

UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS

FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN



Trabajo para optar por el Título Académico de Máster en Ciencia de la  
Computación

# SELECCIÓN DE INSTANCIAS PARA EL CLASIFICADOR ASOCIATIVO GAMMA USANDO LA TEORÍA EXTENDIDA DE LOS CONJUNTOS APROXIMADOS

**Autor:**

Lic. Jarvin Alberto Antón Vargas

**Tutores:**

DRA. YENNY VILLUENDAS REY

DR. ITZAMÁ LÓPEZ YÁÑEZ

2016

---

El que suscribe, Jarvin Alberto Antón Vargas, hace constar que el trabajo titulado "Selección de instancias para el clasificador asociativo Gamma usando la teoría extendida de los Conjuntos Aproximados" se realizó en la Universidad Central "Martha Abreu" de Las Villas como parte de la culminación de los estudios de Maestría en Ciencia de la Computación, autorizando que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

---

Firma del Autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y que el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura a la temática señalada.

---

Firma del Tutor

Dra. Yenny Villuendas Rey

---

Firma del Jefe de Seminario

Dr. Carlos Morell Pérez

---

# Dedicatoria

Dedico esta investigación a mis padres y mis abuelos, a mi tía Mary y a mi padrastro Heriberto por brindarme todas las comodidades indispensables, en lo espiritual y en lo material para que pudiera llevar a cabo esta investigación.

---

# Agradecimientos

Agradezco enormemente el apoyo y guía que me han brindado mis tutores en especial a la Dra. Yenny Villuendas.

Agradezco a mis padres, por sus consejos y su constante apoyo en todas las esferas, lo que me ha servido para levantarme cada día con la seguridad en el resultado que hoy se muestra en esta investigación.

A mi tía por creer siempre en mi capacidad y por su apoyo incondicional.

A a mis abuelos por su ejemplo invaluable de constancia, sacrificio y perseverancia.

A a mis amigos por soportarme en los momentos de mal y buen humor durante el desarrollo de esta investigación, a mis compañeros de trabajo, estudio e investigación por su aporte desinteresado.

A todas aquellas personas que me han brindado su amor, cariño y afecto durante este período de estudio, superación, constancia; que de un modo u otro han hecho más placentero el recorrido hasta el momento en el que nos encontramos hoy.

A todos, mi más profundo agradecimiento.

---

# Resumen

El reconocimiento de patrones constituye una ciencia multidisciplinaria que se ha caracterizado por un continuo desarrollo en disímiles aristas. El enfoque asociativo en los últimos años se ha caracterizado por crear métodos de una marcada sencillez y eficacia, comparable en la mayoría de los casos con muchos de los modelos desarrollados siguiendo otros enfoques, aunque no está exento de sufrir las mismas deficiencias que estos otros métodos.

En el presente trabajo se realiza una propuesta de selección de instancias para el clasificador asociativo Gamma desarrollado por López Yáñez en el 2007, en el Centro de Investigación en Computación del Instituto Politécnico Nacional de México, basado en la teoría extendida de los Conjuntos Aproximados (*Rough Sets*), tomando como base elementos del propio clasificador, logrando eliminar las instancias que pueden afectar la precisión del mismo. De esta forma se propone el uso de la teoría extendida de los Conjuntos Aproximados, lo que constituye la base principal del método de selección de instancias GRIS (*Gamma Rough based Instance Selection*).

Para la validación del nuevo enfoque, se tomaron como medidas, la exactitud lograda por el clasificador asociativo Gamma sobre bases de datos balanceadas, el promedio de la exactitud de la clasificación por clases sobre bases de datos desbalanceadas y el índice de retención de objetos, y se evalúa con otros modelos de selección de instancias como el ENN, GGE, RNGE, MSEditB, basados en el error de clasificación, por lo que se consideran como de propósito similar. La evaluación se realizó sobre un conjunto de bases de datos internacionales. De esta forma los resultados del nuevo método presentaron similar comportamiento con respecto a los otros modelos con respecto a la precisión de la clasificación. De igual forma se obtuvo una mayor reducción de la cantidad de instancias en la mayoría de las bases de datos.

---

# Abstract

Pattern recognition is a multidisciplinary science that has been characterized by a continuous development in dissimilar edges. The associative approach in recent years has been characterized by creating methods with a marked simplicity and efficiency, comparable in most cases with many of the models developed in accordance with other approaches, although not exempt from suffering the same deficiencies as these other methods. In this paper a proposal for instance selection for associative Gamma classifier developed by López Yáñez in 2007, at the Center for Computing Research, National Polytechnic Institute of Mexico, based on the extended theory of rough sets is made, using elements of the classifier itself, achieving eliminate instances that may affect the accuracy of the same. Thus the use of the extended theory of rough sets is proposed to be the main basis of the instance selection method GRIS (Gamma Rough based Instance Selection). To validate the new approach were taken as measures, the accuracy achieved by the associative classifier Gamma, the average of the accuracy and Objects Retention Index, evaluated with other models from the state of the art such as the ENN, GGE, RNGE, MSEditB, all these methods based at classification error, which are considered of similar purpose. The evaluation was conducted on a set of international databases. Thus the results of the new method presented similar behavior with respect to the other models regarding the classification accuracy. By other hand the reduction of the number of instances obtained was superior in most of the databases.

---

# Índice general

Dedicatoria	I
Agradecimientos	II
Resumen	III
Abstract	IV
Introducción	1
<b>1. Selección y construcción de objetos para el mejoramiento de clasificadores supervisados</b>	<b>6</b>
1.1. Clasificación supervisada . . . . .	7
1.1.1. Clasificadores basados en reglas . . . . .	7
1.1.2. Árboles de decisión . . . . .	9
1.1.3. Redes Neuronales Artificiales (RNA) . . . . .	10
1.1.4. Máquinas de Vectores de Soporte ( <i>Support Vector Machines-SVMs</i> ) . . . . .	12
1.1.5. Vecino más cercano . . . . .	13
1.1.6. Algoritmos de tipo pivote . . . . .	15
1.1.7. Clasificadores asociativos . . . . .	16
1.1.8. Clasificador Asociativo Gamma . . . . .	18
1.2. Pre-procesamiento de datos . . . . .	20
1.2.1. Selección de rasgos . . . . .	21
1.2.2. Construcción de objetos . . . . .	24

---

1.2.3. Selección de objetos . . . . .	26
1.3. Conclusiones del capítulo . . . . .	32
<b>2. Selección de objetos para el clasificador Gamma</b>	<b>34</b>
2.1. Propuesta de los conjuntos aproximados basados en la Gamma similitud	35
2.2. Funcionamiento del Clasificador Asociativo Gamma . . . . .	35
2.3. Conjuntos Aproximados basados en la Gamma similitud . . . . .	39
2.4. Algoritmo para la selección de instancias . . . . .	48
2.5. Análisis del costo computacional del algoritmo propuesto . . . . .	49
2.6. Conclusiones del capítulo . . . . .	51
<b>3. Resultados experimentales</b>	<b>53</b>
3.1. Descripción de los bancos de datos de prueba . . . . .	53
3.2. División de los conjuntos de datos . . . . .	54
3.3. Algoritmos utilizados . . . . .	56
3.4. Medidas de desempeño y test estadísticos . . . . .	58
3.4.1. Medidas de desempeño utilizadas . . . . .	58
3.4.2. Test estadísticos . . . . .	59
3.5. Resultados considerando clasificador Gamma con rasgos igualmente ponderados . . . . .	61
3.6. Resultados considerando clasificador Gamma con pesos obtenidos de forma automática . . . . .	66
3.7. Resultados considerando otros clasificadores de aprendizaje perezoso .	71
3.8. Comparación del los resultados con el modelo Edit3RS . . . . .	78
3.9. Conclusiones del capítulo . . . . .	81
<b>Conclusiones</b>	<b>82</b>
<b>Recomendaciones</b>	<b>83</b>
<b>Bibliografía</b>	<b>84</b>

---

# Introducción

En las últimas décadas se ha visto un incremento en la acumulación de información sobre los más disímiles temas. Todo esto es producto de la informatización en los distintos sectores de la sociedad. Pero la información no lo es todo, lo importante es el conocimiento que se puede obtener de la información recopilada.

El conocimiento que viene intrínseco en el conjunto de datos almacenados, que de una forma u otra describen procesos ingenieriles, matemáticos, computacionales, sociales y humanísticos, que pueden estar relacionados con objetos físicos y abstractos, es analizado por una ciencia con carácter multidisciplinario, denominada Reconocimiento de Patrones (RP) [1].

Es posible definirla también, como la forma intuitiva en la que se interpreta el mundo que nos rodea, es el mecanismo que permite distinguir elementos que pueden ser distintos entre sí o relacionados, describir objetos, tomar decisiones e incluso darles explicación a estas decisiones. En fin, es un mecanismo para formalizar y automatizar los procesos de interpretación del mundo real [2].

Debido a la vital importancia que adquieren las tareas de reconocimiento de patrones realizadas cotidianamente, ha impulsado a que en un grupo cada vez más amplio de investigadores, se encaminen al diseño y creación de sistemas de reconocimiento automático de patrones, sobre todo con el advenimiento de los sistemas computacionales modernos [3].

Para poder llevar a cabo este proceso es necesario pasar por un conjunto de fases que involucran elementos de medición, validación, procesamiento y la interpretación de la información en el entorno de estudio, he aquí el carácter multidisciplinario de esta ciencia [4].

Es evidente que una de las principales tareas del Reconocimiento de Patrones es

la clasificación, la regresión o la descripción de una solución [3], dado a que es el momento del cual se pueden obtener importantes conclusiones sobre una investigación determinada.

En la literatura se describen dos ramas específicas de clasificación: la clasificación no supervisada y la supervisada [3][5], esta última se entiende como la estrategia que intenta encontrar una hipótesis que corresponda con el espacio a interpretar, donde se tiene un previo conocimiento de las clases a las que pertenece cada uno de los objetos que conforman el espacio de interpretación [3].

Dependiendo del tipo de problema a analizar y de la propia aplicación que se haga de los modelos de reconocimiento automático de patrones, pueden obtenerse dos intenciones fundamentales: recuperar o clasificar aquellos patrones presentes en el caso de estudio, pero no siempre es posible aplicar el mismo enfoque para todos los problemas, es por esto que en la literatura [6][7], se pueden encontrar los siguientes:

- Enfoque estadístico-probabilístico
- Enfoque sintáctico-estructural
- Enfoque neuronal
- Enfoque lógico combinatorio
- Enfoque asociativo

Es precisamente el asociativo uno de los enfoques que en los últimos años ha atraído la atención de un grupo de investigadores; este interés surge principalmente por el alto grado de sencillez y eficacia de los algoritmos que se han obtenido, comparables en la mayoría de los casos con muchos de los modelos desarrollados siguiendo otros enfoques [7], lo que se aprecia en la exactitud alcanzada por su aplicación a problemas reales y la simplicidad de los modelos resultantes [8][9][10][11].

El enfoque asociativo surge principalmente de los modelos matemáticos que dieron origen a las memorias asociativas, modelos contemporáneos con las primeras redes neuronales [12]. Como se puede constatar en [13], el estado del arte en las memorias asociativas sirven de base a modelos asociativos de reconocimiento de patrones. Estos modelos están constituidos por las memorias asociativas Alfa-Beta, cuyo modelo se basa en dos operaciones simples, denominadas Alfa y Beta. Estas

operaciones son equiparables en sencillez, a las operaciones básicas de la lógica booleana [14].

Es a partir de este enfoque que surge un nuevo clasificador que introduce una extensión al uso de dichas operaciones básicas y que permite abordar problemas de clasificación supervisada en disímiles dominios. Dicho clasificador al que se hará referencia como clasificador asociativo Gamma, fue desarrollado como tema de tesis de maestría en el Centro de Investigación en Computación del Instituto Politécnico Nacional de México por López-Yáñez en el 2007 [13] y luego ha sido caracterizado su funcionamiento en la tesis de doctorado del mismo autor [15], lo que permite establecer un conjunto de fortalezas y debilidades presentes en el mismo y que se exponen a continuación:

- Un patrón o instancia a clasificar, puede ser clasificado con más de una clase, a partir del conjunto de entrenamiento (conjunto fundamental) sobre el que se trabaje.
- Existencia de más de un patrón dentro del conjunto de entrenamiento con valores muy diferentes que pertenezcan a una misma clase.
- Presencia de alejados de la tendencia central (*outliers*), los cuales pueden provocar una asignación incorrecta al patrón a clasificar.
- Presencia de patrones en el conjunto de entrenamiento que formen parte de la frontera de decisión, que causa indecisión en la clasificación de un patrón.

Si se analiza que los datos constituyen el eslabón principal de todo modelo de reconocimiento de patrones y que la calidad de los mismo influye directamente en los resultados que se obtienen, entonces se puede inferir que un elemento indispensable dentro de todo el proceso es el pre-procesamiento de los datos originales. Este proceso ha sido analizado por diversas investigaciones a lo largo del desarrollo del reconocimiento de patrones [16]. Dicho proceso no busca otro objetivo que tratar de refinar el conjunto de entrenamiento original sin que se pierda precisión en la clasificación de los modelos resultantes.

Otro problema importante dentro de los modelos del reconocimiento de patrones, lo constituye la dimensión de la base de casos de entrenamiento, entiéndase por esto,

la cantidad de rasgos que describen cada una de las instancias y la cantidad de casos que componen la base del conocimiento sobre la que se trabaja. Es característico dentro de la amplia gama de clasificadores, que su funcionamiento sea afectado en gran medida durante la fase de entrenamiento y de clasificación, debido a la gran cantidad de instancias que tiene que procesar. Esta afectación es directamente proporcional al costo computacional del modelo, ya dígase costo de ejecución o costo de almacenamiento o ambos a la vez.

A este problema es susceptible el clasificador asociativo Gamma, es por lo que se plantea como **problema de investigación**: ¿Cómo reducir el costo computacional del clasificador asociativo Gamma, sin que se degrade la precisión de la clasificación?

Se propone como **objetivo general**: proponer un modelo de selección de instancias para el clasificador asociativo gamma, haciendo uso de la teoría de los conjuntos aproximados, tomando elementos del propio clasificador, para reducir el costo computacional del mismo, sin que se degrade la precisión de la clasificación.

Como **Objetivos específicos** de la presente investigación se proponen los siguientes:

1. Analizar la influencia de los principales modelos de pre-procesamiento de datos sobre el clasificador Asociativo Gamma.
2. Definir una relación de similaridad para la teoría de los conjuntos aproximados basada en elementos del clasificador asociativo Gamma.
3. Utilizar un modelo de selección de instancias basado en los conjuntos aproximados con la relación de similaridad propuesta.
4. Evaluar experimentalmente el comportamiento del clasificador Gamma a partir del nuevo modelo de selección diseñado con respecto a otros modelos de selección de instancias.

Por todo lo expuesto anteriormente, se puede formular la siguiente **hipótesis**: si se aplica un modelo de selección de instancias usando la teoría de los conjuntos aproximados, que utilice elementos del clasificador asociativo Gamma, entonces se reduce el costo computacional del clasificador, sin degradar la precisión de la clasificación.

Para darle cumplimiento a los objetivos trazados, la presente tesis se divide en tres capítulos. En el primero se realiza un análisis de las principales características de algunos de los clasificadores más representativos, dentro del reconocimiento de patrones, además de abordar los principales conceptos del pre-procesamiento de datos y algunos de sus modelos más representativos. En el segundo se introducen los principales conceptos de la teoría de los conjuntos aproximados, se define el uso de la teoría extendida de los conjuntos aproximados tomando elementos del clasificador asociativo Gamma. Se implementa un modelo de selección de instancias basado en la teoría de los conjuntos aproximados, propuesto por [17] lo que permite establecer un algoritmo de pre-procesamiento de los datos enfocados en el clasificador en cuestión.

Finalmente en el capítulo 3, se expone el protocolo experimental utilizado, así como las medidas de evaluación y los procedimientos estadísticos que se tuvieron en cuenta para analizar el desempeño del nuevo modelo de selección de instancias y su influencia sobre el Clasificador Asociativo Gamma; en dicho capítulo se discuten también los resultados obtenidos. El documento culmina con las conclusiones, recomendaciones, y la bibliografía consultada.

---

# Capítulo 1

## Selección y construcción de objetos para el mejoramiento de clasificadores supervisados

Dentro de la amplia gama de técnicas del Reconocimiento de Patrones (RP), los clasificadores supervisados se encuentran en un lugar preponderante, son modelos que abordan desde diversas aristas la tarea de asignar una clase a un objeto cuya clasificación es desconocida. De forma general, estos algoritmos se pueden dividir en dos grandes grupos: los clasificadores paramétricos y los no paramétricos. De los primeros es necesario conocer las características que cumplen los datos sobre los que trabaja, mientras que los segundos no necesitan conocimiento previo [18], lo que los hace ganar en usabilidad, por lo que abarca gran parte de la atención de los investigadores. Dentro de este grupo se encuentran diversos modelos tales como Clasificadores Basados en Reglas [19], Árboles de Decisión [20][21], Redes Neuronales Artificiales (RNA) [22][23], Máquinas de Vectores de Soporte (*Support Vector Machine* - SVM) [24][25], la familia de algoritmos de vecino más cercano (*Nearest Neighbor* - NN) [26], el modelo ALVOT [27][28] y variantes del enfoque asociativo como el CHAT [29][30] y el Gamma [6].

En la práctica el conjunto de datos sobre los que trabajan estos modelos, presenta información con disímiles problemas, que van desde datos incompletos, incorrectos, hasta información redundante y sin sentido, pues son elementos que degradan la

clasificación de los mismos; es por esto que se hace necesario en la mayoría de los casos realizar un pre-procesamiento de los datos para refinar y filtrar en mayor o menor medida estos problemas [31]. En el presente capítulo se analizarán inicialmente las principales características de los modelos anteriormente mencionados, posteriormente se definirán algunos de los conceptos que formante del pre-procesamiento de los datos, entiéndase por esto, la selección de atributos, la construcción de objetos y la selección de objetos.

## 1.1. Clasificación supervisada

Se puede decir que la clasificación supervisada es una de las áreas más importantes dentro del RP [18], la cual está dividida en dos fases principales, una fase de entrenamiento y una fase de clasificación. La primera de ellas parte de la presencia de un grupo de situaciones u objetos descritos por un conjunto de atributos también llamados propiedades, características o rasgos. Las situaciones pueden referirse a estados, ejemplos, tipificaciones, incluso categorías [32]. Durante el proceso de aprendizaje se le proporciona al modelo una matriz de conocimiento donde viene codificado el conjunto de objetos descritos por sus rasgos, además de la clase a la que pertenece cada uno; este elemento es lo que define el carácter supervisado de esta fase comúnmente llamado aprendizaje supervisado, a diferencia del aprendizaje no supervisado (no se provee la clase a la que pertenece cada objeto) [33]. A partir del conocimiento previo de situaciones históricas, es entonces objetivo de los modelos, determinar una clase o clasificación para cualquier situación nueva y desconocida.

### 1.1.1. Clasificadores basados en reglas

Los clasificadores basados en reglas han sido ampliamente aceptados debido a su fácil interpretabilidad y entendimiento. El descubrimiento de reglas es objeto de estudio desde hace varias décadas y muchos han sido los algoritmos propuestos [34][35]. Los mismos se pueden clasificar según el modelo que adoptan para el descubrimiento de las reglas [36].

- Métodos basados en algoritmos de cubrimiento.

- Métodos basados en árboles de decisión.
- Métodos basados en Asociación.
- Métodos optimales basados en Asociación.

Según [36], de los métodos anteriores, los dos primeros usualmente producen pequeños conjuntos de reglas, en contraste con los dos últimos. Tradicionalmente en la construcción de clasificadores basados en reglas se prefiere un conjunto pequeño de reglas, partiendo del argumento de que las mismas no se sobreajustan al conjunto de entrenamiento y producen mejores resultados de exactitud sobre el conjunto de prueba. Sin embargo el tamaño reducido del conjunto de reglas puede acarrear algunos problemas, el principal de todos es la intolerancia a la presencia de datos perdidos en los datos de prueba, nunca antes vistos. Esto los convierte en clasificadores no robustos, lo cual ha llevado a los investigadores a buscar un modelo que permita obtener modelos relativamente pequeños, pero a la vez capaces de tolerar ciertos inconvenientes.

Los clasificadores basados en reglas usualmente incluyen dos etapas, la etapa de aprendizaje y la etapa de prueba. En la primera el clasificador genera un conjunto de reglas, donde cada regla asocia un patrón con una clase, luego en la otra etapa se aplica el conjunto de reglas a los datos de prueba, esperando que la clase asignada sea la esperada; si esto ocurre, entonces la clasificación es correcta, en otro caso es incorrecta.

En conjunto los modelos para la construcción de clasificadores basados en reglas pueden dividirse en dos grupos [36]:

- Clasificadores basados en reglas ordenadas: las reglas son organizadas en secuencia, por ejemplo en orden descendente de exactitud. De esta forma, cuando se realiza una nueva clasificación, la predicción se realiza por la primera regla en la secuencia que coincida con los valores del objeto a clasificar. Esta secuencia usualmente contiene una regla para una clase por defecto, de esta forma si no se obtiene una regla en toda la secuencia, que coincida con los valores del objeto a clasificar, entonces dicha regla realiza la predicción por defecto. Algoritmos como C4.5rules [37] y CBA [38] aplican este modelo.

- Clasificadores basados en reglas no ordenadas: las reglas no son ordenadas en una secuencia y todas las reglas que coincidan, participan en la determinación de la clase del objeto a clasificar. Una forma directa de hacer esto es aceptando la regla mejor votada como se aplica en CPAR [37]. Otro método más complejo es comparar las exactitudes actuales obtenidas de las múltiples reglas para todas las posibles clases y la predicción se realiza por la regla de mayor exactitud. Algoritmos que emplean este modelo son el CN2 mejorado [39] y el CMAR [40].

El primer modelo es simple y efectivo, realiza la predicción basado en la mejor verosimilitud de las reglas. Esto es debido a que las reglas con mayor valor de precisión preceden a reglas con menor valor de precisión y la misma aproxima la probabilidad condicional cuando el conjunto de datos es grande, lo que permite que la predicción sea lo más precisa posible, cualquiera que sea la dimensión del conjunto de datos [36]. Por otro lado el segundo modelo está en camino aún de convertirse en un modelo maduro, debido a que el concepto de independencia de reglas aún está en desarrollo, aunque ya se han visto intentos como los introducidos por Clark en el CN2 mejorado.

De modo general, es importante destacar que aunque los clasificadores basados en reglas buscan producir un conjunto pequeño de reglas, ya sea por un modelo u otro se hace necesario en todo momento encontrar un predictor robusto, que no es más que generar un conjunto de reglas con un alto valor de precisión en las predicciones sin importar, que el conjunto de prueba no sea tan completo como el conjunto de entrenamiento con el que se generó el conjunto de reglas.

### 1.1.2. Árboles de decisión

Otro grupo importante de clasificadores, lo constituyen los modelos basados en árboles de decisión (*Decision tree* - DT). Un árbol de decisión es un diagrama de flujo con la misma estructura de un árbol, donde cada nodo interno (nodo no hoja) denota una prueba sobre un atributo determinado, cada rama representa un resultado sobre la prueba que se realiza y cada nodo terminal (nodo hoja) contiene una clase [33]. El funcionamiento de los árboles de decisión para la clasificación parte de una tupla  $X$  para la cual la clase asociada es desconocida. Los valores de los atributos de la tupla son probados contra el árbol previamente creado. Una ruta es trazada desde la

raíz hasta el nodo hoja que contiene la clase para la tupla a clasificar. Un elemento importante de los árboles de decisión es que pueden ser fácilmente convertidos en reglas; estas características además de no necesitar conocimiento alguno del dominio sobre el que trabaja y la ausencia de parámetros para poder construirlos, los convierten en uno de los clasificadores con mayor popularidad [33]. Los árboles de decisión pueden manejar datos de distintas dimensiones. Un elemento a su favor es su representación en forma de árbol, lo que permite que sea intuitivo y fácil de asimilar por los humanos. Las fases de aprendizaje y clasificación son simples y rápidas, y por lo general presentan buena exactitud; sin embargo el éxito de su uso puede depender de los datos que se tengan.

