Coevolución de selección de instancias y esquemas de pesos para clasificadores basados en la regla del vecino más cercano

Joaquín Derrac
Dept. CCIA - CITIC
Univ. de Granada
jderrac@decsai.ugr.es

Isaac Triguero
Dept. CCIA - CITIC
Univ. de Granada
triguero@decsai.ugr.es

Salvador García
Dept. Ciencias de la Comp.
Univ. de Jaén
sglopez@ujaen.es

Francisco Herrera
Dept. CCIA - CITIC
Univ. de Granada
herrera@decsai.ugr.es

Resumen

La regla del vecino más cercano es uno de los métodos más representativos en minería de datos. Debido a su popularidad, han aparecido numerosas propuestas para aumentar su rendimiento. Entre ellas, la selección de instancias destaca por su capacidad de mejorar simultáneamente tanto la eficacia del método como su eficiencia, al reducir considerablemente el conjunto de entrenamiento. También puede destacarse la definición de esquemas de pesos (para instancias o para características) en la función de distancia por su efectividad a la hora de mejorar la precisión del clasificador.

Esta propuesta presenta un nuevo algoritmo coevolutivo, como herramienta para hibridar ambas propuestas. Se compara su rendimiento con respecto al de las técnicas evolutivas empleadas por separado, así como con el clasificador básico. Los resultados obtenidos, contrastados mediante técnicas estadísticas, avalan la utilidad de la coevolución como estrategia para aunar esfuerzos entre diferentes técnicas de mejora de la regla del vecino más cercano.

1. Introducción

El clasificador de los k-vecinos más cercanos (k-NN) es uno de los algoritmos más conocidos en minería de datos. Su simplicidad, efectividad y precisión a la hora de abordar problemas de clasificación lo han resaltado como uno de los algoritmos más utilizados en clasificación, contando con gran atención e interés por parte de los investigadores[13].

La selección de instancias es una de las propuestas existentes para mejorar el rendimiento del clasificador k-NN [8]. Su aplicación permite reducir la complejidad espacial del clasificador y mejorar su eficiencia, gracias a la eliminación de ejemplos irrelevantes en el conjunto de entrenamiento, así como incrementar su precisión, mediante la eliminación de ruido.

Otra propuesta interesante es el empleo de esquemas de pesos para ajustar la función de distancia, aplicable tanto a las instancias [1] como a las características [11] del conjunto de entrenamiento. La obtención de un adecuado conjunto de pesos para ponderar la medida de distancia puede ayudar a amoldar el comportamiento del clasificador al dominio en que se emplee, afectando favorablemente a su capacidad de generalización.

En los últimos años han aparecido una gran cantidad de propuestas evolutivas aplicadas a problemas de minería de datos [7]. Gracias a la posibilidad de definir los problemas de la selección de instancias y la obtención de pesos para instancias y características como problemas de búsqueda, estos también pueden ser abordados mediante algoritmos evolutivos, ofreciendo buenos resultados [2].

Recientemente, la aplicación conjunta de varias de estas técnicas sobre un mismo clasificador ha comenzado a ser abordada, mediante el empleo de algoritmos coevolutivos [3]. La coevolución [9] ofrece un marco de trabajo en el que varias técnicas de optimización pueden ser aplicadas de forma simultánea, ofreciendo resultados superiores a los esperados a partir de su aplicación aislada.

En esta contribución presentamos un modelo de Coevolución de Selección de Instancias y Pesos, aplicada al clasificador k-NN (CSIP-NN). El modelo propuesto se compone de 3 poblaciones, cada una dedicada a una tarea de mejora del clasificador (selección de instancias, búsqueda de pesos para instancias y búsqueda de pesos para características). Tras su descripción, presentamos un estudio experimental en el que mostramos la obtención de mejoras significativas respecto a la aplicación de las técnicas evolutivas básicas de forma aislada. Estas mejoras son contrastadas mediante el empleo de test estadísticos no paramétricos, altamente recomendados para el análisis de resultados en problemas de minería de datos.

