste en eliminar instancias, usualmente de la clase mayoritaria, con el fin de reducir el tamaño del conjunto de datos.

- Sobre-muestreo: Implica la creación o replicación de instancias comúnmente de la clase minoritaria.

- Métodos híbridos: Consiste en aplicar tanto técnicas de bajo-muestreo, como técnicas de sobre-muestreo.

La generación de grandes volúmenes de información, a menudo, implica limitaciones de recursos para la clasificación, así como en la necesidad de transformar los conjuntos de datos en formatos adecuados para poder extraer valor de ellos. Por consiguiente, en este estudio nos enfocaremos en los métodos de bajo-muestreo.

Uno de los métodos más usado es el bajo-muestreo aleatorio, RUS por sus siglas en inglés (*Random under-sampling*) [1]; este método equilibra el conjunto de datos mediante la eliminación aleatoria de instancias que pertenecen a la clase mayoritaria. Una limitante de este método es la posibilidad de eliminar información relevante por no tener un mecanismo de control en la eliminación.

Otros métodos son los basados en el vecindario de instancias, los cuales en su mayoría toman los k vecinos más cercanos. Algunos algoritmos de este tipo son el condensado de Hart (CNN) [7], el cual elimina instancias que están lo suficientemente lejos de la frontera de decisión y los enlaces de Tomek (TL) [8], el cual elimina todas aquellas instancias que formen un enlace denominado Tomek, por considerarlas ruidosas al estar la frontera de decisión.

Otras técnicas son los denominados ensembles, los cuales realizan la combinación de un conjunto de clasificadores con alguna técnica de remuestreo, para mejorar el rendimiento de los clasificadores. Un ejemplo es el algoritmo RUSBoost (RBt) [9], el cual combina RUS con un conjunto de clasificadores que buscan reducir el sesgo por la clase mayormente representada. Otro ejemplo es la utilización de filtro de ruido (EEKF) [10], en el que se filtra la clase minoritaria eliminando instancias consideradas como ruido, cuyos vecinos pertenecen a la clase mayoritaria. Después, se entrenan varios modelos de aprendizaje con los diferentes subconjuntos creados. Al final se fusionan las mejores instancias de cada modelo de aprendizaje.

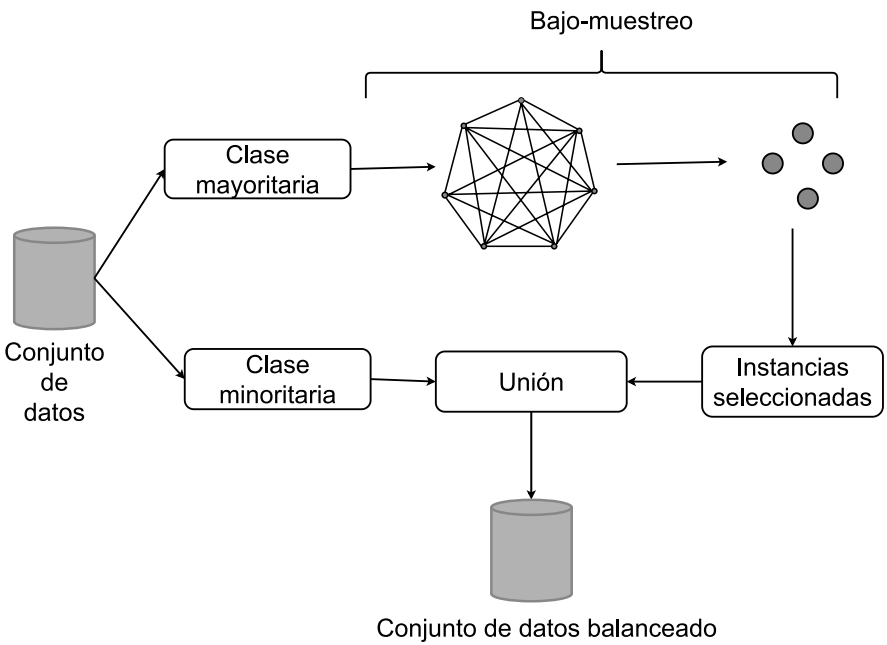
Otras técnicas son basadas en agrupamiento, un ejemplo es el método *Clustering-based undersampling* (CBU) [11], que genera g centros para representar la clase mayoritaria mediante el algoritmo K-Means.