Los árboles de decisión han sido usados para la clasificación en disímiles aplicaciones [21], tales como medicina, producción y manufactura, análisis financieros, ingeniería de software entre otros [41]. Dentro de los algoritmos representativos de esta familia, se pueden encontrar en primer lugar al ID3 (*Iterative Dichotomiser*), desarrollado por J. Ross Quinlan entre los años 1970 y 1980 [42]. Un poco más tarde el mismo autor presenta el C4.5 [20], lo que sería el sucesor del ID3, y se convierte así en el medidor para los futuros algoritmos del aprendizaje supervisado [33]. Por otra parte un grupo de estadísticos (L. Breiman, J. Friedman, R. Olshen, and C. Stone) publicaron un libro donde se describía la generación de árboles binarios de decisión, dicho libro dio nombre a los algoritmos CART (*Classification and Regression Trees*) [43] ampliamente usados en la predicción de valores numéricos (regresión).

### 1.1.3. Redes Neuronales Artificiales (RNA)

Los modelos de redes neuronales artificiales, o simplemente redes neuronales, se conocen por diversos nombres como modelos conexionistas o modelos de procesamiento distribuido paralelo. En lugar de ejecutar un programa secuencialmente como en una arquitectura Von Neumann, la red neuronal explora muchas hipótesis simultáneamente, y usa redes masivamente paralelas compuestas por muchos elementos de procesamiento conectados por enlaces con pesos [44].

Los modelos de redes neuronales son especificados por la topología de la red en dependencia de la estructura y el tipo de enlaces entre sus nodos, las características de los mismos, a lo que se le denomina modelo de la neurona y las reglas de aprendizaje,

mecanismo que usa para ajustar los pesos de los enlaces.

Dicho de otra forma, una red neuronal es un conjunto de unidades computacionales simples, las cuales constituyen neuronas artificiales [33] con varias conexiones de entrada y de salida que les permite comunicarse con las demás neuronas de la red; cada una de estas conexiones tiene asociada un peso  $W_{ij}$ , que indica la significación de la información que llega por esta conexión, o sea, la influencia que tiene la activación alcanzada por la neurona  $U_i$  sobre la neurona  $U_j$ . Cada neurona  $U_i$  calcula una activación a partir de las activaciones de las neuronas conectadas directamente a ella, de los pesos de los arcos a través del que llega cada activación y usa un algoritmo que generalmente es el mismo para todas las unidades [45].

Las RNA, aprenden a partir de ejemplos, ya sea para el reconocimiento de patrones o para la clasificación de datos o cualquier otra tarea; se configura una red a través del proceso de aprendizaje. Uno de los métodos más populares de aprendizaje, característico de las Redes Neuronales Multicapa (*Multilayer Perceptron Networks* - MLP) está basado en el error del gradiente descendente, a través del algoritmo *back-propagation*; que ajusta los pesos de las conexiones entre las neuronas hasta que sea capaz de predecir correctamente la clase de las tuplas de entrenamiento. Se puede decir que la popularidad de este modelo de aprendizaje se debe a que es sencillo sin importar lo complicada que sea la estructura de la red; incluso los demás modelos de entrenamiento pueden ser vistos como modificaciones del mismo [5].

Un elemento importante en la clasificación de las redes neuronales artificiales lo constituye la arquitectura o topología de la misma. La estructura óptima está determinada por el dominio del problema a resolver. Muy a menudo el conocimiento del problema ya, sea informal o heurístico, es fácil de incorporar a la topología de la red, pues solo es necesario especificar el número de capas ocultas, el tipo de neurona, las conexiones de retroalimentación y otros parámetros necesarios [5]. Estas libertades a la hora de construir una red neuronal ha dado cabida al surgimiento de distintas topologías y mecanismos tanto de aprendizaje como de clasificación.

Según [46] se pueden definir dos grandes grupos, divididos principalmente por sus mecanismos de clasificación (redes *Feed-Forward* and redes Recurrentes/*Feedback*). Dentro del primer grupo se encuentran redes tales como *Simple-Layer Perceptron*, *Multilayer Perceptron* y *Radial Basis Function network* que poseen un buen compor-

tamiento en problemas de clasificación, aproximación de funciones y en la construcción de modelos matemáticos [47]. El segundo grupo lo componen redes tales como *Competitive networks*, *Kohonen SMO* y *Hopfield network*, entre otras. Este grupo de redes ha tenido buenos desempeños en la predicción de series de tiempo, donde han encontrado su principal aplicación, aunque en investigaciones recientes se han obtenido resultados exitosos en tareas de clasificación donde es necesario analizar además, elementos propios de la organización de los datos y la influencia de estos en los resultados [48].

Las RNA constituyen una herramienta poderosa y veloz para la construcción de clasificadores. Provee muy buenos resultados sobre una amplia gama de problemas. Su principal deficiencia es lo complejo de su estructura y al poseer una interpretabilidad casi nula, la hace poco intuitiva, de manera tal que no es posible encontrar una explicación clara de componentes que determinan sus resultados. Esto ha impulsado a un número considerable de investigadores a buscar mecanismos para la extracción del conocimiento embebido en las RNA entrenadas. Entre los métodos se incluyen la extracción de reglas a partir de las redes y el análisis de sensibilidad [33]

#### 1.1.4. Máquinas de Vectores de Soporte (*Support Vector Machines-SVMs*)

Las Máquinas de Vectores de Soporte son métodos de aprendizaje supervisados relativamente nuevos, principalmente usados en la clasificación binaria. Este método fue introducido por Vapnik en el año 1995, aunque ya el mismo autor había hecho mención de ello 20 años antes [49]. La idea básica de su funcionamiento es tratar de encontrar un hiperplano que permita dividir perfectamente dos clases a partir de un conjunto de datos en un espacio n-dimensional [50]. Cuando se está en presencia de esto se puede decir que los datos son linealmente separables y el hiperplano a encontrar, es una combinación lineal de puntos del espacio n-dimensional. Es evidente suponer que para un conjunto de datos es posible encontrar múltiples hiperplanos, por lo que uno de los objetivos principales del aprendizaje de las SVMs, es encontrar el hiperplano óptimo que separe los elementos de ambas clases; pero de forma tal que se maximice la distancia geométrica entre el hiperplano y los puntos más cercanos del conjunto de datos. La solución del hiperplano óptimo puede ser escrita como la

combinación de unos pocos puntos de entrada que son llamados vectores de soporte [51].

En la vida real, no siempre es posible encontrar una división lineal, debido a la propia distribución de los datos. En estos casos las SVMs introducen la noción de "función núcleo (*kernel*) del espacio de características", lo que permite calcular el producto punto de los puntos de entrada del espacio original; lo que genera un espacio dimensional superior, donde los datos sean separables.

Las SVM son básicamente clasificadores para 2 clases. Es posible cambiar la formulación del algoritmo que permite calcular el hiperplano delimitador para la clasificación multiclase. Dicho de otra forma, los datos son divididos en dos partes de diferentes formas y una SVM es entrenada para cada forma de división. La clasificación multiclase es hecha combinando la salida de todos los clasificadores [51].

Como fortalezas de este modelo, se puede asegurar que su entrenamiento es relativamente fácil, no es necesario tener conocimiento previo alguno sobre los datos de entrada, incluso datos no tradicionales como cadenas de caracteres y estructuras como árboles pueden ser usados como entrada, además de vectores de características como tradicionalmente se codifican los datos. No es sensible a encontrar óptimos locales como las redes neuronales. Los datos son escalados relativamente bien en espacios dimensionales altos. Un elemento importante es que la relación entre la complejidad del clasificador y el error puede ser controlado explícitamente. Su principal debilidad radica en que se necesita una buena función *kernel*, es decir, se necesitan metodologías eficientes para ajustar los parámetros de inicialización de la SVM [51].

### 1.1.5. Vecino más cercano

El clasificador de los  $k$  vecinos más cercanos (*Nearest Neighbors*, NN, por sus siglas en inglés) constituye uno de los procedimientos de clasificación más atractivos, concebido por Fix y Hodges [52] y es una forma práctica y de fácil aplicación para predecir o clasificar un nuevo dato, basado en observaciones conocidas o históricas. Esta técnica se basa, simplemente, en recordar todos los ejemplos que se vieron en la etapa de entrenamiento. Cuando un nuevo caso se presenta al sistema, este se clasifica según el comportamiento del caso más cercano [53].

A manera de ejemplo, el caso de un médico que está tratando de predecir el

resultado de un procedimiento quirúrgico puede predecir que el resultado de la cirugía del paciente será aquel del paciente más parecido que conoce, que haya sido sometido al mismo procedimiento [54]. Este modelo de clasificación puede ser un tanto extremo, pues en la práctica resultados un tanto menos similares pueden contener la correcta clasificación de una situación determinada y no así el caso más cercano. Por esta razón el método del vecino más cercano se generaliza para usar la regla de los  $k$  vecinos más cercanos. La idea básica sobre la que se fundamenta este paradigma es que un nuevo caso se va a clasificar en la clase más frecuente a la que pertenecen sus  $k$  vecinos más cercanos. El paradigma se fundamenta por tanto en una idea muy simple e intuitiva, lo que unido a su fácil implementación hace que sea un paradigma clasificadorio muy extendido [26].

Los clasificadores del vecino más cercano pueden ser usados también para la predicción numérica, devolviendo un valor real que no sería más que el promedio de todos los valores asociados a la clase de cada uno de los vecinos más cercanos. Si se analiza el comportamiento de estos clasificadores, es necesario determinar cómo se pueden comparar los distintos objetos de un conjunto de datos. Inicialmente puede definirse como caso (situación u objeto) a una tupla donde cada uno de sus componentes corresponde con los valores que lo caracterizan. El primer análisis es posible abordarlo, asumiendo que todos los valores son numéricos, por lo tanto es posible determinar la similitud entre dos objetos a partir de la diferencia entre cada uno de los valores de los atributos que caracterizan ambos objetos, siguiendo el criterio definido por el cálculo de la distancia Euclidiana. En el caso de que los valores sean nominales o categóricos, es usual que se comparen de la siguiente forma: si son idénticos entonces, la diferencia entre esas componentes es 0, si son distintos, entonces la diferencia es 1. Incluso otros métodos más sofisticados pueden emplearse, tanto para valores numéricos como nominales [33].

A pesar de ser uno de los principales modelos de clasificación, por su extendido, facilidad de implementación y sencillez; presenta un grupo de debilidades, principalmente la selección del  $k$  óptimo. Este parámetro suele ser escogido de forma experimental, aunque se han desarrollado estrategias y métodos para esto; pues experimentalmente se ha demostrado que a medida que el número de datos de entrenamiento aumenta, el valor de  $k$  debe ser más grande. Otra variante ha sido la inclusión de

pesos en los rasgos que describen los casos de entrenamiento, puesto que la Exactitud de la Clasificación también sufre cuando trabaja con valores ruidosos (valores perdidos y *outliers*).

### 1.1.6. Algoritmos de tipo pivote

El modelo de algoritmos de clasificación supervisada de tipo pivote denominado ALVOT [55][56] tuvo sus orígenes a mediados de la década de los 60 y su desarrollo se debe a los resultados obtenidos por el especialista ruso Zhuravliov en conjunto con su grupo de trabajo [1]. Sus principales avances han sido desarrollados en el enfoque lógico combinatorio de patrones [55].

La idea de este modelo está basada en el concepto de precedencia parcial, que es lo mismo que analogías parciales, que radica en que la comparación entre dos objetos se puede realizar parte a parte (parcialmente), y no necesariamente entre toda la descripción completa del objeto, lo que permite encontrar a partir de las partes similares, determinadas regularidades en las características de los objetos. El modelo general se resume en el siguiente conjunto de etapas definidas en [1]:

- Sistema de conjuntos de apoyo.
- Función de semejanza.
- Evaluación por fila dado un conjunto de apoyo fijo.
- Evaluación por clase dado un conjunto de apoyo fijo.
- Evaluación por clase para todo el sistema de conjuntos de apoyo.
- Regla de solución.

Para obtener un algoritmo de votación es necesario determinar el conjunto de parámetros necesarios para cada una de las etapas anteriormente descritas.

En la literatura se han propuesto diferentes mejoras al modelo de algoritmos ALVOT, basadas principalmente en optimizar la estrategia de búsqueda para calcular el sistema de conjuntos de apoyo, y en la edición de objetos y de los conjuntos de apoyo manejados por el algoritmo [27] [28]. Un dato curioso consiste en que la mayoría

de los algoritmos reportados en la literatura son modelos no incrementales, y estos algoritmos presentan la necesidad de mantener el conjunto de datos completo en la memoria principal. Pero, si el tamaño del conjunto de datos es grande, entonces la aplicación del algoritmo puede no ser factible. Otro inconveniente es la reclasificación de un objeto cuando la muestra de aprendizaje es incrementada con más objetos (i.e. un conjunto de datos al cual le han sido añadidos nuevos objetos), lo cual implica procesar nuevamente todos los objetos de la muestra de aprendizaje. En [57] se intentan resolver estas deficiencias, y se obtienen buenos resultados con respecto a la ejecución del algoritmo.

### 1.1.7. Clasificadores asociativos

Estos clasificadores basan gran parte de su funcionamiento en memorias asociativas reconocidas por su capacidad de reconocer patrones. La principal razón para ello es que tienen como objetivo recuperar correctamente patrones completos a partir de patrones de entrada los cuales pueden o no estar alterados por ruido [8].

Se entiende como memoria asociativa un sistema  $M$  un patrón de entrada representado como un vector columna denotado  $x$  y un patrón de salida  $y$  representado de la misma forma. Cada uno de los patrones de entrada crea una asociación con el correspondiente patrón de salida. Dicha asociación es similar a un par ordenado; por ejemplo, los patrones  $x$  y  $y$  del esquema anterior forman la asociación  $(x, y)$ .

La memoria asociativa  $M$  se representa mediante una matriz, la cual se genera a partir de un conjunto finito de asociaciones conocidas con anterioridad, a este nuevo conjunto se le conoce como conjunto fundamental de aprendizaje, o simplemente conjunto fundamental representado de la siguiente forma:

$$\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\} \tag{1.1}$$

donde  $p$  es un número entero positivo que representa la cardinalidad del conjunto fundamental.

A los patrones que conforman las asociaciones del conjunto fundamental se les llama patrones fundamentales. La naturaleza del conjunto fundamental proporciona un importante criterio para clasificar las memorias asociativas: una memoria es

Autoasociativa si se cumple que  $x^\mu = y^\mu \forall \mu \in \{1, 2, \dots, p\}$ , por lo que uno de los requisitos que se debe cumplir es que  $n = m$ , es decir, la matriz que se genera debe ser cuadrada. De otra manera si  $\exists \mu \in \{1, 2, \dots, p\}$  para el que se cumple que  $x^\mu \neq y^\mu$  se trata de una memoria Heteroasociativa. Nótese que pueden existir memorias heteroasociativas con  $n = m$ .

En los problemas donde intervienen las memorias asociativas, se consideran dos fases importantes: la fase de aprendizaje, que es donde se genera la memoria asociativa a partir de las  $p$  asociaciones del conjunto fundamental, y la fase de recuperación que es donde la memoria asociativa opera sobre un patrón de entrada.

Los principales modelos de este enfoque son: *Lernmatrix* [58], *Correlograph* de Willshaw, Buneman y Longuet-Higgins [59], *Linear Associator* [60], *Memoria Hopfield* [61], Memoria Asociativa Bidireccional (BAM, por sus siglas en inglés) de Kosko [62] y *Memorias Asociativas Alfa-Beta* [14], siendo los cuatro primeros modelos basados en el anillo de los números racionales con las operaciones de multiplicación y adición, el antepenúltimo modelo esta basado sobre las operaciones morfológicas y el paradigma de suma de productos y el penúltimo utiliza máximos y mínimos de dos operaciones nuevas expresadas en [14] conocidas como operación Alfa y operación Beta y el último basado en nuevas operaciones sustentadas en [63].

Dentro de este grupo de clasificadores, representantes del enfoque asociativo del reconocimiento de patrones; se puede encontrar al Clasificador híbrido asociativo (*Hybrid Associative Classifier* - HAC) creado por Santiago en el 2003 [64]. HAC es un clasificador que combina en la fase entrenamiento el modelo asociativo *Linear Associator*, que tiene su origen en los trabajos pioneros de 1972 publicados por Anderson y Kohonen [8]. En esta fase de aprendizaje, para cada una de las  $p$  asociaciones  $(x^\mu, y^\mu)$  se determina la matriz  $y^\mu \cdot (x^\mu)^t$  de dimensiones  $n \times m$ . Luego se adicionan las  $p$  matrices para obtener la siguiente memoria:

$$M = \sum_{\mu=1}^p y^\mu \cdot (x^\mu)^t = [m_{ij}]_{m \times n} \quad (1.2)$$

En la fase de recuperación o clasificación de este modelo se realiza siguiendo el algoritmo *Lernmatrix* de Steinbuch, que no es más que una memoria heteroasociativa que funciona como un clasificador de patrones binarios; acepta como entrada un patrón binario  $x^\mu \in A^n, A = \{0, 1\}$ , y produce como salida la clase  $y^\mu \in A^p$  que le

corresponde (de entre  $p$  clases diferentes), codificada esta con un método que en la literatura se le ha llamado one-hot [30][8].

El desempeño del clasificador HAC se ve afectado cuando los patrones de entrada están agrupados en un mismo cuadrante, las magnitudes de los patrones de entrada difieren en gran medida y el clasificador tiende a clasificar patrones de menor magnitud en clases de patrones de mayor magnitud. Esta situación tiende a producir errores de clasificación. Algunas de estos problemas son resueltos en una mejora realizada por el mismo autor [64], a través de una traslación de los ejes, lo que permite una mejor clasificación de los patrones de entrada [30], y crea así el Clasificador Híbrido Asociativo con Traslación (CHAT).

### 1.1.8. Clasificador Asociativo Gamma

El Clasificador Asociativo Gamma, como su nombre lo indica, forma parte del conjunto de clasificadores supervisados del enfoque asociativo, se puede decir que es un derivado directo de las Memorias Asociativas Alfa - Beta, dado que su funcionamiento base está compuesto por las operaciones definidas en las mismas.

Inicialmente se parte de un conjunto de entrenamiento (CE) cuyas instancias se codifica siguiendo el algoritmo Jhonson-Möbius Generalizado [6]. Esta codificación permite obtener instancias primeramente con valores numéricos positivos los cuales se codifican a una cadena de *bits*. Esta codificación determina para cada rasgo, la mayor cantidad de dígitos binarios necesarios, para representar los valores que caracterizan las instancias del CE. La determinación de este valor al cual en lo adelante llamaremos  $e_m(i)$  permite obtener el conjunto  $E_m$ . A partir de este punto se realiza el cálculo del parámetro  $\rho$  que no es más que el menor de los valores del conjunto  $E_m$ .

$$e_m(j) = \bigvee_{i=1}^p x_i^j \quad (1.3)$$

$$\rho = \bigwedge_{i=1}^p e_m(j) \quad (1.4)$$

Luego, para poder clasificar el nuevo patrón, es necesario codificarlo también, y se obtiene para cada componente un valor en su respectiva representación binaria.

Un parámetro esencial es el valor  $\theta$  que se inicializa en cero. Posteriormente cada una de las instancias del CE es agrupada según la clase a la que pertenece cada una, lo que permite obtener tantos grupos como clases se tenga en el CE. A partir de este momento se calcula la gamma similitud generalizada  $\gamma_g(x_i, y_i, \theta)$  de cada una de las componentes  $x_i$  codificadas del patrón a clasificar y las componentes  $y_i$  de cada uno de los patrones agrupados anteriormente.

$$\gamma_g(x, y, \theta) = \begin{cases} 1 & \text{si } m - \mu_\beta[\alpha(x, y) \bmod(2)] \leq \theta \\ 0 & \text{e. o. c} \end{cases} \quad (1.5)$$

Donde  $m$  es la dimensión del vector binario que representa el valor de la componente que está siendo analizada,  $\mu_\beta$  y  $\alpha(x, y)$  son los operadores gamma y alfa definidos en [6], en este caso se considera que dos componentes son similares, si el resultado es menor que el valor de  $\theta$ .

Se calcula la suma ponderada de los valores obtenidos de la gamma similitud entre las componentes del patrón a clasificar y los patrones agrupados según sus respectivas clases, de esta forma es posible obtener un valor  $C_i$  que determina la similitud del nuevo patrón a los patrones que representan a cada clases.

$$C_i = \frac{\sum_{\omega=1}^{k_i} \sum_{j=1}^n w_j * \gamma_g(x_j^\omega, p_j, \theta)}{k_i} \quad (1.6)$$

donde  $k_i$  es la cantidad de instancias en la clase  $Y_i$ ,  $n$  la cantidad de rasgos que describen a las instancias,  $w_j$  el peso asociado al  $j$ -ésimo rasgo, con  $w_j \in [0, 1]$ ,  $x_j^\omega$  el valor asociado a al  $j$ -ésima componente del  $\omega$ -ésimo elemento de la clase  $Y_i$  y  $p_i$  a  $j$ -ésima componente del patrón  $p$  a clasificar.

Si del conjunto resultante, se obtiene un valor máximo único, el clasificador asigna al nuevo patrón la clase del grupo al que le corresponde dicho valor máximo. En caso de que no se obtenga un máximo único, entonces se repite el cálculo de la gamma similitud pero incrementando el valor de  $\theta$ .

Este proceso se repite hasta que se encuentra un valor máximo único o mientras el valor de  $\theta$  no exceda el valor de  $\rho$  inicialmente calculado, en cuyo caso se asigna la clase que posea el primer valor máximo  $C_i$ . Este clasificador será explicado con mayor detalle en el próximo capítulo.

Teniendo en cuenta la caracterización realizada en [15], partiendo del uso prolongado del clasificador en conjuntos de datos con disímiles características y con objetivos diferentes, principalmente en tareas de clasificación, para lo cual fue diseñado y en tareas de interpolación y exploración de funciones, estas últimas para las cuales no fue diseñado; se determinó que es posible esperar buenos resultados cuando los datos inducen una función. Dicho de otra forma, si a cada elemento de entrada le corresponde un único patrón de salida o clase, entonces el clasificador es competitivo e incluso superior a otros algoritmos. En el caso contrario, es decir, que para un patrón de entrada existen varios patrones de salida, entonces el algoritmo se encuentra en una situación no prevista por el algoritmo original.

El algoritmo posee un rendimiento competitivo también en el caso de que se presente una secuencia conocida, cuya salida será el valor conocido que mejor corresponda a esa secuencia. Sin embargo, es claro suponer que puede darse el caso de que dicha secuencia no sea exactamente igual a una conocida, sino que quede cercana a la frontera entre dos o más secuencias conocidas. Entonces, se puede obtener una salida cercana a la frontera entre las salidas conocidas correspondientes.

## 1.2. Pre-procesamiento de datos

El pre-procesamiento de los datos usualmente tiene un impacto directo en el comportamiento de los algoritmos de clasificación supervisada. La eliminación del ruido en los datos constituye una de las tareas más difíciles en el intento de obtener una base de casos lo más refinada posible [65]. Cuando se habla de eliminación de ruido, entiéndase por esto al conjunto de valores que caracterizan los objetos de una base de conocimiento y que se desvían en gran medida de los valores normales presentes en dicha base. A este conjunto de problemas en la literatura también lo reconoce como *outliers*. Otro problema importante surge cuando la base de conocimiento es excesivamente grande, lo que provoca que sea prácticamente imposible de realizar el aprendizaje supervisado. Es común también que se tenga que lidiar con valores perdidos durante la fase de preparación de los datos [66]. Incluso la presencia de desbalance existente entre la cantidad de objetos de unas clases con otras, puede provocar que el aprendizaje y la posterior clasificación sea errónea.

