UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

Computación Evolutiva Multi-Objetivo
para Selección de Atributos Y
Clasificación Interpretable

D. Carlos Martínez Cortés

2019

Tesis doctoral

COMPUTACIÓN EVOLUTIVA MULTI-OBJETIVO PARA SELECCIÓN DE ATRIBUTOS Y CLASIFICACIÓN INTERPRETABLE

20 de junio de 2019

Dirigida por:
Fernando Jiménez Barrionuevo
Gracia Sánchez Carpena

Presentada por:
Carlos Martínez Cortés
Universidad de Murcia
Facultad de Informática
Departamento de Ingeniería de la Información y las Comunicaciones
carlos.martinez6@um.es

Agradecimientos

Comienzo agradeciendo la labor de mis directores Fernando Jiménez Barrionuevo y Gracia Sánchez Carpena, la paciencia y el esmero que han tenido guiándome y enseñándome desde cuando empece mi trabajo de fin de Grado en 2014 hasta ahora, ellos lo han hecho todo mucho más sencillo. También agradecer la ayuda de los miembros del departamento de Ingeniería de la Información y las Comunicaciones de la Universidad de Murcia que han colaborado en mayor o menor grado en la realización de los artículos que forman esta tesis, en especial a José Tomás Palma Méndez. Agradecer también a mis padres Pedro y Fuensanta por haberme brindado un apoyo total en todo lo que lo que he emprendido en la vida, haberme enseñado tan buenos valores y haber puesto mi bienestar por delante del suyo siempre. A mi abuela Carmen por todo el cuidado y cariño que me ha dado durante años. A mi hermano Jesús por ser un amigo que siempre esta ahí cuando se le necesita. Por último agradecer a mis compañeros del Judo Club Ciudad de Murcia, con ellos he pasado muy buenos ratos a la vez que aprendía y enfrentaba nuevos retos.

Índice

Índice de figuras

Índice de tablas

Índice de Algoritmos

Resumen

En el contexto del *aprendizaje supervisado*, en esta Tesis Doctoral se han desarrollado modelos de optimización multi-objetivo para los problemas de *selección de atributos* y de *clasificación interpretable*, así como *algoritmos evolutivos multi-objetivo* para sus resoluciones. El problema de la selección de atributos se enmarca dentro de un proceso mas general que es la *reducción de la dimensionalidad* de los datos. Este proceso es fundamental hoy día debido a la gran cantidad de datos que cada vez más se generan con el desarrollo imparable de las tecnologías de la información. El problema de la clasificación o predicción interpretable juega también un papel crucial hoy día, ya que no siempre es aceptable un modelo automático si éste no es entendible y validable por un experto, sobre todo en contextos donde la ética profesional lo requiere, como por ejemplo, la medicina o los negocios. Por otro lado, la *Computación Evolutiva Multi-objetivo* se ha mostrado como un *metaheurística* muy potente para resolver ambos tipos de problemas, y aunque no garantiza soluciones óptimas, éstas pueden resultar más satisfactorias que las proporcionadas con las técnicas clásicas de búsqueda, optimización y aprendizaje.

Los algoritmos evolutivos multi-objetivo desarrollados en esta tesis han sido implementados en la plataforma *Weka* de *machine learning* con los nombres *MultiObjectiveEvolutionarySearch* y *MultiObjectiveEvolutionaryFuzzyClassifier* respectivamente. Para el problema de selección de atributos, la estrategia de búsqueda *MultiObjectiveEvolutionarySearch* puede combinarse con distintos evaluadores para configurar métodos de selección de atributos tanto *filter* como *wrapper*, con diferentes medidas estadísticas, clasificadores y métricas de evaluación, lo que hace que la técnica sea muy flexible y robusta. Los algoritmos *ENORA* y *NSGA-II* han sido implementados como estrategia de búsqueda, resolviendo un problema de optimización booleana con los objetivos de precisión y de cardinalidad de los subconjuntos de atributos. Para el problema de clasificación interpretable, el clasificador *MultiObjectiveEvolutionaryFuzzyClassifier* permite construir clasificadores basados en reglas, tanto *fuzzy* (gaussianos) como *crisp*, con datos numéricos y categóricos, en problemas de clasificación multi-clase, permitiendo configurar distintos evaluadores en la fase de aprendizaje. Los algoritmos *ENORA* y *NSGA-II* han sido implementados para la construcción de clasificadores basados en reglas, resolviendo un problema de optimización combinatoria mixta con restricciones, con los objetivos de precisión y de complejidad del conjunto de reglas, y restricciones de similaridad de los conjuntos fuzzy gaussianos.

Para los experimentos se han utilizado dos campos de fundamentales de aplicación, en el *screening virtual* para el descubrimiento de fármacos, y en la gestión de las habilidades profesionales de agentes en un centro de contacto con datos extraídos de la empresa *GAP S.R.L.* en el norte de Italia. También se han utilizado bases de datos públicas del *UCI Machine Learning Repository* por razones de reproducibilidad. Los resultados han sido analizados siguiendo metodologías propias del *análisis inteligente de datos*, y las conclusiones están abaladas por tests estadísticos, los cuales muestran un excelente comportamiento de las técnicas propuestas tanto para selección de atributos como para clasificación basada en reglas, en comparación con otras técnicas, algoritmos y clasificadores del estado del arte ampliamente consolidados.

Abstract

In the context of *supervised learning*, in this Doctoral Thesis, multi-objective optimization models have been developed for the problems of *feature selection* and *interpretable classification*, as well as *multi-objective evolutionary algorithms* for their resolutions. The problem of feature selection is framed within a more general process that is the *dimensionality reduction* of data. This process is fundamental today due to the large amount of data that is increasingly generated with the unstoppable development of information technologies. The problem of interpretable classification (or prediction) also plays a crucial role today, since an automatic model is not always acceptable if it is not understandable and validated by an expert, especially in contexts where professional ethics requires it, such as, for example, medicine or business. On the other hand, the *Multi-objective Evolutionary Computation* has been shown as a very powerful *metaheuristic* to solve both types of problems, and although it does not guarantee optimal solutions, these can be more satisfactory than those provided with the classic search, optimization and learning techniques.

The multi-objective evolutionary algorithms developed in this thesis have been implemented in the *Weka* platform of *machine learning* with the names *MultiObjectiveEvolutionarySearch* and *MultiObjectiveEvolutionaryFuzzyClassifier* respectively. For the feature selection problem, the search strategy *MultiObjectiveEvolutionarySearch* can be combined with different evaluators to configure feature selection methods both *filter* and *wrapper*, with different statistical measures, classifiers and evaluation metrics,

which makes the technique very flexible and robust. The algorithms *ENORA* and *NSGA-II* have been implemented as search strategy, solving a boolean optimization problem with objectives of precision and attribute subset cardinality. For the problem of interpretable classification, the classifier *MultiObjectiveEvolutionaryFuzzyClassifier* allows to build rule-based classifiers, both *fuzzy* (Gaussian) and *crisp*, with numerical and categorical data, in multi-class classification problems, allowing to configure different evaluators in the learning phase. The algorithms *ENORA* and *NSGA-II* have been again implemented for the construction of rule-based classifiers, solving a mixed combinatorial constrained optimization problem in this case, with the objectives of precision and rule set complexity, and similarity constraints of Gaussian fuzzy sets.

Two fundamental application areas have been used for the experiments, in *virtual screening* for the discovery of drugs, and for the management of the professional skills of agents in a contact center with data extracted from the company *GAP SRL* in the north of Italy. Public databases of the *UCI Machine Learning Repository* have also been used for reproducibility reasons. The results have been analyzed following the methodologies of *intelligent analysis of data*, and the conclusions are supported by statistical tests, which show an excellent behavior of the proposed techniques both for feature selection and for rule-based classification, in comparison with other techniques, algorithms and classifiers of the state-of-the-art widely consolidated.

1. Introducción

Nos encontramos en una época con muchas innovaciones tecnológicas que están produciendo una gran cantidad de datos los cuales requieren de una gestión. Las dificultades más habituales vinculadas a la gestión de datos se centran en la recolección, almacenamiento, búsqueda, compartición, análisis y visualización. La tendencia a manipular datos se debe a la necesidad en muchos casos de incluir dicha información para la creación de informes estadísticos y modelos predictivos utilizados en diversas aplicaciones, como son los sistemas publicitarios, la medicina, el correo basura, el descubrimiento de fármacos, el procesamiento del lenguaje natural, los negocios y muchas otras. Disciplinas como la *minería de datos* [1], el *análisis inteligente de datos* [2], el *machine learning* [3], o el *soft computing* [4], han surgido de forma natural y progresiva para tal fin, y en la actualidad, el *big data* [5].

Dos aspectos fundamentales para el tratamiento de información son, por un lado, la reducción de la dimensionalidad y particularmente, la *selección de los atributos* [6] que tienen mayor incidencia en la salida supervisada de los datos, y por otro, la construcción de clasificadores que permitan, no solo la creación de modelos precisos para los datos observados, sino también la alta capacidad de predicción de nuevas instancias de datos. Además, en determinadas aplicaciones, como son las relacionadas con la medicina o los negocios, se requieren *modelos de clasificación interpretables* [7], de cara tanto a la validación del modelo por parte del experto, como para la explicación del comportamiento del clasificador a los usuarios del mismo. La *computación evolutiva multi-objetivo* [8] se ha consolidado en los últimos años como una técnica muy potente y robusta tanto como estrategia de búsqueda en métodos de selección de atributos como para la construcción de clasificadores interpretables, incluyendo *árboles de decisión* [9] y *sistemas basados en reglas*, tanto *fuzzy* [10] como *crisp* [11]. El algoritmo evolutivo multi-objetivo *ENORA* fué inicialmente desarrollado por los directores de esta Tesis Doctoral para optimización de parámetros reales [12], ha sido incorporado por el doctorando a la plataforma *Weka* [13] como estrategia de búsqueda en métodos de selección de atributos, y como clasificador para la construcción de modelos basados en reglas. El algoritmo *ENORA* y su integración a *Weka* han servido de base en esta tesis para la consecución de las distintas propuestas que se han llevado a cabo, entre las que se hayan la selección de atributos para tareas de clasificación interpretable mediante árboles de decisión y mediante sistemas basados en reglas, tanto crisp con datos categóricos como fuzzy con datos numéricos, las cuales han sido probadas, depuradas y comparadas estadísticamente con otras técnicas del estado del arte.

De esta forma, la Tesis Doctoral se contextualiza entorno al desarrollo, test y validación de nuevas estrategias de búsqueda para selección de atributos así como de algoritmos de clasificación interpretables que permitan, en su conjunto, una predicción eficaz, eficiente e interpretable en ambiente supervisado. Las técnicas propuestas han sido comparadas con las técnicas tradicionales en distintos ámbitos de aplicación, para lo cual se han utilizado bases de datos pertenecientes a diversas disciplinas y con distintas estructuras en lo que se refiere a la cantidad de atributos y de instancias, así como distintos tipos de atributos tanto de entrada como de salida. Se han utilizado dos campos de aplicación, en el *screening virtual* [14] para el descubrimiento de fármacos y en la gestión de las habilidades profesionales de agentes en un *contact center* con datos extraídos de la empresa *GAP S.R.L.* en el norte de Italia. También se han utilizado bases de datos públicas del *UCI Machine Learning Repository* [15] por razones de reproducibilidad.

La tesis doctoral se ha realizado en la modalidad de compendio de publicaciones, regulada por el *Artículo 20 del Reglamento de Doctorado de la Universidad de Murcia* con arreglo al *Artículo 11.6 del Real Decreto 99/2011, de 28 de enero*, en la *Escuela Internacional de Doctorado de la Universidad de Murcia*. Los siguientes tres artículos, claramente relacionados y que justifican la unidad científica de la tesis, han sido publicados en los cuartiles *Q1* y *Q2* de las revistas indizadas del *Journal Citation Reports 2018*, de reconocido prestigio según los indicios de calidad establecidos por la *Agencia Nacional de Evaluación de la Calidad y Acreditación* (*ANECA*) para la rama del conocimiento de *Ingeniería y Arquitectura* en la evaluación de la actividad investigadora:

A1 Fernando Jiménez, Horacio Pérez-Sánchez, José Palma, Gracia Sánchez, Carlos Martínez. A methodology for evaluating multi-objective evolutionary feature selection for classification in the context of virtual screening. *Soft Computing* 2018, https://doi.org/10.1007/s00500-018-3479-0.

A2 Fernando Jiménez, Carlos Martínez, Luis Miralles-Pechuán, Gracia Sánchez, Guido Sciavicco. Multi-Objective Evolutionary Rule-Based Classification with Categorical Data. *Entropy* 2018, 20, 684.

A3 Fernando Jiménez, Carlos Martínez, Enrico Marzano, J. Palma, Gracia Sánchez, Guido Sciavicco. Multi-objective Evolutionary Feature Selection for Fuzzy Classification. *IEEE Transactions on Fuzzy Systems* 2019, DOI: 10.1109/TFUZZ.2019.2892363.

A continuación se muestra un resumen global de los objetivos de la investigación, de la metodología seguida para su consecución, y de las conclusiones finales, en el que se unifican los resultados parciales presentados en cada uno de los trabajos, organizado con los siguientes apartados: la sección 2 muestra los objetivos de la investigación; la sección 3 contiene los antecedentes de la tesis y los materiales y métodos usados en cada uno de los artículos del compendio, incluyendo métodos adicionales que han sido desarrollados pero aún no han sido publicados; la sección 4 resume los principales resultados; la sección 5 resume las principales conclusiones; la sección 6 indica los trabajos de investigación asociados a la tesis que están previstos para el futuro. Finalmente se incluye un apartado con las cartas de aceptación de los artículos, un apartado con las referencias bibliográficas citadas en la memoria, un apéndice que incluye material suplementario (Apéndice A), y el resumen e información adicional sobre los artículos (Apéndice B).

2. Objetivos

Los siguientes objetivos han sido establecidos en esta tesis dentro del marco de la computación evolutiva multi-objetivo para problemas de selección de atributos en tareas de clasificación interpretable:

Objetivos Generales:

- *OG1*: Identificar y analizar modelos de optimización para selección de atributos en tareas de clasificación interpretable.
- *OG2*: Analizar tipos de métodos de selección de atributos en tareas de clasificación interpretable.
- *OG3*: Analizar estrategias de búsqueda para métodos de selección de atributos en tareas de clasificación interpretable.
- *OG4*: Analizar evaluadores para métodos de selección de atributos en tareas de clasificación interpretable.
- *OG5*: Diseñar estrategias de búsqueda basadas en computación evolutiva multi-objetivo para métodos de selección de atributos en tareas de clasificación interpretable.
- *OG6*: Diseñar un sistema de ayuda a la decisión para evaluar y comparar métodos de selección de atributos.
- *OG7*: Identificar y analizar modelos de optimización multi-objetivo con restricciones para clasificación basada en reglas.
- *OG8*: Diseñar clasificadores basados en reglas mediante computación evolutiva multi-objetivo.
- *OG9*: Diseñar métodos de selección de atributos eficaces y eficientes para clasificación basada en reglas fuzzy.
- *OG10*: Aplicar y validar métodos de selección de atributos para clasificación interpretable en bases de datos públicas y del mundo real.

Objetivos Específicos:

- *OE1*: Enunciar y formular, en términos de programación matemática, modelos de optimización multi-objetivo para selección de atributos en tareas de clasificación interpretable.
- *OE2*: Implementar estrategias de búsqueda evolutivas multi-objetivo para métodos de selección de atributos, mediante los algoritmos *ENORA* y *NSGA-II*.
- *OE3*: Implementar métodos de selección de atributos en tareas de clasificación interpretable del tipo filter y wrapper, univariate y multivariate, attribute evaluation (ranker) y subset evaluation, con estrategias de búsqueda deterministas y probabilistas, basadas en optimización multi-objetivo y con un único objetivo.
- *OE4*: Implementar un sistema de ayuda a la decisión para la evaluación y comparación, mediante test estadísticos, de métodos de selección de atributos en tareas de clasificación interpretable.

- *OE5*: Enunciar y formular, en términos de programación matemática, modelos de optimización multi-objetivo con restricciones para clasificación basada en reglas.
- *OE6*: Implementar clasificadores basados en reglas con datos categóricos mediante computación evolutiva multi-objetivo, a través de los algoritmos *ENORA* y *NSGA-II*.
- *OE7*: Implementar clasificadores basados en reglas fuzzy mediante computación evolutiva multi-objetivo, a través de los algoritmos *ENORA* y *NSGA-II*.
- *OE8*: Implementar un método wrapper de selección de atributos para clasificación basada en reglas fuzzy con estrategia de búsqueda y evaluador basados ambos en computación evolutiva multi-objetivo, a través de los algoritmos *ENORA* y *NSGA-II*.
- *OE9*: En el contexto del screening virtual:
  - *OE9.1*: Determinar qué modelo de optimización es más eficaz para selección de atributos, multi-objetivo o con un único objetivo.
  - *OE9.2*: Determinar qué métrica de evaluación es más eficaz para selección de atributos, porcentaje de aciertos o área bajo la curva *ROC*.
  - *OE9.3*: Determinar qué estrategia de búsqueda es más eficaz para selección de atributos, *ENORA* o *NSGA-II*.
  - *OE9.4*: Construir, evaluar, visualizar, testar y validar modelos de clasificación precisos e interpretables basados en árboles de decisión para las bases de datos estándar *tk* y *mr* para screening virtual del *DUD*.
  - *OE9.5*: Comparar estadísticamente los resultados con otras estrategias de búsqueda de la literatura.
- *OE10*: En el contexto de la clasificación basada en reglas con datos categóricos:
  - *OE10.1*: Determinar qué métrica de evaluación es más eficaz para clasificación basada en reglas con datos categóricos, porcentaje de aciertos, área bajo la curva *ROC* o root-mean-square error.
  - *OE10.2*: Determinar qué algoritmo evolutivo multi-objetivo es más eficaz para clasificación basada en reglas con datos categóricos, *ENORA* o *NSGA-II*.
  - *OE10.3*: Construir, evaluar, visualizar, testar y validar modelos de clasificación precisos e interpretables basados en reglas con datos categóricos para las bases de datos públicas del *UCI Machine Learning Repository*.
  - *OE10.4*: Comparar estadísticamente los resultados con otros clasificadores basados en reglas de la literatura.
- *OE11*: En el contexto del *GAP S.R.L. Contact Center*:
  - *OE11.1*: Determinar qué métrica de evaluación es más eficaz para clasificación basada en reglas fuzzy, porcentaje de aciertos, área bajo la curva *ROC* o root-mean-square error.
  - *OE11.2*: Determinar qué algoritmo evolutivo multi-objetivo es más eficaz para clasificación basada en reglas fuzzy, *ENORA* o *NSGA-II*.
  - *OE11.3*: Determinar, mediante test estadísticos, el número de generaciones óptimas, tanto en la estrategia de búsqueda evolutiva como en el evaluador evolutivo, para el método wrapper de selección de atributos.
  - *OE11.4*: Construir, evaluar, visualizar, testar y validar modelos de clasificación precisos e interpretables basados en reglas fuzzy para las bases de datos *INBOUND_AGENTS* y *ALL_AGENTS* del *GAP S.R.L. Contact Center*.
  - *OE11.5*: Comparar estadísticamente los resultados con otros clasificadores fuzzy de la literatura.
  - *OE11.6*: Interpretar y validar el modelo propuesto a manos de un experto (CEO de la empresa *GAP S.R.L.*).

3. Materiales y métodos

En este punto se incluyen los antecedentes que preceden a lo investigado en la tesis (sección 3.1), las herramientas usadas para la realización de la tesis (sección 3.2) y el resumen de los proyectos que forman parte de la tesis (secciones 3.3, 3.4, 3.5 y 3.6).

3.1. Antecedentes

3.1.1. Computación evolutiva multi-objetivo

El término *optimización* [16] se refiere a la selección del mejor elemento, con respecto a algunos criterios, de un conjunto de elementos alternativos. *La programación matemática* [17] se ocupa de la teoría, algoritmos, métodos y técnicas para representar y resolver problemas de optimización. En esta tesis, estamos interesados en una clase de problemas de programación matemática llamados *problemas de optimización multi-objetivo* [18], que se pueden definir formalmente, para l objetivos y m restricciones, como sigue:

$$\begin{array}{lll} Min./Max. & f_i(\mathbf{x}), & i = 1, \ldots, l \\ sujeto\ a & g_j(\mathbf{x}) \leq 0, & j = 1, \ldots, m \end{array} \tag{1}$$

donde $f_i(\mathbf{x})$ y $g_j(\mathbf{x})$ son funciones arbitrarias. Los problemas de optimización pueden separarse naturalmente en dos categorías: aquellos con variables discretas, que llamamos *combinatorios*, y aquellos con variables continuas. En los problemas combinatorios, estamos buscando objetos de un conjunto finito, o infinitamente contable infinito, $\mathcal{X}$, donde los objetos son típicamente enteros, conjuntos, permutaciones o gráficos. En problemas con variables continuas, en cambio, buscamos parámetros reales que pertenecen a algún dominio continuo. En (1), $\mathbf{x} = \{x_1, x_2, \ldots, x_w\} \in \mathcal{X}^w$ representa el conjunto de variables de decisión, donde $\mathcal{X}$ es el dominio para cada variable x_k, $k = 1, \ldots, w$.

Sea $\mathcal{F} = \{\mathbf{x} \in \mathcal{X}^w \mid g_j(\mathbf{x}) \leq 0,\ j = 1, \ldots, m\}$ el conjunto de todas las soluciones posibles para (1). Queremos encontrar un subconjunto de soluciones $\mathcal{S} \subseteq \mathcal{F}$ llamado *conjunto no-dominado* (o *conjunto Pareto optimal*). Una solución $\mathbf{x} \in \mathcal{F}$ es *no-dominada* si no hay otra solución $\mathbf{x}' \in \mathcal{F}$ que *domine* a $\mathbf{x}$ y una solución $\mathbf{x}'$ *domina* a $\mathbf{x}$ si y solo si existe un i $(1 \leq i \leq l)$ tal que $f_i(\mathbf{x}')$ mejora $f_i(\mathbf{x})$, y para todo i $(1 \leq i \leq l)$, $f_i(\mathbf{x})$ no mejora $f_i(\mathbf{x}')$. En otras palabras, $\mathbf{x}'$ *domina* $\mathbf{x}$ si y solo si $\mathbf{x}'$ es mejor que $\mathbf{x}$ en al menos un objetivo y no es peor que $\mathbf{x}$ para cada uno de los otros objetivos. El conjunto $\mathcal{S}$ de soluciones no-dominadas de (1) puede ser formalmente definido como:

$$\mathcal{S} = \{\mathbf{x} \in \mathcal{F} \mid \nexists\, \mathbf{x}'(\mathbf{x}' \in \mathcal{F} \wedge \mathcal{D}(\mathbf{x}', \mathbf{x}))\}$$

donde:

$$\mathcal{D}(\mathbf{x}', \mathbf{x}) = \exists i(1 \leq i \leq l, f_i(\mathbf{x}') < f_i(\mathbf{x})) \wedge \forall i(1 \leq i \leq l, f_i(\mathbf{x}') \leq f_i(\mathbf{x})).$$

Una vez que el conjunto de soluciones óptimas está disponible, se puede elegir la solución más satisfactoria aplicando un criterio de preferencia. Cuando todas las funciones f_i son lineales, el problema es un *problema de programación lineal* [19], que es el problema clásico de programación matemática y para el cual existen algoritmos extremadamente eficientes para obtener la solución óptima (por ejemplo, el *método simplex* [20]). Cuando cualquiera de las funciones f_i es no lineal, tenemos un *problema de programación no lineal* [21]. Un problema de programación no lineal en el que los objetivos son funciones arbitrarias es, en general, intratable. En principio, cualquier algoritmo de búsqueda puede usarse para resolver problemas de optimización combinatoria, aunque no se garantiza que encuentren una solución óptima. Los métodos *metaheurísticos* como los *algoritmos evolutivos* [22] se han usado típicamente para encontrar soluciones aproximadas para problemas complejos de optimización multi-objetivo, incluida la selección de atributos y la clasificación fuzzy.

Los algoritmos evolutivos multi-objetivo *ENORA* [12] y *NSGA-II* [8] utilizan una estrategia $(\mu + \lambda)$ (Algoritmo 1) con $\mu = \lambda = popsize$, donde μ corresponde al número de padres y λ se refiere al número de hijos (*popsize* es el tamaño de la población), con *selección del torneo binario* (Algoritmo 2) y una función de clasificación basada en los frentes de Pareto y *la distancia de crowding* (Algoritmo 3 y Algoritmo 4). La diferencia entre *NSGA-II* y *ENORA* es cómo se realiza el cálculo del ranking de los individuos en la población. En *ENORA* cada individuo pertenece a un slot (según lo establecido en [23]) del espacio de búsqueda objetivo y la posición de un individuo en una población es el nivel de no dominación del individuo en su slot. Por otro lado, en *NSGA-II*, el posición de un individuo en una población es el nivel

de no dominación del individuo en toda la población. *ENORA* y *NSGA-II* usan el mismo algoritmo de ordenación no dominada, el *fast non-dominated sorting* [24]. Este compara cada solución con el resto de las soluciones y almacena los resultados para evitar comparaciones duplicadas entre cada par de soluciones. Para un problema con l objetivos y una población con N soluciones, este método debe realizar $l \cdot N \cdot (N-1)$ comparaciones de las funciones objetivo, lo que significa que tiene una complejidad de tiempo de $O(l \cdot N^2)$ [25]. Sin embargo, *ENORA* distribuye la población en N slots (en el mejor de los casos), por lo tanto, la complejidad de *ENORA* es de $O(l \cdot N^2)$ en el peor de los casos y $O(l \cdot N)$ en el mejor de los casos.

Algoritmo 1 Estrategia de optimización multiobjetivo $(\mu + \lambda)$

Entrada: $T > 1$ {Número de iteraciones}
Entrada: $N > 1$ {Número de individuos en la población}
Inicializa P con N individuos
Evalúa todos los individuos de P
$t \leftarrow 0$
mientras $t < T$ **hacer**
  $Q \leftarrow \emptyset$
  $i \leftarrow 0$
  mientras $i < N$ **hacer**
    $Padre1 \leftarrow$ Selección por torneo binario de P
    $Padre2 \leftarrow$ Selección por torneo binario de P
    $Hijo1, Hijo2 \leftarrow$ Variación adaptativa de $Parent1, Parent2$
    Evalúa $Hijo1$
    Evalúa $Hijo2$
    $Q \leftarrow Q \bigcup \{Hijo1, Hijo2\}$
    $i \leftarrow i + 2$
  fin mientras
  $R \leftarrow P \bigcup Q$
  $P \leftarrow N$ mejores individuos de R de acuerdo a la función Rank-Crowding-Better
  $t \leftarrow t + 1$
fin mientras
devolver devuelve individuos no-dominados de P

Algoritmo 2 Selección por torneo binario

Entrada: P {Población}
$I \leftarrow$ Selección aleatoria desde P
$J \leftarrow$ Selección aleatoria desde P
si I es mejor que J de acuerdo a la función Rank-Crowding-Better en la población P **entonces**
  devolver I
si no
  devolver J
fin si

Algoritmo 3 Función Rank-Crowding-Better

Entrada: P {Población}
Entrada: I, J {Individuos a comparar}
si $rank\,(P, I) < rank\,(P, J)$ **entonces**
  devolver $True$
fin si
si $rank\,(P, J) < rank\,(P, I)$ **entonces**
  devolver $False$
fin si
devolver $crowding\_distance\left(P^I, I\right) > crowding\_distance\left(P^J, J\right)$

La razón principal por la que *ENORA* y *NSGA-II* se comportan de manera diferente es la siguiente: *NSGA-II* nunca selecciona a un individuo dominado por otro en un torneo binario, mientras que en *ENORA*, el individuo dominado puede ser el ganador del torneo. La figura 1 muestra este comportamiento gráficamente. Por ejemplo, si se seleccionan los individuos B y C para un torneo binario con *NSGA-II*, el individuo B gana a C porque B domina a C. A la inversa, el individuo C gana a B con *ENORA* porque el individuo C tiene una mejor posición en su slot que el individuo B. De esta manera, *ENORA* permite que los individuos en cada slot evolucionen hacia el frente de Pareto fomentando la diversidad. Este enfoque genera un hipervolumen mejor que el de *NSGA-II* a lo largo del proceso de evolución.

ENORA ha sido usado en diversos problemas durante la última década. Se ha aplicado *ENORA* a optimización de parámetros reales con restricciones [12], optimización fuzzy [26], clasificación fuzzy [27], selección de atributos para clasificación [28] y selección de atributos para regresión [23]. El algoritmo *NSGA-II* fue diseñado por K. Deb et al. y se ha demostrado que es un algoritmo muy potente y rápido en contextos de optimización multi-objetivo de todo tipo. La mayoría de los investigadores en computación evolutiva multiobjetivo utilizan *NSGA-II* como base para comparar el rendimiento de sus propios algoritmos.