Metodología

Algoritmos basados en grafos

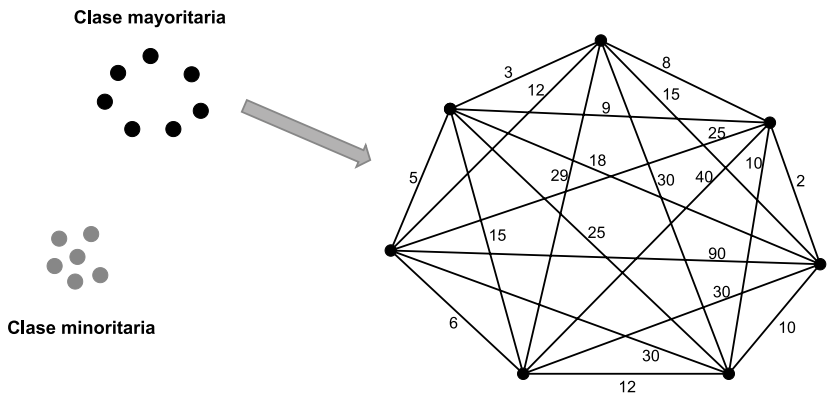
Los algoritmos basados en grafos para abordar el desbalance de clases se fundamentan principalmente en la obtención del contorno de la clase mayoritaria o del núcleo de sus instancias [12]. La idea general se muestra en la Fig. 4.

Fig. 4. Flujo de trabajo de algoritmos basados en grafos.



El proceso de la Fig. 4 inicia con la representación de la clase mayoritaria como un grafo completo ponderado. Esta representación se muestra en la Fig. 5, en la que las instancias de la clase mayoritaria se visualizan como los vértices del grafo y la unión entre cada par de instancias y tienen asignado un valor o peso correspondiente a la distancia euclídea existente entre un vértice y otro, es decir, entre dos instancias.

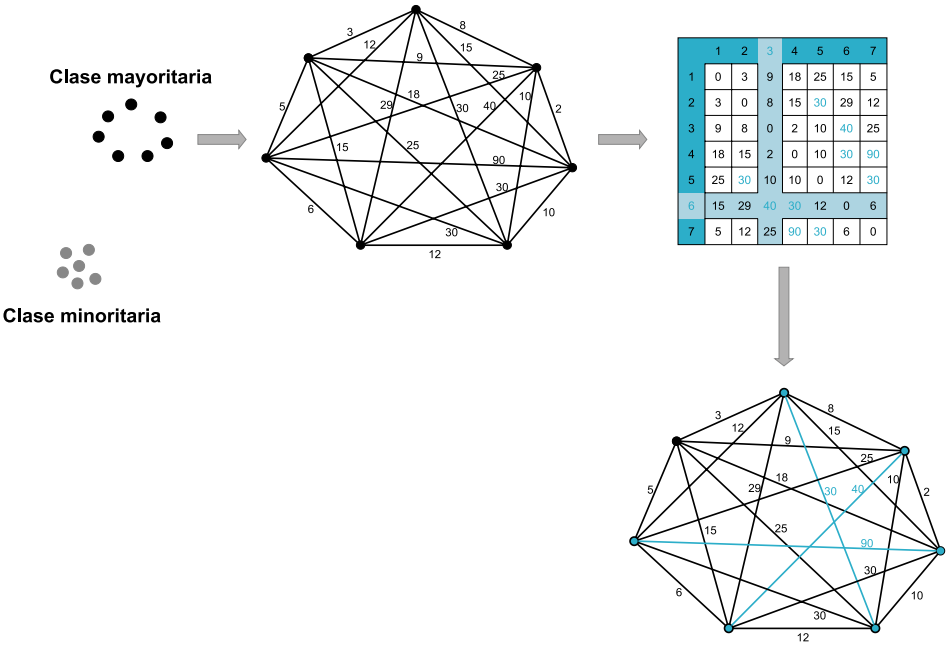
Fig. 5. Representación de la clase mayoritaria en un grafo.