Cuando los investigadores hacen uso de valores provenientes de problemas reales, se enfrentan en muchos casos con que los datos están representados por muchos rasgos y solo unos pocos de ellos guardan relación directa con el objetivo del problema. Puede existir redundancia, donde varios rasgos pueden tener una alta correlación, lo cual hace que no sea necesario incluirlos todos en el modelo. También es válido encontrar interdependencia, de forma tal que dos o más rasgos contengan información relevante, que de excluirse alguno de ellos, puede hacer que no se aproveche dicha información [67]. Es usual que se tengan que enfrentar los problemas de disponibilidad de los datos, que trae consigo en la mayoría de los casos que los objetos no estén balanceados, haciendo que los algoritmos se decanten por la clase mayoritaria e incluyan en esa misma categoría a objetos de clases minoritarias.

### 1.2.1. Selección de rasgos

El problema de la selección de rasgos (*Feature Selection*) está presente de forma permanente en el pre-procesamiento de los datos. El principal beneficio de una correcta selección consiste en el mejoramiento del aprendizaje inductivo como fase inicial en los problemas de clasificación supervisada. Dicha mejora, puede evaluarse tanto en términos de rapidez del aprendizaje como capacidad de generalización o en términos de simplicidad del modelo obtenido [68]. Además existen otros beneficios asociados a un número pequeño de rasgos tales como la obtención de una mayor comprensión del modelo resultante [16].

Existen diversas razones que pueden hacer oscuro el proceso de selección de rasgos, tales como la presencia de rasgos irrelevantes y redundantes, ruido en los datos o interacción entre los atributos. Esto ocurre cuando se está en presencia de cientos o miles de rasgos, como es el caso de los microarreglos de las secuencias de ADN y las cadenas de aminoácidos que representan las proteínas. Otro caso puede ser cuando hay en presencia de bases de casos con un número elevado de rasgos y poca cantidad de casos, lo que provoca que la tarea del aprendizaje sea muy difícil para la mayoría de los algoritmos, pues el espacio de búsqueda estará escasamente poblado, y el modelo no sería capaz de diferenciar correctamente lo que sería ruido o dato relevante [69].

La mayoría de los autores coinciden en que la Selección de Rasgos puede ser divi-

dida en dos grandes tareas: decidir qué rasgos son los que mejor describen el contexto y seleccionar cuál es la mejor combinación de estos que mejore la precisión predictiva; sin embargo, las discrepancias ocurren por los criterios usados para definir tanto la relevancia como la redundancia en los datos [9]. En aras de determinar la relevancia, en el año 1991 Almuallim y Dietterich estuvieron de acuerdo en que la relevancia de la información debe ser considerada como un valor booleano (relevante o no relevante), libre de ruido y definida en términos de distribuciones de probabilidad, de esta forma era posible obtener una estimación confiable que sugiriera la eliminación de rasgos claramente identificables [70].

Existen dos grandes aproximaciones en la selección de rasgos: la evaluación individual y la evaluación por subconjuntos. La evaluación individual también conocida como *Feature Ranking* [71], evalúa individualmente los rasgos asignándoles pesos de acuerdo con el nivel de relevancia del rasgo. Por otro lado la evaluación por subconjunto, parte de la creación de subconjuntos de rasgos candidatos basados en cierta estrategia de búsqueda [16]. Otra clasificación de los métodos de selección de rasgos puede ser la definida en [72], dividiéndolos en tres grupos: filtros (*Filters*), envolturas (*wrappers*) y embebidos (*embedded*).

El primero de ellos incluye a los algoritmos de selección de rasgos que son independientes del algoritmo predictor. Típicamente estos algoritmos buscan identificar aquellos rasgos que tienen poca probabilidad de ser útiles para el análisis de los datos. Los métodos de filtrado se basan en la evaluación de alguna métrica calculada directamente sobre los datos sin recibir retroalimentación del algoritmo predictor; consecuentemente, los algoritmos de filtrado usualmente son computacionalmente menos costosos que los métodos de envoltura (*Wrappers*).

Los métodos *wrapper* funcionan como una envoltura alrededor del algoritmo predictor que entrega en cada iteración un subconjunto diferente de rasgos y recibe la retroalimentación del desempeño alcanzado. Generalmente dicha retroalimentación está en términos de la precisión predictiva alcanzada por el algoritmo predictor usando un subconjunto específico de características. El enfoque de envoltura busca mejorar el desempeño de un predictor específico a través de la búsqueda del subconjunto óptimo de características [9].

El tercer grupo de métodos de selección es muy específico del algoritmo en el

cual están incluidos. La búsqueda del subconjunto de rasgos óptimos es realizada durante el proceso de aprendizaje del clasificador. El proceso de aprendizaje y la selección de rasgos no puede ser separada, pues la estructura de la clase de funciones juegan un papel crucial [73]. Tienen la ventaja de que la selección que realizan está directamente interrelacionada con el modelo de clasificación y es incluida en los rasgos seleccionados; su desventaja radica en que no es tan intensivo como los métodos de envoltura [74].

Teóricamente, los métodos de selección de rasgos buscan sobre el conjunto de rasgos, tratando de encontrar el mejor de los  $2^N$  subconjuntos posibles, de acuerdo con una función de evaluación determinada. Encontrar el mejor subconjunto es usualmente intratable, es por esto que el problema de la selección de rasgos adopta en muchos casos métodos heurísticos de búsqueda, con la idea de reducir la complejidad computacional. Los autores Dash y Liu introdujeron en el año 1997 una plataforma básica para los métodos de selección de rasgos, definida en los siguientes 4 pasos [75]:

1. Procedimiento de generación para generar el próximo subconjunto candidato
2. Función de evaluación para evaluar el subconjunto que se está analizando
3. Criterio de parada para decidir cuándo se detiene la búsqueda
4. Procedimiento de validación para determinar si el subconjunto es válido.

Teniendo en cuenta las diferencias de aplicación de los conjuntos de datos, es posible clasificar los métodos de selección en supervisados y no supervisados. La diferencia entre ellos radica en el problema de clasificación que se vaya a realizar. Si se usa un conjunto de datos con sus respectivas clasificaciones para el aprendizaje de determinados patrones, entonces, se denomina supervisado. Si solamente se usa un conjunto de datos sin su clasificación y su basamento es en su estructura interna, entonces el método es no supervisado.

Dentro de los métodos para la selección supervisada se pueden encontrar los algoritmos agrupados como filtros y envolturas, basados en sus dependencias con sus respectivos métodos inductivos. Dentro de la selección no supervisada se pueden encontrar métodos tales como Análisis de Componentes Principales (PCA - por sus siglas en inglés) [76], Factor de Análisis y Análisis de Componentes Independientes (ACI) [77].

### 1.2.2. Construcción de objetos

La construcción o generación de objetos, se enfoca en la creación de nuevos objetos no presentes necesariamente en el conjunto de entrenamiento, tratando siempre de que contengan la información más relevante, y utilizando estos para la clasificación final del problema en cuestión.

Con este objetivo, existe toda una familia de métodos de construcción de objetos representativos del conjunto original. Son varias las ideas propuestas, como la mezcla de objetos cercanos, el aprendizaje competitivo y los métodos basados en *Bootstrap* [78]. De forma general presentan buenos comportamientos; en este sentido toma especial atención la selección de los parámetros (generalmente por prueba y error) y es necesario realizar suficientes iteraciones. Generalmente no se aplican si existen variables cualitativas, por la dificultad de definir las operaciones de suma, multiplicación por un escalar y promedio con sentido en estos dominios.

El primer creador de un método de construcción de objetos fue Chang en 1974 [79]. Este método se basa en la sustitución de manera iterativa los dos objetos más cercanos de la misma clase por su media ponderada, en dependencia de la degradación o no de la eficacia del clasificador. La ponderación se realiza con los pesos asociados a los objetos, que son inicializados en 1 y luego se suman para obtener el peso resultante al ocurrir una mezcla. De esta manera se garantiza que los objetos resultantes de varias mezclas, se mantengan lo mayor posible en su posición actual. Este proceso, que lleva asociado un alto costo computacional, es optimizado por el propio autor de varias maneras. Debido a que la mezcla de objetos es guiada sólo por la consistencia, el resultado final puede no ser representativo del conjunto original [80].

En 1998, Bezdek et al. [81], fueron los autores de un nuevo método basado en el método de Chang, denominado MCA (*Modified Chang Algorithm*). Una de las modificaciones introducidas es el empleo de un promedio simple entre objetos y no una media ponderada. Además, se modificó el proceso de búsqueda de candidatos para realizar la mezcla mediante el particionado de la matriz de distancias original en submatrices de igual clase. De esta forma se disminuye considerablemente la cantidad de comparaciones que se realizan para buscar los objetos que son más cercanos.

Mollineda et al. en el 2002 [80] introdujeron a su vez una variante de MCA, substituyendo la mezcla de objetos por la mezcla de agrupamientos (clusters). Los

objetos resultantes del método son los centroides de los grupos resultantes. Para agilizar el proceso de búsqueda, se utiliza la desigualdad triangular, es por esto que sólo puede ser aplicable en espacios métricos.

Otro enfoque es el aprendizaje competitivo, dentro del mismo existen varios representantes, tales como el LVQ (*Learning Vector Quantization*) [82], DSM (*Decision Surface Mapping*) [83], LVQ-TC (*Learning Vector Quantization with Training Counters*) [84] y VQ (*Vector Quantization*) [85]. En general estos métodos se basan en la idea de la selección aleatoria de un subconjunto inicial de instancias, y su modificación en varias iteraciones. La estrategia de que "el ganador se lo lleva todo" es utilizado en casi todos, siendo el objeto más cercano a cada objeto del conjunto inicial desplazado. La dirección del desplazamiento de los objetos coincide con la línea recta que une al prototipo con el objeto analizado. La dirección es determinada según las clases de ambos objetos, si son iguales se acercan, si son diferentes se alejan.

La teoría de los Conjuntos Aproximados (Rough Sets Theory - RST), también ha sido utilizada con este objetivo. Entre los modelos desarrollados a partir de esta herramienta matemática se encuentran el Edit2RS y el Edit3RS definidos en [17], que permite la creación de nuevos objetos al ser reetiquetados aquellos que pertenezcan al conjunto frontera de cada clase presente en el conjunto de entrenamiento, aunque también forman parte del grupo de modelos de selección de instancias. El primero realiza la selección de aquellas instancias que se encuentran en la aproximación inferior de cada clase, reetiquetando las instancias del conjunto frontera a partir del análisis de los k-vecinos más cercanos para cada instancia, y de esta forma se logra obtener mejores resultados. El tercer modelo, sigue el mismo esquema, pero reetiqueta el conjunto frontera a partir de la función de pertenencia aproximada definida por Grabowski en el 2003.

Hamamoto et al. en 1997 [86] desarrollaron cuatro métodos de *Bootstrap*. En estos métodos los objetos son construidos por la regla de los k vecinos más cercanos de su misma clase de objetos seleccionados al azar. Es tarea del usuario definir el número de iteraciones que se realizarán, de los cuales se selecciona el resultado con menor error.

### 1.2.3. Selección de objetos

Como ya es sabido en el proceso de clasificación supervisada, la fase entrenamiento es de suma importancia. Dicho aprendizaje está orientado por los datos que contiene la base de casos de entrenamiento. Es usual en los problemas de la vida real, que esta base de casos contenga información vana para el proceso de clasificación; entiéndase por esto: casos superfluos, que pueden contener ruido o que pueden ser redundantes [31]. Es por esto que se hace necesario la eliminación de estos casos de la base de conocimiento inicial.

Dado un conjunto de entrenamiento, el objetivo de los métodos de selección de instancias es obtener un subconjunto que no contenga instancias superfluas de forma tal que no se degrade la Exactitud de la Clasificación que se obtenga de usar el subconjunto de instancias resultante de la selección. Estos métodos pueden generar subconjuntos de forma incremental, adicionando instancias a medida que se explora el espacio de conocimiento. Otra alternativa es el modelo decremental, que parte del conjunto de conocimiento inicial y va eliminando instancias hasta encontrar el subconjunto óptimo de acuerdo con el algoritmo utilizado.

A través de la selección de instancias, el conjunto de entrenamiento es reducido, lo cual pudiera ser útil a la hora de reducir el tiempo durante el proceso de aprendizaje, particularmente en los modelos basados en instancias, donde la clasificación de una nueva instancia hace uso de todo el conjunto de entrenamiento.

Al igual que en la selección de rasgos, en dependencia de la estrategia que se use; la selección de instancia se puede dividir en dos grandes grupos [31]:

- *Wrapper*: el criterio de selección está basado en la precisión obtenida por el clasificador, debido a que comúnmente las instancias que no aportan a la precisión del clasificador, son descartados del conjunto resultante.
- Filtros: el criterio de selección usa una función de selección que no es basada en el clasificador.

#### Métodos de envoltura (Wrapper)

La mayoría de los métodos tipo envoltura que se han propuesto están basados en el clasificador K-NN. El CNN (*Condensed Nearest Neighbor*) [87] viene siendo uno de

los más antiguos, este método sigue el modelo incremental y su rutina inicial consiste en incluir de forma aleatoria en el conjunto resultante  $S$ , una instancia perteneciente a cada una de las clases del problema. Seguidamente cada instancia del conjunto original es clasificado usando  $S$  como conjunto de entrenamiento; si la instancia es clasificada incorrectamente, entonces es incluida en el conjunto  $S$ , de esta forma se persigue la idea de que si existiera otra instancia parecida a esta, entonces se clasificaría correctamente. Uno de los inconvenientes de este modelo es que pudiera retener instancias que constituirían ruido dentro del conjunto resultante. Un enfoque incremental basado en instancias lo constituyen los métodos IB2 e IB3, que toma como base el mismo mecanismo desarrollado por el CNN, tomando como valor de  $k = 1$ .

A partir de este método se derivan un conjunto de métodos entre los que se encuentran el SNN (*Selective Nearest Neighbor*) [88], el cual genera el conjunto  $S$  de forma tal que cada instancia del conjunto original esté más cercano a un miembro del conjunto  $S$  que cualquier otro, esto se pudiera entender como que cada instancia sería correctamente clasificada por el clasificador 1-NN, usando como conjunto de entrenamiento a  $S$ . Otra variante dentro de este grupo es el GCNN (*Generalized Condensed Nearest Neighbor*) [89], su funcionamiento es idéntico al del CNN, solo que incluye un criterio de absorción de acuerdo con un umbral. Esto se traduce en que para cada instancia, la absorción se calcula en términos de los vecinos más cercanos y los enemigos más cercanos (aquellas instancias más cercanas pero que pertenecen a otra clase).

Otro de los métodos de selección de instancia es el ENN (*Edited Nearest Neighbor*) [90] que se enfoca en descartar las instancias ruidosas presentes en el conjunto de entrenamiento. Este método descarta aquellas instancias cuando la clase es diferente de la clase mayoritaria de sus  $k$  vecinos más cercanos (generalmente ENN usa  $k = 3$ ). Una extensión del ENN es el RENN (*Repeted ENN*), método que aplica repetidamente ENN hasta que todas las instancias presentes en el conjunto resultante  $S$  sean de la misma clase que la clase mayoritaria a la que pertenecen sus  $k$  vecinos más cercanos. Otra variante es el *all* K-NN [91], este método parte de iterar la rutina del algoritmo K-NN  $k$  veces etiquetando las instancias que son mal clasificadas. Una vez que las iteraciones se detienen, todas las instancias etiquetadas, se descartan del

conjunto de entrenamiento.

Asociados al K-NN se pueden mencionar también los cinco métodos propuestos por Wilson y Martinez en el año 2000 [92]: DROP1, DROP2, DROP3, DROP4 y DROP5 (*Decremental Reduction Optimization Procedure*). Dichos métodos están basados en el criterio de asociación. Las asociaciones de una instancia determinada son aquellas instancias, sobre las cuales dicha instancia forman parte de sus vecinos más cercanos. En dependencia del orden del modelo, sus rutinas varían en el análisis del conjunto inicial y la aplicación simultánea de un modelo con otro. Otro modelo relacionado con el criterio de asociación es el ICF (*Iterative Case Filtering algorithm*) propuesto por Brighton y Melish en el año 2002, el cual está basado en la accesibilidad y cobertura de cada instancia que no son más que el conjunto de vecinos más cercanos y el conjunto de asociaciones respectivamente. De esta forma las instancias son descartadas si la cardinalidad del conjunto de acceso es mayor que la cardinalidad del conjunto de cobertura, y se considera la instancia como ruido dentro del conjunto de entrenamiento [93].

Existen modelos de clasificadores que también se aplican para la selección de instancias, dentro de este grupo se pueden encontrar las Máquinas de Vectores de Soporte (SVM por sus siglas en inglés), que definen como vector de soporte el subconjunto de instancias seleccionadas por el clasificador, en vez de usar el vector como discriminante de clases. Varios trabajos se pueden encontrar en la literatura, tales como el propuesto por Yuanguí et al. [94] y SV-kNNC (*Support Vector k-Nearest Neighbor Clustering*) [95], que combinan este último con elementos de clasificación no supervisada.

La variedad de técnicas aplicadas a esta tarea ha incluido los modelos de Algoritmos Evolutivos (AE) y bioinspirados. Los primeros se basan en la evolución natural, donde la idea principal consiste en una población inicial de cromosomas correspondiendo con el subconjunto de instancias seleccionadas en dependencia de la estrategia que se escoja. A partir de la población inicial y de acuerdo con la función de evaluación, se seleccionan los mejores cromosomas (subconjunto de instancias); con el objetivo de realizar las combinaciones y mutaciones generando nuevos cromosomas. Este proceso se realiza tantas veces como se haya definido el número de generaciones a generar. En la aproximación biológica se pueden encontrar el método CSA (*Clonal*

*Selection Algorithm*) [96], basado en la idea de un sistema de inmunidad artificial. La idea parte de que si existe un patógeno que ataca a un cuerpo, un conjunto de células (antígenos) trabajan en conjunto para defenderlo. Con el objetivo de proveer una respuesta al patógeno, las células memorizan y mutan para adaptarse al atacante, pero del conjunto completo solo un grupo es capaz de contrarrestar al patógeno. Siguiendo esta idea, el modelo identifica los antígenos (instancias) en el cuerpo (conjunto de entrenamiento) de forma aleatoria; una instancia por clase, entonces un patógeno (conjunto de instancias) ataca el sistema seleccionado y repetidamente un conjunto de clones, parecidos al patógeno muta, con la idea de seleccionar los mejores.

Otra forma de encontrar el subconjunto optimal de instancias es con el método BSE (*Backward Sequential Edition*), que usa la búsqueda secuencial regresiva [97]. Este es un método decremental que evalúa cada una de las instancias a través del uso de un clasificador y desecha la instancia que obtenga el menos valor de contribución a la clasificación para el subconjunto parcial de entrenamiento; este proceso se realiza repetidamente hasta que la precisión de la clasificación empiece a decrecer. Una variante de este mismo método lo constituye el RFOS (*Restricted Floating Object Selection*), adaptando la búsqueda secuencial flotante utilizada por el BSE en su rutina de exploración, pero esta vez de forma restricta, debido al costo computacional de la búsqueda secuencial flotante [98].

## Métodos filtro

En un conjunto de entrenamiento, se pueden encontrar instancias que se sitúan tanto en el borde como en el interior de los grupos de clases. Se puede definir como instancia frontera de una clase determinada, si dicha instancia constituye uno de los vecinos más cercanos de otra clase, de igual manera, el caso contrario, constituye una instancia interior. El conocimiento de las clases fronteras de una clase determinada, brinda una información muy valiosa, pues permite la discriminación de las regiones dentro del conjunto de entrenamiento [99]. Muchos de los métodos de filtrado de instancia, se basan en la información de las instancias fronteras para realizar la selección de las instancias.

Son variadas las propuestas en la literatura; Riquelme et al. presentaron en el año 2003 [100] un modelo basado en las instancias fronteras, el método POP (*Pattern by*

*Ordered Projections*), descarta las instancias interiores y selecciona algunas de las instancias fronteras. Este modelo se apoya en el concepto de debilidad, que no es más que el número de veces que aparece una instancia como frontera en una clase. De acuerdo con este concepto, el método descarta las instancias cuyo valor de debilidad sea igual al número de atributos que describen a las instancias del conjunto de entrenamiento. El método POC-NN (*Pair Opposite Class-Nearest Neighbor*) [101] también selecciona las instancias frontera. El proceso de selección en este método calcula la media de las instancias en cada una de las clases. El cálculo se realiza para cada clase, tomando como referencia las instancias de las demás clases.

Existe otro enfoque, esta vez basado en los árboles k-dimensionales (*kd-trees*), propuesto por Narayan et al. [102]. EN este método se construye un árbol binario donde la raíz contiene todas las instancias; para construir un nuevo hijo, se selecciona un pivote, el cual es la característica con la máxima diferencia entre los valores consecutivos ordenados; los nodos izquierdos contienen aquellas instancias cuyos valores para el correspondiente atributo es menos que el valor de diferencia máxima y las demás instancias anteriores, se incluyen en el nodo derecho. Este proceso se repite hasta que no sea posible la división de los nodos. Finalmente se seleccionan las instancias que quedan en los nodos hojas.

Una idea interesante es intentar realizar selección de instancias a través de los métodos de agrupamiento; modelos surgidos de la clasificación no supervisada. Un primer grupo de los mismos, se basa en el agrupamiento de las instancias del conjunto de entrenamiento, a partir de ahí; se seleccionan las instancias centrales de los grupos creados, tales como el GCM (*Generalized-Modified Chang Algorithm*) creado por Mollineda et al. [80] en el 2002 y el NSB (*Nearest Sub-class Classifier*) desarrollado por Venmann et al. en el 2005 [103]. El otro gran grupo luego de agrupar las instancias, selecciona aquellas que pertenecen a la frontera de los grupos creados, como es el caso del OSC (*Object Selection by Clustering*) creado por Olvera-López et al. en el 2007 [104].

En la literatura, algunos métodos consisten en asignar pesos a las instancias y seleccionar aquellas instancias que posean un peso dentro de un rango aceptable. El método WP (*Weighting Prototypes*) propuesto por Paredes y Vidal [105] usa el gradiente descendiente para calcular los pesos de cada instancia, en términos vecinos

más cercanos y enemigos más cercanos; de esta forma aquellas instancias con mayor peso son eliminadas. De igual manera surge el método PSR (*Prototype Selection by Relevance*), el cual calcula la relevancia de cada clase del conjunto de entrenamiento. Esta relevancia se determina en términos de la similaridad promedio, es decir aquellas instancias más similares dentro de las misma clase, y de ellas las más relevantes. El PSR selecciona un porcentaje de las instancias más relevantes por clase y de ellas las más similares pertenecientes a distintas clases.

Dentro de los modelos desarrollados partiendo de la representación en grafos del conjunto de entrenamiento, se encuentra el método de edición basado en un grafo de Gabriel (GGE - por sus siglas en inglés), el cual como su nombre lo sugiere usa un grafo de Gabriel. Este grafo está definido de la siguiente forma. Por cada par de puntos  $(p_i, p_j)$  en el conjunto de referencia  $\{X, \theta\}$ , se construye la esfera diametral, denotada por,  $S(p_i, p_j)$ , es decir, la esfera de manera que  $(p_i, p_j)$  forma el diámetro de  $S(p_i, p_j)$ . Dos puntos  $p_i, p_j$  se dicen ser vecinos Gabriel si  $S(p_i, p_j)$  está vacía, es decir, si no hay otros puntos de  $\{X, \theta\}$  distintos de  $p_i$  y  $p_j$  se encuentran en  $S(p_i, p_j)$ . El gráfico Gabriel se obtiene al unir un par de puntos con un con una arista si son Vecinos Gabriel [106]. A partir de este grafo, el método de GGE, elimina aquellos objetos que cuyos vecinos no sean de la misma clase.