Algoritmo 4 Función Crowding_distance

Entrada: P {Población}
Entrada: I {Individuo}
Entrada: l {Número of objetivos}
para $j = 1$ a l **hacer**
  $f_j^{max} \leftarrow \max_{I \in P}\{f_j^I\}$
  $f_j^{min} \leftarrow \min_{I \in P}\{f_j^I\}$
  $f_j^{sup_j^I}$ ←valor del j-ésimo objetivo objective para el individuo más alto adyacente al j-ésimo objetivo del individuo I
  $f_j^{inf_j^I}$ ← valor del j-ésimo objetivo objective para el individuo más bajo adyacente al j-ésimo objetivo del individuo I
fin para
para $j = 1$ a l **hacer**
  si $f_j^I = f_j^{max}$ o $f_j^I = f_j^{min}$ **entonces**
    devolver ∞
  fin si
fin para
$CD \leftarrow 0{,}0$
para $j = 1$ a l **hacer**
  $CD \leftarrow CD + \frac{f_j^{sup_j^I} - f_j^{inf_j^I}}{f_j^{max} - f_j^{min}}$
fin para
devolver CD

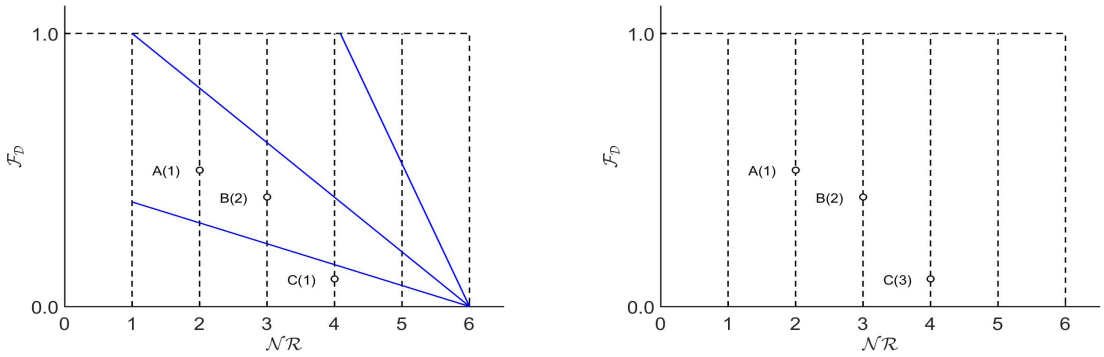


Figura 1: Asignación de ranking de individuos con *ENORA* vs *NSGA-II*.

Aunque *NSGA-II* se desarrolló en 2002, sigue siendo un algoritmo moderno y todavía es un desafío mejorarlo. Actualmente existe una versión mejorada para problemas de *optimización many-objective* llamada *NSGA-III* [29].

3.1.2. Selección de atributos evolutiva multi-objetivo

La selección de atributos se define en [30] como el proceso de eliminar atributos de una base de datos que son irrelevantes para la tarea a realizar. La selección de atributos facilita la comprensión de los datos, reduce los requisitos de medición y almacenamiento, reduce el tiempo de procesamiento computacional y reduce el tamaño del conjunto de datos, de modo que el aprendizaje de modelos se convierte en un proceso más fácil. Los métodos selección de atributos pueden ser *univariate* cuando los atributos se evalúan individual e independientemente, o *multivariate* cuando se evalúan en subconjuntos. La selección de atributos multivariate multi-objetivo se puede formular como una instancia del problema (1) con $l = 2$ de la siguiente forma:

$$\begin{array}{ll} Minimizar & \mathcal{F}(\mathbf{x}) \\ Minimizar & \mathcal{C}(\mathbf{x}) \end{array} \tag{2}$$

donde $\mathbf{x} = \{x_1, x_2, \ldots, x_w\}$ es un conjunto de variables de decisión booleanas, es decir, $x_k \in \{true, false\}$, $k = 1, \ldots, w$, siendo w el número de atributos de la base de datos. El problema (2) es por lo tanto un problema de optimización combinatoria booleano multi-objetivo donde $x_k = 1$ representa que la variable x_k está seleccionada y $x_k = 0$ representa que la variable x_k no está seleccionada, para todo $k = 1, \ldots, w$. Esto generalmente implica un problema *NP-complejo* donde existen 2^w subconjuntos de atributos candidatos. Según el tipo de función $\mathcal{F}(\mathbf{x})$, los métodos de selección de atributos se pueden dividir en *filter* y *wrapper*.

Los primeros aplican medidas estadísticas para evaluar el conjunto de atributos (por ejemplo, *ratio de ganancia* [31], *correlación* [32], *significancia probabilística* [33], etc.). Los métodos wrapper interactúan con un algoritmo de aprendizaje para evaluar el conjunto de atributos utilizando alguna métrica de evaluación [34, 35]. La función $\mathcal{C}(\mathbf{x})$ mide el número de atributos seleccionadas, es decir:

$$\mathcal{C}(\mathbf{x}) = \sum_{k=1}^{w} \mathcal{N}(x_k)$$

donde $\mathcal{N}$ es una función que transforma un valor booleano en numérico. ($true = 1$ y $false = 0$). El frente de Pareto en el problema eq. (2) contiene como máximo w soluciones no dominadas. La figura 2 muestra el frente de Pareto de un problema de selección de atributos multivariate ficticio como en la ecuación (2) con $w = 5$ atributos en su base de datos. El frente de Pareto está compuesto en este ejemplo por tres soluciones no dominadas (subconjuntos de atributos) con 1, 2 y 3 atributos respectivamente. Las soluciones con 4 y 5 atributos son soluciones dominadas.

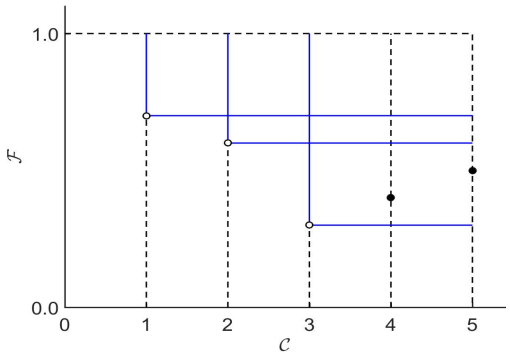


Figura 2: Un frente de Pareto de un problema de selección de atributos multivariate.

El uso de *Algoritmos Genéticos* [36] para la selección de atributos para tareas de clasificación se introdujo en [37]. Desde entonces, los algoritmos genéticos han llegado a ser considerados como una herramienta muy potente para selección de atributos [38] y ha sido propuesta por numerosos autores como una estrategia de búsqueda en métodos filter, wrapper y embedded [39, 40, 41], así como en algoritmos de ranking de atributos de evaluación de subconjuntos de atributos [42, 43]. Una revisión de las técnicas evolutivas para selección de atributos se puede encontrar en [44]. El primer enfoque evolutivo que involucra optimización multi-objetivo para selección de atributos se propuso en [45]. Desde entonces, han aparecido muchos enfoques evolutivos multi-objetivo para selección de atributos en la literatura, tanto en métodos filter como wrapper, y en entornos supervisados y no supervisados. Se puede encontrar un estudio de algoritmos evolutivos multi-objetivo para minería de datos en general en [46, 47]. A continuación se presentan algunos de los trabajos más relevantes aparecidos en la literatura durante la última década.

En [48] se propone un método ensemble que combina un algoritmo multi-objetivo evolutivo y determinación de relevancia automática (ARD) bayesiana, y usa *NSGA* para minimizar el error y la cantidad de atributos. En [49] se combinan la optimización evolutiva multi-objetivo y las máquinas soporte vectorial. *NSGA-II* se usa para minimizar la tasa de falsos positivos, la tasa de falsos negativos y la cantidad de vectores de soporte para reducir la complejidad computacional. En [50] se comparan dos métodos embedded con tres y dos objetivos, respectivamente, aplicados al diagnóstico de cáncer. El algoritmo genético de 3-objetivos optimiza la sensibilidad, la especificidad y la cantidad de genes, y el algoritmo genético de 2-objetivos optimiza la precisión y la cantidad de genes. *NSGA-II* se usa como estrategia de búsqueda y *SVM* se usa para la tarea de clasificación. En [51], se presenta un algoritmo memético multi-objetivo filter combinado con búsqueda local, que es una sinergia del algoritmo evolutivo multi-objetivo y la búsqueda local basada en el método filter de la identificación simultánea de atributos full class relevant (*FCR*) y partial class relevant (*PCR*), donde se usa *NSGA-II* como algoritmo evolutivo multi-objetivo. En [52] se propone un método filter que incluye medidas de consistencia, dependencia, distancia e información, y se usa *NSGA-II* como estrategia de búsqueda. En [53] se propone un método wrapper para el reconocimiento de Entidades Nombradas (NER) en el cual se maximizan la métricas recall y accuracy de modelos basados en máxima entropía, utilizando *NSGA-II* como algoritmo evolutivo multi-objetivo. En [54] se introduce una modificación en la relación de dominancia para tratar con un número arbitrariamente

grande de objetivos, usando *NSGA-II* y regresión logística y Naive Bayes con corrección de Laplace como algoritmos de clasificación. En [55] se propone minimizar el número de atributos y maximizar la precisión del clasificador y/o minimizar los errores obtenidos mediante el uso de máquinas de soporte vectorial, aplicado al diagnóstico cardíaco *SPECT*. Se utiliza el algoritmo genético elitista de frente de Pareto reducido (*RPSGAe*) [56]. *LP-MOGA* se propone en [57] para la selección de atributos en la identificación de fracturas por fatiga, que es un método híbrido entre *NSGA-II* y predicción lineal, para minimizar la tasa de errores, tasa de falsos negativos y el número de atributos seleccionados. En [58] se aplica un sistema inmune artificial bayesiano multi-objetivo (*MOBAIS*) a la selección de atributos en los problemas de clasificación con el objetivo de minimizar tanto el error de clasificación como la cardinalidad del subconjunto de atributos. El operador de mutación tradicional se reemplaza por un modelo probabilístico que representa la distribución de probabilidad de las soluciones más prometedoras encontradas hasta el momento. En [59] se utiliza *SMS-EMOA* [60] en tareas de reconocimiento de género y estilo musical con dos combinaciones de objetivos: recall y especificidad, y accuracy y tasa de las atributos seleccionados.

En [61] se propone un método wrapper para maximizar la precisión de clasificadores *J48* y minimizar la cardinalidad del subconjunto de atributos utilizando *NSGA-II* aplicado a la mortalidad por infección en pacientes de quemaduras severas. En [62] se propone un método wrapper para optimizar la tasa de error y el tamaño del árbol de decisión construido por *CART* o el tamaño de la Feedforward Neural Network, usando un algoritmo evolutivo multi-objetivo elitista basado en [8] y *NSGA*. En [63] se propone un algoritmo de estimación de distribución multi-objetivo para selección de atributos basada en el modelado conjunto de objetivos y variables, denominado multidimensional bayesian network-based estimation of distribution algorithm (*MBN-EDA*). Se utilizan seis medidas de rendimiento diferentes para los clasificadores, y adoptan un enfoque wrapper para evaluar subconjuntos de atributos utilizando naive Bayes y clasificadores tree-augmented naive Bayes. En [64] se propone el *3DCH-EMOA*, un algoritmo de optimización multi-objetivo para maximizar ROC convex hull [65] (*ROCCH*) en 3-D. *3DCH-EMOA* se compara con *NSGA-II*, *GDE3*, *SPEA2*, *MOEA/D* y *SMS-EMOA* en los experimentos. In [66] *RPSGAe* se utiliza para minimizar el número de atributos y maximizar la calidad del clasificador, aplicado a la predicción de bancarrota. En [67] se propone un enfoque de optimización multiobjetivo paralelo para hacer frente a problemas de selección de atributos de alta dimensión. Se proponen varias alternativas evolutivas multi-objetivo paralelas y se evalúan experimentalmente utilizando algunos puntos de referencia sintéticos y *BCI* (Brain-Computer Interface).

La *Evolución Diferencial Multi-objetivo* [68] también se ha aplicado con éxito para selección de atributos en los últimos años. En [69], se ha aplicado un método wrapper (*MODE*) para el reconocimiento de entidades nombradas en textos biomédicos utilizando un clasificador ensemble y F-measure y el número de atributos seleccionados como objetivos, y se compara con los sistemas de reconocimiento de entidades nombradas existentes utilizando los mismos conjuntos de datos. En [70] se ha propuesto un filter (*FAEMODE*) que utiliza un algoritmo de evolución diferencial elitista multi-objetivo para selección de atributos. Se maximiza la dependencia del subconjunto de atributos con la clase objetivo y se minimiza la redundancia de atributos. Los resultados se compararon con los métodos wrapper y filter. Por último, se ha propuesto otro enfoque multi-objetivo basado en la evolución diferencial (*DEMO*) en [71] como un método wrapper de selección de atributos para el reconocimiento de la expresión facial. El número de atributos utilizados y la precisión de reconocimiento de emociones de los clasificadores de máquinas de soporte vectorial se optimizaron simultáneamente. Los resultados han sido comparados con los métodos más modernos, donde se incluye *NSGA-II*.

3.1.3. Sistemas de clasificación evolutivos multi-objetivo basados en reglas fuzzy

La *interpretabilidad de los sistemas de clasificación* se refiere a la capacidad de estos para expresar su comportamiento de una manera que sea fácilmente comprensible para un usuario. Los modelos de clasificación interpretables permiten la validación externa por parte de un experto y, en ciertas disciplinas como la medicina o el negocio, proporcionar información sobre la toma de decisiones es esencial por razones éticas y humanas. Los *sistemas de clasificación basados en reglas fuzzy* [72, 73, 74, 75] se han desarrollado ampliamente en los últimos años y ahora están consolidados como potentes herramientas de clasificación que también permiten la interpretación del modelo de forma directa y clara, ya que utilizan etiquetas lingüísticas de manera similar a como lo hace el razonamiento humano. *Los algoritmos evolutivos multi-objetivo* se han aplicado con éxito en los últimos años para la optimización de *sistemas de clasificación basados en reglas fuzzy*. En [76] se utiliza un algoritmo evolutivo multi-objetivo para aprender simultáneamente la base de reglas y la base de datos de un sistema de clasificación basada en reglas fuzzy. En este caso se consideran dos objetivos: el primero mide la complejidad como la suma de las etiquetas de

variables de entrada utilizadas en cada una de las reglas, y el segundo corresponde al mean squared error. En [77], se generan clasificadores fuzzy para conjuntos de datos desbalanceados y sensibles al costo con un algoritmo evolutivo multi-objetivo de tres objetivos. Los objetivos primero y segundo son sensibilidad y especificidad. El tercer objetivo es una medida de complejidad computada como la suma de las condiciones que componen los antecedentes de las reglas, la cual se minimiza. En el método [78], *PAES-RCS* se utiliza para maximizar la precisión y minimizar el tamaño del conjunto de reglas para la clasificación del tráfico en Internet. En [79], *IT2-PAES-RCS* extiende *PAES-RCS* para emplear conjuntos fuzzy de Tipo 2, donde la sensibilidad, especificidad y el tamaño del conjunto de reglas se optimizan para la clasificación de datos financieros. En [27] se propone un sistema evolutivo de clasificación basado en reglas fuzzy basado en optimización multi-objetivo de parámetros reales con restricciones, que maximiza la precisión y minimiza el número de reglas, e impone una restricción para la similaridad de los conjuntos fuzzy. El número máximo de reglas del modelo, el número máximo de etiquetas lingüísticas para cada variable, y la máxima similaridad de los conjuntos fuzzy, son paramtrizables para que puedan ser establecidos por un usuario con el fin de obtener modelos compactos. Una vez extraído el conjunto de reglas fuzzy, un proceso de etiquetado lingüístico final asigna una etiqueta lingüística a cada conjunto fuzzy. Además, este método de clasificación fuzzy es, en sí mismo, un método de selección de atributos, ya que detecta atributos 'don't care conditions' que pueden ser eliminados del modelo de clasificación.

En [27], el problema se formula como una instancia del problema (1), con $l = 2$ (dos objetivos) y $m = 4$ (cuatro restricciones):

$$\begin{array}{ll} Max./Min. & \mathcal{F}_{\mathcal{D}}(\mathbf{\Gamma}) \\ Min. & \mathcal{NR}(\mathbf{\Gamma}) \\ subject\ to: & \mathcal{NR}(\mathbf{\Gamma}) \geq M_{min} \\ & \mathcal{NR}(\mathbf{\Gamma}) \leq M_{max} \\ & \mathcal{NL}(\mathbf{\Gamma}) \leq L_{max} \\ & \mathcal{S}(\mathbf{\Gamma}) \leq g_s \end{array} \tag{3}$$

donde $\mathbf{\Gamma}$ es un clasificador basado en reglas fuzzy compuesto por $\mathcal{NR}(\mathbf{\Gamma})$ reglas fuzzy. Cada regla fuzzy R_j^{Γ}, $j = 1, \ldots, \mathcal{NR}(\mathbf{\Gamma})$ tiene la siguiente estructura:

$$\begin{array}{lll} R_j^{\Gamma}: & if\ x_1\ es\ A_{1j}^{\Gamma} \wedge \ldots \wedge x_p\ es\ A_{pj}^{\Gamma} \wedge & \\ & y_1\ es\ B_{1j}^{\Gamma} \wedge \ldots \wedge y_q\ es\ B_{qj}^{\Gamma} & \rightarrow z\ es\ C_j^{\Gamma}, \end{array}$$

donde $x_i \in [l_i, u_i] \subset \mathbb{R}$, $i = 1, \ldots, p$, $p \geq 0$, son atributos de entrada reales, $y_i \in \{1, \ldots, v_i\}$, $i = 1, \ldots, q$, $q \geq 0$, $v_i > 1$, son atributos de entrada categóricos, y $z \in \{1, \ldots, w\}$, $w > 1$ es un atributo de salida nominal. Cada conjunto fuzzy A_{ij}^{Γ}, $i = 1, \ldots, p$, $j = 1, \ldots, \mathcal{NR}(\mathbf{\Gamma})$ se define con una *gaussian membership function* [80]. En el problema (3), la función $\mathcal{F}_{\mathcal{D}}(\mathbf{\Gamma})$ es una medida de rendimiento del clasificador $\mathbf{\Gamma}$ sobre la base de datos $\mathcal{D}$. La función $\mathcal{NR}(\mathbf{\Gamma})$ se minimiza, y las restricciones $\mathcal{NR}(\mathbf{\Gamma}) \geq M_{min}$ y $\mathcal{NR}(\mathbf{\Gamma}) \leq M_{max}$ limitan el número de reglas del clasificador $\mathbf{\Gamma}$ al intervalo $[M_{min}, M_{max}]$ (M_{min} se fija al número de clases del atributo de salida, mientras que M_{max} es dado por el usuario). La restricción $\mathcal{NL}(\mathbf{\Gamma}) \leq L_{max}$ limita el número de etiquetas lingüísticas de las variables de entrada reales a L_{max}. Por último, la restricción $\mathcal{S}(\mathbf{\Gamma}) \leq g_s$ asegura la similaridad máxima g_s ($0 < gs \leq 1$) entre los conjuntos fuzzy; el valor de similaridad de un clasificador $\mathbf{\Gamma}$ representa el valor máximo de superposición entre sus conjuntos fuzzy para cualquier variable de entrada. La restricción $\mathcal{S}(\mathbf{\Gamma}) \leq g_s$ es gestionada por el algoritmo evolutivo multi-objetivo por medio de un algoritmo de reparo, que se aplica después de la inicialización de las soluciones, y después del cruce y la mutación.

Como método de razonamiento usamos la *coincidencia máxima* donde, el *grado de compatibilidad* de la regla R_j^{Γ} para el ejemplo $(\mathbf{x}, \mathbf{y})$ se calcula como:

$$\varphi_j^{\Gamma}(\mathbf{x}, \mathbf{y}) = \left(\phi_j^{\Gamma}(\mathbf{y}) + 1\right) \prod_{i=1}^{p} \mu_{\widetilde{A}_{ij}^{\Gamma}}(x_i)$$

donde $\phi_j^{\Gamma}(\mathbf{y})$ es el número de atributos de entrada categóricos, tal que $y_i = B_{ij}^{\Gamma}$. El *grado de compatibilidad* se obtiene aplicando un producto de t-norma al grado de compatibilidad de las cláusulas $x_i\ \ es\ A_{ij}^{\Gamma}$ multiplicado por el número de coincidencias de los datos de entrada categóricos $y_i\ \ es\ B_{ij}^{\Gamma}$. El *grado de asociación* del ejemplo $(\mathbf{x}, \mathbf{y})$ con la clase C, se calcula sumando los grados de compatibilidad de cada

regla R_j^Γ cuyo valor para el atributo de salida categórica C_j^Γ es igual a C, es decir:

$$\lambda_C^\Gamma(\mathbf{x},\mathbf{y}) = \sum_{\substack{j=1,\ldots,M_\Gamma \\ C_j^\Gamma = C}} \varphi_j^\Gamma(\mathbf{x},\mathbf{y})$$

La *clasificación* del ejemplo $(\mathbf{x},\mathbf{y})$ o salida del clasificador Γ, corresponde a la clase C cuyo grado de asociación es máximo, es decir:

$$f_\Gamma(\mathbf{x},\mathbf{y}) = \arg_C \overset{w}{\underset{C=1}{\max}} \lambda_C^\Gamma(\mathbf{x},\mathbf{y})$$

3.2. Herramientas de desarrollo

Para llevar a cabo los objetivos de la tesis, el uso de la herramienta *Weka* ha sido fundamental. *Weka* es una colección de algoritmos de aprendizaje automático para tareas de minería de datos. Contiene herramientas para la preparación de datos, clasificación, regresión, agrupación, extracción de reglas de asociación y visualización. *Weka* es un software *open source* emitido bajo la *Licencia Pública General de GNU*. *Weka* permite a sus usuarios, no sólo poder usar un conjunto de métodos básicos de machine learning incorporados a la herramienta por su equipo de trabajo, sino también poder usar los métodos incorporados por los usuarios/desarrolladores de la herramienta. Cuando un método de machine learning se incorpora a la plataforma de *Weka*, éste queda integrado teniendo acceso a todos los componentes de la herramienta, como son los filtros de *pre-procesamiento*, los métodos de *selección de atributos*, los algoritmos de *clasificación*, *regresión*, *clustering* y *reglas de asociación*, los mecanismos de *evaluación* (*full training*, *supplied test set*, *cross-validation* y *percentage split*) y las herramientas de *visualización*, además de un *entorno de experimentación* que permite hacer test estadísticos paramétricos sobre un amplio conjunto de métricas de evaluación.

Por tanto, un paso crucial en la metodología de la tesis ha sido incorporar los algoritmos propuestos en la plataforma de *Weka*. Concretamente se han incluido, como paquetes oficiales, los paquetes *MultiObjetiveEvolutionarySearch* y *MultiObjectiveEvolutionaryFuzzyClassifier*. La clase *MultiObjetiveEvolutionarySearch* implementa las estrategias de búsqueda *ENORA* y *NSGA-II* para selección de atributos. La clase *MultiObjectiveEvolutionaryFuzzyClassifier* es un clasificador basado en reglas fuzzy el cual permite tanto datos numéricos como categóricos, pudiendo por tanto ser usado también para clasificación basada en reglas (crisp) con datos categóricos y que está implementado también con los algoritmos evolutivos multi-objetivo *ENORA* y *NSGA-II*.

Otra de las ventajas de *Weka* es que, además de la interfaz gráfica, permite desarrollar programas *Java* usando la *API* de *Weka*. De esta forma podemos enlazar todo el proceso de análisis de datos mediante un programa y ejecutarlo en un lote. En esta tesis se ha utilizado la infraestructura de computación facilitada por el *Centro Extremeño de Tecnologías Avanzadas* (*CETA-CIEMAT*), con un computador con 8 procesadores *Intel Xeon X7550 @ 2.00 GHz*, *RAM1TByte* a *1067MHz*, almacenamiento *Lustre Distributed File System v2.5.2* y red de interconexión *Infiniband QDR* (*40Gbps*). Los múltiples procesos requeridos en los experimentos han sido ejecutados en paralelo, paliando de esta forma, en cierta medida, los inconvenientes en el coste de ejecución que sufren los lenguajes interpretados como es *Java*.

Además de la herramienta *Weka*, para la consecución de los objetivos de esta tesis se ha requerido del software *Caret R* para la realización de tests estadísticos, así como del lenguaje *MathLab* para la generación de gráficos a partir de datos y del lenguaje LaTeX para la generación de documentos.

3.3. Una metodología para evaluar métodos evolutivos multi-objetivo de selección de atributos para tareas de clasificación en el contexto del screening virtual

Se ha propuesto una metodología que incluye, para cada base de datos *tk* y *mr* [81], preprocesamiento de los datos, selección de atributos, toma de decisiones, comparación de rendimiento de los optimizadores (basada en la métrica de hipervolumen), evaluación del modelo de clasificación, visualización y tests estadísticos. La metodología propuesta se muestra gráficamente en la Figura 3.

Preprocesamiento: Los atributos que muestran una pequeña varianza se eliminan en el paso de preprocesamiento. El procedimiento *nearZeroVar* de Caret R [82] se ha utilizado para esta tarea. Como resultado, la base de datos *tk* se reduce a 160 atributos y la base de datos *mr* se reduce a 153 atributos.

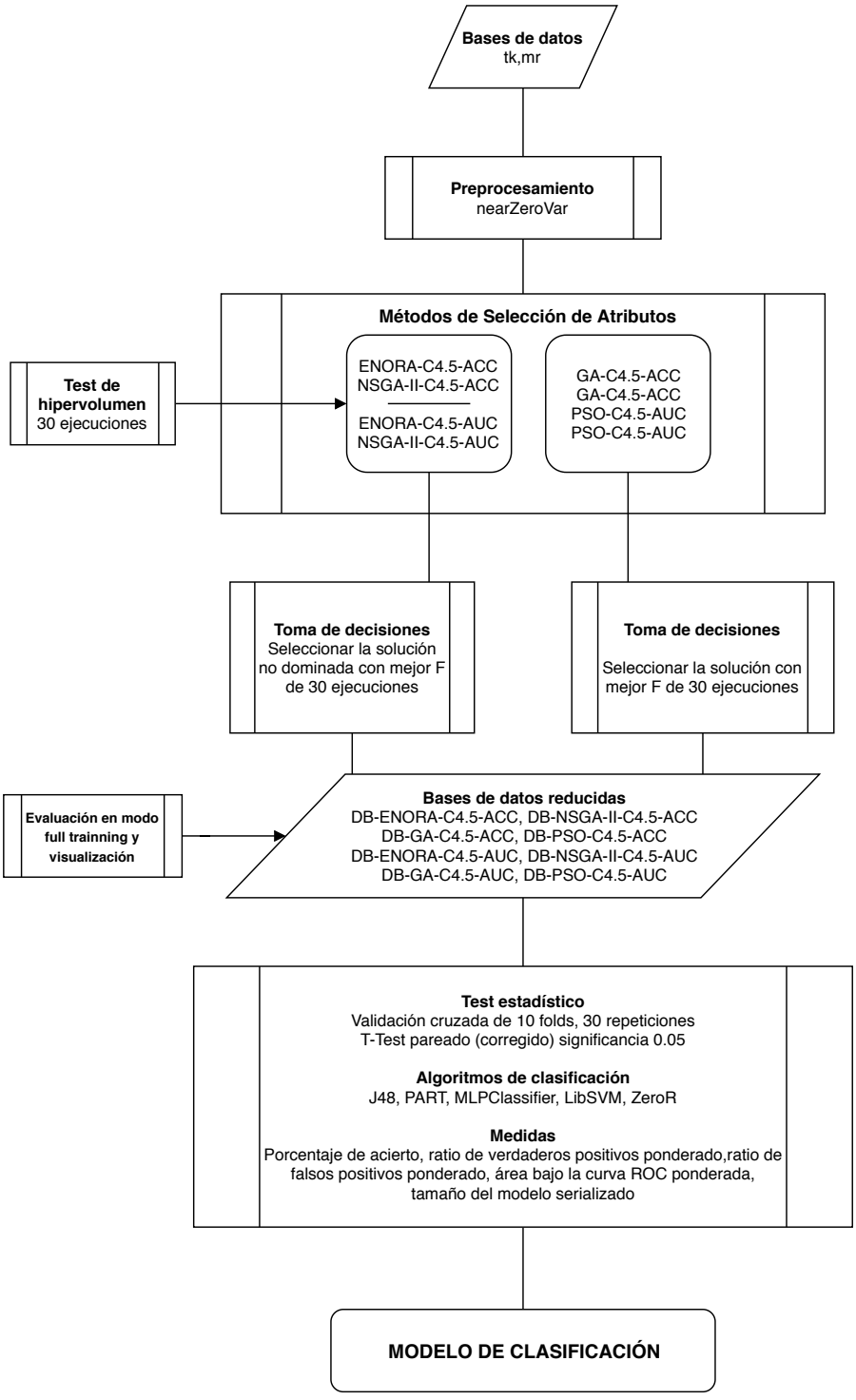


Figura 3: Metodología propuesta para evaluar métodos evolutivos multi-objetivo de selección de atributos para tareas de clasificación en el contexto del screening virtual.

