

# Evolución Diferencial para Reducción de Prototipos y Ponderación de Características

Isaac Triguero, Joaquín Derrac, Salvador García y Francisco Herrera

*Resumen*—Las técnicas de generación de prototipos han demostrado ser muy competitivas mejorando el rendimiento del clasificador del vecino más cercano. Dentro de la metodología de la generación de prototipos cabe destacar los métodos de posicionamiento de prototipos como aquellos que poseen una mayor eficacia. En la literatura se han usado distintos algoritmos evolutivos para optimizar la posición de los prototipos con resultados prometedores. No obstante, estos resultados pueden ser mejorados si otras técnicas de reducción de datos, tales como la selección de prototipos o los esquemas de pesos, son consideradas.

En este trabajo se propone un enfoque híbrido para reducción de prototipos, integrando un esquema de pesos con una de las técnicas más sobresalientes en reducción de prototipos. Concretamente, se aplicará una técnica auto-adaptativa de evolución diferencial con el fin de optimizar el peso asignado a cada característica y la localización de los prototipos. Los resultados obtenidos, contrastados mediante técnicas estadísticas, avalan la utilidad de este enfoque híbrido para mejorar el rendimiento del clasificador del vecino más cercano.

*Palabras clave*—Reducción de prototipos, evolución diferencial, ponderación de características, esquema de pesos, clasificación

## I. INTRODUCCIÓN

La regla del vecino más cercano (*Nearest Neighbor*, NN) [1] es una técnica de clasificación supervisada, caracterizada por su simplicidad y efectividad, que pertenece a la familia de los métodos de aprendizaje basado en instancias [2]. El NN es un clasificador no paramétrico que requiere que todos los datos de entrenamiento estén almacenados. Los ejemplos no conocidos son clasificados en base a la clase de las instancias más cercanas a ellos, en términos de distancia. A pesar de su simplicidad y eficacia, esta técnica tiene algunas debilidades como el tiempo de respuesta, la tolerancia al ruido y el coste de almacenamiento. Además, este clasificador realiza sus predicciones en base a los datos de entrada, asumiendo que estos definen perfectamente la distribución de clases.

Estas debilidades han sido abordadas por diferentes enfoques. Entre ellos, una buena solución que podemos encontrar en la literatura consiste en reducir el conjunto de datos usados para clasificar. Este tipo de técnicas tratan de reducir el conjunto de entrenamiento [3], con el objetivo de eliminar ejemplos

ruidosos, redundantes e irrelevantes. Desde el punto de vista del espacio de las características/atributos, podemos destacar la selección de características [4] como una de las técnicas más importantes, que consiste en elegir un subconjunto representativo de atributos. Por otro lado, desde el punto de vista de las instancias, los métodos de reducción de datos se dividen en dos enfoques, conocidos como selección de prototipos (SP) [5] y generación/abstracción de prototipos (GP) [6]. La SP consiste en seleccionar un subconjunto de los datos de entrenamiento originales, mientras que la GP puede generar nuevos datos artificiales si los necesita. De este modo, la GP no supone que los datos originales delimitan perfectamente las fronteras de decisión entre clases <sup>1</sup>.

Otra propuesta interesante es el empleo de esquemas de pesos aplicados a las características (*Feature Weighting*, FW) [7] que permite asignar un peso a cada característica del dominio del problema para modificar el modo en que las distancias entre ejemplos son calculadas. Esta técnica puede verse como una generalización de los métodos de selección de características, permitiendo dar un grado de relevancia a cada atributo del problema mediante un valor real.

En los últimos años han aparecido una gran cantidad de propuestas evolutivas aplicadas a problemas de minería de datos. Los problemas de GP y FW pueden ser vistos como problemas de optimización continua, por lo que también pueden ser abordados mediante algoritmos evolutivos. Los algoritmos evolutivos para GP se basan en técnicas de ajuste del posicionamiento de prototipos [8], que es una metodología adecuada para optimizar la localización de los prototipos. Específicamente, el algoritmo de Evolución Diferencial (ED) [9] y sus propuestas avanzadas [10] han demostrado ser las técnicas de posicionamiento muy efectivas. Respecto a los métodos de FW, se han propuesto muchas técnicas evolutivas aplicadas al algoritmo NN [11].