En el 2002 Toussaint hizo uso del grado de proximidad de vecindad relativa para la edición de los vecinos más cercanos. Se crea un grafo de proximidad formado por un conjunto de aristas definidas de la siguiente forma  $Edges = \{(p, q) / \wedge_{p,q} \cap T = \emptyset\}$ , donde  $p$  y  $q$  son vértices,  $T$  el conjunto de entrenamiento y  $\wedge_{p,q}$  la intersección entre las hiper-esferas centradas en  $p$  y  $q$  respectivamente, con radio  $|p - q|$  [107]. A partir de este aporte, surge el método de edición RNGE (*Relative Neighborhood Graph Editing*), que calcula el grafo de proximidad de la vecindad relativa sobre un conjunto de entrenamiento y elimina los objetos mal clasificados por sus vecinos en el grafo [108].

Se reportan en la literatura también el enfoque basado en grafos de máxima similaridad (MSG por sus siglas en inglés), el cual es un grafo dirigido donde cada objeto es conectado a su vecino más similar. Formalmente este grafo se define por un par  $G(T, \theta)$ , donde  $T$  es un conjunto de entrenamiento y  $\theta$  un conjunto de aristas. De esta forma, dos objetos  $x, y \in T$  forman un una arista si  $max_{o \in T} \{sim(x, o)\} =$

$sim(x, y)$ , donde  $sim(x, y)$  es una función de similaridad usualmente definida en términos de la distancia Euclidiana, aunque de forma general se puede usar cualquier similaridad. Basados en las características de este tipo de grafos, se definen dos nuevos métodos de selección de instancias, el MSEditA y el MSEditB, ambos calculan inicialmente el grafo de máxima similaridad sobre el conjunto de entrenamiento  $T$  y luego deciden cuál objeto van a eliminar. El MSEditA elimina un objeto si contiene al menos uno de sus más similares (sucesores), perteneciente a una clase diferente. El MSEditB por su parte elimina un objeto si entre sus objetos más similares, la clase mayoritaria es de una clase diferente [108].

Al igual que en los métodos de construcción de objetos también en la selección de instancias, se hace uso de la teoría de los conjuntos aproximados [109]. Este enfoque se basa en la posibilidad que brinda la teoría de obtener dos aproximaciones importantes a partir de un conjunto de entrenamiento original, Aproximación Inferior y Aproximación Superior, a partir de estas aproximaciones es posible obtener un conjunto de instancias que se encuentran en la frontera es estas aproximaciones. En [17] se definen dos modelos de selección de instancias que permiten reducir la cantidad de elementos del conjunto de entrenamiento original Edit1RS. Este modelo, se basa en el cálculo de las aproximaciones inferiores para cada clase y seleccionando aquellas que pertenecen a esta aproximación, de esta forma se eliminarían aquellas instancias que no pertenecen con certeza a la clase aproximada.

### 1.3. Conclusiones del capítulo

En la literatura está demostrada la importancia que posee la clasificación supervisada en la solución de problemas de la vida real. Para realizar esta tarea se han desarrollado disímiles métodos agrupados en dependencia del enfoque que define su comportamiento. En el presente capítulo se realizó una panorámica de algunos de los métodos de los enfoque pertenecientes al Reconocimiento de Patrones. Otro elemento importante y con gran interés en la comunidad científica, es el pre-procesamiento de los datos con los cuales son entrenados los distintos métodos de clasificación. Hay que tener en cuenta que todos adolecen de al menos alguna deficiencia para trabajar con valores ruidosos, perdidos e irrelevantes, los cuales provocan que su precisión se vea

afectada de una u otra manera. En aras de resolver estas deficiencias, se analizaron distintos enfoques para filtrar y refinar los datos de entrenamiento, y se demuestran los resultados positivos obtenidos por cada uno de ellos. Se destaca la carencia de métodos de pre-procesamiento de objetos para el Clasificador Asociativo Gamma, centro de esta investigación.

---

## Capítulo 2

# Selección de objetos para el clasificador Gamma

En la literatura se puede apreciar la amplia gama de algoritmos de selección de instancias [110][31]. La mayoría de los mismos están enfocados a mejorar el rendimiento del clasificador KNN principalmente los métodos de envoltura, como se puede apreciar en el epígrafe 1.2.3 al igual que para otros clasificadores como el ALVOT [28] y las redes neuronales [111]. Debido a los resultados presentados por el Clasificador Asociativo Gamma, es interesante observar el impacto de la selección de instancias para este clasificador, estos resultados se pueden apreciar en [112]. Persiguiendo este objetivo, se propone un nuevo método para la selección de instancias para el Clasificador Asociativo Gamma, tomando como base su funcionamiento, lo que permite refinar el conjunto de entrenamiento sobre el cual se basa durante la fase de entrenamiento y de clasificación. Esta idea está avalada por el supuesto de que un clasificador mejora su exactitud si el conjunto de entrenamiento es pre-procesado, usando elementos específicos del propio clasificador [113].

En este capítulo se introduce un nuevo modelo para el pre-procesamiento de datos mediante la selección de instancias relevantes. El modelo incorpora elementos de la Teoría extendida de los Conjuntos Aproximados, y de la clasificación de patrones basada en el enfoque asociativo. Se definen en este capítulo las bases para calcular los conjuntos aproximados, haciendo uso de una nueva similitud basada en el funcionamiento del Clasificador Asociativo Gamma. A partir de estos elementos teóricos

y tomando como base el modelo de selección de instancias definido en [17], se logra obtener un nuevo algoritmo de selección de instancias.

## 2.1. Propuesta de los conjuntos aproximados basados en la Gamma similitud

Como se mencionaba previamente en la tesis, el clasificador Gamma constituye uno de los clasificadores supervisados con desempeño competitivo dentro del enfoque asociativo [6]. A diferencia de otros clasificadores de este enfoque, el Gamma es un clasificador basado en instancias, y tiene una estrategia de entrenamiento perezosa. Posee, además, una función de similitud propia para realizar las comparaciones entre los objetos. La presencia de esta similitud constituye un punto de contacto de este clasificador con el enfoque extendido de la Teoría de los Conjuntos Aproximados, que utilizan relaciones de similitud en sus definiciones. A continuación, se explica en detalle el funcionamiento del clasificador Gamma, y posteriormente, se abordan las definiciones asociadas a la propuesta de los conjuntos aproximados haciendo uso de esta similitud.

## 2.2. Funcionamiento del Clasificador Asociativo Gamma

El Clasificador Asociativo Gamma consta de dos fases: entrenamiento y clasificación. La fase de entrenamiento de este clasificador inicia con una transformación de los datos, para su conversión a binario, utilizando para ello el código de Johnson-Möbius modificado [6].

Para poder realizar la transformación, es necesario definir la cantidad de bits en los cuales se va a codificar el valor. Supóngase que se desean codificar los siguientes números (5, 8, 10, 3, 6) con 13 bits. Para esto se resta del total de bits el número que se desea representar; el resultado de esta operación corresponde con la cantidad de bits con valor 0 que serán concatenados al inicio de la secuencia y el resto de los bits tendrían el valor 1, como se muestra a continuación:

- Para el número 5 se tendría 13-5 ceros seguidos de 5 unos.
  
- Para el número 8 se tendría 13-8 ceros seguidos de 8 unos.
  
- Para el número 10 se tendría 13-10 ceros seguidos de 10 unos.
  
- Para el número 6 se tendría 13-6 ceros seguidos de 6 unos.

Tabla 2.1: Codificación Johnson Möbius modificado

Número	Codigo Jonhson Möbius modificado
5	0000000011111
8	0000011111111
10	0001111111111
3	0000000000111
6	0000000111111

Posteriormente, este conjunto de entrenamiento ya transformado, y conocido como conjunto fundamental, es almacenado para su uso en la fase de clasificación.

La fase de entrenamiento del Gamma también incluye la realización de la estimación de varios parámetros. En la tabla 2.2 se muestran los parámetros a estimar para dicho clasificador, así como las propuestas que se han realizado para su cálculo.

Tabla 2.2: Parámetros a estimar para el clasificador Gamma

Parámetros	Descripción	Propuesta
$e_m(j)$	Valor que determina el número de dígitos necesarios para la codificación de los rasgos	$\bigvee_{i=1}^p x_j^i$
$\rho$	Parámetro de paro y valor máximo permitido para el parámetro $\theta$ , esto permite continuar buscando la desambiguación de los patrones cercanos a la frontera; de esta forma cuando $\theta = \rho$ , el clasificador deja de iterar y asignará una clase	$\bigvee_{i=0}^n e_m(j) \bigvee_{i=1}^p x_i^j$
$\rho_0$	Parámetro de pausa que permite realizar una evaluación del patrón a clasificar, con el objetivo de determinar si pertenece o no a la clase desconocida	$\bigwedge_{i=0}^n e_m(j) \bigwedge_{i=1}^p x_i^j$
$w$	Vector de pesos para cada uno de los rasgos que describen el patrón, indicando la importancia de cada rasgo para la clasificación	$w_i = 1, i = 1, 2, \dots, n$
$\theta$	Parámetro que determina cuán diferentes son los valores de cada rasgo como para que el operador gamma los considere similares	$\theta = 0$ (valor inicial)
$d$	Variable a ser evaluada para decidir si el patrón a clasificar, pertenece a la clase desconocida o a alguna de las conocidas	$d = \bigvee c_i$

La fase de clasificación del Gamma comienza transformado en patrón a clasificar de la misma forma en que fueron transformados los patrones del conjunto de entrenamiento. Así, se eliminan las componentes negativas, los números decimales y se obtiene, finalmente, un patrón binario. A continuación se muestra un ejemplo.

**Ejemplo:** Supóngase que el patrón a clasificar es el siguiente:  $(4, 8, 3, -2, 5)$  y los siguientes valores de  $e_m$ :  $\{7, 5, 8, 6\}$  donde  $e_i$  es la cantidad de bits en los cuales se deben codificar cada uno de los valores del patrón.

Luego de transformado, el clasificador Gamma comienza un proceso iterativo que

es la clave de su fase de clasificación. Para ello, se utiliza el operador de similitud generalizado ecuación 1.5). Luego, se calcula el promedio de la similitud generalizada para cada clase. Si se encuentra un máximo único, el proceso termina. Si no, se tienen en cuenta los valores de los parámetros de paro y pausa, así como el valor del parámetro  $\theta$ , en un proceso iterativo. En la figura 2.1 se ilustra dicho proceso.

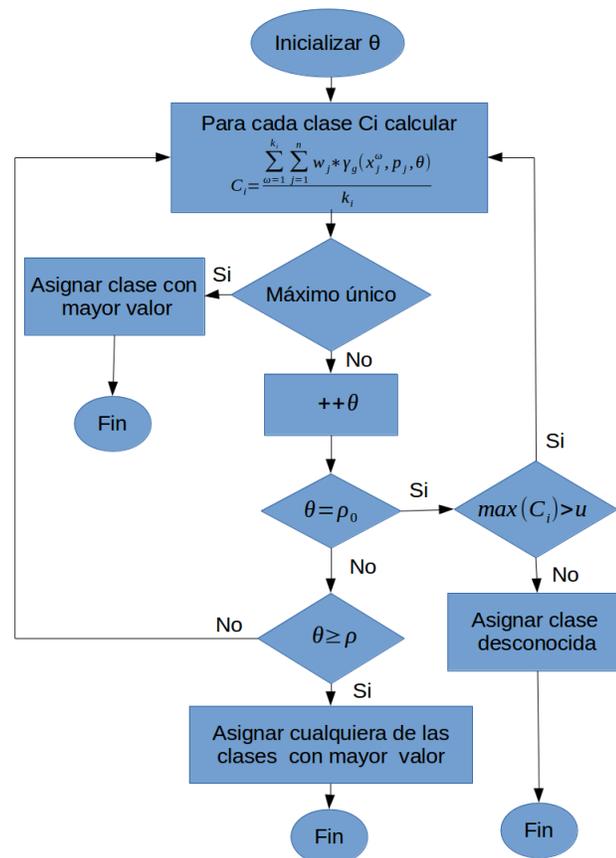


Figura 2.1: Proceso iterativo del clasificador Gamma

Como se aprecia, un elemento crucial en la fase de clasificación del Gamma es la similitud generalizada, que, en última instancia, determina la clasificación de los patrones.

## 2.3. Conjuntos Aproximados basados en la Gamma similitud

La Teoría extendida de los Conjuntos Aproximados se basa en el uso de relaciones de inseparabilidad que no son relaciones de equivalencia [109]. Típicamente, se utilizan relaciones de tolerancia o similitud con el objetivo de manejar valores continuos. Teniendo en cuenta el operador de similitud generalizado del clasificador Gamma, es posible usarlo para definir una relación basada en semejanzas que permita establecer las relaciones entre los objetos. Así, se puede utilizar la información contenida en las aproximaciones de las clases para detectar las instancias relevantes, y poder eliminar aquellas que sean consideradas ruidosas o mal clasificadas. Dado que el clasificador Gamma considera la similitud global de un objeto hacia todas las clases del problema, una definición de aproximaciones basada en vecindad [114][115][116] no se considera adecuada, puesto que no refleja el funcionamiento del clasificador. Por ello, se consideró definir una función de Gamma similitud, y utilizarla en la definición de las aproximaciones superior e inferior del nuevo conjunto Gamma aproximado.

La Gamma similitud propuesta (GBS) utiliza el operador de similitud generalizado, pero considera la desviación estándar de los rasgos en lugar del parámetro  $\theta$ , que toma valores iterativos.

Se decidió utilizar la desviación estándar para sustituir el parámetro  $\theta$  en el cálculo de la similitud, debido a los beneficios que ofrece la desviación estándar como medida de dispersión de los datos. La desviación estándar es una de las medidas más ampliamente utilizadas con este fin [117][118]. Es una medida de variación absoluta que considera la variación real de los datos presente en un conjunto de datos, lo cual ofrece un conocimiento adicional de los datos con los que se trabaja. Así, se considera que si las diferencias entre dos valores son menores que la desviación presente en el conjunto de datos, estos valores (aunque diferentes) son lo suficientemente cercanos como para ser considerados similares.

Sea  $n$  la cantidad de observaciones en un conjunto de datos, y sea  $\tilde{z}$  la media aritmética de dichos valores, la desviación estándar se calcula como:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (z_i - \tilde{z})^2}{n - 1}} \quad (2.1)$$

Sean  $x$  y  $y$  dos instancias, la Gamma similitud entre ellas se calcula como:

$$GBS(x, y) = \sum_{i=1}^p w_i * \gamma_g(x_i, y_i, \sigma_i) \quad (2.2)$$

donde  $p$  es la cantidad de rasgos que describe a las instancias,  $s_i$  es la desviación estándar para el  $i$ -ésimo rasgo,  $x_i$  y  $y_i$  son los dos vectores binarios asociados al  $i$ -ésimo rasgo en las instancias  $x$  y  $y$ , respectivamente y  $w_i$  es el peso asociado al rasgo  $i$ .

Considerando que el operador  $\gamma_g(x_i, y_i, \sigma_i)$  (ecuación 1.5) y el peso  $w_i$  están definidos en  $[0,1]$ , la función queda definida  $0 \leq GBS(x, y) \leq p$ .

$$\gamma_g(x_i, y_i, \sigma_i) = \begin{cases} 1 & m - u_\beta[\alpha(x, y) \bmod 2] \leq \sigma_i \\ 0 & \text{en otro caso} \end{cases} \quad (2.3)$$

**Teorema 1** *La función propuesta es simétrica.*

**Demostración** Por definición  $\gamma_g(x_i, y_i, \sigma_i)$  esta definida como se muestra en la ecuación 2.3. Analizando las definiciones del operador  $u_\beta$  y del operador  $\alpha$  se tiene que la expresión  $m - u_\beta[\alpha(x, y) \bmod 2]$  es equivalente a  $|x_i - y_i|$ , por lo que el operador gamma generalizado quedaría como:

$$\gamma_g(x_i, y_i, \sigma_i) = \begin{cases} 1 & \text{si } (|x_i - y_i|) \leq \sigma \\ 0 & \text{en otro caso} \end{cases} \quad (2.4)$$

De acuerdo con las propiedades del operador módulo, se tiene que  $|x_i - y_i| = |y_i - x_i|$ . Por ello,  $\gamma_g(x_i, y_i, \sigma) = \gamma_g(y_i, x_i, \sigma)$ . Luego queda demostrado que el operador gamma de similitud generalizado es simétrico. Aplicando las propiedades de la suma,  $\sum_{i=1} w_i * \gamma_g(x_i, y_i, \sigma_i) = \sum_{i=1} w_i * \gamma_g(y_i, x_i, \sigma_i)$ , se tiene que la función GBS es simétrica ■

Sea un sistema de decisión  $DS = (U, A \cup C)$ , donde  $U$  es un universo de objetos, descrito por los rasgos presentes en  $A = \{A_1, \dots, A_p\}$ , con un rasgo de decisión  $C$ . Los conceptos asociados a las clases de decisión, para el rasgo  $C$  se denotarán como  $\{C_1, \dots, C_l\}$ .

Se propone una la relación de semejanza  $R$  basada en la función de Gamma similitud (ecuación 2.4), siendo reflexiva dado que cada instancia es similar a sí misma,

siendo  $GBS(x, x) = p$ , donde  $p$  es la cantidad de rasgos que describe a las instancias. Así, se considera que  $xRx$  es siempre verdadero. Dos instancias  $x, y \in U, x \neq y$  se consideran similares de acuerdo con esta relación, si y solo si:

$$xRy \leftrightarrow GBS(x, y) = \max_{z \in U, z \neq x} \{GBS(x, z) > \varepsilon_x\} \quad (2.5)$$

**Definición 1** Sea  $x$  una instancia que pertenece al concepto  $C_i$ , el umbral  $\varepsilon_x$  se calculará como  $\max_{z \in U - C_i} \{GBS(x, z)\}$ .

La clase de semejanza de una instancia  $x \in U$  con respecto a un conjunto de atributos  $B \subseteq A$  en un conjunto Gamma aproximado se denota como  $R_B(x)$  y está formada por el conjunto de objetos que son similares a  $x$  de acuerdo con la relación  $R$  y estará dada por:

$$R_B(x) = \{y \in U : yR_Bx\} \quad (2.6)$$

Teniendo en cuenta que la relación de semejanza propuesta no cumple la propiedad de ser simétrica, se considera la relación inversa de  $R$ , denotada como  $R^{-1}$ , donde  $xR^{-1}y$  significa que “ $y$  es similar a  $x$  de acuerdo con la relación  $R$ ” [119]. Así, la relación inversa  $R^{-1}$  es la clase de las instancias de referencia a las cuales  $x$  es similar:

$$R_B^{-1}(x) = \{y \in U : yR_B^{-1}x\} \quad (2.7)$$

**Teorema 2** La relación de semejanza propuesta NO es simétrica.

**Demostración** Para la demostración de la no simetría de la relación de semejanza propuesta, se utilizará la presentación de un contraejemplo. Sea el siguiente sistema de decisión:

	A1	A2	A3	Clase
O1	3	6	2	A
O2	4	5	1	B
O3	2	5	3	B
O4	3	5	2	A

Las desviaciones estándar para cada rasgo toman los siguientes valores:

	A1	A2	A3
$\sigma_i$	0.82	0.50	0.82

Calculando la función de similitud GBS para cada par de instancias se tiene que:

GBS	O1	O2	O3	O4
O1	$1 + 1 + 1 = 3$	$0 + 0 + 0 = 0$	$0 + 0 + 0 = 0$	$1 + 0 + 1 = 2$
O2	$0 + 0 + 0 = 0$	$1 + 1 + 1 = 3$	$0 + 1 + 0 = 1$	$0 + 1 + 0 = 1$
O3	$0 + 0 + 0 = 0$	$0 + 1 + 0 = 1$	$1 + 1 + 1 = 3$	$0 + 1 + 0 = 1$
O4	$1 + 0 + 1 = 2$	$0 + 1 + 0 = 1$	$0 + 1 + 0 = 1$	$1 + 1 + 1 = 3$

Analizando la relación  $R$  para cada par de instancias, donde por definición que  $xRx$  es siempre verdadero, y  $xRy \leftrightarrow GBS(x, y) = \max_{z \in U, z \neq x} \{GBS(x, z) > \varepsilon_x\}$ , se tiene:

R	O1	O2	O3	O4
O1	Verdadero	Falso	Falso	Verdadero
O2	Falso	Verdadero	Verdadero	<b>Verdadero</b>
O3	Falso	Verdadero	Verdadero	<b>Verdadero</b>
O4	Verdadero	<b>Falso</b>	<b>Falso</b>	Verdadero

Como se aprecia, la relación de semejanza definida no cumple la propiedad de ser simétrica. ■

Para definir las aproximaciones inferior y superior de un concepto (clase)  $C_i$  en el conjunto Gamma aproximado, se analizará la relación que tienen las instancias pertenecientes a este concepto con el resto de las instancias de su clase de semejanza [120]. En este caso, se analizará su Gamma similitud, de acuerdo con el funcionamiento del Clasificador Asociativo Gamma, teniendo en cuenta que la relación de semejanza  $R$  que se establece (ecuación 2.5) está basa en la Gamma similitud.

La aproximación inferior de un concepto  $C_i$  con respecto a un conjunto de atributos  $B \subseteq A$  se denota como  $INF_B(C_i)$  y estará dada por:

$$INF_B(C_i) = \{x \in U : \{R^{-1}(x) - \{x\}\} \neq \emptyset \wedge \subseteq C_i\} \quad (2.8)$$

Considerando una instancia  $x \in C_i$ , se tiene que por definición  $xRy \leftrightarrow GBS(x, y) = \max_{z \in U, z \neq x} \{GBS(x, z) > \varepsilon_x\}$ . Como  $\varepsilon_x = \max_{z \in U - C_i} \{GBS(x, z)\}$ , entonces se tiene que  $xRy \leftrightarrow y \in C_i$ . Así, la definición de aproximación inferior del concepto  $C_i$  puede ser simplificada como:

$$INF_B(C_i) = \{x \in U : \{R^{-1}(x) - \{x\}\} \neq \emptyset\} \quad (2.9)$$

La aproximación superior del concepto  $C_i$  con respecto a un conjunto de atributos  $B \subseteq A$  se denota como  $SUP_B(C_i)$  y estará dada por:

$$SUP_B(C_i) = \{x \in U : R^{-1}(x) \cap C_i \neq \emptyset\} \quad (2.10)$$

La región límite de un concepto  $C_i$  con respecto a un conjunto de atributos  $B \subseteq A$  en un conjunto Gamma aproximado estará dada por:

$$LIM_B(C_i) = SUP_B(C_i) - INF_B(C_i) \quad (2.11)$$

Considérese como ejemplo el sistema de decisión que se muestra en la tabla 2.3, al cual se le ha calculado la desviación estándar de cada rasgo. Se desea obtener las Gamma aproximaciones de los conceptos  $A$  y  $B$ .

Tabla 2.3: Sistema de decisión

Instancia	$A_1$	$A_2$	Clase
$I_0$	1	3	B
$I_1$	6	2	A
$I_2$	7	6	A
$I_3$	6	7	A
$I_4$	2	3	B
$I_5$	6	3	B
$\sigma$	2.7	2	

Se calcula el valor del umbral  $\varepsilon_x$  de cada instancia a partir de la Gamma similitud

entre las instancias del concepto  $A$  con respecto a instancias del concepto  $B$  (tabla 2.4) y viceversa (tabla 2.5):

Tabla 2.4: Similitud de las instancias del concepto  $A$  a las instancias del concepto  $B$

GBS	$I_1$	$I_2$	$I_3$	$\varepsilon_x$
$I_0$	1	0	0	1
$I_4$	1	0	0	1
$I_5$	2	1	1	2

Tabla 2.5: Similitud de las instancias del concepto  $B$  a las instancias del concepto  $A$

GBS	$I_0$	$I_4$	$I_5$	$\varepsilon_x$
$I_1$	1	1	2	2
$I_2$	0	1	2	2
$I_3$	0	0	1	1

Calculando las Gamma similitudes (tabla 2.6) se obtienen las clases de semejanzas de las instancias del ejemplo de la tabla 2.3.

Tabla 2.6: Gamma similitud entre las instancias

GBS	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$xRy$
$I_0$		1	0	0	2	1	$I_4$
$I_1$	1		1	1	1	2	$\emptyset$
$I_2$	0	1		2	0	1	$I_3$
$I_3$	0	1	2		0	1	$I_2$
$I_4$	2	1	0	0		1	$I_0$
$I_5$	1	2	1	1	1		$\emptyset$

Finalmente, analizando la composición de las clases de semejanza, se calculan las aproximaciones inferior y superior de los conceptos tabla 2.7.