Selección de atributos y toma de decisiones: Los siguientes modelos de optimización han sido considerados para selección de atributos:

1. Modelo de optimización multi-objetivo con función objetivo basada en *accuracy*:

$$\begin{aligned} &Minimizar \quad \mathcal{F}(\mathbf{x}) = 1 - \mathcal{ACC}(\mathbf{x}) \\ &Minimizar \quad \mathcal{C}(\mathbf{x}) = \sum_{k=1}^{w} \mathcal{N}(x_k) \end{aligned} \tag{4}$$

   $\mathcal{ACC}(\mathbf{x})$ es la proporción de resultados verdaderos (tanto verdaderos positivos como falsos positivos) entre el número total de instancias examinadas (*accuracy* [83]) obtenida con un clasificador sobre la base de datos con los atributos seleccionados en $\mathbf{x}$.

$$\mathcal{ACC}(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} T(\mathbf{x}, i)$$

   M es el número de instancias de la base de datos y $T(\mathbf{x}, i)$ es el resultado de la clasificación de la instancia i en la base de datos con los atributos seleccionados en $\mathbf{x}$:

$$T(\mathbf{x}, i) = \begin{cases} 1 & if\ \hat{y}_i = y_i \\ 0 & if\ \hat{y}_i \neq y_i \end{cases}$$

   donde $\hat{y}_i$ es el valor predicho de la instancia i-ésima y y_i es el valor verdadero correspondiente. Se usa un el árbol de decisión generado con el algoritmo *C4.5* por razones de interpretabilidad. *C4.5* [84] como una mejora del algoritmo *ID3* para generar un árbol de decisión para clasificación. Es computacionalmente muy eficiente y puede ser interpretable cuando el tamaño del árbol no es excesivamente grande.

2. Modelo de optimización multi-objetivo con función objetivo basada en el *área bajo la curva ROC*:

$$\begin{aligned} &Minimizar \quad \mathcal{F}(\mathbf{x}) = 1 - \mathcal{AUC}(\mathbf{x}) \\ &Minimizar \quad \mathcal{C}(\mathbf{x}) = \sum_{k=1}^{w} \mathcal{N}(x_k) \end{aligned} \tag{5}$$

   $\mathcal{AUC}(\mathbf{x})$ es el *área bajo la curva ROC* de un clasificador (se ha usado *C4.5* como en el modelo (4)). La curva *ROC* (*Receiver Operating Characteristic*) [65] es una representación gráfica de la *sensibilidad* contra la *especificidad* para un sistema clasificador de acuerdo con una variación del *umbral de decisión*. La curva *ROC* se puede utilizar para generar estadísticas que resumen el rendimiento de la eficacia de un clasificador. En [83] se muestra que la curva *ROC* es una descripción empírica simple pero completa del efecto de variación del umbral de decisión, la cual indica todas las combinaciones posibles de las frecuencias relativas de los diversos tipos de decisiones correctas e incorrectas. El área bajo la curva *ROC* se puede formular matemáticamente de la siguiente manera [85]:

$$\mathcal{AUC}(\mathbf{x}) = \int_0^1 F_0(\mathbf{x}, F_1^{-1}(\mathbf{x}, v))dv$$

   donde $F_0(\mathbf{x}, t)$ (sensibilidad) es la proporción de instancias positivas clasificadas como positivas obtenidas con el clasificador en la base de datos con los atributos seleccionados en $\mathbf{x}$, $1 - F_1(\mathbf{x}, t)$ (especificidad) es la proporción de instancias negativas clasificadas como negativas obtenidas con el clasificador en la base de datos con los atributos seleccionados en $\mathbf{x}$, y t es el umbral de decisión.

3. Modelo de optimización con un solo objetivo basado en *accuracy*:

$$Minimizar \quad \mathcal{F}(\mathbf{x}) = 1 - \mathcal{ACC}(\mathbf{x}) \tag{6}$$

4. Modelo de optimización con un solo objetivo basado en el área bajo la curva *ROC*:

$$Minimizar \quad \mathcal{F}(\mathbf{x}) = 1 - \mathcal{AUC}(\mathbf{x}) \tag{7}$$

Usamos el algoritmo *C4.5* tanto para la clasificación en los modelos de optimización (6) y (7) como en los modelos de optimización (4) y (5). Los modelos de optimización (4) y (5) se han resuelto utilizando las estrategias de búsqueda *ENORA* y *NSGA-II*, mientras que los modelos de optimización (6) y (7) se han resuelto con otras dos metaheurísticas basadas en poblaciones: *Genetic Algorithms* [36] y *Particle Swarm Optimization* (*PSO*) [86]. Por lo tanto, se han considerado un total de ocho métodos de selección de atributos, que se resumen en la tabla 1.

Nombre	**Modelos de optimización**
ENORA-C4.5-ACC	Eq. (4)
NSGA-II-C4.5-ACC	Eq. (4)
GA-C4.5-ACC	Eq. (6)
PSO-C4.5-ACC	Eq. (6)
ENORA-C4.5-AUC	Eq. (5)
NSGA-II-C4.5-AUC	Eq. (5)
GA-C4.5-AUC	Eq. (7)
PSO-C4.5-AUC	Eq. (7)

Tabla 1: Nombres de los métodos de selección de atributos y sus modeles de optimización.

Toma de decisiones y bases de datos reducidas: Los métodos de selección de atributos propuestos en este apartado utilizan estrategias de búsqueda probabilísticas, por lo que requieren múltiples ejecuciones con diferentes semillas para la generación de números aleatorios. *ENORA* y *NSGA-II* son estrategias de búsqueda evolutiva multi-objetivo y requieren un proceso de toma de decisiones final para elegir una solución entre las soluciones no dominadas de la población final. Las soluciones no dominadas en la última población de cada ejecución se identifican y la solución con el mejor valor de $\mathcal{F}$ es la elección. El proceso de toma de decisiones para las estrategias de búsqueda *GA* y *PSO* consiste en seleccionar la solución con el mejor valor de $\mathcal{F}$ entre las últimas poblaciones. Cada método de selección de atributos genera una nueva base de datos reducida con los atributos seleccionados.

Test de hipervolumen: En este paso usamos la métrica de *hipervolumen* para comparar estadísticamente el rendimiento de las estrategias de búsqueda *ENORA* y *NSGA-II*. Elegimos la métrica de hipervolumen porque mide tanto la diversidad como la optimalidad de las soluciones no dominadas. Además, la métrica de hipervolumen no requiere el uso de una población óptima, como otras métricas de rendimiento tales como *error ratio*, *generational distance*, *maximum Pareto-optimal front error*, *spread*, *maximum spread*, o *chi-square-like deviation* [8].

La métrica de hipervolumen se define en [8] como el volumen del espacio de búsqueda dominado por una población P, formulado como:

$$H\left(P\right) = \bigcup_{i=1}^{|Q|} v_i$$

donde $Q \subseteq P$ es el conjunto de individuos no dominados de P y v_i es el volumen del individuo i. El *ratio de hipervolumen* se define como la proporción del volumen del espacio de búsqueda no dominado sobre el volumen total del espacio de búsqueda:

$$HR\left(P\right) = 1 - \frac{H\left(P\right)}{VS}$$

donde VS es el volumen total del espacio de búsqueda. El ratio de hipervolumen requiere puntos de referencia que identifican los valores máximo y mínimo para cada objetivo. Para selección de atributos en problemas de clasificación se establecen los siguientes valores mínimos ($\mathcal{F}_{lower}$, $\mathcal{C}_{lower}$) y máximos ($\mathcal{F}_{upper}$, $\mathcal{C}_{upper}$) para cada objetivo (en los modelos de optimización multi-objetivo (4) y (5)):

$$\mathcal{F}_{lower} = 0, \quad \mathcal{F}_{upper} = 1, \quad \mathcal{C}_{lower} = 1, \quad \mathcal{C}_{upper} = w$$

Evaluación en modo full training: En este paso, mostramos los resultados de la evaluación en modo full training con *J48* de las bases de datos reducidas obtenidas con los ocho métodos de selección de atributos y las bases de datos originales *tk* y *mr* usando las métricas estándar proporcionadas por el

paquete *Weka*. Estas medidas son *Percent Correct* ($accuracy \times 100$), *True Positive Rate*, *False Positive Rate*, *Precision*, *Recall*, *F-measure*, *Mathews Correlation Coefficient*, *Area Under ROC Curve* y *Area Under Precision Recall Curve*. Todas estas métricas, excepto *percent correct*, se calculan por clase y promedio ponderado.

Validación cruzada y tests estadísticos: El siguiente paso es testar y comparar las bases de datos reducidas. La eficacia de un modelo está determinada, no por su desempeño en los datos de entrenamiento, sino por su capacidad de predicción en datos vistos por primera vez.

El *overfitting* ocurre cuando un modelo comienza a memorizar datos de entrenamiento en lugar de aprender a generalizar a partir de tendencias. La *validación cruzada* [87] es una forma de predecir el ajuste de un modelo a un conjunto de validación hipotético cuando un conjunto de validación explícito no está disponible. El problema con las evaluaciones en los datos de entrenamiento es que no dan una indicación de cómo de bien se desempeñará el modelo cuando se le pida que haga nuevas predicciones para los datos que aún no han visto. En *validación cruzada de k folds*, la base de datos se divide en k partes y el método fijado se repite k veces. Cada vez, uno de los k subconjuntos se utiliza como conjunto de test y los otros $k-1$ subconjuntos se juntan para formar un conjunto de entrenamiento. Luego se calcula el error promedio en todos los k subconjuntos. Para probar los clasificadores obtenidos con las bases de datos reducidas, *Weka experimenter tool* se configuró utilizando las bases de datos reducidas más el conjunto de datos original, con validación cruzada de 10 fols y 30 iteraciones.

Los siguientes algoritmos de clasificación fueron utilizados para estas pruebas:

- *J48*: Esta prueba es interesante porque nuestros métodos de selección de atributos usaron *J48* como evaluador del método wrapper. *J48* es una implementación Java de código abierto del algoritmo *C4.5* en la herramienta *Weka*.

- *PART*: Este algoritmo de clasificación se incluye dentro de los clasificadores basados en reglas y, por lo tanto, los modelos también pueden ser interpretados por un experto. Utiliza divide y vencerás para construir un árbol de decisión *C4.5* parcial en cada iteración y convierte la mejor hoja en una regla.

- *MLP*: Este algoritmo de clasificación entrena un perceptrón multicapa con una capa oculta minimizando el error cuadrático más una penalización cuadrática con el método *BFGS*.

- *SVM*: Este algoritmo implementa el algoritmo *sequential minimal optimization* (*SMO*) para máquinas de soporte vectorial kernelizadas, ejecutadas con la función *LibSVM* de *Weka*. Este algoritmo es parte de un grupo de clasificadores basados en funciones y hemos utilizado la función de base radial (*RBF*) como tipo de kernel.

- *ZeroR*: Este algoritmo de clasificación se incluye dentro de los clasificadores basados en reglas y es el método de clasificación más simple que se basa en la clase e ignora todos los predictores. El clasificador *ZeroR* simplemente predice la categoría mayoritaria (clase). Aunque no hay poder predictivo en *ZeroR*, es útil para determinar un rendimiento como punto de referencia para otros métodos de clasificación.

Para analizar los resultados de estos experimentos, realizamos *Paired T-Test (corrected)* [88] con una significancia de 0,05. *Corrected Paired T-Test* es una versión corregida de *Paired T-Test* implementada en *Weka* para evitar algunos problemas de la prueba original con validación cruzada. Las bases de datos obtenidas con la estrategia de búsqueda *ENORA* fueron el test base y las métricas *Percent_correct*, *Weighted_avg_true_positive_rate*, *Weighted_avg_false_positive_rate*, *Weighted_avg _area_under_ROC*, *Kappa_statistic* y *Serialized_Model_Size* fueron los campos de comparación.

3.4. Clasificación evolutiva multi-objetivo basada en reglas con datos categóricos

En este apartado, se propone un enfoque para clasificación basada en reglas con datos categóricos mediante optimización multi-objetivo con restricciones.

Clasificación basada en reglas para datos categóricos

Sea Γ un clasificador compuesto por las M reglas, donde cada regla R_i^Γ, $i = 1, \ldots, M$, tiene la siguiente estructura:

$$R_i^\Gamma : \quad IF \quad x_1 = b_{i1}^\Gamma \quad AND \quad \ldots \quad AND \quad x_p = b_{ip}^\Gamma \quad THEN \quad y = c_i^\Gamma \tag{8}$$

donde para $j = 1, \ldots, p$ el atributo b_{ij}^Γ (llamado *antecedente*) toma valores en un conjunto $\{1, \ldots, v_j\}$ ($v_j > 1$), y donde c_i^Γ (llamado *consecuente*) toma valores en $\{1, \ldots, w\}$ ($w > 1$). Sea $\mathbf{x} = \{x_1, \ldots, x_p\}$ un ejemplo observado, con $x_j \in \{1, \ldots, v_j\}$, para cada $j = 1, \ldots, p$. Proponemos *maximum matching* como *método de razonamiento*, donde el *grado de compatibilidad* de la reglas R_i^Γ para la muestra $\mathbf{x}$ (denotado por $\varphi_i^\Gamma(\mathbf{x})$), se calcula como el número de atributos cuyo valor coincide con el del antecedente correspondiente en R_i^Γ, es decir:

$$\varphi_i^\Gamma(\mathbf{x}) = \sum_{j=1}^{p} \mu_{ij}^\Gamma(\mathbf{x})$$

donde:

$$\mu_{ij}^\Gamma(\mathbf{x}) = \begin{cases} 1 & if\ x_j = b_{ij}^\Gamma \\ 0 & if\ x_j \neq b_{ij}^\Gamma \end{cases}$$

El *grado de asociación* para la muestra $\mathbf{x}$ con una clase $c \in \{1, \ldots, w\}$ se calcula sumando los grados de compatibilidad para la muestra $\mathbf{x}$ de cada regla R_i^Γ cuyo consecuente c_i^Γ es igual a la clase c, es decir:

$$\lambda_c^\Gamma(\mathbf{x}) = \sum_{i=1}^{M} \eta_{ic}^\Gamma(\mathbf{x})$$

donde:

$$\eta_{ic}^\Gamma(\mathbf{x}) = \begin{cases} \varphi_i^\Gamma(\mathbf{x}) & if\ c = c_i^\Gamma \\ 0 & if\ c \neq c_i^\Gamma \end{cases}$$

Por lo tanto, la *clasificación* (o salida) del clasificador Γ para la muestra $\mathbf{x}$ corresponde a la clase cuyo grado de asociación es máximo, es decir:

$$f^\Gamma(\mathbf{x}) = \arg_c \max_{c=1}^{w} \lambda_c^\Gamma(\mathbf{x})$$

Un enfoque basado en optimización multi-objetivo con restricciones

Sea $\mathcal{D}$ una base de datos de K instancias con p atributos de entrada categóricos, $p > 0$, y un atributo de salida categórico. Cada atributo de entrada j puede tomar una categoría $x_j \in \{1, \ldots, v_j\}$, $v_j > 1$, $j = 1, \ldots, p$, y el atributo de salida puede tomar una clase $c \in \{1, \ldots, w\}$, $w > 1$. El problema de encontrar un clasificador óptimo Γ, como se describe en la sección anterior, se puede formular como una instancia del problema de optimización multi-objetivo con restricciones (1) con dos objetivos y dos restricciones:

$$\begin{array}{ll} Max./Min. & \mathcal{F}_\mathcal{D}(\Gamma) \\ Min. & \mathcal{NR}(\Gamma) \\ subject\ to: & \mathcal{NR}(\Gamma) \geq w \\ & \mathcal{NR}(\Gamma) \leq M_{max} \end{array} \tag{9}$$

En el problema (9), la función $\mathcal{F}_\mathcal{D}(\Gamma)$ es una medida de rendimiento del clasificador Γ sobre el conjunto de datos $\mathcal{D}$, la función $\mathcal{NR}(\Gamma)$ es el número de reglas del clasificador Γ, y las restricciones $\mathcal{NR}(\Gamma) \geq w$ y $\mathcal{NR}(\Gamma) \leq M_{max}$ limitan el número de reglas del clasificador Γ al intervalo $[w, M_{max}]$, donde w es el número de clases del atributo de salida y M_{max} es dado por el usuario. Ambos objetivos $\mathcal{F}_\mathcal{D}(\Gamma)$ y $\mathcal{NR}(\Gamma)$ están en conflicto. Cuantas menos reglas tenga el clasificador, menor será el número de instancias que puede cubrir, es decir, si el clasificador es más simple, tendrá menos capacidad de predicción. Por lo tanto, existe un conflicto intrínseco entre los objetivos del problema (por ejemplo, maximizar la precisión y minimizar la complejidad del modelo) por lo que no se pueden agregar fácilmente en un solo objetivo. Normalmente, ambos objetivos se optimizan simultáneamente en muchos otros sistemas de clasificación, como las redes neuronales o los árboles de decisión [89, 90].

Tanto *ENORA* como *NSGA-II* han sido adaptados para (9) con representación de longitud variable basada en un enfoque de *Pittsburgh*, *inicialización aleatoria uniforme*, *selección por torneo binario*, *manejo de restricciones* basada en representación *ad hoc*, ranking basado en el *nivel de no-dominación* con distancia de *crowding* como medida de aglomeración, y *operadores de variación adaptativos*. Los operadores de variación adaptativos trabajan en diferentes niveles del clasificador: *cruce de reglas*, *cruce de reglas incremental*, *mutación incremental de reglas* y *mutación entera*.

Representación: Usamos una representación de longitud variable basada en un enfoque de Pittsburgh [91], donde cada individuo I de una población contiene un número variable de reglas M_I. Cada regla R_i^I, $i = 1, \ldots,$ se codifica en los siguientes componentes:

- Valores enteros asociados a los antecedentes $b_{ij}^I \in \{1, \ldots, v_j\}$, para $i = 1, \ldots, M_I$ y $j = 1, \ldots, p$.
- Valores enteros asociados al consecuente $c_i^I \in \{1, \ldots, w\}$, para $i = 1, \ldots, M_I$.

Además, para llevar a cabo el cruce y la mutación adaptativos, cada individuo tiene dos parámetros discretos $d_I \in \{0, \ldots, \delta\}$ y $e_I \in \{0, \ldots, \epsilon\}$ asociados con el cruce y la mutación, donde $\delta \geq 0$ es el número de operadores de cruce y $\varepsilon \geq 0$ es el número de operadores de mutación. Los valores d_I y e_I para la variación adaptativa se generan aleatoriamente en sus correspondientes dominios. La tabla 2 resume la representación de un individuo.

Codificación del conjunto de reglas					**Codificación para cruce adaptativo y mutación**	
Antecedentes				**Consecuente**	**Cruce asociado**	**Mutación asociada**
b_{11}^I	b_{21}^I	$\ldots$	b_{q1}^I	c_1^I		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	d_I	e_I
$b_{1M_I}^I$	$b_{2M_I}^I$	$\ldots$	$b_{qM_I}^I$	$c_{M_I}^I$		

Tabla 2: Codificación de cromosomas para un individuo I.

Manejo de restricciones: Las restricciones $\mathcal{NR}(\mathbf{\Gamma}) \geq w$ y $\mathcal{NR}(\mathbf{\Gamma}) \leq M_{max}$ se satisfacen mediante operadores especializados de inicialización y variación, que siempre generan individuos con una serie de reglas entre w y M_{max}.

Población inicial: La población inicial (Algoritmo 5) se genera aleatoriamente con las siguientes condiciones:

- Los individuos se distribuyen uniformemente con respecto al número de reglas con valores entre w y M_{max}, y con una restricción adicional que especifica que debe haber al menos un individuo para cada número de reglas (pasos del 4 al 8). Esto asegura una adecuada diversidad inicial en el espacio de búsqueda para el segundo objetivo del modelo de optimización.
- Todos los individuos contienen al menos una regla para toda clase de salida entre 1 y w (pasos del 16 al 20).

Evaluación: Dado que el modelo de optimización abarca dos objetivos, cada individuo debe ser evaluado con dos funciones de fitness. Estas corresponden a las funciones objetivo $\mathcal{F}_{\mathcal{D}}(\Gamma)$ y $\mathcal{NR}(\Gamma)$ del problema (9). La selección de los mejores individuos se realiza mediante el concepto de Pareto en un torneo binario.

Operadores de variación: Tanto *ENORA* como *NSGA-II* se han implementado con dos operadores de cruce, *cruce de reglas* (Algoritmo 9) y *cruce de reglas incremental* (Algoritmo 10), y dos operadores de mutación, *mutación incremental de reglas* (Algoritmo 11) y *mutación entera* (Algoritmo 12). El cruce de reglas intercambia aleatoriamente dos reglas seleccionadas de los padres, y la cruce de reglas incremental agrega a cada padre una regla seleccionada al azar del otro padre si su número de reglas es menor que el número máximo de reglas. Por otro lado, la mutación incremental agrega una nueva regla al individuo si el número de reglas del individuo es menor que el número máximo de reglas, mientras que la mutación

entera realiza una mutación uniforme de un antecedente aleatorio que pertenece a una regla seleccionada al azar.

Utilizamos *cruce y mutación adaptativos*, lo que significa que la selección de los operadores se realiza mediante una técnica adaptativa. Cada Individuo I tiene dos parámetros enteros $d_I \in \{0, \ldots, \delta\}$ y $e_I \in \{0, \ldots, \epsilon\}$ para indicar qué cruce o mutación se lleva a cabo. En nuestro caso, $\delta = 2$ y $\epsilon = 2$ (dos operadores de cruce y dos operadores de mutación), así que $d_I, e_I \in \{0, 1, 2\}$). El valor 0 indica que no se realiza cruce o mutación. La variación adaptativa (Algoritmo 6) genera dos hijos a partir de dos padres mediante un cruce adaptativo (Algoritmo 7) y mutación adaptativa (Algoritmo 8). El cruce adaptativo de los individuos I, J y la mutación adaptativa del individuo I son similares entre si. Primero, con una probabilidad p_v, los valores d_I y e_I se reemplazan por un valor aleatorio. Además, en el caso de cruce, el valor d_J se reemplaza por d_I. Luego, se realiza el cruce indicado por d_I o la mutación indicada por e_I. En resumen, si un individuo proviene de un cruce dado o una mutación dada, ese cruce y mutación específicos se conservan para su descendencia con probabilidad p_v, por lo que el valor de p_v debe ser lo suficientemente pequeño para asegurar una evolución controlada (en nuestro caso, usamos $p_v = 0{,}1$). Aunque la probabilidad del cruce y la mutación no está representada explícitamente, puede calcularse como la relación de los individuos para los cuales los valores de cruce y mutación se establecen en 1. A medida que la población evolucione, los individuos con tipos más exitosos de cruce y mutación serán más comunes, por lo que aumentará la probabilidad de seleccionar estos tipos de cruce y mutación. El uso de operadores de cruce y mutación autoadaptativos ayuda a alcanzar los objetivos de mantener la diversidad en la población y mantener la convergencia del algoritmo evolutivo. También elimina la necesidad de establecer una probabilidad de operador *a priori* para cada operador. En otros enfoques (como [92]), las probabilidades de cruce y mutación varían según el valor de adecuación de las soluciones.

3.5. Selección de atributos evolutiva multi-objetivo para clasificación fuzzy

El uso de lógica fuzzy puede no ser suficiente para que el modelo de clasificación sea interpretable. La interpretabilidad no solo implica *transparencia* (capacidad de expresión) sino también *compacidad*. En *sistemas de clasificación basados en reglas fuzzy*, mejorar la compacidad implica reducir el número de atributos, el número de etiquetas lingüísticas para cada variable y el número de reglas. Los sistemas de clasificación basados en reglas fuzzy están diseñados para producir modelos interpretables; sin embargo, en presencia de un gran número de atributos, los clasificadores resultantes pueden ser demasiado complejos para ser interpretados fácilmente (por ejemplo, las reglas con más de cinco atributos pueden ser intratables para un ser humano). En este sentido, un proceso de selección de atributos [30], antes de la fase de extracción del conjunto de reglas fuzzy, puede ser un paso crucial. Aunque el por problema de selección de atributos es *NP-complejo* [35], con un espacio de búsqueda de $O(2^N)$, donde N es el número de atributos, una estrategia de búsqueda heurística o meta-heurística puede obtener buenas soluciones aproximadas en tiempos razonables, reduciendo así la complejidad para construir el clasificador final.

Como ya hemos comentado, los tres esquemas de selección de atributos más comunes son los llamados *filter*, *wrapper* y *embedded*. Los métodos de selección filter aplican medidas estadísticas para evaluar el subconjunto de atributos, mientras que los métodos wrapper interactúan con un clasificador para evaluar el subconjunto de atributos usando alguna métrica de rendimiento. Los métodos filter son computacionalmente más rápidos, pero menos precisos, que los wrappers. Además, una desventaja de los métodos wrapper es que el rendimiento de los subconjuntos seleccionados a menudo dependen mucho del algoritmo de aprendizaje que se utiliza para la evaluación de los subconjuntos, de modo que, por ejemplo, una buena selección de atributos realizada con un wrapper basado en árboles de decisión puede resultar deficiente cuando los atributos seleccionados se usan en una máquina de soporte vectorial. En [93] se propone un método filter de selección de atributos utilizando un algoritmo voraz como estrategia de búsqueda y se emplea una medida de dependencia entre los atributos de decisión y sus definiciones fuzzy para evaluar la importancia de un atributo candidato. En [94] se propone un método wrapper de selección de atributos donde se utiliza *best-first* como estrategia de búsqueda, y el método de Wang y Mendel para generar la base de reglas fuzzy como evaluador. Finalmente, los métodos embedded [84] logran el ajuste del modelo y la selección de atributos simultáneamente.

Sin embargo, aunque el método propuesto en [27] (y usado como una de las bases de esta tesis) detecta los atributos 'don't care conditions', esto puede no ser suficiente en presencia de muchos atributos en la base de datos, y se recomienda un método de selección de atributos previo a la fase de extracción de la reglas fuzzy. Por lo tanto, si la selección de atributos se utiliza para una clasificación posterior basada en reglas fuzzy, la mejor opción sería un método wrapper de selección de atributos que utilice un clasificador basado en reglas fuzzy como evaluador. Sin embargo, esta configuración para un método wrapper no

está exenta de inconvenientes. Debido al alto costo computacional requerido por los sistemas evolutivos basados en reglas fuzzy en la presencia de un gran número de atributos, un método wrapper que utiliza un clasificador evolutivo basado en reglas fuzzy puede no ser viable. Por lo tanto, es necesario analizar cuidadosamente los parámetros tanto de la estrategia evolutiva de búsqueda como del estrategia evolutiva para obtener un buen equilibrio entre la precisión y el tiempo de ejecución.

En esta sección, proponemos un nuevo método wrapper de selección de atributos multivariate para *sistemas de clasificación basados en reglas fuzzy* que presenta las siguientes novedades y beneficios con respecto a los métodos existentes en la literatura:

1. Se propone un método wrapper de selección de atributos previo a la fase de extracción de reglas, donde la estrategia de búsqueda y el evaluador se realizan con algoritmos evolutivos multi-objetivo independientes. En la literatura actual, por lo general, la selección de atributos está incorporada en el propio algoritmo de extracción de reglas fuzzy.

2. El método de extracción de reglas fuzzy está basado en optimización multi-objetivo con restricciones de parámetros reales, en lugar de en optimización combinatoria multi-objetivo, al igual que el resto de los métodos en la literatura. En el resto de los métodos, la base de reglas está separada explícitamente de la base de conocimiento (previamente creada) que contiene la definición de los conjuntos fuzzy. Luego, los conjuntos fuzzy se combinan en la base de reglas utilizando técnicas de optimización combinatoria. Nuestro enfoque no construye una base de conocimiento explícita con las definiciones de los conjuntos fuzzy, sino que los conjuntos fuzzy se incorporan directamente en la base de reglas de forma aleatoria dentro de los dominios de cada variable. Dado que las definiciones de los conjuntos fuzzy son aleatorias, esto puede producir particiones entremezcladas y, por lo tanto, no interpretables. Para evitar esto, se impone una restricción de similaridad para los conjuntos fuzzy en el modelo de optimización, que el algoritmo evolutivo multi-objetivo maneja mediante una técnica de reparo la cual se aplica después de la inicialización, el cruce y la mutación. Los conjuntos fuzzy gaussianos están representados por su centro y varianza como parámetros reales, y por lo tanto los operadores de cruce y mutación utilizados por el algoritmo evolutivo son los de las representaciones de punto flotante, variando los centros y las varianzas de los conjuntos fuzzy gaussianos por separado.

3. La interpretabilidad de la base de reglas se puede ajustar, no sólo imponiendo un máximo de reglas y un máximo de etiquetas lingüísticas, sino también un umbral máximo de similaridad (definido por el usuario) entre los conjuntos fuzzy. Así, cuando el umbral de similaridad es cercano a 0, los conjuntos fuzzy están suficientemente separados, dando lugar a modelos fuzzy descriptivos, mientras que cuando el umbral de similaridad es cercano a 1, los conjuntos fuzzy pueden ser muy similares, produciendo modelos fuzzy aproximativos (no interpretables).

4. El método permite tratar con bases de datos compuestas de atributos tanto numéricos como categóricos (o nominales). Esto es importante porque en muchos problemas de la vida real están presentes ambos tipos de atributos. El algoritmo evolutivo multi-objetivo trata ambos tipos de atributos por separado en la representación de individuos, así como en los operadores de cruce y mutación. Los cromosomas se dividen en dos partes (una para los atributos numéricos y otra para los categóricos). Los atributos numéricos se tratan como conjuntos fuzzy y se optimizan mediante la optimización con restricciones de parámetros reales, y los atributos categóricos se representan con números enteros y se tratan mediante optimización combinatoria. El motor de inferencia fusiona ambos tipos de atributos para proporcionar las predicciones del clasificador.

