

UNIVERSIDAD DE GRANADA



REDES NEURONALES EVOLUTIVAS MULTIOBJETIVO
PARA CLASIFICACIÓN NOMINAL Y ORDINAL.
APLICACIONES

*MULTIOBJECTIVE EVOLUTIONARY NEURAL NETWORKS
FOR NOMINAL AND ORDINAL CLASSIFICATION.
APPLICATIONS*

Doctorado Internacional
Internatinal Doctorate

Doctorando:

MANUEL CRUZ RAMÍREZ

Director:

CÉSAR HERVÁS MARTÍNEZ

Para optar al grado de:

DOCTOR EN INFORMÁTICA

Septiembre 2013

Editor: Editorial de la Universidad de Granada
Autor: Manuel Cruz Ramírez
D.L.: GR 113-2014
ISBN: 978-84-9028-701-9

Manuel Cruz Ramírez: Redes Neuronales Evolutivas Multiobjetivo para Clasificación Nominal y Ordinal. Aplicaciones, *Multiobjective Evolutionary Neural Networks for Nominal and Ordinal Classification. Applications*, Doctor en Informática, Septiembre 2013
DIRECTOR: César Hervás Martínez



La memoria titulada “Redes Neuronales Evolutivas Multiobjetivo para Clasificación Nominal y Ordinal. Aplicaciones”, que presenta D. Manuel Cruz Ramírez para optar al grado de Doctor, ha sido realizada dentro del “Programa Oficial de Doctorado en Tecnología de la Información y la Comunicación” de la UNIVERSIDAD DE GRANADA bajo la dirección del Catedrático de Universidad D. César Hervás Martínez.

El doctorando D. Manuel Cruz Ramírez y el director de la tesis D. César Hervás Martínez garantizamos, al firmar esta Tesis Doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección del director de la Tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, Septiembre de 2013

El Doctorando:

Fdo: Manuel Cruz Ramírez

El Director:

Fdo: César Hervás Martínez



MENCIÓN DE DOCTORADO INTERNACIONAL

Esta Tesis cumple los criterios para la obtención de la “Mención de Doctorado Intenacional” concedida por la Universidad de Granada. Para ello se presentan los siguientes requisitos:

1. Estancia predoctoral de 3 meses realizada fuera de España:
 - Realizada en *School of Computer Science, the University of Birmingham, Birmingham, United Kingdom*, durante los meses de septiembre y diciembre del año 2011, siendo el supervisor de la misma el Dr Huanhuan Chen.
2. Esta Tesis está avalada por dos doctores pertenecientes a centros de investigación no españoles.
 - **Dr. Xin Yao.** *Full Professor* en “School of Computer Science” en la Universidad de Birmingham (Reino Unido) y director del Centro de Excelencia para la Investigación en Inteligencia Computacional y aplicaciones (*Centre of Excellence for Research in Computational Intelligence and Applications, (CER-CIA)*).
 - **Dr. David A. Elizondo.** *Reader in Computer Science* en el departamento de “Computing and Informatics” de la Universidad de De Montfort (Reino Unido).
3. La defensa de Tesis y el texto se han realizado parcialmente en dos idiomas, español e inglés.
4. Entre los miembros del tribunal se encuentra un doctor procedente de un centro de educación superior, tratándose del Dr. Peter Tiño, *Senior Lecturer* en “School of Computer Science” en la Universidad de Birmingham (Reino Unido).

Granada, Septiembre de 2013

El Doctorando:

Fdo: Manuel Cruz Ramírez

Esta Tesis Doctoral ha sido subvencionada parcialmente con cargo a los Proyecto **TIN2008-06681-Co6-03** y **TIN2011-22794** de la Comisión Interministerial de Ciencia y Tecnología (CICYT) y con fondos FEDER.

También ha sido subvencionada parcialmente por los Proyectos de Excelencia **P08-TIC-3745** y **P2011-TIC-7508** de la Junta de Andalucía.

La investigación de Manuel Cruz Ramírez ha sido subvencionada desde diciembre del 2010 por el programa predoctoral de Formación del Profesorado Universitario (FPU, referencia de la beca **AP2009-0487**) del Ministerio de Educación, Cultura y Deporte. Antes de diciembre de 2010, Manuel Cruz Ramírez fue beneficiario de una beca concedida por ASTELAS PHARMAS S.A.



RESUMEN

Este trabajo de Tesis aborda la obtención de clasificadores nominales mediante diferentes modelos de redes neuronales. Para ello, se analizan diferentes métricas asociadas a distintas facetas contrapuestas de la “bondad” de un clasificador multiclase. Esta contraposición nos hace obtener modelos que consideran diferentes características del problema.

La forma clásica de trabajar con los problemas con dos o más objetivos en conflicto ha sido la agregación o ponderación de estos objetivos, de manera que los problemas multiobjetivo eran transformados en problemas con un único objetivo. Una vez realizada esta transformación, los problemas podían ser resueltos mediante cualquier algoritmo, obteniendo finalmente una solución que satisface los objetivos del problema.

Desde la aparición de los algoritmos evolutivos (algoritmos basados en la teoría de la evolución de las especies) y del concepto de óptimo de Pareto, los problemas con múltiples objetivos han sido resueltos por lo que se conoce como algoritmos evolutivos multiobjetivo. Estos algoritmos entrenan simultáneamente un conjunto de modelos durante un número de iteraciones, ordenando los modelos obtenidos tras cada iteración según la relación de dominancia existentes entre ellos. Cuando se habla de dominancia de Pareto, se entiende que un modelo domina a otro si es mejor o igual en todos los objetivos y estrictamente mejor en uno de ellos. Por tanto, la salida obtenida tras finalizar la ejecución del algoritmo multiobjetivo consistirá en un conjunto de modelos que no presentan ninguna relación de dominancia entre ellos, conocidos como primer frente de Pareto. Cualquiera de estos modelos es una solución óptima al problema, aunque no existe forma de determinar cuál de ellas es la mejor.

Para ayudar a la hora de elegir uno de estos modelos, en la presente Tesis se estudian diferentes técnicas de selección de individuos, como la selección de los modelos extremos del frente óptimo de Pareto o la elección del modelo que minimice la distancia euclídea entre su posición en el frente y la solución ideal del problema. Este estudio concluye que la elección de una de estas técnicas depende de las características del problema analizado.

Por otra parte, se ha desarrollado sensibles variaciones de algoritmo evolutivo multiobjetivo PDE desarrollado por H. Abbass. Estas variaciones se centran tanto en evitar una convergencia prematura como en la aplicación del algoritmo de búsqueda local, con el objetivo de mejorar el rendimiento sin aumentar el coste computacional. Junto a estos algoritmos, en la experimentación se han empleado el clásico algoritmo NSGAI y un algoritmo memético desarrollado a partir de este, MPENSGAI.

Una de las razones fundamentales de la construcción de estos clasificadores multiclase ha sido poder aplicarlos a diferentes problemas reales, dando sentido a este documento de Tesis. Así, los algoritmos evolutivos multiobjetivo han sido aplicados en:

- Medicina: Selección del paciente más idóneo en lista de espera que deberá recibir un órgano disponible en los trasplantes hepáticos. Para ello, se ha dispuesto de una base de datos formada por 1003 pares donante/receptor con la que se han obtenido dos modelos de clasificación basados en la supervivencia del injerto un año después de la operación. Las salidas de estos dos modelos se han utilizando como entrada de un sencillo sistema basado en reglas que permite asignar el órgano del donante al receptor situado en lista de espera con una mayor probabilidad de supervivencia.
- Teledetección remota: Determinación, mediante el análisis de imágenes aéreas, del uso del terreno en los campos de olivos, para ayudar a las administraciones andaluza y europea a identificar qué fincas desarrollan una agricultura de conservación (estilo de agricultura que reduce la erosión del terreno existente entre filas de olivos mediante la utilización de cubiertas vegetales vivas o inertes).
- Microbiología predictiva: En respuesta a la demanda de los expertos en alimentación de contar con productos alimenticios más saludables, se ha analizado la capacidad de crecimiento de determinados microorganismos a partir de ciertos factores medioambientales de envasado y/o conservación.

Una segunda propuesta de este documento de Tesis está asociada con la obtención de modelos para la resolución de problemas de clasificación ordinal. Previamente a la obtención de estos modelos, se ha realizado un concienzudo estudio de aquellas métricas ordinales que pueden guiar el proceso evolutivo de un algoritmo multiobjetivo. Esta propuesta surge ante el creciente interés existente por aquellos problemas de clasificación que presentan un cierto orden en sus etiquetas de clase y debido a la falta de trabajos en la literatura del área de *Machine Learning* que resuelvan un problema ordinal mediante un algoritmo multiobjetivo.

ABSTRACT

This Thesis addresses the design of nominal classifiers using different neural network models. To do this, we analyze different metrics associated with different conflicting aspects of the “goodness” of a multiclass classifier. These conflictive aspects make us obtain models which consider different characteristics of the problem.

The traditional way to deal with problems with two or more conflicting objectives is the aggregation or weighting of the objectives, so that multiobjective problems can be transformed into a single objective problem. After this transformation, the problems can be solved by any algorithm, finally obtaining a solution that meets the objectives of the problem.

Since the appearance of evolutionary algorithms (algorithms based on the theory of evolution of species) and the concept of Pareto dominance, multiobjective problems have been solved by what is known as multiobjective evolutionary algorithms. These algorithms simultaneously train a set of models for several iterations, ordering the models obtained after each iteration according to the dominance relation between them. When speaking of Pareto dominance, a model dominates another if it is better or equal in all objectives and strictly better in one of them. Therefore, the output obtained after finishing the execution of multiobjective algorithm consists of a set of models that present no dominance relationship between them. This set is called the first Pareto front. Any one of these models is an optimal solution to the problem, although there is no way to determine which one is the best.

For helping to choose one of these models, in this Thesis, different techniques are studied, such as the selection of the extreme models of the Pareto optimal front or the selection of the model which minimizes the Euclidean distance between its position on the front and the ideal solution of the problem. The final conclusion of this study is that the behaviour of these different techniques depends on the specific characteristics of the problem analyzed.

Moreover, different modifications are applied to the multiobjective evolutionary algorithm considered, the PDE developed by H. Abbass. They are focused on avoiding premature convergence, and on the application of a local search algorithm with the aim of improving the performance without increasing the computational cost. Apart from these algorithms, the classic NSGAI and a memetic algorithm developed from this, MPENSGAI, have been also considered in the experimental study.

One of the main objectives of this Thesis is the application of these algorithms in different real problems, such as:

- **Medicine:** Selecting the most suitable patient on the waiting list for an organ available in liver transplants. For this, we were provided with a dataset consisting of 1003 donor/recipient pairs from which two models were obtained to predict the graft survival one year after the operation. The outputs of these two models were used as the input for a simple rule-based system in order to assign the donor organ to the receiver of the waiting list with a higher probability of survival.
- **Remote Sensing:** Use of aerial images for determining land use in the olive groves, to assist Andalusian and European administrations in the identification of farms that develop conservation agriculture (a farming style which reduces soil erosion between the rows of olive trees by using living or inert groundcovers).
- **Predictive microbiology:** In response to the demand for healthier food products from food experts, we analyzed the growth capacity of certain microorganisms as a consequence of packaging and conservation environmental factors.

A second proposal of this Thesis is associated to the field of ordinal classification problems, specifically to the training of models for this kind of problems. A preliminary thorough study was conducted to determine what ordinal metrics can better guide the evolutionary process of a multiobjective algorithm. The proposal arises from the growing interest for those classification problems where an order in the class labels can be assumed, and also, due to low number of works in the literature of Machine Learning targeted at solving an ordinal problem from a multiobjective perspective.

*Todo hombre que conozco es superior a mí en algún sentido.
En ese sentido aprendo de él.*

— Ralph Waldo Emerson (1803 - 1882)

AGRADECIMIENTOS

En primer lugar me gustaría agradecerle a César Hervás la labor realizada como mi director de Tesis, así como todo su esfuerzo, dedicación y paciencia. Espero poder seguir trabajando muchos años al lado de este gran investigador.

También quiero agradecer a todos mis compañeros de laboratorio tanto por toda la ayuda que me han prestado como por los necesarios momentos de descanso que me habéis proporcionado. Muchas gracias Juan Carlos, Pedro, Javi, María, Francisco y Manolo.

De igual modo, me gustaría mostrar mi agradecimiento a todos aquellos investigadores que me han permitido trabajar con ellos. Gracias Francisca López, Antonio Valero, Javier Briceño y Manuel de la Mata. Sin la colaboración de todos vosotros esta Tesis no hubiera sido posible.

Una mención especial a Daniela, por todo su apoyo y comprensión en estos meses de trabajo. Gracias por apoyarme en los días duros y por ayudarme a desconectar y a disfrutar de los momentos de descanso.

Y finalmente, pero no por ello menos importante, me gustaría mostrar mi agradecimiento a mi familia y amigos por estar siempre en el lugar adecuado en el momento justo.

Muchas gracias a todos por ayudar en la realización de esta Tesis y hacerla posible.

Sinceramente, gracias.

ACRÓNIMOS

ANN	Redes Neuronales Artificiales (<i>Artificial Neural Networks</i>)
AUC	Área bajo la Curva ROC (<i>Area Under Curve Receiver Operating Characteristic</i>)
CCR	Precisión (<i>Correct Classification Rate</i>)
CI	Inteligencia Computacional (<i>Computational Intelligence</i>)
DE	Evolución Diferencial (<i>Differential Evolution</i>)
EA	Algoritmo Evolutivo (<i>Evolutionary Algorithm</i>)
K	Índice Kappa de Cohen (<i>Cohen's Kappa coefficient</i>)
MA	Algoritmo Memético (<i>Memetic Algorithm</i>)
MAE	Error Absoluto Medio (<i>Mean Absolute Error</i>)
MLP	Perceptrón Multicapa (<i>Multilayer Perceptron</i>)
MMAE	Máximo Error Absoluto Medio (<i>Maximum Mean Absolute Error</i>)
MOEA	Algoritmos Evolutivos Multiobjetivo (<i>Multiobjective Evolutionary Algorithms</i>)
MOP	Problema de Optimización Multiobjetivo (<i>Multiobjective Optimization Problem</i>)
MPANN	<i>Memetic Pareto evolutionary approach to Artificial Neural Networks</i>
MPDENN	<i>Memetic Pareto Differential Evolution Neural Networks</i>
MPENSGAII	<i>Memetic Pareto NSGAII</i>
MS	Mínima Sensibilidad (<i>Minimum Sensitivity</i>)
NSGAII	<i>Non-dominated Sorting Genetic Algorithm 2</i>
PDE	<i>Pareto Differential Evolution</i>
PU	Unidades Producto (<i>Product Units</i>)
RBF	Funciones de Base Radial (<i>Radial Basis Functions</i>)

ACRÓNIMOS

RMSE	Raíz del Error Cuadrático Medio (<i>Root Mean Square Error</i>)
SU	Unidades Sigmoide (<i>Sigmoidal Units</i>)

ÍNDICE GENERAL

Resumen	ix
Abstract	xi
Agradecimientos	xiii
Acrónimos	xv
Índice General	xvii
Índice de Figuras	xxi
Índice de Tablas	xxiii
1 INTRODUCCIÓN	1
1.1 Objetivos	4
1.2 Estructura de la Presente Tesis	5
2 INTRODUCTION	7
2.1 Objectives	10
2.2 Structure of this Thesis	11
3 INTRODUCCIÓN A LAS REDES NEURONALES ARTIFICIALES	13
3.1 Taxonomía	16
3.1.1 Redes Neuronales de Unidad Sigmoide	17
3.1.2 Redes Neuronales de Unidad Producto	18
3.1.3 Redes Neuronales de Función de Base Radial	19
4 ALGORITMOS EVOLUTIVOS MULTIOBJETIVO	23
4.1 Optimización Multiobjetivo	23
4.1.1 El Óptimo de Pareto	24
4.1.2 Otras Alternativas	25
4.2 Revisión del Estado del Arte en Clasificación Multiobjetivo	27
4.3 El Algoritmo MPDENN	29
4.3.1 PDE (<i>Pareto Differential Evolution</i>) y MPANN (<i>Memetic Pareto evolutionary approach to ANNs</i>)	30
4.3.1.1 Representación	31
4.3.1.2 El Algoritmo MPANN	31
4.3.2 El Algoritmo MPDENN (<i>Memetic Pareto Differential Evolution Neural Network</i>)	34
4.3.2.1 Hibridación	38
4.3.2.2 <i>iRprop⁺</i> como Algoritmo de Búsqueda Local	38
4.3.2.3 Algoritmos Meméticos Desarrollados	39
4.3.3 Diferencias entre los Algoritmos MPANN y MPDENN	41
4.4 El Algoritmo MPENSGAII	42
4.4.1 Operadores Empleados en el MPENSGAII	45

ÍNDICE GENERAL

4.4.1.1	Ordenamiento Rápido de Individuos	45
4.4.1.2	Distancia <i>Crowding</i>	45
4.4.1.3	Operadores de Mutación	46
4.5	Selección de las Mejores Soluciones del Frente de Pareto	47
5	MÉTRICAS DE RENDIMIENTO EN CLASIFICACIÓN NOMINAL	53
5.1	Métricas Asociadas a la Matriz de Confusión	53
5.2	Métricas no Relacionadas con la Matriz de Confusión	57
5.3	Relación entre la Mínima Sensibilidad y la Precisión	58
6	PROBLEMA DE ASIGNACIÓN DONANTE-RECEPTOR EN LOS TRASPLANTES HEPÁTICOS	63
6.1	Descripción de la Base de Datos Utilizada	64
6.2	Diseño Experimental	65
6.3	Resultados Obtenidos	66
6.4	Sistema Basado en Reglas	68
6.4.1	Ejemplo de Aplicación del Sistema Basado en Reglas	72
6.5	Otros Estudios	73
7	RESOLUCIÓN DE UN PROBLEMA REAL DE TELEDETECCIÓN REMOTA	77
7.1	Diseño Experimental	77
7.2	Resultados	80
8	RESOLUCIÓN DE UN PROBLEMA DE CRECIMIENTO MICROBIANO	87
8.1	Problemas Analizados y Diseño Experimental	87
8.1.1	<i>Staphylococcus Aureus</i>	88
8.1.2	<i>Shiguella Flexneri</i>	88
8.2	Metodología de Resolución del Problema	89
8.3	Resultados	90
8.4	Enfoque Ordinal	94
9	ALGORITMOS MULTIOBJETIVO EN CLASIFICACIÓN ORDINAL	95
9.1	Algoritmos para Clasificación Ordinal	96
9.1.1	Métodos de Umbral	97
9.2	Enfoque Multiobjetivo en Clasificación Ordinal	99
9.2.1	Modelo de Red Neuronal para Clasificación Ordinal	100
9.3	Métricas en Clasificación Ordinal	103
9.3.1	Medidas Ordinales Basadas en Correlación	104
9.3.2	Medidas de Asociación Ordinales	106
9.3.3	Correlación entre las Métricas Ordinales	106
9.3.4	Relación entre el Par de Métricas MAE y MMAE	112
9.3.4.1	Representación Gráfica del Par MMAE-MAE	113
10	COMENTARIOS FINALES Y FUTURAS LÍNEAS DE INVESTIGACIÓN	117
10.1	Conclusiones	117
10.2	Futuras Líneas de Investigación	120
11	CONCLUDING REMARKS AND FUTURE RESEARCH LINES	121

11.1 Conclusions 121
11.2 Future Research Lines 123

BIBLIOGRAFÍA 125

A PUBLICACIONES ASOCIADAS CON LA PRESENTE TESIS 137

A.1 Capítulos de Libro 137

A.2 Artículos en Revista 138

A.3 Artículos en Congresos Internacionales 158

A.4 Artículos en Congresos Nacionales 160

ÍNDICE DE FIGURAS

Figura 3.1	Esquema de una red <i>feedforward</i> con una única capa oculta. . . .	15
Figura 3.2	Esquema de una red <i>feedforward</i> con una única capa oculta simplificado mediante el uso de la función <i>softmax</i>	16
Figura 3.3	Activación de la GRBF en un espacio unidimensional con $w = 0$ y $r = 1$ para diferentes valores de τ	21
Figura 4.1	Ejemplo de frente de Pareto para un problema de minimización de dos objetivos.	25
Figura 4.2	Pseudocódigo del algoritmo MPANN.	32
Figura 4.3	Pseudocódigo del algoritmo MPDENN.	35
Figura 4.4	Flujo de información del algoritmo MPDENN.	36
Figura 4.5	Esquema del procedimiento de búsqueda local seguidos por los algoritmos MPDENN2 y MPDENN3.	40
Figura 4.6	Pseudocódigo del algoritmo MPENSGAII.	43
Figura 4.7	Flujo de información del algoritmo MPENSGAII.	44
Figura 4.8	Métodos de selección de individuos.	48
Figura 4.9	Ejemplo de distancias a las soluciones ideal positiva e ideal negativa.	50
Figura 5.1	Región factible (zona blanca) en el espacio (MS, CCR) para un problema de clasificación.	60
Figura 7.1	Imágenes aéreas de las fincas de Espejo y Montilla.	78
Figura 7.2	Imágenes de los diferentes elementos a discriminar: olivos, suelo desnudo y cubiertas vegetales.	79
Figura 7.3	Frente de Pareto en entrenamiento en el espacio (MS, A) y valores asociados en el espacio (MS, CCR) en generalización para las bases de datos Primavera y Verano.	83
Figura 8.1	Frente de Pareto en entrenamiento en el espacio (MS, A) y valores asociados en el espacio (MS, CCR) en generalización para las bases de datos <i>S. Aureus</i> y <i>S. Flexneri</i>	92
Figura 9.1	Modelo de red neuronal empleado en clasificación ordinal. . . .	101
Figura 9.2	Región factible (zona blanca) en el espacio $(MMAE, MAE)$	114
Figura 9.3	Frente de Pareto en entrenamiento y valores asociados en generalización obtenidos con la base de datos SWD.	115

ÍNDICE DE TABLAS

Tabla 5.1	Matriz de confusión $M(g)$ para un problema con Q clases.	54
Tabla 6.1	Características de la bases de datos donante-receptor.	65
Tabla 6.2	Resultados obtenidos en generalización: media y desviación típica ($Media_{DT}$).	67
Tabla 6.3	Resultados de los mejores modelos generados por los algoritmos MPENSGAII-E (GRBF) y MPENSGAII-MS (GRBF).	67
Tabla 6.4	Resultados del test de Wilconxon para $\alpha = 0,05$	69
Tabla 6.5	Características de los receptores empleados en el ejemplo.	72
Tabla 6.6	Características de los donantes/órganos y factores preoperatorios empleados en el ejemplo.	73
Tabla 6.7	Ejemplo de aplicación del sistema basado en reglas.	74
Tabla 7.1	Características de las bases de datos Primavera y Verano.	78
Tabla 7.2	Resultados obtenidos en entrenamiento (finca de Espejo).	81
Tabla 7.3	Resultados obtenidos en generalización (finca de Montilla).	82
Tabla 7.4	Resultados de los tests estadísticos aplicados sobre los valores obtenidos de generalización (finca de Montilla).	84
Tabla 7.5	Mejor modelo obtenido por el algoritmo MPDENN2-MS para la base de datos Primavera.	85
Tabla 7.6	Mejor modelo obtenido por el algoritmo MPDENN2-E para la base de datos Verano.	86
Tabla 8.1	Diseño factorial para la base de datos <i>S. Flexneri</i>	89
Tabla 8.2	Características de las bases de datos <i>S. Aureus</i> y <i>S. Flexneri</i>	90
Tabla 8.3	Tiempo de ejecución en entrenamiento y resultados obtenidos en generalización: media y desviación típica ($Media_{DT}$).	91
Tabla 8.4	Resultados de los tests estadísticos aplicados sobre los valores obtenidos en generalización.	93
Tabla 9.1	Descomposición binaria para un problema ordinal de 5 clases.	97
Tabla 9.2	Matrices de correlación obtenidas por los diferentes algoritmos.	108
Tabla 9.3	Matriz resumen de los 4 estudios de correlaciones.	109
Tabla 9.4	Resultados obtenidos para la métrica CCR en generalización.	110
Tabla 9.5	Resultados obtenidos para la métrica MAE en generalización.	110
Tabla 9.6	Resultados obtenidos para la métrica MMAE en generalización.	111
Tabla 9.7	Resultados obtenidos para la métrica τ_b en generalización.	111

INTRODUCCIÓN

Una de las características que diferencian al hombre del resto de seres vivos es la búsqueda constante de nuevas vías para mejorar sus condiciones de vida. Ante un determinado problema, el ser humano plantea diferentes soluciones y desarrolla aquella que le reporta más beneficios. Estos beneficios pueden ser físicos, como el empleo de artefactos para el manejo y transporte de mercancías pesadas, o intelectuales, como el uso de máquinas calculadoras para la resolución, de manera rápida y automática, de operaciones aritméticas. Sin embargo, existen diversos problemas que no pueden ser tratados mediante artefactos o máquinas calculadoras y que, por lo tanto, deben ser resueltos por seres humanos expertos en el ámbito concreto del problema. Estos problemas se caracterizan por la falta de un procedimiento algorítmico que permita su resolución. Por ejemplo, algunos de estos problemas son la clasificación de ciertos objetos en base a una serie de características, la predicción de una variable que depende de varios factores o la toma de decisiones en situaciones complejas. La Inteligencia Computacional (*Computational Intelligence*) (CI) es una disciplina centrada en el estudio y desarrollo de sistemas adaptativos con el objetivo de emular el comportamiento de los expertos para permitir, mediante el análisis de grandes bases de datos, la resolución de este tipo de problemas. La CI está compuesta por técnicas como las Redes Neuronales Artificiales (*Artificial Neural Networks*) (ANN) [15, 68], la Computación Evolutiva (*Evolutionary Computation*) [58], la Lógica Difusa (*Fuzzy Logic*) [131] y los métodos bioinspirados como los enjambre de partículas (*particle swarm*) [81]. Todas estas técnicas han sido aplicadas en una gran variedad de problemas complejos en campos tan diversos como la biomedicina, los sistemas de reconocimiento automáticos de patrones, la visión artificial o los sistemas de conducción automáticos.

Una de las técnicas de la CI que más se ha desarrollado en los últimos años son las ANNs. Esta metodología pretende emular ciertas características propias del cerebro de los humanos, como la capacidad de memorizar y de asociar hechos. Si se examinan con atención aquellos problemas que no pueden resolverse de manera algorítmica, se observa que todos tienen una característica en común: la experiencia. El experto es capaz de resolver estas situaciones acudiendo a la experiencia acumulada. Así, parece claro que una forma de aproximarse al problema consista en la construcción de sistemas que sean capaces de reproducir esta característica humana. Las redes neu-

ronales son modelos no lineales simplificados que simulan el comportamiento de las neuronas del cerebro humano y, por consiguiente, son una buena herramienta para la construcción de dichos sistemas. Es bien conocido que los humanos son capaces de aprender, entendiendo aprendizaje como el proceso a través del cual se adquiere más información acerca de un problema, que inicialmente no podía resolverse, hasta alcanzar su resolución. Así, las ANNs constan de unidades de procesamiento que intercambian datos o información, por lo que pueden ser utilizadas para reconocer patrones, incluyendo imágenes, manuscritos o secuencias de tiempo, y tienen la capacidad de aprender y mejorar su funcionamiento mediante este aprendizaje.

El componente principal de las ANNs es la neurona, entendida como una unidad de procesamiento que recibe unos valores de entrada y proporciona un valor de salida. Dentro de las ANNs, las neuronas se estructuran en tres capas: la capa de entrada, que recibe las variables del problema; la capa oculta, que se encarga del procesamiento de la información; y la capa de salida, la cual presenta el valor de la variable dependiente. Por tanto, las ANNs son modelos de clasificación que permiten determinar la clase de pertenencia de los patrones a partir de los valores de las variables de entrada del problema que se esté analizando [133]. Las ANNs necesitan de un proceso de entrenamiento en el que se ajusta tanto la estructura como los coeficientes del modelo a un problema concreto con el objetivo de mejorar la capacidad de predicción para dicho problema. Este proceso de ajuste o adaptación al problema puede ser llevado a cabo mediante diversos métodos, entre los que destacan los métodos de búsqueda local, los métodos de búsqueda global o los métodos híbridos, que combinan la búsqueda local y global. Estos métodos ajustan tanto los pesos de la red neuronal (valor de las conexiones entre neuronas) como la propia estructura de la red (número de nodos en capa oculta y número de conexiones).

Los algoritmos de búsqueda local presentan problemas de estancamiento en óptimos locales y de lentitud en obtener la convergencia hacia el óptimo en problemas multimodales y epistáticos. Sin embargo, si el modelo se encuentra cerca del óptimo global, son los mejores métodos en términos de desempeño y resultados. Entre los métodos de búsqueda local cabe destacar los algoritmos de retropropagación del error de predicción [22].

Para solventar los inconvenientes presentados por los métodos de búsqueda local, se han propuesto muchos métodos heurísticos para el entrenamiento de las ANNs, como la búsqueda tabú (*tabu search*) [63], los métodos basados en enjambres de partículas (*particle swarm*) [81] o los algoritmos evolutivos (*evolutionary algorithms*) [11]. De entre todos los métodos de búsqueda global que se han propuesto, los algoritmos evolutivos son los que han tenido una mayor acogida entre la comunidad científica y han sido los más empleados para el diseño y ajuste de las ANNs.

Un Algoritmo Evolutivo (*Evolutionary Algorithm*) (EA) es un método heurístico de búsqueda basado en poblaciones de soluciones [108]. Esta metodología, basada en la evolución natural de las especies, aplica los operadores evolutivos naturales (selec-

ción, reproducción y mutación) a una población de posibles soluciones al problema (en este caso ANNs). Los individuos de esta población que estén mejor capacitados para la resolución del problema son seleccionados y el resto de individuos son descartados. Las vacantes dentro de la población son ocupadas por otros nuevos individuos generados a partir de una combinación de dos o más individuos de la población o de una mutación sufrida por alguno de los individuos. Tras la finalización del proceso evolutivo, el EA muestra la mejor solución obtenida, aunque no tiene porque ser la óptima. Debido a esto, un EA está considerado como un método estocástico de búsqueda ciega de soluciones cuasi-óptimas. Una de las principales características de los EAs es su capacidad para realizar búsquedas globales sin quedar atrapados en óptimos locales.

La Evolución Diferencial (*Differential Evolution*) (DE) [121] es otro tipo de técnica de optimización global, aplicado ampliamente en la resolución de problemas complejos. La DE mantiene una población de soluciones candidatas (en nuestro caso ANNs), las cuales se recombinan y mutan para producir nuevos individuos. La recombinación se realiza añadiendo la diferencia de dos individuos (padres secundarios) a un tercer individuo (padre principal). Con la DE se asegura la convergencia a un valor óptimo (posiblemente local) de manera consistente a lo largo de una secuencia de ejecuciones independientes.

Es frecuente la utilización de métodos de búsqueda local junto con los EAs. De este modo, los EAs convergen hacia soluciones aceptables dentro del espacio de búsqueda (búsqueda global), mientras que los métodos de búsqueda local terminan de ajustar estas soluciones, dentro de una región pequeña del espacio de búsqueda, para encontrar el óptimo global (o una solución lo más cercana posible). La incorporación de los métodos de búsqueda local en los EAs se conoce como algoritmos meméticos (*memetic algorithms*) [116] o híbridos (*hybrid algorithms*) [64].

En la mayoría de los problemas aparecen varios objetivos contrapuestos a optimizar que deben ser considerados para obtener una solución viable, a menudo, sin ningún conocimiento *a priori* de cómo los objetivos interactúan entre sí. Una forma de resolver este tipo de situaciones es por medio de un tipo de algoritmos conocidos como Algoritmos Evolutivos Multiobjetivo (*Multiobjective Evolutionary Algorithms*) (MOEA) [26], que junto con las ANNs forman lo que se conoce como redes neuronales artificiales evolutivas multiobjetivo (*Multiobjective Evolutionary Artificial Neural Networks*, MOEANN) [77]. La elección de los objetivos a optimizar no siempre es la adecuada cuando queremos obtener un nivel aceptable de buena clasificación en todas las clases. De esta forma, surgen nuevas aproximaciones para la obtención de clasificadores basados en modelos de ANNs donde se optimice tanto la precisión global como el nivel de buena clasificación en todas las clases [55].

Existen otro tipo de problemas donde lo importante, a parte de la correcta clasificación de los patrones, es la relación de orden existente entre las clases del problema. Estos problemas, donde se considera que los patrones están etiquetados mediante

una escala ordinal discreta y finita, son conocidos como problemas de clasificación ordinal [7, 92] y están teniendo un creciente interés en los últimos años dentro de la comunidad de *Machine Learning*. Otro aspecto importante de la clasificación ordinal está relacionado con el coste de los errores de mala clasificación, ya que un error de clasificación en dos clases distantes es más problemático que un error entre clases adyacentes (puesto que se incumplen más restricciones de orden). La principal dificultad al trabajar con problemas de clasificación ordinal radica en la falta de precisión en el cálculo de las distancias entre las diferentes clases, ya que los resultados obtenidos dependen del valor de las etiquetas de clase utilizadas. Ejemplos de problemas de clasificación ordinal pueden ser la evaluación de los alumnos por parte de un profesor (“matricula de honor”, “sobresaliente”, “notable”, “aprobado” y “suspense”) o los resultados obtenidos con una encuesta en escala de Likert (“totalmente en desacuerdo”, “en desacuerdo”, “ni de acuerdo ni en desacuerdo”, “de acuerdo” o “totalmente de acuerdo”).

La clasificación ordinal comparte ciertas similitudes con la clasificación nominal y con la regresión. La clasificación ordinal, al igual que la clasificación nominal, busca clasificar correctamente un conjunto de clases finitas. La diferencia estriba en que la clasificación ordinal existe un orden entre las diferentes clases que debe ser considerado a la hora de realizar la predicción. Con respecto a la regresión, ambos paradigmas pretenden predecir una variable de salida que presenta un cierto orden entre sus valores, sin embargo, en regresión la variable es continua e infinita mientras que en la clasificación ordinal esta variable es discreta y finita.

En la literatura del área de *Machine Learning* no existen muchos trabajos que afronten la clasificación ordinal con un enfoque multiobjetivo. Los problemas ordinales multiobjetivo han sido tradicionalmente tratados como problemas nominales con varios objetivos. Algunos de los trabajos ordinales multiobjetivo que existen son [127], donde se propone la resolución de un problema ordinal considerando 6 objetivos simultáneamente (5 de ellos son ordinales), o [20], donde emplean un SVM (*Support Vector Machines*) para maximizar, independientemente, los márgenes superior e inferior con la finalidad de evitar los errores de clasificación producidos al clasificar un patrón en alguna clase mayor o menor, respectivamente.

En la presente Tesis se han estudiado las modificaciones necesarias para transformar un MOEA nominal en uno ordinal, es decir, en un MOEA que tenga en cuenta el orden existente entre las diferentes clases del problema a la hora de realizar la predicción. Las principales modificaciones están relacionadas con el empleo de un modelo de red neuronal basado en el modelo ordinal POM (*Proportional Odds Model*) y con el empleo de funciones ordinales para guiar la evolución del algoritmo multiobjetivo.

1.1 OBJETIVOS

Los principales objetivos de la presente Tesis son:

- Desarrollo e implementación de un algoritmo evolutivo multiobjetivo, basado en la evolución diferencial para resolver problemas de clasificación binaria y multiclase utilizando diferentes funciones de base (sigmoide, producto, RBF...) en el diseño de modelos redes neuronales, tomando como base los trabajos realizados por H. Abbass [1, 2, 3, 5]. Así mismo, se estudiará el algoritmo evolutivo multiobjetivo desarrollado por K. Deb *et al.* [41] y un algoritmo memético implementado a partir de éste [55].
- Utilizar medidas nominales para medir tanto el rendimiento global de los modelos obtenidos como la clasificación en cada una de las clases. Estas medidas serán utilizadas para guiar el proceso evolutivo de los algoritmos multiobjetivo.
- Aplicación de los algoritmos evolutivos multiobjetivo en problemas reales de diferentes áreas, tales como la teledetección remota, la microbiología predictiva o la toma de decisiones en la asignación donante-receptor dentro de los trasplantes hepáticos.
- Desarrollo e implementación de algoritmos evolutivos multiobjetivo para la resolución de problemas ordinales. Estos algoritmos estarán basados en los algoritmos evolutivos multiobjetivo nominales.
- Estudiar las diferentes métricas empleadas en clasificación ordinal y analizar las posibles relaciones entre ellas para determinar cuáles deben ser empleadas para guiar el proceso evolutivo de un algoritmo multiobjetivo.

1.2 ESTRUCTURA DE LA PRESENTE TESIS

El documento de la presente Tesis se ha organizado siguiendo el siguiente esquema:

- En el Capítulo 3 se desarrollarán los conceptos fundamentales de los modelos de ANNs, detallando tanto la estructura del modelo como los diferentes tipos de funciones de base empleados.
- El Capítulo 4 presenta el concepto de clasificación multiobjetivo junto con una revisión de los algoritmos evolutivos multiobjetivo más empleados en la literatura. Además, se detallan los algoritmos multiobjetivo de H. Abbass y K. Deb y se expone el algoritmo desarrollado por el autor de la presente Tesis. Este algoritmo está basado en los trabajos de H. Abbass. Finalmente, en este Capítulo se muestran diferentes técnicas empleadas para la selección de un modelo concreto perteneciente al frente de Pareto.
- En el Capítulo 5 se muestra una revisión de las métricas empleadas para la evaluación de la calidad de los clasificadores, prestando especial interés a la relación existente entre las métricas precisión (CCR) y mínima sensibilidad (MS).

INTRODUCCIÓN

- En los siguientes Capítulos se aplican los algoritmos descritos en diferentes problemas reales. En el Capítulo 6 se resuelve el problema de asignar el órgano a un paciente en los trasplantes hepáticos. En el Capítulo 7 se resuelve un problema de teledetección remota con el objetivo de ayudar a las administraciones andaluza y europea. En el Capítulo 8 se muestra la resolución de un problema de predicción del crecimiento microbiano en determinadas circunstancias ambientales.
- En el Capítulo 9 se analizan los problemas de clasificación ordinal (problemas donde existe un cierto orden entre las etiquetas de clase), realizando una revisión de los métodos ordinales en la literatura y estudiando la relación existente entre las distintas métricas ordinales, con el objetivo de obtener un par de métricas no correladas para guiar la evolución de un algoritmo evolutivo.
- Los Capítulos 10 y 11 muestran, tanto en español como en inglés, las principales conclusiones alcanzadas durante la elaboración de la presente Tesis, así como las futuras líneas de investigación.
- Finalmente, el Anexo A enumera las publicaciones realizadas durante la elaboración de la presente Tesis.

INTRODUCTION

One of the characteristics that distinguish humans from other living beings is the constant search for new ways to improve their living conditions. Given a particular problem, humans usually propose different solutions and finally consider the one that brings them more benefits. These benefits can be physical, such as the use of devices for handling and transporting heavy goods, or intellectual, as the use of calculating machines to solve, quickly and automatically, arithmetic operations. However, there are several problems that can not be treated by calculating machines or devices and, therefore, should be solved by human experts in the particular field of the problem. These problems are characterized by the lack of an algorithmic process which allows their resolution. For example, some of these problems are the classification of certain objects based on a number of characteristics, the prediction of a variable that depends on several factors or decision making processes in complex situations. The Computational Intelligence (CI) is a discipline focused on the study and development of adaptive systems in order to emulate the behavior of experts to allow the resolution of these problems by analyzing large datasets. CI comprises techniques such as Artificial Neural Networks (ANNs) [15, 68], Evolutionary Computation [58], Fuzzy Logic [131] and bio-inspired methods such as Particle Swarm Optimization [81]. All these techniques have been applied in a variety of complex problems in areas as diverse as those including biomedicine, automatic pattern recognition systems, computer vision or automatic driving systems.

ANNs are one of the CI techniques most widely studied in recent years. This methodology is a way to simulate certain characteristics of the human brain, such as the ability to memorize and associate facts. If we carefully consider all those problems which can not be solved algorithmically, we realize that they share one common feature: experience. The expert is able to solve the situation using their accumulated experience. Thus, it seems clear that one way to approach these problems consists in building systems that are able to reproduce this human characteristic. Neural networks are simplified nonlinear models that simulate the behavior of the neurons of the human brain and, therefore, they are a suitable tool for building such systems. It is well known that humans are able to learn, understanding learning as the process through which more information is acquired about a problem that could not be

solved initially, in order to reach a solution. Thus, ANNs consist of processing units that exchange data or information and can be used to recognize patterns, including images, manuscripts or time sequences, with the ability of learning and improving their performance through this learning.

The main component of the ANNs is the neuron, defined as a processing unit which receives input values and provides an output value. The neurons are usually structured in three layers within the network: the input layer, which receives the variables of the problem; the hidden layer, which is responsible for the information processing; and the output layer, which shows the obtained value of the dependent variable. Therefore, ANNs are classification models for determining the class membership of the patterns from the values of the input variables of the problem that is being analyzed [133]. They require a training process in which both the structure and the coefficients of the model are adjusted to a particular problem to improve the predictive performance for the problem. The process of adjustment or adaptation to the problem can be accomplished by several algorithms, including local search methods, global search methods or hybrid methods, which combine global and local search. These optimize adjust both the weights of the neural network (the value of the connections between neurons) and the structure of the network (number of nodes in the hidden layer and number of connections).

Local search algorithms have the problem of being trapped in local optima and they can need much time to converge to the optimum for multimodal and epistatic problems. However, if the current solution is close to the global optimum, they are the best methods in terms of performance and results. Among the local search methods, the backpropagation error algorithms stand out [22].

To overcome the drawbacks of local search methods, many heuristics have been proposed for the training of ANNs, such as Tabu Search [63], methods based in Particle Swarm Optimization [81] or Evolutionary Algorithms (EA) [11]. From all the proposed global search methods, evolutionary algorithms are maybe the heuristics with greater acceptance from the scientific community, and they have been the most widely used for the design and training of ANNs.

An EA is a heuristic search method based on populations of solutions [108]. This methodology, based on the natural evolution of the species, applies natural evolutionary operators (selection, reproduction and mutation) to a population of possible solutions to the problem (in this case, a population of ANNs). The individuals in the population undergo a process of selection of the individuals better adapted to solve the problem. During this process, less qualified individuals are discarded. Vacancies within the population are occupied by new individuals produced by a combination of two or more individuals in the population or a mutation suffered by any of the individuals. After completion of the evolutionary process, the EA returns the best solution obtained, but not necessarily the optimal solution. Due to this, an EA is considered as

a stochastic blind search of quasi-optimal solutions. One of the main features of the **EAs** is its ability to perform global searches without being trapped in local optima.

The Differential Evolution (**DE**) [121] is another type of global optimization technique, widely applied in complex problem solving. The **DE** maintains a population of candidate solutions (in our case, **ANNs**), which are recombined and mutated to produce new individuals, which will be chosen according to the value of the performance function. Recombination is performed by adding the difference of two individuals (secondary parents) to a third individual (main parent). By using **DE**, convergence to an optimum value (although maybe a local optimum) is consistently assured after a sequence of independent runs.

The use of local search methods together with **EAs** is common in **CI**. Thus, the **EA** tries to find acceptable solutions within the search space (global search), while local search methods improve these solutions within a small region of the search space to find the global optimum (or a solution as close as possible to it). The incorporation of local search methods in the **EAs** is known as memetic algorithms [116] or hybrid algorithms [64].

Most of the optimization problems shows several competitive objectives to optimize which must be considered to obtain a good solution often without any a priori knowledge about how objectives interact with each other. One way to solve this type of situation is by using a type of algorithms known as Multiobjective Evolutionary Algorithms (**MOEAs**) [26], which, together with **ANNs**, result in what is known as Multiobjective Evolutionary Artificial Neural Networks (**MOEANN**) [77]. In the context of classification, the choice of the objectives to optimize is not always suitable when we want to obtain an acceptable level of good classification in all classes. Due to this, there are new approaches for obtaining classifiers based on **ANNs** which optimize both the overall accuracy as a good classification level for all the classes [55].

There are other kind of classification problems in which, the important thing, apart from the correct classification of patterns, is the order relation between the classes of the problem. These problems, which considers that the patterns are labeled by a discrete and finite ordinal scale, are known to as ordinal classification problems [7, 92] and they are receiving an increasing interest in recent years in the community of Machine Learning. One important aspect of ordinal classification is related with the cost of misclassification errors, since a classification error between two distant classes is more problematic than an error between adjacent classes (because of the order restrictions). The main difficulty in ordinal classification problems is the lack of a principled way to calculate the distances between the different classes. The obtained results can depend on the value of the class labels considered. Examples of ordinal classification problems can be the evaluation of students by a teacher ("A+", "A", "B", "C", "D" or "F") or the results obtained with a Likert scale survey ("strongly disagree", "disagree", "neither agree nor disagree", "agree" or "strongly according").

The ordinal classification shares certain similarities with nominal classification and regression. Ordinal classification, as nominal classification, targets at obtaining correct classification from a set of finite classes. The difference is that, in ordinal classification, there is an order between the different classes that must be considered when making the prediction and evaluating the classifier. With respect to regression, both paradigms claim to predict an output variable having a certain order between their values, however, in regression, the variable is continuous and infinite while in ordinal classification this variable is discrete and finite.

In the literature of Machine Learning there are not many works that address ordinal classification with a multiobjective approach. Ordinal multiobjective problems have traditionally been treated as nominal problems with several objectives. Some of the ordinal multiobjective works that exist are the following: in [127], the resolution of an ordinal problem considering 6 objectives simultaneously (5 of them are ordinal) is proposed; or [20], where Support Vector Machine (SVM) is proposed to maximize upgrading and downgrading margins in order to avoid misclassification errors produced when classifying a pattern in a class higher or lower than the real one, respectively.

In this Thesis, we have studied the changes needed to transform a nominal MOEA into an ordinal one, i.e. a MOEA that takes into account the existing order between different classes of the problem when evaluating the classifiers. The main changes are related to the use of a neural network model based on the ordinal Proportional Odds Model (POM) and the use of ordinal metrics to guide the evolution of the multiobjective algorithm.

2.1 OBJECTIVES

The main objectives of this Thesis are:

- Development and implementation of a multiobjective evolutionary algorithm based on differential evolution for solving binary and multiclass classification problems using neural network models with different basis functions (sigmoidal, product, RBF...), based on the work done by H. Abbass [1, 2, 3, 5]. In addition, the multiobjective evolutionary algorithm developed by K. Deb *et al.* [41] and a new memetic algorithm [55] will be studied.
- Use of nominal metrics for measuring both the overall performance of obtained models and the classification level for each of the classes. These measures will be used to guide the evolutionary process of multiobjective algorithms.
- Application of multiobjective evolutionary algorithms on real problems from different areas, such as precision agriculture, predictive microbiology or decision making in donor-recipient allocation for liver transplantation.

- Development and implementation of multiobjective evolutionary algorithms for solving ordinal problems. These algorithms will be based on nominal multiobjective evolutionary algorithms.
- Study of the different metrics considered in ordinal classification and analysis of the possible relationships between them in order to determine which ones should be used to guide the evolutionary process of a multiobjective algorithm.

2.2 STRUCTURE OF THIS THESIS

The Thesis is organized as follows:

- Chapter 3 presents the fundamental concepts of ANNs, detailing both the model structure and the different types of basis functions used.
- Chapter 4 introduces the concept of multiobjective classification and a review of multiobjective evolutionary algorithms most commonly used in the literature. Additionally, the multiobjective algorithms developed by H. Abbass and K. Deb are detailed and the algorithm developed by the author of this Thesis is presented. This algorithm is based on the work of H. Abbass. Finally, this Chapter shows different techniques used for the selection of a particular model belonging to the Pareto front.
- Chapter 5 shows a review of the metrics used for evaluating the quality of classifiers, focusing special attention on the relationship between accuracy (CCR) and minimum sensitivity (MS) metrics.
- In the following Chapters, the described algorithms are applied in different real problems. Chapter 6 discusses the problem of allocating the organ to a patient in liver transplants. Chapter 7 tackles a problem of remote sensing to help Andalusian and European administrations. Finally, Chapter 8 deals with a prediction problem of the microbial growth under certain environmental circumstances.
- In Chapter 9, ordinal classification problems are analyzed (problems where there is a certain order between class labels), carrying out a review of ordinal methods in the literature and studying the relationship between the different ordinal metrics, in order to obtain a pair of uncorrelated metrics to guide the evolution of an evolutionary algorithm.
- Chapters 10 and 11 shown the main conclusions reached from this Thesis, as well as future research, in Spanish and English, respectively.
- Finally, Appendix A lists the publications associated to this Thesis.

3

INTRODUCCIÓN A LAS REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales (ANNs) se definen como modelos no lineales diseñados para resolver problemas emulando el funcionamiento del cerebro humano [15]. De esta manera su interpretación biológica es similar a la activación de las neuronas cerebrales [93]. A diferencia de los algoritmos procedurales convencionales que siguen un flujo secuencial, las computaciones neuronales se realizan a través de un gran número de nodos, de forma distribuida y paralela.

La arquitectura de una ANN es la manera en que se distribuyen sus nodos y cómo se conectan unos con otros mediante las conexiones o enlaces. Los nodos pueden ser de tres tipos diferentes, dependiendo de su situación dentro de la red:

- Nodos de entrada: Reciben los valores de las variables independientes del problema, es decir, las entradas del modelo, y forman la capa de entrada.
- Nodos de salida: Producen las salidas del modelo, esto es, los valores de las variables dependientes del problema, y forman la capa de salida.
- Nodos ocultos: Son el resto de nodos y se agrupan en una o más capas ocultas. Aunque pueden existir varias capas ocultas, normalmente sólo se utiliza una, puesto que se ha demostrado que el uso de varias capas ocultas no mejora los resultados obtenidos.

La información manejada por los nodos de una ANN se puede transmitir de diversas maneras, dependiendo de las funciones usadas y del propósito de la ANN. De este modo, los tipos de transmisión existentes son los siguientes:

- Transmisión hacia delante: Este tipo de transmisión se produce desde los nodos de la capa de entrada hasta los nodos de la capa de salida. Las ANNs que presentan este tipo de transmisión se denominan redes *feedforward* o redes con transmisión de información hacia delante.
- Transmisión lateral: Este tipo de transmisión se produce entre nodos de la misma capa. Las ANNs que presentan este tipo de transmisión se denominan redes con transmisión lateral.

- Transmisión hacia atrás: Este tipo de transmisión (también llamada transmisión *feedback*) se produce desde los nodos de la capa de salida hasta los nodos de la capa de entrada. Las ANN que presentan este tipo de transmisión se denominan redes con realimentación o redes recurrentes.

La forma general de una ANN [68] es la de un modelo de caja negra en el sentido de que la interpretabilidad de las relaciones no lineales causa-efecto no se obtiene fácilmente y donde las características de entrada pertenecen, en general, a un espacio de una alta dimensión. Las neuronas de la capa oculta de la estructura de la red se modelan mediante unidades de proceso o nodos del modelo. Cada unidad de proceso está compuesta por los siguientes elementos:

- Conexiones de entrada ponderadas: Hacen el papel de las conexiones sinápticas. La existencia de conexiones determina si es posible que un nodo influya sobre otro, mientras que el valor de los pesos y el signo de los mismos definen el tipo (excitatorio/inhibitorio) y la intensidad de la influencia.
- Función de activación: Computa la entrada total combinada de todas las conexiones, generalmente como simple suma ponderada de todas las entradas recibidas, es decir, de las entradas multiplicadas por el peso de las conexiones. Llamaremos $f(\mathbf{x}, \mathbf{w})$ al valor obtenido, donde \mathbf{x} es el vector de variables de entrada a la red y \mathbf{w} es el vector de pesos de las conexiones de la capa de entrada con la capa oculta.
- Función de transferencia o salida: Calcula la salida del nodo en función de la activación de la misma. Se representa como $h(\cdot)$ y se usan diferentes tipos de funciones, desde simples funciones de umbral a funciones no lineales. El valor final de salida ϕ_j del nodo j -ésimo será el siguiente:

$$\phi_j(\mathbf{x}) = h(f(\mathbf{x}, \mathbf{w})).$$

En la presente Tesis se utilizan redes *feedforward* con una sola capa oculta y diferentes funciones de transferencia (eligiendo la que mejor se adapta a cada problema). La función asociada con cada uno de los nodos de la capa de salida de las redes *feedforward* empleadas es la siguiente:

$$f_q(\mathbf{x}, \boldsymbol{\theta}_q) = \beta_0^q + \sum_{m=1}^M \beta_m^q \phi_m(\mathbf{x}, \mathbf{w}_m), \quad \text{para } q = 1, \dots, Q,$$

donde q es una neurona de salida de la red (el número máximo de neuronas de salida es Q), $\boldsymbol{\theta}_q = (\beta_0^q, \beta_1^q, \dots, \beta_M^q, \mathbf{w}_1, \dots, \mathbf{w}_M)$ es el vector de pesos de la neurona q , $\mathbf{w}_m = (w_{m0}, w_{m1}, \dots, w_{mK})$ es el vector de pesos de las conexiones entre la capa de entrada y la neurona m de la capa oculta, M es el número de neuronas de la

capa oculta, K es el número de neuronas o características en la capa de entrada, \mathbf{x} es el vector de las entradas de la ANN, $\phi_j(\mathbf{x}, \mathbf{w}_m)$ es la función de base de la neurona m de la capa oculta y β_0^q y w_{m0} son los sesgos del modelo asociado a la neurona q en la capa de salida, y a la neurona m de la capa oculta respectivamente. Una representación de este modelo se puede apreciar en la Figura 3.1.