Una vez construido el grafo completo ponderado de la clase mayoritaria es posible aplicar alguna de las siguientes estrategias [12]:

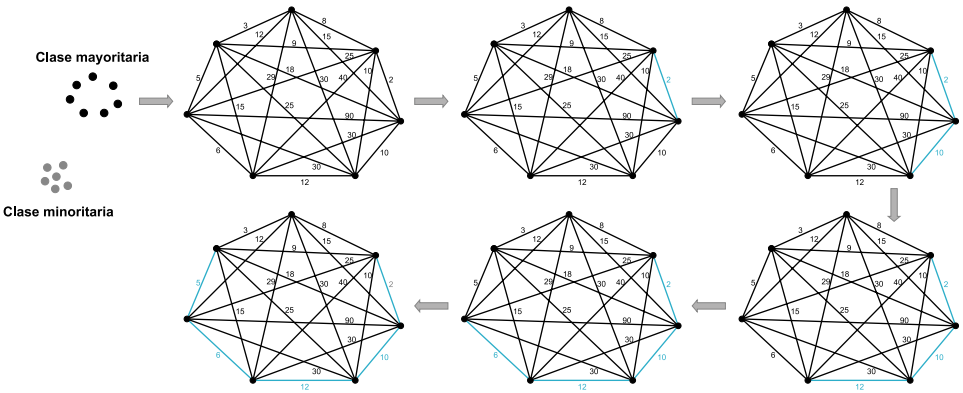
- **Subgrafo inducido (IG-US):** se buscan mantener todas aquellas instancias que están más alejadas unas de las otras, por medio de la matriz de adyacencia del grafo (Fig. 6). Por ejemplo, la fila y columna marcada de naranja hacen referencia a que el par de vértices {3,6} será considerado en el subgrafo inducido, ya que tiene una de las distancias más lejanas entre ellos.

Fig. 6. Construcción de un subgrafo inducido a partir de la clase mayoritaria.



- Árbol de expansión mínimo (MIST-US) [12]: su objetivo es obtener la representación del núcleo de clase mayoritaria descartando todas aquellas instancias que están lo suficientemente cerca de la frontera de decisión (Fig. 7). Para la construcción del árbol, el primer paso es considerar un vértice pivote, el siguiente vértice será aquel cuya arista que comparta con el vértice pivote tenga el menor costo, siempre y cuando no forme un ciclo. Para que éste último sea ahora un vértice pivote, el proceso se repite hasta haber visitado todos los vértices.

Fig. 7. Construcción de un árbol de expansión mínimo a partir de la clase mayoritaria.



Para ambas propuestas, una vez construido el grafo, se toman un número representativo de instancias, considerando la proporción deseada de balance a obtener.

Análisis experimental

Como se mencionó, comúnmente los conjuntos de datos tienen algún problema que deteriora el desempeño de los modelos de clasificación, para evitarlo se han desarrollado diversas estrategias, entre las cuales la teoría de grafo comienza a mostrar resultados prometedores.