5. Tanto el algoritmo evolutivo multi-objetivo para la estrategia de búsqueda como para la extracción de reglas fuzzy se han diseñado con operadores de variación adaptativos. De esta manera, no es necesario realizar experimentos preliminares para ajustar las probabilidades de cruce y mutación.

6. La estrategia de búsqueda y el clasificador fuzzy se han incluido en la plataforma *Weka* [13] como paquetes oficiales. Por lo tanto, el método propuesto puede compararse con el resto de métodos de selección de atributos y clasificadores implementados en la plataforma, así como someterse a tests estadísticas en un amplio conjunto de medidas de rendimiento.

Con respecto a la metodología utilizada para la validación del método propuesto, se han aplicado las siguientes técnicas:

1. Debido al alto costo computacional, los experimentos se han orientado a establecer una configuración aceptable con respecto al número de evaluaciones requeridas por la estrategia de búsqueda y por el clasificador basado en reglas fuzzy. Con este fin, se han llevado a cabo tests estadísticos sobre el tiempo de ejecución, la precisión y la cantidad de atributos seleccionados.

2. Comparamos el método propuesto en su mejor configuración con una amplia gama de métodos de selección de atributos (79 en total) que incluyen métodos filter, wrapper, univariate, y multivariate, con estrategias de búsqueda deterministas y probabilísticas, y con evaluadores de diversa naturaleza.

3. Para comparar el rendimiento de los 79 métodos de selección de atributos, se han utilizado la precisión, el área ponderada bajo la curva *ROC*, el error cuadrático medio y el tamaño del modelo. Para seleccionar los mejores métodos, proponemos un proceso de toma de decisiones multi-objetivo para identificar las soluciones no dominadas de un problema de optimización combinatoria multi-objetivo con 4 objetivos (uno para cada métrica de rendimiento).

4. Finalmente, el modelo de clasificación basado en reglas fuzzy obtenido con el método propuesto se ha evaluado con métricas de rendimiento estándar y se ha comparado con otros clasificadores basados en reglas fuzzy conocidos.

Para la realización de los experimentos utilizamos una base de datos de la vida real. Los datos se han extraído de un *contact center* de tamaño mediano que gestiona comunicaciones tanto entrantes como salientes, con diferentes propósitos, que incluyen atención al cliente y seguimiento, así como marketing y control de calidad. El objetivo de este problema de clasificación es evaluar el desempeño de los agentes. Los datos operativos incluyen toda la información técnica necesaria para reconstruir un historial detallado de los eventos que tienen lugar durante cada comunicación, e incluyen, por ejemplo, el número de teléfono entrante o saliente, los agentes que han estado involucrados, llamadas transferidas, y duración de las llamadas. Por otro lado, los datos del servicio son específicos del servicio en particular para el que se realizó el contacto y pueden incluir, por ejemplo, todas las respuestas proporcionadas por el sujeto entrevistado durante una encuesta de salida. Por lo tanto, podemos definir este problema como un problema de selección de atributos, cuyo propósito es establecer qué subconjunto de variables indica mejor, objetivamente, el rendimiento de un agente. Para este fin, el centro ha recopilado los datos acumulados, representados por el agente, de un período de tiempo significativo y una gama significativa de servicios diferentes, y pidió a tres supervisores independientes que evalúen a cada agente involucrado. Tal evaluación desempeña el papel de la opinión de los expertos de este problema, y el modelo que estamos buscando trata de predecir tal juicio.

Selección de atributos para clasificación fuzzy

Uno de los inconvenientes de la clasificación fuzzy es que genera modelos poco interpretables en presencia de muchos atributos. En estos casos, se requiere un proceso de selección de atributos, previo a la extracción de reglas fuzzy. En esta sección se propone el siguiente método de selección de atributos para su posterior uso en clasificación fuzzy. Se propone el algoritmo evolutivo multi-objetivo *ENORA* como *estrategia de búsqueda*. Como se muestra en [28, 95], el rendimiento de *ENORA* es generalmente mejor que el de *NSGA-II* en términos de *hipervolumen* [96], y mejor que otras estrategias de búsqueda de un solo objetivo. El *evaluador* consiste en un clasificador basado en reglas fuzzy mediante, otra vez, *ENORA*, que supera a *NSGA-II* en esta tarea también [27], conducido por la accuracy (ACC) como *medida de rendimiento*. La búsqueda guiada por ACC ha dado mejores resultados en los experimentos preliminares que la búsqueda usando *área bajo la curva ROC (AUC)*, que obtiene peores valores que ACC. Como *criterio de parada* se usa un límite en el número de generaciones.La figura 4 muestra gráficamente el método se selección de atributos propuesto. Como se puede ver en la figura 4, el método propuesto consiste básicamente en un algoritmo evolutivo multi-objetivo (estrategia de búsqueda *MultiObjectiveEvolutionarySearch*) donde para la evaluación de un subconjunto de atributos candidatos, se usa un wrapper basado en el clasificador *MultiObjectiveEvolutionaryFuzzyClassifier*. Por lo tanto, para evaluar un subconjunto de atributos candidatos, se ejecuta un algoritmo evolutivo multi-objetivo para extraer un conjunto de reglas fuzzy en la base de datos reducida, que se evalúa con una validación cruzada usando la métrica ACC. Los siguientes pasos se realizan para evaluar un subconjunto de atributos $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$, $x_k \in \{true, false\}$, $k = 1, \ldots, N$, de la base de datos $\mathcal{D}$:

1. Eliminar de la base de datos $\mathcal{D}$ aquellos atributos x_k tal que $x_k = false$, obteniendo una base de datos reducida $\mathcal{D}$';

2. Ejecutar *MultiObjectiveEvolutionaryFuzzyClassifier* sobre la base de datos $\mathcal{D}$' para extraer un conjunto de reglas fuzzy.

3. Evaluar el conjunto de reglas fuzzy con validación cruzada utilizando la métrica de rendimiento *ACC*.

4. Devolver el 'mérito' del subconjunto de atributos $\mathbf{x}$.

Teniendo en cuenta la amplitud del espacio de búsqueda ($O(2^N)$) para la selección de atributos, y la complejidad intrínseca de la extracción de reglas fuzzy (para cada intento), es crucial ajustar los parámetros de evolución de ambos algoritmos evolutivos multi-objetivo para obtener una buena compensación entre la precisión y el tiempo de ejecución.

3.6. Un algoritmo evolutivo multi-objetivo basado en hipervolumen para optimización many-objective

Una tendencia actual en computación evolutiva multi-objetivo es el diseño de algoritmos evolutivos que permitan manejar eficientemente muchos objetivos (más de 3). Esto es debido a que los algoritmos evolutivos de optimización multi-objetivo existentes basados en ordenación no dominada comienzan a tener problemas de convergencia a partir de 3 objetivos, además de requerir tiempos de ejecución muy elevados. Este tipo de optimización evolutiva multi-objetivo con muchos objetivos ha recibido recientemente el nombre de *optimización evolutiva many-objective* [97].

Una de las formas de tratar problemas de optimización many-objective con algoritmos evolutivos es mediante el uso de la métrica de evaluación del *hipervolumen*. En el contexto de esta tesis se ha desarrollado un algoritmo evolutivo novel de optimización many-objective, el cual, aunque está basado también en la métrica del hipervolumen como otros métodos publicados en la literatura [98, 99], difiere mucho de éstos por la forma en la que se usa el hipervolumen en la evolución de loas poblaciones.

Nuestro método, que denominamos *HVMOEA*, está inspirado en el algoritmo evolutivo multi-objetivo *ENORA*, e implementa también sustitución generacional *$(\mu + \lambda)$-ES*, con $\mu = \lambda = popsize$, y selección por torneo binario. *HVMOEA*, al igual que *ENORA*, divide el espacio de los objetivos en slots y calcula el slot al cual pertenece cada individuo. La diferencia con respecto a *ENORA* es la siguiente: en lugar de hacer una ordenación no dominada de los individuos en su slot como hace *ENORA*, la ordenación se hace de acuerdo al hipervolumen del individuo en su slot. Si dos individuos tienen igual ranking en su slot, el mejor es de nuevo el que mejor hipervolumen tenga, sin tener que calcular el valor de *crowding* como hace *ENORA*. Para el cálculo del hipervolumen se usan métodos probabilistas del tipo *MonteCarlo*.

La ventajas de *HVMOEA* son claras: dado que el hipervolumen es una métrica que mide tanto optimalidad como diversidad, el algoritmo suprime, por un lado, la ordenación no dominada por frentes, y por otro, el cálculo de las distancias de crowding para los individuos con igual ranking. Esto hace que el tiempo de ejecución requerido por *HVMOEA* sea muy inferior al tiempo requerido por los algoritmos convencionales para un mismo número de evaluaciones de las funciones objetivo.

Con el fin de mejorar el hipervolumen de la población hemos planteado modificar la forma en que el algoritmo divide la población en slots. Tanto *ENORA* como *HVMOEA* distribuyen la población en slots, y en ambos casos dividen el espacio de soluciones usando como eje el punto en el que se encuentra el mejor valor para cada uno de los objetivos. El número de formas de calcular los slots dependerá del número de objetivos que tengamos, pudiendo distribuir el espacio de los objetivos usando como eje cualquier punto límite o usando como división los valores de alguno de los objetivos.

El escenario del estudio tiene dos objetivos $f1$ y $f2$, donde $f1$ se trata la accuracy, que se maximiza con valores entre 0 y 1, y $f2$ es el número de atributos, que se minimiza con valores entre 0 y el número de atributos inicial de la base de datos, n. Para este caso, *ENORA* y *HVMOEA* usan como eje para dividir el espacio en slots el punto $(1, 0)$. La tabla 3 muestra la descripción de todas las formas de calcular los slots que se han estudiado.

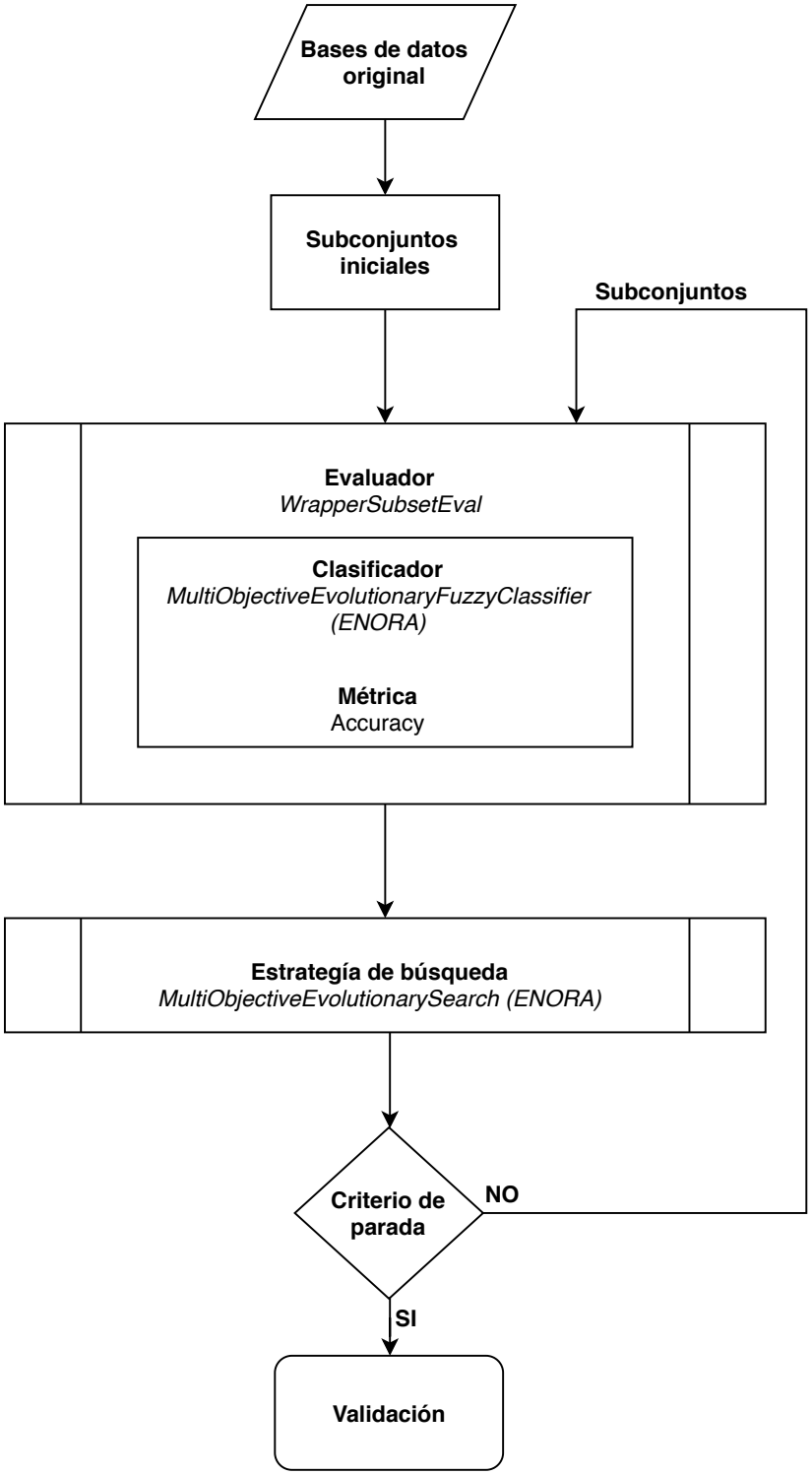


Figura 4: El método wrapper de selección de atributos multivariable propuesto.

Método	Descripción
MAX	Slots transversales desde el punto $(1, n)$
MIN	Slots transversales desde el punto $(0, 0)$
EVEN	Slots transversales desde el punto $(1, 0)$
ODD	Slots transversales desde el punto $(0, n)$
GRID	Slots verticales desde el eje del número de atributos

Tabla 3: Descripción de las distintas formas de calcular los slots.

4. Resultados

En esta sección se analizan los resultados obtenidos en función de las pruebas realizadas en cada uno de los tres escenarios de la tesis. También se han incluido resultados parciales de la investigación realizada no publicada referente a un algoritmo evolutivo multi-objetivo basado en hipervolumen para optimización con restricciones con muchos objetivos y resultados finales de un estudio sobre la influencia de las formas de distribuir la población en slots sobre el hipervolumen.

4.1. En el contexto del screening virtual

En términos del indicador de hipervolumen, las comparaciones nos permiten afirmar que:

1. Comparando los resultados de *ENORA* y *NSGA-II* para selección de atributos evaluada con *J48*, podemos afirmar que *ENORA* proporciona mejores valores que *NSGA-II*. Los intervalos de confianza al 95 % de la media basados en la distribución t se han realizado con muestras de 30 individuos, lo que nos lleva a concluir que las diferencias entre los valores de hipervolúmen obtenidos con ambos algoritmos son estadísticamente significativos. Los valores obtenidos por *ENORA* son (significativamente) mejores que los obtenidos por *NSGA-II* en los conjuntos de datos *tk* y *mr* bajo el modelo de optimización basado en *ACC*, y también en el conjunto de datos *tk* bajo el modelo de optimización basado en *AUC*.

2. La razón principal por la que *ENORA* y *NSGA-II* se comportan de manera diferente es la siguiente. En el torneo binario, *NSGA-II* nunca selecciona un individuo dominado por el otro. En *ENORA*, un individuo dominado por el otro puede ser el ganador del torneo. La figura 1 muestra este comportamiento gráficamente. Por ejemplo, si se seleccionan los individuos B y C para un torneo binario con *NSGA-II*, el individuo B gana al individuo C porque B domina a C. Por el contrario, el individuo C gana al individuo B con *ENORA* porque C tiene una mejor posición en su slot que B. Por tanto, *ENORA* fomenta la diversidad ya que permite que los individuos en cada slot evolucionen hacia el frente de Pareto, aunque estos individuos no sena los mejores cuando se comparan, obteniendo así un hipervolumen mejor que *NSGA-II* a lo largo del proceso de evolución.

Con respecto a la evaluación en full training de los modelos de clasificación, observamos que:

1. Usando el clasificador *C4.5*, todos los métodos de selección de atributos (excepto *GA-C4.5-ACC* y *GA-C4.5-AUC*) mejoran el rendimiento con respecto a las bases de datos originales, para todas las métricas evaluadas, reduciendo además el número de características.

2. En general, los métodos multi-objetivo de selección de atributos obtienen bases de datos reducidas con menor número de atributos que los métodos de un solo objetivo. La razón principal de esto es que las estrategias de búsqueda de *ENORA* y *NSGA-II* minimizan explícitamente el número de atributos.

3. Los mejores valores de *ACC* y *AUC* se obtienen, en general, con las estrategias de búsqueda *ENORA* y *PSO*. No obstante, *ENORA* selecciona menos atributos que *PSO*.

4. Los modelos de optimización basados en *AUC* producen mejores clasificadores que los producidos por los modelos de optimización basados en *ACC* cuando se evalúan en modo full training.

5. El mejor resultado reportado por otros autores para los conjuntos de datos *tk* y *mr* evaluados en full training obtiene un área bajo la curva *ROC* de 0.95 y 0.99 respectivamente, con los métodos *C_RF_RF* (Random Forest) y *C_RF_SVM* (Support Vector Machine), frente a 1.00 (clasificación

perfecta) obtenida en esta tesis con *C4.5* para ambos conjuntos de datos. Es importante indicar que los métodos *C_RF_RF* y *C_RF_SVM* no solo obtienen resultados peores que los obtenidos en este trabajo, sino también que los clasificadores generados no son interpretables.

Con respecto a la evaluación de los modelos de clasificación en validación cruzada de 10 repeticiones con 30 iteraciones y test estadísticos, observamos que:

1. La evaluación con *C4.5* ha resultado mejor que la evaluación con otros clasificadores, como era de esperar, dado que los métodos de selección de atributos están configurados con *C4.5* como evaluador.
2. El tamaño del modelo de los clasificadores producidos por las estrategias de búsqueda *ENORA* y *NSGA-II* son, en general, más pequeños que los producidos por las estrategias de búsqueda *GA* y *PSO*.
3. Los modelos de optimización basados en *ACC* producen mejores clasificadores que los producidos por los modelos de optimización basados en *AUC* cuando se evalúan con validación cruzada. Ya que los modelos de optimización basados en *AUC* fueron mejores que los modelos de optimización basados en *ACC* al evaluarse en full training, podemos afirmar que los modelos de optimización basados en *ACC* son menos susceptibles de overfitting que los modelos de optimización basados en *AUC*. Para estos casos (métodos de selección de atributos basados en *ACC*), la mejor *ACC* y *AUC* se obtuvo con la estrategia de búsqueda *ENORA*.
4. No hay diferencias estadísticamente significativas entre la estrategia de búsqueda *ENORA* y las otras estrategias de búsqueda con respecto a las medidas consideradas con el clasificador *C4.5*, excepto en *Serialized_Model_Size*, donde *ENORA* es mejor en la mayoría de los casos para el conjunto de datos *tk*. Solo *NSGA-II* es mejor que *ENORA* para el conjunto de datos *mr* en *Serialized_Model_Size* cuando se evalúa con *ACC*.
5. El mejor resultado reportado por otros autores para los conjuntos de datos *tk* y *mr* evaluados en validación cruzada obtiene un área bajo la curva *ROC* de 0.95 y 0.98 respectivamente con el método *SVM_AE246* (Support Vector Machine), contra los 0.93 y 0.95 obtenidos en este trabajo con *C4.5*. Si bien los resultados con *C4.5* obtenidos en este trabajo son ligeramente peores en este caso, se debe tener en cuenta que: 1) los árboles de decisión obtenidos con *C4.5* son interpretables, mientras que las *SVM* no lo son, y 2) el método *SVM_AE246* se evaluó en validación cruzada de 5 repeticiones con una sola iteración (sólo 5 modelos), mientras que el árbol de decisión obtenido en este trabajo se evaluó en validación cruzada de 10 repeticiones con 30 iteraciones (300 modelos), por lo que el grado de posible overfitting es considerablemente menor.

Trabajo/Año	**Métrica**	***tk***	***mr***
Wang et al. (2004) *BINDSURF*	–	0.700	0.695
Shoichet et al. (1992) *DOCK*	–	0.521	0.554
Abagyan et al. (1994) *ICM*	–	0.723	0.789
Friesner et al. (2004) *GLIDE*	–	0.681	0.856
Pérez-Sánchez et al. (2014b) *NNET_EE246*	5-fold CV, 1 it.	0.94	0.87
Pérez-Sánchez et al. (2014b) *SVM_AE246*	5-fold CV, 1 it.	0.95	0.98
Roy and Skolnick (2014) *LIGSIFT*	full training set	0.92	0.89
Pereira et al. (2016) *DeepVS-Dock*	leave-one-out CV	0.44	0.55
Pereira et al. (2016) *DeepVS-ADV*	leave-one-out CV	0.54	0.82
Cano et al. (2017) *C_RF_SVM*	full training set	0.94	0.99
Cano et al. (2017) *C_RF_NNET*	full training set	0.94	0.99
Cano et al. (2017) *C_RF_RF*	full training set	0.95	0.98
Esta tesis (2019) *ENORA-C4.5-AUC*	full training set	1.00	1.00
Esta tesis (2019) *NSGA-II-C4.5-AUC*	10-fold CV, 30 it.	0.93	0.95

Tabla 4: Resultados reportados en la literatura para bases de datos *tk* y *tk*.

En general, podemos afirmar que la estrategia de búsqueda multi-objetivo *ENORA* ha obtenido clasificadores buenos e interpretables, con una *ACC* entre 0,9934 y 1,00 y un *AUC* entre 0,96 y 1,00

evaluados en full training, y una *ACC* entre 0,9849 y 0,9940 y un *AUC* entre 0,89 y 0,93 evaluados con 10-fold cross-validation sobre 30 iteraciones, al tiempo que se reduce sustancialmente el tamaño del modelo.

4.2. En el contexto de la clasificación basada en reglas con datos categóricos

Se han realizado diferentes experimentos con distintos modelos de optimización para evaluar el rendimiento general de la técnica propuesta y comparar el efecto de optimizar diferentes objetivos para el mismo problema. Concretamente se han utilizado las métricas *ACC*, *AUC* y *RMSE* como funciones objetivo, y los optimizadores *ENORA* y *NSGA-II*, dando lugar a seis métodos diferentes: *ENORA-ACC*, *ENORA-AUC*, *ENORA-RMSE*, *NSGA-II-ACC*, *NSGA-II-AUC* y *NSGA-II-RMSE*.

De los resultados de estos experimentos, realizados con las bases de datos públicas *Breast Cancer* y *Monk's Problem 2*, podemos deducir que, en primer lugar, *ENORA* mantiene una mayor diversidad de la población y obtiene una mejor relación de hipervolumen con respecto a *NSGA-II*, y en segundo lugar, el uso de *ACC* como primer objetivo genera mejores valores de hipervolumen que el uso de *AUC*, que, a su vez, tiene un mejor desempeño que el uso de *RMSE*. *NSGA-II* identifica menos soluciones que *ENORA* en el frente Pareto, lo que implica menos diversidad y, por lo tanto, una relación de hipervolumen peor, como se muestra en las figuras 5 y 6.

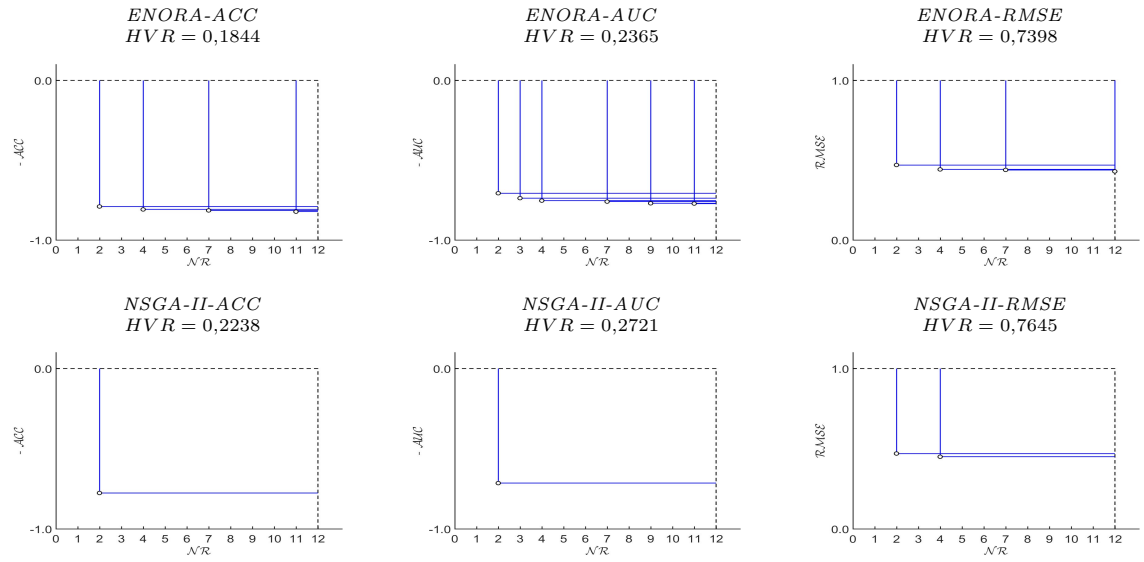


Figura 5: Frentes Pareto de una ejecución de *ENORA* y *NSGA-II*, con $M_{max} = 12$, en el conjunto de datos *Breast Cancer*, y sus respectivos valores de hipervolumen. Nótese que en el caso de la clasificación multi-objetivo donde se maximiza $\mathcal{F}_{\mathcal{D}}$ ($\mathcal{ACC}_{\mathcal{D}}$ y $\mathcal{AUC}_{\mathcal{D}}$), la función $\mathcal{F}_{\mathcal{D}}$ se ha convertido a minimización para una mejor visualización del frente Pareto.

En el siguiente bloque de experimentos, los seis métodos anteriores se han comparado otros con sistemas basados en reglas, concretamente con *PART*, *JRip*, *OneR* y *ZeroR*, usando las base de datos públicas *Breast Cancer*, *Monk's Problem 2*, *Tic-Tac-Toe-Endgame*, *Car*, *kr-vs-kp* y *Nursery*, con los modos de evaluación *full training*, *percentage split* y *validación cruzada de 10 repeticiones* (tablas 5, 6 y 7).

Comparando los resultados en modo full training con los resultados en percentage split o validación cruzada, se hace evidente que nuestra propuesta produce modelos de clasificación que son más resistentes al overfitting. Por ejemplo, el clasificador construido por *PART* con *Monk's Problem 2* presenta un 94.01 % de *ACC* en full training, y cae al 73.51 % en el modo percentage split. Una diferencia similar ocurre con la base de datos *Breast Cancer*; por otro lado, el clasificador contruido con *ENORA-ACC* muestra solo una caída de 5.57 % en un caso, e incluso una mejora en el otro caso. Este fenómeno se explica fácilmente al observar el número de reglas: a más reglas en un clasificador, mayor es el riesgo de un overfitting; *PART* produce clasificadores muy precisos, pero a costa de agregar muchas reglas, lo cual no solo afecta la interpretabilidad del modelo, sino también su resistencia al overfitting. Los resultados con full training parecen indicar que cuando el modelo de optimización es guiado por *RMSE* los clasificadores son más

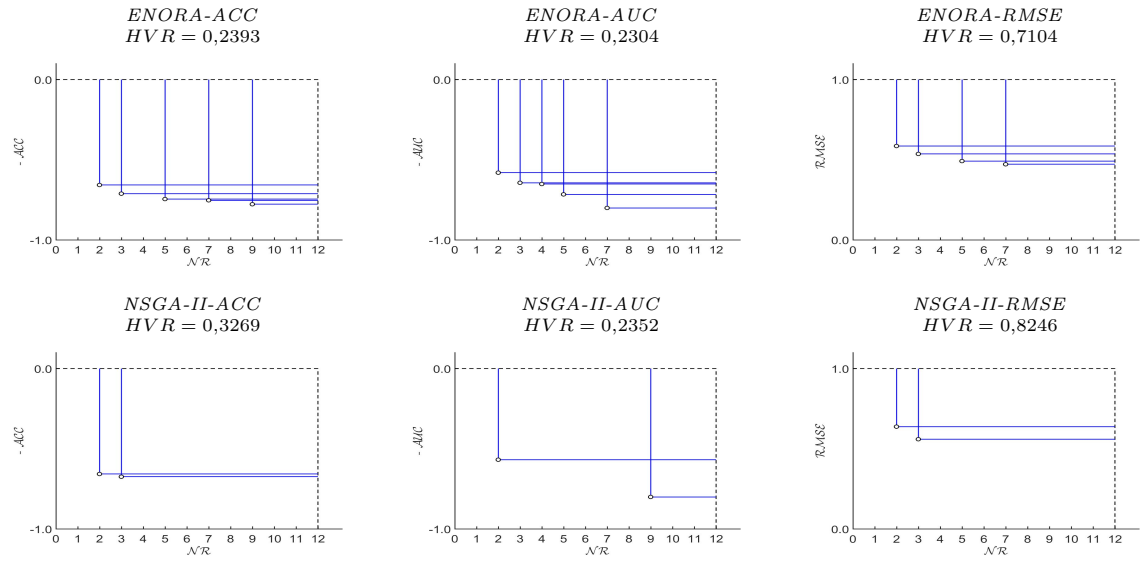


Figura 6: Frentes Pareto de una ejecución de *ENORA* y *NSGA-II*, con $M_{max} = 12$, en el conjunto de datos *Monk's Problem 2*, y sus respectivos valores de hipervolumen. Nótese que en el caso de la clasificación multi-objetivo donde se maximiza $\mathcal{F}_{\mathcal{D}}$ ($\mathcal{ACC}_{\mathcal{D}}$ y $\mathcal{AUC}_{\mathcal{D}}$), la función $\mathcal{F}_{\mathcal{D}}$ se ha convertido a minimización para una mejor visualización del frente Pareto.