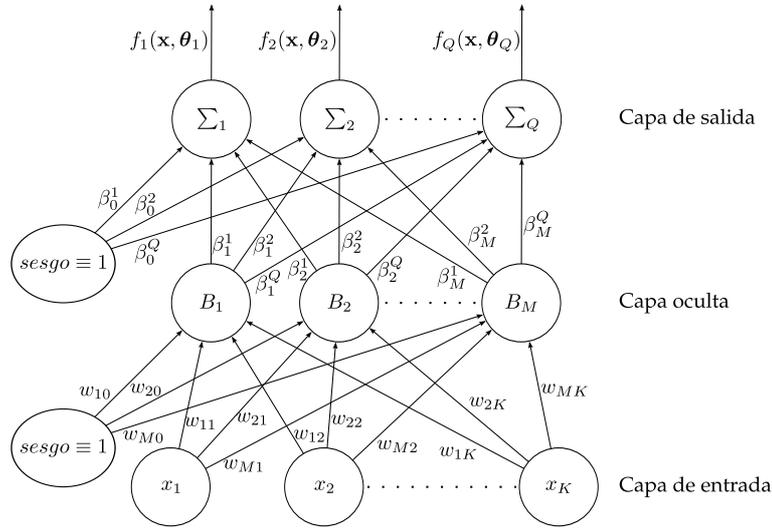


Figura 3.1: Esquema de una red *feedforward* con una única capa oculta.

El enfoque adoptado a la hora de interpretar la salida de una ANN es un enfoque probabilístico. Consideramos un esquema de codificación “1 de Q”, siendo Q el número de clases del problema. Una nueva observación se asigna a la clase correspondiente al valor de salida más grande de cada una de las neuronas que componen la capa de salida (existe una neurona por cada clase). Ese valor se calcula en base a la función *softmax*, que viene dada por la siguiente expresión:

$$g_q(\mathbf{x}, \theta_q) = \frac{\exp f_q(\mathbf{x}, \theta_q)}{\sum_{q=1}^Q \exp f_q(\mathbf{x}, \theta_q)},$$

siendo $g_q(\mathbf{x}, \theta_q)$ la probabilidad de que el patrón \mathbf{x} pertenezca a la clase q . La transformación *softmax* produce estimaciones positivas en todas las salidas, de forma que la suma de todas ellas es la unidad, lo que hace que las salidas puedan ser interpretadas como la probabilidad de pertenencia a la clase correspondiente.

Al tratarse de probabilidades de pertenencia a una clase, está claro que no es necesario calcularlas todas, puesto que la última probabilidad $g_Q(\mathbf{x}, \theta_q)$ se puede calcular en función del resto como $1 - \sum_{i=1}^{Q-1} g_i(\mathbf{x}, \theta_q)$. De esta forma, podemos simplificar el modelo definiendo la salida de una de las neuronas de la capa de salida con un valor constante e igual a 0, es decir, $f_Q(\mathbf{x}, \theta_Q) = 0$. Este cambio en el modelo se puede apreciar en la Figura 3.2.

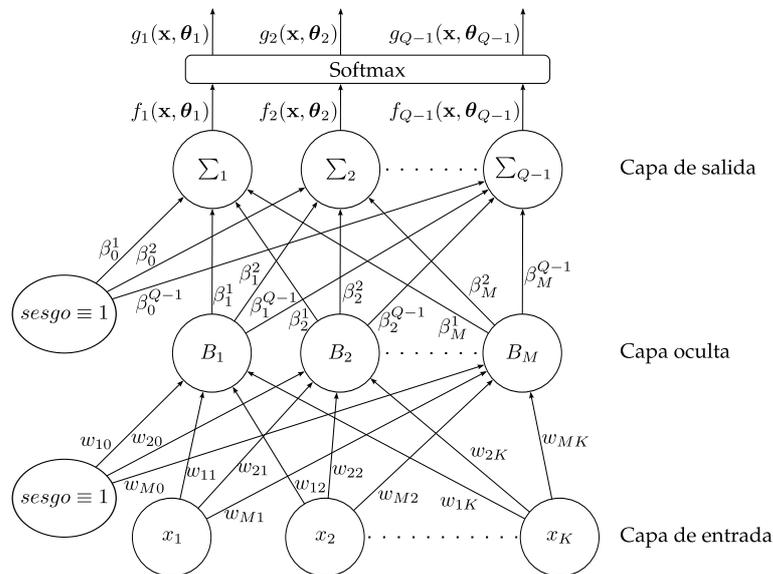


Figura 3.2: Esquema de una red *feedforward* con una única capa oculta simplificado mediante el uso de la función *softmax*.

3.1 TAXONOMÍA

En una ANN, cada función de base tiene como salida el resultado de una función de transferencia cuya entrada es el valor obtenido al aplicar una función de activación sobre las entradas de la red. En general, consideraremos dos tipos de funciones de activación de neuronas:

- Modelo aditivo: Es el caso usado con más frecuencia dentro de la teoría de ANNs. La función de salida de este tipo de nodos es la siguiente:

$$\phi_m(\mathbf{x}, \mathbf{w}_m) = h(w_{m0} + w_{m1}x_1 + w_{m2}x_2 + \dots + w_{mK}x_K),$$

siendo $\mathbf{w}_m = \{w_{m0}, w_{m1}, \dots, w_{mK}\}$ el valor de los coeficientes asociados al nodo, $h(\cdot)$ la función de transferencia y w_{m0} el sesgo o valor umbral de activación del nodo. La elección de la función de transferencia da lugar a distintos tipos de nodos aditivos, entre los cuales caben destacar las neuronas umbral o perceptrón [93] (que utilizan una función tipo escalón), las Unidades Sigmoidales (*Sigmoidal Units*) (SU) (que utilizan funciones de transferencia logísticas sigmoideas, tangente hiperbólica o arcotangente) y los nodos lineales (cuya función de transferencia es la función identidad).

- Modelo multiplicativo: Es un modelo más reciente diseñado para afrontar aquellos casos en los que existe una interacción entre las variables y las regiones de decisión no pueden separarse por hiperplanos [115]. Dentro de este modelo

multiplicativo, la alternativa más generalista es la denominada Unidades Producto (*Product Units*) (PU):

$$\phi_m(\mathbf{x}, \mathbf{w}_m) = x_1^{w_{m1}} \cdot x_2^{w_{m2}} \cdot \dots \cdot x_K^{w_{mK}},$$

siendo en este caso $\mathbf{w}_m = \{w_{m1}, w_{m2}, \dots, w_{mK}\}$, ya que el valor umbral carece de sentido para este modelo. Como puede observarse, la función de transferencia es la función identidad. Los pesos $w_{m1}, w_{m2}, \dots, w_{mK}$ son números reales, por lo que las PUs generalizan a otras unidades de tipo multiplicativo, como las denominadas unidades de orden superior (basadas en monomios con exponentes positivos y enteros).

Por otro lado, también se pueden considerar dos tipos fundamentales de funciones transferencia:

- Funciones de proyección: Son funciones de entorno global, como las SUs o PUs. Al ser globales, presentan dificultades en la aproximación de datos aislados pero suelen actuar mejor en problemas donde el número de variables es alto.
- Funciones de ventana o *kernel*: Son funciones de entorno local, como pueden ser las Funciones de Base Radial (*Radial Basis Functions*) (RBF). Poseen una mayor capacidad de aproximar datos anómalos aislados, pero una mayor dificultad de aproximar datos en entornos globales o cuando el número de entradas es demasiado alto.

3.1.1 Redes Neuronales de Unidad Sigmoide

Las redes de SU o Perceptrón Multicapa (*Multilayer Perceptron*) (MLP) son aquellas que están formadas por nodos tipo SU en su capa oculta, los cuales siguen un modelo aditivo de proyección con la siguiente función de salida:

$$\begin{aligned} \phi_m(\mathbf{x}, \mathbf{w}_m) &= \frac{1}{1 + \exp -(w_{m0} + w_{m1} \cdot x_1 + w_{m2} \cdot x_2 + \dots + w_{mK} \cdot x_K)} = \\ &= \frac{1}{1 + \exp -(w_{m0} + \sum_{k=1}^K w_{jk} \cdot x_k)}. \end{aligned}$$

Las redes MLP poseen una propiedad muy importante: la familia de funciones reales que representan puede aproximar cualquier función continua con suficiente precisión, siempre que el número de nodos de la capa oculta no esté acotado. Esta propiedad proporciona una sólida base de carácter teórico que ha sido esencial para el desarrollo, el estudio y las aplicaciones de estas redes [39, 71, 72]. Junto con esta propiedad, las redes MLP poseen las siguientes características:

- Son fáciles de entrenar, aunque se obtienen con frecuencia óptimos locales.

- Se obtienen modelos eficientes en entornos alisados.
- Son funciones acotadas.

3.1.2 Redes Neuronales de Unidad Producto

Las redes de **PU** [50] son aquellas que están formadas por nodos tipo **PU** en su capa oculta, los cuales siguen el modelo multiplicativo de proyección con la siguiente función de salida:

$$\phi_m(\mathbf{x}, \mathbf{w}_m) = x_1^{w_{m1}} \cdot x_2^{w_{m2}} \cdot \dots \cdot x_K^{w_{mK}} = \prod_{k=1}^K x_k^{w_{mk}},$$

siendo en este caso $\mathbf{w}_m = \{w_{m1}, w_{m2}, \dots, w_{mK}\}$, ya que en este tipo de **ANN** no definimos un sesgo en capa de entrada. Entre las ventajas de las redes con **PU** se encuentran las siguientes [91]:

- Como consecuencia del Teorema de Stone-Weierstrass [13], se ha demostrado que las redes de **PU** son aproximadores universales (obsérvese que las funciones polinómicas de varias variables son un subconjunto de los modelos basados en **PU**).
- La posibilidad de usar exponentes negativos permite expresar cocientes y ratios entre las variables.
- Durbin y Rumelhart [50] demostraron que la capacidad de información de una única unidad de tipo producto (medida como la capacidad para el aprendizaje de patrones booleanos aleatorios) es aproximadamente igual a $3K$, comparado con el valor de $2K$ que corresponde a una unidad de tipo aditivo, siendo K el número de entradas de la unidad.
- Los exponentes del modelo son números reales. Esta característica tiene especial importancia, sobre todo si se tiene en cuenta que son frecuentes las situaciones en el modelado de datos reales en las que la relación entre las variables responde a una estructura de tipo potencial, donde las potencias no están restringidas a los números naturales o enteros [114].
- Permiten implementar funciones polinómicas de orden superior. Han demostrado buenos resultados en el modelado de datos en los que existen interacciones de diferentes órdenes entre las variables independientes del problema. De esta forma, cuando existen interacciones entre las variables que intervienen en un determinado fenómeno, las redes neuronales basadas en **PU** permiten obtener modelos más simplificados que las redes **MLP**.

- Junto a esto, es posible obtener cotas superiores de la dimensión de Vapnik [125] para redes basadas en PU similares a las conocidas para las redes MLP, lo que supone que poseen una capacidad de generalización similar.
- Las funciones derivadas parciales del modelo obtenido a partir de una red con PU son funciones del mismo tipo. Este hecho ayuda con frecuencia al estudio de las tasas de cambio de la variable dependiente del modelo con respecto a cada una de las variables, facilitando la interpretabilidad de los modelos.

Como contrapartida, las redes de PU presentan un importante inconveniente: la superficie de error asociada es especialmente compleja con numerosos óptimos locales y regiones planas, y por tanto existe una mayor probabilidad de que los algoritmos de búsqueda queden atrapados en alguno de los óptimos locales [52]. La estructura potencial del modelo provoca que pequeños cambios en los exponentes tengan como consecuencia un cambio significativo en los valores de la función de activación y en la función de error. Así, los algoritmos de entrenamiento de redes basados en el gradiente quedan con frecuencia y, de forma especial en este tipo de ANNs, atrapados en óptimos locales. Esta dificultad relacionada con el entrenamiento, es una de las razones por las que, a pesar de las ventajas anteriormente señaladas, las PUs han tenido poco desarrollo y han sido menos utilizadas como herramientas para resolver problemas de clasificación o regresión que otros tipos de redes [88].

3.1.3 Redes Neuronales de Función de Base Radial

Los nodos ocultos de las ANNs con estructura RBF presentan funciones de activación de tipo ventana [14, 47]. Cada uno de los nodos RBF realiza una aproximación local independiente del espacio de búsqueda, normalmente mediante una campana de Gauss. La capa de salida de tipo lineal aúna el efecto independiente de cada nodo, sumando cada valor obtenido. La idea es que cada nodo esté situado en un entorno del espacio de búsqueda formado por las variables de entrada y además con un radio determinado. El proceso de aprendizaje de la red de RBF consistirá en ir moviendo dichos nodos a lo largo del espacio de búsqueda, modificando los centros y los radios de los mismos, para ajustarlos de la mejor forma a los datos de entrenamiento.

La función de activación será equivalente a la función de distancia Euclídea (tomando como centro de la RBF el vector \mathbf{w}_m) y la función de transferencia será una función de tipo Gaussiana. Por tanto, la función de transferencia sería la siguiente:

$$\phi_m(\mathbf{x}, \mathbf{w}_m) = \exp -d_m(\mathbf{x}, \mathbf{w}_m), \quad (3.1)$$

siendo

$$d_m(\mathbf{x}, \mathbf{w}_m) = \frac{\|\mathbf{x} - \mathbf{w}_m\|^2}{r_m^2},$$

donde $\mathbf{w}_m = \{w_{m1}, w_{m2}, \dots, w_{mK}\}$ son los centros de las RBFs, r_m es un parámetro escalar definido como el radio de la neurona m y $\|\cdot\|$ es la distancia Euclídea.

Algunas de las características de las redes RBF son:

- Se ha demostrado formalmente que las redes neuronales de RBF son aproximadores universales [97].
- En comparación con las redes MLP, las redes RBF presentan la ventaja de que cada nodo en la capa oculta es un elemento local en el modelo, lo que hace que para un determinado patrón sólo algunas unidades ocultas se activen. Esta característica facilita el entrenamiento, disminuyendo el número de óptimos locales y regiones planas de la superficie de error, al desaparecer gran parte de las interacciones entre los pesos.
- Por último, el proceso de entrenamiento de una red MLP consta de una sola etapa, mientras que las redes RBF suelen entrenarse en dos etapas, siendo las funciones de base aproximadas en primer lugar, mediante técnicas de aprendizaje no supervisado, determinando los centros y el número de funciones de base mediante, por ejemplo, un algoritmo *K-means*, y en segundo lugar los pesos entre las capas oculta y de salida, mediante métodos supervisados [16].

La ecuación (3.1) representa la función de transferencia para una RBF estándar (*Standard RBF*). Las redes RBF estándar presentan una respuesta muy selectiva, con una alta activación para aquellos patrones cercanos a los centros y muy baja para los patrones distantes.

Además de las redes RBF estándar existen otros tipos de redes RBF, como:

- La RBF de Cauchy, definida por:

$$\phi_m(\mathbf{x}, \mathbf{w}_m) = \frac{1}{1 + d_m(\mathbf{x}, \mathbf{w}_m)}.$$

- La RBF inversa multicuadrática, definida por:

$$\phi_m(\mathbf{x}, \mathbf{w}_m) = \frac{1}{(1 + d_m(\mathbf{x}, \mathbf{w}_m))^{\frac{1}{2}}}.$$

La RBF de Cauchy y la RBF inversa multicuadrática tiene colas más largas que la RBF estándar, es decir, su activación para patrones distantes del centro de la RBF es mayor que la activación de la RBF estándar para estos patrones.

- Las qRBF (*q-Gaussian RBF*) representan una familia de funciones locales, en las que la cola de las diferentes funciones desempeña un papel fundamental. La qRBF se define como:

$$\phi_m(\mathbf{x}, \mathbf{w}_m) = e_{q_m}^{-d_m(\mathbf{x}, \mathbf{w}_m)},$$

donde q_m es un valor real y la función q -exponencial de $-d_m(\mathbf{x}, \mathbf{w}_m)$ viene dada por:

$$d_m(\mathbf{x}, \mathbf{w}_m) = \begin{cases} (1 - (1 - q)d_m(\mathbf{x}, \mathbf{w}_m))^{\frac{1}{1-q}} & \text{si } (1 - (1 - q)d_m(\mathbf{x}, \mathbf{w}_m)) \geq 0 \\ 0 & \text{en otro caso} \end{cases}$$

La q RBF puede reproducir diferentes RBF para distintos valores del parámetro q . Por ejemplo, cuando el parámetro q está próximo a 2, la q RBF es una RBF de Cauchy; para $q = 3$ tenemos que la activación de una unidad radial con una RBF inversa multicuadrática para $d_m(\mathbf{x}, \mathbf{w}_m)$ es igual a la activación de una unidad radial con una q RBF para $d_m(\mathbf{x}, \mathbf{w}_m)/2$; finalmente, cuando el valor de q converge a 1, la q RBF converge a la función Gaussiana (RBF estándar).

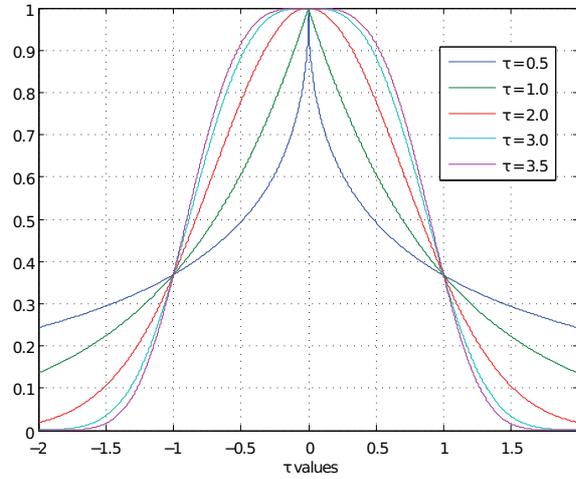


Figura 3.3: Activación de la GRBF en un espacio unidimensional con $w = 0$ y $r = 1$ para diferentes valores de τ .

- La RBF Gaussiana generalizada (GRBF), al igual que la q RBF, se basa en una parametrización de una distribución Gaussiana, la distribución Gaussiana generalizada. Esta distribución puede representar diferentes funciones de distribución según el valor del parámetro τ . Algunas de las funciones de distribución que se pueden representar son: la distribución uniforme, la Gaussiana ($\tau = 2$) o la de Laplace ($\tau = 1$). En la Figura 3.3 se muestran diferentes representaciones obtenidas con diferentes valores de τ . La GRBF se puede definir como:

$$\phi_m(\mathbf{x}, \mathbf{w}_m) = \exp -d_m(\mathbf{x}, \mathbf{w}_m),$$

siendo

$$d_m(\mathbf{x}, \mathbf{w}_m) = \frac{\|\mathbf{x} - \mathbf{w}_m\|^{\tau_j}}{r_m^{\tau_m}}.$$

 ALGORITMOS EVOLUTIVOS MULTI OBJETIVO

Frecuentemente, un número de objetivos contrapuestos tienen que ser tratados para obtener una solución viable a un problema, a menudo sin ningún tipo de conocimiento *a priori* de como los objetivos interactúan entre sí. Esta situación la encontramos en problemas del tipo: maximizar la potencia y minimizar el consumo de gasolina de un vehículo, minimizar el peso y maximizar la fortaleza de un componente, maximizar el confort y minimizar el consumo de un aparato de aire acondicionado, etc. Un problema de este tipo es considerado como un Problema de Optimización Multiobjetivo (*Multiobjective Optimization Problem*) (MOP) [26, 40].

4.1 OPTIMIZACIÓN MULTI OBJETIVO

La optimización con múltiples objetivos (llamada también optimización vectorial o de múltiples criterios) puede definirse como el problema de encontrar un vector de variables de decisión que satisfaga las restricciones y optimice una función vectorial cuyos elementos representen las funciones objetivo. Estas funciones forman una descripción matemática de los criterios de desempeño que, usualmente, están en conflicto entre sí. Por lo tanto, el término “optimizar” significa encontrar una solución tal que proporcione valores para todos los objetivos que resulten aceptables para el decisor.

De manera formal, un MOP se puede definir como la tarea de encontrar un vector

$$\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)^T,$$

que satisfaga las m restricciones de desigualdad

$$g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m, \quad (4.1)$$

las p restricciones de igualdad

$$h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, p, \quad (4.2)$$

y que optimice la función vectorial $f(\mathbf{x})$, definida según:

$$f : F \rightarrow X,$$

$$\mathbf{x} \rightarrow f(\mathbf{x}),$$

$$f(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\}^T,$$

donde \mathbf{x}^* es el vector de variables de decisión. En otras palabras, deseamos determinar, de entre el conjunto F de valores que satisfacen las restricciones 4.1 y 4.2, el conjunto $\{x_1^*, x_2^*, \dots, x_n^*\}$ que produzca los valores óptimos de todas las funciones objetivo.

Las restricciones 4.1 y 4.2 definen la región factible F , de manera que cualquier punto \mathbf{x} en F constituye una solución factible. La función vectorial $f(\mathbf{x})$ transforma el conjunto F en el conjunto X , que representa todos los valores posibles de las funciones objetivo. Las k componentes del vector $f(\mathbf{x})$ representan los criterios no conmensurables que se considerarán. Las restricciones $g_i(\mathbf{x})$ y $h_i(\mathbf{x})$ representan las limitaciones impuestas sobre las variables de decisión. El vector \mathbf{x}^* sirve para denotar las soluciones óptimas (normalmente habrá más de una, ya que los diversos objetivos pueden estar en conflicto, de forma que la optimización de uno de ellos lleve, en general, a un decremento del valor óptimo de otro).

El problema es que el significado de la palabra “óptimo” no está bien definido en este contexto, puesto que en raras ocasiones tenemos una \mathbf{x}^* tal que:

$$\forall i = 1, 2, \dots, k \exists \mathbf{x} \in F / (f_i(\mathbf{x}^*) \leq f_i(\mathbf{x})).$$

Si este fuese el caso, entonces \mathbf{x}^* sería una solución deseable, pero desgraciadamente, en la práctica casi nunca tenemos situaciones como ésta, en las que todas las $f_i(\mathbf{x})$ tengan un óptimo en F localizado en un punto común \mathbf{x}^* .

4.1.1 El Óptimo de Pareto

El concepto de óptimo de Pareto fue formulado por Vilfredo Pareto en el siglo XIX [96] y constituye por sí mismo el origen de la investigación en optimización con múltiples objetivos.

Decimos que un punto $\mathbf{x}^* \in F$ es un óptimo de Pareto si:

$$\forall \mathbf{x} \in F \exists i \in I / (f_i(\mathbf{x}) = f_i(\mathbf{x}^*)),$$

o, hay al menos una $i \in I$ (siendo I el conjunto de funciones objetivo del problema) tal que:

$$f_i(\mathbf{x}) > f_i(\mathbf{x}^*).$$

Esta definición afirma que \mathbf{x}^* es un óptimo de Pareto si no existe un vector factible \mathbf{x} que decremente algún criterio sin causar un incremento simultáneo en al menos otro de ellos. De esta definición obtenemos lo que conocemos como dominancia de

Pareto: una solución domina a otra si es mejor o igual en todos los objetivos y, al menos, estrictamente mejor en uno de ellos.

La curva (para dos objetivos) o superficie (para más de dos objetivos) que describe la compensación óptima entre los objetivos es conocida como Frente de Pareto [61], de tal manera que, una solución factible del frente no puede mejorar un objetivo sin degradar al menos uno de los restantes. Las soluciones que forman parte del Frente de Pareto se conocen con el nombre de “soluciones no dominadas”.

En la Figura 4.1 se muestra un ejemplo de frente de Pareto. Los puntos estrellados son las posibles soluciones al problema de minimización. Cualquiera de estos puntos es una solución óptima al problema, aunque no existe forma de determinar cuál de ellas es la mejor. Sin embargo, una cosa que si se puede afirmar es que son mejores que el resto de las soluciones que no pertenecen al frente.

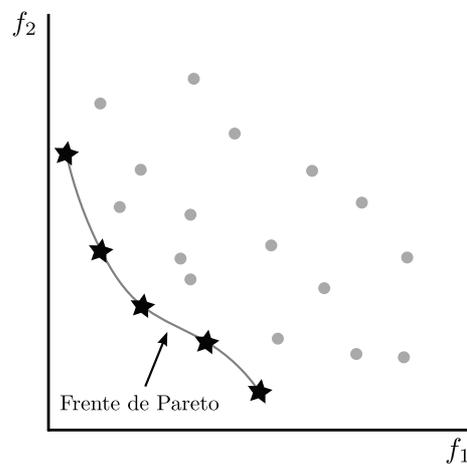


Figura 4.1: Ejemplo de frente de Pareto para un problema de minimización de dos objetivos.

4.1.2 Otras Alternativas

Existen otras alternativas, diferentes al concepto de dominancia de Pareto, empleadas para tratar los MOPs. Estas alternativas tienden a convertir un problema multiobjetivo en un problema de optimización global o en un problema con un único objetivo. Las alternativas más conocidas son:

- Métodos agregativos: Estos métodos combinan todos los objetivos del problema en una única función, la cuál será optimizada para encontrar la solución óptima al problema. El principal inconveniente de este procedimiento es la necesidad de proporcionar información escalar precisa sobre la importancia de los objetivos, para evitar que uno de ellos eclipse a los demás. Además, puede que no todos los objetivos tengan la misma importancia, por lo que debe existir

un término de regulación para cada objetivo. Este término indica cómo influye el valor de un objetivo dentro de la función de agregación. Normalmente, el ajuste de los diferentes términos de regulación se realiza mediante un proceso de prueba y error guiado por la experiencia de un experto.

$$f_{agr} = \sum_{i=1}^k w_i f_i(\mathbf{x}),$$

donde $w_i \geq 0$, $\sum_{i=1}^k w_i = 1$ y k el número de objetivos, obteniéndose una única solución en función del valor de los pesos w_i .

El autor de la presente Tesis ha realizado un estudio [29] en el que, mediante la utilización del método de ayuda en la toma de decisiones TOPSIS (*Technique for Order of Preference by Similarity to Ideal Solution*) [74], calcula cuál es el término de regulación más favorable para la agregación de los siguientes objetivos: el nivel de predicción global y el nivel de clasificación de la clase peor clasificada (estas medidas son analizadas en la Sección 5.3).

- Método de optimización *min-max* con pesos: En este método se efectúa una optimización *min-max*. Para ello se define un vector cuyos términos son las funciones objetivo multiplicadas por diferentes pesos. El MOP se convierte en la minimización con respecto al máximo de un vector \mathbf{x} , de las funciones objetivo ponderadas obtenidas con ese vector, para unos pesos dados *a priori*:

$$\min [\max_{\mathbf{x}} (w_1 \cdot f_1(\mathbf{x}), \dots, w_k \cdot f_k(\mathbf{x}))].$$

- Programación por metas: La idea principal es encontrar soluciones que logren un objetivo predefinido para una o más funciones objetivo. Si la solución al objetivo predefinido no existe, el propósito es minimizar las desviaciones de todos los objetivos. Si consideramos un valor t_i para cada una de las funciones objetivo, la mejor solución es aquella que alcance el valor t_i de cada objetivo, sujeto a que debe cumplir el resto de las restricciones del problema.
- Método de permutación ϵ : Este método reduce un MOP a un problema con un único objetivo con restricciones. Se basa en la minimización de la principal función objetivo considerando el resto de los objetivos como restricciones acotadas por ciertos niveles permisibles ϵ_i ($f_i(\mathbf{x}) \leq \epsilon_i$).
- Métodos de las métricas ponderadas: El fundamento de estos métodos se centra en la minimización de la distancia (por ejemplo, la distancia Euclídea) entre las soluciones obtenidas y un punto utópico o solución ideal.
- Método basado en soluciones óptimas lexicográficas: Este método convierte el MOP, donde todos los objetivos tienen la misma importancia, en un problema

donde los objetivos se sitúan en orden decreciente de importancia. Así, el problema de optimización lexicográfica se formula como una secuencia de k minimizaciones.

4.2 REVISIÓN DEL ESTADO DEL ARTE EN CLASIFICACIÓN MULTIOBJETIVO

Los Algoritmo Evolutivo (*Evolutionary Algorithm*) (EA) son uno de los métodos de optimización más adecuados para la resolución de problemas multiobjetivos, debido a que trabajan con un conjunto de soluciones (conocidas como población), lo que permite generar varios individuos en una única generación. Además, los EAs son menos susceptibles que otros métodos a la discontinuidad y la forma del frente de Pareto [26].

Los Algoritmos Evolutivos Multiobjetivo (MOEAs) son una extensión de los tradicionales EAs, con dos diferencias principales [25]:

- Mecanismos de selección: al trabajar con un problema multiobjetivo el propósito no es seleccionar la solución con el mejor valor de la función objetivo, sino el conjunto de soluciones no dominadas.
- Mantenimiento de la diversidad: los MOEAs necesitan un mecanismo para preservar la diversidad de las soluciones no dominadas y que evite la convergencia hacia una única solución.

Los MOEAs que emplean el concepto de óptimo de Pareto se pueden dividir, de forma general, en métodos de primera o de segunda generación. Los métodos de primera generación no son elitistas, es decir, son métodos que no disponen de mecanismos para preservar las mejores soluciones de una generación a otra. Estos métodos son fáciles de entender y sencillos de implementar, pero actualmente no suelen emplearse. Algunos de estos algoritmos son: MOGA (*Multiple Objective Genetic Algorithm*) [59], NPGA (*Niched-Pareto Genetic Algorithm*) [70] y NSGA (*Non-Dominated Sorting Genetic Algorithm*) [120].

Los métodos de la segunda generación son conocidos como algoritmos elitistas. Este tipo de algoritmos usa diversos mecanismos para preservar los mejores individuos y la diversidad dentro de la población al pasar de una generación a otra. La forma más sencilla de implementar el elitismo consiste en seleccionar los mejores individuos e incorporarlos en la población hija. El número de individuos seleccionados es importante, puesto que si es demasiado grande puede llegar a disminuir la diversidad de la población. Este fenómeno es conocido como presión selectiva. Aunque existe una amplia variedad de MOEAs en la literatura, sólo un número reducido de ellos es empleado de manera habitual. Según [25], los MOEAs más representativos son:

- SPEA (*Strength Pareto Evolutionary Algorithm*) y SPEAII (*Strength Pareto Evolutionary Algorithm 2*): Con el algoritmo SPEA (1999) [135] se inició un nuevo método de nichos para preservar la diversidad en la población sin requerir parámetros adicionales, considerando una población externa como base para el elitismo. El algoritmo SPEA2 (2001) [136] contempla algunas diferencias con respecto a su anterior versión, mejorando la precisión de la aptitud al comparar unos individuos con otros miembros de la población. Se introduce también un operador en la técnica de agrupamiento para una mejor preservación de la diversidad.
- PAES (*Pareto-Archived Evolution Strategy*): Este algoritmo se propuso en 1999 [83] como una aproximación de búsqueda local, compuesto básicamente por tres elementos: un generador de individuos candidatos, la función de aceptación del individuo candidato y un archivo de buenos individuos. La idea es chequear en los individuos el atributo de la dominancia de Pareto y el grado de diversidad del individuo mutado en comparación con el individuo actual y con el archivo de buenos individuos, antes de que dicho individuo y el archivo sean actualizados.
- μ GA (*micro Genetic Algorithm*) y μ GA2 (*micro Genetic Algorithm 2*): En los años 2001 y 2003 se han desarrollado los algoritmos llamados micro genético I y II [106], respectivamente. Son algoritmos genéticos con una población muy pequeña y un proceso de reinicialización. Se usan tres formas de elitismo y una memoria para generar la población inicial, teniendo una complejidad computacional baja. En su versión revisada se suprimen los parámetros de afinamiento y se introduce una selección dinámica a partir de la que se decide cuál es el mejor operador de cruce a utilizar.
- MOEA/D (*Multiobjective Evolutionary Algorithm Based on Decomposition*): Este algoritmo [134], propuesto en 2007, descompone el problema en un conjunto de subproblemas de optimización escalar que son simultáneamente optimizados. El algoritmo MOEA/D es un claro ejemplo de un MOEA que incorpora exitosamente conceptos de programación matemática.
- NSGAII (*Non-Dominated Sorting Genetic Algorithm 2*): El algoritmo NSGAII, presentado en 2002 [41], es una extensión del algoritmo NSGA. La principal mejora consiste en una reducción de la complejidad computacional con respecto a su antecesor, junto con la aparición de dos nuevas características: el ordenamiento rápido de individuos no dominados y la asignación de la distancia de grupo (*crowding distance*) para la preservación de la diversidad. Este algoritmo será visto detalladamente en la Sección 4.4.
- PDE (*Pareto Differential Evolution*) y MPANN (*Memetic Pareto evolutionary approach to ANNs*): Algoritmos desarrollados por H. Abbass en 2001 [1, 5]. Estos

algoritmos emplean un operador de cruce basado en evolución diferencial para la generación de nuevos individuos. Estos algoritmos serán tratados con más detalle en la Sección 4.3.1.

A estos algoritmos, habría que añadir otros como PESA (*Pareto Envelope-based Selection Algorithm*) [27], PESAI (*Pareto Envelope-based Selection Algorithm 2*) [28] o ϵ -MOEA (*epsilon MOEA*) [42].

Estos algoritmos han sido aplicados a una gran variedad de problemas de diversas áreas como el control automático para sistemas de vuelo ligero [6], la selección de características [67] o la detección de objetos [112]. El reconocimiento de imágenes es una de las áreas que más interés despierta en los autores. Por ejemplo, en [128] se detectan y clasifican rostros humanos mediante el uso del algoritmo NSGAI junto con un algoritmo de búsqueda local (esta búsqueda local es aplicada en cada generación).

4.3 EL ALGORITMO MPDENN

En esta sección se describe brevemente el concepto de la evolución diferencial (DE) para, posteriormente, comentar el principal algoritmo evolutivo multiobjetivo basado en esta técnica desarrollado por H. Abbass. Este MOEA es conocido como *Memetic Pareto evolutionary approach to Artificial Neural Networks* (MPANN). A partir de este algoritmo, hemos desarrollado un algoritmo propio llamado *Memetic Pareto Differential Evolution Neural Networks* (MPDENN), el cual será analizado al final de esta sección.

La DE fue propuesta por Storn y Price [121] como una nueva heurística para la minimización de funciones no lineales y no diferenciables en espacios totalmente ordenados. La DE es un tipo de técnica de optimización global que usa selección y cruce como sus operadores primarios para la optimización de problemas sobre dominios continuos, e incluso mutación, aunque este último operador fue introducido posteriormente al trabajo inicial de Storn y Price.

El primer paso para resolver un problema de optimización es determinar si existe un algoritmo diseñado específicamente para el problema en cuestión. Desafortunadamente, los algoritmos tradicionales deterministas son insuficientes cuando la función objetivo posee características como no linealidad, alta dimensionalidad, existencia de múltiples óptimos locales, no diferenciables o ruido. La DE es un método alternativo en la solución a problemas con estas características.

Dentro de las características fundamentales de la DE se encuentran [121]:

- La capacidad de utilizar funciones objetivo no lineales, no diferenciables y multimodales.
- El algoritmo es fácilmente paralelizable y resulta útil cuando la evaluación de la función objetivo es computacionalmente costosa.

- No se requiere predefinir distribuciones de probabilidad como en el caso de las estrategias evolutivas.
- Emplea una codificación real y la precisión está determinada por el formato de punto flotante empleado.
- Puede converger a un valor óptimo de manera consistente (dependiendo de las características del problema) a lo largo de una secuencia de ejecuciones independientes.
- En su forma original, emplea únicamente tres parámetros de control, además de un criterio de terminación o parada.

La **DE** trabaja creando una población inicial aleatoria de soluciones, donde se garantiza mediante reglas de reparación, que el valor de cada variable está dentro de unos límites, de forma tal que se verifiquen las restricciones del problema. Entonces, un individuo es elegido aleatoriamente para ser reemplazado y tres padres diferentes se seleccionan para formar un nuevo hijo. Uno de estos padres es nombrado como padre principal. Cada variable en el padre principal se modifica añadiendo un valor ponderado de la diferencia entre los dos valores de esta variable en los otros dos padres. En esencia, el vector asociado al padre principal es perturbado con el vector formado por la diferencia de los otros dos padres. Esto representa el operador de cruce en la **DE**. Si el individuo resultante es mejor que el individuo escogido para ser reemplazado, se reemplaza. En caso contrario el individuo escogido permanece en la población y el hijo se rechaza. El proceso de la **DE** se detalla más adelante en el pseudocódigo del algoritmo **MPANN**.

4.3.1 *PDE (Pareto Differential Evolution) y MPANN (Memetic Pareto evolutionary approach to ANNs)*

El algoritmo *Pareto Differential Evolution* (**PDE**) fue desarrollado por H. Abbass como un algoritmo de propósito general para la optimización multiobjetivo. La base fundamental de este algoritmo es la **DE** y el concepto de dominancia de Pareto. A partir del algoritmo **PDE**, H. Abbass implementó el algoritmo **MPANN**, introduciendo un proceso de búsqueda local a un porcentaje de individuos de la población. El algoritmo **MPANN** ha sido utilizado para minimizar el error y el número de neuronas en capa oculta de **ANN** [1, 2, 3, 5].

La principal característica del algoritmo **PDE** y de su variante **MPANN** es que ambos incluyen un operador de cruce basado en la elección aleatoria de tres padres. Este operador de cruce es el primer operador de este tipo utilizado por algoritmos evolutivos que trabajan con redes neuronales.

4.3.1.1 Representación

H. Abbass define una representación que puede ser utilizada en cualquier arquitectura sin necesidad de modificaciones. El cromosoma utilizado para representar a los individuos consta de una matriz Ω y un vector ρ . La matriz Ω es de dimensión $(I + O) \times (H + O)$, donde I es el número de variables independientes del problema (número de neuronas de la capa de entrada), O es el número de variables dependientes (número de neuronas de la capa de salida) y H es el número de neuronas en la capa oculta de la red. Cada elemento $w_{ij} \in \Omega$ es el valor del enlace que conecta la neurona i con la j , donde $i = 0, \dots, I - 1$ es la i -ésima neurona de entrada; $i = I, \dots, (I + O - 1)$ es la $(i - I)$ -ésima neurona de salida; $j = 0, \dots, H - 1$ es la j -ésima neurona oculta; y $j = H, \dots, (H + O - 1)$ es la $(j - H)$ -ésima neurona de salida. Esta representación utilizada por H. Abbass tiene dos características adicionales que no son empleadas por los algoritmos PDE y MPANN:

1. Permite la conexión directa de cada neurona de entrada con cada una de las neuronas de salida.
2. Permite conexiones entre neuronas de la capa de salida.

El vector ρ es de dimensión H , donde $\rho_h \in \rho$ es un valor binario que sirve para indicar si la neurona oculta h -ésima existe o no en la red (1=existe; 0=no existe).

4.3.1.2 El Algoritmo MPANN

En esta sección se detallará cómo funciona el algoritmo MPANN (recuérdese que el algoritmo PDE trabaja de manera similar al algoritmo MPANN, eliminando el proceso de búsqueda local). En la Figura 4.2 se puede ver el pseudocódigo del algoritmo MPANN.

Durante la ejecución del algoritmo, suponemos que todas las variables están definidas en el intervalo $[0, 1]$.

El algoritmo comienza con la generación aleatoria de una población inicial de soluciones (ANNs en este caso). Los valores de los pesos de estas soluciones se establecen siguiendo una distribución uniforme $[0, 1]$ (línea 1). Acto seguido, el algoritmo comienza a evolucionar la población a través de las generaciones.

La primera acción que tiene lugar al comienzo de una generación es la evaluación de los individuos (línea 4). La evaluación consiste en calcular los valores de las funciones objetivo para cada uno de los individuos de la población. Una vez evaluados, se separan los individuos dominados de los no dominados (línea 5). Si el grupo de los individuos no dominados tiene menos de 3 componentes, se busca un individuo no dominado dentro del grupo de los dominados y se añade al grupo. Esto se repite hasta que el grupo de los no dominados tenga 3 componentes (líneas 6-8). Una vez

```

1: Inicializar  $P_0$ 
2:  $k \leftarrow 0$ 
3: while no se cumpla la condición de parada do
4:   Evaluar  $P_k$ 
5:   Copiar las soluciones no dominadas de  $P_k$  a  $P_{k+1}$ 
6:   while tamaño( $P_{k+1}$ ) < 3 do
7:     Copiar una solución dominada de  $P_k$  a  $P_{k+1}$ 
8:   end while
9:   Marcar un 20% del conjunto de entrenamiento para validación
10:  while la población no esté completa do
11:    Seleccionar aleatoriamente  $\alpha_1, \alpha_2, \alpha_3 \in P_{k+1}$  donde  $\alpha_1 \neq \alpha_2 \neq \alpha_3$ 
12:    for all neurona do
13:       $pro \leftarrow$  Probabilidad Uniforme  $[0, 1]$ 
14:      if  $pro < pro\_cruce$  then
15:        Aplicar (4.3) y (4.4) o (4.7) según la capa a la que pertenezca la neurona
16:      else
17:        Aplicar (4.5) y (4.6) o (4.8) según la capa a la que pertenezca la neurona
18:      end if
19:    end for
20:    for all neurona do
21:       $pro \leftarrow$  Probabilidad Uniforme  $[0, 1]$ 
22:      if  $pro < pro\_mut$  then
23:        Aplicar (4.9) y (4.10) o (4.11) según la capa a la que pertenezca la neurona
24:      end if
25:    end for
26:    Aplicar la búsqueda local al hijo generado
27:    Evaluar al hijo
28:    if el hijo domina al padre principal then
29:      Añadir el hijo a  $P_{k+1}$ 
30:    end if
31:  end while
32:   $k \leftarrow k + 1$ 
33: end while

```

Figura 4.2: Pseudocódigo del algoritmo MPANN.

formado el grupo de los individuos no dominados, se eliminan todas las soluciones del grupo de dominados.

A continuación, se selecciona al azar un 20% del conjunto de patrones de entrenamiento como conjunto de validación (línea 9). Este conjunto de validación es usado para verificar el procedimiento de búsqueda local. En cada generación del proceso evolutivo se selecciona un conjunto de validación diferente con el objetivo de evitar que los individuos de la población se sobreajusten al conjunto de entrenamiento.

El siguiente paso es el más importante de todo el algoritmo, ya que se trata de la generación de un nuevo hijo. Este paso se repetirá, durante cada una de las generaciones, hasta que la población alcance el tamaño máximo fijado para el problema. La primera acción a realizar es la selección aleatoria de tres padres (línea 11). Uno

de ellos será etiquetado como padre principal (α_1) y los otros dos como secundarios (α_2, α_3). A continuación tiene lugar la aplicación del operador de cruce, seguido del operador de mutación.

En la operación de cruce (líneas 12-19), se obtiene un valor aleatorio de una distribución uniforme $[0, 1]$ para cada una de las neuronas de la capa oculta. Si el valor es inferior a una probabilidad de cruce prefijada (`pro_cruce`), se aplican las operaciones mostradas en (4.3) y (4.4).

$$\rho_h^{\text{hijo}} \leftarrow \begin{cases} 1 & \text{si } (\rho_h^{\alpha_1}) + N(0, 1) (\rho_h^{\alpha_2} - \rho_h^{\alpha_3}) \geq 0,5 \\ 0 & \text{en otro caso} \end{cases} . \quad (4.3)$$

$$w_{ih}^{\text{hijo}} \leftarrow w_{ih}^{\alpha_1} + N(0, 1) (w_{ih}^{\alpha_2} - w_{ih}^{\alpha_3}) . \quad (4.4)$$

En caso de que no se cumpla que el número obtenido sea menor que `pro_cruce`, se aplicarán las operaciones (4.5) y (4.6).

$$\rho_h^{\text{hijo}} \leftarrow \rho_h^{\alpha_1} . \quad (4.5)$$

$$w_{ih}^{\text{hijo}} \leftarrow w_{ih}^{\alpha_1} . \quad (4.6)$$

Una vez terminada la operación de cruce asociada a todas las neuronas de la capa oculta, es el turno de las neuronas de la capa de salida. Se obtiene un valor aleatorio de una distribución uniforme $[0, 1]$ y si se cumple que el valor es inferior a una probabilidad de cruce prefijada (`pro_cruce`), se aplica la operación (4.7).

$$w_{ho}^{\text{hijo}} \leftarrow w_{ho}^{\alpha_1} + N(0, 1) (w_{ho}^{\alpha_2} - w_{ho}^{\alpha_3}) . \quad (4.7)$$

En caso de que no se cumpla que el número obtenido sea menor que `pro_cruce`, se aplica la operación (4.8).

$$w_{ho}^{\text{hijo}} \leftarrow w_{ho}^{\alpha_1} . \quad (4.8)$$

Durante la operación de cruce, al menos un peso de la estructura de la red hijo se debe modificar para que se obtenga un modelo de red distinto al del padre principal.

En la operación de mutación (líneas 20-25), se obtiene un valor aleatorio de una distribución uniforme $[0, 1]$ para cada una de las neuronas del hijo. Si se cumple que el valor es inferior a una probabilidad de mutación prefijada (`pro_mut`), se aplican las operaciones (4.9) y (4.10) si la neurona es de la capa oculta y la operación (4.11) si la neurona es de la capa de salida. Si el valor aleatorio obtenido es menor que `pro_mut`, no se hace nada.

$$\rho_h^{\text{hijo}} \leftarrow \begin{cases} 1 & \text{si } \rho_h^{\text{hijo}} = 0 \\ 0 & \text{en otro caso} \end{cases} . \quad (4.9)$$

$$w_{ih}^{\text{hijo}} \leftarrow w_{ih}^{\text{hijo}} + N(0, \text{RadioMut}). \quad (4.10)$$

$$w_{ho}^{\text{hijo}} \leftarrow w_{ho}^{\text{hijo}} + N(0, \text{RadioMut}). \quad (4.11)$$

El parámetro `RadioMut` toma valores entre 0 y 1, obteniendo mejores resultados cuando su valor está comprendido entre 0 y 0,1, como se indica en [2].

Una vez realizadas las operaciones de cruce y mutación, se aplica la búsqueda local al hijo resultante (línea 26). Este hijo es evaluado y será añadido a la población si presenta una relación de dominancia con respecto al padre principal (líneas 27-30).

Se generarán tantos hijos como sean necesarios hasta que la población alcance su tamaño máximo. Cuando esto ocurra, se dará por finalizada la generación. Este proceso se repetirá a lo largo de las generaciones hasta que se cumpla la condición de parada y concluya la ejecución del algoritmo.

4.3.2 El Algoritmo MPDENN (*Memetic Pareto Differential Evolution Neural Network*)

A continuación se detallará el algoritmo `MPDENN`. Este algoritmo sigue la estructura básica del algoritmo `PDE` propuesto por H. Abbass [1, 2, 3, 5], pero con algunas modificaciones para adaptarlo a los requerimientos de nuestro entorno de trabajo [66] y para resolver los problemas de estancamiento del algoritmo original. La Figura 4.3 muestra el pseudocódigo del algoritmo `MPDENN` y la Figura 4.4 presenta el flujo de la información.

El primer paso del algoritmo consiste en la generación de una población inicial al azar. Los pesos de los individuos pertenecientes a esta población se establecen siguiendo una distribución uniforme $[0, 1]$ (línea 1). Acto seguido, el algoritmo comienza a evolucionar la población a través de las generaciones.

La primera acción que tiene lugar al comienzo de una generación es la evaluación de los individuos (línea 5). Esta evaluación consiste en calcular las funciones objetivo para cada uno de los individuos de la población. Una vez evaluados, se separan los individuos dominados de los no dominados (línea 6). Si el grupo de los individuos no dominados tiene menos de 3 componentes, se busca un individuo no dominado dentro del grupo de los dominados y se añade al grupo. Esto se repite hasta que el grupo de los no dominados tenga 3 componentes (líneas 7-9).

Si por el contrario, el grupo de individuos no dominados supera la mitad del tamaño máximo de la población ($M/2$), será necesario reducir el tamaño de este grupo. Para ello, se calcula la distancia de cada individuo con sus dos vecinos más cercanos y se elimina del grupo aquel individuo con menor distancia (líneas 10-15). El cálculo de la distancia viene dado por:

$$D(x) = \frac{(\text{mín } \|x - x_i\| + \text{mín } \|x - x_j\|)}{2},$$

```

1: Inicializar  $P_0$ 
2:  $k \leftarrow 0$ 
3:  $creados \leftarrow 0$ 
4: while no se cumpla la condición de parada do
5:   Evaluar  $P_k$ 
6:   Copiar las soluciones no dominadas de  $P_k$  a  $P_{k+1}$ 
7:   while tamaño( $P_{k+1}$ ) < 3 do
8:     Copiar una solución dominada de  $P_k$  a  $P_{k+1}$ 
9:   end while
10:  while tamaño( $P_{k+1}$ ) >  $M/2$  do
11:    for all individuo  $\in P_{k+1}$  do
12:      Calcular la distancia a los vecinos más cercanos
13:    end for
14:    Eliminar de  $P_{k+1}$  al individuo con menor distancia
15:  end while
16:  while la población no esté completa do
17:    Seleccionar aleatoriamente  $\alpha_1, \alpha_2, \alpha_3 \in P_{k+1}$  donde  $\alpha_1 \neq \alpha_2 \neq \alpha_3$ 
18:    for all neurona do
19:       $pro \leftarrow$  Probabilidad Uniforme  $[0, 1]$ 
20:      if  $pro < pro\_cruce$  then
21:        Aplicar (4.3) y (4.4) o (4.7) según la capa a la que pertenezca la neurona
22:      else
23:        Aplicar (4.5) y (4.6) o (4.8) según la capa a la que pertenezca la neurona
24:      end if
25:    end for
26:    if hijo = padre principal then
27:      Seleccionar aleatoriamente un peso del hijo ( $peso$ )
28:       $peso \leftarrow peso + N(0, 1)$ 
29:    end if
30:    for all neurona oculta do
31:       $pro \leftarrow$  Probabilidad Uniforme  $[0, 1]$ 
32:      if  $pro < pro\_mut$  then
33:        Aplicar (4.9)
34:      end if
35:    end for
36:    Evaluar al hijo
37:     $creados \leftarrow creados + 1$ 
38:    if el hijo domina al padre principal then
39:      Añadir el hijo a  $P_{k+1}$ 
40:       $creados \leftarrow 0$ 
41:    else if no hay relación de dominancia entre el hijo y el padre principal then
42:      Añadir el hijo a  $P_{k+1}$ 
43:       $creados \leftarrow 0$ 
44:    else if  $creados = M$  then
45:      Añadir el mejor hijo rechazado a  $P_{k+1}$ 
46:       $creados \leftarrow 0$ 
47:    else
48:      Rechazar el hijo
49:    end if
50:  end while
51:   $k \leftarrow k + 1$ 
52: end while

```

Figura 4.3: Pseudocódigo del algoritmo MPDENN.