Habitualmente, los métodos de ajuste de la localización de los prototipos [8] se centran en el proceso de optimizar la posición, sin tomar en consideración la selección más apropiada del número de prototipos por clase. Recientemente, en [12], este problema se aborda aplicando un método memético de SP, conocido como SSMA [13], como etapa previa para proveer una selección apropiada de prototipos por clase

I. Triguero, J. Derrac y F. Herrera son miembros del Dept. de Ciencias de la Computación e Inteligencia Artificial, CITIC-UGR. Universidad de Granada. 18071, Granada, España, E-mails: triguero@decsai.ugr.es, jderrac@decsai.ugr.es, herrera@decsai.ugr.es.

S. García es miembro del Dept. de Informática, Universidad de Jaén, 23071, Jaén, España, E-mail: sglopez@ujaen.es

<sup>1</sup>Más información en el sitio web temático de SCI2S sobre *Prototype Reduction in Nearest Neighbor Classification: Prototype Selection and Prototype Generation* <http://sci2s.ugr.es/pr/>

y posteriormente optimizar su localización mediante ED. Este algoritmo se conoce con el nombre de SSMA-EDGP.

En este trabajo, se propone un método híbrido que combina SSMA-EDGP con un esquema de FW. En este esquema, los pesos de cada característica y la localización de los prototipos se determinan mediante un método auto-adaptativo de ED, conocido con el nombre de SFLSDE [10]. Este método se emplea para abordar los problemas de FW y GP simultáneamente. Los algoritmos de GP evolutivos tienden a sobre-entrenar los datos de entrenamiento, en pocas iteraciones, por lo que la modificación de la función de distancia mediante FW puede evitar este problema, determinando además un grado de relevancia para cada característica. Tras su descripción, se presenta un estudio experimental en el que se muestran las mejoras significativas en comparación con la técnica original de GP y otras técnicas de FW.

El resto del trabajo se organiza como sigue: La Sección II presenta algunos conceptos preliminares sobre las técnicas empleadas. La Sección III describe el modelo propuesto. La Sección IV muestra el estudio experimental y presenta los resultados. Por último, la Sección V concluye el trabajo.

## II. PRELIMINARES

Esta sección repasa algunos conceptos preliminares necesarios. La Sección II-A define formalmente los problemas de selección y generación de prototipos. La Sección II-B comenta el empleo de esquemas de pesos aplicados a las características. Finalmente, la Sección II-C explica el algoritmo de ED.

### A. Selección y Generación de prototipos

Tanto la selección como la generación de prototipos [6] tienen como objetivo obtener un conjunto de entrenamiento lo más reducido posible que permita al NN clasificar con la misma o más calidad que con el conjunto de entrenamiento original. Esto permite reducir la complejidad espacial del método y reducir su coste computacional. Además, en ocasiones puede mejorar su precisión, mediante la eliminación de ruido.

Formalmente, la SP se define como sigue: Sea  $\mathbf{X}$  una instancia donde  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M, \mathbf{c})$ , con  $\mathbf{X}$  perteneciendo a la clase  $\mathbf{c}$  y un espacio  $M$ -dimensional donde  $\mathbf{x}_i$  es el  $i$ -ésimo valor de la instancia  $X$ . Asumamos que existe un conjunto de entrenamiento  $CE$  compuesto por  $N$  instancias, y un conjunto de test  $CT$  compuesto por  $T$  instancias. La selección de prototipos consiste en obtener un conjunto reducido de prototipos ( $CR$ ), tal que  $CR \subseteq CE$ , con el que clasificar los ejemplos conjunto  $CT$  mediante la regla del vecino más cercano.

El propósito de la GP es obtener un  $CR$ , que consiste en  $r$ ,  $r < N$ , prototipos  $\mathbf{P}$  donde  $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_M, \mathbf{c})$ , que son generados a partir de los

ejemplos de  $CE$ . El  $CR$  debe representar eficientemente la distribución de clases y su tamaño debe ser lo suficientemente pequeño para mejorar el tiempo y necesidad de almacenamiento del clasificador NN.

### B. Esquemas de pesos

El objetivo de los métodos de FW es reducir la sensibilidad a características redundantes, irrelevantes o ruidosas en el clasificador NN, mediante la modificación de la medida de distancia con pesos, para mejorar su precisión en clasificación.

Normalmente se usa la distancia euclídea (Ecuación 1) en el clasificador NN, donde  $P$  and  $Q$  son dos ejemplos y  $M$  es el número de características de estos. Durante este estudio se usará dicha distancia, dada su simplicidad, facilidad para optimizarla y su amplio uso en el campo del aprendizaje basado en instancias [2].

$$DistanciaEuclidea(P, Q) = \sqrt{\sum_{i=1}^M (p_i - q_i)^2} \quad (1)$$

Lo métodos de FW suelen extender dicha ecuación aplicando diferentes pesos a cada característica ( $W_i$ ) que modifica la forma en que la medida de distancia es calculada (Ecuación 2).

$$DistanciaFW(P, Q) = \sqrt{\sum_{i=1}^M W_i \cdot (p_i - q_i)^2} \quad (2)$$

Esta técnica ha sido ampliamente usada en la literatura. En [7] podemos encontrar un estudio completo sobre el tema.