Modelo de aprendizaje	**Percent correct**	**Área *ROC***	**Serialized model size**
ENORA-ACC	73.45	0.61	9554.80
ENORA-AUC	70.16	0.62	9554.63
ENORA-RMSE	72.39	0.60	9557.77
NSGA-II-ACC	72.50	0.60	9556.20
NSGA-II-AUC	70.03	0.61	9555.70
NSGA-II-RMSE	73.34	0.60	9558.60
PART	68.92	0.61	55298.13
JRip	71.82	0.61	7664.07
OneR	67.15	0.55	1524.00
ZeroR	70.30	0.50	915.00

Tabla 5: Comparación de los clasificadores en modo validación cruzada de 10 repeticiones con 3 iteraciones, base de datos *Breast Cancer*.

precisos, pero sin embargo, también son más propensos al overfitting, lo que indica que, en promedio, son preferibles los modelos de optimización guiados por *ACC*.

De los tests estadísticos deducimos que no hay diferencias estadísticamente significativas entre las seis variantes del modelo de optimización propuesto, lo que sugiere que las ventajas del método propuesto no dependen directamente del algoritmo evolutivo específico ni de la métrica de evaluación específica que se utiliza para guiar las evoluciones. Como es de esperar, hay diferencias estadísticamente significativas entre los métodos propuestos y el método clásico simple *OneR*. No se han encontrado diferencias estadísticamente significativas entre los métodos propuestos y uno bien consolidado, como es *PART*, pero el precio que paga *PART* para tener resultados similares a los métodos propuestos es un número muy alto de reglas (15 contra 2 en el caso de *Breast Cancer*, 47 contra 7 en el *M*onk's Problem 2).

Hay que resaltar que tanto el conjunto de datos *Breast Cancer* como el conjunto de datos *Monk's problem 2* son difíciles de aproximar con los clasificadores interpretables y que ninguno de los clasificadores analizados obtiene altos índices de precisión utilizando la técnica de validación cruzada. Incluso los potentes clasificadores de caja negra, como *Random Forest* y *Logistic*, obtienen tasas de éxito inferiores a 70 % en una validación cruzada de 10 folds para estos conjuntos de datos. Sin embargo, *ENORA* obtiene un mejor equilibrio entre precisión e interpretabilidad que el resto de los clasificadores. Para el resto de los conjuntos de datos analizados, la precisión obtenida con *ENORA* es sustancialmente mayor. Por ejemplo,

Modelo de aprendizaje	**Percent correct**	**Área *ROC***	**Serialized model size**
ENORA-ACC	76.69	0.70	9586.50
ENORA-AUC	72.82	0.79	9589.30
ENORA-RMSE	75.66	0.68	9585.30
NSGA-II-ACC	70.07	0.59	9590.60
NSGA-II-AUC	67.08	0.70	9619.70
NSGA-II-RMSE	67.63	0.54	9565.90
PART	73.51	0.79	73115.90
JRip	64.05	0.50	5956.90
OneR	65.72	0.50	1313.00
ZeroR	65.72	0.50	888.00

Tabla 6: Comparación de los clasificadores en modo percentage split, para *Monk's problem 2*.

para el conjunto de datos *Tic-Tac-Toe-Endgame*, *ENORA* obtiene un porcentaje de éxito de 98,3299 % con solo 2 reglas en la validación cruzada, mientras que *PART* obtiene 94,2589 % con 49 reglas, y *JRip* obtiene 97,8079 % con 9 reglas. Con respecto a los resultados obtenidos en los conjuntos de datos *Car*, *kr-vs-kp* y *Nursery*, comentar que se puede obtener un mejor porcentaje de éxito si se aumenta el número máximo de evaluaciones, que llevaría a incrementar el número de reglas. Sin embargo, aunque con mayor número de reglas se obtienen mejores porcentajes de éxito, esto va en detrimento de la interpretabilidad de los modelos.

Modelo de aprendizaje	**Número de reglas**	**Percent correct**	***TP* Rate**	***FP* Rate**	**Precisión**	**Recall**	***F*-Measure**	***MCC***	Área ***ROC***	**Área *PRC***	***RMSE***
Monk's problem 2											
ENORA-ACC	7	77.70	0.777	0.360	0.777	0.777	0.762	0.481	0.708	0.695	0.472
PART	47	79.53	0.795	0.253	0.795	0.795	0.795	0.544	0.884	0.893	0.380
JRip	1	62.90	0.629	0.646	0.526	0.629	0.535	-0.034	0.478	0.537	0.482
OneR	1	65.72	0.657	0.657	0.432	0.657	0.521	0.000	0.500	0.549	0.586
ZeroR	-	65.72	0.657	0.657	0.432	0.657	0.521	0.000	0.491	0.545	0.457
Tic-Tac-Toe-Endgame											
ENORA-ACC/RMSE	2	98.33	0.983	0.031	0.984	0.983	0.983	0.963	0.976	0.973	0.129
PART	49	94.26	0.943	0.076	0.942	0.943	0.942	0.873	0.974	0.969	0.220
JRip	9	97.81	0.978	0.031	0.978	0.978	0.978	0.951	0.977	0.977	0.138
OneR	1	69.94	0.699	0.357	0.701	0.699	0.700	0.340	0.671	0.651	0.548
ZeroR	-	65.35	0.653	0.653	0.427	0.653	0.516	0.000	0.496	0.545	0.476
Car											
ENORA-RMSE	14	86.57	0.866	0.089	0.866	0.866	0.846	0.766	0.889	0.805	0.259
PART	68	95.78	0.958	0.016	0.959	0.958	0.958	0.929	0.990	0.979	0.1276
JRip	49	86.46	0.865	0.064	0.881	0.865	0.870	0.761	0.947	0.899	0.224
OneR	1	70.02	0.700	0.700	0.490	0.700	0.577	0.000	0.500	0.543	0.387
ZeroR	-	70.02	0.700	0.700	0.490	0.700	0.577	0.000	0.497	0.542	0.338
kr-vs-kp											
ENORA-RMSE	10	94.87	0.949	0.050	0.950	0.949	0.949	0.898	0.950	0.927	0.227
PART	23	99.06	0.991	0.010	0.991	0.991	0.991	0.981	0.997	0.996	0.088
JRip	16	99.19	0.992	0.008	0.992	0.992	0.992	0.984	0.995	0.993	0.088
OneR	1	66.46	0.665	0.350	0.675	0.665	0.655	0.334	0.657	0.607	0.579
ZeroR	-	52.22	0.522	0.522	0.273	0.522	0.358	0.000	0.499	0.500	0.500
Nursery											
ENORA-ACC	15	88.41	0.884	0.055	0.870	0.884	0.873	0.824	0.915	0.818	0.2153
PART	220	99.21	0.992	0.003	0.992	0.992	0.992	0.989	0.999	0.997	0.053
JRip	131	96.84	0.968	0.012	0.968	0.968	0.968	0.957	0.993	0.974	0.103
OneR	1	70.97	0.710	0.137	0.695	0.710	0.702	0.570	0.786	0.632	0.341
ZeroR	-	33.33	0.333	0.333	0.111	0.333	0.167	0.000	0.500	0.317	0.370

Tabla 7: Comparación del rendimiento de los modelos de aprendizaje en modo de validación cruzada con 10 repeticiones - Bases de datos *Monk's Problem 2*, *Tic-Tac-Toe-Endgame*, *Car*, *kr-vs-kp* y *Nursery*.

4.3. En el contexto del *GAP S.R.L. Contact Center*

En este escenario se han llevado a cabo tres bloques de experimentos usando dos bases de datos *INBOUND_AGENTS* y *ALL_AGENTS* extraidas del *GAP S.R.L. Contact Center*: el primero tiene como objetivo encontrar el número óptimo de generaciones tanto de la estrategia de búsqueda como del evaluador para una compensación adecuada entre el rendimiento y el tiempo de ejecución; en el segundo bloque, el método propuesto se compara con otros métodos de selección de atributos multivariate, univariate, filter y wrapper; en el tercer bloque, se comparan los clasificadores basados en reglas fuzzy obtenidos por el método propuesto con los clasificadores fuzzy obtenidos con otros métodos conocidos. Finalmente, las reglas de los mejores modelos fuzzy obtenidos con nuestra propuesta se interpretan en el contexto del

GAP S.R.L. Contact Center.

4.3.1. Número de generaciones óptimo

Se han establecido tres configuraciones diferentes, mostradas en la tabla 8, las cuales se han ejecutados 10 veces cada una de ellas y se han realizado test estadísticos no paramétricos. La tabla 9 muestra un resumen de los resultados con los promedios del tiempo de ejecución (ms), la accuracy y el número de atributos seleccionados.

# Configuración	Número de generaciones Evolutionary search	Número de generaciones Fuzzy classifier
#1	10	10
#2	10	100
#3	100	10

Tabla 8: Las tres configuraciones de parámetros estudiadas en este trabajo.

Configuración	Tiempo de ejecución	Accuracy	Número de atributos
INBOUND_AGENTS			
#1	158300955.5	0.50607	9.7
#2	1140188340.7	0.57998	6.0
#3	553075217.9	0.56	3.5
ALL_AGENTS			
#1	148756602.7	0.50857	7.6
#2	1200222302.3	0.56128	8.0
#3	688512492.7	0.54831	5.5

Tabla 9: Tiempo medio de ejecución, precisión y número de atributos.

En cuanto al tiempo de ejecución, la configuración #1 fue la mejor, como se esperaba. La configuración #3 se comportó mejor que la configuración #2, aunque las diferencias no son estadísticamente significativas. En cuanto a la accuracy, no se encontraron diferencias estadísticamente significativas entre las configuraciones #2 y #3, y ambas se comportaron mejor que la configuración #1. Finalmente, con respecto al número de atributos seleccionados, no hay diferencias estadísticamente significativas entre las tres configuraciones. Este análisis nos permitió concluir que la peor configuración es la #1 con respecto a la precisión y, ya que no hay diferencias estadísticamente significativas entre las configuraciones #2 y #3, optamos por la configuración #3 de acuerdo con el principio de *minimum description length*. Por lo tanto, la configuración #3 se ha utilizado para la selección de attributos final.

4.3.2. Comparación con otros métodos de selección de atributos

Hemos aplicado sistemáticamente un amplio conjunto de métodos de selección de atributos, del tipo univariate, multivariate, filter y wrapper, todos ellos disponibles en la literatura. Cada método es, en sí mismo, una combinación de una elección específica entre las estrategias de búsqueda, los evaluadores y las métricas de evaluación (en el caso de los métodos wrapper). La figura 7 muestra los métodos de selección de atributos usados en la comparación.

Combinando las distintas opciones mostradas en la figura 7, se obtienen un total de 79 métodos de selección de atributos. Para elegir las mejores bases de datos reducidas, consideramos el siguiente problema de optimización multi-objetivo combinatoria:

$$\begin{aligned} Maximizar \quad & f_1(\mathbf{x}) = \frac{1}{n}\sum_{j=1}^{n} ACC(\mathbf{x}, j) \\ Maximizar \quad & f_2(\mathbf{x}) = \frac{1}{n}\sum_{j=1}^{n} WAUC(\mathbf{x}, j) \\ Minimizar \quad & f_3(\mathbf{x}) = \frac{1}{n}\sum_{j=1}^{n} RMSE(\mathbf{x}, j) \\ Minimizar \quad & f_4(\mathbf{x}) = \frac{1}{n}\sum_{j=1}^{n} MS(\mathbf{x}, j) \end{aligned} \tag{10}$$

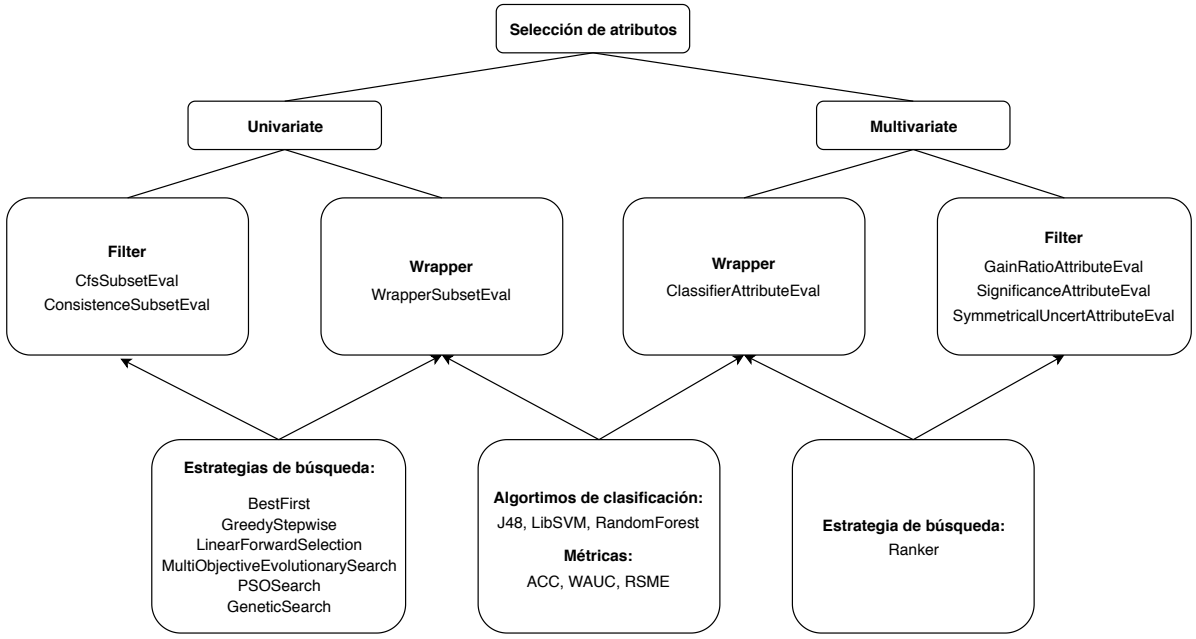


Figura 7: Metodología propuesta para la validación del método propuesto.

donde $\mathbf{x} \in DB$ es una base de datos reducida ($DB = \{1, \ldots, 79\}$), y $n = 3$ es el número de clasificadores (*J48*, *RandomForest* y *LibSVM*). La solución al problema (10) es un conjunto de 4 bases de datos no dominadas para *INBOUND_AGENTS* y un conjunto de 9 bases de datos no dominadas para *ALL_AGENTS*. Estas bases de datos no dominadas han sido comparadas con la base de datos reducida obtenida con el método propuesto y los resultados son claramente favorables a nuestra propuesta, con diferencias estadísticamente significativas en la mitad de los casos, y en en el resto obteniendo siempre mejores resultados.

4.3.3. Comparación con otros clasificadores basados en reglas fuzzy

Se han comparado nuestros resultados de clasificación con los obtenidos por otros clasificadores basados en reglas fuzzy del paquete *R frbs* [100] y de *Weka*, concretamente los algoritmos *FRBCS.CHI*, *FRBCS.W*, *FH.GBML* y *FURIA*. Analizamos la precisión de los clasificadores resultantes tanto en full training como en validación cruzada de 10 repeticiones con 10 iteraciones, así como su estadística kappa, número de reglas, forma del conjunto fuzzy (gaussiano, trapezoidal, triangular), número de etiquetas lingüísticas y tiempo de ejecución en full training.

	MOEFC	*FRBCS.CHI*	*FRBCS.W*	*FH.GBML*	*FURIA*	*MOEFC*	*FRBCS.CHI*	*FRBCS.W*	*FH.GBML*	*FURIA*
	INBOUND_AGENTS					*ALL_AGENTS*				
Precisión del conjunto full training	0.7857	0.6607	0.7143	0.7679	0.5357	0.7403	0.5714	0.5714	0.6364	0.6234
Estadística kappa del conjunto full training	0.666	0.485	0.5762	0.6396	0.2385	0.5877	0.3465	0.3465	0.4111	0.4385
Precisión media 10-fold CV (10 rep.)	0.5940	0.5544	0.5431	0.5723	0.4649	0.6013	0.5459	0.6004	0.5567	0.5714
Estadística kappa media 10-fold CV (10 rep.)	0.3751	0.3018	0.296	0.3243	0.2448	0.3749	0.2728	0.3661	0.2761	0.3265
Máximo número de reglas	14	–	–	14	–	14	–	–	14	–
Número de reglas encontradas	8	21	43	8	3	5	49	46	4	6
Forma del conjunto fuzzy (gaussiano, trapezoidal, triangular)	Trapezoidal	Gaussian	Trapezoidal	Gaussian	Triangular	Trapezoidal				
Número máximo de etiquetas lingüísticas para cada variable.	7	3	21	14	–	7	15	15	4	–
Tiempo de ejecución del conjunto full training	36.12 s.	1.230 s.	0.399 s.	4.11 h.	0.01 s.	40.5 s.	1.552 s.	0.523 s.	4.86 h.	0.01 s.
Número de generaciones	1000	–	–	1000	–	1000	–	–	1000	–
Tamaño de la población	100	–	–	100	–	100	–	–	100	–
Máxima similaridad de los conjuntos fuzzy	0.1	–	–	–	–	0.1	–	–	–	–

Tabla 10: Métricas de rendimiento de los clasificadores basados en reglas fuzzy.

La precisión obtenida en el modo full training por parte de *MultiObjectiveEvolutionaryFuzzyClassifier* (*MOEFC*) es mucho mejor que la obtenida por cualquier otro método; tal diferencia es menor en el

modo de validación cruzada. En términos de interpretabilidad, *MOEFC* encontró (muchas) menos reglas que *FRBCS.CHI* y que *FRBCS.W*, mientras que la cantidad de reglas encontradas por *FH.GBML* es la misma (8 reglas) en el caso de *INBOUND_AGENTS*, y una menos (4 reglas en lugar de 5 reglas) en el caso de *ALL_AGENTS*. Aunque *FURIA* ha encontrado modelos muy compactos (3 y 6 reglas para *INBOUND_AGENTS* y *ALL_AGENTS* respectivamente), la accuracy tanto en full training como en 10-fold cross-validation es mucho peor que la accuracy obtenida con *MOEFC*. Por lo tanto, podemos concluir que *MOEFC* se comporta generalmente mejor que los otros cuatro clasificadores fuzzy. En particular, obsérvese que el único clasificador cuyo rendimiento es comparable con el de *MOEFC* en este problema, que es *FH.GBML*, consumió un tiempo de ejecución muy superior: 36.12 – 40.5 segundos (*MOEFC*) contra 4.11 – 4.86 horas (*FH.GBML*) con el mismo número de generaciones y el mismo tamaño de población.

4.4. En el contexto de un algoritmo evolutivo multi-objetivo basado en hipervolumen para optimización many-objective

Como un primer conjunto de experimentos, el algoritmo *HVMOEA* ha sido testado en las bases de datos *ALL_AGENTS* del *GAP S.R.L. Contact Center* y con la base de datos *hcc survival* de la *UCI Machine Learning Repository*. Se han realizado 30 ejecuciones con los métodos *HVMOEA+J48+ACC*, para *ALL_AGENTS*, y *HVMOEA+RandomForest+ACC*, para *hcc survival*. Cada algoritmo se ha ejecutado con 10000 evaluaciones. Se han calculado los intervalos de confianza al 95 % mediante t-test para las métricas *hipervolumen*, *distancia generacional*, *dispersión* y *tiempo de ejecución* de algoritmo de dos formas, el global y el tiempo sin considerar el consumido en la evaluación de los clasificadores. La razón de este desglose en las métricas del tiempo de ejecución se debe a que, en selección de atributos tipo wrapper, cuanta mayor diversidad de las soluciones hay, mayor es el tiempo de ejecución global ya que hay que evaluar más clasificadores y con más atributos. Por tanto, para aislar el tiempo base del algoritmo *HVMOEA* hemos medido el tiempo del algoritmo suponiendo un tiempo constante para evaluación de las funciones objetivo, y por otro lado el tiempo de ejecución total. Las figuras 8, 9 y 10 muestran los *diagramas de cajas* para las 30 ejecuciones en cada una de las métricas sobre las dos bases de datos testadas.

Las siguientes deducciones pueden extraerse de los diagramas de cajas:

1. El hipervolumen de *HVMOEA* es mejor que el de *NSGA-II*, presentando diferencias estadísticamente significativas.

2. Esta diferencia de hipervolumen entre *HVMOEA* y *NSGA-II* se debe a que la dispersión de las soluciones en *HVMOEA* es mucho mejor (estadísticamente). Sin embargo, la distancia generacional de *NSGA-II* es mejor que la de *HVMOEA*.

3. El tiempo de ejecución de *HVMOEA* sin considerar la evaluación de los clasificadores es, como se esperaba, mucho menor que el tiempo de los algoritmos *ENORA* y *NSGA-II*.

4. El tiempo global de *HVMOEA* es mayor que el tiempo de ambos algoritmos *ENORA* y *NSGA-II*. Esto se debe, como se ha comentado anteriormente, a que la diversidad de los individuos es mayor, teniendo por tanto que evaluarse más clasificadores y con más atributos.

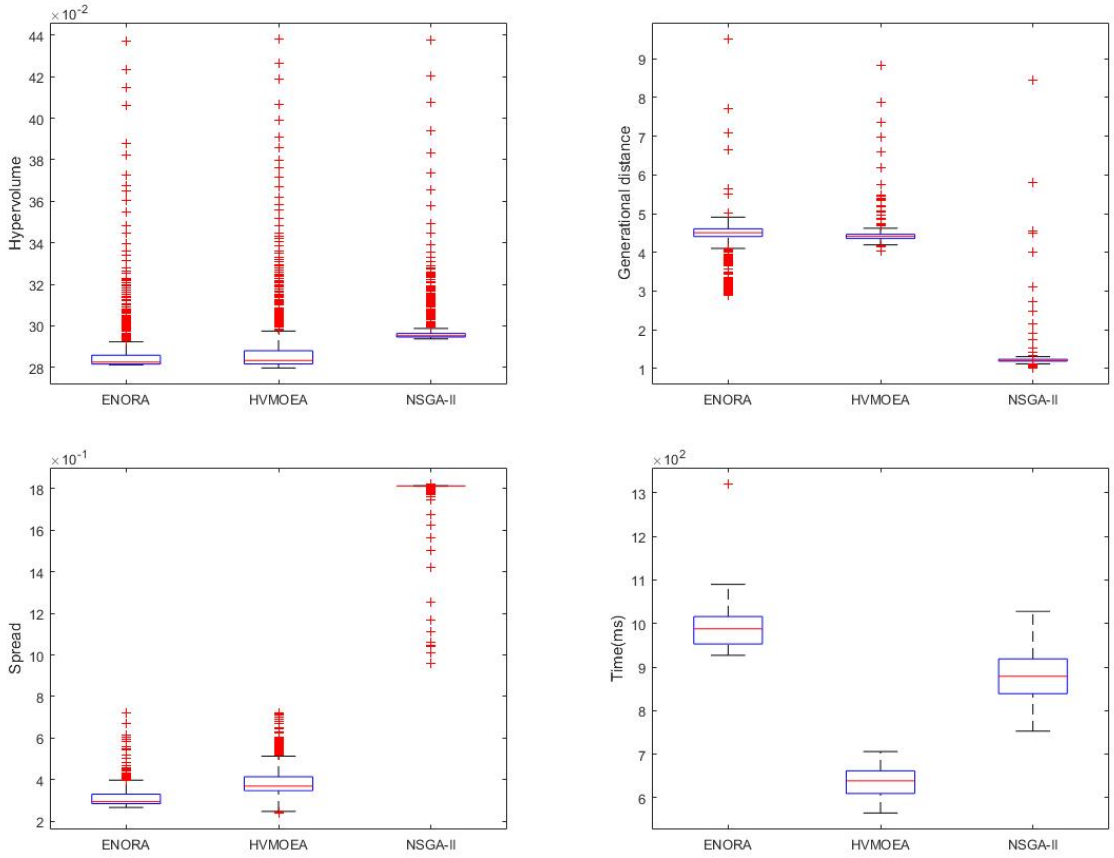


Figura 8: Diagramas de cajas sobre 30 ejecuciones con J48+ACC de *HVMOEA*, *ENORA* y *NSGA-II*, para la base de datos *ALL_AGENTS*.

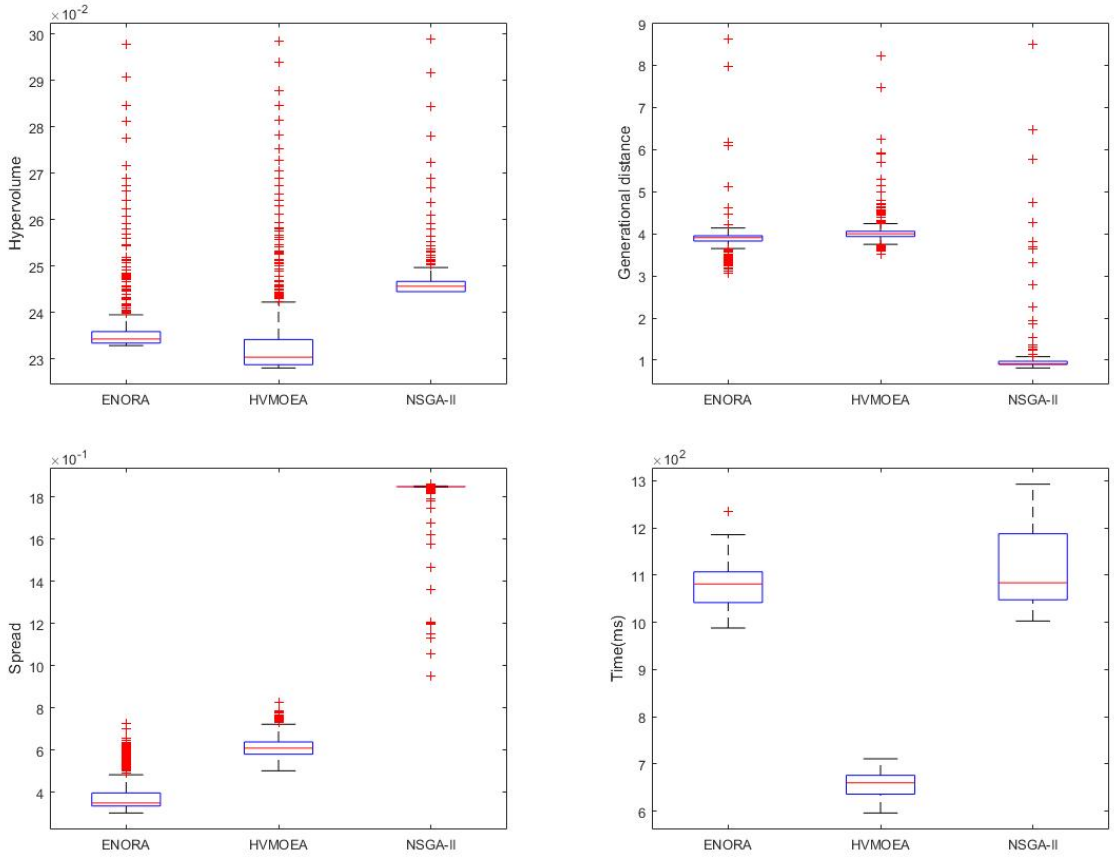


Figura 9: Diagramas de cajas sobre 30 ejecuciones con RandomForest+ACC de *HVMOEA*, *ENORA* y *NSGA-II*, para la base de datos *hcc survival*.

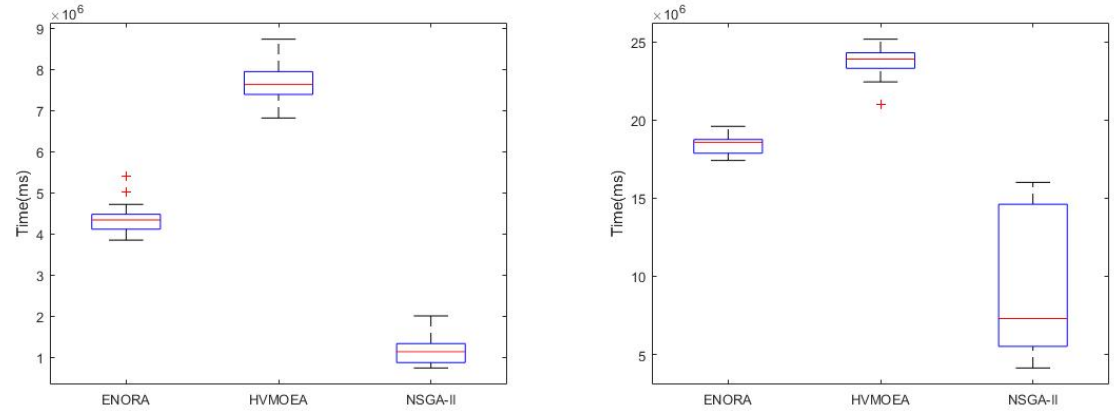


Figura 10: Diagramas de cajas sobre 30 ejecuciones de *HVMOEA*, *ENORA* y *NSGA-II* con J48+ACC (izquierda), *HVMOEA*, *ENORA* y *NSGA-II* con RandomForest+ACC (derecha), para las bases de datos *ALL_AGENTS* y *hcc survival* respectivamente.