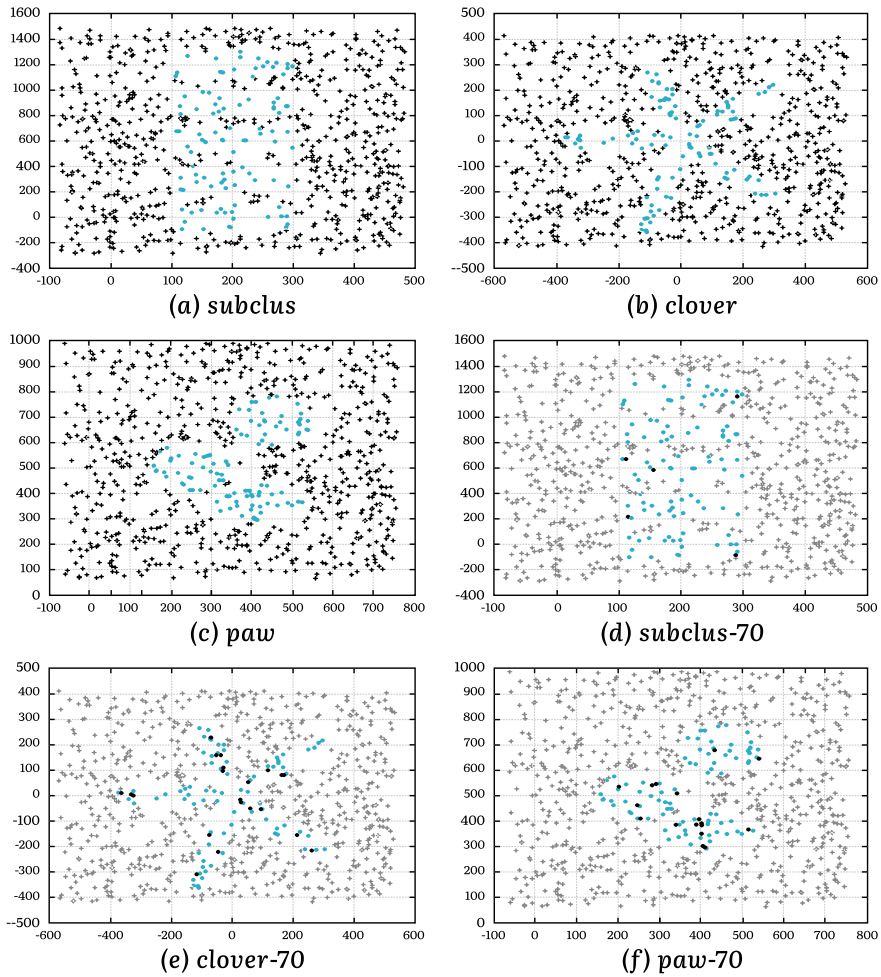
En esta sección se presenta un análisis experimental sobre conjuntos de datos sintéticos en los que se busca analizar los efectos de técnicas de bajo-muestreo en un ambiente controlado.

Conjuntos de datos

Para el análisis presentado en esta sección se utilizaron 6 conjuntos de datos sintéticos denominados *subclus*, *clover* y *paw* obtenidos de *Imbalanced data sets* [14]. Cada conjunto de datos está formado por

800 instancias, un grado de desbalance de 7; 3 de ellos sin ruido y los 3 restantes con un 70% de ruido (ver Fig. 8).

Fig. 8. Distribución de los conjuntos de datos sintéticos con ausencia de ruido y con presencia del 70% de ruido [13].



El ruido hace referencia a instancias mal etiquetadas o a las que se les cambia el valor de una característica sin alterar su etiqueta de clase.

En todos los conjuntos de datos, las instancias de clase minoritaria (puntos azules) están uniformemente rodeados por instancias de clase mayoritaria (puntos negros).

Para el conjunto de datos *subclus* (Fig. 8a) las instancias pertenecientes a la clase minoritaria forman rectángulos de manera disjunta. Mientras que en el conjunto *clover* (Fig. 8b), las instancias de clase minoritaria asemejan una flor con pétalos elípticos. Por último, en el conjunto de datos *paw* (Fig. 8c) la clase minoritaria se ubica en tres subregiones, de las cuales dos están ubicadas cerca una de la otra y una más pequeña está separada. Este último conjunto de datos puede representar de mejor manera datos de algún problema de la vida real, mientras que los conjuntos de datos *subclus* y *clover* constituyen formas más complejas de aprender.