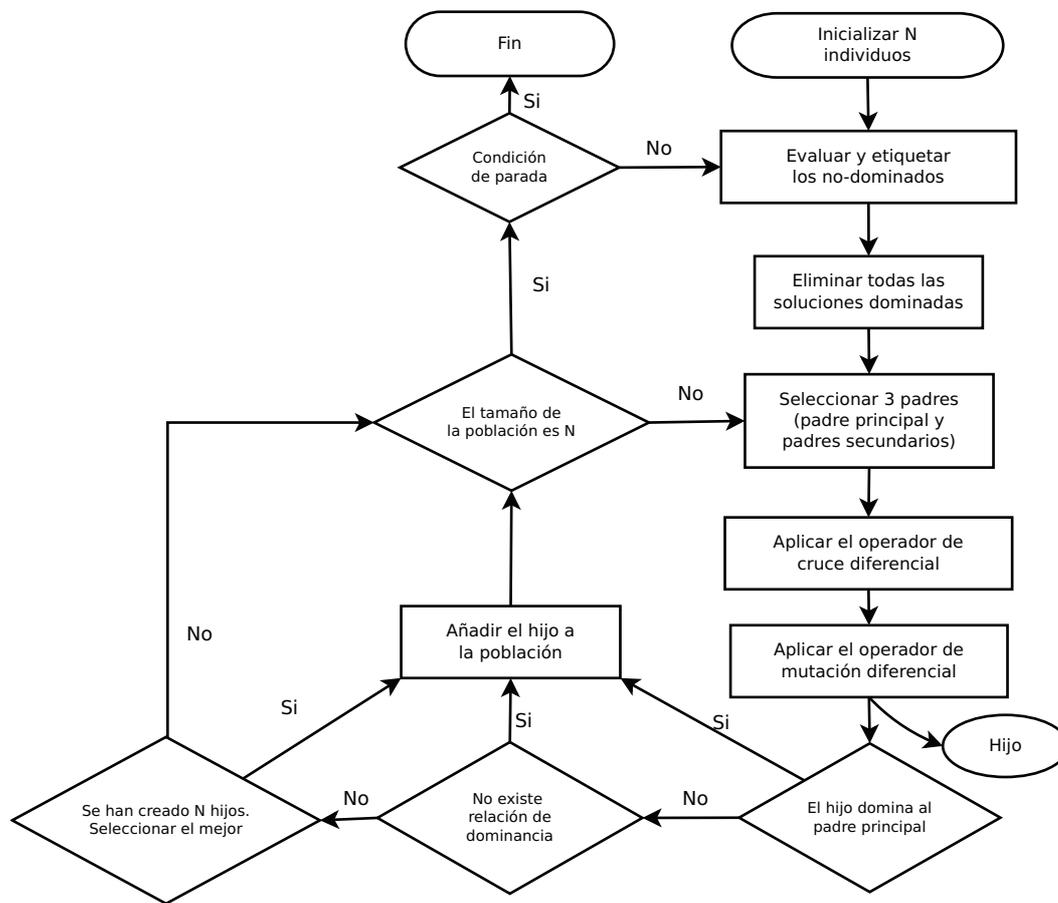


Figura 4.4: Flujo de información del algoritmo MPDENN.

siendo x el individuo sobre el que se va a calcular la distancia a sus dos vecinos más cercanos (x_i y x_j , para $i \neq j$). Esto nos permite mantener un mayor grado de diversidad y favorecer que el algoritmo no quede estancado en el caso de que el número de soluciones no dominadas sea cercano a M , puesto que el tamaño de la población suele ser pequeño (en aras de un menor coste computacional), con lo que obtendríamos de una generación a otra muy pocos individuos nuevos.

Una vez formado el grupo de los individuos no dominados, se eliminan todas las soluciones del grupo de dominados.

A continuación se completa la población actual. Primero se seleccionan aleatoriamente tres individuos diferentes de la población (línea 17). El primero de ellos será considerado como padre principal y los otros dos como padres secundarios. A partir de esos tres individuos, se realizan una serie de operaciones para obtener un nuevo hijo que será añadido a la población. En primer lugar, se aplica el operador de cruce sobre el padre principal y los padres secundarios. Se genera un valor aleatorio de una

distribución uniforme $[0, 1]$ para cada una de las neuronas de la red (líneas 18-25), de manera que si el valor obtenido es menor que `pro_cruce`, se aplican las operaciones (4.3) y (4.4) o (4.7) según si la neurona pertenece a la capa oculta o a la capa de salida, respectivamente. Si el valor obtenido es mayor que `pro_cruce`, se aplican las operaciones (4.5) y (4.6) si la neurona pertenece a la capa oculta o (4.8) si pertenece a la capa de salida.

Si el hijo obtenido es igual al padre principal, porque no se haya producido ningún cambio con las operaciones realizadas anteriormente, se modifica aleatoriamente un peso de su estructura, añadiéndole un valor de una distribución Gaussiana $N(0, 1)$ (líneas 26-29).

A continuación, se aplica el operador de mutación a cada una de las neuronas de la capa oculta del hijo creado. Se genera un valor aleatorio de una distribución uniforme $[0, 1]$ (líneas 30-35) y si se cumple que el número obtenido es menor que `pro_mut`, se aplica la fórmula (4.11), la cual elimina la neurona en caso de que exista, o la crea y la añade al modelo en el caso de que no existiera, estableciendo los pesos de los enlaces de manera aleatoria. En caso de que no se cumpla que el número obtenido sea menor que `pro_mut`, no se hace nada con esa neurona. El operador de mutación ha cambiado con respecto al utilizado por el algoritmo MPANN debido a que usamos una representación vectorial de la red. En el algoritmo MPANN, al utilizar una representación matricial, se pueden recuperar los pesos de las neuronas previamente eliminadas y modificar estos valores con un valor de una distribución Gaussiana $N(0, \text{RadioMut})$. En cambio, nuestro operador de mutación no puede realizar esta operación, por lo que decidimos modificarlo de la manera expuesta.

Una vez realizadas las operaciones de cruce y mutación, se evalúa el hijo resultante (línea 36). Una vez evaluado el nuevo hijo, se incrementa en 1 el valor de la variable `creados`, la cual será usada más adelante en el proceso evolutivo. Concretamente, cuando se hayan creado muchos hijos y ninguno de ellos se pueda añadir a la población.

El nuevo hijo se añade a la población si:

- el hijo domina al padre principal (línea 38).
- no existe relación de dominancia entre el padre principal y el hijo (línea 41).
- el padre principal domina al hijo y `creados = M` (M es el tamaño máximo de la población). En este caso se añade a la población el mejor de los M hijos rechazados (línea 44).
- si no se verifica ninguna de las condiciones anteriores, el hijo se rechaza y no se añade a la población (línea 48).

Siempre que se añade un nuevo individuo a la población, la variable `creados` se iguala a 0 para reiniciar la cuenta.

Se generarán tantos hijos como sean necesarios hasta que la población alcance su tamaño máximo. Cuando esto ocurra, daremos por finalizada una generación. Este proceso se repetirá a lo largo de las generaciones hasta que se cumpla la condición de parada.

En el pseudocódigo del algoritmo **MPDENN** no se ha incluido la búsqueda local, debido a que se han considerado diferentes alternativas, las cuales se verán a continuación.

4.3.2.1 Hibridación

Los **EAs** son suficientemente eficientes explorando el espacio de búsqueda en un problema de optimización, sin embargo, está probada su ineficiencia para encontrar el óptimo global dentro de la región del espacio en la que el algoritmo converge. Numerosos investigadores han demostrado que los **EAs** son una herramienta adecuada para la búsqueda global, ya que son capaces de hallar rápidamente regiones del espacio de búsqueda en las que se encuentran los óptimos locales del problema o bien la solución global, mientras que la convergencia hacia el óptimo correspondiente es con frecuencia muy lenta [73, 78]. Por otro lado, los algoritmos de búsqueda local pueden quedar atrapados en un óptimo local, aunque en ocasiones pueden encontrar el óptimo global cuando la búsqueda se lleva a cabo en pequeñas regiones del espacio. De esta forma, en combinación con los **EAs**, los algoritmos de búsqueda local pueden proporcionar un método para obtener de manera rápida y eficiente buenas soluciones a un problema [116].

Según la forma en la que se realice la combinación entre el **EA** y la búsqueda local, podemos hablar de algoritmo híbrido o de Algoritmo Memético (*Memetic Algorithm*) (**MA**). En los algoritmos híbridos se aplican la optimización local a todos los individuos de la población del **EA** y se almacenan como posibles soluciones. Los **MA**s [94, 95], un caso particular de algoritmo híbrido, son algoritmos evolutivos que incluyen dentro de la estrategia de búsqueda un proceso de optimización o aprendizaje de cada individuo, es decir, se selecciona un individuo de la población, se le aplica la optimización local y se devuelve a la población.

4.3.2.2 *iRprop*⁺ como Algoritmo de Búsqueda Local

El algoritmo de búsqueda local empleado en la presente Tesis es el *iRprop*⁺ (*Improved Resilient Back-propagation*) [76], que es una mejora del algoritmo *Rprop* (*Resilient Back-propagation*) [110]. El algoritmo *Rprop*, como método de búsqueda local, es una de las mejores técnicas conocidas en términos de velocidad de convergencia, precisión y robustez con respecto a sus parámetros. El algoritmo *iRprop*⁺ es un procedimiento basado en gradiente, difiriendo de las técnicas clásicas de propagación hacia atrás del error en que, las derivadas parciales de la función error, sólo son usadas para determinar el sentido en que deben ser corregidos los pesos de la red, pero no las

magnitudes de los ajustes. El modelo de actualización de cada peso w_{ij} de una ANN viene dado por $w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t+1)}$, donde $\Delta w_{ij}^{(t+1)}$ se estima con una función de cambio de signo de la derivada del error entre las iteraciones (t) y $(t-1)$, y en función del tamaño de paso para los pesos Δ_{ij} , tal como se realiza tradicionalmente con las técnicas basadas en gradiente. Así, si el signo de la derivada no cambia en las dos últimas iteraciones, el tamaño de paso es incrementado en un factor η^+ pero limitado a que su valor máximo no supere Δ_{max} , que corresponde al tamaño máximo de la modificación de w_{ij} . Si la derivada es cero no se modifica y si cambia de signo, el tamaño de paso es reducido en un factor η^- pero limitando el tamaño mínimo de modificación de w_{ij} a un valor Δ_{min} . El método *iRprop+* aplica una estrategia de *backtracking* para decidir la dirección de corrección del valor de los pesos de la red, concretamente la idea es que la actualización de los pesos dependa de la evolución del error. De esta manera, el esquema de entrenamiento combina información local con información global (por ejemplo, el valor del error en cada iteración) a la hora de decidir si se debe modificar los pesos. Un cambio de signo en una derivada parcial significa que se ha saltado un mínimo local. Se ha demostrado mediante problemas de prueba o *benchmarks* [75], que el algoritmo *iRprop+* consigue mayor rendimiento que su versión original.

La función de error que será optimizada por el algoritmos *iRprop+* es la entropía cruzada (más información en la Sección 5.2). La elección de esta métrica se debe a que representa el nivel de clasificación global del modelo y que, al ser una función de error, es una medida continua, lo que facilita la convergencia en la búsqueda.

4.3.2.3 Algoritmos Meméticos Desarrollados

Una vez descrito el método de búsqueda local que vamos a emplear, se detallarán los MAs desarrollados. Todos ellos siguen la misma estructura que el algoritmo MPDENN, descrita en la Figura 4.3. Los aspectos que diferencian a los MAs entre sí, son el momento de la evolución dónde se aplica la búsqueda local y qué individuos son optimizados.

Una de las ventajas de los MAs desarrollados con respecto al algoritmo MPANN es la reducción de la carga computacional y, por consiguiente, un tiempo de ejecución menor.

- MPDENN1: En este MA únicamente se aplica el algoritmo de búsqueda local a aquellos hijos que presenten una relación de dominancia con respecto al padre principal. De esta forma solo se aplica la búsqueda local a los hijos que sabemos que serán incluidos en la población. El proceso de optimización tiene lugar en todas las generaciones.
- MPDENN2: En este MA cambiamos de filosofía, ya que ahora no se aplica la búsqueda local a los hijos antes de añadirlos a la población, sino que se aplica

a un grupo de individuos seleccionados de la población, una vez que ésta se ha completado. La búsqueda local se aplicará a un 10% de individuos de la población, los cuales serán seleccionados de manera aleatoria. Además, el proceso de optimización sólo tendrá lugar en ciertas generaciones a lo largo de la ejecución (así, se aplica sólo al principio (1/3 del número total de generaciones), en la mitad del proceso evolutivo (2/3) y al final del mismo (3/3)). Este esquema puede verse en la Figura 4.5. El resto de las generaciones se ejecutan siguiendo el esquema del algoritmo MPDENN.

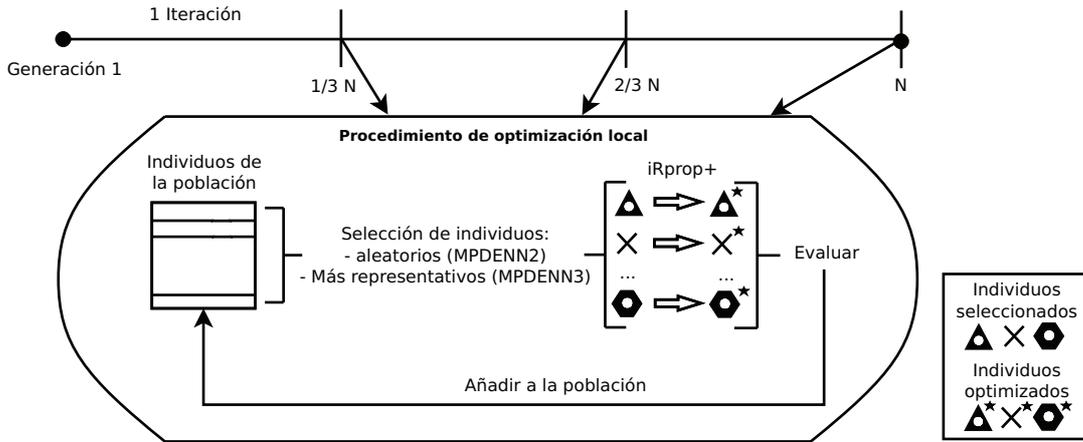


Figura 4.5: Esquema del procedimiento de búsqueda local seguidos por los algoritmos MPDENN2 y MPDENN3.

- MPDENN3: En este caso, la búsqueda local sólo se realiza durante ciertas generaciones a lo largo de la ejecución una vez que la población está completa. Generalmente la búsqueda local se aplica en tres generaciones, de igual manera que el algoritmo MPDENN2.

En el algoritmo MPDENN3 introducimos el concepto de *clustering* o agrupamiento, de manera que la búsqueda local sólo se realiza a los individuos más representativos del frente de Pareto. El procedimiento para seleccionar estos individuos es el siguiente:

- Si el número de individuos del primer frente de Pareto es menor o igual que el número de individuos representativos deseados (num), se les aplica la búsqueda local a los individuos del primer frente sin necesidad de aplicar el algoritmo *K-means* [90].
- Si el número de individuos del primer frente es mayor que num , se aplica el algoritmo *K-means* al primer frente para obtener los num individuos más representativos (aquellos más cercanos a los centroides de los clúster obtenidos), a los cuales se les aplicará la búsqueda local.

4.3.3 Diferencias entre los Algoritmos MPANN y MPDENN

A continuación, se presenta la justificación de las principales modificaciones que presenta el algoritmo **MPDENN** con respecto al algoritmo **MPANN**:

- Se establece un número máximo de individuos no dominados. Esta modificación está justificada por el hecho de que si la totalidad, o casi la totalidad, de la población está formada por individuos no dominados, en cada generación, prácticamente, no se crearán hijos nuevos. En un caso extremo, si el tamaño de la población es M y en la población existen M individuos no dominados, el algoritmo **MPANN** comenzará a pasar generaciones sin crear nuevos hijos hasta la finalización de la ejecución. Para evitar esta situación indeseada, en el algoritmo **MPDENN** se ha llegado a la solución de compromiso de eliminar los individuos no dominados hasta quedarnos con $M/2$. Esta modificación se puede apreciar en las líneas 10-15 del pseudocódigo expuesto en la Figura 4.3.

El método utilizado para determinar que individuos serán eliminados de la población es el de la distancia a los dos vecinos más cercanos. En este método se calcula la distancia de cada individuo a sus dos vecinos más próximos, y después se elimina el individuo con la distancia más pequeña. Se ha utilizado este método porque nos permite reducir el tamaño de la población de individuos no dominados sin una pérdida significativa de diversidad.

- En el algoritmo **MPANN**, existe la restricción de que en el operador de cruce, al menos, el valor de un enlace debe ser modificado. Sin embargo, H. Abbass no especifica qué hacer en el caso de que no se produzca ningún cambio. Por tanto, se ha decidido que en el caso de no existir ningún cambio, se seleccionará un enlace al azar de la red (el enlace puede estar entre la capa de entrada y la oculta, o entre la capa oculta y la de salida) y se modificará añadiendo al valor de su peso una variable aleatoria perteneciente a una distribución Gaussiana $N(0, 1)$. Esta modificación se puede apreciar en las líneas 26-29 del pseudocódigo.
- El operador de mutación ha sido modificado sustancialmente, por causa de la implementación utilizada. H. Abbass utiliza una representación matricial, por lo que siempre conserva la información de todas las neuronas, existan o no. Para determinar que neuronas existen y cuáles no, utiliza un vector binario (un 0 en la posición i significa que la neurona i -ésima no existe y un 1 significa que existe). En cambio, en el algoritmo **MPDENN** se utiliza una representación vectorial, por lo que no se puede mantener la información de las neuronas que no existen. Es por esto que no se pueden modificar los pesos utilizando el parámetro `RadioMut`, puesto que si la neurona existiese se eliminaría (por lo que no tendría sentido modificar sus pesos) y se suprimiría del vector. En el caso de que la neurona no existiera, se crearía una nueva con pesos aleatorios (por lo que tampoco tendría sentido modificar el valor de sus enlaces).

- La condición para añadir hijos a la población en el algoritmo **MPANN** es muy restrictiva, ya que puede darse el caso de que los hijos nunca dominen al padre principal. En un caso extremo, el algoritmo caería en un bucle infinito en el que se generarían hijos no válidos y nunca concluiría la ejecución. Para evitar esta situación indeseada se ha introducido en el algoritmo **MPDENN** la siguiente solución: si el hijo domina al padre principal o si no tiene relación de dominancia con él se añade a la población. En el caso en el que el padre principal domina al hijo, éste último se rechaza, pero se almacena el mejor de todos los hijos rechazados. Después de rechazar M hijos, se añadiría a la población el hijo almacenado. Estas modificaciones se pueden apreciar en las líneas 38-49.
- Todas las estrategias empleadas para seleccionar los individuos que serán objeto de la búsqueda local son menos costosas, computacionalmente hablando, que la estrategia presentada por H. Abbass. Esto se debe a que en el algoritmo **MPANN** se aplica la búsqueda local a todos los individuos creados, vayan a formar parte de la población o no. Sin embargo, en todas las variantes del algoritmo **MPDENN** sólo se aplica la búsqueda local a individuos que ya forman parte de la población.

4.4 EL ALGORITMO MPENSGAII

Otro algoritmo muy utilizado en la literatura es el *Non-dominated Sorting Genetic Algorithm 2* (**NSGAII**) desarrollado por K. Deb *et al.* [41]. Las principales características de este algoritmo son un procedimiento de ordenamiento rápido de soluciones no dominadas para obtener los frentes de Pareto y el cálculo de la distancia *crowding* para favorecer y mantener la diversidad en la población. El algoritmo **NSGAII** ha sido adaptado para poder trabajar con redes neuronales por J.C. Fernández *et al.* [55]. Esta versión del algoritmo **NSGAII** es conocida como *Memetic Pareto NSGAII* (**MPENSGAII**) y, además de adaptar el algoritmo original para entrenar **ANN**, también incorpora un procedimiento de búsqueda local. Este procedimiento de búsqueda local se aplicará, en tres momentos a lo largo de la evolución, a los individuos pertenecientes al primer frente de Pareto. La Figura 4.6 muestra el pseudocódigo del algoritmo **MPENSGAII** y la Figura 4.7 presenta el flujo de información. El pseudocódigo del algoritmo **NSGAII** no ha sido incluido debido a que es similar al del algoritmo **MPENSGAII** (excluyendo el procedimiento de búsqueda local).

El algoritmo comienza con la generación de la población inicial P_k de tamaño M (línea 2). Una vez generada, la población inicial es evaluada (línea 3). A continuación, se aplica el ordenamiento rápido de individuos no dominados. Este procedimiento asigna a cada individuo un valor numérico que depende del frente de Pareto al que pertenece (1 a los individuos del primer frente, 2 a los del segundo, ...) (línea 4).

```

1:  $k \leftarrow 0$ 
2: inicializar( $P_k$ )
3: evaluar( $P_k$ )
4:  $F \leftarrow \text{ordenamiento\_rapido}(P_k)$ 
5: distancia_crowding( $F$ )
6:  $Q_k \leftarrow \emptyset$ 
7: for  $i = 1 \rightarrow M$  do
8:    $q \leftarrow \text{torneo\_binario}(P_k)$ 
9:   mutar( $q$ )
10:   $Q_k \leftarrow Q_k \cup q$ 
11:   $i \leftarrow i + 1$ 
12: end for
13: evaluar( $Q_k$ )
14: while no se cumpla la condición de parada do
15:   $R_k \leftarrow P_k \cup Q_k$ 
16:  if  $k$  es igual a  $2/7, 4/7$  o  $6/7$  del total de generaciones then
17:     $F \leftarrow \text{ordenamiento\_rapido}(R_k)$ 
18:     $F^1 \leftarrow \text{busqueda\_local}(F^1)$ 
19:    evaluar( $F^1$ )
20:    Reemplazar  $F^1$  por  $F^1$  en  $R_k$ 
21:    evaluar( $R_k$ )
22:  end if
23:   $F \leftarrow \text{ordenamiento\_rapido}(R_k)$ 
24:   $f \leftarrow 1$ 
25:   $P_{k+1} \leftarrow \emptyset$ 
26:  while tamaño  $P_{k+1} < M$  do
27:    distancia_crowding( $F^f$ )
28:     $P_{k+1} \leftarrow P_{k+1} \cup F^f$ 
29:     $f \leftarrow f + 1$ 
30:  end while
31:  Ordenar  $P_{k+1}$  y seleccionar los  $M$  primeros individuos
32:   $Q_{k+1} \leftarrow \emptyset$ 
33:  for  $i = 1 \rightarrow M$  do
34:     $q \leftarrow \text{torneo\_binario}(P_{k+1})$ 
35:    mutar( $q$ )
36:     $Q_{k+1} \leftarrow Q_{k+1} \cup q$ 
37:     $i \leftarrow i + 1$ 
38:  end for
39:  evaluar( $Q_{k+1}$ )
40:   $k \leftarrow k + 1$ 
41: end while

```

Figura 4.6: Pseudocódigo del algoritmo MPENSGAII.

Además, se asigna el valor de la distancia *crowding* a cada individuo (línea 5). Estos dos procedimientos se detallarán en la Subsección 4.4.1.

El siguiente paso consiste en generar una población auxiliar Q_k que contendrá los nuevos individuos generados. Estos individuos son el resultado de aplicar los operadores de mutación (ver Subsección 4.4.1) a un individuo de P_k seleccionado mediante un torneo binario. En un torneo binario, se eligen dos individuos de la población al azar y, tras comparar sus distancias *crowding* y los frentes a los que pertenecen, el mejor de ellos es seleccionado. Finalmente, la población Q_k es evaluada (líneas 6-13).

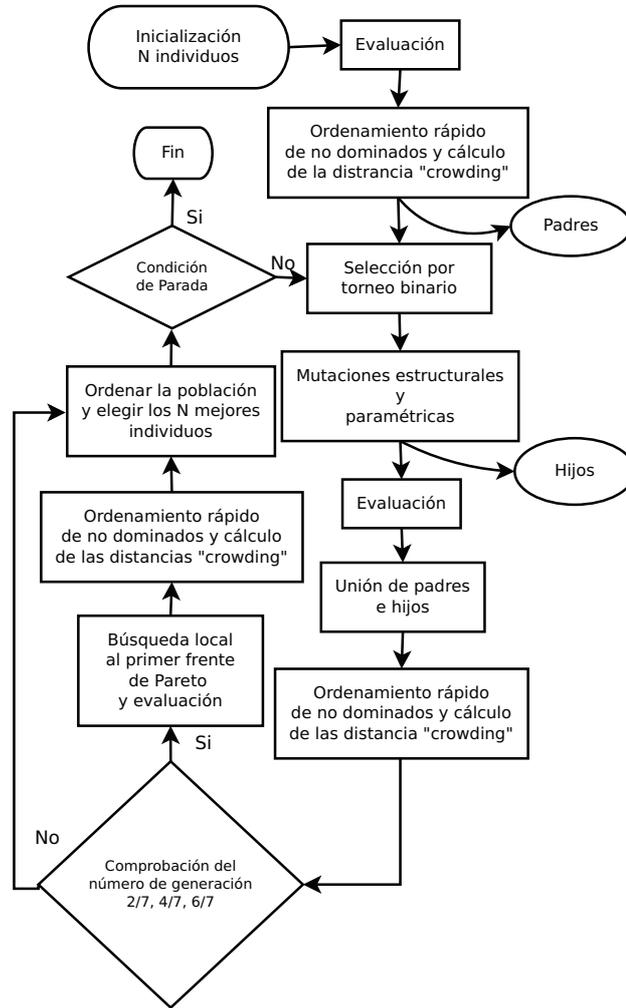


Figura 4.7: Flujo de información del algoritmo MPENSGAIL.

En este momento comienza el proceso evolutivo. Los individuos de la población inicial junto con los de la nueva población generada se ordenan según el frente al que pertenezcan. A continuación, se crea la población P_{k+1} añadiendo los individuos de la lista de frentes obtenidos y en base a las distancias *crowding*, hasta que el tamaño de P_{k+1} sea igual o mayor que M . Cuando la población P_{k+1} esté completa, se ordenan en base al orden del frente y a la distancia *crowding* y se seleccionan los M primeros individuos (líneas 15, 23-31).

Una vez generada la población P_{k+1} , se genera la población auxiliar Q_{k+1} de manera similar a como se generó la población Q_k (líneas 32-39).

El procedimiento de búsqueda local se aplica a todos los individuos del primer frente de Pareto en tres momentos de la evolución: una vez que se inicializó la población y ha comenzado la evolución, en un punto intermedio del proceso evolutivo.

vo y antes del final de la ejecución del algoritmo. Estos momentos se corresponden, respectivamente, con los 2/7, 4/7 o 6/7 del total de generaciones del proceso evolutivo. El algoritmo de búsqueda local empleado es el $iRprop^+$, el mismo que emplea el algoritmo **MPDENN** (líneas 16-22).

Finalmente, el proceso evolutivo finaliza cuando se alcance el número máximo de generaciones.

4.4.1 Operadores Empleados en el MPENSGAII

En esta Subsección se detallarán los principales operadores empleados por el algoritmo MPENSGAII: el ordenamiento rápido de individuos, la distancia *crowding* y los operadores de mutación.

4.4.1.1 Ordenamiento Rápido de Individuos

Para realizar el ordenamiento, este procedimiento mantiene dos variables para cada uno de los individuos de la población ($P[i]$):

- El conjunto S_i de las soluciones dominadas por $P[i]$.
- El número nd_i de soluciones que dominan a $P[i]$.

Para obtener estas variables, se comparan entre sí todos los individuos de la población. Si el individuo $P[i]$ domina al individuo $P[j]$, este último se agrega al conjunto S_i . Por el contrario, si $P[i]$ es dominado por $P[j]$, se incrementa en 1 el valor del contador nd_i . En el caso de que no exista relación de dominancia entre $P[i]$ y $P[j]$, no se actualiza ninguna de las dos variables.

Una vez determinadas estas variables para todos los individuos, se forma el primer frente de soluciones no dominadas, denotado por F^1 , con todos los individuos cuyo contador nd sea igual a 0. El algoritmo prosigue recorriendo el conjunto S_i de cada individuo $P[i] \in F^1$, reduciendo en uno el valor de nd_j de cada individuo $P[j] \in S_i$. Cuando el valor de nd_j es igual a 0, se agrega $P[j]$ a una lista H , inicialmente vacía. Cuando se han recorrido todos los individuos del primer frente, en H quedan los individuos que forman el segundo frente F^2 . El proceso se repite hasta que todos los individuos de la población son asignados a un frente.

4.4.1.2 Distancia Crowding

La distancia *crowding* es empleada para determinar la densidad de soluciones alrededor de un individuo $F^f[i]$ con respecto a los demás individuos del frente F^f .

Para cada individuo $F^f[i]$, se calcula un valor $F^f[i]_{dist}$ que sirve como un estimador del tamaño del cuboide más grande que encierra la solución sin incluir ningún otro individuo de la población. Este valor es lo que se conoce como distancia *crowding*.

Esta distancia se determina para cada individuo del frente considerado y para cada uno de los objetivos, calculando para cada uno de los individuos cuál es el siguiente elemento menor y mayor dentro del frente. Para ello se ordenan de menor a mayor los elementos del frente para cada uno de los objetivos. Luego, el valor de la distancia *crowding* de un elemento $F^f[i]$ se calcula como la suma de las distancias entre los individuos inmediatamente mayor y menor considerando cada objetivo. En este punto hay que tener en cuenta que los diferentes objetivos pueden estar expresados en unidades diferentes, por lo que sería conveniente realizar una normalización de todos los objetivos.

Además de la distancia *crowding*, también se define un operador de comparación por *crowding* (\geq_n). El objetivo de este operador es guiar el proceso de selección en las diferentes etapas del algoritmo hacia un frente de Pareto óptimo uniformemente distribuido.

Asumiendo que cada uno de los individuos en la población tiene dos atributos: la posición en la clasificación por no dominancia calculado con el ordenamiento rápido de individuos ($P[i]_{\text{rank}}$) y su distancia *crowding* ($P[i]_{\text{dist}}$), el orden \geq_n se puede definir como $P[i] \geq_n P[j]$ si:

$$(P[i]_{\text{rank}} < P[j]_{\text{rank}}) \vee ((P[i]_{\text{rank}} = P[j]_{\text{rank}}) \wedge (P[i]_{\text{dist}} > P[j]_{\text{dist}})),$$

donde $P[i]_{\text{rank}} = f$ si $P[i] \in F^f$.

Esto es, se define un orden lexicográfico con dos objetivos, con la posición en la eficiencia en clasificación por no dominancia como el de mayor importancia. Entonces, entre dos individuos con diferente posición en la clasificación por no dominancia se prefiere aquella con la clasificación más baja. De otra forma, si ambas soluciones están localizadas en el mismo frente, se prefiere la que esté ubicada en una región con un menor número de puntos.

4.4.1.3 Operadores de Mutación

Los operadores de mutación empleados por el algoritmo [MPENSGAI](#) se dividen en cuatro mutaciones estructurales y una mutación paramétrica. Las mutaciones estructurales producen un cambio en la estructura de la [ANN](#) mientras que la mutación paramétrica realiza una modificación en los pesos de la [ANN](#). Estos operadores tienen la misma probabilidad de utilizarse a la hora de aplicar la mutación sobre el individuo seleccionado. Por tanto, cada operador de mutación tiene 20% de probabilidad de aplicarse. Si la mutación seleccionada no puede aplicarse debido a situaciones especiales en la estructura de la [ANN](#) (por ejemplo, no se puede eliminar la última neurona de la capa oculta), se selecciona aleatoriamente otro operador de mutación.

- Añadir neuronas (mutación estructural): Con esta mutación se añaden 1 o 2 neuronas a la capa oculta de la red. En el caso extremo de que la capa oculta ya disponga del número máximo de neuronas fijado al inicio de la ejecución,

4.5 SELECCIÓN DE LAS MEJORES SOLUCIONES DEL FRENTA DE PARETO

se considerará que la mutación no ha tenido éxito y se aplicará otro operador distinto. Los pesos de los enlaces de las neuronas añadidas se establecen aleatoriamente.

- Eliminar neuronas (mutación estructural): Esta mutación elimina 1 o 2 neuronas de la capa oculta de la red. Se considerará que la mutación no ha tenido éxito si la capa oculta posee el número mínimo de neuronas fijado al inicio de la ejecución. Si la neurona escogida es la única neurona con la cual está enlazada una neurona de la capa de salida, se seleccionará automáticamente otra neurona de la capa oculta.
- Añadir enlaces (mutación estructural): Este operador añade nuevos enlaces entre dos neuronas de capas consecutivas. En caso de existir un enlace entre las dos neuronas seleccionadas, este será sobrescrito con el nuevo valor del peso. En total, se añadirán tantos enlaces como el 30% del total de enlaces que haya entre la capa de entrada y la capa oculta y el 5% del total de los existentes entre la capa oculta y la capa de salida.
- Eliminar enlaces (mutación estructural): Con este operador de mutación se eliminan el 30% del total de enlaces existentes entre la capa de entrada y la capa oculta y el 5% del total de los que haya entre la capa oculta y la capa de salida. Se considerará que la mutación no ha tenido éxito si el enlace seleccionado es el único de la neurona destino del enlace (para evitar neuronas incomunicadas). También se considerará no exitosa la mutación si el enlace seleccionado es el único enlace de salida de una neurona de la capa oculta.
- Mutación paramétrica: Esta mutación añade a cada uno de los pesos de la red un ruido Gaussiano $N(0, T(t))$, donde $T(t)$ representa la temperatura en la generación t a lo largo de la evolución (el valor de $T(t)$ disminuye produciéndose descensos más drásticos al comienzo de la evolución (exploración) y más lentos al final (explotación)). La temperatura $T(t)$ se define como:

$$T(t) = \begin{cases} rT(t-1) & \text{si } t = kG_t, k = 1, 2, \dots, \frac{G_{\max}}{G_t}, \\ T(t-1) & \text{en otro caso} \end{cases},$$

donde r es el factor de descenso, G_t es el número de generaciones que deben transcurrir para que se actualice el valor de la temperatura y G_{\max} es el número máximo de generaciones del algoritmo.

4.5 SELECCIÓN DE LAS MEJORES SOLUCIONES DEL FRENTA DE PARETO

Una vez finalizada la ejecución de cualquier algoritmo multiobjetivo, surge el problema de seleccionar una solución de entre todas las que forman el primer frente de

Pareto. La elección de un individuo concreto para presentar sus resultados es una tarea compleja que puede depender del problema y del posible uso que tendrá la solución seleccionada. Una alternativa a la selección de una única solución es el uso de *ensembles*. Un *ensemble* es un modelo compuesto, formado a partir de la agregación de varios modelos, y cuya predicción depende de las predicciones de los individuos agregados [89].

A continuación se van a presentar diferentes técnicas empleadas para seleccionar individuos del frente de Pareto.

- Individuos extremos del frente de Pareto: En este caso se seleccionarán los individuos con el mejor valor en cada uno de los objetivos. El principal problema de este método es que no garantiza que estos individuos presenten el mejor rendimiento en el conjunto de generalización, debido a que puede existir un sobreentrenamiento durante la creación de los modelos. Los individuos seleccionados con este procedimiento pueden ser vistos en la Figura 4.8(a).

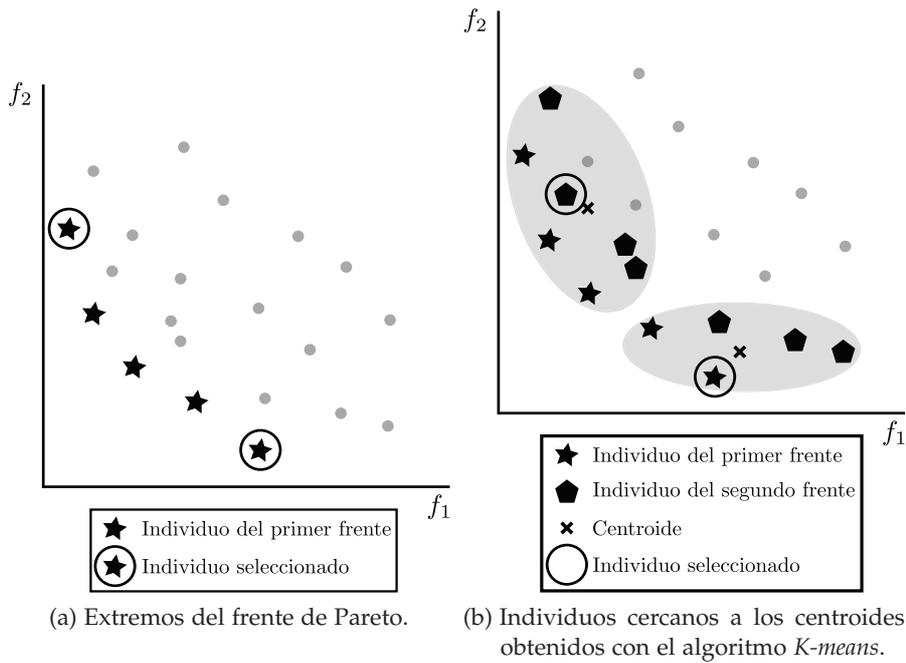


Figura 4.8: Métodos de selección de individuos.

- Selección mediante *clustering*: Este método aplica el algoritmo de *clustering K-means* (con $K = 2$) sobre los individuos pertenecientes a los dos primeros frentes de Pareto para seleccionar los dos individuos más representativos. Una vez finalizado el algoritmo de *clustering*, se seleccionan los individuos más cercanos a los centroides obtenidos, los cuales serán considerados como los individuos

más representativos de la población. A pesar del hecho de que estos individuos no tengan los mejores valores en ninguno de los objetivos en entrenamiento, no significa que no puedan generalizar bien. Este método puede ser visto gráficamente en la Figura 4.8(b).

El segundo frente de Pareto se ha incluido en este método, para aumentar el número de individuos e incrementar la diversidad. Además, los individuos de este frente pueden tener un alto nivel de clasificación en generalización debido a que no sobreentrenan durante la fase de entrenamiento.

- Selección basada en el coeficiente de proximidad: Este método selecciona al individuo del primer frente de Pareto con mayor coeficiente de proximidad. Este coeficiente mide la distancia de una solución con respecto a la solución ideal positiva y la solución ideal negativa. La solución ideal positiva es la solución que presenta los mejores valores para cada uno de los objetivos. Por otro lado, la solución ideal negativa es aquella que presenta los peores valores para todos los objetivos. Para un problema de maximización de dos objetivos, la solución ideal positiva sería el punto (1, 1) y la solución ideal negativa el punto (0, 0). Una vez determinadas la solución ideal positiva y negativa, se calcula la distancia Euclídea entre estas soluciones y cada uno de los individuos del frente de Pareto. Estas distancias serán denotadas como d_i^+ para la distancia del individuo i a la solución ideal positiva y d_i^- para la distancia a la solución ideal negativa. Una vez calculadas estas distancias, el coeficiente de proximidad puede ser calculado según la siguiente expresión:

$$CP_i = \frac{d_i^-}{d_i^+ + d_i^-}.$$

A modo de ejemplo, la Figura 4.9 muestra la distancia a la solución ideal positiva del punto C (d_C^+) y la distancia del punto D a la solución ideal negativa (d_D^-).

Una vez vistos los diferentes métodos para seleccionar individuos de la población, se van a presentar los tres métodos de *ensembles* más conocidos: *majority voting*, *simple averaging* y *winner take all* [123]. Un aspecto importante a la hora de emplear este tipo de técnica es determinar qué y cuántos modelos serán seleccionados para constituir el *ensemble*. Otro aspecto a tener en cuenta es la diversidad existente entre los modelos seleccionados.

Normalmente, cuando se trabaja con MOEAs, el *ensemble* está formado por los individuos pertenecientes al primer frente de Pareto. De manera que el tamaño del primer frente de Pareto determina el tamaño del *ensemble*. La diversidad entre estos individuos queda garantizada, puesto que los MOEAs buscan conseguir frentes lo más diversos posibles (mediante técnicas como la distancia al vecino más cercano o la distancia *crowding*).

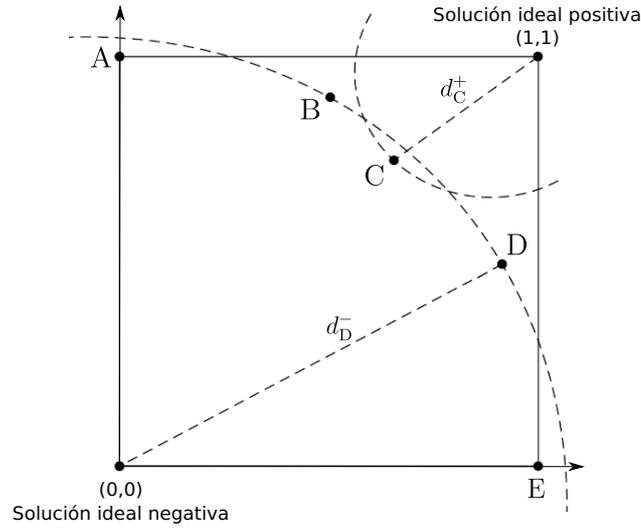


Figura 4.9: Ejemplo de distancias a las soluciones ideal positiva e ideal negativa.

- *Majority voting* o mayoría de votos: En esta técnica, cada clasificador predice (“vota”) a qué clase deberá pertenecer el patrón. Una vez reunidas todas las predicciones, el patrón es asignado a la clase que más veces ha sido predicha por los clasificadores.

Considerando un conjunto de clasificadores $C = \{C_1, \dots, C_T\}$, la decisión del clasificador t para la clase q y el patrón \mathbf{x} es $d_{t,q}(\mathbf{x}) \in \{0, 1\}$ para $t = 1, \dots, T$ y $q = 0, \dots, Q - 1$, donde T el tamaño del *ensemble* y Q es el número de clases. La decisión $d_{t,q}(\mathbf{x})$ será igual a 1 si el clasificador t clasifica el patrón \mathbf{x} en la clase q y 0 en cualquier otro caso. La decisión final de esta técnica de *ensemble* será:

$$d_{MV}(\mathbf{x}) = \arg \max_q \sum_{t=1}^T d_{t,q}(\mathbf{x}) \text{ para } q = 0, \dots, Q - 1.$$

- *Simple averaging* o media simple: Este método de *ensemble* calcula, para cada patrón, la media aritmética de las probabilidades de asignación a cada una de las Q clases obtenidas por cada uno de los miembros del *ensemble*. El patrón será asignado a la clase con la mayor probabilidad media.

Siguiendo con la notación previa, la probabilidad obtenida por el clasificador t para la clase q y el patrón \mathbf{x} es $p_{t,q}(\mathbf{x}) \in [0, 1]$ para $t = 1, \dots, T$ y $q = 0, \dots, Q - 1$. Finalmente, la decisión del método *simple averaging* es:

$$d_{SA}(\mathbf{x}) = \arg \max_q \frac{1}{T} \sum_{t=1}^T p_{t,q}(\mathbf{x}) \text{ para } q = 0, \dots, Q - 1.$$

4.5 SELECCIÓN DE LAS MEJORES SOLUCIONES DEL FRENTE DE PARETO

- *Winner take all* o el ganador lo consigue todo: La decisión de este método de *ensemble* es igual a la decisión tomada por el clasificador con mayor probabilidad de asignación en cualquiera de las clases. De este modo, y siguiendo con la notación anterior, la decisión de este método se define según lo siguiente:

$$d_{WTA}(\mathbf{x}) = \arg \max_q \left[\underset{t}{\text{máx}} (p_{t,q}(\mathbf{x})) \right] \text{ para } t = 1, \dots, T \text{ y } q = 0, \dots, Q - 1.$$

En los artículos [31, 35], el autor de la presente Tesis presenta varios estudios en los que se comparan las diferentes técnicas de selección de individuos. En general, todos los métodos de selección obtienen resultados competitivos, por lo que la elección de uno de ellos dependerá de los objetivos que se persigan en cada problema.

MÉTRICAS DE RENDIMIENTO EN CLASIFICACIÓN NOMINAL

En la literatura existen multitud de métricas de evaluación empleadas para cuantificar el rendimiento de un clasificador [49, 117]. Estas métricas, además, son empleadas para comparar el comportamiento de dos o más clasificadores y para guiar a los Algoritmo Evolutivo (*Evolutionary Algorithm*) (EA).

Las métricas de rendimiento se pueden dividir en dos grupos: las métricas que emplean la matriz de confusión y las que se calculan sin hacer uso de ella.

Además de las métricas de rendimiento que veremos en este capítulo, existen otras medidas empleadas para guiar la evolución de un MOEA. Estas métricas pueden ser:

- de diversidad: Medidas empleadas para favorecer y mantener la diversidad dentro del frente de Pareto.
- de complejidad: Métricas que miden la complejidad del clasificador. Ejemplos de este tipo de métricas son el número de enlaces de la red neuronal o el número de neuronas en la capa oculta.

5.1 MÉTRICAS ASOCIADAS A LA MATRIZ DE CONFUSIÓN

Antes de comenzar a describir las distintas métricas, es necesario definir el concepto de matriz de confusión o contingencia. Primero analizaremos la matriz de confusión obtenida en los problemas de clasificación binaria. En estos problemas, una de las dos clases será considerada como la clase positiva, y la otra como negativa. La matriz de confusión $M(g)$ de un clasificador binario g se define como:

$$\mathbf{M}(g) = \begin{pmatrix} tp & fn \\ fp & tn \end{pmatrix},$$

donde:

- tp (verdaderos positivos, *true positive*) es el número de elementos pertenecientes a la clase positiva y que el clasificador ha predicho como positivos.

- fn (falsos negativos, *false negative*) es el número de elementos de la clase positiva que han sido clasificados como negativos.
- fp (falsos positivos, *false positive*) es el número de elementos pertenecientes a la clase negativa que han sido clasificados como positivos.
- tn (verdaderos negativos, *true negative*) es el número de elementos de la clase negativa que han sido predichos como negativos.

Para un problema con Q clases y N patrones de entrenamiento/generalización, la matriz de confusión $M(g)$, de tamaño $Q \times Q$, se define de la siguiente forma:

$$M(g) = \left\{ n_{ij}; \sum_{i,j=1}^Q n_{ij} = N \right\}.$$

La Tabla 5.1 muestra, de forma extendida, la matriz de confusión $M(g)$, donde n_{ij} representa el número de patrones predichos por el clasificador g en la clase j cuando en realidad pertenecen a la clase i , $n_{i\bullet}$ es el número de patrones que pertenecen a la clase i y $n_{\bullet j}$ es el número de patrones clasificados en la clase j . La diagonal de la matriz de confusión se corresponde con los patrones correctamente clasificados y los elementos de fuera de la diagonal principal con los errores de clasificación.

Tabla 5.1: Matriz de confusión $M(g)$ para un problema con Q clases.

		Clase predicha					
		1	...	j	...	Q	
Clase real	1	n_{11}	...	n_{1j}	...	n_{1Q}	$n_{1\bullet}$
	⋮	⋮	⋱	⋮	⋱	⋮	⋮
	i	n_{i1}	...	n_{ij}	...	n_{iQ}	$n_{i\bullet}$
	⋮	⋮	⋱	⋮	⋱	⋮	⋮
	Q	n_{Q1}	...	n_{Qj}	...	n_{QQ}	$n_{Q\bullet}$
		$n_{\bullet 1}$...	$n_{\bullet j}$...	$n_{\bullet Q}$	N

Una vez detallada la matriz de confusión, se describirán las métricas más influyentes que hacen uso de ella. Primero expondremos las métricas más empleadas en problemas binarios y, posteriormente, para multiclase.

- Precisión (CCR): Rendimiento global de un clasificador, es decir, porcentaje de patrones correctamente clasificados.

$$CCR = \frac{tp + tn}{tp + fn + fp + tn} = \frac{tp + tn}{N}.$$

- Precisión positiva (P): Porcentaje de patrones correctamente clasificados de la clase positiva con respecto a todos los elementos que el clasificador predijo como positivos.

$$p = \frac{tp}{tp + fp}$$

- Sensibilidad (S): Porcentaje de patrones de la clase positiva correctamente clasificados con respecto al total de patrones de la clase positiva. Esta métrica también se conoce por *Recall* o *True Positive Rate* (TPR).

$$S = TPR = \frac{tp}{tp + fn}$$

- Especificidad (SP): Porcentaje de patrones de la clase negativa correctamente clasificados con respecto al total de elementos de la clase negativa.

$$SP = \frac{tn}{fp + tn}$$

- Porcentaje de falsos positivos: Porcentaje de patrones incorrectamente clasificados de la clase negativa con respecto al total de patrones existentes en la clase negativa. También conocida como *False Positive Rate* (FPR). Es un valor complementario a la SP.

$$FPR = \frac{fp}{fp + tn} = 1 - SP$$

- F_{score} : Esta métrica es una media ponderada de las medidas P y TPR. Es la relación entre el total de elementos positivos y los positivos predichos por el clasificador.

$$F_{score} = 2 \frac{P \cdot TPR}{P + TPR} = \frac{2tp}{2tp + fp + fn}$$

- Área bajo la curva ROC (AUC): La curva ROC (*Receiver Operating Characteristic*) es una representación gráfica de los valores de TPR frente a los de FPR según varía el umbral de discriminación (valor a partir del cual un patrón es predicho como positivo). Esta curva se encuadra en un cuadrado de lado la unidad. El AUC se define como el área situada debajo de la curva ROC. Esta métrica permite realizar una comparación gráfica del rendimiento de varios clasificadores. El valor AUC de un clasificador se puede aproximar según la siguiente estimación:

$$AUC = \frac{1 + TPR - FPR}{2}$$

- Media Geométrica (MG): La media geométrica intenta maximizar la precisión de las dos clases del problema de la manera más balanceada posible.

$$MG = \sqrt{TPR - TPR \cdot FPR}.$$

- Kappa de Cohen (K): Esta métrica muestra el grado de acuerdo entre los valores reales del problema y los valores predichos por el clasificador. Los resultados obtenidos con esta métrica demuestran que es una medida más robusta que la precisión en problemas desbalanceados con múltiples clases. El índice Kappa se calcula de la siguiente manera:

$$K = \frac{P_o - P_e}{1 - P_e},$$

donde P_o es la proporción de concordancia observada (CCR del clasificador) y P_e es la proporción de concordancia esperada por puro azar, o lo que es lo mismo:

$$P_e = \frac{(tp + fn)(tp + fp) + (fp + tn)(fn + tn)}{N^2}.$$

Esta métrica toma valores en el rango $[-1, 1]$, donde 1 significa un acuerdo perfecto, -1 un completo desacuerdo y 0 indica que no existe relación.

A continuación, se describirán algunas de las métricas más empleadas en problemas con múltiples clases.

- Precisión (CCR): Como en el caso de los problemas binarios, es el porcentaje de patrones correctamente clasificados.

$$CCR = \frac{1}{N} \sum_{i=1}^Q n_{ii}.$$

- Mínima Sensibilidad (MS): Extendiendo el concepto de sensibilidad visto anteriormente, se puede definir la sensibilidad de la clase i como el número de patrones correctamente predichos de la clase i con respecto al total de patrones de esta clase, $S_i = n_{ii}/n_{i\bullet}$. La sensibilidad para la clase i es un estimador de la probabilidad de predecir correctamente un patrón de esta clase. A partir de las sensibilidades de todas las clases, podemos definir la mínima sensibilidad del clasificador como el mínimo de las sensibilidades de cada una de las clases:

$$MS_i = \text{mín}(S_i; i = 1, \dots, Q).$$

Esta métrica representa el desempeño de la clase peor clasificada.

5.2 MÉTRICAS NO RELACIONADAS CON LA MATRIZ DE CONFUSIÓN

- Macro-Media (MM): La macro-media se define como la media de las sensibilidades de todas las clases. Esta métrica considera que todas las clases del problema tienen la misma importancia.

$$MM = \frac{1}{Q} \sum_{i=1}^Q S_i.$$

- Área bajo la curva ROC (AUC): Se han realizado multitud de aproximaciones, aunque no existe ninguna que esté completamente consolidada [85]. Una de las adaptaciones más comúnmente empleada es la siguiente: primero se calcula el valor de AUC considerando cada una de las clases frente al resto y luego se realiza la media ponderada de los valores AUC obtenidos.
- Kappa de Cohen (K): Esta métrica binaria también puede ser utilizada en problemas multiclase, mediante la generalización de los valores P_o y P_e . El valor P_o es igual al CCR del clasificador multiclase:

$$P_o = \frac{1}{N} \sum_{i=1}^Q n_{ii},$$

y el valor P_e se define como:

$$P_e = \frac{1}{N^2} \sum_{i=1}^Q n_{i \cdot} \cdot n_{\cdot i}.$$

5.2 MÉTRICAS NO RELACIONADAS CON LA MATRIZ DE CONFUSIÓN

Existen otras métricas que no necesitan la matriz de confusión para poder medir la bondad del clasificador. Estas métricas se calculan midiendo el error que produce la salida del clasificador.

- Error cuadrático medio (MSE): Esta medida, también conocida como *Mean Square Error*, muestra la media de los errores cuadráticos cometidos por el clasificador, es decir, el promedio de las diferencias entre los valores predichos y reales al cuadrado. Tradicionalmente, esta medida se empleaba en problemas de regresión, aunque ahora también se emplea ampliamente en clasificación.

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{Y}_i - Y_i)^2,$$

donde \hat{Y}_i e Y_i son, respectivamente, la etiqueta predicha y la etiqueta real del patrón i . Los valores que puede tomar esta métrica están en el rango $[0, \infty)$.

- Raíz del error cuadrático medio (RMSE): Esta métrica es la raíz cuadrada del MSE, conocida también como *Root Mean Square Error*. La medida RMSE proporciona otro promedio del error cometido por un clasificador. Sus valores también están comprendidos en el rango $[0, \infty)$.

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{Y}_i - Y_i)^2}.$$

- Error absoluto medio (MAE): Esta métrica, también conocida como *Mean Absolute Error*, muestra la media de los errores obtenidos por el clasificador en valor absoluto.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{Y}_i - Y_i|.$$

- Entropía cruzada (E): La entropía cruzada está basada en las probabilidades de pertenencia de los patrones de un conjunto de datos a cada una de las clases del problema. Los valores de la entropía cruzada están en el rango $[0, \infty)$.

$$E = -\frac{1}{N} \sum_{n=1}^N \sum_{q=i}^Q y_n^{(q)} \log p_q(\mathbf{x}_n),$$

donde $y_n^{(q)}$ es igual a 1 si el patrón \mathbf{x}_n pertenece a la clase q e igual a 0 en caso contrario y $p_q(\mathbf{x}_n)$ es la probabilidad de pertenencia del patrón \mathbf{x}_n a la clase q .