### C. Evolución diferencial

El algoritmo de ED comienza con una población de  $NP$  individuos. La población inicial debe cubrir el espacio de búsqueda tanto como sea posible. En algunos problemas esto se consigue inicializando los individuos aleatoriamente, pero en otros problemas, como en la GP, existe un conocimiento base que está disponible para ser usado en los mecanismos de inicialización. Las generaciones siguientes se denotan como:  $G = 0, 1, \dots, G_{max}$ . En ED es usual denominar a cada individuo como un vector  $D$ -dimensional  $X_{i,G} = \{x_{i,G}^1, \dots, x_{i,G}^D\}$ , llamado vector objetivo.

Después de la inicialización, el algoritmo ED aplica un operador de mutación para generar un vector mutado  $V_{i,G}$  respecto a cada individuo  $X_{i,G}$ , en la población actual. Para cada vector objetivo  $X_{i,G}$ , su vector mutado asociado es  $V_{i,G} = \{V_{i,G}^1, \dots, V_{i,G}^D\}$ .

En este trabajo nos centramos en el uso de la estrategia *RandToBest/1* que genera un vector mutado de la siguiente forma:

$$V_{i,G} = X_{i,G} + F \cdot (X_{mejor,G} - X_{i,G}) + F \cdot (X_{r_1^i,G} - X_{r_2^i,G}) \quad (3)$$

Donde los índices  $r_1^i, r_2^i$  son enteros mutuamente exclusivos y generados aleatoriamente en el rango  $[1, NP]$ , y son también diferentes del índice base  $i$ . El factor de escala  $F$  es un parámetro de control positivo para escalar la diferencia entre vectores.

Tras la fase de mutación, el operador de cruce se aplica para cada par, vector objetivo  $X_{i,G}$  y su correspondiente vector mutado  $V_{i,G}$ , para generar un nuevo vector, denominado vector de prueba  $U_{i,G}$ . Nos centraremos en el uso del operador de cruce binomial, inicializado con los valores del vector objetivo, y en el cual cada componente del vector de prueba se modifica con los valores del vector mutado si al generar un número aleatorio entre 0 y 1, éste es menor o igual que un ratio de cruce  $RC$ . Finalmente, debemos decidir qué individuo pasa a la siguiente generación  $G + 1$ . Si el vector de prueba obtiene una solución igual o mejor que el vector objetivo, este reemplaza su correspondiente vector objetivo en la siguiente generación; en otro caso, el vector objetivo se mantiene en la población.

### C.1 Propuestas Avanzadas para Evolución Diferencial

El éxito del algoritmo de ED para resolver un problema específico depende crucialmente de la elección apropiada de sus parámetros de control asociados ( $F$  y  $RC$ ) que determinan la convergencia del algoritmo. Por lo tanto, una selección fija de estos parámetros puede producir una baja y/o prematura convergencia dependiendo del problema. Por ello, muchos investigadores se han centrado en desarrollar mecanismos para mejorar el rendimiento del algoritmo básico de ED [14], [15].

Una de las técnicas adaptativas más competitiva es el algoritmo de evolución diferencial con búsqueda local para el factor de escala (Scale Factor Local Search in Differential Evolution, SFLSDE). Este método fue destacado como la técnica más competitiva aplicada a GP en [12].

## III. UN MODELO HÍBRIDO DE ESQUEMAS DE PESOS Y REDUCCIÓN DE PROTOTIPOS

En esta sección se describe el modelo híbrido propuesto y sus principales componentes. En la Sección III-A se presenta el modelo propuesto como esquema de pesos. A continuación, la Sección III-B muestra las principales características del método de partida SSMA-EDGP. Finalmente, la Sección III-C presenta el modelo híbrido propuesto en detalle.

### A. Evolución diferencial como un esquema de pesos

Como se dijo anteriormente, los esquemas de pesos aplicados a las características se pueden entender como un problema de búsqueda en un espacio continuo en el cual necesitamos determinar la ponderación más apropiada para cada característica con el fin de mejorar el clasificador NN. Específicamente, se propone el uso de un algoritmo de ED para obtener

los mejores pesos que permitan, a un conjunto reducido  $CR$ , incrementar su precisión en clasificación sobre el conjunto  $CE$ . Denominamos a este algoritmo con las siglas *EDFW* (Evolución Diferencial para FW).