Tabla 2.7: Aproximaciones superior e inferior de cada concepto

Concepto	INF	SUP
A	$I_0, I_4$	$I_0, I_4, I_5$
B	$I_2, I_3$	$I_1, I_2, I_3$

Una propiedad importante en la Teoría de los Conjuntos Aproximados basados en relaciones de similitud o semejanza es la que se refiere a la existencia de relaciones de monotonía en las aproximaciones [121]. Si el objeto  $x$  pertenece a la aproximación inferior de un conjunto  $X$  dada la relación de semejanza obtenida sobre un conjunto de atributos  $B$  ( $B \subseteq A$ ), entonces  $x$  pertenece a la aproximación inferior de  $X$  con respecto a cualquier conjunto  $R \subseteq A$  siempre que  $B \subseteq R$ .

Dicho en léxico profesional: si una instancia pertenece a la aproximación inferior con un conjunto de atributos  $B$ , tiene que seguir perteneciendo a la aproximación, si se agregan nuevos atributos.

A continuación se demuestra que los Conjuntos Aproximados, haciendo uso de la similitud propuesta en esta tesis, cumplen con esta propiedad.

**Definición 2** Sea  $A$  el conjunto de rasgos de un sistema de decisión  $DS$ ; y sea  $B \subseteq A$ . El universo de objetos con respecto a  $B$  se denota como  $U_B$  y se define como el conjunto de objetos de  $U$  que contienen únicamente rasgos de  $B$ .

**Definición 3** Sea  $A$  el conjunto de rasgos de un sistema de decisión  $DS$ ; el conjunto de índices asociado al conjunto  $A$  se denota como  $I_A$ , y se define como:  $I_A = \{i \in N : 1 \leq i \leq |A|\}$ . Si  $j \in I_A$ , a  $j$  se le denomina el  $j$ -ésimo índice de  $A$ , y se denota como  $I_A(j)$ ; es decir, se cumple que  $I_A(j) = \{j : \forall j \in I_A\}$ .

En ocasiones, es preciso conocer el conjunto de índices de un subconjunto de  $A$ , y la siguiente definición permite obtener ese conjunto de índices:

**Definición 4** Sea  $A$  el conjunto de rasgos de un sistema de decisión  $DS$ , y sea  $B \subseteq A$ . El conjunto de índices asociado al conjunto  $B$  se denota como  $I_B$ , y se define:  $I_B = \{i \in I_A : A_I A(i) \in B\}$ .

**Lema 1** Sea  $A$  el conjunto de rasgos de un sistema de decisión  $DS$ ; sea  $B \subseteq A$  y sean  $I_A$  e  $I_B$  los conjuntos de índices asociados a los conjuntos  $A$  y  $B$ , respectivamente; entonces se cumple la siguiente contención:  $I_B \subseteq I_A$ .

**Demostración** Caso 1:  $I_B \neq \emptyset$ . Sea un índice cualquiera  $p \in I_B$ ; por definición de  $I_B$ , se tiene que  $p \in \{i \in I_A : A_{I_A(i)} \in B\}$ , es decir,  $p \in I_A$ , por lo que se concluye lo que se quiere demostrar:  $I_B \subseteq I_A$ .

Caso 2:  $I_B = \emptyset$ . Por contradicción. Supóngase que la tesis es falsa; es decir  $I_B \subseteq I_A$  no se cumple. Esto significa que  $\exists p \in I_B$  tal que  $p \notin I_A$ , pero eso es imposible, porque  $I_B = \emptyset$  y, como consecuencia, no posee elementos. Por lo que se concluye que la tesis es verdadera:  $I_B \subseteq I_A$ . ■

**Definición 5** El operador Gamma de similitud generalizado  $\gamma_g$  recibe como entradas dos vectores binarios,  $x(p) \in 0, 1^q$  y  $y(p) \in 0, 1^q$ , correspondientes a las  $p$ -ésimas componentes de las instancias  $x$  y  $y$ , y un valor no negativo  $\sigma$ . Tiene como salida un número binario, y se calcula según la ecuación 2.3.

**Definición 6** Sea  $A$  el conjunto de rasgos de un sistema de decisión  $DS$ , sea  $B \subset A$  y sea  $I_B$  el conjunto de índices asociado al conjunto  $B$ . La función de similitud basada en Gamma (GBS) entre dos instancias  $x, y$  del universo  $U_B$ , se define como  $GBS_B : U \times U \rightarrow [0, |B|]$ , tal que  $GBS_B(x, y) = \sum_{p \in I_B} w * \gamma_g(x(p), y(p), \sigma)$ .

**Lema 2** Sea  $A$  el conjunto de rasgos de un sistema de decisión  $DS$ , sea  $B \subset A$  y sean  $I_A$  e  $I_B$  los conjuntos de índices asociados a los conjuntos  $A$  y  $B$ , respectivamente. Sean dos objetos cualesquiera  $x, y \in U_B$ ; considérese un índice  $p \in I_A$  tal que  $p \notin I_B$ , y constrúyase el conjunto de rasgos  $B' = (B \cup A_p) \subseteq A$  con su conjunto asociado de índices  $I_{B'} = I_B \cup \{p\}$ , de modo tal que se tengan dos nuevos objetos en el universo expandido  $x', y' \in U_{B'}$  que resultan de añadir el rasgo  $A_p$  a los objetos  $x, y$  respectivamente. Entonces se cumple la siguiente desigualdad:  $GBS_{B'}(x', y') \geq GBS_B(x, y)$ .

**Demostración** Por definición, se tiene que  $GBS_{B'}(x', y') = \sum_{j \in I_{B'}} w(j) * \gamma_g(x'(j), y'(j), \sigma)$  y dado que  $I_{B'} = I_B \cup \{p\}$ , la expresión previa se transforma en:  $GBS_{B'}(x', y') = \sum_{j \in I_B \cup \{p\}} w(j) * \gamma_g(x'(j), y'(j), \sigma)$ . Es posible separar el término correspondiente al

índice  $p$  de la sumatoria previa, de manera que esta actúe solamente sobre las componentes de los objetos  $x', y'$  que pertenecen a  $B$ ; es decir, sobre las componentes de los objetos  $x, y \in U_B$ . Al hacer esto, la expresión queda con dos sumandos, de la siguiente manera:  $GBS_{B'}(x', y') = [\sum_{j \in I_B} w(j) * \gamma_g(x(j), y(j), \sigma)] + w(p) * \gamma_g(x(p), y(p), \sigma)$ . De acuerdo con la hipótesis relacionada con los objetos  $x, y \in U_B$  y con la definición dada, el primer sumando de la expresión es  $GBS_B(x, y)$ . Sustituyendo  $\sum_{j \in I_B} \gamma_g(x(j), y(j), \sigma) = GBS_B(x, y)$  en la expresión previa, se tiene:  $GBS_{B'}(x', y') = GBS_B(x, y) + \gamma_g(x(p), y(p), \sigma)$ .

Pero por definición se tiene que:  $0 \leq \gamma_g(x(p), y(p), \sigma) \leq 1$ , por lo que se cumple la siguiente desigualdad:  $GBS_B(x, y) + \gamma_g(x(p), y(p), \sigma) \geq GBS_B(x, y)$ , y por transitividad se tiene lo que se quería demostrar:  $GBS_{B'}(x', y') \geq GBS_B(x, y)$ . ■

**Teorema 3** *Sea  $DS$  un sistema de decisión de un Conjunto Gamma Aproximado con un universo de objetos  $U$  y conjunto de rasgos  $A$ ; sea  $B \subset A$  y sean  $I_A$  e  $I_B$  los conjuntos de índices asociados a los conjuntos  $A$  y  $B$ , respectivamente. Considérese un índice  $p \in I_A$  tal que  $p \notin I_B$ , y constrúyase el conjunto de rasgos  $B' = (B \cup A_p) \subseteq A$  con su conjunto asociado de índices  $I_{B'} = I_B \cup \{p\}$ . Sea  $x \in U_B$  un objeto cualquiera en el universo de objetos con respecto a  $B$ , y sea  $x' \in U_{B'}$  el objeto que resulta de añadir el rasgo  $A_p$  al objeto  $x$ . Entonces se cumple la siguiente condicional:  $x \in INF_B(C_i) \rightarrow x' \in INF_{B'}(C_i)$ .*

**Demostración** Sea  $x \in INF_B(C_i)$ . Por las definiciones de  $INF_B(C_i)$  es posible afirmar que  $x \in C_i$  y que además, se cumple lo siguiente:  $\{x \in C_i : R_B^{-1}(x) - \{x\}\} \neq \emptyset \wedge \subseteq C_i$ . Por definición,  $R_B^{-1}(x) = \{y \in U : xR_B y\}$  y  $xRy \leftrightarrow GBS(x, y) = \max_{z \in U, z \neq x} \{GBS(x, z)\} \geq \varepsilon_x$ . Para  $x \in C_i$ , el umbral  $\varepsilon_x$  por definición es  $\max_{z \in U - C_i} \{GBS(x, z)\}$ . Por tanto,  $xRy \leftrightarrow y \in C_i$ . Sean  $x', y' \in U_{B'}$  los objetos que resultan de añadir el rasgo  $A_p$  a los objetos  $x, y \in U_B$ . Por Lema 1, se cumple la siguiente desigualdad:  $GBS_{B'}(x', y') \geq GBS_B(x, y)$ .

Como  $x \in INF_B(C_i)$ , entonces  $R_B^{-1}(x) \in C_i$ , es decir  $\forall y \in R_B^{-1}(x), GBS_B(x, y) > \varepsilon_x$ . Por transitividad  $\forall y' \in R_{B'}^{-1}(x'), GBS_{B'}(x', y') > \varepsilon_x$ . Lo cual, de acuerdo con las definiciones de  $INF_{B'}(C_i)$ , significa que  $x' \in INF_{B'}(C_i)$ , en el universo de objetos respecto a  $B'$ .

Al considerar las expresiones  $INF_B(C_i)$  y  $INF_{B'}(C_i)$ , es posible establecer que, dado un objeto arbitrario en la aproximación inferior de  $C_i$  en el universo respecto

a  $B$ , el objeto  $x' \in U_{B'}$  que le corresponde a  $x$  en el universo respecto a  $B'$  pertenece la aproximación inferior de  $C_i$  en ese mismo universo:  $x \in INF_B(C_i) \rightarrow x' \in INF_{B'}(C_i)$ .

Es decir, se cumple la propiedad de monotonía de las aproximaciones en el conjunto Gamma aproximado. ■

## 2.4. Algoritmo para la selección de instancias

En esta sección se aborda la propuesta de algoritmos para la selección de instancias basados en la teoría de los Conjuntos Aproximados. Considerando la información presente en las aproximaciones de las clases, es posible identificar aquellas instancias que puedan ser ambiguas. Estos elementos han sido analizados de forma genérica [119], al considerar los supuestos para que una instancia sea declarada ambigua o no. Dado un subconjunto de instancias  $X \subseteq U$  y una relación de semejanza  $R$  en  $U$ , una instancia  $x \subseteq X$  no es ambigua si cumple una de las dos condiciones siguientes:

1.  $x$  pertenece a  $X$  sin ambigüedad, es decir,  $x \in X$  y  $R^{-1}(x) \subseteq U$ ; estas instancias se denominan positivas.
2.  $x$  no pertenece a  $X$  sin ambigüedad, es decir,  $x \in U - X$  y  $R^{-1}(x) \cap X = \emptyset$ ; estas instancias se denominan negativas.

Las instancias que no son positivas ni negativas, se consideran ambiguas. Teniendo en cuenta que las instancias ambiguas representan la mayor dificultad para los algoritmos de clasificación, y en particular pueden afectar negativamente al desempeño del clasificador Gamma, se decidió utilizar un algoritmo de selección que las elimine del conjunto de entrenamiento.

El algoritmo propuesto, denominado GRIS (*Gamma Rough based Instance Selection*) se basa en el modelo propuesto por Camallero en el 2007 [17], por lo que se puede definir como un caso especial del algoritmo Edit1RS y consiste en obtener, para cada una de las clases, las instancias positivas y negativas, y eliminar aquellas consideradas ambiguas. Para ello, se utilizó la relación de semejanza propuesta (ecuación 2.5) que se basa en la función GBS introducida como parte de esta investigación (ecuación 2.3).

Se calculan las aproximaciones inferiores de cada una de las clases de decisión, obteniéndose un conjunto aproximado de instancias a partir de la ecuación 2.9 y se eliminan las instancias que no se encuentren en la región positiva de ninguna de las clases.

El pseudocódigo del algoritmo propuesto se ofrece a continuación.

---

**Algoritmo 1:** Pseudocódigo del algoritmo GRIS

---

**Entrada:** Conjunto de entrenamiento  $T$ , con clases de decisión  $Y$

**Salida** : Conjunto  $S$  de instancias seleccionadas

```

1  $S \leftarrow \emptyset$ ;
2 foreach concepto  $Y_i \in Y$  do
3   |   Calcular aproximación inferior  $E = INF(Y_i)$ , (ecuación 2.9);
4   |   foreach instancia  $x \in E$  do
5   |     |   adicionar a  $S$ ,  $S \leftarrow S \cup \{x\}$ ;
6   |   end
7 end
8 return  $S$ ;

```

---

El modelo genérico Edit1RS sobre el que se basa el algoritmo propuesto hace uso de una relación de semejanza reflexiva para el cálculo de los conjuntos aproximados. Por su parte GRIS se diferencia del modelo base precisamente en que la función requerida para obtener las aproximaciones está basada en la función GBS (ecuación 2.5), ya que está diseñado precisamente para el Clasificador Asociativo Gamma.

## 2.5. Análisis del costo computacional del algoritmo propuesto

El costo computacional de los algoritmos depende en numerosas ocasiones de su implementación. En esta tesis se propone una implementación que se considera eficiente, la cual se detalla a continuación.

Se considera como entrada del algoritmo una lista de instancias que constituyen el conjunto de entrenamiento original, donde  $n$  es la cantidad de instancias y  $p$  es la cantidad de rasgos que describen a estas instancias. Debe tenerse en cuenta además, que se calcule en un único paso el valor de la función GBS de similitud para cada par

de instancias y que dichos valores sean almacenados en forma matricial, donde cada índice esté asociado al orden que ocupan los objetos en la lista de entrenamiento, y que se cuente con un arreglo binario de índices donde se marque si una instancia es seleccionada o no.

1. Calcular y almacenar los valores de la función de semejanza GBS propuesta para cada par de instancias.

- a) Crear y llenar un arreglo para el almacenamiento de las desviaciones estándar de cada rasgo (costo de almacenamiento:  $n$ , costo de ejecución:  $n$ )
- b) Crear la matriz: (costo de almacenamiento:  $n^2$ , costo de ejecución: 1)
- c) Para cada par de instancias, calcular y almacenar la semejanza GBS (costo de ejecución del cálculo de la función:  $p$ , costo de ejecución para cada par:  $n^2$ , costo de ejecución de guardar en la matriz: 1)

Costo total del paso 1: Costo de almacenamiento:  $n^2 + n$ , y Costo de ejecución:  $n + 1 + p * n^2 + 1 = 2 + n + p * n^2$

2. Calcular y almacenar las instancias más similares para cada instancia, y marcar las que no pertenecen a la región positiva.

- a) Crear un arreglo booleano para marcar las instancias que serán eliminadas (costo de almacenamiento:  $n$ , costo de ejecución: 1)
- b) Crear una lista para el almacenamiento de las instancias más similares de una instancia  $x$  (costo de almacenamiento:  $n$ , costo de ejecución: 1)
- c) Hallar las instancias más similares a  $x$ , verificar si son o no de su clase y guardarlas (costo de ejecución de encontrar las instancias:  $n$ , costo de ejecución de guardarlas en la lista: 1, costo de ejecución de ver si coincide la clase: 1, costo de almacenamiento de la bandera de verificación: 1)
- d) Marcar la instancia  $x$  para ser eliminada, en caso de que tenga instancias similares que no sean de su clase (costo de ejecución de la verificación: 1, costo de marcar a  $x$  de ser necesario en el arreglo: 1)

Costo total del paso 2: Costo de almacenamiento:  $n + n + 1 = 2n + 1$ , y  
Costo de ejecución:  $1 + n(n + 1) = 1 + n + n^2$ .

3. Copiar las instancias no marcadas a una nueva lista que se devuelve.

a) Crear la nueva lista donde se devolverán las instancias (costo de almacenamiento:  $n$ , costo de ejecución: 1)

b) Para cada instancia si no está marcada copiarla a la nueva lista y devolver la lista (costo ejecución de verificar si la instancia está marcada: 1, costo ejecución de copiar: 1, y finalmente costo de ejecución de devolver la lista: 1)

Costo total del paso 3: Costo de almacenamiento:  $n$ , y Costo de ejecución:  $1 + n(1 + 1) = 2n + 1$ .

Costo computacional total del algoritmo propuesto:

Costo de almacenamiento total:  $n^2 + n + 2n + 1 + n = n^2 + 4n + 1$ . Siendo el término mayor  $n^2$ , se considera que la cota del costo de almacenamiento está dada por  $O(n^2)$ .

Costo de ejecución total:  $2 + n + p * n^2 + 1 + n + n^2 + 2n + 1 = (p + 1) * n^2 + 4n + 4$ .

Considerando que  $p < n$ , el término mayor de la expresión es  $n^2$ , y se considera que la cota del costo de ejecución está dada por  $O(n^2)$ .

Por ello, se considera que la propuesta realizada en esta investigación es tratable desde el punto de vista computacional.

## 2.6. Conclusiones del capítulo

Las propuestas teóricas realizadas en este capítulo permiten arribar a las siguientes conclusiones parciales:

1. Los nuevos conjuntos aproximados se fundamentan en la hibridación de la Teoría extendida de los Conjuntos Aproximados y del Clasificador Asociativo Gamma, y constituyen una nueva herramienta teórica que puede ser utilizada en el Reconocimiento de Patrones.

2. La función de similitud GBS constituye un elemento esencial en los nuevos conjuntos aproximados, y su funcionamiento se inspira en el operador de similitud del Clasificador Asociativo Gamma.
3. El algoritmo GRIS para la selección de instancias relevantes es tratable desde el punto de vista computacional, y puede ser aplicado para el pre-procesamiento de conjuntos de entrenamiento en el Clasificador Asociativo Gamma.

---

## Capítulo 3

# Resultados experimentales

En este capítulo se describen los experimentos numéricos realizados para la evaluación del desempeño del modelo propuesto para la selección de instancias. Se detallan los bancos de datos utilizados, así como el protocolo de experimentación. Finalmente, se ofrecen los test estadísticos que sustentan las comparaciones numéricas realizadas.

### 3.1. Descripción de los bancos de datos de prueba

Teniendo en cuenta que el Clasificador Asociativo Gamma, que se desea mejorar en esta investigación solamente maneja datos numéricos y sin ausencias de información, se seleccionaron 15 bancos de datos del Repositorio de Aprendizaje Automático de la Universidad de California en Irvine [122], para la realización de los experimentos.

Todos los bancos de datos utilizados se corresponden con problemas reales, es decir, los datos fueron obtenidos de diversos escenarios de la vida real. No se utilizaron bancos de datos sintéticos creados *ex profeso* para fines de experimentación. La elección de bancos de datos reales permite modelar mejor el comportamiento del algoritmo, y de cierta forma evaluar cómo será su comportamiento ante situaciones de clasificación de la vida real. Sin embargo, como establecen los teoremas de “*No Free Lunch*” [123], no es posible diseñar un algoritmo que alcance los mejores resultados, para todas las medidas de calidad, en todos los bancos de datos posibles.

A pesar de ello, se considera que la realización de pruebas con datos reales puede ser beneficiosa. En la tabla 3.1 se describen los bancos de datos utilizados.

Tabla 3.1: Descripción de los bancos de datos utilizados en los experimentos.

Bancos de datos	Instancias	Rasgos	Clases	IR	Balanceada
balance-scale	625	4	3	5.87755102	
diabetes	768	8	2	1.865671642	
ecoli	336	7	8	71.5	
hayes-roth	160	4	3	2.096774194	
heart-statlog	270	13	2	1.25	x
ionosphere	351	34	2	1.785714286	
iris	150	4	3	1	x
liver-disorders	345	6	2	1.379310345	x
mfeat-morphological	2000	6	10	1	x
new-thyroid	215	5	3	5	
page-blocks	5473	10	5	175.4642857	
pendigits	10992	16	10	1.08436019	x
spambase	4601	57	2	1.537782681	
vehicle	846	18	4	1.095477387	x
wine	154	13	3	1.395348837	x

La cantidad de instancias en los bancos de datos utilizados varía entre 150 y 10992 instancias, y la cantidad de rasgos varía entre 4 y 57 rasgos. Además, las clases a predecir están en el rango de 2 a 10 clases. Otro criterio tenido en cuenta fue el Índice de Desbalance (*Inbalance Ratio* - IR) presente en cada una de ellas, por lo que se asumieron como bases de datos desbalanceadas aquellas que poseen un valor de IR superior a 1.5 por ser el adoptado en el sitio oficial de la plataforma KEEL (*Knowledge Extraction based in Evolutionary Learning*).

## 3.2. División de los conjuntos de datos

Para la obtención de los conjuntos de entrenamiento y prueba, en esta investigación se procedió a dividir cada uno de los bancos de datos de acuerdo con un

procedimiento estratificado de validación cruzada en 10 hojas.

La validación cruzada es un procedimiento de muestreo ampliamente utilizado en la evaluación de algoritmos de clasificación [124]. El procedimiento consiste en dividir el conjunto de datos en  $k$  hojas o *folde*s (en este caso se utilizaron 10) del mismo tamaño.

Luego de dividido el conjunto de datos, se entrena el algoritmo de clasificación con  $k - 1$  hojas, y se utiliza la restante como conjunto de prueba. Este procedimiento se repite  $k$  veces. Así, cada una de las  $k$  hojas es utilizada una vez como conjunto de prueba. Este procedimiento se ilustra en la figura 3.1.

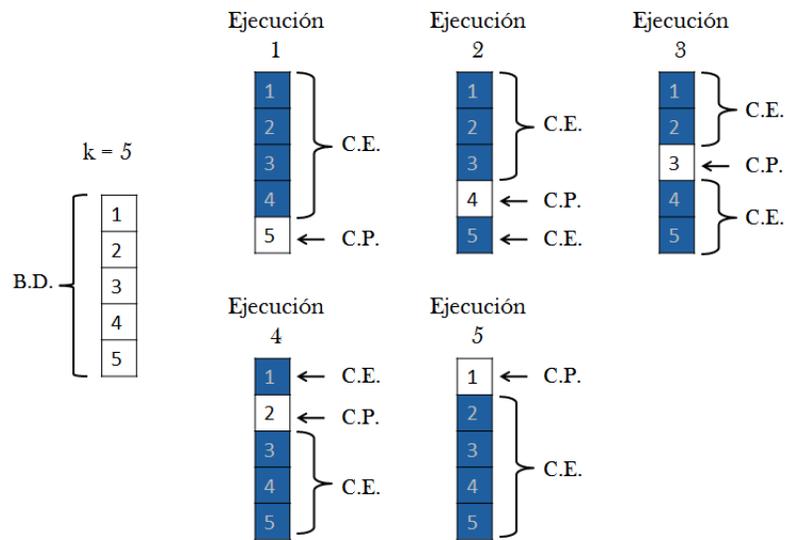


Figura 3.1: Procedimiento de validación cruzada con 5 hojas. C.E. representa las hojas que formarán parte del conjunto de entrenamiento y C.P. las que constituirán el conjunto de prueba.

Usualmente, el proceso de división en hojas se realiza de manera estratificada, garantizando que cada clase esté equitativamente representada en cada una de las hojas. En este caso, la selección de las instancias que formarán parte de cada hoja se realiza por clases. Este procedimiento se ilustra en la figura 3.2.