El resto del trabajo se estructura como sigue: La Sección 2 presenta algunos conceptos preliminares sobre las técnicas empleadas. La Sección 3 describe el modelo propuesto. La Sección 4 muestra el estudio experimental realizado para comparar el rendimiento de CSIP-NN con varias técnicas no coevolutivas. Finalmente, en la Sección 5 se muestran las conclusiones generales del trabajo.

2. Conceptos preliminares

Esta sección repasa algunos conceptos preliminares necesarios. La Sección 2.1 presenta la coevolución y algunas de sus características más interesantes. La Sección 2.2 describe el uso de las técnicas de selección de instancias en clasificación. Finalmente, la Sección 2.3 comenta el empleo de esquemas de pesos para mejorar la precisión de los clasificadores.

2.1. Coevolución

La coevolución es el área de la computación evolutiva relativa a técnicas capaces de gestionar varias poblaciones simultáneamente. Su aplicación consiste en fraccionar el dominio del problema, empleando una estrategia **divide y vencerás** en la que cada población se especializa en resolver una parte.

Dentro de este área, la coevolución cooperativa [9] define elementos de cooperación entre las diferentes poblaciones. Generalmente, esto se consigue empleando funciones de evaluación conjuntas, que requieren de un individuo de cada población para realizar la evaluación. Esto permite beneficiar a aquellos individuos que funcionen especialmente bien en colaboración con el resto de poblaciones, en contraposición a las funciones de evaluación clásicas, dirigidas tan solo a evaluar la calidad de los individuos respecto a un objetivo individual.

La principal motivación de su uso reside, por tanto, en esta capacidad de descomposición, la cual ha demostrado tener la capacidad de sobrepasar la conocida barrera del **No Free Lunch**, presente en la mayoría de problemas de optimización [12].

2.2. Selección de instancias

La selección de instancias [8] tiene como objetivo obtener el menor subconjunto posible de entrenamiento que permita a un algoritmo de minería de datos operar con la misma 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.

Su definición es la siguiente: Sea $\mathbf{X}$ una instancia donde $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_M}, \mathbf{x_c})$, con $\mathbf{X}$ perteneciendo a la clase $\mathbf{c}$, dada por $\mathbf{X_c}$, y un espacio M-dimensional donde $\mathbf{x_i}$ es el i-ésimo valor de la instancia $\mathbf{X}$. Asumamos que existe un conjunto de entrenamiento **CE** compuesto por **N** instancias, y un conjunto de test **CT** compuesto por **T** instancias. Sea $\mathbf{CR} \subseteq \mathbf{CE}$ el subconjunto resultante de la ejecución de un método de selección de instancias. Toda nueva instancia **T** perteneciente a **CT** será clasificada por un algoritmo de minería de datos empleando **CR** como conjunto de entrenamiento.

2.3. Esquemas de pesos

El empleo de pesos es otra técnica interesante que puede emplearse para mejorar el rendimiento de los clasificadores. Aunque existen múltiples propuestas para ello, en este trabajo nos centraremos en aquellas que se emplean para modificar la función de distancia definida por el clasificador.

Así, es posible definir tanto pesos asociados a las características (es decir, valores reales que ponderen la importancia de cada variable de cara al cómputo de la similaridad entre dos instancias) como asociados a las instancias (es decir, valores reales que modulen la distancia efectiva entre dos instancias dependiendo de alguna propiedad asociada, p. ej., la clase a la que pertenezcan). Ambas vertientes han sido ampliamente estudiadas en la literatura [11, 1].

El objetivo final de la inclusión de ambos tipos de esquemas de pesos es incrementar en lo posible la precisión del clasificador base empleado. Por tanto, la mayoría de estos métodos realizan un proceso de optimización para encontrar los mejores esquemas posibles, partiendo del conjunto de entrenamiento original.