5.3 RELACIÓN ENTRE LA MÍNIMA SENSIBILIDAD Y LA PRECISIÓN

La comunidad que investiga en las áreas de estadística y de *Machine Learning* ha utilizado tradicionalmente la Precisión (*Correct Classification Rate*) (CCR) como métrica para medir el rendimiento de un clasificador. Sin embargo, esta medida sólo presenta el nivel de clasificación global, obviando por lo general el porcentaje de buena clasificación de cada una de las clases. Este comportamiento ha sido criticado por diferentes autores, sobre todo en problemas multiclase y/o no balanceados [104], debido a que la métrica CCR no puede capturar los diferentes aspectos de un clasificador, sobre todo al comparar varios clasificadores entre sí.

Suponiendo que el coste de una mala clasificación es igual y que no tenemos preferencia en la clasificación de una determinada clase, entenderemos que un buen clasificador es aquel que presenta un alto nivel de precisión a la vez que un considerable nivel de buena clasificación para cada una de las clases. En problemas reales, estos dos objetivos están en conflicto puesto que para alcanzar un alto nivel de precisión, por lo general, es necesario sacrificar el nivel de precisión de una o varias clases, sobre todo en problemas no balanceados o con múltiples clases [105].

Para solucionar las deficiencias de las medidas globales como la métrica CCR, en [55] se propone el uso de la métrica CCR junto con la Mínima Sensibilidad (*Minimum Sensitivity*) (MS) para guiar la evolución de un MOEA. De esta manera, al evaluar a los individuos de la población se tendrá en cuenta tanto el rendimiento global como el desempeño de la clase peor clasificada.

La selección de la métrica MS como una medida complementaria a la métrica CCR se justifica si consideramos que el valor de la métrica CCR es una media ponderada de las sensibilidades de cada una de las clases (S_i), es decir:

$$CCR = \sum_{i=1}^Q \frac{S_i \cdot n_{i\bullet}}{N}.$$

De esta forma, ambas métricas nos proporcionen una visión intuitiva de las sensibilidades de cada una de las Q clases del problema.

Mediante estas dos métricas, cada clasificador se puede representar como un punto en la región blanca de la Figura 5.1, donde podemos visualizar los resultados obtenidos independientemente del número de clases del problema. Un clasificador (un punto en el espacio (MS, CCR)) domina a otro si está por encima y/o a la derecha, esto es, si tiene mayor valor de CCR y/o mejor valor de MS.

A simple vista, se puede llegar a pensar que las medidas MS y CCR están correladas o relacionadas positivamente. Esto puede ser cierto para valores pequeños de CCR y, por lo tanto, de MS, pero, en general, no ocurre lo mismo cuando los valores de MS y CCR se acercan al punto (1, 1). Es decir, para valores cercanos a 1 de CCR y MS, un incremento en CCR no implica necesariamente un aumento de MS y viceversa.

Es claro que dos cantidades no pueden contemplar toda la información contenida por las $Q(Q - 1)$ proporciones de mala clasificación contenidas en la matriz de confusión; sin embargo, el par (MS, CCR) intenta obviar las desventajas de considerar todas estas proporciones de mala clasificación como, por ejemplo, las dificultades para obtener una representación gráfica (si consideramos estas $Q(Q - 1)$ proporciones) que nos permita ver el rendimiento de los clasificadores, el incremento en la dimensión del frente de Pareto con respecto al número de objetivos o el coste computacional asociado con un problema de optimización multiobjetivo que tiene tantos objetivos. Así, el par (MS, CCR) plantea un compromiso entre el rendimiento de un clasificador basado en una única medida escalar, como es la métrica CCR, y las medidas multidimensionales basadas en las $Q(Q - 1)$ proporciones de mala clasificación.

Proposición 5.3.1 *Considerando un problema de clasificación con Q clases y siendo p^* la frecuencia relativa de la clase minoritaria estimada a priori, es decir, $p^* = \frac{n_{1\bullet}}{N}$, siendo $n_{1\bullet}$ el número de patrones de la clase minoritaria y N el total de patrones, la relación existente entre las medidas CCR y MS se puede definir según las siguientes inecuaciones:*

$$MS \leq CCR \leq 1 - (1 - MS)p^*.$$

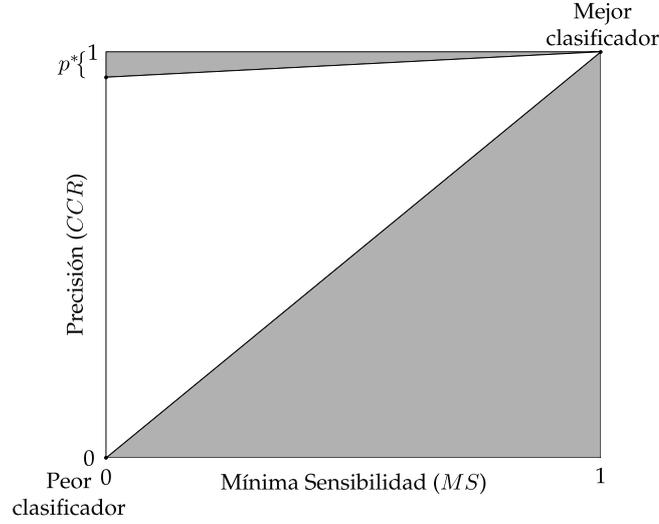


Figura 5.1: Región factible (zona blanca) en el espacio (MS, CCR) para un problema de clasificación.

Demostración Comenzando por el límite superior de la inecuación, teniendo en cuenta las definiciones de CCR y MS y recordando que $\sum_{i=1}^Q n_{i\bullet} = N$, se puede comprobar que:

$$\begin{aligned} CCR &= \sum_{i=1}^Q \frac{S_i \cdot n_{i\bullet}}{N} = MS \frac{n_{1\bullet}}{N} + \sum_{i \neq 1}^Q S_i \frac{n_{i\bullet}}{N} \leq MS \frac{n_{1\bullet}}{N} + \sum_{i \neq 1}^Q \frac{n_{i\bullet}}{N} = \\ &= MS \frac{n_{1\bullet}}{N} + 1 - \frac{n_{1\bullet}}{N} = 1 - (1 - MS) \frac{n_{1\bullet}}{N} = 1 - (1 - MS)p^*. \end{aligned}$$

Por otro lado, el límite inferior se obtiene de la siguiente forma:

$$CCR = \frac{1}{N} \sum_{i=1}^Q n_{ii} = \sum_{i=1}^Q \frac{n_{ii}}{n_{i\bullet}} \frac{n_{i\bullet}}{N} = \sum_{i=1}^Q S_i \frac{n_{i\bullet}}{N} \geq MS \sum_{i=1}^Q \frac{n_{i\bullet}}{N} = MS.$$

Proposición 5.3.2 Considerando un problema de clasificación con Q clases y siendo p^* la frecuencia relativa de la clase minoritaria estimada a priori, cada punto en el espacio (MS, CCR) pertenece al subconjunto:

$$F = \{(MS, CCR) : 0 \leq MS \leq 1, MS \leq CCR \leq 1 - p^* + (MS)p^*\}.$$

Así, el subconjunto completo F se corresponde con la región factible.

Demostración Para demostrar que el subconjunto de puntos F se corresponde con la región factible, probaremos que cada punto de dicho subconjunto se puede obtener a partir de una matriz de confusión. Sea (MS_0, CCR_0) un punto en F , la demostración

5.3 RELACIÓN ENTRE LA MÍNIMA SENSIBILIDAD Y LA PRECISIÓN

consistirá en la construcción de una matriz de confusión con $MS = MS_0$ y $CCR = CCR_0$ para ese punto. Definamos la función continua y lineal $f : [MS_0, 1] \rightarrow \mathbb{R}$ como:

$$f(MS) = (MS_0)p^* + MS(1 - p^*),$$

la cual satisface lo siguiente:

$$f(MS_0) = MS_0 \leq CCR_0, f(1) = 1 - p^* + (MS_0)p^* \geq CCR_0.$$

La continuidad de la función f implica que existe un valor $MS^* \in [MS_0, 1]$ de manera que $f(S^*) = CCR_0$. Definiendo la siguiente matriz de confusión:

$$\begin{array}{c} \text{Clase} \\ \begin{array}{cccccc} 1 & 2 & \dots & Q-1 & Q \\ \left(\begin{array}{cccccc} (MS_0)p^* & (1 - MS_0)p^* & \dots & 0 & 0 \\ \frac{(1-S^*)(1-p^*)}{Q-1} & \frac{S^*(1-p^*)}{Q-1} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \frac{(1-S^*)(1-p^*)}{Q-1} & \frac{S^*(1-p^*)}{Q-1} & 0 \\ 0 & 0 & \dots & \frac{(1-S^*)(1-p^*)}{Q-1} & \frac{S^*(1-p^*)}{Q-1} \end{array} \right) \end{array} \\ \end{array} \begin{array}{l} p^* \\ \frac{1-p^*}{Q-1} \\ \dots \\ \frac{1-p^*}{Q-1} \\ \frac{1-p^*}{Q-1} \end{array}$$

Se puede observar que las sensibilidades cumplen:

$$S_1 = MS_0, S_2 = \dots = S_Q = \frac{S^*(1-p^*)}{Q-1} \frac{Q-1}{1-p^*} = S^*.$$

Por tanto, $MS = \min\{MS_0, S^*\} = MS_0$, ya que $S^* \in [MS_0, 1]$. Por otra parte, se verifica que la precisión, CCR , asociada a la matriz es:

$$CCR = MS_0p^* + (Q-1)S^* \frac{1-p^*}{Q-1} = f(S^*) = CCR_0.$$

Finalmente, a partir de la condición $p^* \leq \frac{1}{Q}$ obtenemos que:

$$p^*Q \leq 1 \implies p^*Q - p^* \leq 1 - p^* \implies p^* \leq \frac{1-p^*}{Q-1},$$

y podemos concluir que la mínima probabilidad de las columnas es p^* .

Proposición 5.3.3 *El área A del conjunto factible F (región blanca de la Figura 5.1) verifica que $A = \frac{1-p^*}{2}$, donde p^* es la frecuencia relativa de la clase minoritaria estimada a priori.*

Demostración La demostración es inmediata a partir de la Proposición 5.3.2. Cuando el número de clases aumenta o el problema es altamente desbalanceado, el valor de p^* disminuye. Por tanto, al disminuir el valor de p^* , el área de F aumenta. Un análisis similar prueba que el valor mínimo del área se alcanza cuando el problema es totalmente balanceado. En consecuencia, en problemas no balanceados, la línea que define el borde superior del conjunto en el plano (MS, CCR) tiende a ser horizontal, y el rango de valores de MS será mayor incluso para valores altos de CCR . En este caso, el valor de CCR como medida única de un clasificador multiclase es insuficiente, ya que esconde muchas posibilidades diferentes para la métrica MS . Estos comentarios muestran que las medidas MS y CCR pueden ser especialmente ventajosas en problemas con cierto desbalanceo o cuando el número de clases es alto, confirmando que, por sí sola, la medida CCR es inadecuada en estas situaciones.

PROBLEMA DE ASIGNACIÓN DONANTE-RECEPTOR EN LOS TRASPLANTES HEPÁTICOS

El trasplante de órganos es una técnica ampliamente utilizada hoy en día como tratamiento para pacientes que presentan problemas hepáticos graves. Sin embargo, dichos trasplantes están seriamente limitados por la disponibilidad de órganos apropiados, de tal forma que existe un fuerte desequilibrio entre la oferta y la demanda. Esta situación, desgraciadamente, se traduce en la muerte de pacientes en lista de espera. En España, a pesar del aumento de donantes en los últimos años (con una tasa que alcanza el 34,4 p.m.p (por millón de personas) en 2009, según datos del Registro Español de Trasplante Hepático (RETH) [109]), se ha asistido también de forma paralela al crecimiento del número de pacientes que precisan un trasplante hepático, sobrepasando así los 2000 candidatos en los últimos años. Y aún cuando el número de trasplantes realizados ha crecido, es inevitable el consiguiente incremento de los pacientes en lista de espera y su mortalidad en la misma [111]. Debido a dicho desequilibrio, se han desarrollado distintos métodos para optimizar la asignación, priorizando a aquellos pacientes que podrían ser más adecuados para recibir un determinado órgano. Para el caso específico de trasplante hepático, se han propuesto distintas metodologías como el DRI (*Donor Risk Index*) [54], definido para establecer un indicador cuantitativo asociado al riesgo de trasplante, usando para ello únicamente características del donante. Otra metodología de asignación ampliamente utilizada es el MELD (*model for end-stage liver disease score*) [79]. Este método está basado en el principio de “el más enfermo primero”, por lo que sólo considera información relativa al receptor. En este sentido, estos métodos pueden considerarse como insuficientes para el problema tratado, puesto que sólo consideran características de los donantes o de los receptores, pero no ambas de forma conjunta, cuando en teoría, a la hora de tratar de predecir el éxito de un trasplante sería lógico considerar las características asociadas a ambos y al trasplante a realizar en sí. Con el empleo de estos métodos pueden surgir incompatibilidades derivadas, por ejemplo, de la edad del donante y/o receptor, el número de días que ha estado el donante en la UCI o el tiempo de isquemia fría del órgano (tiempo que ha estado congelado el órgano).

Con el objetivo de tratar con este problema, Rana *et al.* [107] propusieron un sistema de puntuación llamado SOFT (*Survival Outcomes Following liver Transplantation*)

para predecir la supervivencia del injerto después de tres meses tras el trasplante mediante las características del donante y del receptor. Recientemente, P. Dutkowski *et al.* han propuesto el sistema BAR (*balance of risk score*) [51] basado también en una combinación de las características del donante y del receptor.

Otra solución al desequilibrio entre la oferta y la demanda empleada en los últimos años consiste en el uso de órganos procedentes de donantes con criterios expandidos [17]. Estos donantes son aquellos con valores extremos en algunas variables, como por ejemplo aquellos con una edad superior a 75 años o con un índice de masa corporal mayor que 30, aquellos que han estado más de 4 días hospitalizados en la UCI o si el tiempo de isquemia fría del órgano es superior a 12 horas.

A continuación, se van a presentar los estudios realizados para la obtención de un nuevo sistema de asignación donante-receptor utilizando técnicas de *Machine Learning*, que permite asignar el órgano al paciente en lista de espera, de entre los 5 pacientes con mayor valor de MELD, que tenga la probabilidad más elevada de sobrevivir un número de meses tras el trasplante por una parte y, por otra, la menor probabilidad de rechazar el injerto antes de un número de meses. De este modo, el órgano sería recibido por aquel paciente que mejor lo aprovecharía, lo que significa que al menos el órgano trasplantado perdurará un cierto número de meses (en general, los estudios consideran la supervivencia tras 3 meses o tras 1 año). Estas probabilidades se obtienen mediante dos modelos de ANN obtenidos por un MOEA, las cuales serán combinadas mediante un sencillo sistema basado en reglas para proporcionar la decisión final del sistema. Gracias a este nuevo sistema, se obtiene una predicción que permitirá a los especialistas médicos tomar una decisión más apropiada, sin eliminar los principios de justicia, eficacia y equidad necesarios en todos los procedimientos médicos. En el caso de pacientes muy graves y con necesidad de un trasplante urgente, los expertos pueden decidir no usar este nuevo sistema y asignar el órgano directamente siguiendo el sistema actual.

6.1 DESCRIPCIÓN DE LA BASE DE DATOS UTILIZADA

La base de datos empleada para desarrollar este nuevo sistema está formada por los pacientes de 11 hospitales españoles trasplantados durante los años 2007 y 2008. Todos los receptores eran mayores de 18 años. Los pacientes sometidos a trasplantes combinados o multivisceral fueron excluidos del estudio. A todos los pacientes se les hizo un seguimiento durante un año tras el trasplante o hasta su muerte/perdida del injerto.

En total, la base de datos estaba formada por 1031 trasplantes hepáticos. De cada uno de estos trasplantes se recogieron 16 características del receptor, 20 características del donante y 5 factores preoperatorios. La variable a predecir es la supervivencia o no del injerto una vez transcurrido un año desde la operación. Los datos de 28 trasplantes fueron excluidos debido al excesivo número de valores perdidos. Para el resto

de los trasplantes, los valores perdidos fueron reemplazados por la media en el caso de variables continuas y por la moda en el caso de variables categóricas. Finalmente, cada una de las variables categóricas de la base de datos ha sido transformada en tantas variables binarias como posibles categorías estén presentes en dicha variable.

6.2 DISEÑO EXPERIMENTAL

Para la obtención de los modelos de red neuronal mediante los algoritmos MPENS-GAII y MPDENN2 se ha empleado un procedimiento *k-fold* estratificado ($k=5$) con 6 repeticiones para cada *fold* (de manera que obtenemos 30 ejecuciones diferentes). En cada ejecución, el entrenamiento se ha llevado a cabo con el 80% de los patrones seleccionados aleatoriamente y la generalización con el 20% restante. Durante la creación de estos conjuntos, la proporción 80-20% también se ha mantenido para los patrones de cada uno de los hospitales. Además, esta proporción ha sido mantenida entre los patrones de las dos clases del problema: supervivencia del injerto tras un año de la operación y no supervivencia.

Las funciones de base de los modelos de red neuronal empleadas en este estudio han sido las funciones sigmoideas y las funciones de base radial generalizadas (GRBF). Se ha empleado este último tipo de función debido al buen rendimiento que presentan en problemas con alta dimensionalidad (como es el caso del problema de asignación donante-receptor) [56].

La Tabla 6.1 resume las características de la base de datos: número total de patrones en cada conjunto de datos, número de patrones en entrenamiento y en generalización, número de variables de entrada, número de clases, número total de patrones por clase y valor de p^* (porcentaje de patrones de la clase minoritaria).

Tabla 6.1: Características de la bases de datos donante-receptor.

Total patrones	Patrones entrenam.	Patrones generali.	Variables de entrada	Número de clases	Patrones por clase	p^*
1003	751	252	64	2	890-113	0,1126

La evolución de los dos algoritmos evolutivos es guiada por la entropía (E) y la mínima sensibilidad (MS). Con el objetivo de acotar las dos métricas en el intervalo $[0, 1]$, se ha aplicado sobre la medida E la siguiente transformación estrictamente decreciente $\bar{A} = 1/(1 + E)$. Con esto se definen las dos funciones a maximizar por los diferentes MOEAs. Se ha empleado la medida E en lugar de la métrica CCR debido a que esta última es una medida discontinua, lo cual dificulta la convergencia en algunas bases de datos, sobre todo cuando el número de patrones del conjunto de generalización es pequeño.

Para el algoritmo MPENSGAII se ha empleado un tamaño de población de 150 individuos y una probabilidad de mutación para cada operador de $1/5$, siguiendo las directrices de K. Deb [41]. Los parámetros del algoritmo MPDENN2 se han establecido siguiendo las directrices de H. Abbass [4]: tamaño de población de 50 individuos, probabilidad de cruce de $\text{pro_cruce} = 0,8$ y probabilidad de mutación de $\text{pro_mut} = 0,1$. Además, en el algoritmo MPDENN2 se aplica la búsqueda local 3 veces durante la ejecución (cada 33,33 % de las generaciones) (el Capítulo 4 contiene más información sobre estos algoritmos y sus parámetros).

Para el algoritmo de búsqueda local $iRprop^+$, se utilizan los parámetros recomendados por C. Igel [76]: $\eta^- = 0,5$, $\eta^+ = 1,2$, $\Delta_0 = 0,0125$, $\Delta_{\min} = 0$, $\Delta_{\max} = 50$ y Epochs = 10.

Para iniciar el procesamiento de datos, cada una de las variables de entrada se escalaron en el intervalo $[-2, 2]$, para evitar así la saturación de la señal. Este rango ha sido elegido para permitir el correcto funcionamiento de las redes GRBF.

6.3 RESULTADOS OBTENIDOS

La Tabla 6.2 presenta los resultados obtenidos sobre el conjunto de generalización. Los resultados mostrados se corresponden con el valor medio obtenido tras realizar las 30 ejecuciones. Se han seleccionado tanto los individuos de los extremos de los diferentes frente de Pareto como los individuos más cercanos a los centroides calculados mediante el algoritmo de *clustering K-means* sobre los individuos pertenecientes a los dos primeros frentes de Pareto (en la Sección 4.5 se puede encontrar más información sobre estos métodos de selección). Los modelos con mayor valor en la función A (extremo superior del frente) se han etiquetado como *NombreMétodo-E*, aquellos con el mayor valor de MS (extremo inferior del frente) como *NombreMétodo-MS*, los individuos más cercanos al *cluster* superior como *NombreMétodo-C-E* y los individuos más cercanos al *cluster* inferior como *NombreMétodo-C-MS*. Estos resultados han sido publicados en [34].

Como muestran los resultados, el algoritmo MPDENN2-E (SU) obtiene el mejor valor medio en la métrica CCR, seguido por el algoritmo MPENSGAII-E (GRBF). Para la métrica MS, los métodos con mejor desempeño son el algoritmo MPENSGAII-MS (GRBF) y el algoritmo MPDENN2-MS (GRBF). Con respecto a la métrica AUC, los mejores resultados son obtenidos por los métodos MPENSGAII-MS (GRBF) y MPENSGAII-E (GRBF). Los algoritmos MPENSGAII-E (GRBF) y MPDENN2-E (SU) son los que presentan mejor rendimiento en RMSE. Para la métrica K, el mejor resultado es obtenido por el algoritmo MPENSGAII-MS (GRBF) y el segundo mejor valor por los algoritmos MPENSGAII-C-MS (GRBF) y MPENSGAII-C-E (GRBF). Considerando la complejidad estructural de los modelos, el algoritmo MPENSGAII-C-E (GRBF) obtiene los modelos con el menor número de coeficientes y el algoritmo MPENSGAII-MS (GRBF) produce los modelos con menos neuronas en capa oculta.

Tabla 6.2: Resultados obtenidos en generalización: media y desviación típica ($Media_{DT}$).

Algoritmo	CCR	MS	AUC	RMSE	K	n° Coef.	n° N.O.
MPENSGAII-E (GRBF)	83,63 _{1,06}	3,72 _{2,87}	0,568 _{0,042}	0,379 _{0,021}	0,041 _{0,031}	40,07 _{17,14}	1,03 _{0,18}
MPENSGAII-MS (GRBF)	57,10 _{3,18}	52,04 _{4,32}	0,569 _{0,041}	0,506 _{0,013}	0,069 _{0,038}	38,67 _{17,97}	1,00 _{0,00}
MPENSGAII-C-E (GRBF)	78,44 _{1,68}	15,03 _{5,71}	0,552 _{0,051}	0,429 _{0,029}	0,064 _{0,060}	37,73 _{16,80}	1,07 _{0,25}
MPENSGAII-C-MS (GRBF)	63,39 _{3,32}	41,82 _{7,72}	0,565 _{0,043}	0,479 _{0,013}	0,064 _{0,055}	38,67 _{15,46}	1,03 _{0,18}
MPDENN-E (GRBF)	83,42 _{0,83}	1,24 _{2,37}	0,510 _{0,072}	0,399 _{0,022}	0,006 _{0,030}	124,57 _{47,87}	2,30 _{1,02}
MPDENN2-MS (GRBF)	56,99 _{7,32}	47,99 _{10,65}	0,543 _{0,054}	0,591 _{0,070}	0,056 _{0,053}	122,63 _{43,90}	2,13 _{0,86}
MPDENN2-C-E (GRBF)	80,60 _{5,18}	9,73 _{7,35}	0,520 _{0,062}	0,415 _{0,033}	0,053 _{0,075}	142,10 _{43,92}	2,57 _{0,73}
MPDENN2-C-MS (GRBF)	61,95 _{14,27}	30,85 _{16,04}	0,519 _{0,052}	0,545 _{0,089}	0,027 _{0,062}	130,43 _{40,68}	2,40 _{0,77}
MPENSGAII-E (SU)	82,78 _{1,15}	3,63 _{4,26}	0,541 _{0,068}	0,392 _{0,021}	0,022 _{0,061}	62,67 _{14,83}	1,90 _{0,31}
MPENSGAII-MS (SU)	57,61 _{3,74}	47,84 _{6,15}	0,556 _{0,040}	0,584 _{0,043}	0,046 _{0,051}	57,57 _{16,42}	1,77 _{0,43}
MPENSGAII-C-E (SU)	78,79 _{2,73}	13,87 _{5,82}	0,564 _{0,054}	0,433 _{0,031}	0,062 _{0,076}	57,03 _{17,63}	1,63 _{0,49}
MPENSGAII-C-MS (SU)	65,79 _{3,16}	34,66 _{7,58}	0,543 _{0,050}	0,528 _{0,038}	0,048 _{0,061}	62,70 _{17,06}	1,83 _{0,38}
MPDENN2-E (SU)	83,68 _{0,92}	2,39 _{3,91}	0,534 _{0,065}	0,390 _{0,016}	0,024 _{0,043}	137,10 _{54,92}	2,53 _{1,04}
MPDENN2-MS (SU)	55,18 _{4,70}	44,91 _{9,07}	0,533 _{0,052}	0,585 _{0,047}	0,018 _{0,052}	134,27 _{46,82}	2,40 _{0,86}
MPDENN2-C-E (SU)	81,67 _{2,76}	6,13 _{5,72}	0,536 _{0,055}	0,409 _{0,019}	0,031 _{0,051}	116,80 _{48,10}	2,23 _{1,04}
MPDENN2-C-MS (SU)	61,03 _{6,66}	38,75 _{10,60}	0,528 _{0,059}	0,568 _{0,057}	0,034 _{0,076}	137,63 _{40,27}	2,40 _{0,67}

n° Coef. = número de coeficientes del modelo; n° N.O. = número de neuronas en la capa oculta.

Los mejores resultados están marcadas en **negrita** y los segundos mejores en subrayado.

En general, comparando el rendimiento de ambos algoritmos, se observa que el mejor desempeño lo obtiene el algoritmo MPENSGAII con funciones de base GRBF, ya que presenta los mejores resultados en la mayoría de las métricas analizadas. En otras palabras, el algoritmo MPENSGAII con GRBF produce modelos más simples con resultados aceptables. Analizando únicamente los métodos que emplean los modelos de la parte superior del frente de Pareto (aquellos con el sufijo -E), los mejores resultados son alcanzados por el algoritmo MPENSGAII-E (GRBF). Por otro lado, comparando los métodos que emplean los modelos de la parte inferior del frente de Pareto (aquellos con el sufijo -MS), el algoritmo MPENSGAII-MS (GRBF) presenta el mejor rendimiento. Los resultados obtenidos por los mejores modelos generados por estos dos algoritmos pueden verse en la Tabla 6.3. Estos dos modelos presentan resultados aceptable en términos médicos.

Tabla 6.3: Resultados de los mejores modelos generados por los algoritmos MPENSGAII-E (GRBF) y MPENSGAII-MS (GRBF).

Algoritmo	CCR	MS	AUC	RMSE	K	n° Coef.	n° N.O.
MPENSGAII-E (GRBF)	84,50	3,13	0,652	0,352	0,051	57	1
MPENSGAII-MS (GRBF)	60,00	59,38	0,605	0,502	0,115	56	1

n° Coef. = número de coeficientes del modelo; n° N.O. = número de neuronas en la capa oculta.

Con el fin de determinar la significación estadística de las diferencias observadas entre los distintos algoritmos usados, diferentes tests estadísticos han sido aplicados a cada una de las métricas. En primer lugar, se ha aplicado el test estadístico de Kolmogorov-Smirnov (K-S) con un nivel de significación de $\alpha = 0,05$ para evaluar si los valores de las diferentes métricas siguen una distribución normal. Sólo 2 de los 16 algoritmos obtienen un *p-valor* inferior que el nivel crítico para las métricas CCR y K, 4 para la medida MS, 0 para las métricas AUC, RMSE y n° Coef. y 11 para el n° N.O.. Por tanto, no se puede asumir en ninguno de los casos una distribución normal. Como consecuencia de ello, el test no paramétrico de Friedman ha sido aplicado para comprobar si los métodos empleados afectan significativamente a los resultados obtenidos. El test concluye que existen diferencias significativas con un *p-valor* = 0,00. Para terminar, utilizamos el test de rangos de Wilcoxon. Para este test, el nivel de significación ha sido ajustado para controlar la tasa de error por familia dividiendo el valor de α entre el número total de comparaciones realizadas menos uno [44]:

$$\alpha_{\text{Wilcoxon}} = \frac{\alpha}{\frac{n\text{Algs}*(n\text{Algs}-1)}{2} - 1},$$

donde nAlgs es el número de algoritmos usados. Los resultados obtenidos se muestran en la Tabla 6.4. Para cada algoritmo, estos resultados incluyen el número de algoritmos a los que se gana significativamente (victorias, V), el número de algoritmos con los que no presenta diferencias significativas (empates, E) y el número de algoritmos con los que se pierde significativamente (derrotas, D).

Analizando los resultados, los algoritmos MPENSGAII-E (GRBF) y MPENSGAII-MS (GRBF) destacan por ser los enfoques más competitivos, obteniendo cada uno de ellos el mejor rendimiento en 4 de las 7 medidas analizadas.

6.4 SISTEMA BASADO EN REGLAS

En nuestra vida cotidiana, muchas situaciones complejas están regidas por reglas deterministas, como los sistemas de control de tráfico, las decisiones médicas, sistemas de seguridad y transacciones bancarias. Los sistemas basados en reglas [86] automatizan la resolución de problemas y proporcionan un medio para capturar y refinar el conocimiento humano. Los expertos tienden a expresar sus conocimientos en términos de un conjunto de reglas del tipo “condición-acción”. Estas reglas, que son los cimientos de cualquier sistema basado en reglas, se componen de dos partes: la condición (situación o antecedente) y la acción (conclusión o consecuente), es decir:

SI (condiciones) **ENTONCES** (acciones)

A partir de los mejores modelos obtenidos por los métodos MPENSGAII-E (GRBF) y MPENSGAII-MS (GRBF) se ha diseñado un sencillo sistema basado en reglas para

Tabla 6.4: Número de victorias (V), empates (E) y derrotas (D) cuando se comparan los diferentes algoritmos usando el test de Wilcoxon para $\alpha = 0,05$.

Algoritmo	CCR	MS	AUC	RMSE	K	n° Coef.	n° N.O.
MPENSGAII-E (GRBF)	11/4/0	0/5/10	0/15/0	12/3/0	<u>1/14/0</u>	11/4/0	12/3/0
MPENSGAII-MS (GRBF)	0/5/10	13/2/0	0/15/0	4/2/9	3/12/0	11/4/0	12/3/0
MPENSGAII-C-E (GRBF)	8/1/6	5/2/8	0/15/0	8/3/4	<u>1/14/0</u>	<u>10/5/0</u>	12/3/0
MPENSGAII-C-MS (GRBF)	3/4/8	8/6/1	0/15/0	6/1/8	<u>1/14/0</u>	<u>10/5/0</u>	12/3/0
MPDENN2-E (GRBF)	<u>10/5/0</u>	0/3/12	0/15/0	9/6/0	0/10/5	0/7/8	0/11/4
MPDENN2-MS (GRBF)	0/6/9	<u>10/5/0</u>	0/15/0	0/4/11	<u>1/14/0</u>	0/7/8	0/11/4
MPDENN2-C-E (GRBF)	9/4/2	2/5/8	0/15/0	8/6/1	0/15/0	0/7/8	0/8/7
MPDENN2-C-MS (GRBF)	0/7/8	8/4/3	0/15/0	0/7/8	0/15/0	0/7/8	0/11/4
MPENSGAII-E (SU)	<u>10/5/0</u>	0/5/10	0/15/0	<u>10/5/0</u>	0/15/0	8/3/4	0/11/4
MPENSGAII-MS (SU)	0/5/10	<u>10/5/0</u>	0/15/0	0/4/11	0/15/0	8/5/2	1/10/4
MPENSGAII-C-E (SU)	8/2/5	5/2/8	0/15/0	8/4/3	0/15/0	8/7/0	<u>2/9/4</u>
MPENSGAII-C-MS (SU)	4/3/8	8/3/4	0/15/0	3/3/9	0/15/0	8/3/4	1/10/4
MPDENN2-E (SU)	11/4/0	0/4/11	0/15/0	<u>10/5/0</u>	0/14/1	0/7/8	0/11/4
MPDENN2-MS (SU)	0/5/10	9/5/1	0/15/0	0/4/11	0/14/1	0/7/8	0/11/4
MPDENN2-C-E (SU)	<u>10/5/0</u>	1/4/10	0/15/0	8/6/1	0/15/0	0/7/8	0/11/4
MPDENN2-C-MS (SU)	0/7/8	8/6/1	0/15/0	0/5/10	0/15/0	0/7/8	0/10/5

n° Coef. = número de coeficientes del modelo; n° N.O. = número de neuronas en la capa oculta.

Los mejores resultados están marcadas en **negrita** y los segundos mejores en subrayado.

determinar cual de los primeros 5 pacientes en lista de espera (ordenados de forma descendente por su valor de MELD) recibirá el órgano. Este sistema basado en reglas puede llegar a ser una herramienta útil para los expertos médicos a la hora de asignar un órgano a un determinado receptor. Las entradas de dicho sistema son las respuestas de los dos mejores modelos de ANN frente a un determinado par Donante-Receptor (DR) (cada par DR está compuesto por la información de un receptor, de un donante/órgano y de los factores preoperatorios). El modelo obtenido con el algoritmo MPENSGAII-E (GRBF) es empleado para estimar la probabilidad de pertenencia a la clase "Supervivencia del injerto", mientras que el modelo obtenido con el algoritmo MPENSGAII-MS (GRBF) se usa para calcular la probabilidad de pertenencia a la clase 'No supervivencia del injerto'. Destacar que estas dos probabilidades no son complementarias, debido a que son obtenidas mediante modelos diferentes.

El sistema basado en reglas está formado por cinco reglas y dos precondiciones. Estas precondiciones son usadas para determinar si existen diferencias significativas entre las diferentes predicciones. El rango de significación ha sido calculado a partir de la siguiente hipótesis:

$$\left. \begin{array}{l} H_0 : p = p_0 \\ H_1 : p \neq p_0 \end{array} \right\}$$

El intervalo de confianza asociado con esta hipótesis es:

$$CCR_0 = \left(p_0 - z_{\alpha/2} \frac{\sqrt{p_0 q_0}}{\sqrt{n}}; p_0 + z_{\alpha/2} \frac{\sqrt{p_0 q_0}}{\sqrt{n}} \right).$$

Si $\hat{p} \in C_0 \Rightarrow$ se acepta la hipótesis nula, es decir, se acepta que el valor de p es igual a p_0 .

Considerando un valor de $\alpha = 0,10$ y $z_{\alpha/2} = 1,645$ y bajo la hipótesis de que $p_0 = 0,8363$ (valor medio de CCR obtenido por el algoritmo MPENSGAII-E (GRBF)), el intervalo de confianza resultante para la métrica CCR es:

$$CCR_0 = (0,7892, 0,8834),$$

donde:

$$z_{\alpha/2} \frac{\sqrt{p_0 q_0}}{\sqrt{n}} = 0,0471.$$

A partir de este intervalo, se puede deducir que las probabilidades obtenidas por dos pares DR no son significativamente diferentes si la diferencia entre ellas es menor que 0,0471, es decir, si $p_1^* - p_2^* \in (0 \pm 0,0471)$.

El intervalo de confianza para la medida MS, bajo la hipótesis de que $p_0^* = 0,5204$ (valor medio de MS obtenido por el algoritmo MPENSGAII-MS (GRBF)) es:

$$MS_0 = (0,4623, 0,5785),$$

donde:

$$z_{\alpha/2} \frac{\sqrt{p_0^* q_0^*}}{\sqrt{n}} = 0,0581.$$

Para este intervalo se deduce que dos probabilidades no son significativamente diferentes si la diferencia entre ellas es menor que 0,0581, es decir, si $p_1^* - p_2^* \in (0 \pm 0,0581)$.

Una vez obtenidos estos valores, las precondiciones pueden ser definidas de la siguiente manera:

- **SI** la diferencia en las probabilidades (diferencias significativas) de pertenecer a la clase "Supervivencia del injerto" de dos pares DR obtenidas con el modelo MPENSGAII-E (GRBF) es menor que 4,71 (dado en %),
ENTONCES se considera que estos pares tienen la misma probabilidad de pertenecer a dicha clase.
- **SI** la diferencia en las probabilidades (diferencias significativas) de pertenecer a la clase "No supervivencia del injerto" de dos pares DR obtenidas con el modelo MPENSGAII-MS (GRBF) es menor que 5,81 (dado en %),
ENTONCES se considera que estos pares tienen la misma probabilidad de pertenecer a dicha clase.

Una vez que han sido definidas las precondiciones, las 5 reglas del sistema pueden ser presentadas. Mediante estas reglas, el sistema puede ofrecer una respuesta ante cualquier situación. Las 4 primeras reglas se aplican secuencialmente para cada par DR, empezando con el par formado por el receptor con mayor valor de MELD. Finalmente, si no se ha cumplido ninguna de estas reglas con ninguno de los 5 pares DR, se aplicará la quinta regla.

- **SI** un par DR tiene la probabilidad más alta de pertenecer a la clase “Supervivencia del injerto” con el modelo MPENSGAII-E (GRBF) y **SI** tiene la probabilidad menor de pertenecer a la clase “No supervivencia del injerto” con el modelo MPENSGAII-MS (GRBF),
ENTONCES se selecciona este par.
- **SI** un par DR tiene la probabilidad más alta de pertenecer a la clase “Supervivencia del injerto” con el modelo MPENSGAII-E (GRBF) y **SI** no existen diferencias significativas con la probabilidad menor de pertenecer a la clase “No supervivencia del injerto” con el modelo MPENSGAII-MS (GRBF),
ENTONCES se selecciona este par.
- **SI** un par DR no tiene diferencias significativas con el par con la probabilidad más alta de pertenecer a la clase “Supervivencia del injerto” con el modelo MPENSGAII-E (GRBF) y **SI** tiene la probabilidad menor de pertenecer a la clase “No supervivencia del injerto” con el modelo MPENSGAII-MS (GRBF),
ENTONCES se selecciona este par.
- **SI** un par DR no tiene diferencias significativas con el par con la probabilidad más alta de pertenecer a la clase “Supervivencia del injerto” con el modelo MPENSGAII-E (GRBF) y **SI** no existen diferencias significativas con la probabilidad menor de pertenecer a la clase “No supervivencia del injerto” con el modelo MPENSGAII-MS (GRBF),
ENTONCES se selecciona este par.
- **SI** un par DR tiene la probabilidad más alta de pertenecer a la clase “Supervivencia del injerto” con el modelo MPENSGAII-E (GRBF) o no tiene diferencias significativas con el par con mayor probabilidad, y **SI** otro par DR distinto tiene la probabilidad menor de pertenecer a la clase “No supervivencia del injerto” con el modelo MPENSGAII-MS (GRBF) o no tiene diferencias significativas con el par con menor probabilidad,
ENTONCES se selecciona el par cuyo receptor tenga el mayor valor de MELD.

6.4.1 Ejemplo de Aplicación del Sistema Basado en Reglas

El objetivo del siguiente ejemplo es facilitar la comprensión del sistema propuesto y observar su aplicación en una situación real. Para este ejemplo, se han seleccionado 5 receptores al azar de la base de datos, con valores de MELD comprendidos entre 20 y 27. Este rango se corresponde con los valores más comunes de MELD. Las características de los receptores seleccionados pueden verse en la Tabla 6.5. De igual modo, se han seleccionado aleatoriamente 20 potenciales donantes de la base de datos, con los que se analizará la respuesta de los receptores. Diez de estos 20 donantes no poseen criterios expandidos, es decir, no son casos extremos o no presentan problemas serios. Por otro lado, los otros diez donantes si poseen criterios expandidos. La Tabla 6.6 muestra las características de los donantes empleados en el ejemplo.

Tabla 6.5: Características de los receptores empleados en el ejemplo.

	E	D	S	IMC	DP	HTA	DI	EP	H	TP	TLE	MI	MT	TIP	SH	CAS
Rec1 (27)	60	0	0	26	0	0	0	1	0	0	143	27	27	0	0	0
Rec2 (24)	41	1	0	28	0	1	0	2	1	0	84	24	24	0	0	0
Rec3 (22)	59	0	0	26,57	0	0	0	6	1	0	59	22	22	0	0	1
Rec4 (21)	58	0	1	21,63	0	1	1	0	0	0	465	21	21	0	0	0
Rec5 (20)	58	1	1	29,14	2	0	1	6	0	0	264	20	20	0	0	0

Abreviaturas: E: edad; D: diabetes; S: sexo; IMC: índice de masa corporal; DP: diagnóstico principal; HTA: hipertensión arterial; DI: diálisis; EP: etiología principal; H: hepatocarcinoma; TP: trombosis portal; TLE: tiempo en lista de espera; MI: MELD inicial; MT: MELD en el trasplante; TIP: derivación portosistémica intrahepática transyugular; SH: síndrome hepatorenal; CAS: cirugía abdominal superior.

La Tabla 6.7 presenta tanto las probabilidades de cada par DR de pertenecer a la clase "Supervivencia del injerto" empleando el modelo MPENSGAII-E (GRBF) como las probabilidades de pertenecer a la clase "No supervivencia del injerto" calculadas mediante el modelo MPENSGAII-MS (GRBF). A partir de estas probabilidades, el sistema basado en reglas selecciona el receptor más favorable para el órgano ofertado.

A partir de los resultados mostrados en la Tabla 6.7 se observa que el receptor 1 recibe el órgano 3 veces al considerar los donantes sin criterios expandidos y 7 veces al considerar los donantes con criterios expandidos (3 + 7), el receptor 2 recibe el órgano en (4 + 2) casos, el receptor 3 no recibe el órgano en ningún caso (0 + 0), el receptor 4 recibe el órgano (2 + 0) veces y el receptor 5 lo recibe en (1 + 1) ocasiones. Según estos resultados, el sistema se comporta de manera similar considerando o sin considerar donantes con criterios expandidos, por lo que se puede afirmar que el sistema trabaja de manera eficaz en cualquier situación. Es importante tener en cuenta que el sistema basado en reglas ha sido diseñado para complementar el sistema de asignación MELD, pero puede seleccionar candidatos con menor valor de MELD

Tabla 6.6: Características de los donantes/órganos y factores preoperatorios empleados en el ejemplo.

	E	S	IMC	CE	D	HTA	UCI	EH	I	C	NA	AST	ALT	BT	AHB	HCV	SP	A	DM	THD	TC	IC/P	TIF	ABO	LI/R
Don1	55	0	27,78	1	0	0	3	0	0	0,7	146	11	19	0,5	1	0	1	0	1	0	0	0	0	1	0
Don2	34	1	26,22	0	0	0	3	0	0	0,8	151	17	14	0,3	1	0	1	0	1	0	0	0	1	1	1
Don3	63	1	29,38	1	0	1	1	0	0	1	160	29	16	0,7	1	0	1	0	1	0	0	0	1	1	1
Don4	52	0	24,22	1	0	1	1	0	0	5,3	131	13	21	0,5	1	0	0	0	1	0	0	0	0	1	0
Don5	25	0	25,93	0	0	0	2	0	0	1	181	19	31	0,4	0	0	0	0	1	0	0	0	1	1	1
Don6	67	1	27,77	0	0	0	3	0	0	0,96	148	62	53	0,89	1	0	0	0	1	1	1	0	0	1	0
Don7	31	1	25,39	1	0	0	1	0	0	0,7	187	51	37	0,33	1	0	0	0	1	0	0	0	0	1	0
Don8	60	0	24,84	1	0	1	1	0	0	1	146	51	30	0,5	1	0	1	0	1	0	0	0	1	1	0
Don9	57	0	26,12	1	1	1	2	0	0	2,6	135	126	236	0,67	0	0	1	0	1	0	0	0	0	1	0
Don10	35	0	29,3	1	0	0	0	0	0	0,73	153	16	20	1,92	1	0	1	0	1	0	0	0	1	1	0
Don11	81	1	31,25	1	0	0	5	0	1	1	141	43	43	0,3	1	0	1	0	0	0	0	0	1	0	0
Don12	79	1	29,52	1	0	0	8	0	1	1	159	24	26	0,3	1	0	1	0	1	0	0	0	0	1	0
Don13	78	1	41,12	1	0	0	8	0	0	0,85	159	42	30	0,3	1	0	1	0	1	0	0	0	1	1	0
Don14	76	1	31,11	1	1	1	1	0	1	0,6	138	40	15	0,8	1	0	1	0	0	0	0	0	1	1	0
Don15	67	0	31,14	1	0	1	30	0	1	1,1	158	363	300	2,6	1	0	1	0	1	0	0	0	0	1	1
Don16	35	0	31,14	0	0	0	6	0	1	0,8	150	105	80	0,7	1	0	1	0	1	0	0	0	2	1	0
Don17	42	0	29,39	1	0	1	13	1	1	4,4	156	82	66	4,2	1	0	1	0	1	0	0	0	2	1	1
Don18	83	0	31,25	1	0	1	2	0	1	0,8	149	53	97	0,7	1	0	1	0	0	0	0	0	1	1	0
Don19	68	1	31,11	1	0	1	9	0	1	0,6	149	27	20	0,26	0	0	1	0	1	0	0	0	1	1	0
Don20	85	1	27,06	1	0	0	0	0	0	1	143	1	9	1,2	1	0	1	0	0	0	0	0	2	1	1

Abreviaturas: E: edad; S: sexo; IMC: índice de masa corporal; CE: causa de exitus; D: diabetes; HTA: hipertensión arterial; UCI: días de hospitalización en la unidad de cuidados intensivos; EH: episodios de hipotensión > 1h < 60mmHg; I: alto uso de drogas inotrópicas; C: concentración plasmática de creatinina; NA: concentración plasmática de sodio; AST: nivel de aspartato transaminasa; ALT: nivel plasmático aminotransferasa alanina; BT: bilirrubina total; AHB: hepatitis B (Ab positivo); HCV: hepatitis C (serología positiva); SP: solución de preservación; A: asistolia; DM: donante multiorgánico; THD: trasplante de hígado dividido; TC: trasplante combinado; IC/P: injerto completo o parcial; TIF: tiempo de isquemia fría; ABO: incompatibilidad ABO; LI/R: grado de la lesión por isquemia-reperusión.

cuando las probabilidades de pertenecer a una clase u otra son significativamente favorables.

6.5 OTROS ESTUDIOS

En esta Sección se enumeran otros estudios realizados relacionados con la mejor asignación del órgano en los trasplantes hepáticos.

- La prevalencia del injerto después de 3 meses desde la operación es estudiada en [18]. La metodología seguida en este estudio es similar a la presentada en este Capítulo. El principal aporte de este trabajo consiste en la comparación, mediante las curvas ROC, del desempeño del sistema basado en reglas con los resultados obtenidos por otras metodologías de asignación empleadas en los

Tabla 6.7: Ejemplo de aplicación del sistema basado en reglas.

Donantes sin criterios expandidos										
Modelo MPENSGAII-E (GRBF) - Probabilidad de aceptación en %										
	Don1	Don2	Don3	Don4	Don5	Don6	Don7	Don8	Don9	Don10
Rec1(27)	90,16	92,52	<u>97,34</u>	78,84	64,15	75,84	80,99	<u>96,52</u>	87,98	<u>93,52</u>
Rec2(24)	<u>95,74</u>	<u>96,50</u>	<u>98,02</u>	91,67	84,63	90,42	<u>92,51</u>	<u>97,77</u>	<u>95,02</u>	<u>96,82</u>
Rec3(22)	90,35	92,66	<u>97,37</u>	79,24	64,70	76,28	81,36	<u>96,57</u>	88,22	<u>93,64</u>
Rec4(21)	<u>96,61</u>	<u>97,12</u>	<u>98,12</u>	<u>93,88</u>	88,98	<u>93,03</u>	<u>94,44</u>	<u>97,96</u>	<u>96,13</u>	<u>97,33</u>
Rec5(20)	97,76	97,93	98,24	96,83	95,13	96,54	97,03	98,19	97,60	98,00
Modelo MPENSGAII-MS (GRBF) - Probabilidad de rechazo en %										
Rec1(27)	<u>3,59</u>	<u>3,29</u>	<u>1,24</u>	<u>5,33</u>	9,67	11,57	<u>5,61</u>	<u>1,63</u>	<u>3,84</u>	<u>2,82</u>
Rec2(24)	<u>2,28</u>	<u>2,11</u>	<u>0,94</u>	<u>3,27</u>	<u>5,83</u>	<u>7,00</u>	<u>3,43</u>	<u>1,17</u>	<u>2,42</u>	<u>1,85</u>
Rec3(22)	<u>2,98</u>	<u>2,74</u>	<u>1,10</u>	<u>4,36</u>	7,89	9,48	<u>4,59</u>	<u>1,41</u>	<u>3,17</u>	<u>2,36</u>
Rec4(21)	<u>1,61</u>	<u>1,51</u>	<u>0,78</u>	<u>2,21</u>	<u>3,77</u>	<u>4,50</u>	<u>2,31</u>	<u>0,93</u>	<u>1,70</u>	<u>1,35</u>
Rec5(20)	1,06	1,01	0,65	1,35	2,08	2,43	1,39	0,72	1,10	0,93
Decisión del sistema basado en reglas										
	Rec2	Rec2	Rec1	Rec4	Rec5	Rec4	Rec2	Rec1	Rec2	Rec1
	Regla4	Regla4	Regla4	Regla4	Regla1	Regla4	Regla4	Regla4	Regla4	Regla4
Donantes con criterios expandidos										
Modelo MPENSGAII-E (GRBF) - Probabilidad de aceptación en %										
	Don11	Don12	Don13	Don14	Don15	Don16	Don17	Don18	Don19	Don20
Rec1(27)	<u>96,96</u>	<u>95,29</u>	<u>96,27</u>	<u>97,52</u>	<u>96,70</u>	69,69	86,51	<u>97,54</u>	<u>96,83</u>	81,88
Rec2(24)	<u>97,91</u>	<u>97,38</u>	<u>97,69</u>	<u>98,08</u>	<u>97,83</u>	87,59	<u>94,51</u>	<u>98,08</u>	<u>97,87</u>	<u>92,85</u>
Rec3(22)	<u>96,99</u>	<u>95,37</u>	<u>96,32</u>	<u>97,54</u>	<u>96,74</u>	70,20	86,79	<u>97,56</u>	<u>96,86</u>	82,24
Rec4(21)	<u>98,05</u>	<u>97,70</u>	<u>97,91</u>	<u>98,16</u>	<u>98,00</u>	91,07	<u>95,79</u>	<u>98,16</u>	<u>98,02</u>	<u>94,67</u>
Rec5(20)	98,22	98,12	98,18	98,24	98,20	95,87	97,48	98,25	98,21	97,10
Modelos MPENSGAII-MS (GRBF) - Probabilidad de rechazo en %										
Rec1(27)	<u>2,32</u>	<u>2,24</u>	<u>2,21</u>	<u>1,76</u>	<u>1,38</u>	<u>5,73</u>	<u>2,22</u>	<u>1,70</u>	<u>1,71</u>	<u>4,96</u>
Rec2(24)	<u>1,56</u>	<u>1,52</u>	<u>1,50</u>	<u>1,25</u>	<u>1,02</u>	<u>3,50</u>	<u>1,51</u>	<u>1,21</u>	<u>1,21</u>	<u>3,06</u>
Rec3(22)	<u>1,96</u>	<u>1,90</u>	<u>1,88</u>	<u>1,52</u>	<u>1,21</u>	<u>4,69</u>	<u>1,88</u>	<u>1,47</u>	<u>1,48</u>	<u>4,07</u>
Rec4(21)	<u>1,17</u>	<u>1,14</u>	<u>1,13</u>	<u>0,97</u>	<u>0,83</u>	<u>2,35</u>	<u>1,14</u>	<u>0,95</u>	<u>0,96</u>	<u>2,08</u>
Rec5(20)	0,84	0,83	0,82	0,74	0,67	1,41	0,82	0,73	0,73	1,28
Decisión del sistema basado en reglas										
	Rec1	Rec1	Rec1	Rec1	Rec1	Rec5	Rec2	Rec1	Rec1	Rec2
	Regla4	Regla4	Regla4	Regla4	Regla4	Regla5	Regla4	Regla4	Regla4	Regla4

La mejor probabilidad está marcada en **negrita** y las probabilidades sin diferencias significativas en subrayado.

trasplantes hepáticos (MELD, D-MELD, DRI, P-SOFT, SOFT y BAR). Los resultados muestran que, en la mayor parte de la curva ROC, el sistema propuesto es el mejor método de asignación.