El *EDFW* comienza con una población de  $NP$  individuos  $X_{i,G}$ . Este algoritmo codifica un vector de pesos en cada individuo, como un vector  $M$ -dimensional de números reales, que contiene la ponderación de cada característica mediante un valor comprendido en el intervalo  $[0,1]$ . Esto implica, que cada individuo de la población codifica una solución completa para el problema de FW. Siguiendo las ideas establecidas en [10], la población inicial se genera de forma aleatoria dentro del rango definido anteriormente.

Después del proceso de inicialización, el método *EDFW* entra en un bucle en el cual los operadores de mutación y cruce guían el proceso de optimización de los pesos en las características generando nuevos vectores  $U_{i,G}$ . Tras aplicar dichos operadores, como se explicaba en la Sección II-C, debemos verificar si los valores generados pertenecen al rango  $[\theta, 1]$ . Si el valor es mayor que uno, este se truncará a uno. Además, basándonos en el trabajo de Kira y Rendell [18], si este valor es menor que un umbral  $\theta$ , se considera que esta característica es irrelevante, y por tanto se establece a 0. En nuestros experimentos, este umbral se ha establecido empíricamente en 0,2.

Por último, el operador de selección decide si el vector generado debe entrar en la siguiente generación  $G + 1$ . La regla del NN guía este operador para cubrir nuestros propósitos. Los ejemplos de  $CE$  son clasificados con los prototipos de un  $CR$  dado, pero en este caso, la medida de distancia usada en el clasificador NN es modificada de acuerdo con la Ecuación 2, donde los pesos  $W_i$  son obtenidos de  $X_{i,G}$  y  $U_{i,G}$ . La precisión o acierto en clasificación se mide como el número de éxitos de clasificación dividido entre el número total de clasificaciones realizadas. Nuestro objetivo es maximizar dicho valor, por ello, el operador de selección se puede ver como:

$$X_{i,G+1} = \begin{cases} U_{i,G} & \text{if } Precision(CR, U_{i,G}) \\ & \geq Precision(CR, X_{i,G}) \\ X_{i,G} & \text{En otro caso} \end{cases} \quad (4)$$

En caso de empate entre los valores de precisión obtenidos, seleccionamos el vector  $U_{i,G}$  con el fin de dar al individuo mutado la posibilidad de entrar en la población.

Para evitar que los parámetros más importantes en ED, ( $F$  y  $RC$ ), provoquen una convergencia lenta o prematura implementamos el método SFLSDE [10] como esquema auto-adaptativo de ED.

### B. SSMA-EDGP: Combinando selección y generación de prototipos

El enfoque SSMA-EDGP nace como una combinación de SP y GP. El objetivo de este método es el de abordar las principales debilidades de cada uno de ellos por separado. La principal desventaja de los métodos de SP es que asumen que los ejemplos más representativos se pueden obtener como un subconjunto del conjunto de entrenamiento original. Sin embargo, la GP puede generar nuevos ejemplos para definir mejor la fronteras de decisión entre clases. Específicamente, los métodos de ajuste de la localización de prototipos tienen por objetivo corregir la posición de un subconjunto de prototipos del  $CR$ .

No obstante, estos métodos de GP sufren ciertas debilidades:

- Abordan un problema más complejo que la SP, por lo que el espacio de búsqueda puede ser más difícil de explorar. Como resultado, encontrar una solución prometedora requiere un mayor coste que el de un método de SP.
- Usualmente los métodos de ajuste del posicionamiento inicializan el conjunto a generar  $CR$  con un número fijo de prototipos obtenidos aleatoriamente del  $CR$ . Esta característica es una de las principales debilidades de estos métodos, ya que este parámetro puede ser muy dependiente del problema abordado.

Para abordar dichos problemas, SSMA-EDGP usa un algoritmo de SP con el fin de realizar la elección más apropiada de prototipos por clase. Concretamente, el algoritmo SSMA [13] se aplica y el conjunto reducido resultante es introducido como uno de los individuos de la población en el algoritmo SFLSDE, que en este caso, actúa como un método de GP. A continuación, se realizan las operaciones de mutación y cruce para generar nuevas soluciones candidatas. De nuevo, se usa la regla del vecino más cercano para guiar el operador de selección. Finalmente, el método SSMA-EDGP devuelve el mejor posicionamiento de los prototipos que mejora índice de clasificación. La Figura 1 muestra el esquema del modelo SSMA-EDGP.