El segundo conjunto de experimentos ha consistido en comparar el hipervolumen de cada algoritmo con el mismo algoritmo, pero usando una forma distinta de distribuir la población en slots, por lo que la misma población inicial evolucionará de distinta forma. En este caso se ha comparado la métrica de *hipervolumen* usando t-test con intervalos de confianza del 95 %. Los test estadísticos no obtienen diferencias significativas en el caso de *ENORA* y, en *HVMOEA*, obtienen que la distribución de slots *MAX* es peor que todas las demás, entre las cuales tampoco existen diferencias significativas. En las figuras 11 y 12 se observan los diagramas de cajas del experimento sobre las dos bases de datos.

Dados estos resultados podemos concluir que, para el escenario de estudio descrito, no existe una mejora significativa en el hipervolumen modificando la forma de distribuir la población en slots de los algoritmos *ENORA* y *HVMOEA*.

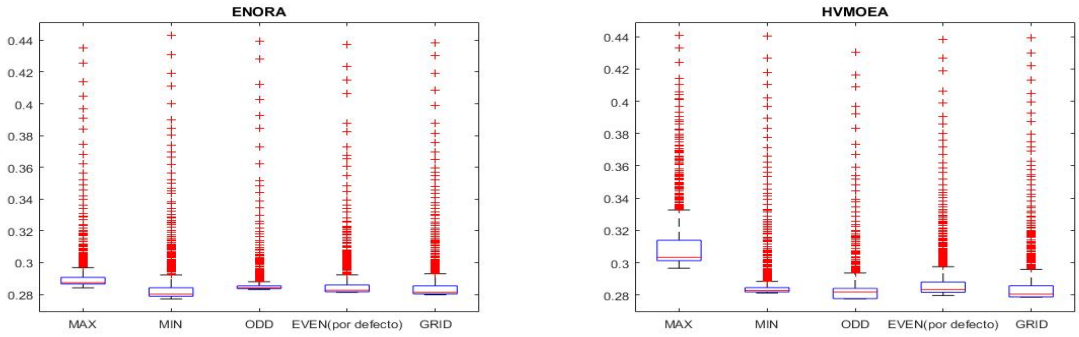


Figura 11: Diagramas de cajas sobre 30 ejecuciones con J48+ACC de *ENORA* y *HVMOEA* para la base de datos *ALL_AGENTS* usando 5 formas distintas de distribuir los individuos en los slots.

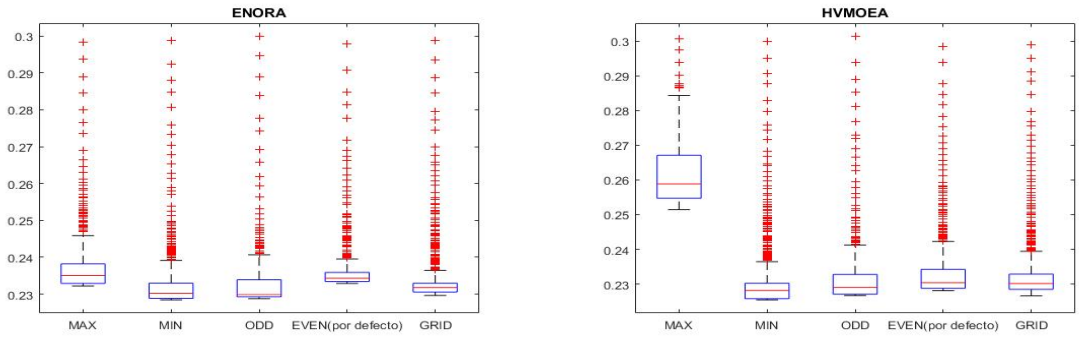


Figura 12: Diagramas de cajas sobre 30 ejecuciones con RandomForest+ACC de *ENORA* y *HVMOEA* para la base de datos *hcc survival* usando 5 formas distintas de distribuir los individuos en los slots.

Actualmente se están realizando nuevas pruebas con *HVMOEA* tanto en selección de atributos con más de dos objetivos, como en problemas many-objective de optimización de parámetros reales con restricciones.

5. Conclusiones

Dos procesos clave del análisis de datos, como son la selección de atributos y la clasificación interpretable, son abordados en esta tesis desde la perspectiva de la computación evolutiva multi-objetivo. La seleccción de atributos permite reducir la dimensionalidad de los datos al mismo tiempo que aumenta la capacidad de predicción de los sistemas clasificadores y disminuye la complejidad de estos y su tiempo de entrenamiento y respuesta. La clasificación interpretable permite a los expertos validar los sistemas de clasificación, así como ayudar a la toma de decisiones y a su justificación en contextos donde los aspectos éticos lo requieren. La computación evolutiva multi-objetivo encuentra soluciones aproximadas en problemas complejos,

intratables con técnicas clásicas, resultando técnicas muy robustas, eficaces, eficientes y escalables en contextos de búsqueda y optimización multi-objetivo no lineales de alta dimensionalidad.

En esta tesis doctoral, la computación evolutiva multi-objetivo se ha utilizado para selección de atributos en tareas de clasificación interpretable, tanto con árboles de decisión como con sistemas de clasificación basada en reglas, ambas fuzzy (con datos numéricos) y crisp (con datos categóricos). El algoritmo evolutivo multi-objetivo *ENORA*, que se diseñó inicialmente para optimización multi-objetivo de parámetros reales, ha sido adaptado en esta tesis para selección de atributos y clasificación basada en reglas. La selección de atributos se ha abordado como un problema de optimización multi-objetivo booleana, mientras que la clasificación basada en reglas ha consistido en un problema de optimización multi-objetivo mixta (combinatoria y de parámetros reales) con restricciones. El algoritmo *ENORA* ha sido aplicado en esta tesis para el descubrimiento de fármacos y para la clasificación de las habilidades profesionales de los agentes de un centro de contacto en una empresa de Italia. Además se han utilizado bases de datos públicas del *UCI Machine Learning Repository* lo que posibilita la reproducibilidad de los resultados. *ENORA* ha sido ampliamente comparado con el famoso algoritmo evolutivo multi-objetivo *NSGA-II*, así como con otras estrategias de búsqueda, tanto deterministas (*BestFirst*, *GreedyStepwise*, *LinearForwardSelection*, *Ranker*) como probabilistas (*GeneticSearch*, *PSOSearch*), con otros métodos de screening virtual (*BINDSURF*, *DOCK*, *ICM*, *GLIDE*, *NNET_EE246*, *SVM_AE246*, *LIGSIFT*, *DeepVS-Dock*, *DeepVS-ADV*, *C_RF_SVM*, *C_RF_NNET*, *C_RF_RF*), y con otros sistemas de clasificación basados en reglas (*JRip*, *PART*, *OneR ZeroR*) y basados en reglas fuzzy (*FRBCS.CHI*, *FRBCS.W*, *FH.GBML*, *FURIA*). Para las comparaciones se han realizado tests estadísticos, tanto paramétricos como no paramétricos, sobre las métricas *Hipervolumen*, *Accuracy*, *TP Rate*, *FP Rate Precision*, *Recall*, *F-Measure*, *MCC*, *ROC Area*, *PRC Area*, *RMSE*, *Serialized Model Size*, y los modelos se han evaluado tanto en *full training* como en *validación cruzada de 10 repeticiones* y *pertentage split*, de cara a analizar el grado de *overfitting* y la capacidad de predicción. El algoritmo *ENORA* se ha integrado en la plataforma *Weka* en dos paquetes oficiales (*MultiObjectiveEvolutionarySearch* como estrategia de búsqueda de métodos de selección de atributos, y *MultiObjectiveEvolutionaryFuzzyClassifier* como clasificador basado en reglas fuzzy y crisp con datos numéricos y categóricos), bajo *Licencia Pública General de GNU*.

Del amplio conjunto de experimentos que se han realizado en los distintos escenarios de la tesis, podemos destacar las siguientes conclusiones:

1. La optimización multi-objetivo, en donde además de optimizarse una métrica de rendimiento se minimiza el número de atributos, en el caso del problema de la selección de atributos, o el número de reglas, en clasificación basada en reglas, supone una forma natural de afrontar estos problemas, ya que siempre es deseable reducir la complejidad de los modelos de clasificación para una mayor interpretabilidad y para una reducción del tiempo de aprendizaje y de respuesta. Los algoritmos evolutivos multi-objetivo son una herramienta ideal para este tipo de problemas.

2. El amplio conjunto de tests estadísticos realizados con los algoritmos *ENORA* y *NSGA-II* nos hacen concluir que *ENORA* obtiene mejores valores del indicador de hipervolumen que *NSGA-II*, tanto en problemas de selección atributos como para la construcción de clasificadores basados en reglas, ambas fuzzy y crisp. Esta superioridad se debe fundamentalmente a que *ENORA* refuerza la diversidad de las poblaciones mediante un sistema de búsqueda basado en la división en slots del espacio de búsqueda de los objetivos.

3. Los clasificadores basados en reglas, tanto fuzzy como crisp, construidos con el método *MultiObjectiveEvolutionaryFuzzyClassifier* mediante el algoritmo *ENORA*, son poco propensos al overfitting, en comparación con otros clasificadores basados en reglas, según los experimentos realizados sobre las bases de datos analizadas.

4. Aunque no se han detectado diferencias estadísticamente significativas, la métrica *ACC* (accuracy) ha mostrado los mejores resultados, frente a las métricas *AUC* (área bajo la curva *ROC*) y *RMSE* (root-mean-square error), como métrica de evaluación en métodos wrapper multivariate de selección de atributos y como función de evaluación en sistemas de clasificación basada en reglas tanto fuzzy como crisp.

5. El método wrapper multivariate de selección de atributos formado por la estrategia de búsqueda *MultiObjectiveEvolutionarySearch* mediante *ENORA*, y el evaluador *WrapperSubsetEval* con el clasificador *J48* y la métrica *ACC*, de *Weka*, se ha comportado satisfactoriamente en la búsqueda de árboles de decisión compactos y precisos para el descubrimiento de fármacos, obteniendo un mejor equilibrio interpretabilidad-precisión que los trabajos reportados en la literatura.

6. El clasificador *MultiObjectiveEvolutionaryFuzzyClassifier*, que puede construir reglas fuzzy, crisp y mixtas según sean el tipo de datos de los atributos de la base de datos, ha demostrado un comportamiento muy satisfactorio en la búsqueda de clasificadores interpretables y precisos, en comparación con otros clasificadores basados en reglas de la literatura ampliamente consolidados.

7. El método wrapper multivariate de selección de atributos formado por la estrategia de búsqueda *MultiObjectiveEvolutionarySearch* mediante *ENORA*, y el evaluador *WrapperSubsetEval* con el clasificador *MultiObjectiveEvolutionaryFuzzyClassifier* mediante *ENORA* y la métrica *ACC*, de *Weka*, es altamente aconsejable como paso previo en problemas de clasificación basada en reglas fuzzy para la obtención de clasificadores interpretables y precisos. Debido a la complejidad computacional del método se aconseja un número de evaluaciones del orden de 100 para *MultiObjectiveEvolutionarySearch* y del orden de 10 para *MultiObjectiveEvolutionaryFuzzyClassifier* en la fase de selección de atributos, y un número de evaluaciones superior a 1000 en la fase de clasificación final.

Estas conclusiones nos llevan a sugerir y aconsejar, desde esta tesis doctoral, que los métodos *MultiObjectiveEvolutionarySearch* y *MultiObjectiveEvolutionaryFuzzyClassifier* sean tenidos en cuenta en la búsqueda de clasificadores compactos, interpretables y precisos en ambientes supervisados para cualquier tipo de aplicación que requiera una explicación razonable del comportamiento del modelo. No obstante, los métodos deben ser configurados adecuadamente para un buen rendimiento, para lo cual se debe hacer uso de las guías de configuración incluidas en las publicaciones que componen la tesis.

6. Trabajos futuros

Los siguientes trabajos están siendo actualmente realizados o previstos para el fututo:

1. Se tienen resultados parciales del nuevo algoritmo evolutivo multi-objetivo basado en hipervolumen *HVMOEA* para problemas de optimización con restricciones de parámetros reales con muchos objetivos y para selección de atributos con más de dos objetivos. Estos trabajos se están actualmente preparando para su publicación en revistas internacionales del *JCR*.

2. La *Evolución Diferencial Multi-Objetivo* (*MODE*) es una técnica evolutiva que actualmente está siendo objeto de mucha atención por la comunidad científica. Nuestro equipo de investigación está implementado cuatro algoritmos de la familia *MODE*, en concreto *DE/best/1/bin* y *DE/rand/1/bin*, con representaciones binaria y real en ambos casos. Es importante disponer de estos nuevos algoritmos integrados en nuestra plataforma de desarrollo software para usarlos en las comparaciones en los nuevos algroitmos que se van desarrollando.

3. Además de las cuatro variantes de la familia *MODE*, estamos desarrollando un nuevo algoritmo *MODE* con refuerzo de la diversidad inspirado en el algoritmo *ENORA*, en donde el espacio de los objetivos se divide el slots y los individuos se ordenan según el ranking en su slot. Este nuevo algoritmo será aplicado tanto en optimización multi-objetivo con restricciones de parámetros reales, como en selección de atributos y clasificación basada en reglas.

4. Tenemos prevista la publicación inminente "*Multi-objective Performance Evaluation of Ranking Feature Selection Methods*", en donde se proponen dos nuevos métodos multivariate de selección de atributos basados en ranking. Normamente los métodos de selección de atributos basados en ranking son univariate (con algunas excepciones como *ReliefF*) y por tanto no tienen en cuenta las interacciones entre los atributos. Además proponemos una métrica multi-objetivo para evaluar los métodos de selección de atributos basados en ranking, la cual usa un amplio conjunto de métricas tanto *subset evaluation* como *attribute evaluation*, de tipo filter y de tipo wrapper. La métrica evalúa los distintos métodos, incluidos métodos *ensemble*, y proporciona un ranking de los mismos, por lo que resulta, en sí misma, un método de selección de atributos (los atributos selecciónados por el mejor método en el ranking).

5. También muy desarrollado tenemos un algoritmo evolutivo multi-objetivo para selección de instancias, en tareas tanto de clasificación como de regresión. Este algoritmo sigue un modelo de optimización similar al de selección de atributos, es decir, optimizar una métrica de rendimiento y minimizar el número de instancias, pudiéndose configurar tanto de tipo filter (con por ejemplo, funciones de consistencia), como de tipo wrapper usando algún clasificador específico y una métrica de evaluación. Para problemas de clasificación se requiere añadir restricciones al modelo de optimización para asegurar que se selecciona al menos una instancia para cada clase.

6. Una vez validado el algoritmo de selección de instancias, se extenderá a la selección simultánea de instancias y atributos, considerando en este caso un problema de optimización de 3 objetivos. En este punto será necesario comprobar que la optimización simultánea instancias + atributos es más eficiente que realizar ambos procesos secuencialmente.

7. En el campo de los sistemas evolutivos fuzzy tenemos en mente tres posibles extensiones: *i)* considerar conjuntos fuzzy *type-2*, *ii)* extender la representación de las reglas para que cada regla pueda contener un número variable de atributos, y *iii)* considerar sistemas de regresión basados en reglas fuzzy, tipo *TSK*. En este último apartado, aunque está ya desarrollado, nos disponemos a su integración en *Weka* con el nombre *MultiObjectiveEvolutionaryFuzzyRegression*.

8. Finalmente, actualmente se está trabajando en la aplicación de *MultiObjectiveEvolutionarySearch* para la selección de atributos en problemas de pronóstico de series temporales con aplicación en la resistencia de los antibióticos. El método *MultiObjectiveEvolutionaryFuzzyRegression* será requerido en este caso para la interpretación de los modelos de pronóstico de las series temporales.

Cartas de aceptación

A methodology for evaluating multi-objective evolutionary feature selection for classification in the context of virtual screening

Asunto: SOCO: Editor Decision - Accept
Fecha: 13 Aug 2018 18:49:55 -0400
De: Soft Computing (SOCO) <em@editorialmanager.com>
Responder a: Soft Computing (SOCO) <parthiban.gurusamy@springer.com>
Para: Fernando J. Jiménez <fernan@um.es>

Ref.: Ms. No. SOCO-D-17-00891R1
A methodology for evaluating multi-objective evolutionary feature selection for classification in the context of virtual screening
Soft Computing

Dear Dr. Jiménez,

I am pleased to tell you that your work has now been accepted for publication in Soft Computing.

Comments from the Editor and Reviewers can be found below.

Thank you for submitting your work to this journal.

With kind regards

Raffaele Cerulli
Managing Editor
Soft Computing

Figura 13: Carta de aceptación de Soft Computing

Referencia completa

Fernando Jiménez, Horacio Pérez-Sánchez, José Palma, Gracia Sánchez y Carlos Martínez. A methodology for evaluating multi-objective evolutionary feature selection for classification in the context of virtual screening. Soft Computing, 2018, p. 1-26.

Clasificación evolutiva multi-objetivo basada en reglas con datos categóricos

Asunto: [Entropy] Manuscript ID: entropy-343135 - Accepted for Publication
Fecha: 06/09/18 (07:06:19 CET)
De: Shayna Tang
Para: Fernando Jiménez
Cc: Carlos MartínezLuis Miralles-PechuánGracia SánchezGuido SciaviccoEntropy Editorial OfficeShayna Tang

Dear Dr. Jiménez,

We are pleased to inform you that the following paper has been officially accepted for publication:

Manuscript ID: entropy-343135
Type of manuscript: Article
Title: Multi-Objective Evolutionary Rule-Based Classification with Categorical Data
Authors: Fernando Jiménez *, Carlos Martínez, Luis Miralles-Pechuán, Gracia Sánchez, Guido Sciavicco
Received: 30 July 2018
E-mails: fernan@um.es, carlos.martinez6@um.es, luis.miralles@ucd.ie, gracia@um.es, scvgdu@unife.it
Statistical Machine Learning for Human Behaviour Analysis
http://www.mdpi.com/journal/entropy/special_issues/Statistical_Machine_Learning
http://susy.mdpi.com/user/manuscripts/review_info/f4a758029e9ca50a65e81f89fdbdcb81

We will now make the final preparations for publication, then return the manuscript to you for your approval.

We also invite you to contribute to Encyclopedia (https://encyclopedia.pub), a scholarly platform providing accurate information about the latest research results. You can adapt parts of your paper to provide valuable reference information for others in the field.

Kind regards,
Ms. Shayna Tang
Assistant Editor
E-Mail: shayna.tang@mdpi.com

MDPI AG
Entropy Editorial Office
E-Mail: entropy@mdpi.com
http://www.mdpi.com/journal/entropy/

Figura 14: Carta de aceptación de Entropy

Referencia completa

Fernando Jiménez, Carlos Martínez, Luis Miralles-Pechuán, Gracia Sánchez, y Guido Sciavicco. Multi-Objective Evolutionary Rule-Based Classification with Categorical Data. Entropy, 2018, vol. 20, no 9, p. 684.

Selección de atributos evolutiva multi-objetivo para clasificación fuzzy

Asunto: Review of Manuscript TFS-2018-0183.R2
Fecha: Mon, 17 Dec 2018 11:24:20 +0000
De: Transactions on Fuzzy Systems <onbehalfof@manuscriptcentral.com>
Responder a: Robert.John@nottingham.ac.uk
Para: fernan@um.es

Our Ref: DR1

17-Dec-2018

TFS-2018-0183.R2
Multi-objective Evolutionary Feature Selection for Fuzzy Classification
Special Issue Paper

Dear Prof. Jiménez:

I am pleased to inform you that your manuscript has been accepted for publication in IEEE Transactions on Fuzzy Systems.

This acceptance is, of course, contingent upon you revising the paper to address any final comments made by the referees – please see below, if the referees have attached a review file you will need to retrieve it from your 'author centre'.

Your final manuscript must be returned within ** 10 days** of receipt of this email.

The final version of your manuscript must comply with IEEE Periodicals requirements for text and graphics processing. Detailed information regarding the format of your final submission can be found here:
http://www.ieee.org/web/publications/authors/transjnl/index.html

We look forward to seeing your paper in IEEE Transactions on Fuzzy Systems.

Sincerely,

Figura 15: Carta de aceptación de IEEE Transactions on Fuzzy System

Referencia completa

Fernando Jiménez, Carlos Martínez, Enrico Marzano, José Palma, Gracia Sánchez y Guido Sciavicco. Multi-objective Evolutionary Feature Selection for Fuzzy Classification. IEEE Transactions on Fuzzy Systems, 2019.

Referencias

[1] I. H. Witten, E. Frank, M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition, Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, 2011 (2011).

[2] M. Berthold, M. Berthold, D. Hand, Intelligent Data Analysis: An Introduction, Springer Nature Book Archives Millennium, Springer, 2003 (2003).

[3] E. Alpaydin, Introduction to Machine Learning, 2nd Edition, The MIT Press, 2010 (2010).

[4] V. Kecman, Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models, MIT Press, Cambridge, MA, USA, 2001 (2001).

[5] S. S. Roy, P. Samui, R. Deo, S. Ntalampiras, Big Data in Engineering Applications, 1st Edition, Springer Publishing Company, Incorporated, 2018 (2018).

[6] Y. Li, T. Li, H. Liu, Recent advances in feature selection and its applications, Knowl. Inf. Syst. 53 (3) (2017) 551–577 (Dec. 2017). doi:10.1007/s10115-017-1059-8.

[7] C. Molnar, Interpretable Machine Learning, Lean Publishing, 2019 (2019).
URL http://leanpub.com/interpretable-machine-learning

[8] K. Deb, Multi-objective optimization using evolutionary algorithms, Wiley, London, UK, 2001 (2001).

[9] M. P. Basgalupp, A. C. P. L. F. D. Carvalho, R. C. Barros, D. D. Ruiz, A. A. Freitas, Lexicographic multi-objective evolutionary induction of decision trees, Int. J. Bio-Inspired Comput. 1 (1/2) (2009) 105–117 (Jan. 2009).

[10] M. Antonelli, P. Ducange, F. Marcelloni, A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers, Information Sciences 283 (2014) 36 – 54, new Trend of Computational Intelligence in Human-Robot Interaction (2014).

[11] S. Dehuri, S. Patnaik, A. Ghosh, R. Mall, Application of elitist multi-objective genetic algorithm for classification rule generation, Applied Soft Computing 8 (1) (2008) 477 – 487 (2008).

[12] F. Jiménez, A. Gómez-Skarmeta, G. Sánchez, K. Deb, An evolutionary algorithm for constrained multi-objective optimization, in: Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress, Vol. 2 of CEC '02, IEEE Computer Society, Washington, DC, USA, 2002, pp. 1133–1138 (2002).

[13] I. H. Witten, E. Frank, M. A. Hall, Introduction to weka, in: Data Mining: Practical Machine Learning Tools and Techniques (Third Edition), The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, Boston, 2011, pp. 403 – 406 (2011).

[14] C. McInnes, Virtual screening strategies in drug discovery, Current opinion in chemical biology 11 (5) (2007) 494–502 (2007).

[15] D. Dua, E. Karra Taniskidou, Uci machine learning repository [http://archive. ics. uci. edu/ml]. irvine, ca: University of california, School of Information and Computer Science (2017).

[16] T. Hubertus, M. Klaus, T. Eberhard, Optimization theory, Kluwer Academic, Dordrecht, 2004 (2004).

[17] S. Sinha, Mathematical Programming: Theory and Methods, Elsevier Science Limited, 2006 (2006).

[18] Y. Collette, P. Siarry, Multiobjective Optimization: Principles and Case Studies, Springer Berlin Heidelberg, 2004 (2004).

[19] H. Karloff, Linear Programming, Birkhauser Basel, Boston, MA, 1991 (1991).

[20] I. Maros, G. Mitra, Simplex algorithms, Oxford Science, 1996, Ch. 1, pp. 1–46 (1996).

[21] D. Bertsekas, Nonlinear Programming (Second ed.), Athena Scientific, Cambridge, MA, 1999 (1999).

[22] F. Jiménez, J. L. Verdegay, Computational Intelligence in Theory and Practice, Physica-Verlag HD, Heidelberg, 2001, Ch. Evolutionary Computation and Mathematical Programming, pp. 167–182 (2001).

[23] F. Jiménez, G. Sánchez, J. García, G. Sciavicco, L. Miralles, Multi-objective evolutionary feature selection for online sales forecasting, Neurocomputing 234 (2017) 75–92 (2017).

[24] K. Deb, A. Pratab, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii., IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182 – 197 (2002).

[25] C. Bao, L. Xu, E. D. Goodman, L. Cao, A novel non-dominated sorting algorithm for evolutionary multi-objective optimization, Journal of Computational Science 23 (2017) 31 – 43 (2017). doi:https://doi.org/10.1016/j.jocs.2017.09.015.
URL http://www.sciencedirect.com/science/article/pii/S1877750317310530

[26] F. Jiménez, G. Sánchez, P. Vasant, A multi-objective evolutionary approach for fuzzy optimization in production planning, J. Intell. Fuzzy Syst. 25 (2) (2013) 441–455 (2013).

[27] F. Jiménez, G. Sánchez, J. M. Juárez, Multi-objective evolutionary algorithms for fuzzy classification in survival prediction, Artificial Intelligence in Medicine 60 (3) (2014) 197–219 (2014).

[28] F. Jiménez, E. Marzano, G. Sánchez, G. Sciavicco, N. Vitacolonna, Attribute selection via multi-objective evolutionary computation applied to multi-skill contact center data classification, in: Proc. of the IEEE Symposium on Computational Intelligence in Big Data (IEEE CIBD 15), IEEE, 2015, pp. 488–495 (2015).

[29] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints, IEEE Transactions on Evolutionary Computation 18 (4) (2014) 577–601 (Aug 2014). doi:10.1109/TEVC.2013.2281535.

[30] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, Norwell, MA, USA, 1998 (1998).

[31] A. G. Karegowda, A. S. Manjunath, M. A. Jayaram, Comparative study of attribute selection using gain ratio and correlation based feature selection, International Journal of Information Technology and Knowledge Management 2 (2) (2010) 271–277 (2010).

[32] M. A. Hall, Correlation-based feature selection for machine learning, Tech. rep., University of Waikato (1999).

[33] A. Ahmad, L. Dey, A feature selection technique for classificatory analysis, Pattern Recognition Letters 26 (1) (2005) 43–56 (2005).

[34] N. Japkowicz, M. Shah, Evaluating Learning Algorithms: A Classification Perspective, Cambridge University Press, New York, NY, USA, 2011 (2011).

[35] E. Amaldi, V. Kann, On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems, Theoretical Computer Science 209 (1) (1998) 237 – 260 (1998). doi:https://doi.org/10.1016/S0304-3975(97)00115-1.
URL http://www.sciencedirect.com/science/article/pii/S0304397597001151

[36] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989 (1989).

[37] W. Siedlecki, J. Sklansky, A note on genetic algorithms for large-scale feature selection, Pattern Recognition Letters 10 (5) (1989) 335 – 347 (1989).

[38] H. Vafaie, K. De Jong, Genetic algorithms as a tool for feature selection in machine learning, in: Tools with Artificial Intelligence, 1992. TAI '92, Proceedings., Fourth International Conference on, 1992, pp. 200–203 (Nov 1992).

[39] M. ElAlami, A filter model for feature subset selection based on genetic algorithm, Knowledge-Based Systems 22 (5) (2009) 356 – 362 (2009).

[40] R. Anirudha, R. Kannan, N. Patil, Genetic algorithm based wrapper feature selection on hybrid prediction model for analysis of high dimensional data, in: Industrial and Information Systems (ICIIS), 2014 9th International Conference on, 2014, pp. 1–6 (Dec 2014).

[41] J. Huang, Y. Cai, X. Xu, A hybrid genetic algorithm for feature selection wrapper based on mutual information, Pattern Recognition Letters 28 (13) (2007) 1825–1844 (2007).

[42] A. F. Gómez-Skarmeta, F. Jiménez, J. Ibánez, S. Paredes, Evolutionary variable identification, in: Proceedings of 7th European congress on intelligent techniques and soft computing (EUFIT'99), 1999 (1999).

[43] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, Intelligent Systems and their Applications, IEEE 13 (2) (1998) 44–49 (Mar 1998).

[44] S. Dreyer, Evolutionary feature selection, in: Norwegian University of Science and Technology, Department of Computer and Information Science, Institutt for datateknikk og informasjonsvitenskap, 2013, p. 76 (2013).

[45] H. Ishibuchi, T. Nakashima, Multi-objective pattern and feature selection by a genetic algorithm, in: Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation, Morgan Kaufmann Publishers Inc., 2000, pp. 1069–1076 (2000).

[46] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. C. Coello, A survey of multiobjective evolutionary algorithms for data mining (part I), IEEE Transactions on Evolutionary Computation 18 (1) (2014) 4–19 (2014).

[47] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. C. Coello, A survey of multiobjective evolutionary algorithms for data mining (part II), IEEE Transactions on Evolutionary Computation 18 (1) (2014) 20–35 (2014).

[48] H. Chen, X. Yao, Evolutionary multiobjective ensemble learning based on bayesian feature selection, in: Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, 2006, pp. 267–274 (2006).