- En [37] también se estudia la supervivencia del injerto una vez transcurridos 3 meses desde la operación. En este caso, el sistema basado en reglas ha sido modificado para recibir como entradas el error medio producido por dos conjuntos de modelos de redes neuronales, en lugar de recibir las salidas de dos modelos. Uno de estos conjuntos está formado por los individuos correspondientes al extremo superior de los frentes de Pareto obtenidos en cada una de las iteraciones y el otro conjunto por los individuos correspondientes al extremo inferior. Este sistema basado en reglas ha sido probado con 4 conjuntos de receptores: receptores con valor de MELD entre 20 y 23 (receptores favorables), con valores entre 24 y 26 (receptores normales), con valores entre 28 y 39 (receptores graves) y todos los receptores con el mismo valor de MELD (MELD=27).
- En [99] se han analizado aquellos pacientes en los que el injerto no sobrepasa los 3 meses de duración. Los datos se han dividido en dos clases: si el fallo del injerto se produce antes de que transcurran 15 días desde la operación o si el fallo se produce antes de 90 días. Los modelos resultantes en este estudio pueden ser empleados junto con los modelos obtenidos en otros estudios, para determinar con mayor precisión cuando se producirá el fallo del injerto, es decir, primero se usarían los modelos de los estudios previos y, para aquellos casos en los que se determine que el injerto no sobrevivirá más de 3 meses, se emplearán los modelos de este estudio para predecir si el fallo se producirá en las dos primeras semanas o no.
- La extensión de este sistema de asignación español ha sido presentada en [101]. En este trabajo se emplean datos de 822 trasplantes hepáticos realizados en el hospital King's College de Londres entre 2002 y 2010. Estos datos se han estudiado conjuntamente con los datos de 615 trasplantes realizados en 7 hospitales españoles (se han seleccionado sólo 7 hospitales de los 11 que componían la base de datos original (descrita en la Sección 6.1) con el objetivo de mejorar la calidad de la base de datos conjunta). En este análisis se han considerado 4 clases: pérdida del injerto antes de 15 días tras la operación (clase 1), pérdida del injerto ocurrida entre 15 días y 3 meses (clase 2), pérdida del injerto producida entre 3 meses y 1 año (clase 3), y supervivencia del injerto de, al menos, 1 año (clase 4). La metodología seguida posee un carácter ordinal, ya que se considera la relación de orden entre las clases del problema (en el Capítulo 9 se muestra más información sobre clasificación ordinal). La técnica de asignación presentada en este trabajo predice la clase de pertenencia de los 5 pares formados por el órgano disponible y los 5 primeros pacientes en lista de espera para, finalmente, asignar el órgano al paciente cuya clase predicha sea mayor. En el caso de que existen dos o más pares con la misma clase predicha (empatan en la predicción), el órgano es asignado al receptor empatado con mayor valor de MELD.

RESOLUCIÓN DE UN PROBLEMA REAL DE TELEDETECCIÓN REMOTA MEDIANTE UN ALGORITMO MULTIOBJETIVO

Las administraciones andaluza y europea dictan subvenciones para aquellos agricultores que desarrollen una agricultura de conservación, esto es, que reduzcan la erosión en los campos de olivos protegiendo el suelo existente entre filas con cubiertas vegetales (vivas o inertes). El objetivo de este estudio es desarrollar modelos que permitan ayudar a las administraciones, a partir de imágenes aéreas tomadas con sistemas de teledetección, a decidir qué fincas cumplen los requisitos para obtener la subvención.

Para desarrollar estos modelos se han empleado dos bases de datos: una con los datos obtenidos en primavera y otra con los obtenidos en verano. Los datos recogidos en estas bases de datos han sido tomados en dos fincas de olivos de dos localidades de Córdoba (Andalucía): Espejo y Montilla. Una imagen aérea de estas fincas puede verse en la Figura 7.1. En estas bases de datos, los patrones pertenecen a una de estas tres clases: olivo, suelo desnudo o cubiertas vegetales. En la figura 7.2 se observan cuatro imágenes de los diferentes elementos a discriminar. En primavera, las variables de entrada se corresponden con los valores de azul, verde y rojo de los píxel de la imagen (valores RGB). En verano, las variables de entrada son las mismas que en primavera, incluyendo, además, el valor de infrarrojo de cada píxel.

7.1 DISEÑO EXPERIMENTAL

En este estudio se han empleado los algoritmos PDENN y NSGAI y los algoritmos meméticos MPDENN2 y MPENSGAI para el entrenamiento de modelos de red neuronal de base sigmoide, junto con el algoritmo MLP (*Multilayer Perceptron*) proporcionado por Weka [129]. Este estudio puede encontrarse en [36].

La Tabla 7.1 muestra las características de cada una de las base de datos utilizadas: número total de patrones en cada conjunto de datos, número de patrones en entrenamiento y en generalización, número de variables de entrada, número de clases, número total de patrones por clase y valor de p^* (porcentaje de patrones de la clase minoritaria).

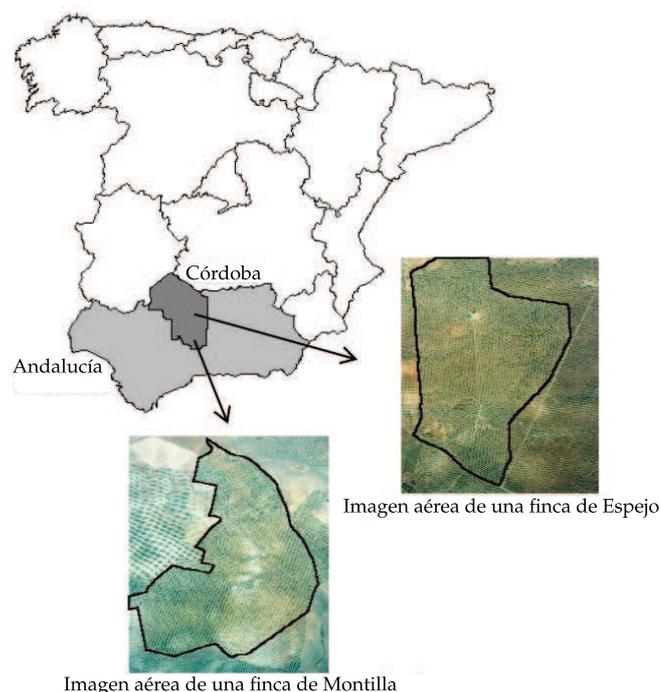


Figura 7.1: Imágenes aéreas de las fincas de Espejo y Montilla.

Tabla 7.1: Características de las bases de datos Primavera y Verano.

Base de datos	Total patrones	Patrones entrenam.	Patrones generali.	Variables de entrada	Número de clases	Patrones por clase	p^*
Primavera	6000	3000	3000	3	3	2000-2000-2000	0,3333
Verano	6000	3000	3000	4	3	2000-2000-2000	0,3333

El diseño experimental seguido es un proceso de validación cruzada (*holdout*) con 30 ejecuciones, donde los conjuntos de entrenamiento, tanto en primavera como en verano, están formados por los datos obtenidos en una finca de Espejo y los conjuntos de generalización por los datos obtenidos en una finca de Montilla. Esto es así porque en la finca de Espejo existe mayor diversidad de tipos de cubiertas vegetales, lo cual proporciona un marco excepcional para el entrenamiento de los modelos. La generalización se hace con los datos de la finca de Montilla, pero se podría haber hecho con los datos de cualquier otra finca, ya que lo que se pretende es que el modelo obtenido clasifique bien cualquier finca, para que sea de ayuda a las administraciones andaluza y europea.

La evolución de los cuatro algoritmos evolutivos es guiada por la entropía (E) y la mínima sensibilidad (MS). Con el objetivo de acotar las dos métricas en el intervalo



Figura 7.2: Imágenes de los diferentes elementos a discriminar: olivos, suelo desnudo y cubiertas vegetales.

$[0, 1]$, se ha aplicado sobre la medida E la siguiente transformación estrictamente decreciente $A = 1/(1 + E)$. Con esto se definen las dos funciones a maximizar por los diferentes MOEAs.

Los parámetros empleados por todos los algoritmos son los mismos para las dos bases de datos. Los algoritmos PDENN y MPDENN2 tienen un tamaño de población de 50 individuos, una probabilidad de cruce de $pro_cruce = 0,8$ y una de mutación de $pro_mut = 0,1$, siguiendo las directrices de H. Abbass [4]. Además, en el algoritmo MPDENN2 se aplica la búsqueda local 3 veces durante la ejecución (cada 33,33 % de las generaciones). Para los algoritmos NSGAI y MPENSGAI se emplea un tamaño de población de 150 individuos y una probabilidad de mutación para cada

operador de $1/5$, siguiendo las directrices de K. Deb [41] (el Capítulo 4 contiene más información sobre estos algoritmos y sus parámetros). El algoritmo MLP disponible en Weka se ha utilizado con los valores por defecto.

Para el algoritmo de búsqueda local $iRprop^+$, se utilizan los parámetros recomendados por C. Igel [76]: $\eta^- = 0,5$, $\eta^+ = 1,2$, $\Delta_0 = 0,0125$, $\Delta_{\min} = 0$ y $\Delta_{\max} = 50$. Para los algoritmos PDENN y MPDENN2 se ejecutan 5 iteraciones (Epochs = 5) y para los algoritmos NSGAI y MPENSGAI 10 (Epochs = 10).

Para iniciar el procesamiento de datos, cada una de las variables de entrada se escalaron en el intervalo $[-1, 1]$, para evitar así la saturación de la señal.

7.2 RESULTADOS

La Tabla 7.2 presenta los resultados obtenidos sobre el conjunto de entrenamiento (finca de Espejo) y la Tabla 7.3 aquellos obtenidos en generalización (finca de Montilla). Los resultados mostrados se corresponden con el valor medio obtenido tras realizar las 30 ejecuciones. En cada ejecución se seleccionan los modelos situados en los dos extremos del frente de Pareto (la Sección 4.5 muestra más información sobre las técnicas de selección de individuos). Los modelos con mayor valor en A (extremo superior del frente) se han etiquetado como *NombreMétodo-E*, y aquellos con el mayor valor en MS (extremo inferior del frente) como *NombreMétodo-MS*.

Se puede observar que el mejor método en entrenamiento es el algoritmo MLP, con valores de CCR, MS y K superiores al 92 %, 90 % y 0,8903 en Primavera y al 83 %, 70 % y 0,7514 en Verano, respectivamente. El resto de algoritmos obtienen unos resultados inferiores (Tabla 7.2). En contraste, el mejor rendimiento en generalización es obtenido por los algoritmos MPDENN2-MS y MPDENN2-E, con valores de CCR, MS y K de 81,28 %, 53,71 % y 0,7193 en Primavera y de 86,78 %, 70,18 % y 0,8017 en Verano, respectivamente (Tabla 7.3). Esto es debido a que el algoritmo MLP se ajusta demasiado a los datos de entrenamiento y no consigue generalizar bien, en cambio en el algoritmo MPDENN2 no se produce este sobreajuste, por lo que presenta mayor capacidad de generalización.

En la Figura 7.3 se muestran los resultados gráficos obtenidos por el algoritmo MPDENN2 en las bases de datos Primavera y Verano en entrenamiento en el espacio bidimensional (MS, A) y en generalización en el espacio (MS, CCR). Para el espacio (MS, A), se ha seleccionado el frente de Pareto de la ejecución que presenta el mejor valor en A dentro de las 30 realizadas. En las gráficas de generalización se muestran los valores de MS y CCR obtenidos por los individuos de la población sobre el conjunto de generalización. Cabe destacar que en generalización, los individuos no forman, necesariamente, frentes de Pareto y que los individuos que en entrenamiento estaban localizados en el primer frente de Pareto, ahora pueden estar situados en una región peor del espacio (MS, CCR). En general, la estructura del frente de Pareto obtenido en entrenamiento no se mantiene en generalización.

Tabla 7.2: Resultados obtenidos en entrenamiento (finca de Espejo): media y desviación típica ($Media_{DT}$).

Algoritmo	CCR	MS	K
Primavera			
PDENN-E	88,01 _{1,48}	82,83 _{3,34}	0,8201 _{0,0223}
PDENN-MS	88,52 _{1,67}	87,84 _{1,73}	0,8278 _{0,0251}
MPDENN2-E	87,69 _{1,15}	82,87 _{3,02}	0,8153 _{0,0173}
MPDENN2-MS	87,98 _{1,72}	87,25 _{1,73}	0,8197 _{0,0259}
NSGAI-E	87,33 _{1,09}	80,91 _{4,23}	0,8101 _{0,0165}
NSGAI-MS	89,34 _{1,56}	88,97 _{1,62}	0,8401 _{0,0235}
MPENSGAI-E	88,85 _{1,02}	84,56 _{1,73}	0,8328 _{0,0154}
MPENSGAI-MS	90,36 _{1,22}	89,95 _{1,29}	0,8554 _{0,0183}
MLP	92,69_{0,34}	90,10_{2,03}	0,8903_{0,0051}
Verano			
PDENN-E	81,32 _{0,95}	69,91 _{2,40}	0,7198 _{0,0143}
PDENN-MS	81,73 _{1,04}	<u>73,04_{1,18}</u>	0,7260 _{0,0156}
MPDENN2-E	81,81 _{0,90}	70,85 _{2,00}	<u>0,7272_{0,0135}</u>
MPDENN2-MS	81,75 _{2,10}	73,49_{1,27}	0,7263 _{0,0315}
NSGAI-E	77,35 _{2,23}	57,78 _{10,06}	0,6603 _{0,0335}
NSGAI-MS	79,12 _{2,19}	70,44 _{1,66}	0,6869 _{0,0329}
MPENSGAI-E	80,03 _{1,30}	65,12 _{5,70}	0,7004 _{0,0195}
MPENSGAI-MS	80,22 _{1,38}	71,58 _{0,92}	0,7033 _{0,0208}
MLP	83,43_{0,44}	70,31 _{4,42}	0,7514_{0,0067}

Los mejores resultados están marcadas en **negrita** y los segundos mejores en subrayado.

Con el fin de determinar si existen diferencias significativas en los resultados obtenidos para las métricas CCR y MS con los nueve algoritmos empleados sobre los conjuntos de generalización (en Primavera y en Verano), se han aplicado diversos tests estadísticos. Estos tests no han sido aplicados sobre la métrica K debido a que esta medida presenta un comportamiento similar a la medida CCR. En primer lugar, con el test estadístico de Kolmogorov-Smirnov (K-S) se llegó a la conclusión de que las distribuciones para las métricas CCR y MS siguen una distribución normal con un nivel de significación asintótica estándar ($\alpha = 0,05$). A continuación, se ha aplicado un test de Levene para evaluar si existen diferencias significativas en las varianzas, pudiendo concluir que las diferencias si son significativas en todos los caso salvo para la métrica CCR en Primavera (segunda columna de la Tabla 7.4). Sobre la base de esta hipótesis, se ha realizado un test de análisis de la varianza de un factor (*ANalysis Of the VAriance of one factor*, ANOVA I), donde el factor consiste en el algoritmo aplicado y las variables de control son los valores de CCR y MS obtenidos en los conjuntos de generalización (tanto en Primavera como en Verano). En la tercera columna de la Tabla 7.4 se muestran los valores de significación del test F de Snedecor. Según estos

Tabla 7.3: Resultados obtenidos en generalización (finca de Montilla): media y desviación típica (Media_{DT}).

Algoritmo	CCR	MS	K
Primavera			
PDENN-E	78,75 _{3,17}	46,89 _{8,31}	0,6812 _{0,0476}
PDENN-MS	80,69 _{2,37}	<u>50,46_{9,11}</u>	0,7104 _{0,0356}
MPDENN2-E	78,30 _{3,05}	49,65 _{7,93}	0,6745 _{0,0458}
MPDENN2-MS	81,28_{4,43}	53,71_{13,20}	0,7193_{0,0665}
NSGA2-E	79,31 _{2,51}	49,91 _{8,50}	0,6897 _{0,0378}
NSGA2-MS	81,01 _{2,47}	50,18 _{8,21}	0,7152 _{0,0371}
MPENSGAII-E	80,21 _{1,34}	49,18 _{4,90}	0,7032 _{0,0206}
MPENSGAII-MS	<u>81,17_{3,75}</u>	<u>50,09_{7,72}</u>	<u>0,7176_{0,0562}</u>
MLP	76,05 _{6,36}	33,26 _{5,23}	0,6408 _{0,0955}
Verano			
PDENN-E	82,07 _{10,30}	55,11 _{29,73}	0,7311 _{0,1546}
PDENN-MS	79,10 _{12,53}	55,75 _{28,86}	0,6865 _{0,1879}
MPDENN2-E	86,78_{9,14}	70,18_{23,16}	0,8017_{0,1371}
MPDENN2-MS	<u>83,03_{12,52}</u>	<u>63,80_{24,74}</u>	<u>0,7455_{0,1878}</u>
NSGA2-E	75,36 _{12,80}	35,99 _{34,62}	0,6304 _{0,1920}
NSGA2-MS	70,78 _{11,13}	34,20 _{27,67}	0,5618 _{0,1669}
MPENSGAII-E	79,94 _{10,06}	46,68 _{31,98}	0,6991 _{0,1509}
MPENSGAII-MS	77,55 _{10,41}	48,62 _{27,25}	0,6633 _{0,1561}
MLP	76,05 _{2,80}	56,67 _{17,70}	0,6408 _{0,0420}

Los mejores resultados están marcadas en **negrita** y los segundos mejores en subrayado.

valores, podemos concluir que existen diferencias significativas en media tanto para la métrica CCR como para MS ($p\text{-valor} < 0,05$).

Para ambas bases de datos, hemos empleado un test paramétrico de comparaciones múltiples con el fin de clasificar las medias de CCR y MS obtenidas por cada uno de los algoritmos. El test de Tukey o Tamhane ha sido aplicado y sus resultados se muestran en la cuarta columna de la Tabla 7.4, en función de que las varianzas sean iguales o distintas. Esta columna primero muestra el orden de las medias de los nueve algoritmos y después las comparaciones de pares que presentan diferencias significativas.

Finalmente se incluyen en la Tabla 7.4 los $p\text{-valores}$ de un test t de Student comparando el algoritmo MPDENN2-MS con el algoritmo MPENSGAII-MS y con el algoritmo PDENN-MS en Primavera y el algoritmo MPDENN2-E con el algoritmo MPDENN2-MS en Verano (en este último caso se aplica un test t de Student con dependencias, "Test t dep."). Se han seleccionado estos pares porque son los formados por los mejores algoritmos en cada caso.

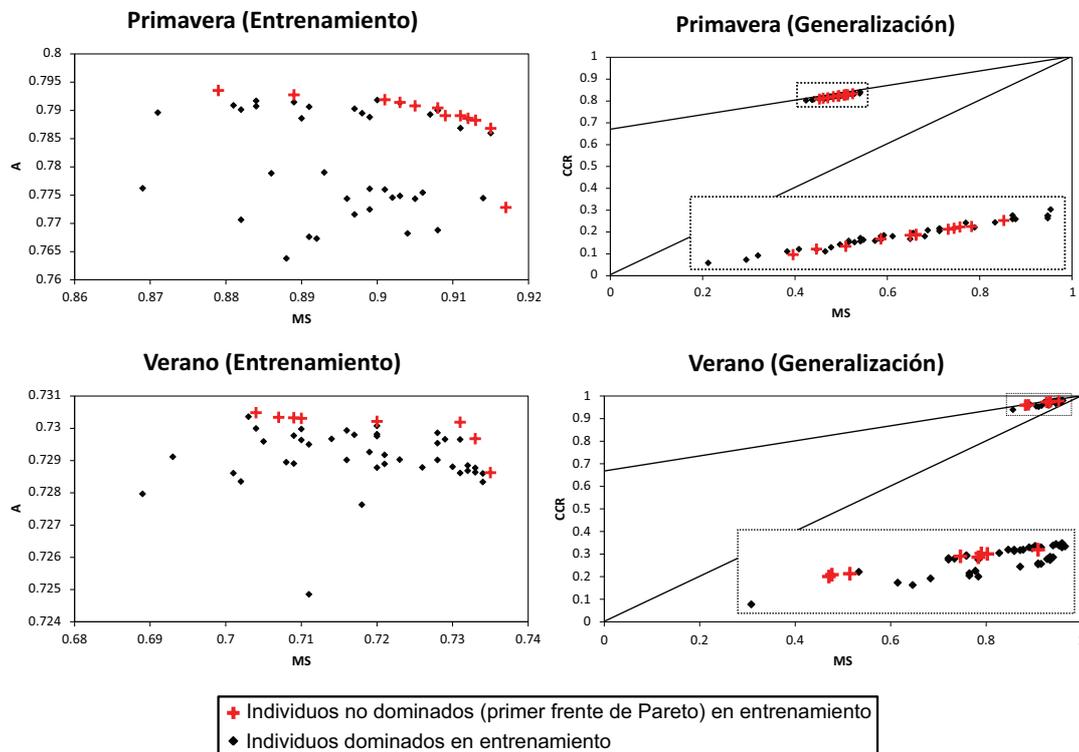


Figura 7.3: Frente de Pareto en entrenamiento en el espacio (MS, A) y valores asociados en el espacio (MS, CCR) en generalización para las bases de datos Primavera y Verano. En las imágenes de generalización se ha aplicado un zoom sobre la zona marcada para facilitar la visualización de los datos.

A la vista de los resultados, podemos afirmar que el mejor algoritmo en Primavera sobre el conjunto de generalización es el algoritmo MPDENN2-MS. Esto indica que en Primavera, al intentar mejorar el porcentaje de clasificación de la clase peor clasificada, se mejora el rendimiento de todas las clases, ya que este algoritmo es el que obtiene los mejores valores en todas las métricas (como se muestra en el test de orden de la Tabla 7.4). Por otro lado, el test t de Student indica que no existen diferencias significativas con respecto al algoritmo que presenta el segundo mejor rendimiento en CCR (MPENSGAII-MS, $p\text{-valor} = 0,92$) y en MS (PDENN-MS, $p\text{-valor} = 0,27$). En Verano, el mejor método en generalización es el algoritmo MPDENN2-E, ya que obtiene los mejores valores en todas las métricas y las diferencias de medias son significativas ($p\text{-valor} = 0,01$ en CCR y $0,03$ en MS) cuando comparamos con el algoritmo MPDENN2-MS mediante el test t de Student (en esta comparación las poblaciones son dependientes, ya que se obtienen con el mismo algoritmo).

Tabla 7.4: *p-value* del test Levene de igualdad de varianzas, *p-value* del test F de Snedecor, orden de medias del test estadístico de comparaciones múltiples de Tukey o Tamhane sobre el conjunto de generalización (finca de Montilla) usando los nuevos algoritmos y *p-value* del test t de Student comparando MPDENN2-MS con MPENSGAII-MS y MPDENN2-MS con PDENN-MS en Primavera y MPDENN2-E con MPDENN2-MS en Verano.

Conjunto	Lev.	Test F	Test de orden Tukey o Tamhane	Todos los algoritmos		
				MPDENN2-MS vs MPDENN2-MS	MPDENN2-E vs MPDENN2-MS	MPDENN2-E vs MPDENN2-MS
CCR en Primavera	0,18	0,00*	$\mu_{MPDENN2-MS} \geq \mu_{MPENSGAII-MS} \geq \mu_{NSGAII-MS} \geq \mu_{PDENN-MS} \geq \mu_{MPENSGAII-E} \geq \mu_{NSGAII-E} \geq \mu_{PDENN-E} \geq \mu_{MPDENN2-E} \geq \mu_{MILP}$	0,92	-	-
			$\mu_{MPDENN2-MS} > \mu_{PDENN-MS}; \mu_{NSGAII-MS} > \mu_{MPDENN2-E}$			
MS en Primavera	0,00*	0,00*	$\mu_{MPDENN2-MS} \geq \mu_{PDENN-MS} \geq \mu_{NSGAII-MS} \geq \mu_{MPENSGAII-MS} \geq \mu_{NSGAII-E} \geq \mu_{MPDENN2-E} \geq \mu_{MPENSGAII-E} \geq \mu_{PDENN-E} > \mu_{MILP}$	-	0,27	-
			$\mu_{MPDENN2-MS} > \mu_{PDENN-MS}; \mu_{NSGAII-MS} > \mu_{MPDENN2-E}$			
CCR en Verano	0,00*	0,00*	$\mu_{MPDENN2-E} \geq \mu_{MPDENN2-MS} \geq \mu_{PDENN-E} \geq \mu_{MPENSGAII-E} \geq \mu_{PDENN-MS} \geq \mu_{MPENSGAII-MS} \geq \mu_{MILP} \geq \mu_{NSGAII-E} \geq \mu_{NSGAII-MS};$	-	-	0,01 (*)
			$\mu_{MPDENN2-E} > \mu_{MPENSGAII-MS}; \mu_{PDENN-MS} > \mu_{NSGAII-MS}$			
MS en Verano	0,00*	0,00*	$\mu_{MPDENN2-E} \geq \mu_{MPDENN2-MS} \geq \mu_{MILP} \geq \mu_{PDENN-MS} \geq \mu_{PDENN-E} \geq \mu_{MPENSGAII-MS} \geq \mu_{MPENSGAII-E} \geq \mu_{NSGAII-E} \geq \mu_{NSGAII-MS};$	-	-	0,03 (*)
			$\mu_{MPDENN2-E} > \mu_{MILP} > \mu_{NSGAII-MS}; \mu_{MPDENN2-MS} > \mu_{PDENN-MS}$			

* La diferencia de medias es significativa con *p-value* = 0,05.
 $\mu_A \geq \mu_B$: la topología A obtiene mejores resultados que la B, pero las diferencias no son significativas.
 $\mu_A > \mu_B$: la topología A obtiene mejores resultados que la B con diferencias significativas.
 Las relaciones $> \mu \geq \mu$ no son transitivas.

El mejor modelo obtenido por el método MPDENN2-MS sobre la base de datos de Primavera puede verse en la Tabla 7.5. En la Tabla 7.6 se muestra el mejor modelo obtenido por el método MPDENN2-E en la base de datos de Verano. En estas Tablas se muestran tanto los modelos obtenidos como su comportamiento en entrenamiento y generalización. Estos modelos constituyen una poderosa herramienta para las administraciones andaluza y europea a la hora de discriminar las cubiertas vegetales, los olivos y el suelo desnudo tanto en primavera y como en verano. Por tanto, si falla la adquisición de imágenes durante la primavera (debido a los días nublados), se podría programar la obtención de imágenes en verano, cuando hay muchos más días soleados. Además, esta doble posibilidad se podría utilizar para volver a supervisar en verano una finca dudosa analizada en primavera. De esta forma, estos modelos podrían ayudar a la administración a la hora del reparto o mantenimiento de las subvenciones.

Tabla 7.5: Mejor modelo obtenido por el algoritmo MPDENN2-MS para la base de datos Primavera.

$p_{CV}(\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{f_1(\mathbf{x}, \boldsymbol{\theta})}}{1 + \sum_{i=1}^2 e^{f_i(\mathbf{x}, \boldsymbol{\theta})}}; p_O(\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{f_2(\mathbf{x}, \boldsymbol{\theta})}}{1 + \sum_{i=1}^2 e^{f_i(\mathbf{x}, \boldsymbol{\theta})}}; p_{SD}(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{1 + \sum_{i=1}^2 e^{f_i(\mathbf{x}, \boldsymbol{\theta})}}$		
$f_1(\mathbf{x}, \boldsymbol{\theta}) = 5,71 - 4,96MLP_1 - 4,93MLP_2 - 5,00MLP_3 - 5,00MLP_4$		
$f_2(\mathbf{x}, \boldsymbol{\theta}) = 1,58 - 5,00MLP_1 + 5,00MLP_2 - 2,58MLP_3 - 5,00MLP_4$		
$f_3(\mathbf{x}, \boldsymbol{\theta}) = 0$		
$MLP_1 = \phi(-2,59 + 0,69B^* - 0,20G^* + 5,00R^*)$		
$MLP_2 = \phi(0,30 + 3,00B^* - 4,98G^* - 1,51R^*)$		
$MLP_3 = \phi(-3,33 + 3,01B^* + 1,50G^* + 4,98R^*)$		
$MLP_4 = \phi(-1,72 - 2,37G^* + 4,79R^*)$		
$B^*, G^*, R^* \in [-1, 1]$		
$CCR_E = 91,66\%; MS_E = 91,50\%; K_E = 0,8750 \quad E \equiv \text{finca de Espejo}$		
$CCR_G = 83,33\%; MS_G = 52,50\%; K_G = 0,7500 \quad G \equiv \text{finca de Montilla}$		
$n^\circ \text{ de neuronas} = 4; n^\circ \text{ de coeficientes} = 25$		
$M_E = \begin{pmatrix} C_1 & C_2 & C_3 \\ 916 & 63 & 21 \\ 78 & 915 & 7 \\ 80 & 1 & 919 \end{pmatrix}; M_G = \begin{pmatrix} C_1 & C_2 & C_3 \\ 525 & 185 & 290 \\ 17 & 975 & 8 \\ 0 & 0 & 1,000 \end{pmatrix}$		
$\text{Clase 1}(C_1) \equiv \text{Cubiertas vegetales (CV)}; \text{Clase 2}(C_2) \equiv \text{Olivos (O)}; \text{Clase 3}(C_3) \equiv \text{Suelo desnudo (SD)}$		

Tabla 7.6: Mejor modelo obtenido por el algoritmo MPDENN2-E para la base de datos Verano.

$p_{CV}(\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{f_1(\mathbf{x}, \boldsymbol{\theta})}}{1 + \sum_{i=1}^2 e^{f_i(\mathbf{x}, \boldsymbol{\theta})}}; p_O(\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{f_2(\mathbf{x}, \boldsymbol{\theta})}}{1 + \sum_{i=1}^2 e^{f_i(\mathbf{x}, \boldsymbol{\theta})}}; p_{SD}(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{1 + \sum_{i=1}^2 e^{f_i(\mathbf{x}, \boldsymbol{\theta})}}$
$f_1(\mathbf{x}, \boldsymbol{\theta}) = -1,39 - 4,97MLP_1 + 3,82MLP_2 + 0,21MLP_3 + 4,89MLP_4$
$f_2(\mathbf{x}, \boldsymbol{\theta}) = 3,75 + 4,96MLP_1 - 5,00MLP_2 - 5,00MLP_3 + 4,94MLP_4$
$f_3(\mathbf{x}, \boldsymbol{\theta}) = 0$
$MLP_1 = \phi(2,56 + 5,00B^* - 2,07G^* - 4,99R^* + 1,26NIR^*)$
$MLP_2 = \phi(-4,22 - 4,86B^* + 4,95G^* + 4,94R^* - 0,17NIR^*)$
$MLP_3 = \phi(1,42 + 4,36B^* - 4,72G^* + 4,78R^* + 3,15NIR^*)$
$MLP_4 = \phi(0,11 - 2,72B^* - 1,15G^* - 4,99R^* - 4,93NIR^*)$
$B^*, G^*, R^*, NIR^* \in [-1, 1]$
$CCR_E = 82,20\%; MS_E = 73,30\%; K_E = 0,7330 \quad E \equiv \text{finca de Espejo}$
$CCR_G = 97,80\%; MS_G = 95,20\%; K_G = 0,9710 \quad G \equiv \text{finca de Montilla}$
$n^\circ \text{ de neuronas} = 4; n^\circ \text{ de coeficientes} = 30$
$M_E = \begin{pmatrix} C_1 & C_2 & C_3 \\ 733 & 0 & 267 \\ 0 & 1,000 & 0 \\ 265 & 2 & 733 \end{pmatrix}; M_G = \begin{pmatrix} C_1 & C_2 & C_3 \\ 952 & 0 & 48 \\ 0 & 1,000 & 0 \\ 10 & 0 & 990 \end{pmatrix}$
$\text{Clase 1}(C_1) \equiv \text{Cubiertas vegetales (CV)}; \text{Clase 2}(C_2) \equiv \text{Olivos (O)}; \text{Clase 3}(C_3) \equiv \text{Suelo desnudo (SD)}$

RESOLUCIÓN DE UN PROBLEMA DE CRECIMIENTO MICROBIANO EN MICROBIOLOGÍA PREDICTIVA

Los modelos de crecimiento/no crecimiento han aparecido en el campo de la microbiología predictiva como un método para determinar la capacidad de crecimiento de los microorganismos. En este sentido, muchos estudios han sido publicados en los últimos años debido a la necesidad de obtener más conocimientos sobre el comportamiento microbiano en condiciones limitantes que impidan el crecimiento. Por tanto, los modelos matemáticos proporcionados por estos estudios pueden llevarnos a estimaciones más realistas de los riesgos de seguridad de los alimentos y pueden proporcionarnos datos cuantitativos útiles para el desarrollo de procesos que permitan la producción de artículos alimenticios de manera más segura [62, 84].

Los principales problemas en el modelado de sistemas microbianos están relacionados con las transiciones abruptas, es decir, con el gran cambio que se produce en el valor de las probabilidades de crecimiento (p) debido a la estrecha gama de factores ambientales que se encuentran entre las condiciones de crecimiento. Por tanto, para una correcta definición del sistema, el crecimiento o no crecimiento solo es considerado si ocurre en todas las replicas del experimento (o al menos en un alto porcentaje). En los trabajos desarrollados en la presente Tesis se han considerado cuatro posibles respuestas del sistema microbiano, basadas en la probabilidad de crecimiento de los microorganismos: $p = 1$, crecimiento (C); $0,5 \leq p < 1$, crecimiento con alta probabilidad (CAP); $0 < p < 0,5$, crecimiento con baja probabilidad (CBP); y $p = 0$, no crecimiento (NC). La existencia de las clases CAP y CBP están claramente justificada, ya que en ciertas zonas del dominio del modelo las respuestas microbianas son muy variables y, por tanto, una clasificación como C o NC no puede ser realizada.

8.1 PROBLEMAS ANALIZADOS Y DISEÑO EXPERIMENTAL

En este Capítulo se va a trabajar con dos problemas de crecimiento microbiano relacionados con los patógenos *Staphylococcus Aureus* y *Shigella Flexneri*.

8.1.1 *Staphylococcus Aureus*

El patógeno *S. Aureus* se ha reconocido como un indicador de la deficiencia de higiene en los alimentos y una causa importante de gastroenteritis en todo el mundo [118]. Para conocer los límites de crecimiento de *S. Aureus* se ha utilizado un diseño factorial fraccional [124]. Cuidadosamente, se ha elegido un subconjunto (fracción) de los experimentos de un diseño factorial completo con el fin de reducir el tiempo de experimentación y los recursos necesarios y mejorar el proceso de entrenamiento. La selección se basó en la delimitación de los niveles de los factores ambientales estudiados para el dominio de crecimiento/no crecimiento de *S. Aureus*. Dado que no se ha detectado un crecimiento por debajo de 7,5 °C, los datos se recopilaron a 8, 10, 13, 16 y 19 °C, y a niveles de pH de 4,5 a 7,5 (en intervalos de 0,5) y en 19 niveles de actividad de agua (aW) (desde 0,856 a 0,999 en intervalos regulares).

El patógeno inicial (287 condiciones) se dividió en dos partes: un conjunto de entrenamiento con 146 condiciones, cubriendo los valores extremos de las variables predictoras o factores ambientales, y un conjunto de generalización, con 141 condiciones, dentro de la región de interpolación del modelo. Para las diferentes condiciones de los factores ambientales, se produjeron 117 casos de C, 45 casos de CAP, 12 casos de CBP y 113 casos de NC. El objetivo de esta selección fue definir un patógeno centrado en las regiones extremas del dominio de crecimiento/no crecimiento, que representen las zonas fronterizas. En este estudio, el número de réplicas por condición ($n = 30$) se incrementó en comparación con otros estudios, para obtener de forma más precisa el punto de transición de crecimiento/no crecimiento.

8.1.2 *Shigella Flexneri*

El patógeno *S. Flexneri* es un importante agente causante de enfermedades gastrointestinales [132]. Para evaluar los efectos de la temperatura se utilizó un diseño factorial incompleto con valores de 12, 15, 19, 28, 37 °C, un pH inicial de 5,5, 6,0, 6,5, 7,0, 7,5, cloruro de sodio con valores de 0,5, 2,5 y 4,0 %, y nitrito de sodio con valores de 0, 50, 100, 200 y 1000 ppm. Los datos se obtuvieron a partir de 123 combinaciones de variables. El número de réplicas de cultivos probados para cada combinación de variables se muestra en la Tabla 2 del artículo [132]. Estos datos se utilizaron para obtener los modelos y predecir el crecimiento anaeróbico de *S. Flexneri* en función de la temperatura, las concentraciones de cloruro de sodio, nitrito de sodio y el pH inicial. Por ejemplo, como se muestra en la Tabla 8.1, para determinar que patrones que pertenecen al conjunto de entrenamiento o generalización de la base de datos *S. Flexneri*, las variables cloruro de sodio y nitrito de sodio han sido seleccionadas alternativamente manteniendo el mismo nivel de temperatura y pH.

La cinética de crecimiento de los datos para cada combinación de variables se resume en la Tabla 2 del artículo al que antes se ha hecho referencia. El crecimiento de *S.*

Tabla 8.1: Diseño factorial para la base de datos *S. Flexneri*.

		NaNO ₂ (ppm)				
		0	50	100	200	1000
NaCl (%)	0,5	○	◆	○	◆	○
	2,5	◆	○	◆	○	◆
	4,0	○	◆	○	◆	○

NaNO₂ = Nitrito de sodio, NaCl = Cloruro de sodio.

○ = patrones para entrenamiento, ◆ = patrones para generalización.

Flexneri no se observó bajo las condiciones correspondientes a 40 de las combinaciones de variables estudiadas. Se estudiaron 15 combinaciones adicionales de variables originadas en entornos ambientales, donde algunas de las replicas del cultivo crecieron, mientras que otras no lo hicieron (detallado en la Tabla 3 del artículo citado). De las diferentes condiciones, existen 39 casos de C, 8 casos de CAP, 7 casos de CBP y 69 casos de NC.

El diseño experimental seguido en este estudio es un proceso de validación cruzada (*holdout*) con 30 ejecuciones del MOEA, donde las bases de datos se han dividido mediante un diseño factorial fraccional en un conjunto de datos de entrenamiento y otro de generalización. Estas particiones se pueden ver en [124] para *S. Aureus* y en [132] para *S. Flexneri*. Este diseño experimental es muy común en microbiología predictiva.

8.2 METODOLOGÍA DE RESOLUCIÓN DEL PROBLEMA

En este estudio se ha empleado el algoritmo MPDENN3 para entrenar modelos de red neuronal de base sigmoide que permitan determinar la capacidad de crecimiento de un microorganismo en determinadas condiciones ambientales. Para ello se han utilizado las bases de datos *S. Aureus* y *S. Flexneri* descritas anteriormente. El rendimiento del algoritmo MPDENN3 se ha comparado con los resultados obtenidos por los algoritmos PDENN, MPANN y MPENSGAII. Una ampliación de este estudio se encuentra en [30].

En la Tabla 8.2 se muestran las características de cada una de las bases de datos utilizadas: número total de patrones en cada conjunto de datos, número de patrones en entrenamiento y en generalización, número de variables de entrada, número de clases, número total de patrones por clase y valor de p^* (porcentaje de patrones de la clase minoritaria).

La evolución de los cuatro algoritmos evolutivos es guiada por la entropía (E) y la mínima sensibilidad (MS). Con el objetivo de acotar las dos métricas en el intervalo [0,1], se ha aplicado sobre la medida E la siguiente transformación estrictamente

Tabla 8.2: Características de las bases de datos *S. Aureus* y *S. Flexneri*.

Base de datos	Total patrones	Patrones entrenam.	Patrones general.	Variables de entrada	Número de clases	Patrones por clase	p*
<i>S. Aureus</i>	287	146	141	3	4	117-45-12-113	0,0418
<i>S. Flexneri</i>	123	76	47	5	4	39-8-7-69	0,0569

decreciente $A = 1/(1 + E)$. Con esto se definen las dos funciones a maximizar por los diferentes MOEAs.

Los parámetros empleados por todos los algoritmos son los mismos para las dos bases de datos. Para los tres algoritmos diferenciales (MPDENN3, PDENN y MPANN), el tamaño máximo de la población es de 25 individuos, la probabilidad de cruce se estableció a $pro_cruce = 0,8$ y la de mutación a $pro_mut = 0,1$. Estos parámetros se han fijado siguiendo las directrices de H. Abbass [4]. Para el algoritmo MPENSGAII se emplea un tamaño de población de 150 individuos y una probabilidad de mutación para cada operador de $1/5$, siguiendo las directrices de K. Deb [41] (el Capítulo 4 contiene más información sobre estos algoritmos y sus parámetros).

Para el algoritmo de búsqueda local $iRprop^+$, se utilizan los parámetros recomendados por C. Igel [76]: $\eta^- = 0,5$, $\eta^+ = 1,2$, $\Delta_0 = 0,0125$, $\Delta_{min} = 0$, $\Delta_{max} = 50$ y Epochs = 25. En el algoritmo MPDENN3, se aplica 3 veces la búsqueda local durante la ejecución (cada 33,33% de las generaciones) y el número de individuos representativos deseados es $num = 5$.

Para iniciar el procesamiento de datos, cada una de las variables de entrada se escalaron en el intervalo $[-1, 1]$, para evitar así la saturación de la señal.

8.3 RESULTADOS

La Tabla 8.3 presenta el tiempo empleado por cada algoritmo en entrenamiento y los resultados obtenidos sobre el conjunto de generalización. Los resultados mostrados se corresponden con el valor medio obtenido tras realizar las 30 ejecuciones. En cada ejecución se seleccionan los modelos situados en los dos extremos del frente de Pareto (la Sección 4.5 muestra más información sobre las técnicas de selección de individuos). Los modelos con mayor valor en A (extremo superior del frente) se han etiquetado como *NombreMétodo-E*, y aquellos con el mayor valor en MS (extremo inferior del frente) como *NombreMétodo-MS*.

Analizando los resultados se observa que el algoritmo MPDENN3 (tanto con la metodología -E como -MS) consigue buenos resultados para la base de datos *S. Aureus* y resultados similares al algoritmo con mejor desempeño en *S. Flexneri* (MPENSGAII), pero con menor tiempo de ejecución

Tabla 8.3: Tiempo de ejecución en entrenamiento y resultados obtenidos en generalización: media y desviación típica ($Media_{DT}$).

Algoritmo	CCR	MS	RMSE	K	Tiempo(seg.)
<i>Staphylococcus Aureus</i>					
MPDENN3-E	<u>73,03_{1,53}</u>	0,00 _{0,00}	<u>0,30_{0,01}</u>	<u>0,56_{0,03}</u>	48,72_{3,70}
MPDENN3-MS	55,01 _{7,97}	21,81_{14,02}	0,39 _{0,03}	0,37 _{0,09}	48,72_{3,70}
PDENN-E	71,27 _{2,50}	0,00 _{0,00}	0,32 _{0,01}	0,53 _{0,03}	<u>49,34_{5,24}</u>
PDENN-MS	53,52 _{6,57}	<u>20,59_{10,98}</u>	0,38 _{0,02}	0,39 _{0,05}	<u>49,34_{5,24}</u>
MPANN-E	74,44_{1,41}	0,00 _{0,00}	0,29_{0,01}	0,59_{0,02}	309,14 _{11,51}
MPANN-MS	66,71 _{8,87}	9,36 _{10,82}	0,33 _{0,04}	0,49 _{0,11}	309,14 _{11,51}
MPENSGAII-E	69,65 _{2,92}	0,00 _{0,00}	0,35 _{0,01}	0,49 _{0,05}	158,92 _{36,13}
MPENSGAII-MS	56,71 _{5,09}	<u>20,00_{10,50}</u>	0,39 _{0,01}	0,39 _{0,06}	<u>158,92_{36,13}</u>
<i>Shighella Flexneri</i>					
MPDENN3-E	<u>87,02_{1,16}</u>	0,00 _{0,00}	<u>0,22_{0,01}</u>	<u>0,76_{0,02}</u>	<u>249,12_{13,99}</u>
MPDENN3-MS	86,59 _{2,80}	2,22 _{8,45}	0,21_{0,01}	<u>0,76_{0,02}</u>	<u>249,12_{13,99}</u>
PDENN-E	83,75 _{1,04}	0,00 _{0,00}	<u>0,22_{0,01}</u>	0,73 _{0,04}	234,90_{17,93}
PDENN-MS	84,04 _{3,77}	9,77_{14,82}	<u>0,22_{0,02}</u>	0,73 _{0,04}	234,90_{17,93}
MPANN-E	<u>87,02_{0,85}</u>	0,00 _{0,00}	0,23 _{0,01}	<u>0,76_{0,02}</u>	1956,93 _{45,28}
MPANN-MS	<u>87,02_{0,85}</u>	0,00 _{0,00}	0,23 _{0,01}	<u>0,76_{0,02}</u>	1956,93 _{45,28}
MPENSGAII-E	88,63_{1,63}	<u>5,55_{12,63}</u>	0,21_{0,01}	0,79_{0,03}	1681,41 _{203,78}
MPENSGAII-MS	88,63_{1,63}	<u>5,55_{12,63}</u>	0,21_{0,01}	0,79_{0,03}	1681,41 _{203,78}

Los mejores resultados están marcadas en **negrita** y los segundos mejores en subrayado.

De hecho, desde un punto de vista meramente descriptivo, el algoritmo MPDENN3 es el algoritmo más rápido y obtiene el mejor valor para la métrica MS en *S. Aureus* (además obtiene los segundos mejores resultados en el resto de métricas). En *S. Flexneri*, es el segundo mejor algoritmo, debido a que presenta el segundo mejor resultado en 4 de las 5 medidas.

En la Figura 8.1 se muestran los resultados gráficos obtenidos por el algoritmo MPDENN3 para las bases de datos *S. Aureus* y *S. Flexneri* en entrenamiento en el espacio (MS, A) y en generalización en el espacio (MS, CCR). Para el espacio (MS, A), se ha seleccionado el frente de Pareto de la ejecución que presenta el mejor valor en A dentro de las 30 realizadas. En las gráficas de generalización se muestran los valores de MS y CCR obtenidos por los individuos de la población sobre el conjunto de generalización. Cabe destacar que en generalización los individuos no forman necesariamente frentes de Pareto y que los individuos que en entrenamiento estaban localizados en el primer frente de Pareto, ahora pueden estar situados en una región menos deseable del espacio (MS, CCR).

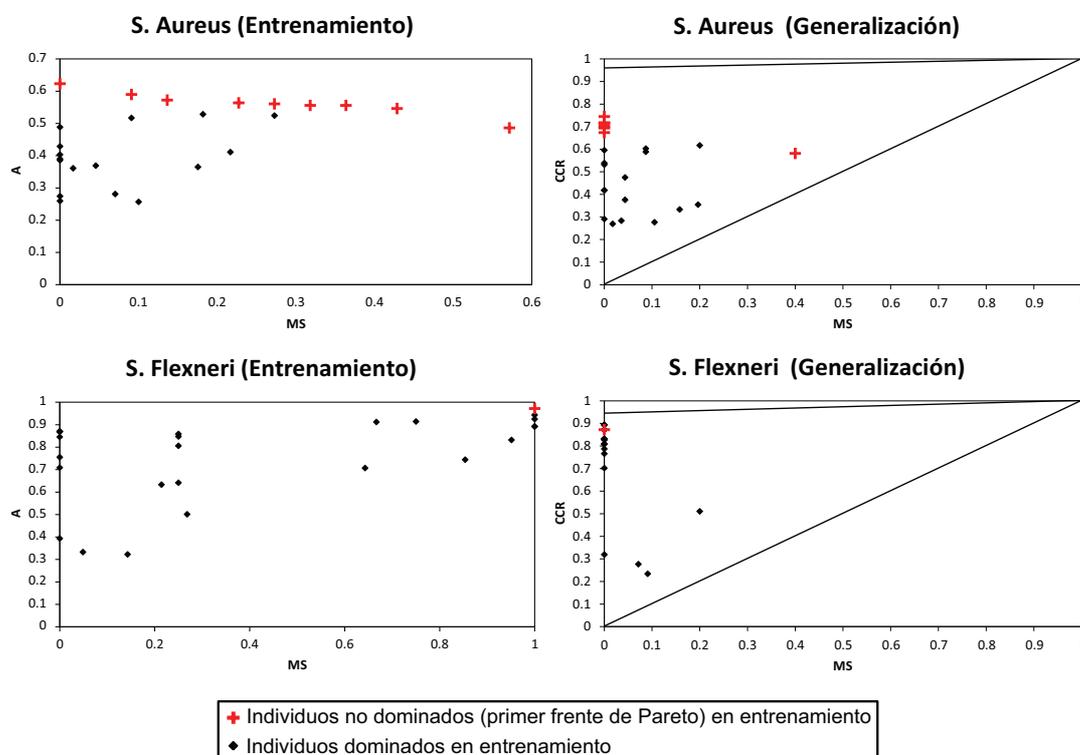


Figura 8.1: Frente de Pareto en entrenamiento en el espacio (MS, A) y valores asociados en el espacio (MS, CCR) en generalización para las bases de datos *S. Aureus* y *S. Flexneri*.

Con el fin de determinar el mejor algoritmo para el entrenamiento de redes neuronales MLP, se ha llevado a cabo un test de análisis de la varianza de un factor (*ANalysis Of the VAriance of one factor*, ANOVA I) y el test no paramétrico de Kruskal-Wallis. Estos tests son utilizados en función de la satisfacción de las hipótesis de normalidad para los valores de CCR, RMSE y K y para el tiempo de ejecución. El factor F_i analiza el efecto que tiene el nivel i -ésimo del factor sobre la media de cada una de las métricas, donde F_i representa los diferentes algoritmos empleados, con niveles: $i = 1$, MPDENN3-E (M3E); $i = 2$, MPDENN3-MS (M3MS); $i = 3$, PDENN-E (PE); $i = 4$, PDENN-MS (PMS); $i = 5$, MPANN-E (ME); $i = 6$, MPANN-MS (MMS); $i = 7$, MPENSGAII-E (MPE); $i = 8$, MPENSGAII-MS (MPMS). No es posible realizar estas pruebas para MS, porque algunos algoritmos presentan valores de cero en ambas bases de datos. Para el tiempo de ejecución, los resultados obtenidos por los algoritmos son los mismos para ambas metodologías de selección (tanto con la metodología -E como -MS), por lo que solo existen cuatro poblaciones. Los tests muestran que el efecto, en media, del método usado para clasificación es estadísticamente sig-

nificativo para un nivel de significación del 5% en ambas bases de datos. La primera parte de la Tabla 8.4 muestra los resultados obtenidos con estos tests.

Tabla 8.4: p -valor del análisis ANOVA I o del test de Kruskal-Wallis y orden de medias del test estadístico de comparaciones múltiples de Tukey o del test de pares de Mann-Whitney.

	<i>Staphylococcus Aureus</i>	<i>Shigella Flexneri</i>
Tests iniciales: ANOVA I o test de Kruskal-Wallis		
CCR	0,000*	0,000°
MS	0,000°	0,000°
RMSE	0,000*	0,008*
K	0,000*	0,000°
Tiempo	0,000*	0,000*
Test <i>post hoc</i> : Tukey o Mann-Whitney		
CCR	$\mu_{ME} \geq \mu_{M_3E} \geq \mu_{PE} \geq \mu_{MPE} \geq \mu_{MMS} > \mu_{MPMS} \geq \mu_{M_3MS} \geq \mu_{PMS}$ $\mu_{ME} > \mu_{MPE}; \mu_{PE} > \mu_{MMS}$	$\mu_{MPE} \geq \mu_{ME}; \mu_{MPE} \geq \mu_{M_3E}; \mu_{MPE} > \mu_{PE}; \mu_{M_3E} > \mu_{PE}$
MS	$\mu_{M_3MS} \geq \mu_{PMS}; \mu_{M_3MS} \geq \mu_{MPMS}; \mu_{M_3MS} > \mu_{MMS}$	$\mu_{PMS} \geq \mu_{MPMS}; \mu_{PMS} \geq \mu_{M_3MS}; \mu_{MPMS} \geq \mu_{M_3MS}$
RMSE	$\mu_{ME} \leq \mu_{M_3E} < \mu_{PE} \leq \mu_{MMS} < \mu_{MPE} < \mu_{PMS} \leq \mu_{M_3MS} \leq \mu_{MPMS}; \mu_{PMS} < \mu_{MPMS}$	$\mu_{MPE} \leq \mu_{MPMS} \leq \mu_{PE} \leq \mu_{M_3E} \leq \mu_{M_3MS} \leq \mu_{PMS} \leq \mu_{ME} \leq \mu_{MMS}; \mu_{MPMS} < \mu_{PMS}$
K	$\mu_{ME} \geq \mu_{M_3E} \geq \mu_{PE} \geq \mu_{MMS} \geq \mu_{MPE} > \mu_{PMS} \geq \mu_{MPMS} \geq \mu_{M_3MS}; \mu_{ME} > \mu_{PE}; \mu_{M_3E} > \mu_{MMS}$	$\mu_{MPE} \geq \mu_{ME}; \mu_{MPE} \geq \mu_{M_3E}; \mu_{MPE} \geq \mu_{PE}; \mu_{ME} \geq \mu_{M_3E}; \mu_{ME} \geq \mu_{PE}; \mu_{M_3E} \geq \mu_{PE}$
Tiempo	$\mu_{MPDENN} \leq \mu_{PDE} < \mu_{MPENSGA_2} < \mu_{MPANN}$	$\mu_{PDE} \leq \mu_{MPDENN} < \mu_{MPENSGA_2} < \mu_{MPANN}$

* Análisis ANOVA I. ° Test de Kruskal-Wallis.

$\mu_A \geq \mu_B$: la topología A obtiene mejores resultados que la B, pero las diferencias no son significativas.

$\mu_A > \mu_B$: la topología A obtiene mejores resultados que la B con diferencias significativas.

Las relaciones $>$ y \geq no son transitivas.