El porcentaje de ruido en cada conjunto de datos está distribuido sobre ambas clases, en las Fig. 8(d), 8(e) y 8(f) la presencia de ruido se ilustra en color gris.

Medidas de evaluación

En los experimentos se utilizaron dos de los clasificadores más populares en aprendizaje automático: regla del vecino más cercano (1NN) y árbol de decisión (J48), ambos con los parámetros predeterminados por el *software* de código abierto WEKA [15].

Las medidas de evaluación se obtienen de la matriz de confusión, la cual permite analizar por separado la tasa de aciertos de la clase positiva o Verdaderos Positivos (VP) y los aciertos de la clase negativa o Verdaderos Negativos (VN). Con estos datos es posible obtener la media geométrica, medida utilizada en escenarios de desbalance de clase lo cual se puede calcular de la siguiente manera (Ec. 1):

$$\text{Gmean} = \sqrt{\frac{VP}{(VP+FN)} \cdot \frac{VN}{(VN+FP)}} \tag{1}$$

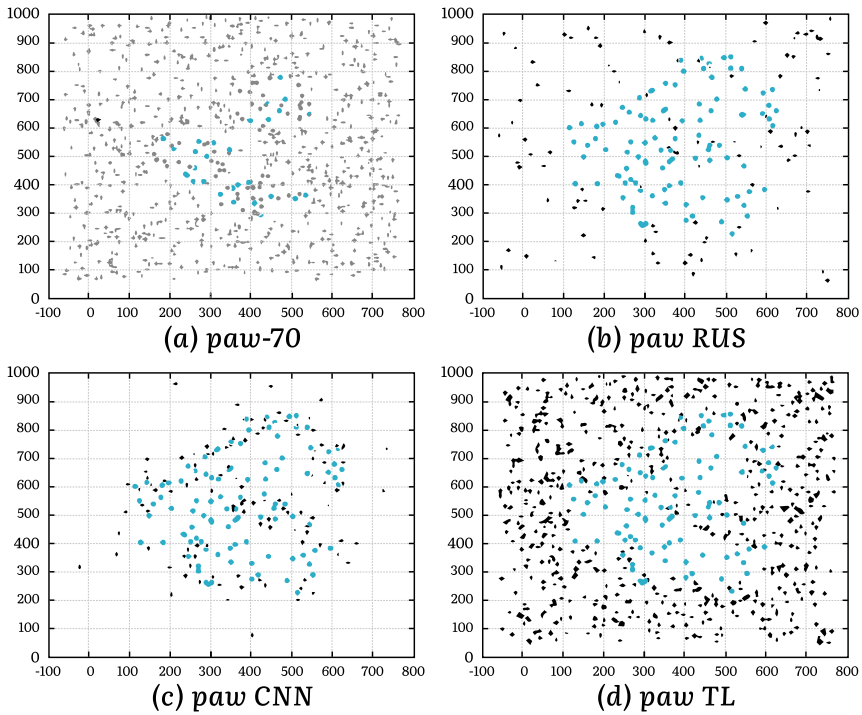
Para determinar las diferencias estadísticamente significativas entre más de dos métodos se aplica la prueba de Friedman, la cual asigna

valor de 1 al mejor resultado, en sucesión al resto de los métodos. En caso de tener empates se asigna un rango promedio entre los métodos empatados.