3. Propuesta

En esta sección se presenta el modelo coevolutivo CSIP-NN. Concretamente, la Sección 3.1 describe los subcomponentes que forman el modelo. La Sección 3.2 describe la función de evaluación empleada. Finalmente, la Sección 3.3 plantea el modelo general.

3.1. Subcomponentes empleados

El funcionamiento de CSIP-NN se basa en buscar, simultáneamente, el mejor subconjunto posible de entrenamiento y los mejores esquemas de pesos posibles tanto para instancias como para características. Para ello, emplea tres poblaciones, cada una dirigida a realizar una tarea de optimización concreta:

- Población de Selección de Instancias (SI): Trata de encontrar el mejor subconjunto de instancias posible.
- Población de Pesos de Instancias (PI): Busca el mejor esquema posible de pesos para instancias.
- Población de Pesos de Características (PC): Busca el mejor esquema posible de pesos para características.

Aunque las tres poblaciones realizan una tarea de búsqueda similar, guiadas por un algoritmo genético, existe una serie de características que pueden ser empleadas para diferenciarlas. La Tabla 1 las resume:

Población	SI	PI	PC
Ámbito	Instancias	Instancias	Carac.
Codificación	Binaria	Real	Real
Granularidad	Individual	Clase	Individual
Época	Simple	Multiple	Multiple
Objective	Acier. + Red.	Acierto	Acierto

Tabla 1: Caracteristicas de las poblaciones

- **Ámbito**: Cada población está dirigida a una dimensión concreta: Instancias o características.
- **Codificación**: La codificación de los cromosomas puede ser binaria ($(0, 1)$) o real ($[0, 1]$) dependiendo de si se están seleccionando elementos (0 = no seleccionado, 1 = seleccionado) o si se están definiendo pesos, respectivamente
- **Granularidad**: Cada elemento del cromosoma puede referirse a una única instancia/característica (Simple) o a todas las instancias de una clase (Clase).
- **Época**: Para cada población es posible especificar una duración de época (numero de generaciones a completar antes de pasar a la siguiente población). Esta duración puede ser de una generación (simple) o de varias (múltiple).
- **Objetivo**: El objetivo común de las poblaciones es incrementar el acierto obtenido por el clasificador. Además, otro objetivo puede ser obtener la mayor reducción posible del conjunto de entrenamiento.

Para la población de selección de instancias, se ha empleado el algoritmo CHC [4], partiendo de la configuración mostrada en [2], donde fue seleccionado como un algoritmo eficaz para esta tarea. Para mejorar su capacidad de reducción, se ha partido de una inicialización sesgada, en la que solo **prob**1 instancias del cromosoma comenzaban seleccionadas, y se ha modificado el operador de cruce HUX original para que solamente mantenga **prob**0**to**1 instancias seleccionadas de las que originalmente aparecerían al aplicar el cruce.