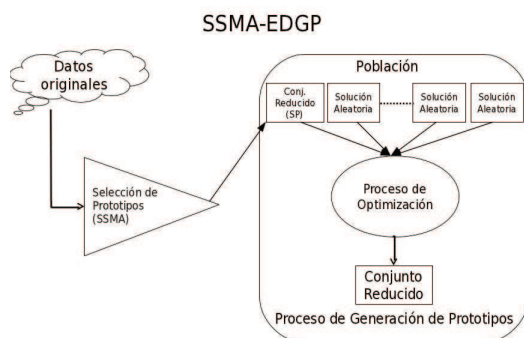


Fig. 1. SSMA-EDGP: Combinando SP y GP

- 1:  $CR[1] = SSMA()$ ;
- 2: Generar los individuos  $CR[2..NP]$  aleatoriamente con la misma estructura que  $CR[1]$
- 3:  $Pesos[1..M] = 1.0$
- 4: Determinar el mejor  $CR$
- 5: **for**  $i = 1$  to  $MAXITER$  **do**
- 6:  $Pesos[1..M] = EDFW(CR[best], Pesos)$
- 7:  $CR[1..NP] = SFLSDE(CR[1..NP], Pesos)$
- 8: Determinar el mejor  $CR$
- 9: **end for**
- 10: **return**  $CR[best]$ ,  $Pesos$

Fig. 2. Híbrido SSMA-EDGPFW

### C. Híbrido SSMA-EDGP y esquema de pesos

El modelo híbrido que compone SSMA-EDGP y EDFW puede ser descrito básicamente como la combinación de SP, GP y FW. La figura 2 muestra el pseudo-código de esta propuesta, denominada SSMA-EDGPFW. Para hibridar el esquema de pesos propuesto con SSMA-EDGP, es necesario aplicar en primera instancia una selección apropiada de prototipos por clase, usando SSMA como algoritmo de SP (Instrucción 1). A continuación, el resto de individuos es generado aleatoriamente, extrayendo prototipos del  $CR$  (y manteniendo la misma estructura que el conjunto seleccionado por SSMA) (Instrucción 2). En esta etapa, establecemos a 1,0 el grado de relevancia de todas las características (Instrucción 3).

La instrucción 4 determina el mejor individuo de la población en base a su índice de clasificación en  $CR$ . A continuación, el modelo híbrido entra un ciclo en el cual se intenta encontrar la ponderación más apropiada para cada atributo y el mejor posicionamiento para cada prototipo. Inicialmente, el método EDFW es aplicado con el mejor  $CR$  encontrado hasta el momento. Nótese, que los "Pesos" actuales son introducidos como un individuo de la población de la población del método EDFW para asegurar que el método FW no degrada en ningún momento el rendimiento del  $CR$  (Instrucción 6). Ahora, en la instrucción 7, una nueva etapa de optimización es aplicada sobre todos los individuos de la población, de modo que el proceso de optimización debe tomar en consideración los nuevos pesos para calcular las distancias entre prototipos (Ver Ecuación 2). La idea subyacente de esta instrucción es que el algoritmo SFLSDE debería generar un conjunto  $CR$  diferente debido al hecho de que la medida de distancia ha cambiado, por lo tanto, el espacio de búsqueda se ha modificado.

Finalmente, tras un número previamente fijado de iteraciones, el modelo devuelve el mejor  $CR$  con sus respectivos pesos, que está listo para ser usado como conjunto de referencia por el clasificador NN.

#### IV. ESTUDIO EXPERIMENTAL Y ANÁLISIS DE RESULTADOS

Para tratar de caracterizar el rendimiento del híbrido SSMA-EDGPFW, se ha realizado un estudio experimental sobre diferentes problemas de clasificación. La Sección IV-A describe los conjuntos de datos utilizados. La Sección IV-B enumera los algoritmos de comparación considerados y describe sus parámetros. La Sección IV-C presenta y analiza los resultados obtenidos, junto con un estudio estadístico para contrastarlos.

##### A. Conjuntos de datos

En este estudio, se han empleado 30 conjuntos de datos representando problemas de clasificación. Han sido tomados del repositorio *KEEL-dataset Repository*<sup>2</sup> [19]. La Tabla I describe sus principales características: Nombre, número de instancias (#In), número de atributos (#At) y número de clases (#Cl).