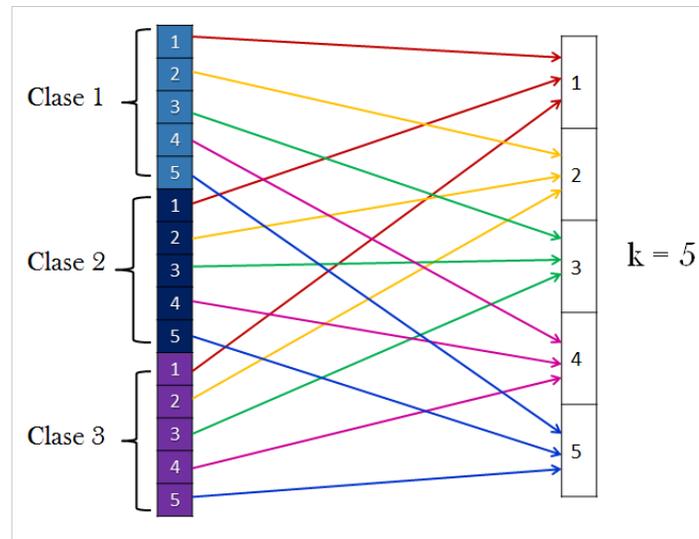


Figura 3.2: Selección de instancias en un proceso de validación cruzada estratificada de 5 hojas.

### 3.3. Algoritmos utilizados

Para la realización de las comparaciones experimentales, se analizó el comportamiento del Clasificador Asociativo Gamma en dos escenarios:

1. Considerando que todos los rasgos tienen la misma importancia, siendo por ende su peso  $w_i = 1$  para todo rasgo  $A_i \in A$ .
2. Considerando que los rasgos no tienen la misma importancia, y estimando los valores de sus pesos automáticamente.

En el segundo caso, se implementó el algoritmo GammaED propuesto en [6], que utiliza la metaheurística de Evolución Diferencial para obtener los pesos de los rasgos en el clasificador Gamma. En la tabla 3.2 se muestran los valores utilizados para la ejecución de este algoritmo. Los valores de estos parámetros se definieron de igual manera que la utilizada en [6].

Tabla 3.2: Descripción de los parámetros utilizados en el algoritmo GammaED.

<b>Parámetro</b>	<b>Valor</b>
Tamaño de la población	10
Número de iteraciones	50
Codificación	Real
Probabilidad de cruce	1
Estrategia de selección	Elitista

En cada uno de los dos escenarios, se calculó la Exactitud del clasificador Gamma utilizando el conjunto de entrenamiento original (sin seleccionar instancias). Este valor fue considerado como línea base en el análisis del comportamiento de los algoritmos de selección de instancias.

Como métodos de selección de instancias, se evaluaron: el algoritmo GRIS, propuesto en esta investigación, y los algoritmos ENN [90], GGE , RNGE [107] y MSEditB [108]. Como se aprecia, todos estos algoritmos son de edición basada en el error de clasificación, por lo que se consideran como de propósito similar. Así, es posible comparar al algoritmo propuesto con otros de su tipo.

Los algoritmos GGE y RNGE fueron hallados entre los mejores del tipo edición basada en el error en estudios experimentales realizados recientemente [125]. De igual manera, el algoritmo ENN se consideró como altamente competitivo a pesar de ser el primero de este tipo reportado en la literatura, en fecha tan temprana como 1972 [90]. También el MSEditB ha tenido buenos resultados experimentales en estudios realizados con métodos híbridos [108].

En todos estos algoritmos, con el objetivo de compararlos en igualdad de condiciones y disminuir la posible influencia de la función de similitud utilizada, se utilizó la propuesta GBS como función de semejanza entre instancias. En cuanto a los parámetros de los algoritmos comparados, el único que presenta parámetros es el ENN, que necesita de la definición de la cantidad de vecinos a considerar. En este caso, se escogió un valor de  $k = 3$ , por ser este el utilizado en las experimentaciones de [125].

## 3.4. Medidas de desempeño y test estadísticos

### 3.4.1. Medidas de desempeño utilizadas

Para la realización de las comparaciones experimentales, se tuvieron en cuenta varias medidas de desempeño:

1. Exactitud de la Clasificación obtenida al utilizar como conjunto de entrenamiento el conjunto de instancias seleccionadas por los diferentes algoritmos para las bases de datos balanceadas.
2. Promedio de la Exactitud por clase obtenida al utilizar como conjunto de entrenamiento el conjunto de instancias seleccionadas por los diferentes algoritmos para las bases de datos desbalanceadas.
3. Reducción en la cardinalidad del conjunto de entrenamiento original al aplicar los métodos de selección de instancias.

La Exactitud de la Clasificación se midió considerando la cantidad de instancias del conjunto de prueba que fueron correctamente clasificadas, con respecto al total de instancias de dicho conjunto. Formalmente, sea  $X$  el conjunto de instancias de prueba, y sea  $d(x)$  la clase real de la instancia  $x \in X$ , y  $g(x)$  la clase asignada por el clasificador Gamma. La Exactitud de la Clasificación se calcula como:

$$Acc = \frac{|\{x \in X : d(x) = g(x)\}|}{|X|} * 100 \quad (3.1)$$

El Promedio de la Exactitud por clases (Avg\_Acc), se midió teniendo en cuenta el promedio del número de instancias por cada clase correctamente clasificadas con respecto a la cantidad de instancias de dicha clase. Formalmente, sea  $K = \{K_1, \dots, K_i\}$  el conjunto de clases del conjunto de entrenamiento, el Promedio de la Exactitud por clases se calcula según la siguiente expresión:

$$Avg\_Acc = \frac{1}{|K|} \sum_{i=1}^{|K|} \frac{|\{x \in K_i : d(x) = g(x)\}|}{|K_i|} \quad (3.2)$$

Por otra parte, la reducción en el conjunto de entrenamiento se midió el porcentaje de instancias conservadas en dicho conjunto, por los algoritmos de selección.

Formalmente, sea  $T$  el conjunto de entrenamiento, y sea  $E$  el conjunto de instancias seleccionadas, con  $E \subseteq T$ . La retención de objetos (*Object Retention* - OR) se calcula como:

$$OR = \frac{|E|}{T} * 100 \quad (3.3)$$

Así, mientras menor sea la retención de objetos de un algoritmo, mejor se considerará su desempeño.

Teniendo en cuenta que el desempeño de los algoritmos se analiza con respecto a dos objetivos, la comparación entre métodos se lleva a cabo siguiendo el criterio de dominancia de Pareto [126].

La dominancia de Pareto se utiliza para comparar soluciones en problemas multi-objetivos [127][128][129]. Se considera que una solución  $a$  supera a otra solución  $b$ , si y solo si, existe al menos un objetivo en que  $a$  es mejor que  $b$ , y no existe ningún objetivo en que  $b$  sea superior a  $a$ . De lo contrario, cuando una de las soluciones es mejor en un objetivo, pero peor en otro, se dice que ambas soluciones son Pareto óptimas, y que constituyen soluciones de compromiso.

En la figura 3.3 se muestra un ejemplo de soluciones definidas por los dos objetivos analizados en esta investigación (Exactitud y retención), y se señalan aquellas que son dominadas (rombos) y Pareto óptimas (cuadrados).

### 3.4.2. Test estadísticos

Para la determinación de la existencia o no de diferencias significativas en el desempeño de los algoritmos, es necesario realizar pruebas estadísticas. La comparación estadística del desempeño de algoritmos sobre diversos conjuntos de datos es un caso particular de la comparación estadística de muestras relacionadas. Como no es posible garantizar que los resultados del desempeño de los algoritmos sigan una distribución normal, es necesario aplicar pruebas no paramétricas.

Para la comparación de múltiples muestras relacionadas, se recomienda utilizar el test de Friedman [130]. Este test es no paramétrico y determina si existen o no diferencias en el conjunto de muestras, aunque, en caso de detectar las diferencias, no establece entre qué pares de algoritmos se aprecian. Es por ello que, una vez

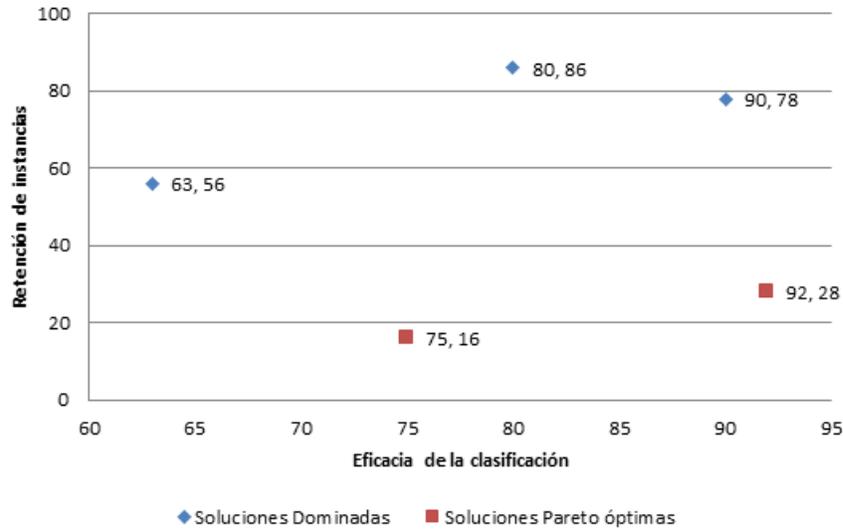


Figura 3.3: Ejemplo de soluciones. Las mejores soluciones son aquellas que tienen una elevada Exactitud (eje X) y poca retención de objetos (eje Y). Nótese como las soluciones (92,28) y (75,16) son no dominadas entre sí, por lo que se consideran Pareto óptimas.

detectada la existencia de diferencias significativas, se recomienda aplicar un test post-hoc, para establecer directamente entre qué algoritmos están las diferencias.

El test de Friedman es un test no paramétrico para evaluar las diferencias de medias entre más de dos muestras relacionadas. En el caso de la comparación de múltiples algoritmos en múltiples bancos de datos, como se requiere en esta investigación, este test realiza un *ranking* de cada uno de los algoritmos para cada banco de datos por separado. El mejor algoritmo recibe rango 1, el segundo mejor rango 2, y así sucesivamente. En caso de empates, se asigna el promedio de los rangos.

Sea  $r_i^j$  el rango obtenido por el  $j$ -ésimo algoritmo en el  $i$ -ésimo banco de datos, y sea  $k$  la cantidad de algoritmos a comparar, y  $N$  la cantidad de bancos de datos en que se aplicaron. El test de Friedman va a comparar el promedio de los rangos de los algoritmos, calculado como  $R_j = \frac{1}{N}(\sum_i r_i^j)$ . Si se considera la hipótesis nula (no existen diferencias entre las medias de los algoritmos), entonces el estadígrafo de Friedman sigue una distribución de  $X_F^2$  con  $k - 1$  grados de libertad, si se cumple que  $N > 10$  y  $k > 5$  [130] donde:

$$X_F^2 = \frac{12N}{k(k-1)} \left[ \sum_j R_j^2 - \frac{k(k-1)}{4} \right] \quad (3.4)$$

Si la hipótesis nula es rechazada (típicamente para un valor de probabilidad menor que 0.05), entonces es necesario realizar un test post-hoc para encontrar entre qué clasificadores están las diferencias.

La utilidad de estos procedimientos post-hoc esta basado en que son capaces de calcular el P-Valor Ajustado (PVA), a partir de un estimador de distribución normal  $z$  (ecuación 3.5) considerando la familia de hipótesis completa para cada pareja de algoritmos comparados [131]. Entre los más utilizados se pueden citar los test de Bonferroni, Holm y Hochberg.

$$z = (R_i - R_j) / \sqrt{\frac{k(k+1)}{6n}} \quad (3.5)$$

donde  $R_i$  es el ranking promedio según Friedman para el clasificador  $i$ ,  $k$  es el número de clasificadores y  $n$  el total de bases de datos [130].

En esta investigación, se aplicó el test de Friedman teniendo en cuenta la Exactitud de la Clasificación (Accuracy) para las bases de datos balanceadas y el Promedio de la Exactitud por clases (Avg\_Acc) para las bases de datos desbalanceadas; también se tuvo en cuenta el análisis de la retención de objetos, para determinar la existencia o no de diferencias significativas entre los algoritmos de selección de instancias. En este caso, se utilizó un nivel de significación de 0.05, para un 95 % de confianza. Luego para aquellas pruebas donde existió diferencias estadísticamente significativas, se procedió a aplicar el test de post-hoc de Holms a partir de un algoritmo de control.

### 3.5. Resultados considerando clasificador Gamma con rasgos igualmente ponderados

El primer experimento fue llevado a cabo considerando que todos los rasgos eran igualmente importantes para el proceso de clasificación. Así, se consideró que todos los pesos de los rasgos serán tomados como de valor uno, en el cálculo de la Gamma similitud.

En la tabla 3.3 se muestran los resultados de la Exactitud de clasificación (bases de datos balanceadas) obtenida a partir del preprocesamiento con los distintos algoritmos de selección y en la tabla 3.4 se muestran los resultados del Promedio de la Exactitud (bases de datos desbalanceadas), de igual forma en la tabla 3.5, se muestra la Retención de Objetos obtenida por cada algoritmo de selección. Se resaltan en negritas los mejores resultados para cada banco de datos.

Tabla 3.3: Exactitud de la Clasificación obtenida por los diferentes algoritmos, sin considerar pesos en los rasgos.

Bancos de datos	Gamma	ENN	GGE	RNGE	MSEditB	GRIS
heart-statlog	81.85	81.85	82.59	82.22	<b>82.96</b>	<b>82.96</b>
iris	88.67	<b>90.00</b>	<b>90.00</b>	<b>90.00</b>	<b>90.00</b>	<b>90.00</b>
liver-disorders	<b>57.70</b>	55.06	56.55	54.50	55.39	57.69
mfeat-morphological	46.00	43.50	<b>46.65</b>	41.85	41.65	42.35
pendigits	64.46	64.48	64.47	64.48	<b>64.55</b>	<b>64.55</b>
vehicle	<b>59.59</b>	58.64	56.15	57.82	58.05	56.75
wine	72.45	71.86	70.29	<b>73.01</b>	72.45	72.48
Cantidad de veces mejor	2	1	2	2	<b>3</b>	<b>3</b>

Tabla 3.4: Promedio de la Exactitud de la Clasificación obtenida por los diferentes algoritmos, sin considerar pesos en los rasgos.

Bancos de datos	Gamma	ENN	GGE	RNGE	MSEditB	GRIS
balance-scale	62.99	63.94	62.24	<b>65.56</b>	65.23	65.33
diabetes	60.97	<b>62.07</b>	61.20	62.00	61.50	61.00
ecoli	48.24	48.11	45.94	48.95	47.88	<b>51.64</b>
hayes-roth	71.39	68.41	63.81	65.17	<b>74.10</b>	69.37
ionosphere	<b>76.84</b>	50.00	50.00	*	*	<b>76.84</b>
new-thyroid	81.85	79.54	82.11	82.27	84.01	<b>85.02</b>
page-blocks	<b>68.29</b>	64.35	68.28	64.29	62.80	60.61
spambase	63.70	62.58	<b>68.81</b>	67.06	62.29	62.58
Cantidad de veces mejor	2	1	1	1	1	<b>3</b>

\* Estos métodos eliminaron todas las instancias del conjunto de entrenamiento.

Como se puede observar en los resultados, el comportamiento del Clasificador Asociativo Gamma es similar en varias de las bases de casos, incluso luego de haber sido preprocesado el conjunto de entrenamiento por varios algoritmos de selección de instancias. Teniendo en cuenta las medidas de evaluación analizadas para cada tipo de base de datos (balanceadas y desbalanceadas), se puede apreciar que el algoritmo propuesto, ha sido superior en varias de las bases de datos analizadas con respecto al clasificador Gamma sin haberse realizado la selección de instancias.

En el caso del banco de datos *ionosphere*, los algoritmos RNGE y MSEditB mostraron un comportamiento patológico, dado que eliminan todas las instancias del conjunto de entrenamiento.

Tabla 3.5: Retención de objetos obtenida por los diferentes algoritmos, sin considerar pesos en los rasgos.

<b>Bancos de datos</b>	<b>Gamma</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>	<b>GRIS</b>
balance-scale	100.00	91.24	84.71	89.48	77.74	<b>58.79</b>
diabetes	100.00	84.66	66.94	77.04	66.90	<b>61.73</b>
ecoli	100.00	84.42	88.78	75.39	<b>67.48</b>	74.67
hayes-roth	100.00	86.60	64.65	27.78	64.51	<b>46.04</b>
heart-statlog	100.00	88.72	77.24	85.19	76.30	<b>70.41</b>
ionosphere	100.00	64.10	<b>35.90</b>	*	*	100.00
iris	100.00	95.48	97.26	94.30	93.41	<b>92.44</b>
liver-disorders	100.00	83.80	53.66	73.95	58.00	<b>48.89</b>
mfeat-morphological	100.00	82.58	93.59	71.51	63.86	<b>60.36</b>
new-thyroid	100.00	98.24	96.69	98.04	95.66	<b>94.16</b>
page-blocks	100.00	97.86	95.83	96.32	95.77	<b>94.60</b>
pendigits	100.00	99.70	99.71	99.53	99.19	<b>98.92</b>
spambase	100.00	95.24	<b>85.61</b>	86.98	90.20	86.57
vehicle	100.00	83.77	85.07	77.74	67.52	<b>62.21</b>
wine	100.00	98.60	<b>88.91</b>	98.85	95.10	92.39
Cantidad de veces mejor	0	0	3	0	1	<b>11</b>

\* Estos métodos eliminaron todas las instancias del conjunto de entrenamiento.

En cuanto a la retención de objetos, los resultados muestran que el algoritmo propuesto fue el más reductor en 11 de los 15 bancos de datos. Sin embargo, en el banco de datos *ionosphere*, no eliminó ninguna instancia, a diferencia de los algoritmos RNGE y MSEditB, que las eliminaron todas. Este comportamiento se debe, a juicio del autor, a que las estrategias de vecindad utilizadas por estos algoritmos no

son adecuadas para este banco de datos.

Para establecer la existencia o no de diferencias estadísticamente significativas en el desempeño de del Clasificador Asociativo Gamma, sobre bases de datos preprocesada con el modelo GRIS, con respecto al desempeño del clasificador sobre bases de datos preprocesadas por los otros algoritmos, se realizó el test de Friedman, con un nivel de significación de 0.05, para un 95 % de confianza. Para el banco de datos *ionosphere*, donde los algoritmos RNGE y MSEditB no obtuvieron resultados, se tomaron como valores de desempeño los correspondientes al conjunto de entrenamiento original. En el caso de la Exactitud de la Clasificación (*Accuracy*) sobre bases de datos balanceadas y del Promedio de la exactitud por clase sobre las bases de datos desbalanceadas (*Avg\_Acc*), este test devolvió un valor de probabilidad superior a 0.05, por lo cual no se rechaza la hipótesis nula. De esta forma, se considera que en cuando a Exactitud del clasificador y Promedio de exactitud por clase, no existen diferencias significativas entre los algoritmos comparados, incluyendo al clasificador usando el conjunto de entrenamiento original como se puede observar en las tablas 3.6 y 3.7.

Tabla 3.6: Average Rankings of the algorithms (Accuracy - Gamma)

Algorithm	Ranking
GRIS	2.714285714285714
MSEditB	3.214285714285714
Gamma	3.571428571428571
RNGE	3.785714285714286
ENN	3.8571428571428568
GGE	3.857142857142857

Friedman statistic considering reduction performance (distributed according to chi-square with 5 degrees of freedom: 2.0816326530612344.

P-value computed by Friedman Test: 0.8377382263643209.

Tabla 3.7: Average Rankings of the algorithms (Avg\_Acc - Gamma)

Algorithm	Ranking
RNGE	2.6875
GRIS	3.125
Gamma	3.4375
MSEditB	3.4375
ENN	4.0
GGE	4.3125

Friedman statistic considering reduction performance (distributed according to chi-square with 5 degrees of freedom: 3.9285714285714284.

P-value computed by Friedman Test: 0.5597451598709065.

De igual forma se aplicó el mismo test teniendo en cuenta la retención de objetos, donde se obtuvo un valor de probabilidad inferior a 0.05, por lo que se acepta la hipótesis nula y se considera que existen diferencias estadísticamente significativas. En la tabla 3.8 se muestran estos resultados.

Tabla 3.8: Average Rankings of the algorithms (OR - Gamma)

Algorithm	Ranking
GRIS	1.4999999999999998
MSEditB	2.4333333333333333
GGE	3.2666666666666675
RNGE	3.5000000000000004
ENN	4.4
Gamma	5.9000000000000001

Friedman statistic considering reduction performance (distributed according to chi-square with 5 degrees of freedom: 50.409523809523925.

P-value computed by Friedman Test: 1.1871116312178742E-9.

A partir de este resultado se procedió a analizar el test de Holm (tabla 3.9) para

determinar la superioridad del algoritmo GRIS con respecto a los demás algoritmos de selección analizados. El resultado de este test evidenció que existen diferencias estadísticamente significativas en la comparación realizada, dado que el *p-value* obtenido es inferior al valor  $\alpha$  ajustado para cada caso.

Tabla 3.9: Holm / Hochberg Table for  $\alpha = 0,05$  (OR)

$i$	algorithm	$z = (R_0 - R_i)/SE$	$p$	Holm/Hochberg/Hommel
5	Gamma	6.440940481460321	1.187354316414293E-10	0.01
4	ENN	4.24516531732612	2.184323166727037E-5	0.0125
3	RNGE	2.927700218845601	0.0034147911781178394	0.016666666666666666
2	GGE	2.5861351933136145	0.009705885293027716	0.025
1	MSEditB	1.3662601021279466	0.17185733906279932	0.05

Bonferroni-Dunn's procedure rejects those hypotheses that have a p-value  $\leq 0,01$ .

Holm's procedure rejects those hypotheses that have a p-value  $\leq 0,05$ .

Hochberg's procedure rejects those hypotheses that have a p-value  $\leq 0,025$ .

Hommel's procedure rejects those hypotheses that have a p-value  $\leq 0,05$ .

Estos resultados permiten afirmar que el método propuesto supera al resto de los algoritmos comparados en cuanto a retención de objetos, debido a que se encontraron diferencias estadísticamente significativas a su favor en todos los casos. Analizando el criterio de dominancia de Pareto, se considera que el algoritmo propuesto GRIS domina a los algoritmos comparados, puesto que tiene mejores resultados en cuanto a retención de objetos y resultados similares en cuanto a eficacia de la clasificación.

### 3.6. Resultados considerando clasificador Gamma con pesos obtenidos de forma automática

En esta sección, se compararon los resultados obtenidos por los algoritmos de selección de instancias, pero considerando que los rasgos no están igualmente ponderados. Para determinar los pesos adecuados para cada rasgo, se utilizó el algoritmo GammaED [6] que se basa en el uso de la meta-heurística de Evolución Diferencial [132] para la obtención de los pesos de forma automatizada.

En las tablas 3.10, 3.11 y 3.12 se muestran los resultados obtenidos en cuanto a Exactitud de la Clasificación, el Promedio de la Exactitud de la Clasificación y retención de objetos, respectivamente. Se resaltan en negritas los mejores resultados.

Tabla 3.10: Exactitud de la Clasificación obtenida por los diferentes algoritmos, considerando los pesos en los rasgos.

<b>Bancos de datos</b>	<b>Gamma</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>	<b>GRIS</b>
heart-statlog	83.70	83.70	83.33	84.07	<b>84.44</b>	84.07
iris	<b>93.33</b>	<b>93.33</b>	<b>93.33</b>	<b>93.33</b>	92.67	92.67
liver-disorders	<b>58.54</b>	55.34	56.50	56.51	56.21	56.55
mfeat-morphological	46.15	43.10	<b>46.25</b>	43.85	43.90	44.75
pendigits	60.81	60.86	60.76	60.84	60.93	<b>65.04</b>
vehicle	<b>58.41</b>	57.94	55.08	56.51	56.40	55.45
wine	67.45	66.90	65.75	66.90	68.01	<b>69.15</b>
Cantidad de veces mejor	<b>3</b>	1	2	1	1	2

Tabla 3.11: Promedio de la Exactitud de la Clasificación obtenida por los diferentes algoritmos, considerando los pesos en los rasgos.