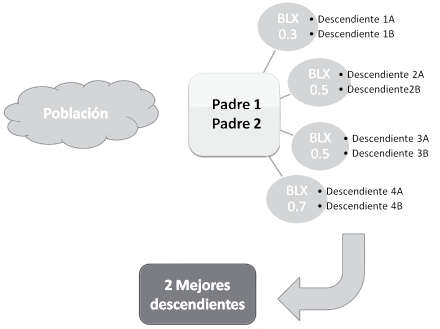

Figura 1: Cruce con múltiples descendientes

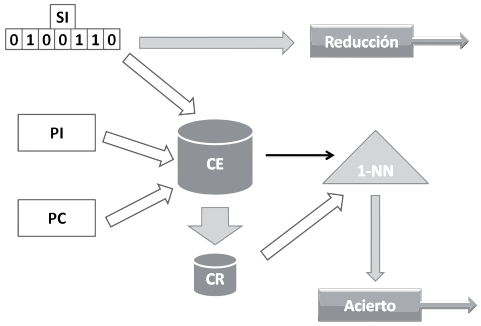

Figura 2: Función de evaluación

Para las poblaciones de pesos, se ha aplicado un algoritmo genético estacionario con codificación real. Se ha empleado un operador de cruce de múltiples descendientes [10] por su gran capacidad de convergencia. De los modelos sugeridos en [10], los mejores resultados se han obtenido con el operador 2BLX0.3-4BLX0.5-2BLX0.7 , basado en el operador BLX-α. La Figura 1 muestra un esquema de su aplicación, donde se realizan 4 cruces con diferentes valores del parámetro alpha, y se toman los dos mejores descendientes encontrados. Como operador de mutación se ha empleado el operador no uniforme, siguiendo las recomendaciones dadas en [10].

3.2. Función de evaluación

La función de evaluación de CSIP-NN depende de dos componentes:

- **Acierto**: Precisión del clasificador base (1-NN) sobre el conjunto de entrenamiento (es decir, se mide el error leave-one-out de clasificación con la configuración de instancias y pesos evaluada).
- **Reducción**: Porcentaje de reducción del conjunto de entrenamiento seleccionado, respecto del conjunto inicial.

Para realizar una evaluación, son necesarios un cromosoma por cada población. Si definimos **H** como un cromosoma de lo población SI, **I** como un cromosoma de lo población PI, y **J** como un cromosoma de lo población PC, el valor asignado a cada cromosoma es el siguiente:

$$\begin{aligned} \mathbf{Fitness}(\mathbf{H}) &= \alpha \cdot \mathbf{Ac}(\mathbf{H},\mathbf{I},\mathbf{J}) \\ &\quad + (1-\alpha) \cdot \mathbf{Red}(\mathbf{H}) \\ \mathbf{Fitness}(\mathbf{I}) &= \mathbf{Ac}(\mathbf{H},\mathbf{I},\mathbf{J}) \\ \mathbf{Fitness}(\mathbf{J}) &= \mathbf{Ac}(\mathbf{H},\mathbf{I},\mathbf{J}) \end{aligned} \quad (1)$$

Donde $\mathbf{Ac}(\mathbf{H},\mathbf{I},\mathbf{J})$ es el acierto estimado del clasificador, $\mathbf{Red}(\mathbf{H})$ es la reducción obtenida por el cromosoma **H**, y α es un valor [0,1] para ponderar ambos objetivos (usualmente, $\alpha = 0{,}5$, siguiendo las recomendaciones de [2]).

La Figura 2 muestra un esquema del proceso de evaluación. Para estimar el acierto, se preprocesa el conjunto de entrenamiento, CE, obteniendo las instancias indicadas en el cromosoma de la población SI, y se les asignan los pesos indicados por los cromosomas de las poblaciones PI y PC. Como resultado, se obtiene el conjunto reducido (CR). Dicho conjunto se emplea como conjunto de entrenamiento de un clasificador 1-NN y se estima el acierto de clasificación sobre el conjunto de entrenamiento original, $\mathbf{Ac}(\mathbf{H},\mathbf{I},\mathbf{J})$.

Como se ha comentado, el clasificador empleado para estimar el acierto es un 1-NN. Este clasificador emplea una versión modificada de la distancia Euclídea,(Ecuación 2):

$$\mathbf{D}(\mathbf{x},\mathbf{y}) = \mathbf{PI}_{\mathbf{c(y)}} \sum_{i=0}^{M} \mathbf{PC}_i \cdot \sqrt{(\mathbf{x}_i - \mathbf{y}_i)^2} \quad (2)$$

Donde **x** es la instancia a clasificar, **y** es una instancia del conjunto reducido, $\mathbf{PI}_{\mathbf{c(y)}}$ denota

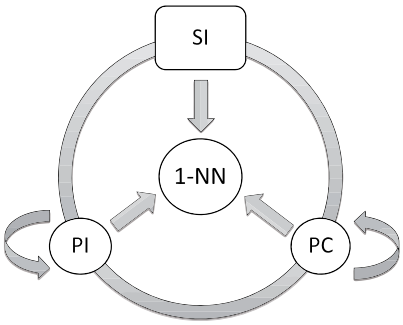

Figura 3: Esquema de poblaciones de CSIP-NN

el peso asociado a la clase a la que pertenece la instancia **y**, y $\mathbf{PC_i}$ denota el peso asociado a la característica **i**.