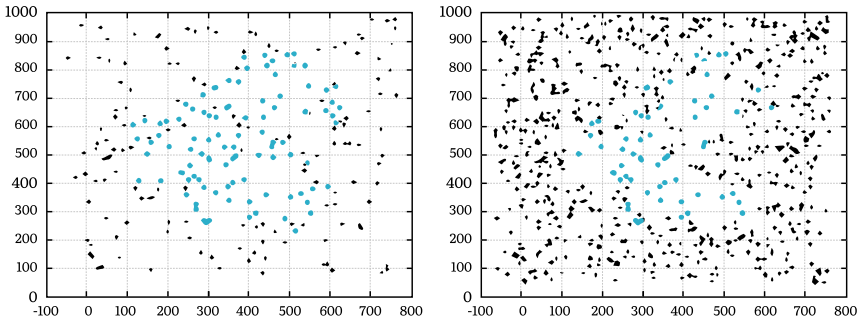
Resultados y discusión

Una vez preprocesados los conjuntos de datos por 8 técnicas de bajo-muestreo[2], en la Fig. 9 se presenta la dispersión final de los conjuntos de datos *paw* con 70% de ruido.

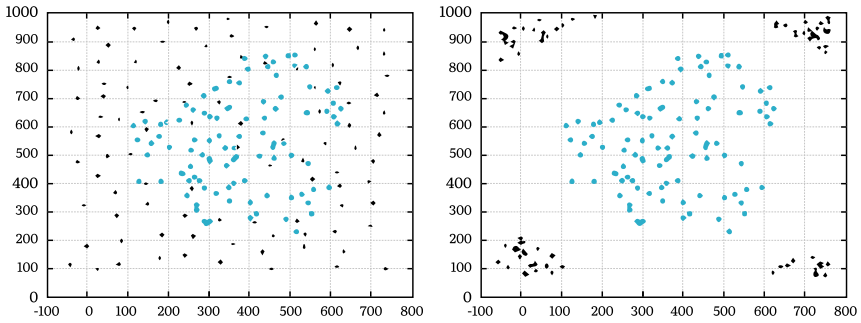
Fig. 9. Distribución de los conjuntos de datos sintéticos con 70% de ruido posterior a aplicar estrategias basadas en grafos [13].



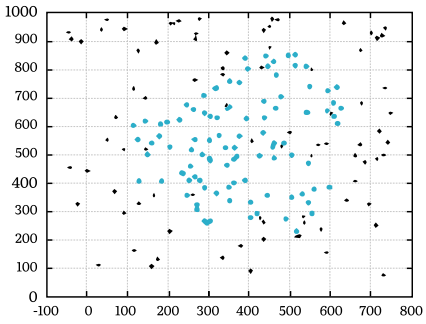
[2] Randon Under-sampling (RUS), Condensado de Hart (CNN), Enlaces de Tomek (TL), RUSBoost (RBT), Filtro de ruido (EEKF), Clustering-based undersampling (CBU), Subgrafo inducido (IG-US) y Árbol de expansión mínimo (MIST-US).



(e) paw RBT (f) paw EEKF



(g) paw CBU (h) paw IG-US



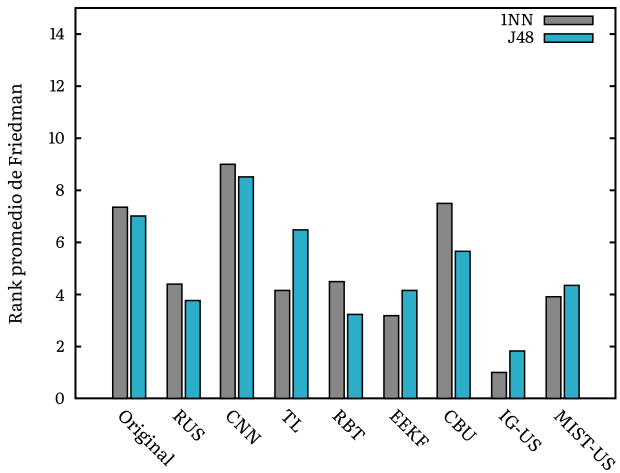
(i) paw MIST-US

Como se puede observar en la Fig. 9, el comportamiento del método CNN hace que la clase mayoritaria aparentemente sea más pequeña que la clase minoritaria. En tanto que el método TL y EEKF muestran un deficiente balance de clases, ya que en ambos casos el tamaño de

la clase mayoritaria parece conservarse. Los métodos que claramente balancean el conjunto de datos son RUS, RBT, CBU y los métodos basados en grafos (IG-US y MIST-US).