TABLA I  
CONJUNTOS DE DATOS UTILIZADOS

Conjunto	#In	#At.	#Cl.	Conjunto	#In	#At.	#Cl.
abalone	4.174	8	28	mammographic	961	5	2
appendicitis	106	7	2	marketing	8.993	13	9
banana	5.300	2	2	movement_libras	360	90	15
bupa	345	6	2	newthyroid	215	5	3
chess	3.196	36	2	nursery	12.690	8	5
contraceptive	1.473	9	3	pima	768	8	2
german	1.000	20	2	segment	2.310	19	7
glass	214	9	7	sonar	208	60	2
haberman	306	3	2	spectfheart	267	44	2
hayes-roth	133	4	3	splice	3.190	60	3
heart	270	13	2	tae	151	5	3
iris	150	4	3	texture	5.500	40	11
led7digit	500	7	10	thyroid	7.200	21	3
lym	148	18	4	wisconsin	683	9	2
magic	19.020	10	2	zoo	101	16	7

Todos los conjuntos han sido particionados, empleando la técnica de validación cruzada de 10 partes. Además, sus valores han sido normalizados en el intervalo  $[0, 1]$ , para igualar la influencia de todos los atributos con respecto a la medida de distancia del clasificador.

##### B. Algoritmos y parámetros

Se han empleado tres métodos de comparación para realizar un estudio completo de nuestra propuesta. El primero de ellos es el algoritmo base SSMA-EDGP, y dos algoritmos bien conocidos en el ámbito de los esquemas de pesos: TS/KNN [20] y ReliefF [21]. Además, se ha considerado también el clasificador 1-NN como referencia básica. La Tabla II muestra los parámetros empleados. Todos los métodos estocásticos se han ejecutado tres veces por partición.

##### C. Resultados obtenidos y estudio estadístico

En este estudio experimental se analizan los resultados en términos de precisión o acierto obtenido durante la clasificación de los datos de test.

<sup>2</sup><http://www.keel.es/datasets.php>

TABLA II  
PARÁMETROS EMPLEADOS

Método	Parámetros
SSMA-EDGPFW	<i>MAXITER</i> = 20, PoblaciónSFLSDE = 40, Iteraciones ED = 50 PoblaciónEDFW=25, Iteraciones EDFW=200, iterSFGSS=8, iterSFHC=20, F1=0.1, Fu=0.9
SSMA-EDGP	Población=40, Iteraciones ED = 500, iterSFGSS=8, iterSFHC=20, F1=0.1, Fu=0.9
TSKNN	Evaluaciones = 10000, M= 10, N= 2, P= $\text{ceil}(\sqrt{\#Caracteristicas})$
ReliefF	Valor de <i>K</i> para contribuciones= El mejor en [1,20] Clasificador base: 1-NN

La Tabla III muestra los resultados obtenidos. Para cada conjunto se muestra el valor medio obtenido en cada conjunto por cada algoritmo (Ac.) y su correspondiente desviación típica (Des.). Además, se ha remarcado en negrita el mejor resultado en acierto obtenido en cada conjunto.

Para contrastar los resultados experimentales se van a emplear tests no paramétricos de comparaciones múltiples. Su empleo en minería de datos está recomendado en los casos en que se intenten comparar los resultados de un nuevo algoritmo con respecto a varios métodos simultáneamente [16].

Específicamente, se ha seleccionado el test de Friedman como método para detectar la existencia de diferencias significativas entre los resultados de acierto, y el método de Holm como test *post-hoc* para caracterizar las diferencias encontradas [17]<sup>3</sup>.

El test de Friedman detecta diferencias significativas ( $p - \text{valor} < 10^{-6}$ ) sobre los resultados. A partir de los rangos obtenidos, se selecciona a SSMA-EDFWGP como método de control (aquel que obtiene el rango más bajo), y se aplica el test de Holm. La Tabla IV resume los resultados del estudio estadístico.

A partir de los resultados, se pueden extraer las siguientes conclusiones:

- SSMA-EDGPFW obtiene mejor acierto medio que el resto. Además, obtiene el mejor resultado en 17 de los 30 problemas considerados.
- Comparando SSMA-EDGPFW con SSMA-EDGP, se aprecia una buena sinergia entre GP y FW, ya que los resultados medios han aumentado considerablemente, siendo un algoritmo más robusto ante diferentes tipos de problemas.
- SSMA-EDGPFW mejora estadísticamente los resultados obtenidos por el resto, con un nivel de significancia  $\alpha = 0,05$ . Por tanto, contrastan la afirmación de que la mejora obtenida en precisión de clasificación por SSMA-EDGPFW sobre el resto de métodos es significativa.