3.3. Modelo coevolutivo

CSIP-NN agrupa los componentes descritos en las secciones anteriores en un único modelo. Las tres poblaciones evolucionan cíclicamente, consumiendo cada población una época (un número fijo de generaciones/evaluaciones) antes de pasar a la siguiente.

La Figura 3 representa el esquema completo: El ciclo evolutivo es iniciado por la población SI, realizando una generación (época simple). A su fin, comienza el turno de la población PI, que realiza varias generaciones hasta agotar la duración de época especificada (época múltiple). Tras ella, la población PC evoluciona durante otra época múltiple, dando fin a un ciclo del modelo completo.

El Algoritmo 1 resume el esquema general del modelo. Al final de cada ciclo, los mejores individuos de cada población son seleccionados. Su cometido será apoyar las evaluaciones de los nuevos individuos generados durante la búsqueda. De este modo, a la hora de evaluar un nuevo individuo de cualquiera de las poblaciones, se emplean los mejores individuos encontrados por las otras 2 poblaciones, completando así los 3 individuos requeridos por la función de evaluación.

Esta configuración es óptima para modelar la cooperación entre las diferentes poblaciones. La evaluación conjunta de cada individuo junto a los mejores representantes del resto de poblaciones permite guiar la búsqueda hacia zonas más prometedoras, que representen las propiedades más deseables de cada una de las técnicas. El empleo del modelo de épocas y de la función de evaluación común permiten controlar la progresión de la búsqueda en cada componente, impidiendo así que alguna de las poblaciones progrese demasiado en solitario (alcanzando soluciones óptimas desde el punto de vista individual, pero subóptimas desde el punto de vista cooperativo).

```
Generar poblaciones SI, PI y PC;
Seleccionar mejores individuos de cada población;
Mientras queden evaluaciones:
     Realizar época en la población SI;
     Realizar época en la población PI;
     Realizar época en la población PC;
     Actualizar mejores individuos;
Fin
Devolver mejores individuos de cada población;
```

Algoritmo 1: Esquema general de CSIP-NN

4. Estudio experimental

Para tratar de caracterizar el rendimiento de CSIP-NN, se ha realizado un estudio experimental sobre diferentes problemas de clasificación. La Sección 4.1 describe los conjuntos de datos utilizados. La Sección 4.2 enumera los algoritmos de comparación considerados y describe sus parámetros. La Sección 4.3 presenta y analiza los resultados obtenidos. Finalmente, la Sección 4.4 muestra el estudio estadístico realizado para contrastar los resultados.

4.1. Conjuntos de datos

En éste estudio, se han empleado 30 conjuntos de datos representando problemas de clasificación. Han sido tomados de los repositorios **UCI Machine Learning Repository** [1] y **KEEL-dataset Repository** [2]. La Tabla 2 describe sus principales características: Nombre, número de instancias, número de atributos (características) y número de clases.

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html

[2] http://www.keel.es/datasets.php

Conjunto	# In.	# At.	# Cl.
Australian	690	14	2
Balance	625	4	3
Bands	539	19	2
Breast	286	9	2
Bupa	345	6	2
Car	1728	6	4
Cleveland	303	13	5
Contraceptive	1473	9	3
Dermatology	366	34	6
German	1000	20	2
Glass	214	9	7
Hayes-roth	160	4	3
Housevotes	435	16	2
Iris	150	4	3
Lymphography	148	18	4
Monk-2	432	6	2
Movement	360	90	15
New Thyroid	215	5	3
Pima	768	8	2
Saheart	462	9	2
Sonar	208	60	2
Spectfheart	267	44	2
Tae	151	5	3
Tic-tac-toe	958	9	2
Vehicle	846	18	4
Vowel	990	13	11
Wine	178	13	3
Wisconsin	699	9	2
Yeast	1484	8	10
Zoo	101	16	7

Tabla 2: Conjuntos de datos utilizados

Todos los conjuntos han sido particionados, empleando la técnica de validación cruzada de 10 campos. 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.