Particularmente, al obtener el IG-US la dispersión de la clase mayoritaria se aleja completamente de la clase minoritaria, dado que se toman en consideración las instancias que están lo más alejadas unas de las otras, contrario al resultado obtenido con MIST-US, en el que las instancias se mantienen cerca de la frontera de decisión. La Fig. 10 muestra el análisis estadístico de los resultados de la clasificación. Se presentan los promedios obtenidos del rango de Friedman por cada método de bajo-muestreo utilizado.

Fig. 10. Rango promedio de Friedman para los métodos de bajo-muestreo.



En la Fig. 10 el método IG-US presentó mejores resultados para ambos clasificadores, lo que sugiere que al conservar instancias alejadas entre sí se logra una representación adecuada de la clase mayoritaria. No obstante, se puede apreciar que el método RBT obtiene mejor rendimiento para el clasificador J48 en comparación de RUS, esto sugiere que incorporar un sistema de múltiples clasificadores al balanceo aleatorio mejora los resultados. Por último, el comportamiento de

CNN es el más deficiente frente al resto, esto sucede por la enorme cantidad de instancias extraídas.

Conclusiones

Desafortunadamente es inevitable la presencia de factores negativos en conjuntos de datos. Por lo tanto, se necesitan técnicas para hacer frente a estos factores, tales como la limpieza de datos, la imputación de valores faltantes, la normalización y estandarización, y el remuestreo.

Actualmente, la teoría de grafos se convierte en una potencial área de uso en técnicas de minería de datos, ya que los problemas pueden esquematizarse y tratarse como un grafo.

En este capítulo se ha presentado un estudio exhaustivo que contempla 8 métodos ampliamente usados en la literatura para el tratamiento del desbalance, incluidas dos técnicas basadas en grafos.

Puntualmente, del estado del arte se observa que las técnicas basadas en vecindario dependen de un número k *a priori* de vecinos, con la desventaja de no establecer un valor general para todos los conjuntos de datos, mientras que las técnicas que combinan métodos *ensemble* o *clustering*, mantienen una dependencia en establecer en principio un número de grupos o vecinos, que a su vez no es el mismo para todos los conjuntos. Mientras que las técnicas basadas en grafos dependen únicamente de la dispersión de las muestras, adicionalmente, son técnicas deterministas, lo que permite obtener invariablemente resultados consistentes bajo condiciones iniciales similares.

Los resultados obtenidos permiten observar que los métodos basados en grafos obtienen conjuntos de datos reducidos sin pérdida de información útil con un mejor comportamiento en términos de media geométrica, en comparación de otros métodos. Por lo que el uso de teoría de grafos es prometedora al esquematizar el conjunto de datos en un grafo completo y permite mantener información de todo el conjunto de datos. Adicionalmente, la reducción del tamaño del conjunto sin pérdida de información es ideal frente a los nuevos retos de grandes volúmenes de datos.

Las líneas abiertas de estudio apuntan a probar las técnicas basadas en grafos en modelos de aprendizaje no supervisado, así como la necesidad de trasladar la concepción en problemas con más de dos clases. Por último, estudiar el comportamiento de algoritmos basados en grafos para grandes volúmenes de datos.

Referencias

[1] A. Guzmán-Ponce, R. M. Valdovinos, J. S. Sánchez, y J. R. Marcial-Romero, "A new under-sampling method to face class overlap and imbalance", Applied Sciences (Switzerland), vol. 10, núm. 15, 2020, doi: 10.3390/app10155164.