Debido a que existen diferencias significativas en media usando el análisis ANOVA I o el test Kruskal-Wallis, se realiza un test de comparaciones múltiples *post hoc* de la media de las diferentes métricas con los diferentes niveles de cada factor (segunda parte de la Tabla 8.4). Primero, se emplea el test de Levene para evaluar si existen diferencias significativas en las varianzas. Entonces, si las varianzas son iguales estadísticamente, se aplica el test de Tukey. En el resto de casos se aplica el test no paramétrico por pares de Mann-Whitney. En estos tests se ha considerado un p -valor de 0,05.

Analizando los resultados para la medida CCR, se observa que los resultados obtenidos por el algoritmo MPDENN3-E son, en media, similares a los obtenidos por

los mejores algoritmos (MPANN-E en *S. Aureus* y MPENSGAII-E en *S. Flexneri*), para un nivel de significación del 5%. Los resultados de la métrica MS muestran que el algoritmo MPDENN3-MS obtiene el mejor desempeño en *S. Aureus* y un desempeño similar en *S. Flexneri* al obtenido por el mejor algoritmo (PDENN-MS) (con un nivel de significación del 5%). Para las medidas RMSE y K, los resultados muestran que el método MPDENN3-E no presenta diferencias significativas con los mejores algoritmos (MPANN-E en *S. Aureus* y MPENSGAII-E en *S. Flexneri*), con un nivel de significación del 5%. En cuanto al tiempo de ejecución, los algoritmos PDENN y MPDENN3 obtienen resultados similares y significativamente mejores que los obtenidos por los algoritmos MPANN y MPENSGAII en ambas bases de datos.

Los resultados estadísticos obtenidos muestran que algoritmo MPDENN3 es muy competitivo para ser considerado en microbiología predictiva, donde hay una alta necesidad de obtener una buena precisión en la clasificación. Los modelos obtenidos con este algoritmo pueden predecir, a partir de ciertos factores medioambientales de envasado y/o conservación, el grado de crecimiento de un patógeno, considerando 4 posibles clases de pertenencia: crecimiento, crecimiento con alta probabilidad, crecimiento con baja probabilidad y no crecimiento. Esta división en 4 clases puede ayudar a los investigadores de microbiología predictiva a definir mejor los límites de crecimiento de los microorganismos y también a modelar de forma más flexible la variabilidad microbiana asociada a las condiciones de envasado y conservación.

8.4 ENFOQUE ORDINAL

Los problemas de crecimiento microbiano pueden ser tratados desde un punto de vista ordinal, ya que en estos problemas se han considerado cuatro clases que poseen un orden inherente entre ellas: $NC \prec CBP \prec CAP \prec C$. Este enfoque ha sido estudiado en [33]. Los algoritmos de clasificación ordinal emplean esta información de orden entre las diferentes clases para mejorar la predicción y para minimizar la magnitud de los errores (no es lo mismo predecir que un patrón de la clase NC pertenece a la clase CBP que a la clase C. Aunque en los dos casos se ha cometido error, la predicción del primer caso se encuentra más cercana a la clase real que la predicción del segundo caso y, por tanto, es un error menos grave). En el siguiente Capítulo se detalla el concepto de clasificación ordinal y las consideraciones a tener en cuenta a la hora de trabajar en ella.

ALGORITMOS MULTIOBJETIVO EN CLASIFICACIÓN ORDINAL

En el mundo real existen muchos problemas de aprendizaje supervisado relacionados con la clasificación ordinal, en los cuales los patrones son etiquetados mediante una escala ordinal discreta y finita. Por ejemplo, existe un orden inherente en la escala que emplea un profesor para evaluar a sus alumnos: “matricula de honor”, “sobresaliente”, “notable”, “aprobado” y “suspenso”. Los problemas de clasificación ordinal son frecuentes en multitud de áreas de investigación, como la medicina [12, 53, 98], interfaz cerebro-computador [130], valoración de crédito [46, 82] o reconocimiento facial [113]. Algunos ejemplos concretos de problemas ordinales pueden ser la gravedad de un paciente en medicina (“sano”, “leve”, “moderado” o “grave”), la eficiencia energética de los electrodomésticos (“A++”, “A+”, “A”, “B”, “C”, “D”, “E”, “F” o “G”) o la clásica escala de Likert empleada en multitud de encuestas (“totalmente en desacuerdo”, “en desacuerdo”, “ni de acuerdo ni en desacuerdo”, “de acuerdo” o “totalmente de acuerdo”). Estos problemas de clasificación ordinal también son conocidos en la literatura como problemas de regresión ordinal.

La clasificación ordinal se encuentra a medio camino entre la clasificación nominal y la regresión, ya que presenta características comunes a ambos paradigmas, pero también presenta algunas diferencias. La clasificación ordinal, al igual que la regresión, pretende predecir una variable de salida que presenta un cierto orden entre sus valores, sin embargo, en regresión la variable es continua e infinita mientras que en la clasificación ordinal esta variable es discreta y finita. Con respecto a la clasificación nominal, ambos paradigmas buscan clasificar correctamente un conjunto de clases finitas. La diferencia estriba en que la clasificación ordinal existe un orden entre las diferentes clases que debe ser considerado a la hora de realizar la predicción.

El principal inconveniente que presentan los problemas de clasificación ordinal es la falta de precisión en el cálculo de las distancias entre las diferentes clases del problema. Los resultados obtenidos dependen del valor de las etiquetas de clase que se consideren.

Como primera propuesta para tratar con los problemas ordinales, se podría pensar en emplear los algoritmos de clasificación nominal, aunque no harían uso de la información de orden contenida en las etiquetas de clase. Otro aspecto que los algoritmos nominales no consideran es que el coste de los errores de mala clasificación

debe ser diferente dependiendo de la gravedad del error. Volviendo al ejemplo de la evaluación de un alumno por parte del profesor, está claro que un alumno “sobresaliente” asignado a la clase “suspense” es un error mayor y, por tanto, debe estar más penalizado, que si se hubiese asignado a la clase “notable”. Es decir, cuanto más alejada esté la clase predicha de la clase real, más grave es el error cometido. Una solución común a este último aspecto es el empleo de matrices de coste, donde los errores más graves son más penalizados (esta solución conlleva el problema de fijar la ponderación de los errores). Como se observa, los algoritmos nominales no van a proporcionar, en general, soluciones óptimas, debido a que no tienen en cuenta la información ordinal proporcionada por el problema.

9.1 ALGORITMOS PARA CLASIFICACIÓN ORDINAL

Los problemas ordinales han sido tratados tradicionalmente en estadística mediante una función de enlace sobre un modelo lineal, de manera que dicha función estima las probabilidades de pertenencia a cada clase respetando las restricciones de orden [9]. Este tipo de modelos pertenecen al grupo conocido como métodos de umbral, en el sentido de que aprenden una proyección lineal o no lineal que sitúa todos los patrones en una línea recta y un vector de umbrales que estiman el rango que corresponde a cada categoría. En la última década, han empezado a surgir métodos de aprendizaje automático, que, con una filosofía similar, adaptan mecanismos de clasificación nominal bien conocidos para abordar los problemas de clasificación ordinal, como por ejemplo, las máquinas de vectores soporte (SVMs) [24, 69], los procesos Gaussianos [23] o el análisis discriminante [122], mostrando, en general, un mejor rendimiento que los métodos nominales tradicionales. En la Subsección 9.1.1 se muestra una descripción más detallada de estos métodos ordinales.

Existen otras alternativas, a parte de los métodos de umbral, para trabajar con problemas ordinales, como la descomposición del problema en subproblemas binarios o la clasificación binaria aumentada.

Tres aspectos deben ser tenidos en cuenta a la hora de trabajar con métodos basados en descomposición binaria. El primero de ellos está basado en cómo se realiza la descomposición binaria. En la Tabla 9.1 se muestran algunas de las descomposiciones más utilizadas para trabajar con problemas ordinales. El segundo aspecto a considerar es el número de modelos que serán entrenados por el algoritmo. Algunos métodos entrenan un modelo para cada subproblema [60] mientras que otros entrenan un único modelo para resolver todos los subproblemas [45]. El último aspecto está relacionado con la regla empleada para predecir nuevos patrones, una vez obtenidos los valores de decisión de todos los subproblemas, es decir, cómo se calculará la decisión final a partir de los resultados obtenidos en los subproblemas. Un caso peculiar de descomposición se muestra en [100], donde en lugar de realizar una descomposición binaria, se utiliza una descomposición ternaria. Las tres clases consi-

Tabla 9.1: Descomposición binaria para un problema ordinal de 5 clases.

Particiones ordenadas	Uno contra el siguiente	Uno contra los siguientes	Uno contra los anteriores
$\begin{pmatrix} -, -, -, - \\ +, -, -, - \\ +, +, -, - \\ +, +, +, - \\ +, +, +, + \end{pmatrix}$	$\begin{pmatrix} -, , , \\ +, -, , \\ , +, - \\ , , +, - \\ , , , + \end{pmatrix}$	$\begin{pmatrix} -, , , \\ +, -, , \\ +, +, - \\ +, +, +, - \\ +, +, +, + \end{pmatrix}$	$\begin{pmatrix} +, +, +, + \\ +, +, +, - \\ +, +, -, \\ +, -, , \\ -, , , \end{pmatrix}$

El símbolo + está asociado con la clase positiva y el símbolo - con la clase negativa. Si no se incluye ningún símbolo, es que la clase no ha sido usada en el subproblema.

deradas se corresponden con las clases anteriores a la clase q , la clase q y las clases posteriores a la clase q (en la primera y última clase solo se obtienen dos clases). Una vez realizadas las descomposiciones, se entrena un clasificador con cada uno de los subproblemas y, finalmente, se obtiene la predicción final a partir de las decisiones de estos clasificadores.

Los métodos de clasificación binaria aumentada también transforman el problema ordinal en un problema binario. En este caso, los patrones del problema ordinal son reemplazados por $Q - 1$ transformaciones binarias extendidas de ellos mismos [87]. Para la transformación q ($1 \leq q \leq Q - 1$), se incorpora al patrón original la codificación binaria de q y se reemplaza la variable de clase del patrón original por una variable binaria (-1 o 1), la cual representa el orden relativo entre la clase del patrón original y la clase q . Una vez aplicada esta transformación en todos los patrones, se entrena un clasificador binario. Finalmente, la ordenación de las clases se obtiene a partir de las salidas de este clasificador binario. El algoritmo más conocido que emplea este procedimiento es el REDSVM (*REDuction applied to Support Vector Machines*) [87].

9.1.1 Métodos de Umbral

A menudo, en el contexto de la clasificación ordinal, se asume que existe una variable latente que origina las etiquetas de clase con las que nos encontramos (por ejemplo, si las etiquetas de clase son “bebe”, “niño”, “adulto” y “anciano”, la variable latente que generó estas etiquetas podría ser la edad de la persona). Los métodos que aprovechan dicha hipótesis son denominados métodos de umbral y se puede considerar que son los algoritmos ordinales más populares entre todos los existentes [65, 126]. Estos métodos estiman:

- Una función $f(\mathbf{x})$ que intenta predecir la naturaleza de estos valores reales subyacentes.

- Un conjunto de umbrales $\beta = (\beta_1, \beta_2, \dots, \beta_{Q-1}) \in \mathbb{R}^{Q-1}$ que representan intervalos en el rango de $f(\mathbf{x})$ y que deben satisfacer la restricción $\beta_1 \leq \beta_2 \leq \dots \leq \beta_{Q-1}$. Estos umbrales delimitan cada una de las clases.

Se pueden considerar dos familias de métodos de umbral:

- Modelos discriminativos: Son métodos que estiman directamente la probabilidad *a posteriori* $P(y|\mathbf{x})$ o que aprenden una función para transformar \mathbf{x} en las etiquetas de clase. En esta familia, las propuestas más influyentes son los modelos de enlace con probabilidades acumuladas, las máquinas de vectores soporte y al análisis discriminante.
 - Modelos de enlace con probabilidades acumuladas (*Cumulative Link Models*, CLMs) [8]: Estos modelos provienen de un contexto puramente estadístico, siendo el más conocido el modelo de posibilidades proporcionales (*Proportional Odds Model*, POM) [92]. Los CLMs extienden la regresión logística binaria para predecir categorías en una escala ordinal. De esta forma, se basan en estimar las probabilidades acumuladas de pertenencia a varias categorías $P(y \preceq \mathcal{C}_j|\mathbf{x})$ que se pueden relacionar de forma directa con las probabilidades de pertenencia a una sola categoría:

$$P(y \preceq \mathcal{C}_q|\mathbf{x}) = P(y = \mathcal{C}_1|\mathbf{x}) + \dots + P(y = \mathcal{C}_q|\mathbf{x}),$$

$$P(y = \mathcal{C}_q|\mathbf{x}) = P(y \preceq \mathcal{C}_q|\mathbf{x}) - P(y \preceq \mathcal{C}_{q-1}|\mathbf{x}),$$

donde $1 \leq q \leq Q$ y considerando por definición que $P(y \preceq \mathcal{C}_Q|\mathbf{x}) = 1$. La formulación general de este tipo de modelos puede definirse como:

$$g^{-1}(P(y \preceq \mathcal{C}_q|\mathbf{x})) = \beta_q - \mathbf{w}^T \mathbf{x}, 1 \leq q \leq Q,$$

donde $g^{-1} : [0, 1] \rightarrow (-\infty, +\infty)$ es una función monótona a menudo conocida como la función de enlace inversa y β_q es el umbral para la clase \mathcal{C}_q . Este método de umbral considera por tanto un modelo lineal para la función $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$. De esta forma, se predice la etiqueta \mathcal{C}_q si y solo si $f(\mathbf{x}) \in [\beta_{q-1}, \beta_q]$, donde la función f y $\beta = (\beta_0, \beta_1, \dots, \beta_{Q-1}, \beta_Q)$ deben determinarse a partir de los datos de entrenamiento, asumiendo que $\beta_0 = -\infty$ y $\beta_Q = +\infty$. La función inversa de enlace más comúnmente utilizada es la función *logit* (inversa de la función logística) aunque existen otras posibilidades [8].

- Máquinas de vectores soporte (*Support Vector Machines*, SVMs): Es bien conocido que las SVMs son formuladas de forma natural para clasificación binaria y que su extensión a problemas multiclase nominales se basa en descomposiciones de tipo “uno contra uno” o “uno contra todos”. La extensión de las SVMs a problemas de clasificación ordinal es directa y

natural, de manera que el problema de optimización de las SVMs puede reformularse como la búsqueda de $Q - 1$ hiperplanos paralelos, todos ellos representados por un vector de proyección común \mathbf{w} y los umbrales escalares $\beta_1 \leq \dots \leq \beta_{Q-1}$, separando de esta forma los datos en clases ordenadas. En este sentido, Chu y Keerthi [24] proponen dos enfoques para optimizar \mathbf{w} y todos los umbrales, asegurando la restricción $\beta_1 \leq \dots \leq \beta_{Q-1}$. El primer enfoque incluye restricciones de desigualdad explícitas en los umbrales (*Support Vector Ordinal Regression with Explicit Constraints, SVOR-EX*). La segunda aproximación (*SVOR with Implicit Constraints, SVOR-IM*), no incorpora las restricciones de desigualdad debido a que los patrones de todas las clases contribuyen al cálculo de los errores de cada umbral.

- Análisis discriminante para clasificación ordinal [122]: Esta metodología introduce una restricción que impone una separación mínima en las proyecciones más cercanas de dos puntos pertenecientes a clases adyacentes en la escala ordinal. Esta restricción es empleada para preservar la información de orden en las categorías. Este método es conocido como *Kernel Discriminant Learning for Ordinal Regression, (KDLOR)* [98, 122].
- Modelos generativos: Estos modelos aprenden un modelo de la probabilidad conjunta $P(\mathbf{x}, y)$ de los patrones con características de entrada \mathbf{x} y la etiqueta y , para luego, mediante un procedimiento Bayesiano, realizar la predicción en base a una estimación de la probabilidad $P(y|\mathbf{x})$. Dentro de esta familia, el método más importante es el de procesos Gaussianos para clasificación ordinal (*Gaussian Processes for Ordinal Regression, GPOR*) [23], el cual modela la variable latente $f(\mathbf{x})$ utilizando procesos Gaussianos y , posteriormente, estima todos los parámetros mediante un procedimiento Bayesiano. Se asume que los valores de la función latente $\{f(\mathbf{x}_i)\}$ son dados por variables aleatorias indexadas por el patrón de entrada en un proceso Gaussiano de media cero. Para aproximar la covarianza entre las funciones correspondientes a dos patrones se utilizan funciones tipo *kernel*.

9.2 ENFOQUE MULTI OBJETIVO EN CLASIFICACIÓN ORDINAL

El primer acercamiento de la clasificación ordinal a la clasificación multiobjetivo consiste en la transformación del problema ordinal multiobjetivo en un problema de optimización global o en un problema con un único objetivo. Estas transformaciones se pueden realizar con las mismas técnicas empleadas en clasificación nominal (en la Subsección 4.1.2 se describen estas transformaciones).

En la literatura existen pocos trabajos que afronten la clasificación ordinal desde un punto de vista multiobjetivo. En [127] se resuelve un problema ordinal considerando 6 objetivos simultáneamente: MZE, MAE y cuatro formulaciones diferentes de

la exactitud del rango esperado. En [20] emplean el método SVM para maximizar, independientemente, los márgenes superior e inferior con la finalidad de evitar los errores de clasificación producidos al clasificar un patrón en alguna clase mayor o menor, respectivamente.

En la presente Tesis se propone la modificación de ciertos elementos de los MOEAs nominales para permitir la resolución de problemas ordinales. Estas modificaciones radican principalmente en dos elementos: cambiar el modelo y emplear funciones objetivo ordinales.

- Modificar el modelo de ANN para aprovechar la información ordinal contenida en la base de datos y favorecer la obtención de una predicción que respete el orden existente entre las diferentes clases. El modelo de ANN para clasificación ordinal empleado puede ser visto en detalle en la Subsección 9.2.1.
- Utilizar funciones objetivo ordinales para guiar la evolución del MOEA y poder evaluar correctamente los modelos ordinales obtenidos. Una revisión sobre las diferentes métricas ordinales existentes en la literatura es presentada en la Sección 9.3. Además, en dicha Sección, se realiza un estudio para determinar el par de métricas ordinales que mejor podría guiar la evolución de un MOEA. Recordemos que para favorecer la diversidad en el frente de Pareto y garantizar la obtención de soluciones aceptables es necesario que las funciones objetivo empleadas en el MOEA no estén correladas.
- Además de los dos aspectos anteriores, es necesario modificar algunos de los elementos de los MOEAs (operadores de cruce y mutación) con el fin de asegurar que las restricciones de orden impuestas por el modelo de ANN ordinal se vean cumplidas.

9.2.1 Modelo de Red Neuronal para Clasificación Ordinal

Uno de los aspectos principales de la clasificación ordinal es el desconocimiento de la distancia precisa entre las diferentes clases. Este inconveniente se suele solventar etiquetando los patrones del problema mediante un conjunto de categorías que mantienen un orden entre ellas. El modelo de red neuronal ordinal empleado en la presente Tesis está basado en el modelo ordinal POM [92], puesto que tiene en cuenta el orden de las etiquetas a la hora de realizar la clasificación. La Figura 9.1 muestra el modelo de red neuronal basado en el modelo POM [48]. Como se puede observar, este modelo está compuesto por dos elementos: una red neuronal y una capa lineal adicional.

- El primer elemento es una red neuronal con cualquier tipo de nodo en capa oculta y con un único nodo en la capa de salida.

9.2 ENFOQUE MULTIOBJETIVO EN CLASIFICACIÓN ORDINAL

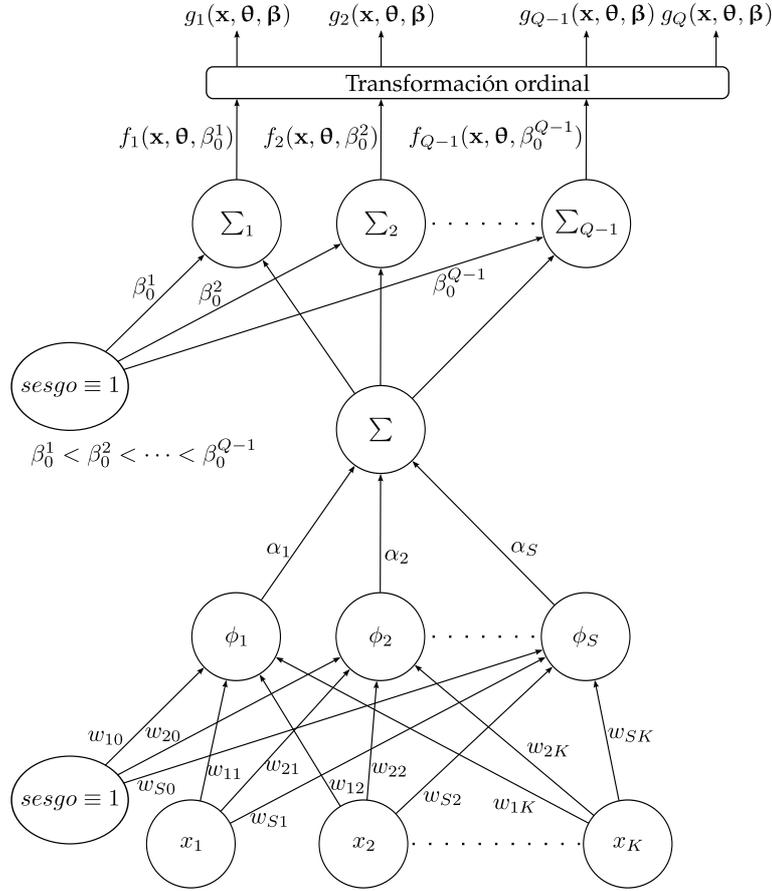


Figura 9.1: Modelo de red neuronal empleado en clasificación ordinal.

- El segundo elemento es una capa lineal con tantos nodos como el número de clases menos uno. La entrada de esta capa lineal es la salida de la red neuronal anterior y un valor β para cada uno de los nodos. Estos valores de β permitirán diferenciar las clases entre si. Las salidas de esta capa son trasladadas a una recta de manera ordenada, lo que facilita la clasificación ordinal.

La regla de clasificación de este modelo puede ser representada, de manera genérica, según la siguiente expresión:

$$C(\mathbf{x}) = \begin{cases} C_1, & \text{si } f(\mathbf{x}, \boldsymbol{\theta}) \leq \beta_0^1 \\ C_2, & \text{si } \beta_0^1 < f(\mathbf{x}, \boldsymbol{\theta}) \leq \beta_0^2, \\ \dots & \\ C_Q, & \text{si } f(\mathbf{x}, \boldsymbol{\theta}) > \beta_0^{Q-1} \end{cases}, \quad (9.1)$$

donde el conjunto de sesgos o umbrales $\beta = (\beta_0^1, \dots, \beta_0^{Q-1})$ satisface la siguiente restricción ordinal: $\beta_0^1 \leq \beta_0^2 \leq \dots \leq \beta_0^{Q-1}$. \mathbf{x} es el vector de variables predictoras del patrón a clasificar, θ es el vector de parámetros del modelo y $f(\mathbf{x}, \theta)$ es una función de ordenamiento. De hecho, el análisis de la expresión (9.1) revela la idea general presentada anteriormente: los patrones son proyectados a una recta usando la función de ordenamiento y separados en clases ordenadas mediante los sesgos o umbrales.

Antes de describir el modelo, es necesario definir formalmente el problema de clasificación ordinal. La finalidad del problema es obtener, tras el proceso de entrenamiento, un modelo que sea capaz de predecir las etiquetas de clase, $\mathcal{C} = \{C_1, C_2, \dots, C_Q\}$, para nuevos patrones. La diferencia entre un problema ordinal y uno nominal es la relación de orden $C_1 \prec C_2 \prec \dots \prec C_Q$ impuesta en el problema (el símbolo \prec indica el orden entre las distintas clases).

Siendo $D = \{(\mathbf{x}_i, y_i), i = 1, \dots, N\}$ el conjunto de patrones empleados para entrenar el modelo, donde $\{y_1, y_2, \dots, y_N\}$ es el conjunto de etiquetas dadas en la base de datos, las etiquetas predichas por un clasificador se definen como $\{y_1^*, y_2^*, \dots, y_N^*\}$, donde $y_i \in \mathcal{C}$ y $y_i^* \in \mathcal{C}$, y $\mathcal{C} = \{C_1, C_2, \dots, C_Q\}$, $1 \leq i \leq N$. En general, las predicciones de los clasificadores serán una de las categorías, pero, para algunas métricas, estas categorías se convertirán en valores enteros mediante el uso de la función $\mathcal{O}(y_i^*)$, la cual establece la posición de la escala ordinal de la etiqueta predicha, siendo $\mathcal{O}(C_q) = q$ y si $y_i = C_q$ entonces $\mathcal{O}(y_i) = q$, $1 \leq q \leq Q$.

El modelo para cada clase se puede definir formalmente como:

$$f_q(\mathbf{x}, \theta, \beta_0^q) = f(\mathbf{x}, \theta) - \beta_0^q, \quad 1 \leq q \leq Q - 1,$$

donde la función de proyección $f(\mathbf{x}, \theta)$ es estimada usando S funciones de base sigmoide, $f(\mathbf{x}, \theta) = \sum_{s=1}^S \alpha_s \phi_s(\mathbf{x}, \mathbf{w}_s)$, reemplazando $\phi_s(\mathbf{x}, \mathbf{w}_s)$ por la ecuación sigmoideal:

$$\phi_s(\mathbf{x}, \mathbf{w}_s) = \frac{1}{1 + \exp\left(-w_{s0} - \sum_{k=1}^K w_{sk} x_k\right)}, \quad 1 \leq s \leq S,$$

donde K es el número de entradas.

En el modelo POM [92], esta proyección puede ser usada para obtener las probabilidades acumuladas:

$$P(Y \leq q) = P(Y = 1) + \dots + P(Y = q), \quad 1 \leq q \leq Q - 1,$$

$$P(Y \leq Q) = 1,$$

las posibilidades (*odds*) acumuladas:

$$\text{odds}(Y \leq q) = \frac{P(Y \leq q)}{1 - P(Y \leq q)}, \quad 1 \leq q \leq Q - 1,$$

y las transformaciones *logits* acumuladas:

$$\text{logit}(Y \leq q) = \ln \left(\frac{P(Y \leq q)}{1 - P(Y \leq q)} \right) = f(\mathbf{x}, \boldsymbol{\theta}) - \beta_0^q = f_q(\mathbf{x}, \boldsymbol{\theta}, \beta_0^q),$$

siendo

$$P(Y \leq q) = \frac{1}{1 + \exp(f(\mathbf{x}, \boldsymbol{\theta}) - \beta_0^q)} = \frac{1}{1 + \exp(f_q(\mathbf{x}, \boldsymbol{\theta}, \beta_0^q))},$$

donde $1 \leq q \leq Q - 1$, $P(Y = q)$ es la probabilidad que tiene el patrón \mathbf{x} de pertenecer a la clase q -ésima, $P(Y \leq q)$ es la probabilidad que tiene el patrón \mathbf{x} de pertenecer a las clases comprendidas entre la primera y la j -ésima y la transformación *logit* se modela usando la función de ordenamiento, $f(\mathbf{x}, \boldsymbol{\theta})$, y el correspondiente umbral, β_0^q .

Podemos volver a $P(Y = q)$ desde $P(Y \leq q)$ según:

$$P(y_n^{(q)} = 1 | \mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\beta}) = P(Y = q) = g_q(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\beta}) = P(Y \leq q) - P(Y \leq q - 1)$$

para $q = 1, \dots, Q$, donde $y_n^{(q)} = 1$ si el patrón \mathbf{x} pertenece a la clase q y 0 en otro caso. La probabilidad final del modelo puede ser expresada como:

$$g_q(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\beta}) = \frac{1}{1 + \exp(f_q(\mathbf{x}, \boldsymbol{\theta}, \beta_0^q))} - \frac{1}{1 + \exp(f_{q-1}(\mathbf{x}, \boldsymbol{\theta}, \beta_0^{q-1}))},$$

para $1 \leq q \leq Q - 1$. Para la última clase, la probabilidad final se expresa como:

$$g_Q(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\beta}) = 1 - \frac{1}{1 + \exp(f_{Q-1}(\mathbf{x}, \boldsymbol{\theta}, \beta_0^{Q-1}))} = \frac{\exp(f_{Q-1}(\mathbf{x}, \boldsymbol{\theta}, \beta_0^{Q-1}))}{1 + \exp(f_{Q-1}(\mathbf{x}, \boldsymbol{\theta}, \beta_0^{Q-1}))}.$$

De esta forma, obtenemos un modelo POM generalizado donde la función de transformación o proyección $f(\mathbf{x}, \boldsymbol{\theta})$ ya no es lineal sino sigmoideal (o de cualquier otro tipo de función de base).

9.3 MÉTRICAS EN CLASIFICACIÓN ORDINAL

Considerando un problema de clasificación ordinal en el que las distintas clases están etiquetadas de la siguiente forma: “bajo”, “medio-bajo”, “medio”, “medio-alto”, “alto”; es claro ver que el error producido al predecir un patrón de la clase “bajo” como de la clase “medio-alto” es más grave que si se hubiera predicho como de la clase “medio-bajo”. Este pequeño ejemplo muestra la necesidad de evaluar los problemas ordinales con métricas específicas. Generalmente, las métricas ordinales están desarrolladas basadas en la correlación entre la clase real y la clase predicha o a partir de la asociación entre los valores reales y los predichos.

Las métricas ordinales basadas en la correlación se centran en la distancia absoluta entre la clase real y la clase predicha. Las más comunes en *Machine Learning* son el

error absoluto medio (*Mean Absolute Error*, MAE) [10, 43], la raíz del error cuadrático medio (*Root Mean Square Error*, RMSE) [43] y la tasa de error (*Mean Zero-One Error*, MZE) [43], donde $MZE = 1 - CCR$. Sin embargo, estas tres medidas no son adecuadas para evaluar el desempeño de los clasificadores en bases de datos ordinales desbalanceadas [10]. Para solventar este inconveniente, existen otras métricas como el valor medio de los MAEs de cada clase (*Average MAE*, AMAE) o el máximo de los MAEs (*Maximum MAE*, MMAE). Estas métricas se detallan en la Subsección 9.3.1.

Las medidas de asociación se basan en la relación de orden entre la clase real y la clase predicha, sin necesidad de medir la distancia entre ellas. Es decir, estas medidas no se ven influenciadas por la codificación numérica de la clase. El coeficiente de correlación de Spearman (r_s) [119] y el τ_b de Kendall [80] son las métricas de asociación más comunes en la literatura (estas medidas se tratan con más detalle en la Subsección 9.3.2).

A continuación se presentarán las métricas basadas en la correlación y, acto seguido, las medidas de asociación.

9.3.1 Medidas Ordinales Basadas en Correlación

Todas estas medidas analizan la distancia existente entre la clase real y la clase predicha por el clasificador. Para el cálculo de las distancias se utiliza una matriz de confusión similar a la mostrada en la Tabla 5.1 de la Sección 5.1, considerando un problema ordinal con Q clases y N patrones.

- Error absoluto medio (*Mean Absolute Error*, MAE): Desviación absoluta media entre la clase predicha y la real [10]:

$$MAE = \frac{1}{N} \sum_{q,k=1}^Q |q - k|n_{qk} = \frac{1}{N} \sum_{i=1}^n e(x_i),$$

donde $e(x_i) = |\mathcal{O}(y_i) - \mathcal{O}(y_i^*)|$ es la distancia entre el valor real (y_i) y el predicho (y_i^*) y n_{qk} es el número de patrones de la clase q predichos como de la clase k . Esta métrica toma valores en el rango $[0, Q - 1]$.

- MAE medio (*Average MAE*, AMAE): Esta métrica (propuesta por Baccianella *et al.* [10]) se define como la media de los MAEs de cada una de las clases. El AMAE es especialmente útil para mitigar el efecto producido por el desbalanceo de las clases del problema.

El MAE de la clase q -ésima se define como:

$$MAE_q = \frac{1}{n_{q\bullet}} \sum_{k=1}^Q |q - k|n_{qk} = \frac{1}{n_{q\bullet}} \sum_{i=1}^{n_{q\bullet}} e_q(x_i), 1 \leq q \leq Q,$$

donde $n_{q\bullet}$ es el número de patrones de la clase q . De modo que:

$$MAE = \frac{1}{N} \sum_{q=1}^Q n_{q\bullet} MAE_q.$$

Por tanto, el AMAE se puede definir como:

$$AMAe = \frac{1}{Q} \sum_{q=1}^Q MAE_q.$$

Esta medida toma valores en el rango $[0, Q - 1]$.

- MAE máximo (*Maximum MAE*, MMAE): El MMAE se define como el máximo de los MAEs, es decir, el valor de MAE de la clase con la mayor distancia entre los valores predichos y los reales. Esta métrica ha sido propuesta por el autor de la presente Tesis en [38].

$$MMAE = \max\{MAE_q; q = 1, \dots, Q\}.$$

Esta métrica toma valores en el rango $[0, Q - 1]$ y es una extensión natural de la métrica MS para problemas de clasificación ordinal.

- Índice Kappa de Cohen ponderado (*Weighted Kappa*, WK): Modificación del índice Kappa que permite asignar diferentes pesos a los diferentes niveles de agregación entre las clases predichas y reales [57]:

$$WK = \frac{p_o(w) - p_e(w)}{1 - p_e(w)},$$

donde

$$p_o(w) = \frac{1}{N} \sum_{q,k=1}^Q w_{qk} n_{qk},$$

y

$$p_e(w) = \frac{1}{N^2} \sum_{q,k=1}^Q w_{qk} n_{q\bullet} n_{\bullet k},$$

donde el peso $w_{qk} = |q - k|$ mide el grado de discrepancia entre la clase real y la predicha. Esta métrica, al igual que el índice kappa original, toma valores en el rango $[-1, 1]$, donde 1 significa un acuerdo perfecto, -1 un completo desacuerdo y 0 indica que no existe relación.

Aunque en la presente Tesis sólo se analicen estas métricas ordinales basadas en correlación, existen otras en la literatura como el coeficiente r_{int} [103] o el índice de clasificación ordinal [19].

9.3.2 Medidas de Asociación Ordinales

Estas métricas miden la relación de orden entre las clases predichas y las reales, pero sin emplear la distancia entre las clases. Por tanto, estas métricas no necesitan utilizar la matriz de confusión.

- r_S : El coeficiente de correlación de Spearman es una medida no paramétrica de la dependencia entre dos variables [119]. Para el caso de problemas ordinales, estas variables son las etiquetas de clase reales y predichas.

$$r_S = \frac{\sum_{i=1}^N (\mathcal{O}(y_i) - \overline{\mathcal{O}(y)}) (\mathcal{O}(y_i^*) - \overline{\mathcal{O}(y^*)})}{\sqrt{\sum_{i=1}^N (\mathcal{O}(y_i) - \overline{\mathcal{O}(y)})^2} \sqrt{\sum_{i=1}^N (\mathcal{O}(y_i^*) - \overline{\mathcal{O}(y^*)})^2}},$$

donde $\overline{\mathcal{O}(y)}$ y $\overline{\mathcal{O}(y^*)}$ representan, respectivamente, la media de $\mathcal{O}(y_i)$ y $\mathcal{O}(y_i^*)$, $i = 1, \dots, N$. Los valores de esta métrica están en el rango $[-1, 1]$.

- τ_b : El τ_b de Kendall es un estadístico usado para medir el grado de asociación entre dos variables [80]. Al igual que con la métrica r_S , las dos variables son las etiquetas de clase reales y predichas.

$$\tau_b = \frac{\sum_{i,j=1}^N c_{ij}^* c_{ij}}{\sqrt{\left(\sum_{i,j=1}^N c_{ij}^{*2}\right) \left(\sum_{i,j=1}^N c_{ij}^2\right)}},$$

donde c_{ij}^* es $+1$ si $\mathcal{O}(y_i^*) > \mathcal{O}(y_j^*)$, 0 si $\mathcal{O}(y_i^*) = \mathcal{O}(y_j^*)$, y -1 si $\mathcal{O}(y_i^*) < \mathcal{O}(y_j^*)$ para $i, j = 1, \dots, N$, y similar para c_{ij} . Esta métrica toma valores en el rango $[-1, 1]$.

Existen otras medidas de asociación ordinales a parte de las tratadas en la presente Tesis. Una de las más conocidas es la curva ROC ordinal [127], calculada a partir de la proyección de todos los patrones sobre una recta.

9.3.3 Correlación entre las Métricas Ordinales

Estudiar las relaciones existentes entre las diferentes métricas empleadas en la clasificación ordinal resulta útil, puesto que las métricas menos relacionadas guiarán el proceso evolutivo de un algoritmo multiobjetivo de una manera más eficaz, dado que si los objetivos (las métricas en nuestro caso) están correlados no tiene sentido plantear un algoritmo multiobjetivo. El estudio realizado consiste en el análisis de una matriz de correlaciones obtenida a partir de los resultados producidos por 4 algoritmos de la literatura sobre 10 bases de datos ordinales (Automobile, Balance-scale, Bondrate, ERA, ESL, Eucalyptus, LEV, SWD, Toy, Winequality-red). En el estudio experimental se ha empleado un 30-*holdout* estratificado, donde aproximadamente el

75 % de los patrones son empleados para entrenar y el 25 % restante para generalizar. Las métricas que serán analizadas en este estudio son las presentadas en las Subsecciones anteriores junto con las medidas nominales CCR y MS. Estas medidas nominales han sido incluidas en el estudio debido a que son las funciones objetivo empleadas por los MOEAs nominales presentados en la presente Tesis.

Los 4 algoritmos empleados son ampliamente usados en la literatura y han sido seleccionados debido al buen desempeño que presentan generalmente. El primero de ellos es el algoritmo SVM nominal (en la presente Tesis se ha empleado el software LibSVM [21], considerando la métrica MAE como criterio para la selección de los hiperparámetros durante el procedimiento de validación cruzada), mientras que los otros 3 algoritmos son específicos para clasificación ordinal: REDSVM, SVOR-EX y SVOR-IM (más información sobre estos algoritmos ordinales en la Sección 9.1).

El estudio se ha realizado separadamente para cada uno de los algoritmos, con la finalidad de tomar en consideración los efectos de cada clasificador en los resultados. Estas matrices son generadas de la siguiente manera: primero, se obtienen 30 modelos con cada algoritmo (un modelo por cada *holdout*). A continuación, la correlación entre cada par de métricas es calculada obteniendo un total de 10 matrices de correlación de dimensión 8×8 (una matriz por cada base de datos). Entonces, si el valor de la correlación entre un par de métricas es mayor que 0,75, se suma un punto al par de métricas correspondiente. Finalmente, el total de puntos obtenidos por cada par es dividido por el número total de comparaciones (10) para obtener el porcentaje de veces que este par de métricas obtiene una correlación superior a 0,75. Cuando una de las dos métricas del par analizado (o ambas) presenta un valor constante en los 30 modelos, el par es ignorado, dado que la correlación no puede ser calculada. Por ejemplo, la correlación entre las medidas CCR y MS con los resultados del método LibSVM es 16,67, debido a que sólo 1 comparación es mayor que 0,75 y 4 comparaciones no han podido ser calculadas, por lo que solo se han realizado $10 - 4 = 6$ comparaciones ($1/6 = 0,1667 = 16,67\%$).

Este proceso es repetido para cada uno de los algoritmos y los resultados obtenidos se muestran en la Tabla 9.2. Un resumen de estos cuatro estudios ha sido incluido en la Tabla 9.3. Esta matriz ha sido generada teniendo en cuenta los resultados presentados por los cuatro métodos conjuntamente. Como se puede observar, las conclusiones de los 5 estudios (4 individuales y 1 conjunto) son muy similares.

Según los resultados mostrados en la Tabla 9.3, los pares de medidas que están menos correlados (con un valor por debajo del 20 %) son CCR-MMAE, MAE-MMAE, MMAE- r_S , MMAE- τ_b y MMAE-WK. De los cinco pares, los tres últimos son muy similares, debido a que las métricas r_S , τ_b y WK están altamente correladas (valores superiores al 90 %). Por tanto, en nuestro estudio solo consideraremos, de estos tres pares, el par formado por MMAE y τ_b . La elección de τ_b se debe a que proporciona una visión intuitiva de los datos y a que es una de las métricas más usadas en clasificación ordinal.

Tabla 9.2: Matrices de correlación obtenidas por los diferentes algoritmos. Cada elemento es igual al porcentaje de veces (de un total de 10 comparaciones) donde la correlación entre las métricas es mayor que 0,75.

Algoritmo LibSVM										Algoritmo REDSVM									
CCR	MS	MAE	AMAE	MMAE	t_S	t_b	WK	CCR	MS	MAE	AMAE	MMAE	t_S	t_b	WK				
CCR	100,00	16,67	80,00	30,00	10,00	40,00	60,00	80,00	CCR	100,00	*	77,78	33,33	*	33,33	44,44	77,78		
MS	—	100,00	16,67	33,33	33,33	16,67	16,67	16,67	MS	—	100,00	*	16,67	33,33	*	*	*		
MAE	—	—	100,00	30,00	10,00	80,00	80,00	80,00	MAE	—	—	100,00	22,22	*	88,89	88,89	88,89		
AMAE	—	—	—	100,00	50,00	50,00	50,00	60,00	AMAE	—	—	—	100,00	55,56	33,33	33,33	44,44		
MMAE	—	—	—	—	100,00	20,00	10,00	10,00	MMAE	—	—	—	—	100,00	11,11	11,11	11,11		
t_S	—	—	—	—	—	—	100,00	100,00	90,00	t_S	—	—	—	—	—	100,00	100,00	77,78	
t_b	—	—	—	—	—	—	—	100,00	100,00	t_b	—	—	—	—	—	—	100,00	77,78	
WK	—	—	—	—	—	—	—	100,00	WK	—	—	—	—	—	—	—	100,00		
Algoritmo SVOR-EX										Algoritmo SVOR-IM									
CCR	MS	MAE	AMAE	MMAE	t_S	t_b	WK	CCR	MS	MAE	AMAE	MMAE	t_S	t_b	WK				
CCR	100,00	40,00	90,00	40,00	20,00	40,00	50,00	80,00	CCR	100,00	40,00	90,00	40,00	20,00	40,00	50,00	80,00		
MS	—	100,00	40,00	40,00	80,00	40,00	40,00	40,00	MS	—	100,00	40,00	40,00	80,00	40,00	40,00	40,00		
MAE	—	—	100,00	40,00	20,00	80,00	90,00	80,00	MAE	—	—	100,00	40,00	20,00	80,00	90,00	100,00		
AMAE	—	—	—	100,00	60,00	30,00	50,00	60,00	AMAE	—	—	—	100,00	60,00	50,00	50,00	60,00		
MMAE	—	—	—	—	100,00	20,00	20,00	20,00	MMAE	—	—	—	—	100,00	20,00	20,00	20,00		
t_S	—	—	—	—	—	—	100,00	100,00	t_S	—	—	—	—	—	100,00	100,00	90,00		
t_b	—	—	—	—	—	—	—	100,00	t_b	—	—	—	—	—	—	100,00	90,00		
WK	—	—	—	—	—	—	—	100,00	WK	—	—	—	—	—	—	—	100,00		

* significa que una de las métricas comparadas (o ambas) tenía valores constantes para todas las comparaciones y, por tanto, la correlación no puede ser calculada.
Las correlaciones menores que el 20% están marcadas en **negrita**.

Tabla 9.3: Matriz resumen de los 4 estudios de correlaciones. Cada elemento es igual al porcentaje de veces (de un total de 40 comparaciones) donde la correlación entre las métricas es mayor que 0,75.

	CCR	MS	MAE	AMAE	MMAE	r_s	τ_b	WK
CCR	100,00	22,73	84,62	35,90	12,82	38,46	51,28	79,49
MS	—	100,00	22,73	31,82	54,55	22,73	22,73	22,73
MAE	—	—	100,00	33,33	12,82	82,05	87,18	87,18
AMAE	—	—	—	100,00	56,41	41,03	46,15	56,41
MMAE	—	—	—	—	100,00	17,95	15,38	15,38
r_s	—	—	—	—	—	100,00	100,00	89,74
τ_b	—	—	—	—	—	—	100,00	92,31
WK	—	—	—	—	—	—	—	100,00

Las correlaciones menores que el 20 % están marcadas en **negrita**.

Lógicamente, cualquiera de los tres pares de medidas seleccionados (CCR-MMAE, MAE-MMAE y MMAE- τ_b) podrían guiar de manera satisfactoria la evolución de un MOEA, debido a la baja correlación entre ellas, lo que implícitamente significa que son objetivos no cooperativos. A continuación, intentaremos discernir con cuál de estos pares de métricas se obtienen mejores resultados para poder analizarlo en más profundidad.

Para realizar la comparación de los tres pares de métricas, guiaremos el algoritmo MPDENN presentado en la Subsección 4.3.2 con cada uno de estos pares y analizaremos los resultados obtenidos sobre las 10 bases de datos ordinales utilizadas en el estudio anterior. Nueve de ellas son bases de datos de referencia¹ y otra (Toy) ha sido generada siguiendo las indicaciones dadas en [102].

Puesto que el algoritmo MPDENN es estocástico, se ha usado un diseño 10-*holdout* estratificado con 3 repeticiones por cada *holdout* (obteniendo un total de $10 \times 3 = 30$ resultados distintos), donde aproximadamente el 75 % de los patrones son empleados para entrenar y el 25 % restante para generalizar. De esta forma consideramos aleatoriedad en los datos y en el algoritmo evolutivo.

Los principales parámetros del algoritmo han sido obtenidos empleando un procedimiento de validación cruzada, considerando la métrica MAE como el criterio para determinar la mejor combinación. La búsqueda en malla *grid* desarrollada emplea los valores {10, 25, 50} para el tamaño de la población, {100, 150, 200} para el número de generaciones, y {4, 5, 6} para el número de nodos sigmoides en la capa oculta.

La Tabla 9.4 muestra los valores de CCR obtenidos después de guiar el MOEA con los tres pares de métricas. Los resultados se corresponden con las medias y las desviaciones típicas obtenidas en generalización por los 30 modelos (cada modelo es

¹ Bases de datos disponibles en <http://weka.wikispaces.com/Datasets> y <http://mldata.org/>.

obtenido en una ejecución distinta y se corresponde con uno de los extremos del frente de Pareto. Más información en la Sección 4.5). Además, la Tabla 9.4 incluye el *ranking* medio obtenido por cada metodología sobre todas las bases de datos. De igual modo, las Tablas 9.5, 9.6 y 9.7 presentan los valores obtenidos para las métricas MAE, MMAE y τ_b , respectivamente.

Tabla 9.4: Resultados obtenidos para la métrica CCR en generalización: media y desviación típica (Media_{DT}) y *ranking* medio (\bar{R}_{CCR}).

Base de datos	Par CCR-MMAE		Par MAE-MMAE		Par τ_b -MMAE	
	Mod. CCR	Mod. MMAE	Mod. MAE	Mod. MMAE	Mod. τ_b	Mod. MMAE
Automobile	43,916,57	41,54 _{6,26}	42,12 _{7,02}	41,60 _{6,13}	31,86 _{9,39}	31,54 _{9,06}
Balance-scale	90,47 _{2,49}	90,57 _{2,65}	90,57 _{2,62}	<u>90,62_{2,60}</u>	90,64_{2,15}	<u>90,62_{2,13}</u>
Bondrate	49,11 _{10,13}	48,89 _{10,11}	52,89_{10,20}	<u>51,56_{10,20}</u>	32,67 _{16,73}	33,78 _{15,46}
ERA	23,80 _{2,21}	23,13 _{2,37}	25,25_{2,54}	<u>24,25_{2,55}</u>	22,05 _{3,42}	23,08 _{2,67}
ESL	<u>62,40_{7,32}</u>	61,26 _{7,28}	64,26_{6,68}	62,16 _{9,36}	30,68 _{16,87}	37,40 _{15,30}
Eucalyptus	44,51_{2,74}	<u>43,44_{2,50}</u>	42,99 _{4,18}	42,37 _{4,17}	40,62 _{7,80}	41,23 _{7,23}
LEV	58,31 _{3,53}	52,96 _{4,94}	59,55_{2,89}	54,92 _{3,56}	49,48 _{5,66}	49,57 _{4,91}
SWD	<u>53,71_{2,94}</u>	50,29 _{3,96}	54,93_{2,57}	52,35 _{3,70}	48,29 _{7,86}	50,17 _{3,86}
Toy	<u>68,80_{13,60}</u>	<u>68,80_{13,60}</u>	73,78_{10,88}	73,78_{10,88}	59,11 _{21,78}	59,11 _{21,62}
Winequality-red	56,17_{2,82}	43,24 _{8,46}	55,35 _{2,01}	41,02 _{10,07}	24,82 _{6,89}	27,37 _{6,80}
\bar{R}_{CCR}	2,45	3,70	1,80	2,80	5,35	4,90

Los mejores resultados están marcadas en **negrita** y los segundos mejores en subrayado.

Tabla 9.5: Resultados obtenidos para la métrica MAE en generalización: media y desviación típica (Media_{DT}) y *ranking* medio (\bar{R}_{MAE}).

Base de datos	Par CCR-MMAE		Par MAE-MMAE		Par τ_b -MMAE	
	Mod. CCR	Mod. MMAE	Mod. MAE	Mod. MMAE	Mod. τ_b	Mod. MMAE
Automobile	0,7955 _{0,1332}	0,8179 _{0,1217}	0,7750_{0,1384}	<u>0,7833_{0,1283}</u>	0,9224 _{0,1726}	0,9282 _{0,1698}
Balance-scale	0,1081 _{0,0313}	0,1064 _{0,0328}	0,1047 _{0,0296}	0,1042_{0,0293}	<u>0,1045_{0,0265}</u>	0,1047 _{0,0263}
Bondrate	0,6778 _{0,1576}	0,6911 _{0,1501}	0,5956_{0,1249}	<u>0,6178_{0,1174}</u>	0,8822 _{0,2832}	0,8533 _{0,2634}
ERA	1,3613 _{0,0978}	1,4272 _{0,1466}	1,2576_{0,0849}	<u>1,3584_{0,1118}</u>	1,5276 _{0,2507}	1,4584 _{0,1472}
ESL	0,4243 _{0,0976}	0,4352 _{0,0936}	0,3888_{0,0776}	<u>0,4197_{0,1282}</u>	0,9680 _{0,3775}	0,7850 _{0,2870}
Eucalyptus	0,8529 _{0,0586}	0,8476 _{0,0507}	0,7911 _{0,0721}	0,7989 _{0,0673}	0,7989 _{0,1467}	0,7774_{0,1111}
LEV	<u>0,4525_{0,0396}</u>	0,5321 _{0,0715}	0,4393_{0,0341}	0,5040 _{0,0452}	0,5737 _{0,0740}	0,5753 _{0,0733}
SWD	<u>0,4961_{0,0390}</u>	0,5473 _{0,0574}	0,4741_{0,0344}	0,5211 _{0,0497}	0,5821 _{0,1228}	0,5512 _{0,0521}
Toy	<u>0,3449_{0,1698}</u>	0,3449 _{0,1698}	0,2711_{0,1186}	0,2711_{0,1186}	0,4196 _{0,2313}	0,4178 _{0,2261}
Winequality-red	0,4840_{0,0349}	0,6880 _{0,1287}	0,4874 _{0,0233}	0,7262 _{0,1504}	0,9940 _{0,1964}	0,9261 _{0,1677}
\bar{R}_{MAE}	3,25	4,05	1,50	<u>2,40</u>	5,15	4,65

Los mejores resultados están marcadas en **negrita** y los segundos mejores en subrayado.

Tabla 9.6: Resultados obtenidos para la métrica MMAE en generalización: media y desviación típica (Media_{DT}) y *ranking* medio (\bar{R}_{MMAE}).