4.2. Algoritmos y parámetros

Además de CSIP-NN, se han empleado como algoritmos de comparación los 3 métodos empleados en cada una de las poblaciones, por separado: El algoritmo CHC para Selección de Instancias (CHC-SI), un Algoritmo Genético Estacionario con múltiples descendientes para Pesos en Características (AGE-PC) y un Algoritmo Genético Estacionario con múltiples descendientes para Pesos en Instancias (AGE-PI). Además, se ha considerado también el clasificador 1-NN como referencia básica.

La experimentación se ha realizado con el apoyo de la plataforma KEEL [3], en el lenguaje de programación JAVA. La Tabla 3 muestra los parámetros empleados.

[3] http://www.keel.es

Método	Parametros
CSIP-NN	α: 0.5, prob0to1: 0.25, prob1: 0.25, Prob. mutación: 0.05 por cromosoma Duración de épocas: 40 evaluaciones
CHC-SI	α: 0.5, prob0to1: 0.25, prob1: 0.25
AGE-PC	Prob. mutación: 0.05 por cromosoma
AGE-PI	Prob. mutación: 0.05 por cromosoma
Comunes	Evaluaciones: 10000, Tam. poblac.: 50, Op. cruce: 2BLX0.3-4BLX0.5-2BLX0.7, Op. cruce (binario): HUX modificado Clasificador base: 1-NN

Tabla 3: Parámetros empleados

4.3. Resultados obtenidos

En el estudio experimental, se han tenido en cuenta como medidas de rendimiento el acierto en la fase de test (acierto a la hora de clasificar ejemplos no contemplados en la fase de entrenamiento del modelo) y la reducción obtenida sobre el conjunto de entrenamiento, para aquellos métodos que han realizado reducción (CSIP-NN y CHC-SI).

La Tabla 4 muestra los resultados obtenidos. Para cada conjunto se muestra el valor medio obtenido en cada conjunto por cada algoritmo, y medida de rendimiento. Además, se ha remarcado en negrita el mejor resultado en acierto obtenido en cada conjunto.

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

- El método propuesto, CSIP-NN, obtiene mejor acierto medio que el resto. Además, obtiene el mejor resultado en 18 de los 30 problemas considerados.
- Todos los métodos mejoran considerablemente el acierto de 1-NN.
- Tanto CSIP-NN como CHC-SI reducen ampliamente el tamaño del conjunto de datos original, sin que esto afecte negativamente a su precisión.

Estos resultados ponen de manifiesto la utilidad de las técnicas de selección de instancias y de los esquemas de pesos a la hora de mejorar el rendimiento del clasificador 1-NN. Esta mejora es especialmente apreciable en el caso del modelo coevolutivo, el cual aporta, simultáneamente, una precisión superior a la obtenida mediante el resto de propuestas por se-