#### V. CONCLUSIONES

En este trabajo se ha presentado un modelo híbrido que combina selección y generación de prototipos con un esquema de pesos para mejorar la regla del

<sup>3</sup>Más información en el sitio web temático de SCI2S sobre *Statistical Inference in Computational Intelligence and Data Mining* <http://sci2s.ugr.es/sicidm/>

TABLA III  
RESULTADOS OBTENIDOS

Conjunto	1-NN		SSMA-EDGP		SSMA-EDGPFW		TS/KNN		Relieff	
	Ac.	Des.	Ac.	Des.	Ac.	Des.	Ac.	Des.	Ac.	Des.
abalone	19,91	1,60	<b>25,66</b>	1,71	25,61	1,34	24,65	1,43	14,71	1,85
appendicitis	79,36	11,51	83,27	7,60	84,27	11,39	<b>85,00</b>	10,40	80,27	7,64
banana	87,51	1,03	89,55	1,14	<b>89,94</b>	1,16	89,51	0,84	68,53	2,76
bupa	61,08	6,88	66,00	7,80	<b>67,41</b>	7,96	62,44	7,90	56,46	4,37
chess	84,70	2,36	90,61	2,18	95,56	1,69	95,94	0,40	<b>96,09</b>	0,57
contraceptive	42,77	3,69	48,74	4,46	<b>50,17</b>	3,35	42,70	0,22	39,99	6,05
german	70,50	4,25	71,90	3,11	<b>72,10</b>	5,13	71,40	2,20	69,30	1,42
glass	73,61	11,91	71,98	9,47	73,64	8,86	76,42	13,21	<b>80,65</b>	12,04
haberman	66,97	5,46	71,53	6,38	73,18	2,61	<b>74,15</b>	5,07	63,34	8,42
hayes-roth	35,70	9,11	75,41	10,57	76,41	10,49	54,36	11,56	<b>80,20</b>	10,67
heart	77,04	8,89	82,22	8,25	<b>85,19</b>	8,11	81,48	6,42	78,15	9,72
iris	93,33	5,16	94,00	4,67	<b>94,67</b>	4,99	94,00	4,67	94,00	5,54
led7digit	40,20	9,48	71,40	4,90	<b>71,80</b>	4,77	10,80	3,12	63,20	5,53
lym	73,87	8,77	80,29	15,48	<b>81,76</b>	9,83	74,54	8,95	70,43	22,52
magic	80,59	0,90	82,31	0,65	83,24	0,96	<b>83,25</b>	0,68	76,68	5,46
mammographic	73,68	5,59	81,27	5,48	81,86	6,03	<b>82,62</b>	4,76	70,76	4,28
marketing	27,38	1,34	31,39	0,70	<b>31,90</b>	1,34	24,05	1,33	26,45	1,91
movement	<b>81,94</b>	4,34	74,72	4,88	74,44	5,24	71,11	3,12	11,11	3,03
newthyroid	97,23	2,26	<b>97,68</b>	2,32	96,73	3,64	93,48	2,95	97,25	4,33
nursery	82,67	0,92	85,38	1,09	<b>92,99</b>	0,76	82,67	0,88	78,94	32,09
pima	70,33	3,53	74,89	5,81	73,23	5,43	<b>75,53</b>	5,85	70,32	5,65
segment	96,62	0,70	95,15	1,28	<b>95,93</b>	0,62	82,86	2,47	95,71	1,26
sonar	<b>85,55</b>	7,51	79,29	10,23	80,29	5,85	82,10	6,19	83,57	3,47
spectfheart	69,70	6,55	79,02	7,31	<b>79,68</b>	10,73	76,01	10,12	78,30	11,92
splice	74,95	1,15	78,37	4,44	<b>82,51</b>	4,80	71,72	1,72	78,24	1,30
tae	40,50	8,43	56,54	15,86	<b>58,38</b>	11,80	30,54	2,56	49,12	3,77
texture	<b>99,05</b>	0,41	96,80	0,78	96,93	0,71	96,18	0,47	99,02	0,47
thyroid	92,58	0,81	94,58	0,55	<b>96,93</b>	2,39	95,87	0,61	92,57	0,26
wisconsin	69,65	5,09	75,05	5,27	<b>76,30</b>	4,97	70,02	4,76	68,45	7,06
zoo	92,81	6,57	95,33	6,49	<b>95,83</b>	9,72	66,25	8,07	96,83	2,78
<b>Promedio</b>	69,68	5,16	75,15	5,31	<b>76,33</b>	5,10	70,46	4,78	68,52	7,19

TABLA IV  
TEST ESTADÍSTICO NO PARAMÉTRICO

Método	Rangos	<i>p</i> -valor de Holm
SSMA-EDGPFW	1,8000	—
SSMA-EDGP	2,6000	0,0478
TS/KNN	3,2500	0,0007
Relieff	3,6667	0,0000
1-NN	3,6833	0,0000
<i>P</i> -valor de Friedman= 0.0000003		

vecino más cercano. De esta forma el modelo híbrido propuesto supera a los métodos de forma aislada. Los resultados obtenidos muestran que esta combinación mejora significativamente al modelo original y a otras propuestas anteriores.