[49] Y. Jin (Ed.), Multi-Objective Machine Learning, Vol. 16 of Studies in Computational Intelligence, Springer, Warsaw, Poland, 2006 (2006).

[50] J. García-Nieto, E. Alba, L. Jourdan, E. Talbi, Sensitivity and specificity based multiobjective approach for feature selection: Application to cancer diagnosis, Information Processing Letters 109 (16) (2009) 887 – 896 (2009).

[51] Z. Zhu, Y.-S. Ong, J.-L. Kuo, Feature selection using single/multi-objective memetic frameworks, in: C.-K. Goh, Y.-S. Ong, K. Tan (Eds.), Multi-Objective Memetic Algorithms, Vol. 171 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2009, pp. 111–131 (2009).

[52] M. Venkatadri, K. Srinivasa Rao, A multiobjective genetic algorithm for feature selection in data mining, International Journal of Computer Science and Information Technologies 1 (5) (2010) 443–448 (2010).

[53] A. Ekbal, S. Saha, C. Garbe, Feature selection using multiobjective optimization for named entity recognition, in: Pattern Recognition (ICPR), 2010 20th International Conference on, 2010, pp. 1937–1940 (Aug 2010).

[54] A. P. Reynolds, D. W. Corne, M. J. Chantler, Feature selection for multi-purpose predictive models: a many-objective task, in: Parallel Problem Solving from Nature, PPSN XI, Springer, 2010, pp. 384–393 (2010).

[55] A. Gaspar-Cunha, Feature selection using multi-objective evolutionary algorithms: Application to cardiac spect diagnosis, in: M. Rocha, F. Riverola, H. Shatkay, J. Corchado (Eds.), Advances in Bioinformatics, Vol. 74 of Advances in Intelligent and Soft Computing, Springer Berlin Heidelberg, 2010, pp. 85–92 (2010).

[56] A. Gaspar-Cunha, J. A. Covas, Rpsgae - reduced pareto set genetic algorithm: Application to polymer extrusion, in: X. Gandibleux, M. Sevaux, K. Sorensen, V. T kindt (Eds.), Metaheuristics for Multiobjective Optimisation, Vol. 535 of Lecture Notes in Economics and Mathematical Systems, Springer Berlin Heidelberg, 2004, pp. 221–249 (2004).

[57] L. Li, M. Li, Y. Lu, Y. Zhang, A new multi-objective genetic algorithm for feature subset selection in fatigue fracture image identification, JCP 5 (7) (2010) 1105–1111 (2010).

[58] P. A. Castro, F. J. Von Zuben, Multi-objective feature selection using a bayesian artificial immune system, International Journal of Intelligent Computing and Cybernetics 3 (2) (2010) 235–256 (2010).

[59] I. Vatolkin, M. Preuß, G. Rudolph, Multi-objective feature selection in music genre and style recognition tasks, in: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO '11, ACM, New York, NY, USA, 2011, pp. 411–418 (2011).

[60] N. Beume, B. Naujoks, M. Emmerich, Sms-emoa: Multiobjective selection based on dominated hypervolume, European Journal of Operational Research 181 (3) (2007) 1653 – 1669 (2007).

[61] A. Jara, R. Martínez, D. Vigueras, G. Sánchez, F. Jiménez, Attribute selection by multiobjective evolutionary computation applied to mortality from infection in severe burns patients, in: HEALTHINF 2011 - Proceedings of the International Conference on Health Informatics, Rome, Italy, 26-29 January, 2011, 2011, pp. 467–471 (2011).

[62] B. Krishna, B. Kaliaperumal, Efficient genetic-wrapper algorithm based data mining for feature subset selection in a power quality pattern recognition application, Int. Arab J. Inf. Technol. 8 (4) (2011) 397 – 405 (2011).

[63] H. Karshenas, P. Larrañaga Múgica, Q. Zhang, C. Bielza, An interval-based multiobjective approach to feature subset selection using joint modeling of objectives and variables, Tech. rep., Facultad de Informática, Universidad Politécnica de Madrid (2012).

[64] J. Zhao, V. B. Fernandes, L. Jiao, I. Yevseyeva, A. Maulana, R. Li, T. Bäck, M. T. M. Emmerich, Multiobjective optimization of classifiers by means of 3-d convex hull based evolutionary algorithm, CoRR abs/1412.5710 (2014).

[65] T. Fawcett, An introduction to roc analysis, Pattern Recogn. Lett. 27 (8) (2006) 861–874 (Jun. 2006).

[66] A. Gaspar-Cunha, G. Recio, L. Costa, C. Estébanez, Self-adaptive moea feature selection for classification of bankruptcy prediction data, The Scientific World Journal 2014 (2014).

[67] D. Kimovski, J. Ortega, A. Ortiz, R. Banos, Parallel alternatives for evolutionary multi-objective optimization in unsupervised feature selection, Expert Systems with Applications 42 (9) (2015) 4239 – 4252 (2015).

[68] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (4) (1997) 341–359 (Dec 1997). doi:10.1023/A:1008202821328.

[69] U. K. Sikdar, A. Ekbal, S. Saha, Mode: multiobjective differential evolution for feature selection and classifier ensemble, Soft Computing 19 (12) (2015) 3529–3549 (Dec 2015). doi:10.1007/s00500-014-1565-5.

[70] S. K. Nayak, P. K. Rout, A. K. Jagadev, T. Swarnkar, Elitism based multi-objective differential evolution for feature selection: A filter approach with an efficient redundancy measure, Journal of King Saud University - Computer and Information Sciences (2017). doi:https://doi.org/10.1016/j.jksuci.2017.08.001.

[71] U. Mlakar, I. Fister, J. Brest, B. Potocnik, Multi-objective differential evolution for feature selection in facial expression recognition systems, Expert Systems with Applications 89 (2017) 129 – 137 (2017). doi:https://doi.org/10.1016/j.eswa.2017.07.037.

[72] L. X. Wang, J. M. Mendel, Generating fuzzy rules by learning from examples, in: Proceedings of the 1991 IEEE International Symposium on Intelligent Control, 1991, pp. 263–268 (1991).

[73] Z. Chi, H. Yan, T. Pham, Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition, World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1996 (1996).

[74] H. Ishibuchi, T. Nakashima, Effect of rule weights in fuzzy rule-based classification systems, IEEE Transactions on Fuzzy Systems 9 (4) (2001) 506–515 (2001).

[75] H. Ishibuchi, T. Yamamoto, T. Nakashima, Hybridization of fuzzy gbml approaches for pattern classification problems, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 35 (2) (2005) 359–365 (2005).

[76] M. Gacto, R. Alcalá, F. Herrera, Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems, Soft Computing 13 (5) (2009) 419–436 (Dec. 2009).

[77] P. Ducange, B. Lazzerini, F. Marcelloni, Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets, Soft Computing 14 (7) (2010) 713–728 (2010).

[78] P. Ducange, G. Mannara, F. Marcelloni, R. Pecori, M. Vecchio, A novel approach for internet traffic classification based on multi-objective evolutionary fuzzy classifiers, in: Fuzzy Systems (FUZZ-IEEE), 2017 IEEE International Conference on. IEEE, 2017, pp. 1–6 (07 2017).

[79] M. Antonelli, D. Bernardo, H. Hagras, F. Marcelloni, Multiobjective evolutionary optimization of type-2 fuzzy rule-based systems for financial data classification, IEEE Trans. Fuzzy Systems 25 (2) (2017) 249–264 (2017). doi:10.1109/TFUZZ.2016.2578341.
URL https://doi.org/10.1109/TFUZZ.2016.2578341

[80] V. Kreinovich, C. Quintana, L. Reznik, Gaussian membership functions are most adequate in representing uncertainty in measurements, in: Proceedings of the NAFIPS'92, North American Fuzzy Information Processing Society Conference, Puerto Vallarta, 1992, pp. 618–624 (1992).

[81] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. N. Shindyalov, P. E. Bourne, The protein data bank, Nucleic acids research 28 (1) (2000) 235–242 (2000).

[82] Package caret, http://cran.r-project.org/web/packages/caret/caret.pdf (2015).

[83] C. E. Metz, Basic principles of ROC analysis, Seminars in Nuclear Medicine 8 (1978) 283–298 (1978).

[84] S. Salzberg, C4.5: Programs for machine learning by J. Ross Quinlan, Machine Learning 16 (3) (1994) 235–240 (1994).

[85] D. J. Hand, Measuring classifier performance: a coherent alternative to the area under the roc curve, Machine Learning 77 (1) (2009) 103–123 (Oct 2009). doi:10.1007/s10994-009-5119-5.

[86] A. Moraglio, C. Di Chio, R. Poli, Geometric particle swarm optimisation, in: M. Ebner, M. OŃeill, A. Ekárt, L. Vanneschi, A. Esparcia-Alcázar (Eds.), Genetic Programming, Vol. 4445 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 125–136 (2007).

[87] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 1137–1143 (1995).

[88] C. Nadeau, Y. Bengio, Inference for the generalization error, Machine Learning (2001).

[89] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. A. C. Coello, A survey of multiobjective evolutionary algorithms for data mining: Part i, IEEE Transactions on Evolutionary Computation 18 (1) (2014) 4–19 (Feb 2014). doi:10.1109/TEVC.2013.2290086.

[90] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. A. C. Coello, Survey of multiobjective evolutionary algorithms for data mining: Part ii, IEEE Transactions on Evolutionary Computation 18 (1) (2014) 20–35 (Feb 2014). doi:10.1109/TEVC.2013.2290082.

[91] H. Ishibuchi, T. Murata, I. Turksen, Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems, Fuzzy Sets and Systems 89 (2) (1997) 135 – 150 (1997).

[92] M. Srinivas, L. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Transactions on Systems, Man, and Cybernetics 24 (4) (1994) 656–667 (1994).

[93] C. Wang, M. Shao, Q. He, Y. Qian, Y. Qi, Feature subset selection based on fuzzy neighborhood rough sets, Knowledge-Based Systems 111 (2016) 173 – 179 (2016). doi:https://doi.org/10.1016/j.knosys.2016.08.009.
URL http://www.sciencedirect.com/science/article/pii/S0950705116302714

[94] M. Cintra, H. Camargo, M.-C. Monard, Fuzzy feature subset selection using the wang & mendel method, in: Proceedings - 8th International Conference on Hybrid Intelligent Systems, HIS 2008, 2008, pp. 590–595 (10 2008).

[95] F. Jiménez, R. Jodár, G. Sánchez, M. Martín, G. Sciavicco, Multi-objective evolutionary computation based feature selection applied to behaviour assessment of children, in: Proc. of the 2016 International Conference on Educational Data Mining (ICEDM), Vol. 2(6), 2016, pp. 1888–1897 (2016).

[96] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evolutionary Computation 8 (2) (2000) 173 – 195 (2000).

[97] K. Li, R. Wang, T. Zhang, H. Ishibuchi, Evolutionary many-objective optimization: A comparative study of the state-of-the-art, IEEE Access 6 (2018) 1–1 (05 2018). doi:10.1109/ACCESS.2018.2832181.

[98] J. Bader, E. Zitzler, Hype: An algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. 19 (1) (2011) 45–76 (Mar. 2011).

[99] A. Menchaca-Mendez, C. A. Coello Coello, An alternative hypervolume-based selection mechanism for multi-objective evolutionary algorithms, Soft Comput. 21 (4) (2017) 861–884 (Feb. 2017).

[100] L. Riza, C. Bergmeir, F. Herrera, J. Benítez, frbs: Fuzzy rule-based systems for classification and regression in R, Journal of Statistical Software, Articles 65 (6) (2015) 1–30 (2015).

Anexos

A. Material Suplementario

A.1. Selección de atributos en *Weka*

En la herramienta de código abierto *Weka* [13] la selección de atributos se realiza con el paquete *weka.attributeSelection.AttributeSelection* de la clase *weka.attributeSelection* a través de dos componentes: la *estrategia de búsqueda* (clase abstracta *weka.attributeSelection.ASSearch*) y el *evaluador* (clase abstracta *weka.attributeSelection.ASEvaluation*). Esto permite a los usuarios y programadores configurar una multitud de métodos diferentes para la selección de atributos, tanto filter como wrapper, univariate y multivariate. Los evaluadores con nombres que terminan en *SubsetEval* configuran métodos de evaluación de subconjuntos de atributos, mientras que aquellos con nombres que terminan en *AttributeEval* configuran métodos de evaluación de atributos de forma individual. Para los métodos wrapper multivariate, el paquete *weka.attributeSelection* tiene la clase *weka.attributeSelection.WrapperSubsetEval* que evalúa los conjuntos de atributos utilizando un esquema de aprendizaje con validación cruzada y una medida de rendimiento. Para métodos wrapper univariate, la clase *weka.attributeSelection.ClassifierAttributeEval* evalúa el valor de un atributo utilizando un clasificador especificado por el usuario, una validación cruzada y una medida de evaluación de desempeño para usar para seleccionar los atributos.

Dado que tanto la selección de atributos como los procesos de clasificación deben ejecutarse en modo por lotes, *Weka* ofrece el meta-clasificador *weka.classifiers.meta.AttributeSelectedClassifier*, que reduce la dimensionalidad de los datos al realizar una selección de atributos antes de pasar los datos a un clasificador. El meta-clasificador *weka.classifiers.meta.AttributeSelectedClassifier* implementa la clase *weka.classifiers.SingleClassifierEnhancer* que es una clase de utilidad abstracta para manejar configuraciones comunes a los meta clasificadores que usan un solo aprendiz de base, extendiéndose a su vez clasificador abstracto *weka.classifiers.AbstractClassifier*.

A.2. El paquete *MultiObjectiveEvolutionarySearch* de *Weka*

El paquete de *Weka weka.attributeSelection* contiene la clase *MultiObjectiveEvolutionarySearch*. Esta clase hereda de la clase abstracta *weka.attributeSelection.ASSearch*, e implementa las interfaces *weka.attributeSelection.StartSetHandler* y *weka.core.OptionHandler*. *MultiObjectiveEvolutionarySearch* es un método de búsqueda que explora subconjuntos de atributos, por lo que debe ejecutarse junto con evaluadores que implementan la interfaz *weka.attributeSelection.SubsetEvaluator* (métodos de selección multivarite), y permite el uso de los métodos de búsqueda *ENORA* y *NSGA-II* con las siguientes opciones, entre otras (tenga en cuenta que no hay opciones para establecer las probabilidades de cruce y mutación, ya que estas son adaptativas):

- *Algorithm*, para establecer el algoritmo (*ENORA* o *NSGA-II*);
- *PopulationSize*, para establecer el número de individuos en la población;
- *Generations*, para estableces el número de generaciones en las que evolucionará la población;
- *StartSet*, para establecer un punto de partida para la búsqueda (se incluye como un elemento de la población inicial).

A.3. El paquete *MultiObjectiveEvolutionaryFuzzyClassifier* de *Weka*

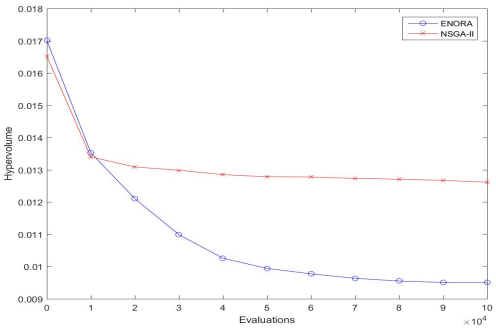
El paquete *Weka weka.classifiers.rules* contiene la clase *MultiObjectiveEvolutionaryFuzzyClassifier* que hereda de la clase abstracta *weka.classifiers.AbstractClassifier* e implementa la interfaz *weka.core.OptionHandler*. Construye un clasificador basado en reglas fuzzy usando *ENORA* o *NSGA-II* con las opciones siguientes, entre otras (tenga en cuenta que no hay opciones para establecer las probabilidades de cruce y mutación ya que son adaptativas):

- *Algorithm*, para establecer el algoritmo (*ENORA* o *NSGA-II*);
- *EvaluationMeasure*, para establecer la medida en la que se evaluara el desempeño del clasificador;
- *PopulationSize*, para establecer el número de individuos de la población;

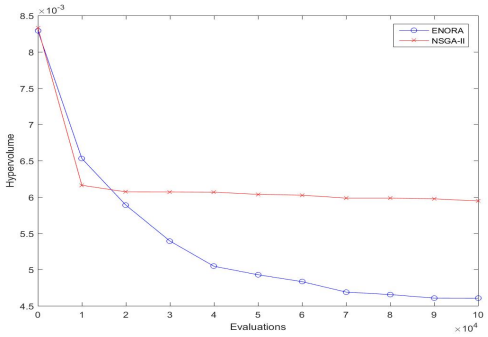
- *Generations*, para estableces el número de generaciones en las que evolucionará la población;
- *MaxRules*, para establecer el valor del número máximo de reglas;
- *MaxLabels*, para establecer el número de etiquetas lingüísticas para los atributos de entrada reales;
- *MaxSimilarity*, para establecer el valor máximo de similaridad de los conjuntos fuzzy;
- *MinV*, para establecer el valor para el cual el dominio de una variable se divide para obtener la varianza mínima;
- *MaxV*, para establecer el valor para el cual el dominio de una variable se divide para obtener la varianza máxima;

Los algoritmos *ENORA* y *NSGA-II* han sido implementados con *representación de longitud variable* con coma flotante y variables de entrada categóricas con enfoque de *Pittsburgh*, *inicialización aleatoria uniforme*, *selección por torneo binario*, *manejo de restricciones* usando un *algoritmo de reparo*, ranking basado en el *nivel de no-dominación* con *crowding distance*, *operadores de variación adaptativos* que trabajan en diferentes niveles del clasificador fuzzy: *cruce de conjuntos fuzzy*, *cruce de reglas*, *cruce de reglas incremental*, *mutación del centro del conjunto gaussiano*, *mutación de la varianza del conjunto gaussiano*, *mutación del conjunto fuzzy*, *mutación incremental de reglas*, y *mutación entera* (para datos categóricos). Una vez extraído el conjunto de reglas fuzzy, se asigna una etiqueta lingüística a cada conjunto fuzzy.

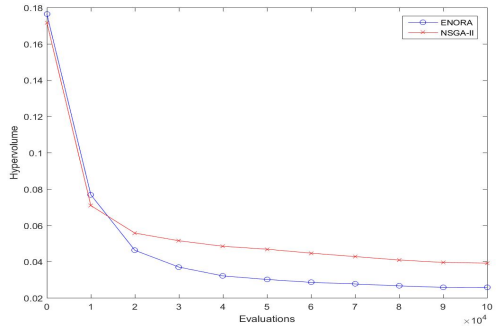
A.4. Una metodología para evaluar métodos evolutivos multi-objetivo de selección de atributos para tareas de clasificación en el contexto del screening virtual



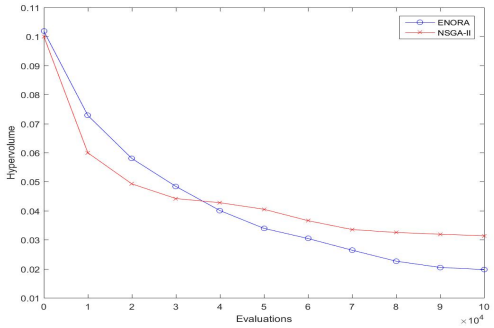
tk data set
Modelo de optimización eq. (4)



mr data set
Modelo de optimización eq. (4)

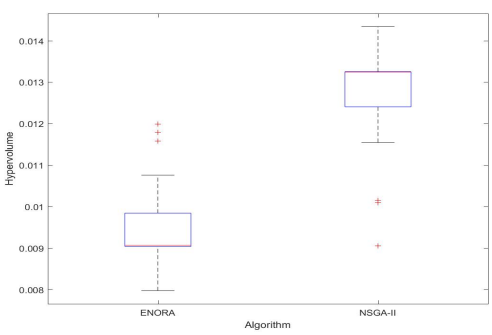


tk data set
Modelo de optimización eq. (5)

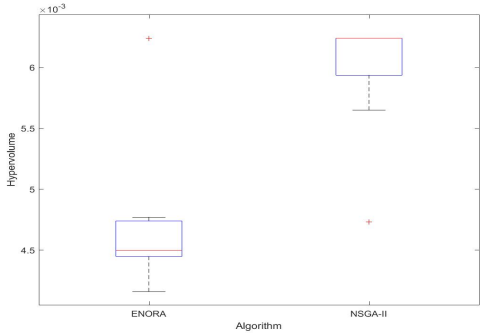


mr data set
Modelo de optimización eq. (5)

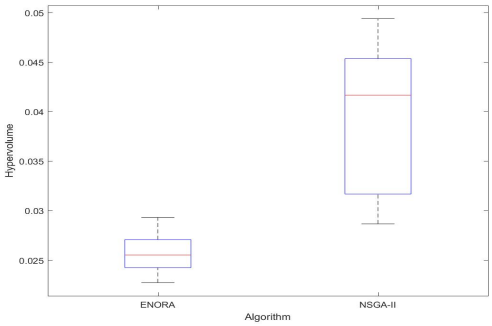
Figura 16: Evolución del hipervolumen medio obtenido con 30 ejecuciones de las estrategias de búsqueda *ENORA* y *NSGA-II* para las bases de datos *tk* y *mr*.



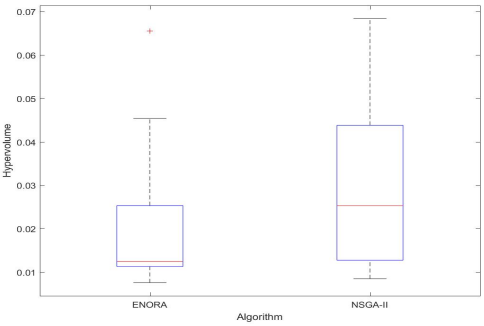
tk data set
Modelo de optimización eq. (4)



mr data set
Modelo de optimización eq. (4)



tk data set
Modelo de optimización eq. (5)



mr data set
Modelo de optimización eq. (5)

Figura 17: Diagramas de caja para el hipervolumen obtenido con 30 ejecuciones de los algotitmos *ENORA* y *NSGA-II* para las bases de datos *tk* y *mr*.

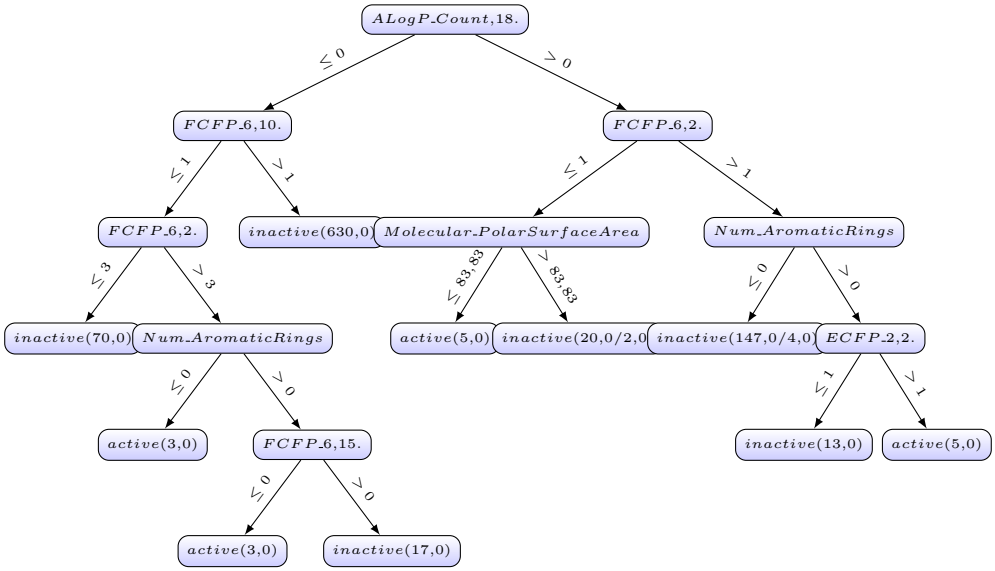


Figura 18: Árbol de decisión de *ENORA-C4.5-ACC* para la base de datos *tk*.

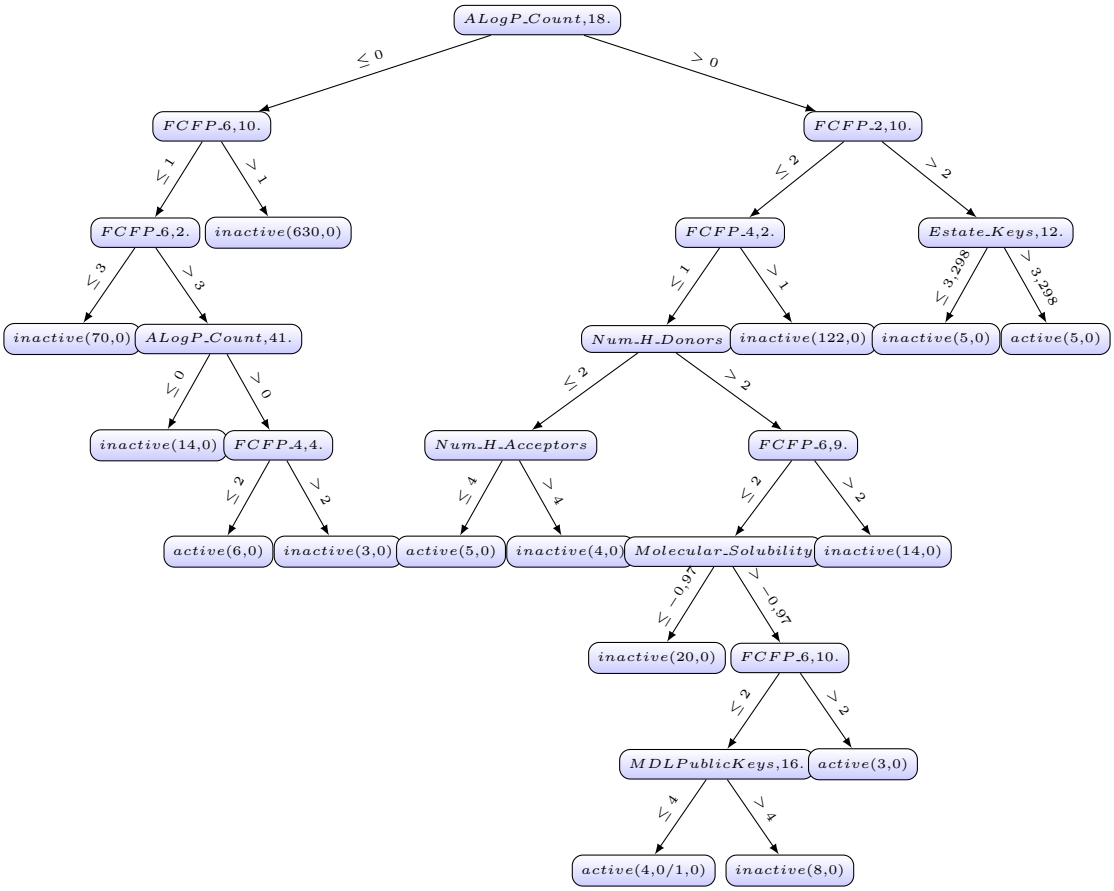


Figura 19: Árbol de decisión de *ENORA-C4.5-AUC* para la base de datos *tk*.

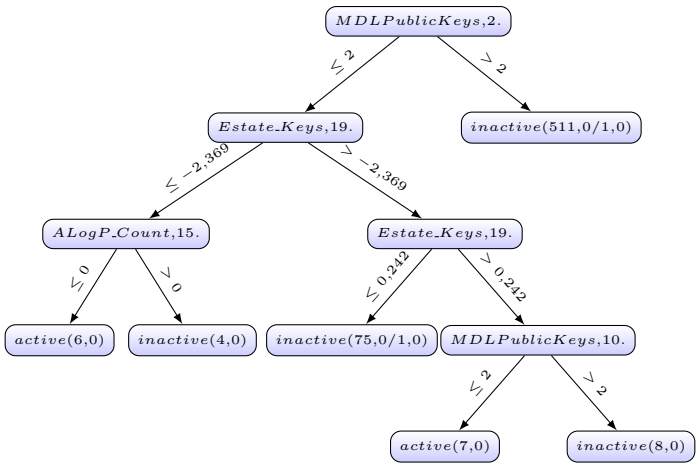


Figura 20: Árbol de decisión de *ENORA-C4.5-ACC* para la base de datos *mr*.

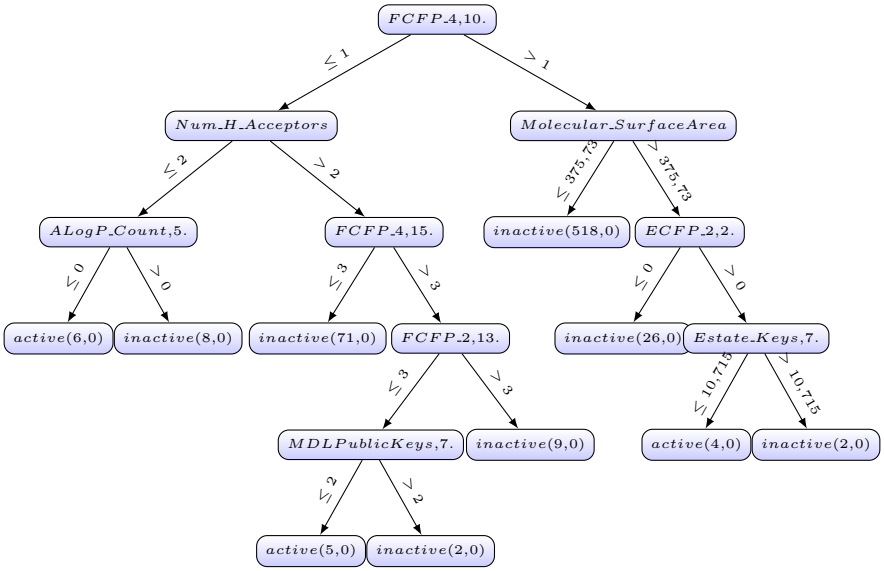


Figura 21: Árbol de decisión de *ENORA-C4.5-AUC* para la base de datos *mr*.