Medida	Acierto (%)					Reducción (%)	
Conjunto	CSIP-NN	CHC-SI	AGE-PC	AGE-PI	1-NN	CSIP-NN	CHC-SI
Australian	81.74	81.45	81.01	80.87	81.45	93.66	97.67
Balance	85.75	79.04	73.76	80.33	79.04	94.24	96.62
Bands	75.52	74.04	72.75	72.92	74.04	95.49	97.28
Breast	70.62	66.04	63.06	69.98	65.35	97.86	97.71
Bupa	60.95	62.51	62.91	62.29	61.08	95.36	96.55
Car	95.89	85.65	94.91	86.34	85.65	83.78	95.87
Cleveland	56.43	53.14	52.48	56.45	53.14	97.14	98.13
Contraceptive	45.22	42.63	44.06	44.61	42.77	84.36	97.04
Dermatology	96.72	95.35	96.45	94.26	95.35	96.02	96.45
German	72.10	70.50	69.50	71.90	70.50	89.13	97.99
Glass	75.72	74.50	72.36	69.35	73.61	93.25	93.51
Hayes-roth	72.15	71.01	69.96	73.03	35.70	91.92	92.34
Housevotes	94.93	91.24	93.78	91.23	91.24	97.80	98.24
Iris	93.33	93.33	94.00	94.00	93.33	96.37	95.93
Lymphography	79.30	73.87	76.54	77.34	73.87	94.23	94.67
Monk-2	100.00	95.32	100.00	75.09	77.91	93.29	95.40
Movement	83.06	86.39	86.67	88.06	81.94	74.69	88.09
New Thyroid	95.82	97.23	96.28	95.84	97.23	96.95	97.62
Pima	71.24	70.33	70.71	70.59	70.33	92.09	97.09
Saheart	65.37	64.49	64.06	64.28	64.49	96.34	97.88
Sonar	87.00	85.55	85.07	86.02	85.55	91.67	93.11
Spectfheart	77.92	69.70	74.63	78.68	69.70	98.17	97.96
Tae	65.71	65.04	68.38	63.04	40.50	93.82	94.41
Tic-tac-toe	87.37	82.07	91.33	73.07	73.07	88.67	95.62
Vehicle	71.28	70.10	71.16	66.55	70.10	90.28	94.48
Vowel	98.28	99.39	99.29	98.38	99.39	74.97	84.01
Wine	97.16	95.52	96.63	97.75	95.52	96.88	96.69
Wisconsin	96.00	95.57	95.57	96.42	95.57	94.74	99.21
Yeast	52.76	52.23	50.81	52.63	50.47	83.49	97.19
Zoo	97.50	96.83	96.83	95.58	92.81	89.99	89.34
Average	80.09	78.00	78.83	77.56	74.69	91.89	95.47

Tabla 4: Resultados obtenidos

parado, y una reducción considerable del conjunto de entrenamiento (que conlleva una mayor eficiencia del clasificador, tanto en términos de espacio de almacenamiento como en tiempo de ejecución).

4.4. Estudio estadístico

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 [6].

Específicamente, se ha seleccionado el test de Friedman como método para detectar la existencia de diferencias significativas entre los resultados de acierto, y los métodos de Holm, Hochberg y Finner como test **post-hoc** para caracterizar las diferencias encontradas [5] [4].

[4] Para más información, puede consultarse el sitio web temático de SCI2S sobre **Statistical Inference in Computational Intelligence and Data Mining** `http://sci2s.ugr.es/sicidm/`

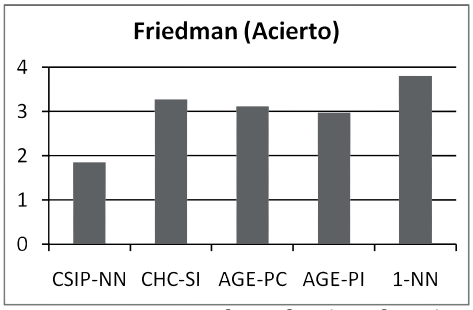

Figura 4: Rangos obtenidos (Friedman)

El test de Friedman detectó diferencias significativas (**p** = 0**,**00006) sobre los resultados. A partir de los rangos obtenidos (Figura 4), se seleccionó a CSIP-NN como método de control (aquel que obtuvo el rango más bajo), y se aplicaron los 3 métodos **post-hoc**.