#### AGRADECIMIENTOS

Subvencionado por los proyectos TIN2011-28488 y TIC-2010-6858. I. Triguero y J. Derrac disfrutaron de una beca FPU del Ministerio de Educación y Ciencia.

#### REFERENCIAS

- [1] Cover T.M., Hart P.E, *Nearest neighbor pattern classification*. IEEE Transactions on Information Theory 13,

21–27, 1967.

- [2] Aha D.W, Kibler D., Albert, M.K, *Instance-Based Learning Algorithms*. Machine Learning 1 (6), 37–66, 1991.
- [3] Pyle D., *Data Preparation for Data Mining*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, 1999.
- [4] Liu H., Motoda H. (Eds.), *Computational Methods of Feature Selection*. Chapman & Hall/Crc Data Mining and Knowledge Discovery Series, 2007.
- [5] S. García, J. Derrac, J.R. Cano, F. Herrera, *Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study*. IEEE Transactions on Pattern Analysis and Machine Intelligence, in press, doi: 10.1109/TPAMI.2011.142, 2011.
- [6] Triguero I, Derrac J, García S., Herrera F, *A Taxonomy and Experimental Study on Prototype Generation for Nearest Neighbor Classification*. IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews, doi: 10.1109/TSMCC.2010.2103939, in press, 2011.
- [7] D. Wettschereck, D. W. Aha, and T. Mohri, *A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms*. Artificial Intelligence Review 11, 273–314, 1997.
- [8] Fernández F., Isasi, P, *Evolutionary Design of Nearest Prototype Classifiers*. Journal of Heuristics 10 (4), 431–454, 2004.
- [9] Storn R., Price K. V, *A simple and efficient heuristic for global optimization over continuous spaces*. Journal of Global Optimization 11 (10), 341–359, 1997.
- [10] Ferrante N., Tirronen V, *Scale factor local search in differential evolution*. Memetic Computing 1 (2), 153–171, 2009.

- [11] Fernández F., Isasi, P, *Local feature weighting in a nearest prototype classification*. IEEE Transactions on Neural Networks 19 (1) 40–53, 2008.
- [12] Triguero I, García S., Herrera F, *Differential Evolution for Optimizing the Positioning of Prototypes in Nearest Neighbor Classification*. Pattern Recognition 44 (4), 901–916, 2011.
- [13] García S., Cano J.R., Herrera F, *A memetic algorithm for evolutionary prototype selection: A scaling up approach*. Pattern Recognition 41 (8), 2693–2709, 2008.
- [14] J. Zhang, A.C. Sanderson, *JADE: adaptive differential evolution with optional external archive*. IEEE Transactions on Evolutionary Computation 13 (5) 945–958, 2009.
- [15] S. Das, A. Abraham, U.K. Chakraborty, A. Konar, *Differential evolution using a neighborhood-based mutation operator*. IEEE Transactions on Evolutionary Computation 13 (3) 526–553, 2009.
- [16] García S., Herrera F, *An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons*. Journal of Machine Learning Research 9, 2677–2694, 2008.
- [17] García S., Fernández A., Luengo J., Herrera F, *Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental Analysis of Power*. Information Sciences 180, 2044–2064, 2010.
- [18] Kira K, Rendell L.A, *A practical approach to feature selection*. Proceedings of the Ninth International Conference on Machine Learning, Aberdeen, Scotland, 249–256, 1992.
- [19] Alcalá-Fdez J., Fernandez A., Luengo J., Derrac J., García S., Sánchez L., Herrera F.: KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing, in press (2010).
- [20] Kononenko I, *Estimating attributes: Analysis and extensions of RELIEF*. Proceedings of the 1994 European Conference on Machine Learning, Catania, Italy, 171–182, 1994.
- [21] Tahir M. A., Bouridane A., Kurugollu F, *Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier*. Pattern Recognition Letters 28 (4) 438–446, 2007.