[2] L. E. B. Ferreira, J. P. Barddal, F. Enembreck, y H. M. Gomes, "An Experimental Perspective on Sampling Methods for Imbalanced Learning from Financial Databases", Proceedings of the International Joint Conference on Neural Networks, Institute of Electrical and Electronics Engineers Inc., 2018, doi: 10.1109/IJCNN.2018.8489290.

[3] Á. Arnaiz-González, A. González-Rogel, J. F. Díez-Pastor, y C. López-Nozal, "MR-DIS: democratic instance selection for big data by MapReduce", Progress in Artificial Intelligence, vol. 6, núm. 3, pp. 211–219, 2017, doi: 10.1007/s13748-017-0117-5.

[4] T. Washio y H. Motoda, "State of the art of graph-based data mining", ACM SIGKDD Explorations Newsletter, vol. 5, núm. 1, pp. 59–68, 2003, doi: 10.1145/959242.959249.

[5] J. L. Gross y J. Yellen, "Graph theory and its applications, second edition", Graph Theory and Its Applications, Second Edition, pp. 1–800, 2005, doi: 10.1201/9781420057140/GRAPH-THEORY-APPLICATIONS-JAY-YELLEN-JONATHAN-GROSS.

[6] S. García, J. Luengo, y F. Herrera, "Intelligent Systems Reference Library 72 Data Preprocessing in Data Mining", Cham, Switzerland: Springer International Publishing, 2015, http://www.springer.com/series/8578.

[7] P. E. Hart, "The condensed nearest neighbor rule. IEEE Transactions on Information Theory", Scientific Research Publishing, vol. 14, pp.

515-516. 1968, https://scirp.org/reference/referencespapers.aspx?referenceid=29777 (consultado el 27 de marzo de 2023).
[8] I. Tomek, "Two Modifications of CNN", IEEE Trans Syst Man Cybern, vol. SMC-6, núm. 11, pp. 769–772, 1976, doi: 10.1109/TSMC.1976.4309452.
[9] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, y A. Napolitano, "RUSBoost: A Hybrid Approach to Alleviating Class Imbalance", Systems and Humans, vol. 40, núm. 1, 2010, doi: 10.1109/TSMCA.2009.2029559.
[10] Q. Kang, X. S. Chen, S. S. Li, y M. C. Zhou, "A Noise-Filtered Under-Sampling Scheme for Imbalanced Classification", IEEE Trans Cybern, vol. 47, núm. 12, pp. 4263–4274, 2017, doi: 10.1109/TCYB.2016.2606104.
[11] W. C. Lin, C. F. Tsai, Y. H. Hu, y J. S. Jhang, "Clustering-based undersampling in class-imbalanced data", Inf. Sci., vol. 409–410, pp. 17–26, 2016, doi: 10.1016/J.INS.2017.05.008.
[12] A. Guzmán-Ponce, J. R. Marcial-Romero, R. M. Valdovinos-Rosas, y J. S. Sánchez-Garreta, "Weighted Complete Graphs for Condensing Data", Electronic Notes Theoretical Computer Science, vol. 354, pp. 45–60, 2020, doi: 10.1016/J.ENTCS.2020.10.005.
[13] A. Guzmán-Ponce, "Nuevos algoritmos basados en grafos y clustering para el tratamiento de complejidades de los datos", Tesis Doctoral, pp. 1–171, 2021.
[14] "KEEL: A software tool to assess evolutionary algorithms for Data Mining problems (regression, classification, clustering, pattern mining and so on)". https://sci2s.ugr.es/keel/datasets.php.
[15] C. Pal, I. Witten, E. Frank, y M. Hall, "Weka 3 - Data Mining with Open-Source Machine Learning Software in Java", 2016, https://www.cs.waikato.ac.nz/ml/weka/index.html.