<b>Bancos de datos</b>	<b>Gamma</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>	<b>GRIS</b>
balance-scale	53.84	63.94	62.24	<b>65.56</b>	65.23	65.33
diabetes	60.18	<b>62.07</b>	61.20	62.00	61.50	61.00
ecoli	39.15	48.11	45.94	48.95	47.88	<b>51.64</b>
hayes-roth	58.70	68.41	63.81	65.17	<b>74.10</b>	69.37
ionosphere	76.19	50.00	50.00	*	*	<b>76.84</b>
new-thyroid	77.79	79.54	82.11	82.27	84.01	<b>85.02</b>
page-blocks	67.51	64.35	<b>68.28</b>	64.29	62.80	60.61
spambase	65.04	62.58	<b>68.81</b>	67.06	62.29	62.58
Cantidad de veces mejor	0	1	2	1	1	<b>3</b>

\* Estos métodos eliminaron todas las instancias del conjunto de entrenamiento.

Como se aprecia, en este experimento, al igual que en el experimento anterior, los algoritmos RNGE y MSEditB, eliminaron todas las instancias de entrenamiento en

el banco de datos ionosphere. Igualmente, en estos casos, para la realización de los test estadísticos se consideraron los valores de Exactitud de clasificación, Promedio de la Exactitud de clasificación y retención de objetos correspondientes al conjunto original.

Tabla 3.12: Retención de objetos obtenida por los diferentes algoritmos, considerando pesos en los rasgos.

<b>Bancos de datos</b>	<b>Gamma</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>	<b>GRIS</b>
balance-scale	100.00	91.24	84.71	89.48	77.74	<b>16.50</b>
diabetes	100.00	84.66	66.94	77.04	66.90	<b>16.88</b>
ecoli	100.00	84.42	88.78	75.39	<b>67.48</b>	74.67
hayes-roth	100.00	86.60	64.65	<b>27.78</b>	64.51	46.04
heart-statlog	100.00	88.72	77.24	85.19	76.30	<b>70.41</b>
ionosphere	100.00	64.10	<b>35.90</b>	*	*	67.84
iris	100.00	95.48	97.26	94.30	93.41	<b>92.44</b>
liver-disorders	100.00	83.80	53.66	73.95	58.00	<b>48.89</b>
mfeat-morphological	100.00	82.58	93.59	71.51	63.86	<b>60.36</b>
new-thyroid	100.00	98.24	96.69	98.04	95.66	<b>94.16</b>
page-blocks	100.00	97.86	95.83	96.32	95.77	<b>94.60</b>
pendigits	100.00	99.70	99.71	99.53	99.19	<b>51.27</b>
spambase	100.00	95.24	<b>85.61</b>	86.98	90.20	86.57
vehicle	100.00	83.77	85.07	77.74	67.52	<b>62.21</b>
wine	100.00	98.60	88.91	98.85	95.10	<b>88.46</b>
Cantidad de veces mejor	0	0	2	1	1	<b>11</b>

\* Estos métodos eliminaron todas las instancias del conjunto de entrenamiento.

En cuanto a la retención de objetos, el algoritmo propuesto superó al resto en 11 de los 15 bancos de datos. Las mayores reducciones se alcanzaron en los bancos de datos balance-scale y diabetes, donde el método GRIS solamente seleccionó el 16 % de las instancias. Es de destacar que, en estos bancos de datos, el GRIS además obtuvo los mejores valores de eficacia del clasificador.

Para establecer la existencia o no de diferencias significativas en el desempeño de los algoritmos, nuevamente se realizó el test de Friedman, con un nivel de significación

de 0.05, para un 95 % de confianza. En la tabla 3.13 se ofrecen los resultados teniendo en cuenta la Exactitud de la Clasificación y en la tabla 3.14 se muestran los resultados del análisis a partir del Promedio de la Exactitud por clases.

Tabla 3.13: Average Rankings of the algorithms (Accuracy-Gamma)

Algorithm	Ranking
Gamma	2.714285714285714
GRIS	2.8571428571428568
MSEditB	3.3571428571428568
RNGE	3.5
ENN	4.071428571428571
GGE	4.5

Friedman statistic considering reduction performance (distributed according to chi-square with 5 degrees of freedom: 4.755102040816325.

P-value computed by Friedman Test: 0.4464928920455401.

Tabla 3.14: Average Rankings of the algorithms (Avg\_acc-Gamma)

Algorithm	Ranking
RNGE	2.625
GRIS	2.8125
MSEditB	3.375
ENN	3.625
GGE	3.8125
Gamma	4.75

Friedman statistic considering reduction performance (distributed according to chi-square with 5 degrees of freedom: 6.696428571428571.

P-value computed by Friedman Test: 0.24421375754882935.

El análisis de la Retención de Objetos, teniendo en cuenta los pesos, se puede apreciar en la tabla 3.15, y se observa que existen diferencias estadísticamente significativas, dado que el valor obtenido por el test es inferior a 0.05, lo que conllevó a

analizar el test de Holm para probar la superioridad del algoritmo propuesto con respecto a los demás algoritmos de selección de instancias. Como se puede apreciar en la tabla 3.16, para cada comparación se obtuvo un *p-value* inferior al valor de  $\alpha$  ajustado por el test.

Tabla 3.15: Average Rankings of the algorithms (OR)

Algorithm	Ranking
GRIS	1.3333333333333333
MSEditB	2.4666666666666667
GGE	3.3333333333333334
RNGE	3.5333333333333337
ENN	4.4
Gamma	5.9333333333333345

Friedman statistic considering reduction performance (distributed according to chi-square with 5 degrees of freedom: 53.666666666666674.

P-value computed by Friedman Test: 2.7834734517284687E-10.

Tabla 3.16: Holm / Hochberg Table for  $\alpha = 0,05$

$i$	algorithm	$z = (R_0 - R_i)/SE$	$p$	Holm/Hochberg/Hommel
5	Gamma	6.733710503344882	1.653900925968585E-11	0.01
4	ENN	4.4891403355632535	7.151117679110054E-6	0.0125
3	RNGE	3.22047024073016	0.001279804793030722	0.016666666666666666
2	GGE	2.927700218845601	0.0034147911781178394	0.025
1	MSEditB	1.6590301240125067	0.09710972225903586	0.05

Bonferroni-Dunn's procedure rejects those hypotheses that have a p-value  $\leq 0,01$ .

Holm's procedure rejects those hypotheses that have a p-value  $\leq 0,05$ .

Hochberg's procedure rejects those hypotheses that have a p-value  $\leq 0,025$ .

Hommel's procedure rejects those hypotheses that have a p-value  $\leq 0,05$ .

Estos resultados permiten afirmar que el método propuesto supera al resto de los algoritmos comparados en cuanto a retención de objetos, dado a que se encontraron diferencias estadísticamente significativas a su favor en todos los casos. Analizando

el criterio de dominancia de Pareto, se considera que el algoritmo propuesto GRIS domina a los algoritmos comparados, puesto que tiene mejores resultados en cuanto a retención de objetos y resultados similares en cuanto a eficacia de la clasificación.

### 3.7. Resultados considerando otros clasificadores de aprendizaje perezoso

En esta sección, se compararon los resultados obtenidos por los algoritmos de selección de instancias, pero considerando otros clasificadores de aprendizaje perezoso. Se analizaron los desempeños de los clasificadores del vecino más cercano (1-NN) [26] y ALVOT [28]. No se consideraron pesos en los atributos, para el cálculo de la función de similitud GBS, que rige el comportamiento del algoritmo propuesto. Si se tuvieron en cuenta las medidas de evaluación para las bases de datos balanceadas (Exactitud de clasificación) y desbalanceadas (Promedio de la Exactitud de la Clasificación por clases).

En las tablas 3.17 y 3.18 se muestran los resultados obtenidos en cuanto a la Exactitud de la Clasificación y el Promedio de la Exactitud de la Clasificación para los clasificador 1-NN respectivamente. Se ofrecen los resultados sin selección de instancias (1-NN) y luego de la selección de instancias con los algoritmos evaluados (ENN, GGE, RNGE, MSEditB y GRIS). Se resaltan en negritas los mejores resultados.

Tabla 3.17: Exactitud de la Clasificación, considerando el clasificador del vecino más cercano.

<b>Bancos de datos</b>	<b>1-NN</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>	<b>GRIS</b>
heart-statlog	75.19	78.89	73.33	<b>80.74</b>	76.30	78.89
iris	95.33	95.33	<b>96.00</b>	<b>96.00</b>	95.33	95.33
liver-disorders	62.86	60.55	54.79	62.87	62.87	<b>64.03</b>
mfeat-morphological	65.80	70.10	67.00	<b>71.20</b>	71.00	70.80
pendigits	<b>99.35</b>	99.32	99.33	99.34	99.30	99.29
vehicle	69.63	67.26	<b>71.53</b>	68.69	67.74	67.98
wine	96.08	96.08	94.93	<b>96.63</b>	96.08	96.08
Cantidad de veces mejor	1	0	2	<b>4</b>	0	1

Tabla 3.18: Promedio de la Exactitud de la Clasificación, considerando el clasificador del vecino más cercano.

<b>Bancos de datos</b>	<b>1-NN</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>	<b>GRIS</b>
balance-scale	56.81	59.63	56.23	60.17	62.16	<b>65.11</b>
diabetes	65.56	68.96	70.49	72.17	71.96	<b>72.32</b>
ecoli	63.75	69.32	69.98	68.39	69.06	<b>75.96</b>
hayes-roth	<b>71.82</b>	61.37	53.39	53.54	65.63	68.35
ionosphere	50.00	50.00	50.00	*	*	<b>84.39</b>
new-thyroid	95.74	89.75	<b>96.59</b>	92.34	92.25	91.42
page-blocks	80.21	77.07	<b>84.00</b>	75.52	72.84	73.11
spambase	<b>90.32</b>	89.54	88.53	89.70	89.60	89.76
Cantidad de veces mejor	2	0	2	0	0	<b>4</b>

\* Estos métodos eliminaron todas las instancias del conjunto de entrenamiento.

En las tabla 3.19 y 3.20 se presentan igualmente los resultados de la Exactitud de la Clasificación y del Promedio de la Exactitud de la Clasificación por clase para el clasificador ALVOT respectivamente, de igual forma, se exponen primeramente los resultados de la aplicación del clasificador (ALVTO) sobre los datos originales y luego de ser procesados por los distintos algoritmos de selección de instancias.

Tabla 3.19: Exactitud de la Clasificación, considerando el clasificador ALVOT.

Bancos de datos	ALVOT	ENN	GGE	RNGE	MSEditB	GRIS
heart-statlog	63.77	<b>80.00</b>	78.89	<b>80.00</b>	79.63	79.63
iris	94.67	<b>95.33</b>	94.00	94.67	<b>95.33</b>	94.67
liver-disorders	53.65	53.66	<b>55.73</b>	49.59	50.43	53.89
mfeat-morphological	70.20	69.80	<b>70.80</b>	69.20	69.40	69.50
pendigits	76.04	76.07	76.06	76.06	<b>76.14</b>	76.12
vehicle	47.99	47.28	47.51	47.64	47.75	<b>48.11</b>
wine	96.60	97.19	97.16	<b>97.75</b>	97.19	97.19
Cantidad de veces mejor	0	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	1

Tabla 3.20: Promedio de la Exactitud de la Clasificación, considerando el clasificador ALVOT.

Bancos de datos	ALVOT	ENN	GGE	RNGE	MSEditB	GRIS
balance-scale	66.31	68.87	<b>72.35</b>	63.55	63.68	64.04
diabetes	<b>69.92</b>	65.55	65.72	64.38	64.84	64.51
ecoli	58.37	57.95	51.74	64.10	65.21	<b>68.68</b>
hayes-roth	<b>68.45</b>	59.73	49.15	44.05	64.24	59.87
ionosphere	80.81	50.00	50.00	*	*	<b>83.19</b>
new-thyroid	<b>82.66</b>	76.72	76.72	75.88	76.72	76.72
page-blocks	60.32	60.69	<b>61.01</b>	60.95	59.88	60.23
spambase	<b>83.48</b>	81.89	78.98	83.21	82.46	82.14
Cantidad de veces mejor	<b>4</b>	0	2	0	0	2

Como se aprecia, en el desempeño del clasificador del vecino más cercano, el algoritmo propuesto obtuvo resultados similares, teniendo en cuenta las dos medidas de evaluación del clasificador. Igualmente, para el clasificador ALVOT.

No se realizó un análisis estadístico de la reducción obtenida en estos casos, pues esta se aprecia en la tabla 3.5, donde queda mostrada la significativa superioridad del algoritmo GRIS con respecto a otros de su tipo, en cuanto a reducción de instancias, sin considerar pesos. Sí se aplicó el test de Friedman para establecer o no la

existencia de diferencias significativas en el desempeño de los algoritmos en cuanto a la Exactitud de la Clasificación y al Promedio de la Exactitud de la Clasificación.

En la tabla 3.21 se muestran los resultados del test de Friedman para el clasificador 1-NN, en la tabla 3.22 se pueden observar el mismo test sobre los valores del Promedio de la Exactitud.

Tabla 3.21: Average Rankings of the algorithms - (Accuracy-NN)

Algorithm	Ranking
RNGE	1.7142857142857142
GRIS	3.5
1-NN	3.714285714285714
MSEditB	3.785714285714286
GGE	4.071428571428571
ENN	4.2142857142857135

Friedman statistic considering reduction performance (distributed according to chi-square with 5 degrees of freedom: 8.306122448979579.

P-value computed by Friedman Test: 0.14015209880755652.

Tabla 3.22: Average Rankings of the algorithms - (Avg-acc-NN)

Algorithm	Ranking
GRIS	2.25
1-NN	3.5
GGE	3.5625
RNGE	3.75
MSEditB	3.875
ENN	4.0625

Friedman statistic considering reduction performance (distributed according to chi-square with 5 degrees of freedom: 4.767857142857142.

P-value computed by Friedman Test: 0.4448630579871087.

A continuación la tabla 3.23 se muestran los resultados luego de realizar el test de

Friedman para el clasificador ALVOT, en la tabla 3.22 se pueden observar el mismo test sobre los valores del Promedio de la Exactitud.

Tabla 3.23: Average Rankings of the algorithms - (Accuracy - ALVOT)

Algorithm	Ranking
MSEditB	2.7857142857142856
GRIS	2.8571428571428568
ENN	3.2857142857142856
ALVOT	3.7142857142857144
RNGE	4.142857142857143
GGE	4.2142857142857135

Friedman statistic considering reduction performance (distributed according to chi-square with 5 degrees of freedom: 3.8775510204081343.

P-value computed by Friedman Test: 0.5671770246893313.

Tabla 3.24: Average Rankings of the algorithms - (Avg-acc - ALVOT)

Algorithm	Ranking
ALVOT	2.25
GRIS	3.3125
MSEditB	3.5625
GGE	3.75
ENN	3.875
RNGE	4.25

Friedman statistic considering reduction performance (distributed according to chi-square with 5 degrees of freedom: 5.410714285714286.

P-value computed by Friedman Test: 0.36783573421607363.

A partir del análisis de los resultados, se puede asegurar que no existen diferencias significativas en la Exactitud de la Clasificación obtenida por los clasificadores 1-NN y ALVOT sobre las bases de datos originales y las bases de datos preprocesadas con los distintos algoritmos de selección de instancias. Este resultado puede apreciarse también teniendo en cuenta el Promedio de la Exactitud de la Clasificación por clases.

Como se aprecia, para los clasificadores 1-NN y ALVOT, el algoritmo propuesto GRIS obtiene resultados similares que con los algoritmos analizados (ENN, GGE y MSEditB), se observa un mejor resultado sobre las bases de datos desbalanceadas con el clasificador 1-NN, donde se logra mejorar la eficacia del clasificador. Estos resultados avalan la calidad de la propuesta realizada, y muestran que es similar a otros de su tipo y en algunos casos es superior utilizando el criterio de análisis multiobjetivo de Pareto tanto para la medida de evaluación Exactitud de la Clasificación (Tabla 3.25) y el Promedio de la clasificación (Tabla 3.26)

Tabla 3.25: Comparación de GRIS respecto a otros algoritmos, usando el criterio de análisis multiobjetivo de Pareto, a partir de la Exactitud de la Clasificación.

<b>Clasificador Gamma sin pesos</b>					
<b>GRIS vs.</b>	<b>Gamma</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>
Objetivo1: Exactitud	Igual	Igual	Igual	Igual	Igual
Objetivo2: Retención de objetos	Mejor	Mejor	Mejor	Mejor	Mejor
Decisión de Pareto: GRIS es	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>
<b>Clasificador Gamma con pesos</b>					
<b>GRIS vs.</b>	<b>Gamma</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>
Objetivo1: Exactitud	Igual	Igual	Igual	Igual	Igual
Objetivo2: Retención de objetos	Mejor	Mejor	Mejor	Mejor	Mejor
Decisión de Pareto: GRIS es	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>
<b>Clasificador 1-NN</b>					
<b>GRIS vs.</b>	<b>1-NN</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>
Objetivo1: Exactitud	Igual	Igual	Igual	Igual	Igual
Objetivo2: Retención de objetos	Mejor	Mejor	Mejor	Mejor	Mejor
Decisión de Pareto: GRIS es	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Igual</b>	<b>Mejor</b>
<b>Clasificador ALVOT</b>					
<b>GRIS vs.</b>	<b>ALVOT</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>
Objetivo1: Exactitud	Igual	Igual	Igual	Igual	Igual
Objetivo2: Retención de objetos	Mejor	Mejor	Mejor	Mejor	Mejor
Decisión de Pareto: GRIS es	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>

Tabla 3.26: Comparación de GRIS respecto a otros algoritmos, usando el criterio de análisis multiobjetivo de Pareto, a partir del Promedio de la clasificación.

<b>Clasificador Gamma sin pesos</b>					
<b>GRIS vs.</b>	<b>Gamma</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>
Objetivo1: Promedio de la Exactitud	Igual	Igual	Igual	Igual	Igual
Objetivo2: Retención de objetos	Mejor	Mejor	Mejor	Mejor	Mejor
Decisión de Pareto: GRIS es	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>
<b>Clasificador Gamma con pesos</b>					
<b>GRIS vs.</b>	<b>Gamma</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>
Objetivo1: Promedio de la Exactitud	Igual	Igual	Igual	Igual	Igual
Objetivo2: Retención de objetos	Mejor	Mejor	Mejor	Mejor	Mejor
Decisión de Pareto: GRIS es	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>
<b>Clasificador 1-NN</b>					
<b>GRIS vs.</b>	<b>1-NN</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>
Objetivo1: Promedio de la Exactitud	Igual	Igual	Igual	Igual	Igual
Objetivo2: Retención de objetos	Mejor	Mejor	Mejor	Mejor	Mejor
Decisión de Pareto: GRIS es	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>
<b>Clasificador ALVOT</b>					
<b>GRIS vs.</b>	<b>ALVOT</b>	<b>ENN</b>	<b>GGE</b>	<b>RNGE</b>	<b>MSEditB</b>
Objetivo1: Promedio de la Exactitud	Igual	Igual	Igual	Igual	Igual
Objetivo2: Retención de objetos	Mejor	Mejor	Mejor	Mejor	Mejor
Decisión de Pareto: GRIS es	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>	<b>Mejor</b>

Así, se puede afirmar que el uso de la teoría extendida de los conjuntos aproximados en la selección de instancias, mejora la efectividad del Clasificador Asociativo Gamma, y además la de otros clasificadores perezosos, como el vecino más cercano y ALVOT en distintos tipos de bases de datos.

### 3.8. Comparación del los resultados con el modelo Edit3RS

Dado que la propuesta presentada en esta tesis está basada en el modelo Edit1RS propuesta en [17], la misma autora ha propuesto varias mejoras al modelo, de esta forma surge el Edit3RS. Como se había hecho referencia anteriormente, este modelo incluye una estrategia de reetiquetado de instancias, lo que permite cambiarle el valor del atributo de decisión al conjunto de instancias que se encuentran en el conjunto frontera del sistema de decisión. Esto arroja como resultado una menor reducción del conjunto de entrenamiento, pero se beneficia la Exactitud de la Clasificación.

A pesar de que este enfoque no forma parte de la estrategia seguida en esta investigación, se hace conveniente realizar una comparación del comportamiento de este modelo con la propuesta de esta tesis. Con este objetivo se muestran los resultados obtenidos para la Exactitud de la Clasificación sin pesos y con pesos en los rasgos (Tablas 3.27 y 3.29) de igual forma se muestran los resultados para el Promedio de la Exactitud de la Clasificación (Tablas 3.28 y 3.30).

Tabla 3.27: Exactitud de la Clasificación obtenida a partir de la selección con los algoritmos GRIS y Edit3RS, sin considerar el peso de los rasgos.

<b>Bancos de datos</b>	<b>GRIS</b>	<b>Edit3RS</b>
heart-statlog	<b>82.96</b>	81.85
iris	<b>90.00</b>	59.99
liver-disorders	57.69	<b>58.78</b>
mfeat-morphological	<b>42.35</b>	28.85
pendigits	64.55	<b>65.17</b>
vehicle	<b>56.75</b>	50.13
wine	72.48	<b>74.15</b>
Cant. de veces mejor	<b>4</b>	3

Tabla 3.28: Promedio de la Exactitud de la Clasificación obtenida a partir de la selección con los algoritmos GRIS y Edit3RS, sin considerar el peso de los rasgos.

<b>Bancos de datos</b>	<b>GRIS</b>	<b>Edit3RS</b>
balance-scale	<b>65.33</b>	62.41
diabetes	<b>61.00</b>	55.08
ecoli	<b>51.64</b>	44.27
hayes-roth	69.37	<b>75.33</b>
ionosphere	76.84	<b>77.93</b>
new-thyroid	<b>85.02</b>	67.19
page-blocks	<b>60.61</b>	58.82
spambase	62.58	<b>65.94</b>
Cant. de veces mejor	<b>5</b>	3

Tabla 3.29: Exactitud de la Clasificación obtenida a partir de la selección con los algoritmos GRIS y Edit3RS, considerando el peso de los rasgos.

<b>Bancos de datos</b>	<b>GRIS</b>	<b>Edit3RS</b>
heart-statlog	<b>84.07</b>	81.85
iris	<b>92.67</b>	59.99
liver-disorders	56.55	<b>58.78</b>
mfeat-morphological	<b>44.75</b>	28.85
pendigits	65.04	<b>65.17</b>
vehicle	<b>55.45</b>	50.13
wine	69.15	<b>74.15</b>
Cant. de veces mejor	<b>4</b>	3

Tabla 3.30: Promedio de la Exactitud de la Clasificación obtenida a partir de la selección con los algoritmos GRIS y Edit3RS, considerando el peso de los rasgos.

<b>Bancos de datos</b>	<b>GRIS</b>	<b>Edit3RS</b>
balance-scale	<b>65.33</b>	62.41
diabetes	<b>61.00</b>	55.08
ecoli	<b>51.64</b>	44.27
hayes-roth	69.37	<b>75.33</b>
ionosphere	76.84	<b>77.93</b>
new-thyroid	<b>85.02</b>	67.19
page-blocks	<b>60.61</b>	58.82
spambase	62.58	<b>65.94</b>
Cant. de veces mejor	<b>5</b>	3

A partir de estos valores se procedió a realizar el test de Friedman para determinar si existen diferencias estadísticamente significativas. Los resultados obtenidos se muestran en las siguientes tablas:

Tabla 3.31: Comparación sin considerar pesos en los rasgos

Algorithm	Accuracy	Avg_Accuracy
GRIS	1.4285714285714284	1.375
Edit3RS	1.5714285714285712	1.625
P-value	0.7054569861114506	0.7054569861114506

Tabla 3.32: Comparación considerando pesos en los rasgos

Algorithm	Accuracy	Avg_Accuracy
GRIS	1.4285714285714284	1.375
Edit3RS	1.5714285714285712	1.625
P-value	0.7054569861114506	0.4795001221882962

Como se puede apreciar, los resultados son similares en cuanto a la precisión de la clasificación, aunque se muestra un resultado relativamente superior por parte del algoritmo GRIS, a consideración del autor, este comportamiento está debido a la forma en la que el clasificador determina los objetos similares, incluso ajustando

aquellos que pueden provocar una ambigüedad en la clasificación. De esta forma se refuerza la hipótesis de que al usar métodos específicos para un clasificador, ofrece mejores resultados que al usar métodos pensados para otros clasificadores.