A.5. Clasificación evolutiva multi-objetivo basada en reglas con datos categóricos

Algoritmo 5 Población inicial para clasificación basada en reglas con datos categóricos

Entrada: $p > 0$ {Número de atributos de entrada categóricos}
Entrada: $v_1, \ldots, v_p$, $v_j > 1$, $j = 1, \ldots, p$ {Número de categorías para los atributos de entrada}
Entrada: $w > 1$, {Número de clases para el atributo de salida}
Entrada: $\delta > 0$ {Número de operadores de cruce}
Entrada: $\epsilon > 0$ {Número de operadores de mutación}
Entrada: $M_{max} \geq w$ {Número máximo de reglas}
Entrada: $N > 1$ {Número de individuos en la población}
$P \leftarrow \emptyset$
para $k = 1$ a N **hacer**
  $I \leftarrow$ nuevo individuo
  si $k \leq M_{max} - w + 1$ **entonces**
    $M_I \leftarrow k + w - 1$
  si no
    $M_I \leftarrow$ Int $aleatorio(w, M_{max})$
  fin si
  {Regla aleatoria R_i^I}
  para $i = 1$ a M_I **hacer**
    {Valores enteros aleatorios asociados a los antecedentes}
    para $j = 1$ a p **hacer**
      $b_{ij}^I \leftarrow Random(1, v_j)$
    fin para
    {Valor entero aleatorio asociado al consecuente}
    si $i < w$ **entonces**
      $c_i^I = j$
    si no
      $c_i^I \leftarrow Random(1, w)$
    fin si
  fin para
  {Valores enteros aleatorios para la variación adaptativa}
  $d_I \leftarrow Random(0, \delta)$
  $e_I \leftarrow Random(0, \epsilon)$
  $P \leftarrow P \cup I$
fin para
devolver P

Algoritmo 6 Variación en clasificación basada en reglas con datos categóricos

Entrada: $Padre1$, $Padre2$ {Individuos a cambiar}
$Hijo1 \leftarrow Padre1$
$Hijo1 \leftarrow Padre2$
Cruce adaptativo $Hijo1$, $Hijo2$
Mutación adaptativa $Hijo1$
Mutación adaptativa $Hijo2$
devolver $Hijo1, Hijo2$

Algoritmo 7 Cruce adaptativo en clasificación basada en reglas con datos categóricos

Entrada: I, J {Individuos a cruzar}
Entrada: p_v $(0 < p_v < 1)$ {Probabilidad de cruce}
Entrada: $\delta > 0$ {Número de operadores de cruce diferentes}
si Una variable aleatoria de Bernoulli p_v toma el valor 1 **entonces**
  $d_I \leftarrow Random(0, \delta)$
fin si
$d_J \leftarrow d_I$
Realiza el cruce especificado por d_I:
  {0: No hay cruce}
  {1: Cruce de reglas}
  {2: Cruce de reglas incremental}

Algoritmo 8 Mutación adaptativa en clasificación basada en reglas con datos categóricos

Entrada: I {Individuo a mutar}
Entrada: p_v $(0 < p_v < 1)$ {Probabilidad de mutación}
Entrada: $\epsilon > 0$ {Número de operadores de mutación diferentes}
si Una variable aleatoria de Bernoulli p_v toma el valor 1 **entonces**
  $e_I \leftarrow Random(0, \epsilon)$
fin si
Realiza el mutación especificado por e_I:
  {0: No hay mutación}
  {1: Mutación incremental de reglas}
  {2: Mutación}

Algoritmo 9 Cruce de reglas en clasificación basada en reglas con datos categóricos

Entrada: I, J {Individuos a cruzar}
1: $i \leftarrow Random(1,M_I)$
2: $j \leftarrow Random(1,M_J)$
3: Intercambia las reglas R_i^I y R_j^J

Algoritmo 10 Cruce de reglas incremental en clasificación basada en reglas con datos categóricos

Entrada: I, J {Individuos a cruzar}
Entrada: M_{max} {Máximo número de reglas}
1: **si** $M_I < M_{max}$ **entonces**
2: $j \leftarrow Random(1,M_J)$
3: Añade R_j^J al individuo I
4: **fin si**
5: **si** $M_J < M_{max}$ **entonces**
6: $i \leftarrow Random(1,M_I)$
7: Añade R_i^I al individuo J
8: **fin si**

Algoritmo 11 Mutación de reglas incremental en clasificación basada en reglas con datos categóricos

Entrada: I {Individuo a mutar}
Entrada: M_{max} {Máximo número de reglas}
1: **si** $M_I < M_{max}$ **entonces**
2: Agrega una nueva regla aleatoria a I
3: **fin si**

Algoritmo 12 Mutación entera en clasificación basada en reglas con datos categóricos

Entrada: I {Individuo a mutar}
Entrada: $p > 0$ {Número de atributos de entrada categóricos}
Entrada: $v_1, \ldots, v_p$, $v_j > 1$, $j = 1, \ldots, p$ {Número de categorías para los atributos de entrada}
1: $i \leftarrow Random(1,M_I)$
2: $j \leftarrow Random(1,p)$
3: $b_{ij}^I \leftarrow Random(1,v_j)$

A.6. Selección de atributos evolutiva multi-objetivo para clasificación fuzzy

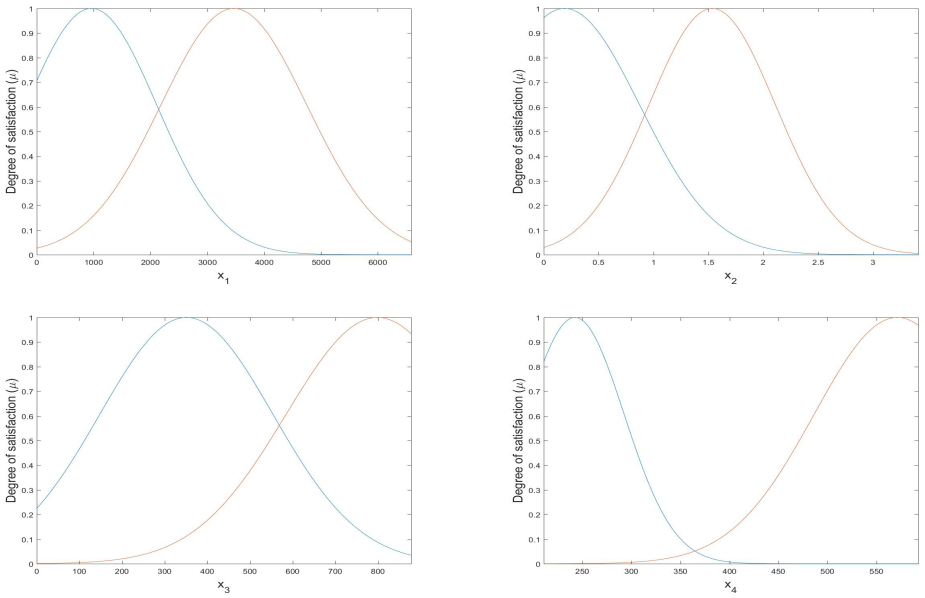


Figura 22: Conjuntos fuzzy gaussianos para *INBOUND_AGENTS*.

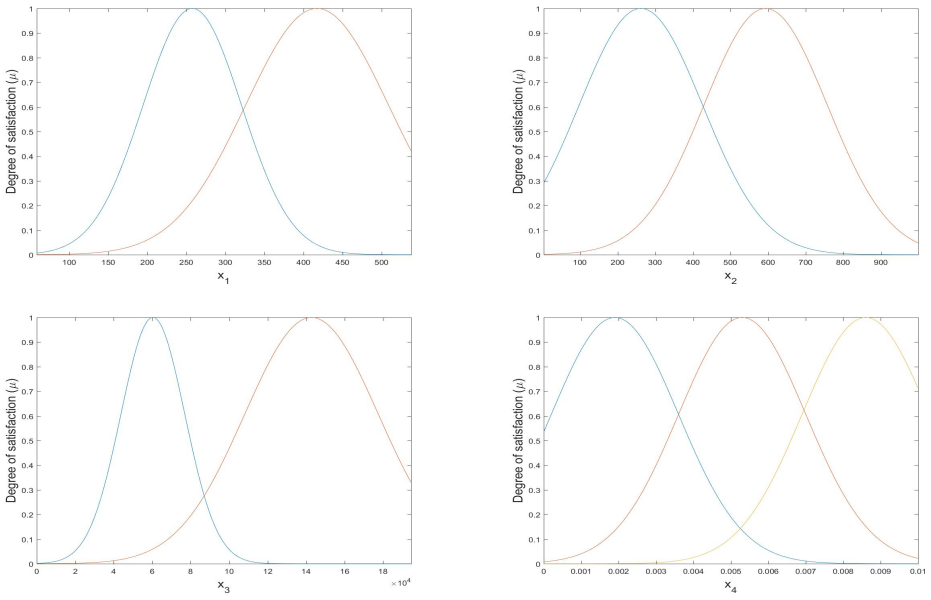


Figura 23: Conjuntos fuzzy gaussianos para *ALL_AGENTS*.

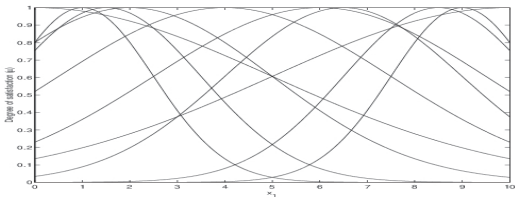
Simbolo	Definición
	Ecuación 1: Optimización multi-objetivo con restricciones
x_k	k-ésima variable de decisión
$\mathbf{x}$	Conjunto de variables de decisión
$f_i(\mathbf{x})$	i-ésima función objetivo
$g_j(\mathbf{x})$	j-ésima restricción
$l > 0$	Número de objetivos
$m > 0$	Número de restricciones
$w > 0$	Número de variables de decisión
$\mathcal{X}$	Dominio para cada variable de decisión x_k
$\mathcal{X}^w$	Dominio para el conjunto de variables de decisión
$\mathcal{F}$	Conjunto de todas las soluciones viables
$\mathcal{S}$	Soluciones no dominadas o conjunto óptimo de Pareto
$\mathcal{D}(\mathbf{x}', \mathbf{x})$	Función de dominación de Pareto
	Ecuación 2: Problema de optimización combinatoria booleano multiobjetivo para selección de atributos
$\mathcal{D}$	Base de datos
x_k	k-ésima variable de decisión: verdadera si k esta seleccionado
$\mathbf{x}$	Conjunto de variables de decisión
$\mathcal{F}_{\mathcal{D}}(\mathbf{x})$	Función objetivo de rendimiento de un conjunto de atributos $\mathbf{x}$ en la base de datos $\mathcal{D}$
$\mathcal{C}_{\mathcal{D}}(\mathbf{x})$	Número de atributos seleccionados en $\mathbf{x}$
$\eta(x_k)$	Función que transforma un valor booleano en numérico ($\eta(true) = 1$ y $\eta(false) = 0$)
$N > 0$	Número de variables de decisión (número de atributos en la base de datos $\mathcal{D}$)
	Ecuación 3: Problema de optimización multi-objetivo para la Clasificación Fuzzy con restricciones
$\mathcal{D}$	Base de datos
x_i	i-ésimo atributo de entrada real en la base de datos $\mathcal{D}$
$\mathbf{x}$	Atributos reales de entrada en la base de datos $\mathcal{D}$
y_i	i-ésimo atributo de entrada categórico en la base de datos $\mathcal{D}$
$\mathbf{y}$	Atributos de entrada categóricos en la base de datos $\mathcal{D}$
z	Atributo de salida categórico en la base de datos $\mathcal{D}$
$[l_i, u_i]$	Dominio del i-ésimo atributo de entrada real en la base de datos $\mathcal{D}$
$\{1, \ldots, v_i\}$	Dominio del i-ésimo atributo de entrada categórico en la base de datos $\mathcal{D}$
$\{1, \ldots, w\}$	Dominio del atributo de salida categórico en la base de datos $\mathcal{D}$
$p \geq 0$	Número de atributos de entrada reales en la base de datos $\mathcal{D}$
$q \geq 0$	Número de atributos de entrada categóricos en la base de datos $\mathcal{D}$
$\mathbf{\Gamma}$	Clasificador basado en reglas fuzzy
$R_j^{\mathbf{\Gamma}}$	j-ésimo clasificador basado en reglas fuzzy $\mathbf{\Gamma}$
A_{ij}^{Γ}	Conjunto fuzzy para el i-ésimo atributo de entrada real y el j-esimo clasificador basado en reglas fuzzy $\mathbf{\Gamma}$
B_{ij}^{Γ}	Categoría para el i-ésimo atributo de entrada categórico y la j-ésima regla del clasificador fuzzy $\mathbf{\Gamma}$
C_j^{Γ}	Categoría para el atributo de salida categórico y la j-ésima regla del clasificador fuzzy $\mathbf{\Gamma}$
$\varphi_j^{\Gamma}(\mathbf{x}, \mathbf{y})$	Grado de compatibilidad de la regla R_j^{Γ} para la muestra $(\mathbf{x}, \mathbf{y})$
$\phi_j^{\Gamma}(\mathbf{y})$	Número de atributos de entrada categóricos, por lo que $y_i = B_{ij}^{\Gamma}$
$\lambda_C^{\Gamma}(\mathbf{x}, \mathbf{y})$	Grado asociado del la muestra $(\mathbf{x}, \mathbf{y})$ con la clase C
$f_{\Gamma}(\mathbf{x}, \mathbf{y})$	Clasificación o salida del clasificador Γ para la muestra $(\mathbf{x}, \mathbf{y})$
$\mathcal{F}_{\mathcal{D}}(\mathbf{\Gamma})$	Función objetivo de rendimiento del clasificador fuzzy $\mathbf{\Gamma}$ en la base de datos $\mathcal{D}$
$\mathcal{NR}(\mathbf{\Gamma})$	Número de reglas fuzzy del clasificador fuzzy $\mathbf{\Gamma}$
$\mathcal{NL}(\mathbf{\Gamma})$	Número de etiquetas lingüísticas del clasificador fuzzy $\mathbf{\Gamma}$
$\mathcal{S}(\mathbf{\Gamma})$	Similaridad del clasificador fuzzy $\mathbf{\Gamma}$
M_{min}	Número mínimo de reglas fuzzy permitidas para los clasificadores fuzzy
M_{max}	Número máximo de reglas fuzzy permitidas para los clasificadores fuzzy
L_{max}	Número máximo de etiquetas lingüísticas permitidas para los clasificadores fuzzy
$g_s \in]0, 1[$	Máxima similaridad permitida para los clasificadores fuzzy
	Problema de optimización combinatoria multi-objetivo para elegir la mejor base de datos
$\mathbf{x} \in DB$	Decision variable
$DB = \{1, \ldots, 79\}$	Conjunto de bases de datos
$n = 3$	Número de clasificadores (*J48*, *RandomForest* y *LibSVM*)
$ACC(\mathbf{x}, j)$	Accuracy del clasificador j para la base de datos $\mathbf{x}$
$WAUC(\mathbf{x}, j)$	Área ponderada bajo la curva ROC del clasificador j para la base de datos $\mathbf{x}$
$RMSE(\mathbf{x}, j)$	Error cuadrático medio del clasificador j para la base de datos $\mathbf{x}$
$MS(\mathbf{x}, j)$	Tamaño del modelo del clasificador j para la base de datos $\mathbf{x}$

Tabla 11: Tabla de nomenclatura.

Nombre	Descripción
weka.attributeSelection	Paquete para la selección de atributos
weka.attributeSelection.AttributeSelection	Clase para la selección de atributos
weka.attributeSelection.ASSearch	Clase abstracta para estrategia de búsqueda
weka.attributeSelection.ASEvaluation	Clase abstracta para evaluación
weka.classifiers.AbstractClassifier	Clasificador Abstracto
weka.classifiers.SingleClassifierEnhancer	Clase de utilidad abstracta, hereda de *AbstractClassifier*
weka.classifiers.meta.AttributeSelectedClassifier	Meta-clasificador para selección de atributos + clasificación / regresión, hereda de *SingleClassifierEnhancer*
weka.attributeSelection.BestFirst	Clase para la estrategia de búsqueda best first, hereda de *ASSearch*
weka.attributeSelection.GreedyStepwise	Clase para la estrategia de búsqueda greedy stepwise, hereda de *ASSearch*
weka.attributeSelection.LinearForwardSelection	Clase para la estrategia de búsqueda linear forward selection, hereda de *ASSearch*
weka.attributeSelection.MultiObjectiveEvolutionarySearch	Clase para la estrategia de búsqueda multi-objective evolutionary search, hereda de *ASSearch*
weka.attributeSelection.PSOSearch	Clase para la estrategia de búsqueda particle swarm optimization, hereda de *ASSearch*
weka.attributeSelection.GeneticSearch	Clase para la estrategia de búsqueda genetic search, extends *ASSearch*
weka.attributeSelection.Ranker	Clase para clasificar los atributos en métodos de selección de atributos univariate, hereda de *ASSearch*
weka.attributeSelection.WrapperSubsetEval	Clase para clasificar los atributos en métodos de selección de atributos wrapper multivariate, hereda de *ASEvaluation*
weka.attributeSelection.ClassifierAttributeEval	Clase para métodos de selección de atributos wrapper univariate, hereda de *ASEvaluation*
weka.attributeSelection.CfsSubsetEval	Clase para los métodos de selección de atributos filter multivariate, hereda de *ASEvaluation*
weka.attributeSelection.ConsistencySubsetEval	Clase para los métodos de selección de atributos filter multivariate, hereda de *ASEvaluation*
weka.attributeSelection.GainRatioAttributeEval	Clase para los métodos de selección de atributos filter univariate, hereda de *ASEvaluation*
weka.attributeSelection.SymmetricalUncertAttributeEval	Clase para los métodos de selección de atributos filter univariate, hereda de *ASEvaluation*
weka.classifiers.trees.J48	Clase para generar árbol de decisión C4.5 podado o sin podar, hereda de *AbstractClassifier*
weka.classifiers.functions.LibSVM	Clase wrapper para las herramientas libsvm, hereda de *weka.classifiers.RandomizableClassifier*
weka.classifiers.trees.RandomForest	Clase para la construcción de un bosque de árboles aleatorios, hereda de *weka.classifiers.meta.Bagging*
weka.classifiers.rules.MultiObjectiveEvolutionaryFuzzyClassifier	Clase para construir un clasificador basado en reglas fuzzy, hereda de *AbstractClassifier*

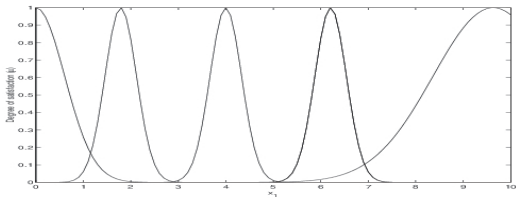
Tabla 12: Paquetes y clases para selección de atributos en *Weka* utilizados en este documento.

	x_1	y
$R_1:$	$A_{11} = (0{,}0; 5{,}0)$	1
$R_2:$	$A_{12} = (10{,}0; 5{,}0)$	1
$R_3:$	$A_{13} = (6{,}0; 3{,}5)$	2
$R_4:$	$A_{14} = (2{,}0; 3{,}0)$	1
$R_5:$	$A_{15} = (1{,}0; 1{,}5)$	1
$R_6:$	$A_{16} = (9{,}0; 1{,}5)$	2
$R_7:$	$A_{17} = (6{,}5; 2{,}5)$	2
$R_8:$	$A_{18} = (4{,}0; 3{,}5)$	1
$R_9:$	$A_{19} = (8{,}5; 2{,}0)$	2
$R_{10}:$	$A_{110} = (1{,}5; 2{,}0)$	1



Clasificador inicial:
10 reglas. 10 conjuntos fuzzy. $\mathcal{S}(\mathbf{\Gamma}) = 0{,}997$.

	x_1	y
$R'_1:$	$A'_{11} = (0{,}0; 0{,}6)$	1
$R'_2:$	$A'_{12} = (9{,}6; 1{,}3)$	1
$R'_3:$	$A'_{13} = (1{,}8; 0{,}3)$	1
$R'_4:$	$A'_{14} = (6{,}2; 0{,}3)$	1
$R'_5:$	$A'_{15} = (4{,}0; 0{,}3)$	1
$R'_6:$	$A'_{16} = (6{,}2; 0{,}3)$	2



Clasificador después de algoritmo de reparo:
6 reglas. 5 conjuntos fuzzy. $\mathcal{S}(\mathbf{\Gamma}) = 0{,}074$.

Tabla 13: Ejemplo de algoritmo de reparo para un problema ficticio con una sola variable de entrada real con dominio en $[0, 10]$ y una sola variable de entrada binaria. El algoritmo de reparo se ejecuta con un umbral de similaridad de $g_s = 0{,}1$ y $L_{max} = 7$. Se repara un clasificador inicial no factible con 10 reglas, 10 conjuntos fuzzy y $\mathcal{S} = 0{,}997 > g_s$. Durante la reparación, varios conjuntos fuzzy se separan y se fusionan y se eliminan varias reglas. Después de ejecutar el algoritmo de reparo, se obtiene un clasificador factible con 6 reglas, 5 conjuntos fuzzy y $\mathcal{S} = 0{,}074 \leq g_s$.

B. Publicaciones que componen la tesis doctoral

B.1. A methodology for evaluating multi-objective evolutionary feature selection for classification in the context of virtual screening

Resumen:

Se ha demostrado que los métodos de *Virtual Screening* (VS) aumentan la tasa de éxito en muchas campañas de descubrimiento de fármacos, cuando se complementan con enfoques experimentales, como son los métodos de cribado de alto rendimiento o los enfoques de la química medicinal clásica. Sin embargo, la capacidad predictiva de VS aún no es óptima, principalmente debido a las limitaciones en los principios físicos subyacentes que describen los fenómenos de *drug binding*. Un enfoque que puede mejorar los métodos de VS es la ayuda de los métodos de aprendizaje automático. Cuando hay suficientes datos experimentales disponibles para entrenar tales métodos, la capacidad predictiva puede aumentar considerablemente. En este trabajo de investigación mostramos cómo una estrategia de búsqueda evolutiva multi-objetivo para selección de atributos, que puede proporcionar árboles de decisión pequeños y precisos, fáciles de comprender para los químicos, puede aumentar drásticamente la aplicabilidad y la capacidad predictiva de estas técnicas y, por lo tanto, ser una ayuda considerable en el problema de descubrimiento de fármacos. Con la metodología propuesta, encontramos modelos de clasificación con una precisión entre 0.9934 y 1.00 y un área bajo la curva ROC entre 0.96 y 1.00 evaluados en modo *full trainning*, y una precisión entre 0.9849 y 0.9940 y un área bajo la curva ROC entre 0.89 y 0.93 evaluados con 10-fold cross-validation sobre 30 iteraciones, mientras que reduce sustancialmente el tamaño del modelo.

Título	A methodology for evaluating multi-objective evolutionary for classification in the context of virtual screening
Autores	Fernando Jiménez, Horacio Pérez-Sánchez, José Palma, Gracia Sánchez y Carlos Martínez
Tipo	Revista
Revista Factor de impacto JCR 2017 Rank JCR 2017	Soft Computing 2.367 Computer Science, Artificial Intelligence: 45/132 (Q2) Computer Science, Interdisciplinary Applications: 40/105 (Q2)
Editor	Springer
Fecha ISNN	1 de Septiembre de 2018 1433-7479
DOI	https://doi.org/10.1007/s00500-018-3479-0
Estado	Publicado
Aportación	Software, validación, investigación y visualización

B.2. Multi-Objective Evolutionary Rule-Based Classification with Categorical Data

Resumen:
La interpretabilidad de un modelo de clasificación es esencial para su validación. A veces es necesario explicar claramente el proceso de clasificación de las predicciones de un modelo. Los modelos que son intrínsecamente más fáciles de interpretar pueden relacionarse sin esfuerzo con el contexto del problema, y sus predicciones pueden, si es necesario, ser evaluadas ética y legalmente. En este documento, proponemos un método novel para generar clasificadores basados en reglas a partir de datos categóricos que se pueden interpretar fácilmente. Los clasificadores se generan utilizando un enfoque de optimización multi-objetivo que se centra en dos objetivos principales: maximizar el rendimiento del clasificador aprendido y minimizar su número de reglas. Los algoritmos evolutivos multi-objetivo *ENORA* y *NSGA-II* se han adaptado para optimizar el rendimiento del clasificador basándose en tres métricas de aprendizaje automático diferentes: *accuracy*, *área bajo la curva ROC* y *raíz cuadrada del error cuadrático medio*. Se han comparado ampliamente los clasificadores generados utilizando nuestro método propuesto con clasificadores generados utilizando métodos clásicos tales como *PART*, *JRip*, *OneR* y *ZeroR*. Los experimentos se han realizado en modo *full trainning*, en *10-fold cross-validation* y dividiendo los datos para entrenamiento y prueba. Para hacer que los resultados sean reproducibles, hemos utilizado los conjuntos de datos conocidos y disponibles públicamente *Breast Cancer*, *Monk's Problem 2*, *Tic-Tac-Toe-Endgame*, *Car*, *kr-vs-kp* y *Nursery*. Después de realizar tests estadísticos sobre nuestros resultados, llegamos a la conclusión de que el método propuesto es capaz de generar modelos de clasificación altamente precisos y fáciles de interpretar.

Título	Multi-Objective Evolutionary Rule-Based Classification with Categorical Data
Autores	Fernando Jiménez, Carlos Martínez, Luis Miralles-Pechuán, Gracia Sánchez y Guido Sciavicco
Tipo	Revista
Revista Factor de impacto JCR 2017 Rank JCR 2017	Entropy 2.305 Physics, Multidisciplinary: 22/78 (Q2)
Editor	MDPI
Fecha ISNN	7 de Septiembre de 2018 1099-4300
DOI	https://doi.org/10.3390/e20090684
Estado	Publicado
Aportación	Software y validación

B.3. Multi-objective Evolutionary Feature Selection for Fuzzy Classification

Resumen:

La interpretabilidad de los sistemas de clasificación se refiere a la capacidad de estos para expresar su comportamiento de una manera que sea fácilmente comprensible para un usuario. Los modelos de clasificación interpretables permiten la validación externa por parte de un experto y, en ciertas disciplinas como la medicina o los negocios, proporcionar información sobre la toma de decisiones es esencial por razones éticas y humanas. Los sistemas de clasificación basados en reglas fuzzy son potentes herramientas de clasificación consolidadas basadas en la lógica fuzzy y diseñadas para producir modelos interpretables. Sin embargo, en presencia de una gran cantidad de atributos, incluso los modelos basados en reglas tienden a ser demasiado complejos para ser interpretados fácilmente. En este trabajo, proponemos un nuevo método de selección de atributos multivariate en el que tanto la estrategia de búsqueda como el evaluador se basan en computación evolutiva multi-objetivo. Diseñamos un conjunto de experimentos para establecer una configuración aceptable del número de evaluaciones requeridas por la estrategia de búsqueda y por el clasificador, y probamos nuestra estrategia en un conjunto de datos de la vida real. Hemos comparado nuestros resultados con una amplia gama de métodos de selección de atributos que incluyen métodos *filter*, *wrapper*, *multivariate* y *univariate*, con estrategias de búsqueda deterministas y probabilistas, y con evaluadores de diversa naturaleza. Finalmente, el modelo de clasificación basado en reglas fuzzy obtenido con el método propuesto se ha evaluado con métricas de rendimiento estándar y se ha comparado con otros clasificadores basados en reglas fuzzy conocidos. Hemos utilizado dos conjuntos de datos de la vida real extraídos de un centro de contacto; en un caso, con el método propuesto obtuvimos una precisión de 0.7857 con 8 reglas y el mejor clasificador fuzzy comparado obtuvo 0.7679 con 8 reglas, y en el segundo caso obtuvimos una precisión de 0.7403 con 5 reglas, mientras que el mejor clasificador fuzzy comparado obtuvo 0.6364 con 4 reglas.

Título	Multi-objective Evolutionary Feature Selection for Fuzzy Classification
Autores	Fernando Jiménez, Carlos Martínez, Enrico Marzano, José Palma, Gracia Sánchez y Guido Sciavicco
Tipo	Revista
Revista Factor de impacto JCR 2017 Rank JCR 2017	IEEE Transactions on Fuzzy Systems 8.415 Computer Science, Artificial Intelligence: 4/132 (Q1) Engineering, Electrical & Electronic: 7/260 (Q1)
Editor	IEEE
Fecha ISNN	10 de Enero de 2019 1941-0034
DOI	https://doi.org/10.1109/TFUZZ.2019.2892363
Estado	Publicado
Aportación	Metodología, software, validación, investigación y visualización