Base de datos	Par CCR-MMAE		Par MAE-MMAE		Par τ_b -MMAE	
	Mod. CCR	Mod. MMAE	Mod. MAE	Mod. MMAE	Mod. τ_b	Mod. MMAE
Automobile	1,7944 _{0,4953}	1,7440 _{0,5143}	1,5084 _{0,5173}	1,4812 _{0,5201}	1,4387 _{0,3641}	1,4376_{0,3670}
Balance-scale	0,2452 _{0,1463}	0,2379 _{0,1494}	0,1936_{0,1115}	0,1936_{0,1115}	0,2310 _{0,1707}	<u>0,2287_{0,1723}</u>
Bondrate	2,2667 _{0,7739}	2,2722 _{0,7637}	1,9278_{0,5735}	<u>1,9722_{0,5290}</u>	2,1111 _{0,6966}	2,0333 _{0,7076}
ERA	2,2279 _{0,4034}	<u>2,1109_{0,3283}</u>	2,2040 _{0,3435}	2,0302_{0,2491}	2,4887 _{0,8828}	2,1717 _{0,5972}
ESL	1,5805 _{0,5762}	<u>1,4550_{0,5092}</u>	1,4950 _{0,5366}	1,4617 _{0,5353}	1,4654 _{0,3896}	1,3330_{0,3681}
Eucalyptus	1,3289 _{0,1281}	1,2623 _{0,1385}	1,2147 _{0,1495}	<u>1,2102_{0,1583}</u>	1,2216 _{0,2791}	1,1425_{0,2526}
LEV	1,1739 _{0,2591}	0,9645 _{0,3168}	1,1893 _{0,2506}	<u>0,9205_{0,2780}</u>	0,9371 _{0,2309}	0,8921_{0,2261}
SWD	0,9947 _{0,2100}	<u>0,7985_{0,1606}</u>	1,0620 _{0,1493}	0,8127 _{0,2039}	0,8600 _{0,2084}	0,7520_{0,1496}
Toy	<u>0,7009_{0,3358}</u>	<u>0,7009_{0,3358}</u>	0,6466_{0,2953}	0,6466_{0,2953}	0,8305 _{0,2869}	0,8246 _{0,2813}
Winequality-red	2,0794 _{0,2073}	1,6396 _{0,4024}	2,0272 _{0,1567}	1,5385 _{0,3866}	<u>1,3964_{0,3793}</u>	1,2858_{0,3262}
\bar{R}_{MMAE}	5,35	3,85	3,70	<u>2,20</u>	3,90	2,00

Los mejores resultados están marcadas en **negrita** y los segundos mejores en subrayado.

Tabla 9.7: Resultados obtenidos para la métrica τ_b en generalización: media y desviación típica (Media_{DT}) y *ranking* medio (\bar{R}_{τ_b}).

Base de datos	Par CCR-MMAE		Par MAE-MMAE		Par τ_b -MMAE	
	Mod. CCR	Mod. MMAE	Mod. MAE	Mod. MMAE	Mod. τ_b	Mod. MMAE
Automobile	0,4590 _{0,1288}	0,4584 _{0,1238}	0,4830 _{0,1389}	0,4761 _{0,1330}	0,5488_{0,0882}	<u>0,5446_{0,0879}</u>
Balance-scale	0,8991 _{0,0302}	0,9008 _{0,0316}	<u>0,9036_{0,0271}</u>	0,9040_{0,0268}	0,9031 _{0,0250}	0,9030 _{0,0249}
Bondrate	0,2497 _{0,2671}	0,2315 _{0,2576}	0,3634_{0,1931}	0,3509 _{0,1829}	0,3429 _{0,2036}	<u>0,3538_{0,1738}</u>
ERA	0,4479 _{0,0369}	0,4448 _{0,0394}	0,4726_{0,0394}	0,4602 _{0,0364}	<u>0,4705_{0,0318}</u>	0,4627 _{0,0288}
ESL	0,8388 _{0,0293}	0,8391 _{0,0292}	<u>0,8466_{0,0252}</u>	0,8455 _{0,0253}	0,8594_{0,0157}	0,8449 _{0,0252}
Eucalyptus	0,4833 _{0,0396}	0,4891 _{0,0390}	0,5325 _{0,0551}	0,5313 _{0,0528}	0,5872_{0,0754}	<u>0,5756_{0,0740}</u>
LEV	0,6363 _{0,0394}	0,6277 _{0,0341}	0,6398 _{0,0277}	0,6358 _{0,0291}	0,6530_{0,0283}	<u>0,6475_{0,0295}</u>
SWD	0,5004 _{0,0415}	0,4971 _{0,0471}	<u>0,5171_{0,0428}</u>	0,5079 _{0,0329}	0,5241_{0,0339}	0,5168 _{0,0345}
Toy	0,7970 _{0,1112}	0,7970 _{0,1112}	0,8526_{0,0654}	0,8526_{0,0654}	0,8487 _{0,0709}	<u>0,8492_{0,0713}</u>
Winequality-red	0,4551 _{0,0387}	0,4304 _{0,0474}	0,4434 _{0,0366}	0,4305 _{0,0455}	0,4910_{0,0309}	<u>0,4887_{0,0338}</u>
\bar{R}_{τ_b}	5,05	5,65	<u>2,25</u>	3,45	1,90	2,70

Los mejores resultados están marcadas en **negrita** y los segundos mejores en subrayado.

Desde un punto de vista descriptivo, para la métrica CCR, el mejor *ranking* es obtenido por el par MAE-MMAE (modelo MAE) y el segundo mejor por el par CCR-MMAE (modelo CCR). Para la métrica MAE, los dos mejores *rankings* son alcanzados por el par MAE-MMAE. Para la medida MMAE, el mejor *ranking* es obtenido por el par τ_b -MMAE (modelo MMAE) y el segundo mejor por el par MAE-MMAE (modelo MAE). Por último, para la métrica τ_b , el mejor *ranking* es alcanzado por el

par τ_b -MMAE (modelo τ_b) y el segundo mejor por el par MAE-MMAE (modelo MAE). Estos resultados muestran que el MOEA presenta un mejor desempeño en las bases de datos ordinales cuando la evolución es guiada por el par MAE-MMAE. En general, el modelo MAE obtiene buenos resultados en las métricas de clasificación de carácter global (CCR, MAE y τ_b) mientras que el modelo MMAE se centra más en el comportamiento de la clase peor clasificada (MMAE). En la siguiente Subsección se analiza la relación existente entre estas dos métricas y su comportamiento como funciones objetivo de un MOEA.

En un análisis previo a este estudio se empleó el par formado por las métricas AMAE y MMAE [32] para resolver un problema de clasificación ordinal en el área de la microbiología predictiva [33]. Este par de métricas no va a ser estudiado en la presente Tesis debido a que el par MAE-MMAE presenta menos correlación entre las métricas y, por consiguiente, obtiene mejores resultados.

9.3.4 Relación entre el Par de Métricas MAE y MMAE

Según los resultados vistos anteriormente, el par más prometedor para guiar la evolución de un MOEA es el formado por las métricas MAE y MMAE. A continuación se muestra el procedimiento seguido para estudiar la relación entre estas dos medidas.

Proposición 9.3.1 *Considerando un problema de clasificación ordinal con Q clases, siendo las métricas MAE y MMAE dos medidas asociadas a un clasificador ordinal y considerando, sin pérdida de generalidad, que la clase con el máximo MAE se corresponde con la última ($MMAE = MAE_Q$), entonces:*

$$p_Q^* MMAE \leq MAE \leq MMAE, \quad (9.2)$$

donde p_Q^* es la frecuencia relativa de la clase Q -ésima estimada a priori:

$$p_Q^* = \frac{n_{Q\bullet}}{N},$$

donde $n_{Q\bullet}$ es el número de patrones de la clase Q .

Demostración Comenzando por el límite superior de la inecuación, se puede comprobar que en general:

$$0 \leq MAE_q \leq MMAE \leq Q - 1,$$

de manera que:

$$MAE = \frac{\sum_{q=1}^Q n_{q\bullet} MAE_q}{N} = \frac{n_{1\bullet} MAE_1 + n_{2\bullet} MAE_2 + \dots + n_{Q\bullet} MAE_Q}{N} \leq$$

$$\leq \frac{n_{1\bullet} \text{MMAE} + n_{2\bullet} \text{MMAE} + \dots + n_{Q\bullet} \text{MMAE}}{N} = \text{MMAE} \left(\frac{\sum_{q=1}^Q n_{q\bullet}}{N} \right) = \text{MMAE}.$$

Por otro lado, el límite inferior se obtiene de la siguiente forma:

$$\begin{aligned} \text{MAE} &= \frac{n_{1\bullet} \text{MAE}_1 + n_{2\bullet} \text{MAE}_2 + \dots + n_{Q\bullet} \text{MAE}_Q}{N} \geq \frac{n_{Q\bullet} \text{MAE}_Q}{N} = \\ &= \frac{n_{Q\bullet} \text{MMAE}}{N} = p_Q^* \text{MMAE}, \end{aligned}$$

puesto que:

$$n_{1\bullet} \text{MAE}_1 + n_{2\bullet} \text{MAE}_2 + \dots + n_{(Q-1)\bullet} \text{MAE}_{Q-1} \geq 0.$$

Corolario 9.3.2 *Si la base de datos está completamente balanceada, es decir, todas las clases tienen el mismo número de patrones, entonces:*

$$n_{1\bullet} = n_{2\bullet} = \dots = n_{Q\bullet} = \frac{N}{Q} \text{ y } p_q^* = \frac{1}{Q}.$$

Por consiguiente:

$$\frac{\text{MMAE}}{Q} \leq \text{MAE} \leq \text{MMAE}.$$

9.3.4.1 Representación Gráfica del Par MMAE-MAE

El par de métricas MMAE-MAE nos permite representar el desempeño de un clasificador ordinal en un espacio de dos dimensiones. La métrica MMAE es representada en el eje horizontal y la medida MAE en el eje vertical, de manera que un clasificador (un punto en el espacio (MMAE, MAE)) domina a otro si se encuentra por debajo y a la izquierda, es decir, si tiene valores más pequeños de MMAE y MAE. Además, todos los clasificadores estarán localizados en la región blanca de la Figura 9.2. Esta región se obtiene a partir de la inecuación 9.2.

Existen algunos puntos de esta región factible que requieren una atención especial. El peor clasificador se encuentra localizado en el extremo superior derecho (el punto $(Q-1, Q-1)$). En el punto $(0,0)$ se localiza el clasificador óptimo, es decir, aquel que ha realizado una clasificación perfecta. El último punto a destacar es el punto $(Q-1, p_Q^*(Q-1))$, que se corresponde con un clasificador que ha predicho correctamente todas las clases salva una, en la que se ha cometido el mayor error de clasificación posible. Considerando un problema con 5 clases (“bajo”, “medio-bajo”, “medio”, “medio-alto”, “alto”), un clasificador en el punto $(Q-1, p_Q^*(Q-1))$ habrá clasificado todos los patrones de la clase “bajo” en la clase “alto”, o viceversa. Destacar que es posible encontrar clasificadores con un nivel bajo de MAE, pero con un valor elevado de MMAE, especialmente cuando el número de clases aumenta. Por

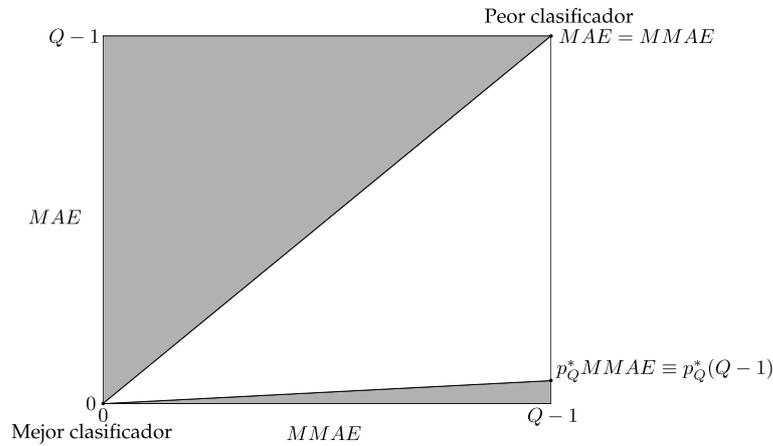


Figura 9.2: Región factible (zona blanca) en el espacio (MMAE, MAE).

tanto, minimizar estas dos funciones de error simultáneamente produce una situación de compromiso en la que los modelos obtenidos presentan buenos valores de clasificación global y valores aceptables de clasificación para la clase peor clasificada (la clase que tiene mayor error entre las clases reales y las predichas), lo que, en general, es muy difícil de obtener en problemas de clasificación ordinal.

Al analizar la región factible, observamos que una reducción del valor del MAE no implica necesariamente una disminución del valor del MMAE. De igual modo, una reducción del valor del MMAE no significa una disminución del valor del MAE. Por otro lado, destacar que para un valor fijo de MAE, un clasificador con dicho valor es mejor cuanto más cerca se encuentre de la diagonal del cuadrado $(Q - 1) \times (Q - 1)$.

La Figura 9.3 muestra gráficamente los resultados obtenidos en una de las 30 ejecuciones para la base de datos SWD al emplear el par MAE-MMAE para guiar la evolución del MOEA. Se observa que el frente de Pareto obtenido en entrenamiento (Figura 9.3(b)) no se mantiene con los patrones de generalización (Figura 9.3(c)). En general, la estructura del frente de Pareto en entrenamiento no se mantiene en generalización.

9.3 MÉTRICAS EN CLASIFICACIÓN ORDINAL

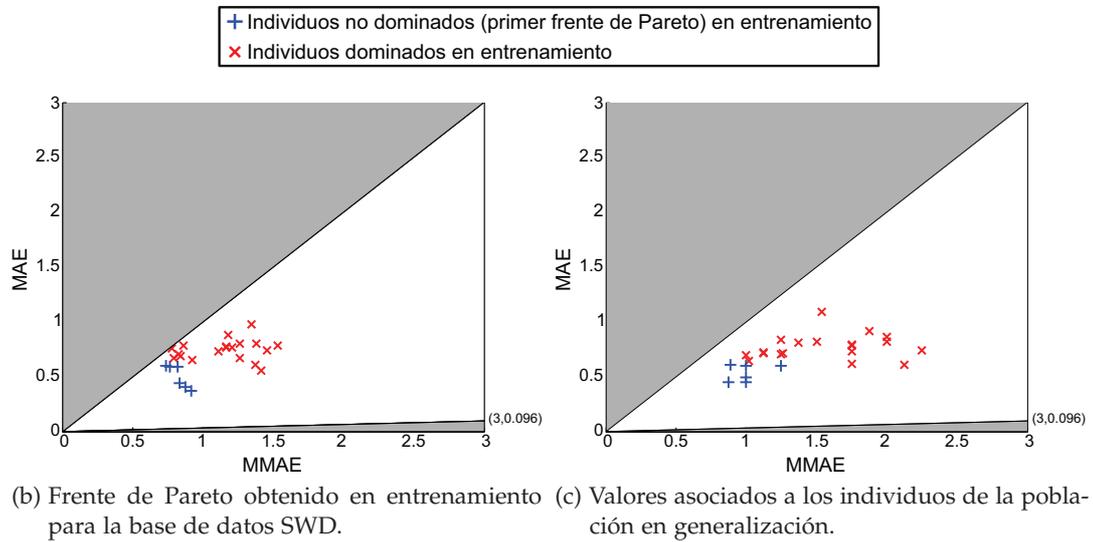


Figura 9.3: Frente de Pareto en entrenamiento y valores asociados en generalización obtenidos con la base de datos SWD.

COMENTARIOS FINALES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

En este Capítulo se expondrán las principales conclusiones alcanzadas durante la realización de la presente Tesis, así como las futuras líneas de investigación que serán estudiadas a corto o medio plazo.

10.1 CONCLUSIONES

- Se ha realizado una breve descripción del uso de las redes neuronales artificiales en los problemas de clasificación, detallando la estructura del modelo y los distintos tipos de funciones de base empleados. Algunas de estas funciones de base son muy novedosas en *Machine Learning*.
- Se han estudiado las diferentes técnicas empleadas para hacer frente a la resolución de problemas con múltiples objetivos, haciendo especial hincapié en los algoritmos basados en el concepto de dominancia de Pareto.
- Así mismo, se ha realizado una revisión bibliográfica de los principales algoritmos evolutivos multiobjetivo existentes en la literatura que emplean el concepto de dominancia de Pareto. De todos estos algoritmos, se han analizado en detalle el algoritmo PDE (*Pareto Differential Evolution*), basado en la evolución diferencial, y el clásico algoritmo NSGAI2 (*Non-dominated Sorting Genetic Algorithm 2*) desarrollados por H. Abbass y K. Deb, respectivamente. Además se ha estudiado un algoritmo memético desarrollado a partir del NSGAI2 llamado MPENSGAI2 (*Memetic Pareto NSGAI2*).
- En la fase de diseño de algoritmos evolutivos multiobjetivo, se ha desarrollado e implementado un algoritmo evolutivo tomando como base los trabajos realizados por H. Abbass, al que se ha llamado PDENN (*PDE with Neural Networks*). El algoritmo PDENN optimiza de manera conjunta la topología y los pesos de red. Además, el coste computacional asociado a la obtención del frente de Pareto final es mucho menor que en los algoritmos de H. Abbass, ya que se ha

utilizado una manera más flexible de incorporar y sustituir los individuos de la población tras finalizar una generación del proceso evolutivo del algoritmo.

- Se han propuesto varias versiones mejoradas del algoritmo anterior, las cuales incorporan un procedimiento de búsqueda local en diferentes puntos de la evolución. Estos algoritmos aplican el algoritmo de búsqueda local *iRprop⁺* a diferentes individuos de la población, como a los pertenecientes al primer frente de Pareto o a los más representativos de este frente. Estos algoritmos meméticos obtienen resultados competitivos con respecto al algoritmo memético propuesto por H. Abbass (MPANN) con un tiempo de computo inferior.
- Una vez construido el frente óptimo de Pareto, es necesario seleccionar un modelo perteneciente a dicho frente que presente un compromiso entre las métricas que evalúan la calidad del clasificador. Esta no es una elección trivial, pues depende de las necesidades del problema que se esté resolviendo. En la presente Tesis se han presentado las técnicas de selección más empleadas en la literatura, así como los métodos de *ensembles* más conocidos. Además, se proponen dos métodos de selección basados en *clustering*, los cuales buscan seleccionar los individuos más representativos de la población.
- También se ha realizado una revisión de las métricas empleadas en el análisis del rendimiento de un clasificador multiclase. Estas métricas son utilizadas para guiar el entrenamiento de los modelos de red neuronal mediante el algoritmo evolutivo. En concreto, se ha estudiado la relación entre las medidas de precisión (CCR) y mínima sensibilidad (MS) desde un punto de vista multiobjetivo. El uso de estas dos métricas nos permite obtener clasificadores con un alto nivel de precisión global, así como un aceptable nivel de clasificación para cada clase, aspecto muy importante en problemas reales, como en medicina o en microbiología predictiva.
- Una de las características fundamentales de la presente Tesis ha sido la utilización de los algoritmos y modelos presentados en la resolución de complejos problemas reales de diferentes áreas de conocimiento, como en medicina, donde se ha desarrollado un sistema de ayuda a la decisión que permite determinar qué paciente tendrá mayor probabilidad de aceptar el órgano tras la operación; en teledetección remota, donde se ha determinado el uso del terreno en diferentes fincas de olivos a través de imágenes aéreas, con el objetivo de ayudar a las administraciones andaluza y europea a identificar aquellas fincas que desarrollan una agricultura de conservación; o en microbiología predictiva, donde se han obtenido modelos que permiten predecir la capacidad de crecimiento de un microorganismo en determinadas circunstancias ambientales.
- Un segundo enfoque de la presente Tesis es abordar los problemas de entrenamiento de modelos para clasificación ordinal, dado que es una temática que

en los últimos años está teniendo un importante auge en el área de *Machine Learning*. Así, en un primer acercamiento a esta temática se ha realizado una revisión bibliográfica sobre este tema, prestando especial interés en los algoritmos relacionados con los problemas multiobjetivo. Tras comprobar que no existen muchos trabajos al respecto, se han estudiado qué modificaciones son necesarias para permitir que los algoritmos multiobjetivo nominales resuelvan problemas de clasificación ordinal. Estas modificaciones se centran, principalmente, en dos elementos: el modelo utilizado y las funciones objetivo que guían el proceso evolutivo.

- En una primera aproximación, se ha propuesto un modelo de red neuronal basado en el modelo ordinal POM (*Proportional Odds Model*). Este modelo pertenece a la familia ordinal de modelos de umbral y es entrenado teniendo en cuenta la relación de orden existente entre las diferentes clases del problema.
- Una parte muy significativa de este segundo enfoque ha consistido en revisar y analizar las diferentes métricas ordinales existentes en la literatura empleadas para evaluar la bondad de un clasificador ordinal. Tras esta revisión se descubrió que no existía en la literatura ninguna métrica ordinal que midiera el rendimiento de la clase peor clasificada. Por este motivo, en la presente Tesis se ha expuesto la métrica MMAE (*Maximum Mean Absolute Error*), utilizada para evaluar el desempeño de la clase con mayor desviación absoluta media entre los valores predichos y los reales.
- Una vez realizada la revisión, ha sido necesario realizar un análisis para determinar el par de métricas ordinales que mejor guiará la evolución de un algoritmo multiobjetivo. El par que presenta los resultados más competitivos es el formado por las métricas MAE y MMAE. Cuando se emplean estas métricas para guiar el entrenamiento de un algoritmo evolutivo multiobjetivo se obtienen modelos ordinales que presentan un alto nivel de clasificación global, así como un aceptable nivel de clasificación para la clase peor clasificada. Esto último es muy importante en problemas desbalanceados, ya que permite que la clase minoritaria no sea olvidada durante el proceso evolutivo.
- Por último, para favorecer la comprensión de los resultados obtenidos por los algoritmos multiobjetivo sobre los problemas de clasificación ordinal, se ha estudiado el espacio bidimensional (MAE, MMAE), definiendo la región factible donde están localizados todos los modelos obtenidos con el algoritmo multiobjetivo. La representación gráfica de los modelos en este espacio permite una lectura rápida y sencilla del rendimiento de los clasificadores.

10.2 FUTURAS LÍNEAS DE INVESTIGACIÓN

- Diseñar e implementar un algoritmo evolutivo multiobjetivo centrado en clasificación ordinal. Este algoritmo deberá emplear durante el entrenamiento de los modelos la información de orden contenida en las etiquetas de clase. Del mismo modo, tendrá que utilizar funciones ordinales para evaluar el rendimiento de los modelos durante la fase de entrenamiento, por lo que es necesario seguir investigando en el desarrollo de métricas contrapuestas que contemplen esta información de orden.
- Incorporar en el algoritmo mencionado un sistema de remuestreo dinámico (remuestreo a lo largo de la evolución) con el que mejorar el rendimiento de las clases minoritarias del problema ordinal. Normalmente, las clases minoritarias se corresponden con las clases de los extremos, sobre todo si la variable de respuesta está medida en una escala de Likert, y suelen ser ignoradas por los modelos durante el proceso evolutivo, por lo que representan el peor valor en la métrica MMAE, es decir, el valor de MMAE del modelo acaba correspondiendo con el valor de MAE de la clase minoritaria. Este problema puede solucionarse empleando una versión ordinal del conocido algoritmo de remuestreo SMOTE (*Synthetic Minority Oversampling Technique*). La implementación de esta versión ordinal debe tener en cuenta la relación de orden entre las distintas clases a la hora de crear los nuevos patrones de la clase minoritaria.
- Resolución de problemas de clasificación ordinal mediante los algoritmos evolutivo multiobjetivo propuestos en la presente Tesis. Estos algoritmos hacen uso de un modelo ordinal que captura la relación entre las diferentes clases del problema, y de funciones objetivo ordinales, como el par de métricas MAE y MMAE.
- Implementación de técnicas de comparación de frentes de Pareto para decidir si un frente es mejor que otro y poder comparar el desempeño alcanzado por los diferentes algoritmos evolutivos multiobjetivo.
- Estudio de diferentes métodos de *ensembles* y de su posible adaptación a clasificación ordinal, de manera que se garantice que la solución proporcionada por el *ensemble* respete la relación de orden entre las diferentes clases.

CONCLUDING REMARKS AND FUTURE RESEARCH LINES

This Chapter presents the main conclusions achieved from this Thesis, as well as future research lines that will be studied in the short or medium term.

11.1 CONCLUSIONS

- A brief description of the use of artificial neural networks in classification problems has been done, detailing the structure of the models and the different types of basis functions considered. Some of these basis functions are quite novel for the Machine Learning community.
- The different techniques used to deal with the resolution of multiobjective problems have been studied, with particular emphasis on algorithms based on the concept of Pareto dominance.
- A literature review of the main existing multiobjective evolutionary algorithms which use the concept of Pareto dominance has been done. From these algorithms, we have analyzed in detail the PDE (Pareto Differential Evolution) algorithm, which is based on differential evolution, and the classic NSGAI (Non-dominated Sorting Genetic Algorithm 2) algorithm proposed by H. Abbass and K. Deb, respectively. In addition, a memetic algorithm based on NSGAI called MPENSGAI (Memetic Pareto NSGAI) has been studied.
- An evolutionary algorithm has been proposed and implemented based on the work done by H. Abbass, which has been called PDENN (PDE with Neural Networks). The PDENN algorithm jointly optimizes the topology and network weights. Furthermore, the computational cost associated with obtaining the Pareto front is much lower than in the algorithms of H. Abbass, because the new algorithm uses a more flexible manner of incorporating and replacing individuals in the population after finishing a generation of the evolutionary process.
- Several memetic algorithms generated from the PDENN algorithm have been proposed, which incorporate a local search procedure at different points of evo-

lution. These algorithms use the local search algorithm *iRprop*⁺ applied to different individuals in the population, such as those belonging to the first Pareto front or the most representative of this front. These memetic algorithms obtain competitive results compared to the memetic algorithm proposed by H. Abbass (MPANN) with a lower computation time.

- Once the optimal Pareto front is obtained, the selection of a model belonging to this front which satisfies a compromise between the metrics measuring the “goodness of the classifier” was necessary. This is not a trivial challenge, because it usually depends on the requirements of the problem. In this Thesis, the selection techniques used in the literature have been presented along with the most popular ensembles methods. In addition, we propose two selection methods based on clustering, which try to select the most representative individuals of the population.
- The Thesis has also reviewed the metrics used in the analysis of the performance of a multiclass classifier which guide the training of neural network models using the evolutionary algorithm. In particular, we studied the relationship between the measures accuracy (CCR) and minimum sensitivity (MS) from a multi-objective perspective. The use of these two metrics allows us to obtain classifiers with a high level of overall accuracy and an acceptable level of accuracy for each class, which is very important on real-world problems, such as in medicine or predictive microbiology.
- One of the fundamental characteristics of this Thesis has been the use of algorithms and models presented in the resolution of complex real-world problems from different areas of knowledge, such as medicine, where a decision support system has been developed to determine which patient is more likely to accept the organ after a liver transplantation; remote sensing, where land use in different olive farms was determined through aerial images, with the aim of helping Andalusian and European administrations to identify those farms that develop conservation agriculture; or predictive microbiology, where the models obtained are used to determine the ability of a microorganism to grow under certain environmental circumstances.
- A second important point of this Thesis is that related to training models for ordinal classification, because it is a field of study which is having an important development in the area of Machine Learning during recent years. Thus, in a first approximation to this issue, a literature review on this subject has been made, with particular interest in algorithms related to multiobjective problems. After verifying that there are no many works in this direction, we have studied what changes could be made in nominal multiobjective algorithms to allow the

resolution of ordinal problems. These changes focus mainly on two elements: the model used and the objective functions that guide the evolutionary process.

- In a first approach, a neural network model based on the ordinal model POM (Proportional Odds Model) has been proposed. This model belongs to the family of threshold models and it is trained taking into account the order relation existing between the different classes of the problem.
- One significant part of this second approach has been reviewing and analyzing the different ordinal metrics used in the literature to evaluate the goodness of an ordinal classifier. After this review, it was discovered that there was no ordinal metric that measures the performance of the worse classified class. For this reason, this Thesis discussed the MMAE metric (Maximum Mean Absolute Error), used to evaluate the performance of the class with the highest average absolute deviation between predicted and real labels.
- Besides studying ordinal metrics, an analysis to determine the ordinal metric pair could better guide the evolution of a multiobjective algorithm has been carried out. The pair that presents the most competitive results is formed by the metric MAE and MMAE. When these metrics are used to guide the training of a multiobjective evolutionary algorithm, ordinal models obtained show a high level of overall classification and a classification level acceptable for the worst classified class. This is very important in unbalanced problems because it allows the minority class not to be forgotten during the evolutionary process.
- Finally, to facilitate understanding of the results obtained by the multiobjective algorithms on the ordinal problems, the two-dimensional space (MAE, MMAE) has been studied, determining the feasible region where all the models obtained with the multiobjective algorithm are located. The graphical representation of the models in this space allows a quick and easy reading of the performance of classifiers.

11.2 FUTURE RESEARCH LINES

- To design and implement a multiobjective evolutionary algorithm focused on ordinal classification. The algorithm should consider the order information contained in the class labels for training of the models. Furthermore, the algorithm should use ordinal metrics to evaluate the performance of the models during the training phase, so further research is needed in the development and study of conflicting metrics which consider this information of order.
- The aforementioned algorithm could include a dynamic resampling method (resampling patterns along the evolution) to improve the performance for the

minority classes of the ordinal problem. Usually, minority classes correspond to classes of the extremes, especially if the response variable is measured on a Likert scale, and they are often ignored by the models during the evolutionary process, so a high value is obtained for the MMAE metric, i.e. the MMAE value of the model corresponds, in general, with the value of the MAE for the minority class. This problem can be solved using an ordinal version of the well-known resampling algorithm Synthetic Minority Oversampling Technique (SMOTE). The implementation of this ordinal version should consider the order relation between classes when creating new patterns of the minority class.

- To solve ordinal problems using the multiobjective evolutionary algorithms proposed in this Thesis. These algorithms use an ordinal model which captures the relationship between the different classes of the problem, and also ordinal objective functions, such as the pair of metrics MAE and MMAE.
- Implementation of techniques for comparing Pareto fronts in order to decide whether a front is better than another and to compare the performance achieved by different multiobjective evolutionary algorithms.
- Study of different ensemble methods and their possible adaptation to ordinal classification, in order to ensure that the predictions provided by the ensemble consider the order relation between the different labels.

BIBLIOGRAFÍA

- [1] H.A. Abbass. A memetic Pareto evolutionary approach to artificial neural networks. En *Proceedings of the 14th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence in Lecture Notes In Computer Science*, págs. 1–12, 2001.
- [2] H.A. Abbass. The self-adaptive Pareto differential evolution algorithm. En *Congress on Evolutionary Computation, CEC02*, págs. 831–836, 2002.
- [3] H.A. Abbass. Pareto neuro-evolution: constructive ensemble of neural networks using multi-objective optimization. En *IEEE Congress on Evolutionary Computation CEC2003*, volumen 3, págs. 2074–2080, 2003.
- [4] H.A. Abbass y R. Sarker. The Pareto differential evolution algorithm. *International Journal on Artificial Intelligence Tools*, 11(4):531–552, 2002.
- [5] H.A. Abbass, R. Sarker y C. Newton. PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems. En *Proceedings of the 2001 Congress on Evolutionary Computation*, volumen 2, págs. 971–978, 2001.
- [6] S.F. Adra, A.I. Hamody, I. Griffin y P.J. Fleming. A hybrid multi-objective evolutionary algorithm using an inverse neural network for aircraft control system design. En *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, volumen 1, págs. 1–8, 2005.
- [7] A. Agresti. *Analysis of ordinal categorical data*. New York: Wiley, 1984.
- [8] A. Agresti. *Categorical Data Analysis*. John Wiley and Sons, 2ª edición, 2002.
- [9] J.A. Anderson. Regression and ordered categorical variables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(1):1–30, 1984.
- [10] S. Baccianella, A. Esuli y F. Sebastiani. Evaluation measures for ordinal regression. En *Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications, ISDA'09*, págs. 283–287, 2009.
- [11] T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, UK, 1996.

BIBLIOGRAFÍA

- [12] R. Bender y U. Grouven. Using binary logistic regression models for ordinal data with non-proportional odds. *Journal of Clinical Epidemiology*, 51(10):809–816, 1998.
- [13] C.M. Bishop. A generalization of the stone-weierstrass theorem. *Pacific Journal of Mathematics*, 11(3):777–783, 1961.
- [14] C.M. Bishop. Improving the generalization properties of radial basis function neural networks. *Neural Computation*, 3(4):579–581, 1991.
- [15] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1ª edición, 1995.
- [16] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [17] J. Briceño, G. Solorzano y C. Pera. A proposal for scoring marginal liver grafts. *Transplant International*, 13:S249–S252, 2000.
- [18] J. Briceño, M. Cruz-Ramírez, M. Prieto, M. Navasa, J. Ortiz de Urbina, R. Orti, M.A. Gómez-Bravo, A. Otero, E. Varo, S. Tomé, G. Clemente, R. Bañares, R. Bárcena, V. Cuervas-Mons, G. Solórzano, C. Vinaixa, Á. Rubín, J. Colmenero, A. Valdivieso, R. Ciria, C. Hervás-Martínez y M. de la Mata. Application of artificial intelligence as an innovative donor-recipient matching model for liver transplantation: results from a multicenter spanish study. *Transplantation*, enviado, 2013.
- [19] J.S. Cardoso y R. Sousa. Measuring the performance of ordinal classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(8):1173–1195, 2011.
- [20] E. Carrizosa y B. Martín-Barragán. Maximizing upgrading and downgrading margins for ordinal regression. *Mathematical Methods of Operations Research*, 74(3):381–407, 2011.
- [21] C.C. Chang y C.J. Lin. LibSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.
- [22] Y. Chauvin y D.E. Rumelhart. *Backpropagation: Theory, Architectures, and Applications*. Lawrence Erlbaum Associates Inc, Mahwah, NJ, USA, 1995.
- [23] W. Chu y Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005.
- [24] W. Chu y S.S. Keerthi. Support vector ordinal regression. *Neural Computation*, 19(3):792–815, 2007.

- [25] C.A. Coello Coello. Evolutionary multi-objective optimization: basic concepts and some applications in pattern recognition. En *Proceedings of the Third Mexican conference on Pattern recognition, MCPR'11*, págs. 22–33, 2011.
- [26] C.A. Coello Coello, G.B. Lamont y D.A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., 2ª edición, 2007.
- [27] D.W. Corne, J.D. Knowles y M.J. Oates. The Pareto envelope-based selection algorithm for multiobjective optimization. En *Proceedings of the parallel problem solving from nature VI conference*, págs. 839–848, 2000.
- [28] D.W. Corne, J.D. Knowles, N.R. Jerram y M.J. Oates. PESAI: Region-based selection in evolutionary multiobjective optimization. En *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, págs. 283–290, 2001.
- [29] M. Cruz-Ramírez, J.C. Fernández, J. Sánchez-Monedero, F. Fernández-Navarro, C. Hervás-Martínez, P.A. Gutiérrez y M.T. Lamata. Ensemble determination using the TOPSIS decision support system in multi-objective evolutionary neural network classifiers. En *10th International Conference on Intelligent Systems Design and Applications (ISDA2010)*, págs. 513–518, 2010.
- [30] M. Cruz-Ramírez, J. Sánchez-Monedero, F. Fernández-Navarro, J. Fernández y C. Hervás-Martínez. Memetic Pareto differential evolutionary artificial neural networks to determine growth multi-classes in predictive microbiology. *Evolutionary Intelligence*, 3(3-4):187–199, 2010.
- [31] M. Cruz-Ramírez, J. C. Fernández, F. Fernández-Navarro, J. Sánchez-Monedero y C. Hervás-Martínez. Selecting the best artificial neural network model from a multi-objective Differential Evolution Pareto front. En *IEEE Symposium on Differential Evolution (IEEE Symposium Series on Computational Intelligence) (SDE (SSCI2011))*, págs. 96–103, 2011.
- [32] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero y P. A. Gutierrez. A preliminary study of ordinal metrics to guide a multi-objective evolutionary algorithm. En *11th International Conference on Intelligent Systems Design and Applications, ISDA 2011*, págs. 1176–1181, 2011.
- [33] M. Cruz-Ramírez, J.C. Fernández, A. Valero, P.A. Gutiérrez y C. Hervás-Martínez. Multiobjective Pareto ordinal classification for predictive microbiology. En *Soft Computing Models in Industrial and Environmental Applications*, volumen 188 de *Advances in Intelligent Systems and Computing*, págs. 153–162, 2012.
- [34] M. Cruz-Ramírez, C. Hervás-Martínez, J.C. Fernandez-Caballero, J. Briceño y M. de la Mata. Multi-Objective Evolutionary Algorithm for Donor-Recipient

BIBLIOGRAFÍA

- Decision System in Liver Transplants. *European Journal of Operational Research*, 222(2):317–327, 2012.
- [35] M. Cruz-Ramírez, C. Hervás-Martínez, P.A. Gutiérrez, M. Pérez-Ortiz, J. Briceño y M. de la Mata. Memetic Pareto differential evolutionary neural network used to solve an unbalanced liver transplantation problem. *Soft Computing*, 17(29):275–284, 2012.
- [36] M. Cruz-Ramírez, C. Hervás-Martínez, M. Jurado-Expósito y F. López-Granados. A multi-objective neural network based method for cover crop identification from remote sensed data. *Expert Systems with Applications*, 39(11):10038–10048, 2012.
- [37] M. Cruz-Ramírez, C. Hervás-Martínez, J.C. Fernández, J. Briceño y M. de la Mata. Predicting patient survival after liver transplantation using evolutionary multi-objective artificial neural networks. *Artificial Intelligence in Medicine*, 58(1):37–49, 2013.
- [38] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero y P.A. Gutiérrez. Metrics to guide a multi-objective evolutionary algorithm for ordinal classification. *Neurocomputing*, aceptado, 2013.
- [39] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.
- [40] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, LTD, 2004.
- [41] K. Deb, A. Pratab, S. Agarwal y T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA2. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [42] K. Deb, M. Mohan y S. Mishra. Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolutionary Computation*, 13(4):501–525, 2005.
- [43] K. Dembczyński, W. Kotłowski y R. Slowiński. Ordinal classification with decision rules. En *Proceedings of the ECML/PKDD'07 workshop on Mining Complex Data, Warsaw, PL*, págs. 169–181, 2007.
- [44] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
- [45] W.Y. Deng, Q.H. Zheng, S. Lian, L. Chen y X. Wang. Ordinal extreme learning machine. *Neurocomputing*, 74(1–3):447–456, 2010.

- [46] H. Dikkers y L. Rothkrantz. Support vector machines in ordinal classification: An application to corporate credit scoring. *Neural Network World*, 15(6):491–507, 2005.
- [47] D.L. Donoho y I.M. Johnstone. Projection-based approximation and a duality with kernel methods. *The Annals of Statistics*, 17(1):58–106, 1989.
- [48] M. Dorado-Moreno, P.A. Gutiérrez y C. Hervás-Martínez. Ordinal Classification Using Hybrid Artificial Neural Networks with Projection and Kernel Basis Functions. En *7th International Conference on Hybrid Artificial Intelligence Systems (HAIS2012)*, págs. 319–330, 2012.
- [49] R.O. Duda, P.E. Hart y D.G. Stork. *Pattern Classification*. Wiley Interscience, New York, 2ª edición, 2000.
- [50] R. Durbin y D. Rumelhart. Products units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation*, 1(1):133–142, 1989.
- [51] P. Dutkowsky, C.E. Oberkofler, K. Slankamenac, M.A. Puhán, E. Schädde, B. Müllhaupt, A. Geier y P.A. Clavien. Are there better guidelines for allocation in liver transplantation? A novel score targeting justice and utility in the model for end-stage liver disease era. *Annals of Surgery*, 254(5):745–753, 2011.
- [52] A. Engelbrecht y A. Ismail. Training product unit neural networks. *Stability and Control: Theory and Applications*, 2(1–2):59–74, 1999.
- [53] Y. Fan. Ordinal ranking for detecting mild cognitive impairment and alzheimer’s disease based on multimodal neuroimages and csf biomarkers. En *Multimodal Brain Image Analysis*, volumen 7012 de *Lecture Notes in Computer Science*, págs. 44–51. 2011.
- [54] S. Feng, N.P. Goodrich, J.L. Bragg-Gresham, D.M. Dykstra, J.D. Punch, M.A. DebbRoy, S.M. Greenstein y R.M. Merion. Characteristics associated with liver graft failure: The concept of a donor risk index. *American Journal of Transplantation*, 6(4):783–790, 2006.
- [55] J.C. Fernández, F.J. Martínez, C. Hervás y P.A. Gutiérrez. Sensitivity versus accuracy in multi-class problems using memetic Pareto evolutionary neural networks. *IEEE Transactions on Neural Networks*, 21(5):750–770, 2010.
- [56] F. Fernández-Navarro, C. Hervás-Martínez, J. Sánchez-Monedero y P.A. Gutiérrez. MELM-GRBF: A modified version of the extreme learning machine for generalized radial basis function neural networks. *Neurocomputing*, 74(16):2502–2510, 2011.

BIBLIOGRAFÍA

- [57] J.L. Fleiss, J. Cohen y B.S. Everitt. Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin*, 72(5):323–327, 1969.
- [58] D.B. Fogel, A.J. Owens y M.J. Wals. *Artificial Intelligence through Simulated Evolution*. New York Wiley, 1966.
- [59] C.M. Fonseca y P.J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. En *Proceedings of the Fifth International Conference on Genetic Algorithms*, págs. 416–423, 1993.
- [60] E. Frank y M. Hall. A simple approach to ordinal classification. En *Proceedings of the 12th European Conference on Machine Learning, EMCL '01*, págs. 145–156, 2001.
- [61] J.E. Friedlsend y S. Singh. Pareto evolutionary neural networks. *IEEE Transactions Neural Networks*, 16(2):338–354, 2005.
- [62] D. Garcia, A.J. Ramos, V. Sanchis y S. Marin. Predicting mycotoxins in foods: a review. *Food Microbiology*, 26:757–769, 2009.
- [63] F. Glover y C. McMillan. The general employee scheduling problem: an integration of MS and AI. *Computers & Operations Research*, 13(5):563–573, 1986.
- [64] C. Grosan, A. Abraham y H. Ishibuchi. *Hybrid Evolutionary Algorithms*. Springer, 2007.
- [65] P.A. Gutiérrez, M. Pérez-Ortiz, F. Fernández-Navarro, J. Sánchez-Monedero y C. Hervás-Martínez. An experimental study of different ordinal regression methods and measures. En *7th International Conference on Hybrid Artificial Intelligence Systems (HAIS2012), Lecture Notes in Computer Science*, volumen 7209, págs. 296–307, 2012.
- [66] P.A. Gutiérrez y A. Ruiz. *Proyecto Fin de Carrera: Diseño de modelos de redes neuronales de unidades híbridas utilizando algoritmos de Computación Evolutiva*. Universidad de Córdoba, 2006.
- [67] T.M. Hamdani, J.M. Won, A.M. Alimi y F. Karray. Multi-objective feature selection with NSGA II. En *Adaptive and Natural Computing Algorithms*, volumen 4431 de *Lecture Notes in Computer Science*, págs. 240–247. 2007.
- [68] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2ª edición, 1998.
- [69] R. Herbrich, T. Graepel y K. Obermayer. *Large margin rank boundaries for ordinal regression, Advances in Large Margin Classifiers*. MIT Press, 2000.

- [70] J. Horn, J. Nafpliotis y D.E. Golberg. A niched Pareto genetic algorithm for multiobjective optimization. En *Proceedings of the 1994 Congress on Evolutionary Computation*, volumen 1, págs. 82–87, 1994.
- [71] K. Hornik, M. Stinchcombe y H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- [72] K. Hornik, M. Stinchcombe y H. White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3(5):551–560, 1990.
- [73] C.R. Houck, J.A. Joines, M.G. Kay y J.R. Wilson. Empirical investigation of the benefits of partial lamarckianism. *Evolutionary Computation*, 5:31–60, 1997.
- [74] C. Hwang y K. Yoon. *Multiple attribute decision making: Methods and application*. Springer-Verlag, 1981.
- [75] C. Igel y M. Hüsken. Empirical evaluation of the improved rprop learning algorithms. *Neurocomputing*, 50(6):105–123, 2003.
- [76] C. Igel y M. Hüsken. Improving the rprop learning algorithm. *Proc. Proceedings of the Second International ICSC Symposium on Neural Computation (NC 2000)*, ICSC Academic Press, págs. 115–121, 2000.
- [77] Y. Jin y B. Sendhoff. Pareto multiobjective machine learning: An overview and case studies. *IEEE Transaction on Systems, Man and Cybernetics, Part. C: Applications and reviews*, 38(3):397–415, 2008.
- [78] J.A. Joines y M.G. Kay. Utilizing hybrid genetic algorithms. En *Evolutionary Optimization*, volumen 48 de *International Series in Operations Research & Management Science*, págs. 199–228. 2003.
- [79] P.S. Kamath y W.R. Kim. The Model for End-stage Liver Disease (MELD). *Hepatology*, 45(3):797–805, 2007.
- [80] M.G. Kendall. *Rank Correlation Methods*. New York: Hafner Press, 1962.
- [81] J. Kennedy y R. Eberhart. Particle swarm optimization. En *Proceeding of the IEEE International Conference on Neural Networks*, págs. 1942–1948, 1995.
- [82] K.J. Kim y H. Ahn. A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach. *Computers and Operations Research*, 39(8):1800–1811, 2012.
- [83] J.D. Knowles y D.W. Corne. The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization. En *Proceedings of the 1999 Congress on Evolutionary Computation*, págs. 98–105, 1999.

BIBLIOGRAFÍA

- [84] K. Koutsoumanis, P. Taoukis y G. Nychas. Development of a safety monitoring and assurance system for chilled food products. *International Journal of Food Microbiology*, 100(1-3):253–260, 2005.
- [85] N. Lachiche y P.A. Flach. Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. En *Proceedings of the 20th International Conference on Machine Learning , ICML'03*, págs. 416–423, 2003.
- [86] A. Ligeza. *Logical Foundations for Rule-Based Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [87] H.T. Lin y L. Li. Reduction from cost-sensitive ordinal ranking to weighted binary classification. *Neural Computation*, 24(5):1329–1367, 2012.
- [88] R.P. Lippmann. Pattern classification using neural networks. *IEEE Communications Magazine*, 27(11):47–64, 1989.
- [89] T. Löfström, U. Johansson y H. Boström. Ensemble member selection using multi-objective optimization. En *2009 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009*, págs. 245–251, 2009.
- [90] J. MacQueen. Some methods for classification and analysis of multivariate observations. En *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, págs. 281–297, 1967.
- [91] A.C. Martínez. *Modelos de Regresión Basados en Redes Neuronales de Unidades Producto Diseñadas y Entrenadas Mediante Algoritmos de Optimización Híbrida. Aplicaciones*. Tesis doctoral, Universidad de Granada, 2005.
- [92] P. McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society, Series B (Methodological)*, 42(2):109–142, 1980.
- [93] W.S. McCulloch y W.H. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
- [94] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Informe Técnico 826, Caltech Concurrent Computation, Catech, Pasadena, California, 1989.
- [95] P. Moscato y C. Cotta. *A Gentle Introduction To Memetic Algorithms, Handbook On Metaheuristics*, págs. 105–144. Kluwer Academics, 2003.
- [96] V. Pareto. *Cours D'Economie Politique*. Droz, Genève, 1896.
- [97] J. Park y I.W. Sandberg. Approximation and radial-basis-function networks. *Neural Computation*, 5(2):305–316, 1993.

- [98] M. Pérez-Ortiz, P.A. Gutiérrez, C. García-Alonso, L.S. Carulla, J.A. Salinas Pérez y C. Hervás-Martínez. Ordinal classification of depression spatial hot-spots of prevalence. En *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA 2011)*, págs. 1170–1175, 2011.
- [99] M. Pérez-Ortiz, M. Cruz-Ramírez, J.C. Fernández-Caballero y C. Hervás-Martínez. Hybrid multi-objective machine learning classification in liver transplantation. En *Proceedings of the 7th international conference on Hybrid Artificial Intelligent Systems - Volume Part I, HAIS12*, págs. 397–408, 2012.
- [100] M. Pérez-Ortiz, P.A. Gutiérrez, C. Hervás-Martínez, J. Briceño y M. de la Mata. An ensemble approach for ordinal threshold models applied to liver transplantation. En *Proceedings of the the 2012 International Joint Conference on Neural Networks (IJCNN)*, págs. 2795–2802, 2012.
- [101] M. Pérez-Ortiz, M. Cruz-Ramírez, M.D. Ayllón-Terán, N. Heaton, R. Ciria y C. Hervás-Martínez. An organ allocation system for liver transplantation based on ordinal regression. *Applied Soft Computing*, enviado, 2013.
- [102] J.F. Pinto da Costa y J.S. Cardoso. Classification of ordinal data using neural networks. En *Proceedings of the 16th European conference on Machine Learning, ECML'05*, págs. 690–697, 2005.
- [103] J.F. Pinto da Costa, H. Alonso y J.S. Cardoso. The unimodal model for the classification of ordinal data. *Neural Networks*, 21:78–91, 2008.
- [104] F. Provost y T. Fawcett. Analysis and visualization of the classifier performance: comparison under imprecise class and cost distribution. En *Proceedings of the Third International Conference on Knowledge Discovery (KDD97) and Data Mining*, págs. 43–48, 1997.
- [105] F. Provost y T. Fawcett. Robust classification system for imprecise environments. En *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, págs. 706–713, 1998.
- [106] G.T. Pulido y C.A. Coello. The micro genetic algorithm 2: Towards online adaptation in evolutionary multiobjective optimization. En *Evolutionary Multicriterion Optimization: The Second International Conference (EMO2003)*, págs. 252–266, 2003.
- [107] A. Rana, M.A. Hardy, K.J. Halazun, D.C. Woodland, L.E. Ratner, B. Samstein, J.V. Guarrera, R.S. Brown y J.C. Emond. Survival outcomes following liver transplantation (SOFT) score: a novel method to predict patient survival following liver transplantation. *American Journal of Transplantation*, 8(12):2537–46, 2008.

BIBLIOGRAFÍA

- [108] G.J.E. Rawlins. *Foundations of Genetic Algorithms*. Morgan Kaufman, 1991.
- [109] Registro Español de Trasplante Hepático. Sociedad española de trasplante hepático. Organización Nacional de Trasplantes, (1984–2009).
- [110] M. Riedmiller y H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. En *Proceedings of the 1993 IEEE International Conference on Neural Networks*, págs. 586–591, 1993.
- [111] H.R. Rosen, M. Prieto, T. Casanovas-Taltavull, V. Cuervas-Mons, O. Guckelberger, P. Muiesan, R.W. Strong, W.O. Bechstein, J. O’grady, A. Zaman, B. Chan, J. Berenguer, R. Williams y N. Neuhaus P. Heaton. Validation and refinement of survival models for liver retransplantation. *Hepatology*, 38(2):460–469, 2003.
- [112] S. Roth, A. Gepperth y C. Igel. Multi-objective neural network optimization for visual object detection. *Studies in Computational Intelligence*, 16:629–655, 2006.
- [113] O. Rudovic, V. Pavlovic y M. Pantic. Multi-output laplacian dynamic ordinal regression for facial expression recognition and intensity estimation. En *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2012)*, págs. 2634–2641, 2012.
- [114] K. Saito y R. Nakano. Numeric law discovery using neural networks. En *Proceedings of the IVth International Conference on Neural Information Processing (ICONIP97)*, págs. 843–846, 1997.
- [115] M. Schmitt. On the complexity of computing and learning with multiplicative neural networks. *Neural Computation*, 14:241–301, 2002.
- [116] J.E. Smith. Coevolving memetic algorithms: A review and progress report. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(1):6–17, 2007.
- [117] M. Sokolova y G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45:427–437, 2009.
- [118] J.M. Soriano, G. Font, J.C. Moltó y J. Mañes. Enterotoxigenic staphylococci and their toxins in restaurant foods. *Trends of Food Science and Technology*, 13:60–67, 2002.
- [119] C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:72–101, 1904.
- [120] N. Srivinas y K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Journal of Evolutionary Computation*, 2(3):221–248, 1994.

- [121] R. Storn y K. Price. Differential Evolution. A fast and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11: 341–359, 1997.
- [122] B.Y. Sun, J. Li, D.D. Wu, X.M. Zhang y W.B. Li. Kernel discriminant learning for ordinal regression. *IEEE Transactions on Knowledge and Data Engineering*, 22 (6):906–910, 2010.
- [123] S. Theodoridis y K. Koutroumbas. *Pattern Recognition*. Elsevier, Academic Press, 3ª edición, 2006.
- [124] A. Valero, F. Pérez-Rodríguez, E. Carrasco, J.M. Fuentes-Alventosa, R.M. García-Gimeno y G. Zurera. Modelling the growth boundaries of staphylococcus aureus: Effect of temperature, ph and water activity. *International Journal Food Microbiology*, 133:186–194, 2009.
- [125] V.N. Vapnik y A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.
- [126] J. Verwaeren, W. Waegeman y B. De Baets. Learning partial ordinal class memberships with kernel-based proportional odds models. *Computational Statistics & Data Analysis*, 56(4):928–942, 2012.
- [127] W. Waegeman, B. De Baets y L. Boullart. ROC analysis in ordinal regression learning. *Pattern Recognition Letters*, 29(1):1–9, 2008.
- [128] S. Wiegand y C. Igel. Evolutionary multi-objective optimization of neural networks for face detection. *International Journal of Computation Intelligence and Applications*, 4(3):237–253, 2004.
- [129] I.H. Witten y E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Sys. Morgan Kaufmann, 2ª edición, 2005.
- [130] J.W. Yoon, S.J. Roberts, M. Dyson y J.Q. Gan. Bayesian inference for an adaptive ordered probit model: An application to brain computer interfacing. *Neural Networks*, 24(7):726–734, 2011.
- [131] L. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.
- [132] L.L. Zaika, E. Moulden, L. Weimer, J.G. Phillips y R.L. Buchanan. Model for the combined effects of temperature, initial ph, sodium chloride and sodium nitrite concentrations on anaerobic growth of shigella flexneri. *International Journal of Food Microbiology*, 23:345–358, 1994.