Método de control: CSIP-NN (Rango: 1.850)				
Método	Rangos	Holm	Hochberg	Finner
CHC-SI	3.267	0.00156	0.00156	0.00104
AGE-PC	3.117	0.00384	0.00384	0.00256
AGE-PI	2.967	0.00623	0.00623	0.00623
1-NN	3.800	0.00001	0.00001	0.00001

Tabla 5: Estudio estadístico no paramétrico

La Tabla 5 resume los resultados del estudio. CSIP-NN mejora estadísticamente los resultados obtenidos por el resto, con un nivel de significancia $\alpha = 0{,}01$ (los tres métodos **post-hoc** obtienen p-valores inferiores a 0.01 en todos los casos). Por tanto, contrastan la afirmación de que la mejora obtenida en acierto por CSIP-NN sobre el resto de métodos es significativa.

5. Conclusión y trabajo futuro

En este trabajo se ha presentado una propuesta de hibridación de varios métodos de preprocesamiento y ajuste para k-NN mediante algoritmos coevolutivos. Los resultados obtenidos muestran que el empleo de la coevolución permite obtener resultados muy positivos, que mejoran significativamente aquellos conseguidos de forma aislada.

Como trabajo futuro, se plantean varias vertientes, que incluyen la comparación del modelo con técnicas no evolutivas de selección de instancias y de pesos y la evaluación del rendimiento en conjuntos de mayor tamaño. Además se plantea el desarrollo de nuevas funciones de evaluación, más eficientes, para mejorar el coste computacional del método.

Referencias

[1] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," **Artificial Intelligence Review**, vol. 11, pp. 11–73, 1997.

[2] J. R. Cano, F. Herrera, and M. Lozano, "Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study," **IEEE Transactions on Evolutionary Computation**, vol. 7, no. 6, pp. 561–575, 2003.

[3] J. Derrac, S. García, and F. Herrera, "IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule," **Pattern Recognition**, vol. 43, no. 6, pp. 2082–2105, 2010.

[4] L. J. Eshelman, "The CHC adaptative search algorithm: How to have safe search when engaging in nontraditional genetic recombination," in **Foundations of Genetic Algorithms**, G. J. E. Rawlins, Ed. San Mateo, California.: Morgan Kaufmann, 1991, pp. 265–283.

[5] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," **Information Sciences**, vol. 180, pp. 2044–2064, 2010.

[6] S. García, D. Molina, M. Lozano, and F. Herrera, "A Study on the Use of Non-Parametric Tests for Analyzing the Evolutionary Algorithms' Behaviour: A Case Study on the CEC'2005 Special Session on Real Parameter Optimization," **Journal of Heuristics**, vol. 15, pp. 617–644, 2009.

[7] A. Ghosh and L. C. Jain, Eds., **Evolutionary Computation in Data Mining**. Springer-Verlag, 2005.

[8] H. Liu and H. Motoda, Eds., **Instance Selection and Construction for Data Mining**, ser. The Springer International Series in Engineering and Computer Science. Springer, 2001.

[9] M. A. Potter and K. A. D. Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," **Evolutionary Computation**, vol. 8, no. 1, pp. 1–29, 2000.

[10] A. M. Sánchez, M. Lozano, P. Villar, and F. Herrera, "Hybrid crossover operators with multiple descendents for real-coded genetic algorithms: Combining neighborhood-based crossover operators," **International Journal on Intelligent Systems**, vol. 24, no. 5, pp. 540–567, 2009.

[11] D. Wettschereck, D. W. Aha, and T. Mohri, "A review and empirical evaluation of feature weigthing methods for a class of lazy learning algorithms," **Artificial Intelligence Review**, vol. 11, pp. 273–314, 1997.

[12] D. H. Wolpert and W. G. Macready, "Coevolutionary free lunches," **IEEE Transactions on Evolutionary Computation**, vol. 9, no. 6, pp. 721–735, 2005.

[13] X. Wu and V. Kumar, Eds., **The Top Ten Algorithms in Data Mining**, ser. Data Mining and Knowledge Discovery. Chapman & Hall/CRC, 2009.