### **3.9. Conclusiones del capítulo**

Las experimentaciones numéricas realizadas en este capítulo permiten arribar a las siguientes conclusiones parciales:

1. El algoritmo GRIS propuesto tiene un desempeño competitivo, y contribuye al mejoramiento del Clasificador Asociativo Gamma.
2. El uso o no de ponderaciones de los rasgos no influye en el comportamiento del algoritmo propuesto, ya que los resultados fueron similares en ambos escenarios.
3. El algoritmo GRIS propuesto supera a otros del estado del arte, siendo significativamente superior a estos en cuanto a retención de objetos, y similar en cuanto a Exactitud de la Clasificación.
4. El algoritmo GRIS propuesto mejora también el desempeño de otros clasificadores perezosos del estado del arte, como 1-NN y ALVOT.
5. El algoritmo GRIS al ser diseñado específicamente para el Clasificador Asociativo Gamma, presenta mejores resultados que otros métodos de su mismo tipo, al estar estos enfocados a otros clasificadores.

---

# Conclusiones

A partir de los resultados obtenidos durante el desarrollo de esta investigación, se pudo constatar la influencia de varios algoritmos de selección de instancias. Esta influencia se evidencia con un incremento en la precisión de la clasificación del clasificador asociativo Gamma, por varios de ellos sobre distintas bases de casos, lo que evidenció la viabilidad del uso de un modelo de selección de instancias ajustado a este clasificador.

Se propuso una novedosa relación de semejanza, así como una función de similitud entre instancias, basada en el funcionamiento del Clasificador Asociativo Gamma. Esto permitió la definición de las aproximaciones inferiores y superiores dentro de la teoría extendida de los conjuntos aproximados ajustados.

Se hizo uso de un modelo basado en la teoría de los conjuntos aproximados, el cual fue ajustado para el mejoramiento del Clasificador Asociativo Gamma de forma tal que se ajustara a su similitud, haciendo que la selección no fuera genérica, sino que estuviera enfocada en el propio clasificador.

La selección de instancias utilizada es competitiva con respecto a otros del estado del arte, lo cual fue corroborado mediante experimentos numéricos sobre bancos de datos internacionales. Los test estadísticos realizados confirman la validez de la propuesta realizada, y su superioridad, en cuanto a retención de instancias y precisión de la clasificación obtenida por el clasificador asociativo Gamma. Así mismo, se corroboró la utilidad de este modelo en el pre-procesamiento de otros clasificadores de aprendizaje perezoso, como NN y ALVOT.

---

## Recomendaciones

- Realizar estudios experimentales con los conjuntos aproximados definidos por la relación de similitud basada en la Gamma similitud, en los modelos de selección de rasgos para el clasificador asociativo Gamma.

---

# Bibliografía

- [1] J. Shulcloper, A. Guzmán, and J.F. Martínez. *Enfoque Lógico Combinatorio al Reconocimiento de Patrones I, Selección de Variables y Clasificación Supervisada*. IPN, 1999.
- [2] G. Dougherty. *Pattern Recognition and Classification. An Introduction*. Springer, 2013.
- [3] J. P. Marques de Sá. *Pattern Recognition. Concepts, Methods and Applications*. Springer, 2001.
- [4] J. A. Carrasco and J. F. Martínez. Reconocimiento de patrones. *Komputer Sapiens*, 2(5), 2011.
- [5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.
- [6] Antonio Ramirez, Itzama Lopez, Yenny Villuendas, and Cornelio Yanez. Evolutionary improvement of parameters in an associative classifier. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 13:1550–1555, 2015.
- [7] A. V. Uriarte-Arcia, I. López Yáñez, C. Yáñez Márquez, J. Gama, and O. Camacho-Nieto. Data stream classification based on the gamma classifier. *Mathematical Problems in Engineering*, 2015.
- [8] C. López-Martin, I. López Yáñez, and C. Yáñez Márquez. Application of gamma classifier to development effort prediction of software projects. *Appl. Math*, 6(3):411–418, 2012.

- 
- [9] M. Aldape-Pérez, C. Yáñez Márquez, and & Ferreira-Santiago A. Camacho-Nieto, O. Feature selection using associative memory paradigm and parallel computing. *Computación y Sistemas*, 17(1):41–52, 2013.
- [10] I. López-Yáñez, A. J. Argüelles-Cruz, O. Camacho-Nieto, and C. Yáñez Márquez. Pollutants time-series prediction using the gamma classifier. *International Journal of Computational Intelligence Systems*, 4(4):680–711, 2011.
- [11] J. ABREU and S. FAJARDO. Clasificador automático de recurrencia de cáncer. Technical report, INSTITUTO POLITÉCNICO NACIONAL, 2009.
- [12] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.
- [13] I. López Yáñez. Clasificador automático de alto desempeño. Master’s thesis, Instituto Politécnico Nacional de México, 2007.
- [14] C. Yáñez. *Memorias Asociativas basadas en Relaciones de Orden y Operadores Binarios*. PhD thesis, Centro de Investigación en Computación, México, 2002.
- [15] I. López Yáñez. *Teoría y aplicaciones del clasificador asociativo Gamma*. PhD thesis, Instituto Politécnico Nacional de México, 2011.
- [16] V. Bolón-Canedo, N. Sánchez-Marroño, and A. Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowl Inf Syst*, 34:483–519, 2013.
- [17] Y. Caballero, R. Bello, Y. Salgado, and M. M. García. A method to edit training set based on rough sets. *International Journal of Computational Intelligence Research*, 3(3):219–229, 2007.
- [18] Y. Villuendas Rey, M. García Borroto, and C. Ruiz Shulcloper. Selección combinada de rasgos y objetos para el mejoramiento de clasificadores nn. *Reporte Técnico. Reconocimiento de Patrones*, 2010.
- [19] Prerna Chikersal, Soujanya Poria, and Erik Cambria. Sentu: Sentiment analysis of tweets by combining a rule-based classifier with supervised learning. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval 2015)*, 2015.

- 
- [20] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [21] B. Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4):261–283, 2013.
- [22] D. Graupe. *Principles of artificial neural networks*, volume 7. World Scientific, 2013.
- [23] S. Lek and J. Guégan. *Artificial neuronal networks: Application to Ecology and Evolution*. Springer Science & Business Media, 2012.
- [24] L. Zhu. Support vector machine. 2013.
- [25] Zhiquan Qi, Yingjie Tian, and Yong Shi. Robust twin support vector machine for pattern classification. *Pattern Recognition*, 46(1):305–316, 2013.
- [26] A. Moujahid, I. Inza, and P. Larrañaga. Clasificadores k-nn. Departamento de Ciencias de la Computación e Inteligencia Artificial. Universidad del País Vasco-Euscal Herriko Unibertsitatea. Disponible en: <http://www.sc.ehu.es/ccwbytes/docencia/mmcc/docs/t9knn.pdf>. Consultado: Julio, 2012.
- [27] J. Carrasco Ochoa and J. F. Martínez Trinidad. Editing and training for alvot, an evolutionary approach. *Springer-Verlag. Hong Kong*, pages 452–456, 2003.
- [28] M. A. Medina-Pérez, M. García-Borroto, Y. Villuendas-Rey, and J. Ruiz-Shulcloper. Selecting objects for alvot. In *Progress in Pattern Recognition, Image Analysis and Applications*, pages 606–613. Springer, 2006.
- [29] M. Aldape-Pérez, C. Yáñez Márquez, O. Camacho-Nieto, and A. J. Argüelles-Cruz. An associative memory approach to medical decision support systems. *Computer methods and programs in biomedicine*, 106(3):287–307, 2012.
- [30] A. V. Uriarte-Arcia, I. López Yáñez, and C. Yáñez Márquez. One-hot vector hybrid associative classifier for medical data classification. *PloS one*, 9(4), 2014.

- 
- [31] J. Olvera López, J. Carrasco Ochoa, J. Martínez Trinidad, and J. Kittler. A review of instance selection methods. *Artif Intell Rev*, 34:133 – 143, 2010.
- [32] E. Burke and G. Kendall, editors. *SEARCHMETHODOLOGIES. Introductory Tutorials in Optimization and Decision Support Techniques*. Springer, 2005.
- [33] J. Han, M. Kamber, and J. Pei. *Data Mining. Concepts and Techniques*. Elsevier, 2012.
- [34] S. Datta and S. Bose. Discovering association rules partially devoid of dissociation by weighted confidence. In *Recent Trends in Information Systems (ReTIS), 2015 IEEE 2nd International Conference on*, pages 138–143. IEEE, 2015.
- [35] B. Balaji and V. Rao. Improved classification based association rule mining. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(5):2211–2221, 2013.
- [36] H. Hong and J. Li. Using association rules to make rule-based classifiers robust. In *Proceedings of Australian Database Conference*, 2005.
- [37] H. Rahmat, A. Mustapha, M. Said, and N. Amit. Measuring differences in accuracy, compactness, and speed between c 4. 5 and cpar in classification. *Journal of Communication and Computer*, 9(1):42–46, 2012.
- [38] F. Kargarfard, A. Sami, and E. Ebrahimie. Knowledge discovery and sequence-based prediction of pandemic influenza using an integrated classification and association rule mining (cba) algorithm. *Journal of biomedical informatics*, 57:181–188, 2015.
- [39] Osama Othman and Christopher H. Bryant. Preceding rule induction with instance reduction methods. In *Machine Learning and Data Mining in Pattern Recognition*, pages 209–218. Springer, 2013.
- [40] M. Antonelli, P. Ducange, F. Marcelloni, and A. Segatori. A novel associative classification model based on a fuzzy frequent pattern mining algorithm. *Expert Systems with Applications*, 42(4):2086–2097, 2015.

- 
- [41] L. Rokach and O. Maimon. *Data mining with decision trees: theory and applications*. World scientific, 2014.
- [42] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.
- [43] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.
- [44] L. Lanzarini, W. Hasperué, C. Estrebou, F. Ronchetti, A. Villa Monte, G. Aquino, F. Quiroga, L. Rojas, and P. Jimbo Santana. Redes neuronales artificiales. In *XVII Workshop de Investigadores en Ciencias de la Computación*, 2015.
- [45] M. M. García Lorenzo, R. Bello Pérez, A. Diaz Sardiñaz, and A. Reinoso Lobato. *REDES NEURONALES ARTIFICIALES*. Prometeo Editores, 2003.
- [46] A. K. Jain, J. Mao, and K. M. Mohiuddin. Artificial neural networks - a tutorial. *Computer*, pages 31–44, 1996.
- [47] M. W. Gardner and S. R. Dorling. Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric science. *Atmospheric Environment*, 3:2627–2637, 1998.
- [48] I. Grau, G. Nápoles, I. Bonet, and M. M. García. Backpropagation through time algorithm for training recurrent neural networks using variable length instances. *Computación y Sistemas*, 17:15–24, 2013.
- [49] C. BURGES. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.
- [50] D. Boswell. Introduction to support vector machines. 2002.
- [51] G. A. BETANCOURT. Las máquinas de soporte vectorial (svms). *Scientia et Technica*, pages 67–72, 2005.
- [52] E. Fix and J. L. Hodges. Discriminatory analysis, non-parametric discrimination: consistency properties. *U. S. A. F. S. A. Medicine*, Ed. Randolph Field, Texas: Project 21-49-004, Contract AF 41(128)-31, Rep. 4., 1951.

- 
- [53] F. Moreno. Clasificadores eficaces basados en algoritmos rápidos de búsqueda del vecino más cercano. Technical report, Departamento de lenguajes y sistemas informáticos, Universidad de Alicante, 2004.
- [54] J. Mora Florez, G. Morales España, and R. Barrera Cárdenas. Evaluación del clasificador basado en los k vecinos más cercanos para la localización de la zona en falla en los sistemas de potencia. *REVISTA INGENIERÍA E INVESTIGACIÓN*, 28:81–86, 2008.
- [55] J. F. Martínez Trinidad and A. Guzman Arenas. The logical combinatorial pattern recognition an overview through selected works. *Pattern Recognition*, 34:741–751, 2001.
- [56] J. Ruiz Shulcloper and M. Lazo Cortés. Mathematical algorithms for the supervised classification based on fuzzy partial precedence. *Mathematical and Computer Modeling*, 29:111–119, 1999.
- [57] U. Escobar Franco and G. Sánchez Díaz. Algoritmo de votación incremental inc-alvot para clasificación supervisada. *Rev. Fac. Ing. Univ. Antioquia*, pages 195–204, 2009.
- [58] K. Steinbuch. Die lernmatrix. *Kybernetik*, 1(1):36–45, 1961.
- [59] M. H. Hassoun. Associative neural memories. *Oxford University Press*, 1993.
- [60] T. Kohonen. Correlation matrix memories. *IEEE Transactions on Computers*, 4:353–359, 1972.
- [61] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Sciences*, volume 79, pages 2554–2558, 1982.
- [62] B. Kosko. Bidirectional associative memories. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):49–60, 1988.
- [63] H. Sossa, R. Barrón, and R. Vázquez. New associative memories to recall realvalued patterns. *CIARP*, pages 195–202, 2004.

- [64] R. Santiago. Clasificador híbrido de patrones basado en la lernmatrix de steinbuch y el linear associator de anderson kohonen. Master's thesis, Centro de Investigación en Computación, IPN, 2003.
- [65] L. García, A. de Carvalho, and A Lorena. Noisy data set identification. In *Hybrid Artificial Intelligent Systems*, pages 629–638. Springer, 2013.
- [66] S. García, J. Luengo, and F. Herrera. *Data Preprocessing in data mining*. Springer, 2015.
- [67] G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.
- [68] L.A. Belanche and F.F. González. Review and evaluation of feature selection algorithms in synthetic problems. 2011.
- [69] F. Provost. Distributed data mining: scaling up and beyond. In *Advances in distributed data mining*. Morgan Kaufmann, 2000.
- [70] H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. *AAAI*, pages 547–552, 1991.
- [71] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR Special Issue on Variable and Feature Selection*, pages 1157–1182, 2003.
- [72] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh. *Feature extraction, foundations and applications*. Springer, Heidelberg, 2006.
- [73] T. Navin Lal, O. Chapelle, J. Weston, and A. Elisseeff. Embedded methods. In *Feature extraction*, pages 137–165. Springer, 2006.
- [74] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23:2507–2517, 2007.
- [75] M. Dash. Feature selection for classification. *Intelligent Data Analysis*, 1:131–156, 1997.
- [76] J. Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

- 
- [77] P. Liu, N. Wu, J. Zhu, J. Yin, and W. Zhang. A unified strategy of feature selection. *LNAI*, pages 457–464, 2006.
- [78] Michael R Chernick and Robert A LaBudde. *An introduction to bootstrap methods with applications to R*. John Wiley & Sons, 2014.
- [79] C.L. Chang. Finding prototypes for nearest neighbor classifier. *IEEE Transactions on Computers C-23*, pages 1179–1184, 1974.
- [80] R.A. Mollineda, F.J. Ferri, and E. Vidal. An efficient prototype merging strategy for the condensed 1-nn rule through class conditional hierarchical clustering. *Pattern Recognition*, 35:2771–2782, 2002.
- [81] J.C. Bezdek, T.R. Reichherzer, G.S. Lim, and A. Y. Multiple-prototype classifier design. *IEEE Transactions on Systems, Man and Cybernetics SMC-28*, pages 67–79, 1998.
- [82] T. Kohonen. *Self - Organizing maps*. Springer, 1995.
- [83] S. Geva and J. Sitte. Adaptive nearest neighbor pattern classification. *IEEE Transactions on Neural Networks*, 2:318–322, 1991.
- [84] R. Odorico. Learning vector quantization with training counters (lvqtc). *Neural Networks*, 10:1083–1088, 1997.
- [85] Q. Xie, C.A. Laszlo, and R.K. Ward. Vector quantization technique for nonparametric classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-15*, pages 1326–1330, 1993.
- [86] Y. Hamamoto, S. Uchimura, and S. Tomita. A bootstrap technique for nearest neighbor classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-19*, pages 73–79, 1997.
- [87] P. E. Hart. The condensed nearest neighbor rule. *IEEE Trans Inf Theory*, 14:515–516, 1968.

- 
- [88] G. L. Ritter, H.B. Woodruff, S. R. Lowry, and T. L. Isenhour. An algorithm for a selective nearest neighbor decision rule. *IEEE Trans Inf Theory*, 21:665–669, 1975.
- [89] C. Chien-Hsing, Bo-Han K., and C. Fu. The generalized condensed nearest neighbor rule as a data reduction method. In *18th international conference on pattern recognition*, pages 556–559, 2006.
- [90] D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans Syst Man Cybern*, 2:408–421, 1972.
- [91] I. Tomek. An experiment with the edited nearest-neighbor rule. *IEEE Trans Syst Man Cybern*, 6:448–452, 1976.
- [92] D. R. Wilson and T. R. Martínez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–286, 2000.
- [93] H. Brighton and C. Mellish. Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery*, 6:153–172, 2002.
- [94] L. Yuangui, H. Zhonhui, C. Yunze, and Z. Weidong. Support vector based prototype selection method for nearest neighbor rules. In *Wang L et al (eds) ICNC 2005, LNCS 3610*, pages 528–535, 2005.
- [95] A. Srisawat, T. Phienthrakul, and B. Kijisirikul. Sv-knnc: an algorithm for improving the efficiency of knearest neighbor. In *Yang Q, Webb G (eds) PRICAI 2006:LNAI 4099*, pages 975–979, 2006.
- [96] U. Garain. Prototype reduction using an artificial immune model. *Pattern Anal Appl*, 11:353–363, 2008.
- [97] J. A. Olvera-López, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad. Sequential search for decremental edition. *LNCS 3578: IDEAL 2005*, pages 280–285, 2005.
- [98] J. A. Olvera-López, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa. Restricted sequential floating search applied to object selection. In Perner P, editor, *MLDM 2007:LNAI 4571*, pages 694–702, 2007.

- 
- [99] D. R. Wilson and T. R. Martínez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–286, 2000.
- [100] J. C. Riquelme, J. S. Aguilar-Ruíz, and M. Toro. Finding representative patterns with ordered projections. *Pattern Recognition*, 36:1009–1018, 2003.
- [101] T. Raicharoen and C. Lursinsap. A divide-and-conquer approach to the pairwise opposite class-nearest neighbor (poc-nn) algorithm. *Pattern Recognition Letters*, 26:1554–1567, 2005.
- [102] B. L. Narayan, C. A. Murthy, and S. K. Pal. Maxdiff kd-trees for data condensation. *Pattern Recognition Letters*, 27:187–200, 2006.
- [103] C. J. Venmann, M. J. T. Reinders, and E. Backer. A maximum variance clustering algorithm. *IEEE Trans Pattern Anal Mach Intell*, 24:1273–1280, 2002.
- [104] J. A. Olvera-López, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad. Object selection based on clustering and border objects. In Kurzynski M et al, editor, *Computer recognition systems*, pages 27–34, 2007.
- [105] R. Paredes and E. Vidal. Weighting prototypes. a new editing approach. In *Proceedings of the international conference on pattern recognition ICPR*, volume 2, pages 25–28, 2000.
- [106] B.K. Bhattacharya, R. S. Poulsen, and G. T. Toussaint. Application of proximity graphs to editing nearest neighbor decision rules. In *International Symposium on Information Theory, Santa Monica*, 1981.
- [107] G. T. Toussaint. Proximity graphs for nearest neighbor decision rules: Recent progress. In *34 Symposium on Computing and Statistics INTERFACE-2002*, 2002.
- [108] M. García, Y. Villuendas, J. A. Carrasco, and J. F. Martinez. Using maximum similarity graphs to edit nearest neighbor classifiers. *Lecture Notes on Computer Science*, 5856:489–496, 2009.

- 
- [109] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron. A rough set perspective on data mining and knowledge. *The hand book of Data Mining and Knowledge Discovery*, 1999.
- [110] J. Rico-Juan and J. Iñesta. Adaptive training set reduction for nearest neighbor classification. *Neurocomputing*, 138:316–324, 2014.
- [111] H. Ishibuchi, T. Nakashima, and M. Nii. Learning of neural networks with ga based instance selection. In *IFSA World Congress and 20th NAFIPS International Conference*, volume 4, pages 2102–2107, 2001.
- [112] J. Antón-Vargas, Y. Villuendas-Rey, and I. López-Yáñez. Instance selection to improve gamma classifier. *Polibits*, 2015.
- [113] M. García-Borroto and J. Ruiz-Shulcloper. Object selection based on subclass error correcting for alvot. In *In Progress in Pattern Recognition, Image Analysis and Applications*, pages 496–505. Springer Berlin Heidelberg, 2007.
- [114] Q. Hu, D. Yu, J. Liu, and C. Wu. Neighborhood rough set based heterogeneous feature subset selection. *Information sciences*, 178(18):3577–3594, 2008.
- [115] G. Lin, Y. Qian, and J. Li. Nmgrs: Neighborhood-based multigranulation rough sets. *International Journal of Approximate Reasoning*, 53(7):1080–1093, 2012.
- [116] Y. L. Zhang, C. Q. Li, M. L. Lin, and Y. J. & Lin. Relationships between generalized rough sets based on covering and reflexive neighborhood system. *Information Sciences*, 2015.
- [117] K. Willick, B. Storer, and S. Wesolkowski. A new principal curve algorithm and standard deviation clouds for non-parametric ordered data analysis. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 1459–1466. IEEE, 2013.
- [118] F. Ramsey and D. Schafer. *The statistical sleuth: a course in methods of data analysis*. Cengage Learning, 2012.
- [119] R. Słowiński, S. Greco, and B. Matarazzo. *Rough-set-based decision support*, chapter Search Methodologies, pages 557–609. Springer US, 2014.

- 
- [120] R. Slowinski and D. Vanderpooten. A generalized definition of rough approximations based on similarity. *IEEE Transactions on Knowledge & Data Engineering*, 2:331–336, 2000.
- [121] S. Greco, B. Matarazzo, and R. Słowiński. *Rough membership and bayesian confirmation measures for parameterized rough sets*. Springer Berlin Heidelberg, 2005.
- [122] K. Bache and M. Lichman. Uci machine learning repository. university of california, school of information and computer science. irvine, ca., 2013.
- [123] D. H. Wolpert. The supervised learning no-free-lunch theorems. *Soft Computing and Industry*, pages 25–42, 2002.
- [124] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *In International Joint Conference on Artificial Intelligence, IJCAI'95*, 14:1137–1145, 1995.
- [125] S. García, J. Derrac, J. R. Cano, and F. Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:417–435, 2012.
- [126] M. Voorneveld. Characterization of pareto dominance. *Operations Research Letters*, 31(1):7–11, 2003.
- [127] C. C. António and L. N. Hoffbauer. Bi-level dominance ga for minimum weight and maximum feasibility robustness of composite structures. *Composite Structures*, 135:83–95, 2016.
- [128] Y. Yun, H. Nakayama, and M. Yoon. Generation of pareto optimal solutions using generalized dea and pso. *Journal of Global Optimization*, 64(1):49–61, 2016.
- [129] X. Blasco, Reynoso-Meza G., E. A. S. Pérez, and J. V. S. Pérez. Asymmetric distances to improve n-dimensional pareto fronts graphical analysis. *Information Sciences*, 2016.

- 
- [130] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.
- [131] Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. Un tutorial sobre el uso de test estadísticos no paramétricos en comparaciones múltiples de metaheurísticas y algoritmos evolutivos.
- [132] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization. *Journal of Global Optimization*, 11:341–359, 1997.