BIBLIOGRAFÍA

- [133] G.P. Zhang. Neural networks for classification: A survey. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 30(4):451–462, 2000.
- [134] Q Zhang y H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.
- [135] E. Zitzler y L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions Neural Networks*, 3(4):414–417, 1999.
- [136] E. Zitzler, M. Laumanns y L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. En *Proceedings of the International Conference on Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, págs. 95–100, 2001.

A

PUBLICACIONES ASOCIADAS CON LA PRESENTE TESIS

En este Anexo se van a enumerar las publicaciones realizadas durante la elaboración de la presente Tesis.

- La Sección [A.1](#) muestra la información relativa a un trabajo publicado como capítulo de libro.
- En la Sección [A.2](#) se presentan los trabajos enviados y publicados en revistas científicas.
- En la Sección [A.3](#) se muestran las aportaciones realizadas en congresos internacionales.
- Finalmente, la Sección [A.4](#) enumera las comunicaciones presentadas en congresos nacionales.

A.1 CAPÍTULOS DE LIBRO

- **M. Cruz-Ramírez** y C. Hervás-Martínez. *New trends on intelligent systems and soft computing*, capítulo *Memetic multi-objective algorithm based on differential evolution to design artificial neural networks: applications*, págs. 13–28. Universidad de Granada, 2012.

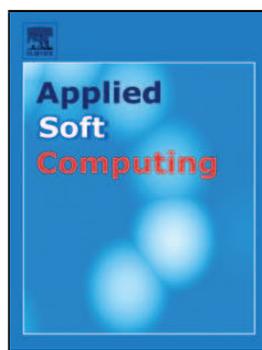
A.2 ARTÍCULOS EN REVISTA

TÍTULO:

An organ allocation system for liver transplantation based on ordinal regression.

AUTORES:

M. Pérez-Ortiz, **M. Cruz-Ramírez**, M.D. Ayllón-Terán, N. Heaton, R. Ciria y C. Hervás-Martínez.



Applied Soft Computing, Enviado (2013).

RANKING:

- A. Índice de Impacto (JCR 2011): 2,612.
- B. Área de conocimiento: *Computer Science, Artificial Intelligence*. Ranking 13/111. Primer Cuartil.
- C. Área de conocimiento: *Computer Science, Interdisciplinary Applications*. Ranking 16/99. Primer Cuartil.

© Elsevier Publishing Group. Reprinted with permission.

An organ allocation system for liver transplantation based on ordinal regression

M. Pérez-Ortiz^{a,*}, M. Cruz-Ramírez^a, M. D. Ayllón-Terán^b, N. Heaton^c, R. Ciria^b, C. Hervás-Martínez^a

^a*Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain*

^b*Liver Transplantation Unit, Reina Sofía Hospital, Córdoba, Spain*

^c*Liver Transplantation Unit, King's College, London, United Kingdom*

Abstract

Liver transplantation is nowadays a widely-accepted treatment for patients who present a terminal liver disease. Nevertheless, transplantation is greatly hampered by the un-availability of suitable liver donors; several methods have been developed and applied to find a better system to prioritize recipients on the waiting list, although most of them only consider donor or recipient characteristics (but not both). This paper proposes a novel donor-recipient liver allocation system constructed to predict graft survival after transplantation by means of a dataset comprised of donor-recipient pairs from different centres (seven Spanish and one UK hospitals). The best model obtained is used in conjunction with the Model for End-stage Liver Disease score (MELD), one of the current assignation methodology most used globally. This problem is assessed using the ordinal regression learning paradigm due to the natural ordering in the classes of the problem, via a cascade binary decomposition methodology and the Support Vector Machine methodology. The methodology proposed has shown competitiveness in all the metrics selected, when compared to other machine learning techniques and efficiently complements the MELD score based on the principles of efficiency and equity. Finally, a simulation of the proposal is included, in order to visualize its performance in realistic situations. This simulation has shown that there are some determining factors in the characterization of the survival time after transplantation (concerning both donors and recipients) and that the

*Corresponding author at: Department of Computer Science and Numerical Analysis, University of Córdoba, Rabanales Campus, Albert Einstein Building 3rd Floor, 14071 Córdoba, Spain. Tel.: +34 957 218 349; Fax: +34 957 218 630.

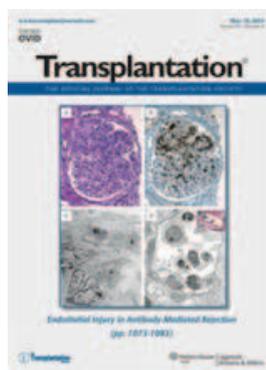
E-mail addresses: i82perom@uco.es, mcruz@uco.es, lolesat83@hotmail.com, rubenciria@hotmail.com, nigel.heaton@nhs.net, javibriceno@hotmail.com, mdelamatagarcia@gmail.com, chervas@uco.es.

TÍTULO:

Application of artificial intelligence as an innovative donor-recipient matching model for liver transplantation: results from a multicenter Spanish study.

AUTORES:

J. Briceño, **M. Cruz-Ramírez**, M. Prieto, M. Navasa, J. Ortiz de Urbina, R. Orti, M.A. Gómez-Bravo, A. Otero, E. Varo, S. Tomé, G. Clemente, R. Bañares, R. Bárcena, V. Cuervas-Mons, G. Solórzano, C. Vinaixa, Á. Rubín, J. Colmenero, A. Valdivieso, R. Ciria, C. Hervás-Martínez y M. de la Mata.



Transplantation, Enviado (2013).

RANKING:

- A. Índice de Impacto (JCR 2011): 4,003.
- B. Área de conocimiento: *Immunology*. Ranking 36/139. Segundo Cuartil.
- C. Área de conocimiento: *Surgery*. Ranking 11/199. Primer Cuartil.
- D. Área de conocimiento: *Transplantation*. Ranking 4/24. Primer Cuartil.

© Lippincott Williams & Wilkins. Reprinted with permission.

TITLE PAGE**APPLICATION OF ARTIFICIAL INTELLIGENCE AS AN INNOVATIVE DONOR-RECIPIENT
MATCHING MODEL FOR LIVER TRANSPLANTATION: RESULTS FROM A MULTICENTER
SPANISH STUDY**

AUTHORS: Javier Briceño¹, Manuel Cruz-Ramírez², Martín Prieto³, Miguel Navasa⁴, Jorge Ortiz de Urbina⁵, Rafael Orti¹, Miguel-Angel Gómez-Bravo⁶, Alejandra Otero⁷, Evaristo Varo⁸, Santiago Tomé⁸, Gerardo Clemente⁹, Rafael Bañares⁹, Rafael Bárcena¹⁰, Valentin Cuervas-Mons¹¹, Guillermo Solórzano¹², Carmen Vinaixa³, Ángel Rubín³, Jordi Colmenero⁴, Andres Valdivieso⁵, Ruben Ciria¹, Cesar Hervás-Martínez², Manuel de la Mata¹.

KEY WORDS: artificial intelligence / allocation / survival / prediction / optimization

WORD COUNT: Title: 120 characters / Abstract: 248 / Document: 3075

NUMBER OF TABLES / FIGURES: Tables: 3 (2 B&W / 1 Color) / Figures: 2 (1 B&W / 1 Color)

ADDRESS FOR CORRESPONDENCE:

- Javier Briceño.
- Liver Transplantation Unit. University Hospital Reina Sofía. Córdoba, SPAIN.
- Phone: 0034957010439 / Fax: 0034957010949
- Email: javibriceno@hotmail.com

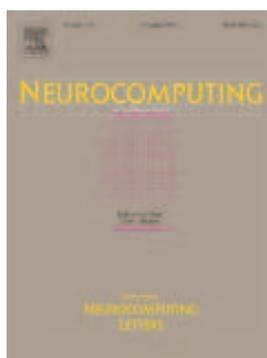
PUBLICACIONES ASOCIADAS CON LA PRESENTE TESIS

TÍTULO:

Metrics to guide a multi-objective evolutionary algorithm for ordinal classification.

AUTORES:

M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero y P.A. Gutiérrez.



Neurocomputing, Aceptado (2013).

RANKING:

- A. Índice de Impacto (JCR 2011): 1,580.
- B. Área de conocimiento: *Computer Science, Artificial Intelligence*. Ranking 39/111. Segundo Cuartil.

© Elsevier Publishing Group. Reprinted with permission.

Metrics to Guide a Multi-Objective Evolutionary Algorithm for Ordinal Classification

M. Cruz-Ramírez*, C. Hervás-Martínez, J. Sánchez-Monedero, P.A. Gutiérrez

Department of Computer Science and Numerical Analysis. University of Córdoba, Spain

Abstract

Ordinal classification or ordinal regression are classification problems in which the labels have an ordered arrangement between them. Due to this order, alternative performance evaluation metrics are needed to be used in order to consider the magnitude of errors. This paper presents an study of the use of a multi-objective optimization approach in the context of ordinal classification. We contribute a study of ordinal classification performance metrics, and propose a new performance metric, the Maximum Mean Absolute Error (*MMAE*). *MMAE* considers per-class distribution of patterns and the magnitude of the errors, both issues being crucial for ordinal regression problems. In addition we empirically show that some of the performance metrics are competitive objectives, which justifies the use of multi-objective optimization strategies. In our case, a multi-objective evolutionary algorithm optimizes a artificial neural network ordinal model with different pairs of metrics combinations, and we conclude that the pair of the Mean Absolute Error (*MAE*) and the proposed *MMAE* is the most favorable. A study of the relationship between the metrics of this proposal is performed, and the graphical representation in the 2 dimensional space where the search of the evolutionary algorithm takes place is analyzed. The results obtained show a good classification performance, opening new lines of research in the evaluation and model selection of ordinal classifiers.

Keywords: mean absolute error, multi-objective evolutionary algorithm, ordinal measures, ordinal classification, ordinal regression, proportional odds model

1. Introduction

Ordinal classification or ordinal regression is a supervised learning problem of predicting categories that have an ordered arrangement. Although classification and regression metric problems have been thoroughly investigated in the literature, the ordinal regression problems have not received as much attention as nominal (binary or multiclass) classification. For example, people can be classified by considering whether they are high, medium, or low on some attribute or in a set of categories varying from strong agreement to strong disagreement with respect to some attitude item.

Hodge and Treiman [1], to analyze social class identification, scored responses as follows: “Respondents identifying with the lower, working, middle, upper middle, and upper class were assigned the scores 1, 2, 3, 4, and 5, respectively”. Though sequential numbers may be assigned to such categories, the numbers assigned serve only to identify the ordering of the categories. In contrast to regression metric problems, these ranks are finite types and the metric distances between the ranks are not defined, in general; in contrast to classification problems, these ranks are also different from the labels of multiple classes due to the existence of the ordering information [2].

In the previous example, it is straightforward to think that predicting class *lower* when the real class is *upper middle* should be considered as a more severe error than the one associated to a *working* prediction. Thereby, ordinal classification problems should be evaluated with specific metrics. On a first consideration, various measures of ordinal association and product-moment correlation and regression seem to rely on very different

*Corresponding author at: Department of Computer Science and Numerical Analysis, University of Córdoba, Rabanales Campus, Albert Einstein Building 3rd Floor, 14071 Córdoba, Spain. Tel.: +34 957 218 349; Fax: +34 957 218 630.

E-mail addresses: {mcruz, chervas, jsanchezm, pagutierrez}@uco.es

**This paper is a significant extension of the work “A Preliminary Study of Ordinal Metrics to Guide a Multi-Objective Evolutionary Algorithm” appearing in the 11th International Conference on Intelligent Systems Design and Applications (ISDA2011).

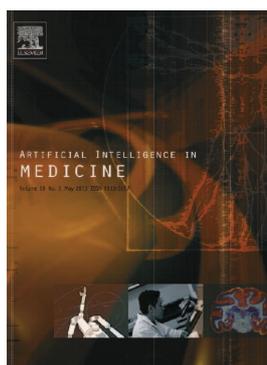
PUBLICACIONES ASOCIADAS CON LA PRESENTE TESIS

TÍTULO:

Predicting patient survival after liver transplantaton using evolutionary multi-objective artificial neural networks.

AUTORES:

M. Cruz-Ramírez, C. Hervás-Martínez, J.C. Fernández, J. Briceño y M. de la Mata.



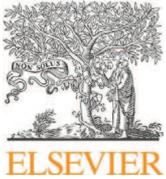
Artificial Intelligence in Medicine, Volumen 58(1), 37-49 (2013).

RANKING:

- A. Índice de Impacto (JCR 2011): 1,345.
- B. Área de conocimiento: *Computer Science, Artificial Intelligence*. Ranking 48/111. Segundo Cuartil.
- C. Área de conocimiento: *Engineering, Biomedical*. Ranking 48/72. Tercer Cuartil.
- D. Área de conocimiento: *Medical Informatics*. Ranking 15/23. Tercer Cuartil.

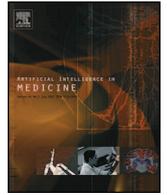
DOI: [10.1016/j.artmed.2013.02.004](https://doi.org/10.1016/j.artmed.2013.02.004)

© Elsevier Publishing Group. Reprinted with permission.



Contents lists available at SciVerse ScienceDirect

Artificial Intelligence in Medicine

journal homepage: www.elsevier.com/locate/aiim

Predicting patient survival after liver transplantation using evolutionary multi-objective artificial neural networks



Manuel Cruz-Ramírez^{a,*}, César Hervás-Martínez^a, Juan Carlos Fernández^a,
Javier Briceño^{b,c}, Manuel de la Mata^{b,c}

^a Department of Computer Science and Numerical Analysis, University of Córdoba, Rabanales Campus, Albert Einstein Building 3rd Floor, 14071 Córdoba, Spain

^b Liver Transplantation Unit, University Hospital Reina Sofía, Avenue Menéndez Pidal, 14004 Córdoba, Spain

^c CIBERehd, Spain

ARTICLE INFO

Article history:

Received 14 September 2011

Received in revised form 4 February 2013

Accepted 5 February 2013

Keywords:

Making decisions rule-based
Multi-objective evolutionary algorithm
Radial basis function neural networks
Liver transplantation
Organ allocations

ABSTRACT

Objective: The optimal allocation of organs in liver transplantation is a problem that can be resolved using machine-learning techniques. Classical methods of allocation included the assignment of an organ to the first patient on the waiting list without taking into account the characteristics of the donor and/or recipient. In this study, characteristics of the donor, recipient and transplant organ were used to determine graft survival. We utilised a dataset of liver transplants collected by eleven Spanish hospitals that provides data on the survival of patients three months after their operations.

Methods and material: To address the problem of organ allocation, the memetic Pareto evolutionary non-dominated sorting genetic algorithm 2 (MPENSGA2 algorithm), a multi-objective evolutionary algorithm, was used to train radial basis function neural networks, where accuracy was the measure used to evaluate model performance, along with the minimum sensitivity measurement. The neural network models obtained from the Pareto fronts were used to develop a rule-based system. This system will help medical experts allocate organs.

Results: The models obtained with the MPENSGA2 algorithm generally yielded competitive results for all performance metrics considered in this work, namely the correct classification rate (*C*), minimum sensitivity (*MS*), area under the receiver operating characteristic curve (*AUC*), root mean squared error (*RMSE*) and Cohen's kappa (*Kappa*). In general, the multi-objective evolutionary algorithm demonstrated a better performance than the mono-objective algorithm, especially with regard to the *MS* extreme of the Pareto front, which yielded the best values of *MS* (48.98) and *AUC* (0.5659).

The rule-based system efficiently complements the current allocation system (model for end-stage liver disease, MELD) based on the principles of efficiency and equity. This complementary effect occurred in 55% of the cases used in the simulation. The proposed rule-based system minimises the prediction probability error produced by two sets of models (one of them formed by models guided by one of the objectives (entropy) and the other composed of models guided by the other objective (*MS*)), such that it maximises the probability of success in liver transplants, with success based on graft survival three months post-transplant.

Conclusion: The proposed rule-based system is objective, because it does not involve medical experts (the expert's decision may be biased by several factors, such as his/her state of mind or familiarity with the patient). This system is a useful tool that aids medical experts in the allocation of organs; however, the final allocation decision must be made by an expert.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Liver transplantation is an accepted treatment for patients with end-stage chronic liver disease, but it is strongly limited due to the

limited availability of suitable liver donors. The imbalance between supply and demand unfortunately results in many waiting-list deaths. Several efforts have been made to expand the donor pool and to better prioritise recipients on waiting lists. Some examples of these efforts are the use of extended criteria donors (donors with extreme values of age, days in the intensive care unit (ICU), inotrope usage, body mass index (BMI) and cold ischemia time) and the adoption of the model for end-stage liver disease (MELD) score [1].

* Corresponding author. Tel.: +34 957 218 349; fax: +34 957 218 630.
E-mail address: mcruz@uco.es (M. Cruz-Ramírez).

PUBLICACIONES ASOCIADAS CON LA PRESENTE TESIS

TÍTULO:

Memetic pareto differential evolutionary neural network used to solve an unbalanced liver transplantation problem.

AUTORES:

M. Cruz-Ramírez, C. Hervás-Martínez, P.A. Gutiérrez, M. Pérez-Ortiz, J. Briceño y M. de la Mata.



Soft Computing, Volumen 17(29), 275-284 (2012).

RANKING:

- A. Índice de Impacto (JCR 2011): 1,880.
- B. Área de conocimiento: *Computer Science, Artificial Intelligence*. Ranking 30/111. Segundo Cuartil.
- C. Área de conocimiento: *Computer Science, Interdisciplinary Applications*. Ranking 24/99. Primer Cuartil.

DOI: [10.1007/s00500-012-0892-7](https://doi.org/10.1007/s00500-012-0892-7)

© Springer Publishing Group. Reprinted with permission.

Memetic Pareto differential evolutionary neural network used to solve an unbalanced liver transplantation problem

M. Cruz-Ramírez · C. Hervás-Martínez ·
P. A. Gutiérrez · M. Pérez-Ortiz · J. Briceño ·
M. de la Mata

Published online: 24 July 2012
© Springer-Verlag 2012

Abstract Donor–recipient matching constitutes a complex scenario difficult to model. The risk of subjectivity and the likelihood of falling into error must not be underestimated. Computational tools for the decision-making process in liver transplantation can be useful, despite the inherent complexity involved. Therefore, a multi-objective evolutionary algorithm and various techniques to select individuals from the Pareto front are used in this paper to obtain artificial neural network models to aid decision making. Moreover, a combination of two pre-processing methods has been applied to the dataset to offset the existing imbalance. One of them is a resampling method and the other is an outlier deletion method. The best model

obtained with these procedures (with $AUC = 0.66$) give medical experts a probability of graft survival at 3 months after the operation. This probability can help medical experts to achieve the best possible decision without forgetting the principles of fairness, efficiency and equity.

Keywords Artificial neural networks · Liver transplantation · Multi-objective evolutionary algorithm · Organ allocations · Pre-processing

1 Introduction

Liver transplantation is an accepted treatment for patients suffering end-stage chronic liver disease. Numerous donor and recipient risk factors interact and influence the probability of survival 3 months after liver transplantation. It is critical to balance waiting-list mortality and post-transplant mortality. The objective is to devise a ranking system that predicts 3-month recipient survival following liver transplantation to complement the model for end-stage liver disease score (MELD) (Wiesner et al. 2003) in order to predict waiting-list mortality.

Most current organ allocation systems are based on the principle that the sickest patients should be treated first. The models thus developed to estimate the risk of death consider the underlying disease and urgency of the recipient, assuming that all donor livers imply the same risk of failure. This, however, is not the case: it has been shown in recent years that the risk of graft failure, and even patient death, after transplantation, differs from one recipient to another. While some patients may “tolerate” and overcome the initially poor functioning of a compromised donor organ, others may not have the same tolerance. Increasing awareness of the diversity in donor organ quality has

This paper is a significant extension of the work “Memetic Pareto differential evolutionary neural network for donor-recipient matching in liver transplantation” appearing in the International Work-Conference on Artificial Neural Networks 2011 (IWANN’11).

M. Cruz-Ramírez (✉) · C. Hervás-Martínez ·
P. A. Gutiérrez · M. Pérez-Ortiz
Department of Computer Science and Numerical Analysis,
University of Córdoba, Córdoba, Spain
e-mail: mcruz@uco.es

C. Hervás-Martínez
e-mail: chervas@uco.es

P. A. Gutiérrez
e-mail: pagutierrez@uco.es

M. Pérez-Ortiz
e-mail: i82perom@uco.es

J. Briceño · M. de la Mata
Liver Transplantation Unit, Hospital Reina Sofía, CIBERehd,
Córdoba, Spain
e-mail: javibriceno@hotmail.com

M. de la Mata
e-mail: mdelamatagarcia@gmail.com

PUBLICACIONES ASOCIADAS CON LA PRESENTE TESIS

TÍTULO:

Multi-objective evolutionary algorithm for donor-recipient decision system in liver transplants.

AUTORES:

M. Cruz-Ramírez, C. Hervás-Martínez, J.C. Fernandez-Caballero, J. Briceño y M. de la Mata.



European Journal of Operational Research, Volumen 222(2), 317–327 (2012).

RANKING:

- A. Índice de Impacto (JCR 2011): 1,815.
- B. Área de conocimiento: *Operations Research & Management Science*. Ranking 6/77. Primer Cuartil.

DOI: [10.1016/j.ejor.2012.05.013](https://doi.org/10.1016/j.ejor.2012.05.013)

© Elsevier Publishing Group. Reprinted with permission.



Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Decision Support

Multi-objective evolutionary algorithm for donor–recipient decision system in liver transplants

Manuel Cruz-Ramírez^{a,*}, César Hervás-Martínez^a, Juan Carlos Fernández^a, Javier Briceño^b, Manuel de la Mata^b^a Department of Computer Science and Numerical Analysis, University of Córdoba, Spain^b Liver Transplantation Unit, Hospital Reina Sofía, CIBERehd, Córdoba, Spain

ARTICLE INFO

Article history:

Received 20 October 2011

Accepted 4 May 2012

Available online 15 May 2012

Keywords:

Artificial neural networks

Generalised radial basis function

Liver transplantation

Multi-objective evolutionary algorithms

Organ allocations

Rule-based system

ABSTRACT

This paper reports on a decision support system for assigning a liver from a donor to a recipient on a waiting-list that maximises the probability of belonging to the *survival graft* class after a year of transplant and/or minimises the probability of belonging to the *non-survival graft* class in a two objective framework. This is done with two models of neural networks for classification obtained from the Pareto front built by a multi-objective evolutionary algorithm – called MPENSGA2. This type of neural network is a new model of the generalised radial basis functions for obtaining optimal values in C (Correctly Classified Rate) and MS (Minimum Sensitivity) in the classifier, and is compared to other competitive classifiers. The decision support system has been proposed using, as simply as possible, those models which lead to making the correct decision about receptor choice based on efficient and impartial criteria.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Today, hepatic transplantation is a well-accepted treatment for patients with terminal liver disease, but transplantation is strongly limited by the availability of suitable liver donors. The imbalance between supply and demand unfortunately results in many waiting-list deaths. Several efforts have been made to expand the donor pool and to prioritise recipients on the waiting-list better. The use of extended criteria donors (donors with extreme values of age, days in the intensive care unit, existence of inotropes, BMI and cold ischaemia time) as well as the adoption of the Model for End-stage Liver Disease (MELD) score [1] for prioritisation are good examples of these attempts worldwide.

In recent years, more relaxed criteria have been used for donors, with an accompanying increased risk in recipient and/or graft losses in comparison to livers from non-extended criteria donors [2]. The MELD score model (based on the sickest-first principle) is the cornerstone of the current allocation policy and has been widely validated [3]. These methods have been developed to estimate the risk of death, taking into consideration the underlying disease and urgency of the receiving patient and assuming that all donor livers carry the same risk of failure, whereas, in general,

this assumption is not true. However, the combination of multiple marginal factors can mean the loss of the graft [4], so these risks should be carefully analysed. Thus, these methods are poor predictors of mortality following transplants, because they have not jointly considered the characteristics of the donor, recipient and transplant. Based on this concept, Feng [5] has proposed a donor risk index (DRI), aimed at establishing the quantitative risk associated with the sole use of combinations of donor characteristics.

Predicting the first 1-year post-transplant survival graft could potentially play a critical role in understanding and improving the matching procedure between the recipient, the donor and the transplant. Our objective is to devise a liver allocation system, based on donor, recipient and transplant matching, that predicts recipient survival 1 year following liver transplantation. There are numerous motivations for developing this system: (1) current selection/allocation systems are based on the risk of waiting-list patient deaths and do not recognise distinctions in donor organ quality; (2) efforts to increase the number of organ donations are likely to result in a relatively high proportion of extended criteria donors; (3) matching donors and recipients may offer the prospect of predicting outcome at the time when a specific donor liver is allocated to a specific recipient; (4) differences in local acceptance rates and policies may be diminished; and (5) overall outcome and efficacy may improve.

This liver allocation system will be developed using Machine Learning and Soft Computing methods, specifically Evolutionary Artificial Neural Networks (ANNs) [6,7]. ANNs have been an object

* Corresponding author. Address: Department of Computer Science and Numerical Analysis, University of Córdoba, Rabanales Campus, Albert Einstein Building 3rd Floor, 14071 Córdoba, Spain. Tel.: +34 957 218 349; fax: +34 957 218 630.

E-mail address: mcruz@uco.es (M. Cruz-Ramírez).

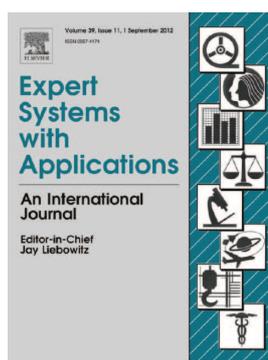
PUBLICACIONES ASOCIADAS CON LA PRESENTE TESIS

TÍTULO:

A multiobjective neural network based method for cover crop identification from remote sensed data.

AUTORES:

M. Cruz-Ramírez, C. Hervás-Martínez, M. Jurado-Expósito y F. López-Granados.



Expert Systems with Applications, Volumen 39(11), 10038-10048 (2012).

RANKING:

- A. Índice de Impacto (JCR 2011): 2,203.
- B. Área de conocimiento: *Computer Science, Artificial Intelligence*. Ranking 22/111. Primer Cuartil.
- C. Área de conocimiento: *Engineering, Electrical & Electronic*. Ranking 41/245. Primer Cuartil.
- D. Área de conocimiento: *Operations Research & Management Science*. Ranking 5/77. Primer Cuartil.

DOI: [10.1016/j.eswa.2012.02.046](https://doi.org/10.1016/j.eswa.2012.02.046)

© Elsevier Publishing Group. Reprinted with permission.



Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A multi-objective neural network based method for cover crop identification from remote sensed data

M. Cruz-Ramírez^{a,*}, C. Hervás-Martínez^a, M. Jurado-Expósito^b, F. López-Granados^b

^a Department of Computer Science and Numerical Analysis, University of Córdoba, Rabanales Campus, Albert Einstein Building 3rd Floor, 14071 Córdoba, Spain

^b Institute for Sustainable Agriculture, IAS-CSIC, Apdo. 4084, 14080 Córdoba, Spain

ARTICLE INFO

Keywords:

Cruciferous
Grass weed
Multi-classification
Multi-objective evolutionary
Multispectral
Neural networks
Olive orchard
Remote sensing data

ABSTRACT

One of the objectives of conservation agriculture to reduce soil erosion in olive orchards is to protect the soil with cover crops between rows. Andalusian and European administrations have developed regulations to subsidise the establishment of cover crops between rows in olive orchards. Current methods to follow-up the cover crops systems by administrations consist of sampling and on ground visits of around 1% of the total olive orchards surface at any time from March to late June. This paper outlines a multi-objective neural network based method for the classification of olive trees (OT), bare soil (BS) and different cover crops (CC), using remote sensing data taken in spring and summer.

The main findings of this paper are: (1) the proposed models performed well in all seasons (particularly during the summer, where only 48 pixels of CC are confused with BS and 10 of BS with CC with the best model obtained. This model obtained a 97.80% of global classification, 95.20% in the class with the worst classification rate and 0.9710 in the KAPPA statistics), and (2) the best-performing models could potentially decrease the number of complaints made to the Andalusian and European administrations. The complaints in question concern the poor performance of current on-ground methods to address the presence or absence of cover crops in olive orchards.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Olive (*Olea europaea* L.) is an essential crop in the Mediterranean basin, where it covers around 9.2 M ha of which 2.4 M ha are in Spain and 1.5 M ha in Andalusia, southern Spain (MAPA, 2007). Traditionally, olive trees are separated between 10 and 12 m from each other and soil management is mainly based on intensive tillage operations between rows. Such operations have serious agro-environmental implications and consequences for land erosion and degradation, desertification, sediment transport and increase of atmospheric CO₂ (Hill, Mégier, Mehl, & degradation, 1995; Schlesinger, 2000). To avoid these negative effects, the European Union (EU) and Andalusian administrations only subsidise the implementation of certain conservation agro-environmental techniques, which mainly consist of altering the natural soil as little as possible and protecting it with cover crops. The EU and Andalusian administrations have developed regulations to subsidise the establishment of cover crops between rows in olive orchards (EU Council Regulation 1257, 1999; EU Council Regulation 1259, 1999; Spanish Royal Decree 4, 2001). The Cover Crop Program in perennial crops is being developed as a part of these agro-environmental

measures. The use of cover crops is one of the most effective alternative strategies to manage soil while preventing erosion. Cover crops (2–6 m wide) are sown (grass and broad-leaved species [Alcántara, Sánchez, Pujadas, & Saavedra, 2009]; [Castro et al., 2008]; [Hernández, Lacasta, & Pastor, 2005]) in autumn each year (mid-November in Mediterranean conditions). At the beginning of spring (the end of March in Mediterranean conditions), the cover crops are mowed, grazed or desiccated by the application of post-emergence, non-residual herbicides just before cover crops begin to compete with olive trees for water and nutrients. Furthermore, dead cover crops consisting of the remains of the corresponding olive spring pruning are also considered in some olive orchards. A complete description of soil management practices can be found in Castro et al. (2008).

Currently, to control these agrarian policy actions, the EU and local administrations require precise records of the presence or absence of cover crops. Current methods to examine the cover crop system and to map cover crop soil coverage by the Andalusian administration consist of sampling and ground visits of around 1% of the total olive orchards' surface at any time between March and late June (Annex I, CE no. 1782/2003). This procedure is time-consuming, very expensive, and infeasible in field areas with difficult access, and sometimes delivers inconsistent results due to its coverage of relatively small areas or only very few target fields.

* Corresponding author. Tel.: +34 957 218 349; fax: +34 957 218 630.

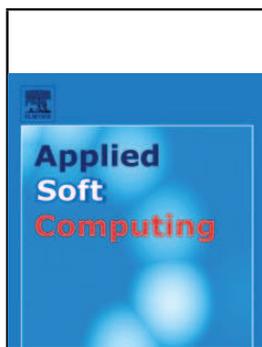
E-mail address: macruz@uco.es (M. Cruz-Ramírez).

TÍTULO:

Evolutionary q-Gaussian radial basis function neural network to determine the microbial growth/no growth interface of Staphylococcus Aureus.

AUTORES:

Francisco Fernández-Navarro, César Hervás-Martínez, **M. Cruz-Ramírez**, Pedro A. Gutiérrez, Antonio Valero.



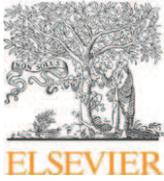
Applied Soft Computing, Volumen 11(3), 3012-3020 (2011).

RANKING:

- A. Índice de Impacto (JCR 2011): 2,612.
- B. Área de conocimiento: *Computer Science, Artificial Intelligence*. Ranking 13/111. Primer Cuartil.
- C. Área de conocimiento: *Computer Science, Interdisciplinary Applications*. Ranking 16/99. Primer Cuartil.

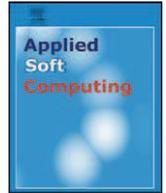
DOI: [10.1016/j.asoc.2010.11.027](https://doi.org/10.1016/j.asoc.2010.11.027)

© Elsevier Publishing Group. Reprinted with permission.



Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Evolutionary q -Gaussian Radial Basis Function Neural Network to determine the microbial growth/no growth interface of *Staphylococcus aureus*

Francisco Fernández-Navarro^{a,*}, César Hervás-Martínez^a, M. Cruz-Ramírez^a,
Pedro Antonio Gutiérrez^a, Antonio Valero^b

^a Department of Computer Science and Numerical Analysis, University of Córdoba, Campus de Rabanales, Albert Einstein Building, 3rd Floor, 14071 Córdoba, Spain

^b Department of Food Science and Technology, University of Córdoba, Campus de Rabanales, Darwin Building, 14014 Córdoba, Spain

ARTICLE INFO

Article history:

Received 10 June 2010

Received in revised form 7 October 2010

Accepted 23 November 2010

Available online 9 December 2010

Keywords:

Imbalanced datasets

Synthetic Minority Over-Sampling

Technique (SMOTE)

q -Gaussian Radial Basis Function Neural

Network

Predictive Microbiology

Memetic Algorithm

ABSTRACT

In this paper, q -Gaussian Radial Basis Functions are presented as an alternative to Gaussian Radial Basis Function. This model is based on q -Gaussian distribution, which parametrizes the Gaussian distribution by adding a new parameter q . The q -Gaussian Radial Basis Function allows different Radial Basis Functions to be represented by updating the new parameter q . For example, when the q -Gaussian function takes a value of $q \rightarrow 1$, it represents the standard Gaussian Radial Basis Function. The model parameters are optimized through a Memetic Algorithm that evolves both its structure and connections. To evaluate the effectiveness of the model, it is tested with a real problem of predictive microbiology. The problem consists of determining the growth boundaries of *Staphylococcus aureus*, a food borne pathogen responsible for several outbreaks. The data from the study of [1] belongs to growth/no growth conditions of *S. aureus* whose temperature, pH and water activity (a_w) has been divided into three categorical classes: growth (G), growth transition (GT) and no growth (NG). Due to the imbalanced nature of the problem, it has been necessary to apply an over-sampling algorithm. The over-sampling procedure selected was the Synthetic Minority Over-Sampling Technique (SMOTE) algorithm. This algorithm has been applied to the patterns in the minority class in order for the performance of the classifier in this class to be acceptable (the minority class in this problem is of vital interest).

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

The use of Artificial Neural Networks (ANNs) as an alternative to other techniques in predictive microbiology has been significant due to their flexibility and high degree of accuracy in fitting to experimental data, all of which has been the object of several research studies [2,3]. Our study focuses on Radial Basis Function Neural Networks (RBFNNs), which have been successfully employed in a variety of pattern recognition problems such as the determination of the microbial growth/no growth interface [4]. Several common types of functions are used as transfer functions, for example, the standard Gaussian (SRBF), the Multiquadratic (MRBF), the Inverse Multiquadratic (IMRBF), and the Cauchy (CRBF).

This paper evaluates a novel RBF based on q -Gaussian distribution which parametrizes standard normal distribution by replacing

* Corresponding author at: Department of Computer Science and Numerical Analysis, University of Córdoba, 14014 Córdoba, Spain. Tel.: +34 957 21 83 49; fax: +34 957 21 83 60.

E-mail address: i22fenaf@uco.es (F. Fernández-Navarro).

the exponential expressions with q -exponential expressions [5], and maximizing Tsallis entropy [6] under certain constraints [7]. q -Gaussian distributions are applicable to a variety of complex signals and systems, and have been applied in a broad range of fields [8] especially including thermodynamics, biology, economics, and quantum mechanics.

Vignat and Plastino [9] show that if the input data exhibit elliptical symmetry and the input data are normalized, the ensuing normalized input will always be a q -Gaussian probability law. The class of elliptically distributed random vectors plays an important role in statistics, and recently garnered a lot of attention in financial mathematics for being especially useful in risk management [8,10] because the normalization processing of data induces a change in its distribution, from a Gaussian to a q -Gaussian one.

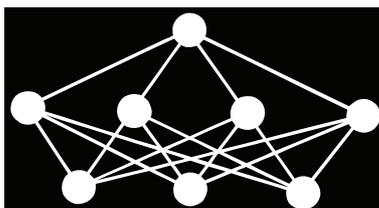
This novel basis function incorporates a real q parameter (besides the centers and widths of the RBF) which can relax or contract the shape of the kernel. This basis function matches both the shape of the kernel and the distribution of the distances better, since the modification of the q parameter allows the representation of different basis functions, among others, Cauchy RBF (CRBF), the standard Gaussian RBF (SRBF), and the Inverse Multiquadratic

TÍTULO:

Evaluating the performance of evolutionary extreme learning machines by a combination of sensitivity and accuracy measures.

AUTORES:

J. Sánchez-Monedero, C. Hervás-Martínez, P.A. Gutiérrez, M. Carbonero-Ruz, M.C. Ramírez-Moreno y **M. Cruz-Ramírez.**



Neural Network World, Volumen 20(7), 899–912 (2010).

RANKING:

- A. Índice de Impacto (JCR 2010): 0,511.
- B. Área de conocimiento: *Computer Science, Artificial Intelligence*. Ranking 94/108. Cuarto Cuartil.

© Academy of Sciences of the Czech Republic. Reprinted with permission.



EVALUATING THE PERFORMANCE OF EVOLUTIONARY EXTREME LEARNING MACHINES BY A COMBINATION OF SENSITIVITY AND ACCURACY MEASURES

*J. Sánchez-Monedero**, *C. Hervás-Martínez**, *P. A. Gutiérrez**,
Mariano Carbonero Ruz†, *M. C. Ramírez Moreno†*, *M. Cruz-Ramírez**

Abstract: Accuracy alone can be deceptive when evaluating the performance of a classifier, especially if the problem involves a high number of classes. This paper proposes an approach used for dealing with multi-class problems, which tries to avoid this issue. The approach is based on the Extreme Learning Machine (ELM) classifier, which is trained by using a Differential Evolution (DE) algorithm. Two error measures (Accuracy, C , and Sensitivity, S) are combined and applied as a fitness function for the algorithm. The proposed approach is able to obtain multi-class classifiers with a high classification rate level in the global dataset with an acceptable level of accuracy for each class. This methodology is evaluated over seven benchmark classification problems and one real problem, obtaining promising results.

Key words: *Accuracy, differential evolution, extreme learning machine, multiclass classification, multiobjective, neural networks, sensitivity*

Received: 20th September 2010

Revised and accepted: 13th November 2010

1. Introduction

In recent years, the imbalanced learning problem has drawn a significant amount of interest. The fundamental issue with the imbalanced learning problem is the ability of imbalanced data to significantly compromise the performance of most standard learning algorithms [1]. If the training methods are not proper, the features representing the classes that have a small number of examples in the training

*J. Sánchez-Monedero, C. Hervás-Martínez, P. A. Gutiérrez, M. Cruz-Ramírez
Department of Computer Science and Numerical Analysis, University of Córdoba, 14071,
Córdoba, Spain, E-mail: {i02samoj, chervas, pagutierrez, i42crram}@uco.es

†Mariano Carbonero Ruz, M. C. Ramírez Moreno
Department of Management and Quantitative Methods, ETEA, Córdoba, Spain, E-mail:
{mariano, mcramirez}@etea.com

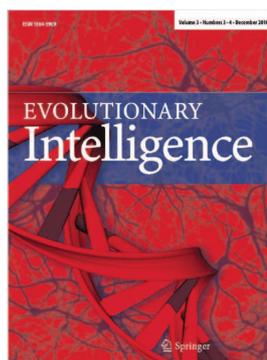
PUBLICACIONES ASOCIADAS CON LA PRESENTE TESIS

TÍTULO:

Memetic pareto differential evolutionary artificial neural networks to determine growth multi-classes in predictive microbiology.

AUTORES:

M. Cruz-Ramírez, J. Sánchez-Monedero, F. Fernández-Navarro, J.C. Fernández y C. Hervás-Martínez.



Evolutionary Intelligence, Volumen 3(3-4), 187-199 (2010).

RANKING:

A. Índice de Impacto (JCR 2009): No indexada.

DOI: [10.1007/s12065-010-0045-9](https://doi.org/10.1007/s12065-010-0045-9)

© Springer Publishing Group. Reprinted with permission.

Memetic pareto differential evolutionary artificial neural networks to determine growth multi-classes in predictive microbiology

M. Cruz-Ramírez · J. Sánchez-Monedero ·
F. Fernández-Navarro · J. C. Fernández ·
C. Hervás-Martínez

Received: 6 October 2010 / Accepted: 6 October 2010 / Published online: 26 October 2010
© Springer-Verlag 2010

Abstract The main objective of this research is to automatically design Artificial Neural Network models with sigmoid basis units for multiclassification tasks in predictive microbiology. The classifiers obtained achieve a double objective: a high classification level in the dataset and high classification levels for each class. The Memetic Pareto Differential Evolution Neural Network chosen to learn the structure and weights of the Neural Networks is a Differential Evolutionary approach based on the Pareto Differential Evolution multiobjective evolutionary algorithm. The Pareto Differential Evolution algorithm is augmented with a local search using the improved Resilient Backpropagation with backtracking-*iRprop*⁺ algorithm. To analyze the robustness of this methodology, it has been applied to two complex classification problems in predictive microbiology (*Staphylococcus aureus* and *Shigella flexneri*). The results obtained show that the generalization ability and the classification rate in each class can be more efficiently improved within this multiobjective algorithm.

Keywords Differential evolution · Memetic algorithms · Multiclassification · Multiobjective · Neural networks · Predictive microbiology

This paper is a very significant extension of a contribution appearing in the 23rd International Conference on Industrial and Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE2010).

M. Cruz-Ramírez (✉) · J. Sánchez-Monedero ·
F. Fernández-Navarro · J. C. Fernández · C. Hervás-Martínez
Department of Computer Science and Numerical Analysis,
University of Córdoba, Rabanales Campus, Albert Einstein
building 3 floor, 14071 Córdoba, Spain
e-mail: mcruz@uco.es

1 Introduction

Growth/No-Growth models have appeared in the predictive microbiology field as an approach to determine the growth ability of microorganisms. In this respect, many studies have been published in recent years for both spoilage and pathogenic microorganisms. This fact is mainly due to the need to gain more knowledge about microbial behaviour in limiting conditions that prevent growth, by using mathematical models. Consequently, these mathematical models may lead to more realistic estimations of food safety risks and can provide useful quantitative data for the development of processes which lead to the production of safer food products [14].

The main problems in modeling microbial interface are related to its abrupt transition, i.e., the great change in the value of growth probability (p) within the very narrow range of environmental factors encountered between growth and no-growth conditions. Thus, to more properly define the interface, growth and no-growth should only be considered if all replicas (reproductions of the same conditions in environmental parameters), or at least a very high percentage of them, grow or do not grow, respectively. In this paper four observed microbial responses are obtained based on the growth probability of a microorganism, ($p = 1$ (growth), G; $0.5 \leq p < 1$ (high growth probability), GHP; $0 < p < 0.5$ (low growth probability), GLP, and $p = 0$ (no-growth), NG).

A classifier design method is usually an algorithm that develops a classifier to approximate an unknown input-output mapping function from finitely available data, i.e., training samples. Once this classifier has been designed, it can be used to predict class labels (G, GHP, GLP or NG) that correspond to unseen samples. Hence, the objective in developing a good classifier is to ensure high prediction

A.3 ARTÍCULOS EN CONGRESOS INTERNACIONALES

- **M. Cruz-Ramírez**, J.C. Fernández, A. Valero, P.A. Gutiérrez y C. Hervás-Martínez. Multi-objective Pareto ordinal classification for predictive microbiology. En *Soft Computing Models in Industrial and Environmental Applications (SOCO2012)*, volumen 188 de *Advances in Intelligent Systems and Computing*, págs. 153–162, 2012.
- P.A. Gutiérrez, J. Sánchez-Monedero, C. Hervás-Martínez, **M. Cruz-Ramírez**, J.C. Fernández y F. Fernández-Navarro. Approaching system administration as a group project in computer engineering higher education. En *International Joint Conference CISIS12-ICEUTE12-SOCO12 Special Sessions*, volumen 189 de *Advances in Intelligent Systems and Computing*, págs. 331–340, 2012.
- M. Pérez-Ortiz, **M. Cruz-Ramírez**, J.C. Fernández-Caballero y C. Hervás-Martínez. Hybrid multi-objective machine learning classification in liver transplantation. En *Proceedings of the 7th international conference on Hybrid Artificial Intelligent Systems (HAIS2012)*, volumen 1, págs. 397–408, 2012.
- **M. Cruz-Ramírez**, C. Hervás-Martínez, J. Sánchez-Monedero y P.A. Gutiérrez. A Preliminary Study of Ordinal Metrics to Guide a Multi-Objective Evolutionary Algorithm. En *11th International Conference on Intelligent Systems Design and Applications (ISDA2011)*, págs. 1176–1181, 2011.
- J. Briceño, **M. Cruz-Ramírez**, M. Prieto, M. Navasa, J. Ortiz, R. Orti, M.A. Gómez, A. Otero, S. Tome, E. Varo, G. Clemente, R. Bañares, R. Bárcena, V. Cuervas, G. Solórzano, C. Hervás y M. de la Mata. Donor-recipient matching in liver transplantation based on a rule-system built on a multiobjective artificial neural network. En *15th Congress of the European Society for Organ Transplantation (ESOT2011)*, volumen 24, págs. 110–110, 2011.
- **M. Cruz-Ramírez**, J.C. Fernández, F. Fernández-Navarro, J. Briceño, M. de la Mata y C. Hervás-Martínez. Memetic Evolutionary Multi-Objective Neural Network Classifier to Predict Graft Survival in Liver Transplant Patients. En *Genetic and Evolutionary Computation Conference (GECCO2011)*, págs. 479–486, 2011.
- J. Briceño, **M. Cruz-Ramírez**, M. Prieto, M. Navasa, J. Ortiz, R. Orti, M.A. Gómez, A. Otero, S. Tome, E. Varo, G. Clemente, R. Bañares, R. Bárcena, V. Cuervas, G. Solórzano, C. Hervás y M. de la Mata. Donor-recipient matching in liver transplantation based on a rule-system built on a multiobjective artificial neural network. En *Congress of ILTS, ELITA y LICAGE (ILTS2011)*, volumen 17, págs. S90–S91, 2011.

- J. Briceño, **M. Cruz-Ramírez**, M. Prieto, M. Navasa, J. Ortiz, R. Orti, M.A. Gómez, A. Otero, S. Tome, E. Varo, G. Clemente, R. Bañares, R. Bárcena, V. Cuervas, G. Solórzano, C. Hervás y M. de la Mata. Donor-recipient matching in liver transplantation based on a rule-system built on a multiobjective artificial neural network. En *American Transplant Congress (ATC2011)*, volumen 11, págs. 423–423, 2011.
- **M. Cruz-Ramírez**, C. Hervás-Martínez, P.A. Gutiérrez, J. Briceño y M. de la Mata. Memetic Pareto Differential Evolutionary Neural Network for Donor-Recipient Matching in Liver Transplantation. En *International Work conference on Artificial Neural Networks (IWANN2011)*, págs. 129–136, 2011.
- **M. Cruz-Ramírez**, J.C. Fernández, F. Fernández-Navarro, J. Sánchez-Monedero y C. Hervás-Martínez. Selecting the best artificial neural network model from a multi-objective Differential Evolution Pareto front. En *IEEE Symposium on Differential Evolution (IEEE Symposium Series on Computational Intelligence) (SDE (SS-CI2011))*, págs. 96–103, 2011.
- **M. Cruz-Ramírez**, J.C. Fernández, J. Sánchez-Monedero, F. Fernández-Navarro, C. Hervás-Martínez, P.A. Gutiérrez y M.T. Lamata. Ensemble determination using the TOPSIS decision support system in multi-objective evolutionary neural network classifiers. En *10th International Conference on Intelligent Systems Design and Applications (ISDA2010)*, págs. 513–518, 2010.
- J. Sánchez-Monedero, **M. Cruz-Ramírez**, F. Fernández-Navarro, J.C. Fernández, P.A. Gutiérrez y C. Hervás-Martínez. On the suitability of Extreme Learning Machine for gene classification using feature selection. En *10th International Conference on Intelligent Systems Design and Applications (ISDA2010)*, págs. 507–512, 2010.
- J. Sánchez-Monedero, P.A. Gutiérrez, C. Hervás-Martínez, **M. Cruz-Ramírez**, J.C. Fernández y F. Fernández-Navarro. Methodology for the recognition and diagnosis of students performance by discriminant analysis and artificial neural networks. En *1st International Conference on European Transnational Education (ICEUTE2010)*, págs. 111–119, 2010.
- F. Fernández-Navarro, C. Hervás-Martínez, P.A. Gutiérrez, **M. Cruz-Ramírez** y M. Carbonero-Ruz. Evolutionary q-Gaussian Radial Basis Functions for Binary-Classification. En *5th International Conference on Hybrid Artificial Intelligence Systems (HAIS2010)*, págs. 280–287, 2010.
- J. Sánchez-Monedero, C. Hervás-Martínez, F.J. Martínez-Estudillo, M. Carbonero-Ruz, M.C. Ramírez-Moreno y **M. Cruz-Ramírez**. Evolutionary Learning using a Sensitivity-Accuracy Approach for Classification. En *5th International Conference on Hybrid Artificial Intelligence Systems (HAIS2010)*, págs. 288–295, 2010.

- **M. Cruz-Ramírez**, C. Hervás-Martínez, J.C. Fernández y J. Sánchez-Monedero. Learning Artificial Neural Networks Multiclassifiers by Evolutionary Multiobjective Differential Evolution Guided by Statistical Distributions. En *International Joint Conference on Neural Networks (IJCNN2010)*, págs. 2540–2547, 2010.
- **M. Cruz-Ramírez**, J. Sánchez-Monedero, F. Fernández-Navarro, J.C. Fernández y C. Hervás-Martínez. Hybrid Pareto Differential Evolutionary Artificial Neural Networks to determined growth multi-classes in Predictive Microbiology. En *23rd International Conference on Industrial and Engineering and Other Applications of Applied Intelligent Systems (IEA-AIE2010)*, págs. 646–655, 2010.
- J.C. Fernández, C. Hervás-Martínez, F.J. Martínez-Estudillo y **M. Cruz-Ramírez**. Design of Artificial Neural Networks using a Memetic Pareto Evolutionary Algorithm using as objectives Entropy versus Variation Coefficient. En *2009 Ninth International Conference on Intelligent Systems Design and Applications (ISDA2009)*, págs. 408–413, 2009.
- J.C. Fernández, C. Hervás-Martínez, F.J. Martínez-Estudillo, P.A. Gutiérrez y **M. Cruz-Ramírez**. Memetic Pareto Differential Evolution for designing Artificial Neural Networks in Multiclassification Problems using Cross-Entropy versus Sensitivity. En *4th International Conference on Hybrid Artificial Intelligence Systems and LNAI 5572 (HAIS2009)*, págs. 433–441, 2009.

A.4 ARTÍCULOS EN CONGRESOS NACIONALES

- C. Hervás-Martínez, M. Pérez-Ortiz, **M. Cruz-Ramírez** y P.A. Gutiérrez. Asignación de órganos en trasplante hepático mediante regresión ordinal. En *VI Simposio de Teoría y Aplicaciones de Minería de Datos (TAMIDA2013)*, enviado, 2013.
- J. Briceño, **M. Cruz-Ramírez**, M. Prieto, M. Navasa, J. Ortiz De Urbina, R. Ortiz, M.A. Gómez-Bravo, A. Otero, E. Varo, S. Tomé, G. Clemente, R. Bañares, R. Bárcena, V. Cuervas-Mons, G. Solórzano, C. Vinaixa, A. Rubin, J. Colmenero, A. Valdivieso, C. Hervás y M. De La Mata. Aplicación de redes neuronales artificiales multiobjetivo en el emparejamiento donante-receptor en trasplante hepático. En *Congreso sociedad española de trasplante hepático (SETH2011)*, 2011.
- **M. Cruz-Ramírez**, C. Hervás-Martínez y M.T. Lamata. Estudio de un Modelo de Ensemble formado por los Extremos del Frente de Pareto en Algoritmos Evolutivos Multi-objetivo Utilizando la Técnica TOPSIS. En *VII Congreso Español sobre Metaheurísticas y Algoritmos Evolutivos y Bioinspirados (MAEB2010)*, págs. 219–226, 2010.
- F. Fernández-Navarro, C. Hervás-Martínez, P.A. Gutiérrez, **M. Cruz-Ramírez** y J. Sánchez-Monedero. Aprendizaje híbrido de redes neuronales q-Gaussianas

en clasificación binaria. En *VII Congreso Español sobre Metaheurísticas y Algoritmos Evolutivos y Bioinspirados (MAEB2010)*, págs. 227–234, 2010.

- F. Fernández-Navarro, P.A. Gutiérrez y **M. Cruz-Ramírez**. Técnica de Hibridación de un Algoritmo Evolutivo y una Búsqueda Local basada en Análisis Cluster para la Optimización de Redes Neuronales RBF. En *VI Congreso Español sobre Metaheurísticas y Algoritmos Evolutivos y Bioinspirados (MAEB2009)*, págs. 317–324, 2009.