RNPS No. 2142
ISSN 2072-6287
Versión Digital

SERIE AZUL

REPORTE TÉCNICO
Reconocimiento de Patrones

Selección combinada de rasgos y objetos para el mejoramiento de clasificadores NN

MSc. Yenny Villuendas-Rey,
MSc. Milton García-Borroto,
Dr. C. José Ruiz-Shulcloper

RT_022 **enero 2010**

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
Ciudad de La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

RNPS No. 2142
ISSN 2072-6287
Versión Digital

SERIE AZUL

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

REPORTE TÉCNICO
Reconocimiento de Patrones

Selección combinada de rasgos y objetos para el mejoramiento de clasificadores NN

MSc. Yenny Villuendas-Rey,
MSc. Milton García-Borroto,
Dr. C. José Ruiz-Shulcloper

RT_022 **enero 2010**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
Ciudad de La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

Selección combinada de rasgos y objetos para el mejoramiento de clasificadores NN

MSc. Yenny Villuendas-Rey, MSc. Milton García-Borroto, Dr. C. José Ruiz-Shulcloper

Facultad de Informática, Universidad de Ciego de Ávila, Carretera a Morón km 9 ½, Ciego de Ávila, Cuba
yennyv@bioplantas.cu

RT_022 CENATAV
Fecha del camera ready: 30 de octubre de 2009

Resumen: Los clasificadores de la familia del vecino más cercanos constituyen una de las técnicas de clasificación supervisada no paramétrica más populares, debido a su simpleza conceptual y a sus excelentes resultados en la práctica. Sin embargo, el elevado costo computacional, tanto de almacenamiento como de clasificación, y la sensibilidad al ruido, constituyen los principales limitantes de estos clasificadores. En este reporte se realiza un análisis de las técnicas de selección de rasgos y objetos de forma simultánea o combinada, para el mejoramiento de los clasificadores de esta familia.

Palabras clave: selección de rasgos y objetos, vecino más cercano, clasificación supervisada

Abstract: Nearest Neighbor classifiers are one of the most popular non parametric techniques for supervised classification, due to their conceptual simplicity and high performance in real applications. However, the computational cost, for both storage and classification, and their sensitivity to noisy, are the main drawbacks of these classifiers. In this Technical Report we review the techniques for combined or simultaneous objects and features selection for the enhancement of Nearest Neighbor classifiers.

Keyword: Objects and Features Selection, Nearest Neighbor, Supervised Classification

1 Introducción

La clasificación supervisada es una de las áreas más importantes dentro del Reconocimiento de Patrones. En ella, generalmente se tiene un conjunto de descripciones de objetos que pertenecen a determinadas clases, estas descripciones de objetos constituyen la muestra o matriz de entrenamiento. El objetivo de la clasificación supervisada es, dado la descripción de un nuevo objeto, estimar a cuál(es) clase(s) pertenece. De manera general los clasificadores supervisados pueden agruparse en dos categorías: paramétricos y no paramétricos [1]. Los clasificadores paramétricos necesitan conocimiento acerca de las distribuciones de los datos, mientras que los no paramétricos no utilizan ninguna información de esta naturaleza, lo que los hace más populares en muchos casos.

Entre los clasificadores supervisados no paramétricos más utilizados, se encuentra el clasificador (regla) del Vecino Más Cercano (*Nearest Neighbor*) NN, por sus siglas en inglés [2]. Este clasificador, en su versión original, para clasificar un nuevo objeto, compara su descripción con la de todos los objetos de la matriz de entrenamiento y luego le asigna la clase del objeto cuya descripción sea más cercana. Este clasificador fue extendido a la regla de los k Vecinos

Más Cercanos (k-NN) donde se buscan las k descripciones más cercanas y se asigna al objeto que se desea clasificar la clase mayoritaria[1] entre todos los vecinos más cercanos. En caso de empate, se asigna la clase de forma aleatoria. Numerosas variantes para la determinación de los vecinos más cercanos han sido propuestas, pero la idea general se ha mantenido [3]. Es por esto que en la literatura generalmente se hace referencia a la familia NN, incluyendo así todas las variantes.

Entre las ventajas de los clasificadores de la familia NN se pueden mencionar las siguientes:

1. No utiliza conocimiento de las distribuciones de los datos para la clasificación. Así, es posible estimar las clases de los objetos sin tener ningún conocimiento probabilístico de los mismos, logrando gran flexibilidad y ampliando su área de aplicación.
2. Es muy simple e intuitivo. Su estrategia de clasificación resulta muy fácil de comprender, y su sencillez hace que sea uno de los clasificadores más populares dentro del Reconocimiento de Patrones y la Inteligencia Artificial.
3. Permite explicar su comportamiento. A diferencia de otros clasificadores supervisados como las Redes Neuronales Artificiales, los clasificadores NN explican la asignación de una clase devolviendo los vecinos más cercanos del objeto clasificado.
4. El error asintótico es menor que el doble del error de Bayes [4]. Esta propiedad ha sido demostrada con funciones de distancia y hace que su superioridad con respecto a otros clasificadores más complejos no sea cuestionada en muchos casos.

Sin embargo, esta familia no está exenta de desventajas, entre ellas:

1. Costo computacional que se incrementa con las dimensiones de la matriz de entrenamiento. En este caso, los clasificadores NN tienen asociados dos costos: el costo de almacenamiento y el costo de clasificación, que es lineal con respecto a la cantidad de objetos almacenados en la matriz de entrenamiento.
2. Deterioro de la eficacia en presencia de ruido. Debido a su estrategia de funcionamiento, la presencia de objetos ruidosos en la matriz de entrenamiento hacen que todos los objetos cercanos a un objeto ruidoso tiendan a ser mal clasificados.
3. Asume, en su versión original, que los objetos están representados en espacios numéricos, por lo que no admite datos mezclados ni ausencia de información. Se han desarrollado versiones, como la Regla del Vecino Más Similar, que eliminan esta restricción.

Para la solución de los problemas antes mencionados de la familia NN, desde la década del 60 se han reportado diversas estrategias, con enfoques diferentes:

1. Seleccionar rasgos [5-7]. Estos métodos hallan subconjuntos de rasgos relevantes y eliminan los irrelevantes. Aquí las variaciones están en el concepto de relevancia y en la forma de calcularlos.
2. Seleccionar objetos [8-13]. Estos métodos se dividen en dos categorías fundamentales:
    a. Métodos de condensación. Tratan de eliminar la mayor cantidad de objetos posibles, sin degradar significativamente la eficacia del clasificador.

[1] Esto se debe a la imposición de clasificar en sólo una clase. Soluciones alternativas que en muchos casos puede ser de mucha utilidad es la abstención o la multiclasificación.

b. Métodos de edición basada en el error (o simplemente edición). Tratan de eliminar objetos ruidosos.

3. Aplicar de forma secuencial de métodos de selección de objetos y de rasgos y viceversa.
4. Seleccionar de forma simultánea o combinada rasgos y objetos.

Se considera que esta última estrategia puede obtener resultados superiores, pues permite en un mismo algoritmo la selección tanto de rasgos como de objetos, y este proceso puede utilizar toda la información contenida en la matriz de entrenamiento, a diferencia de la aplicación secuencial de métodos de selección de objetos y de métodos de selección de rasgos o viceversa. En estos casos, el método que se aplique primero tendrá acceso a toda la información, pero el segundo sólo a los resultados del primer método (ver Figura 1). Experimentos realizados por Kuncheva y Jain [14] apoyan estas consideraciones.

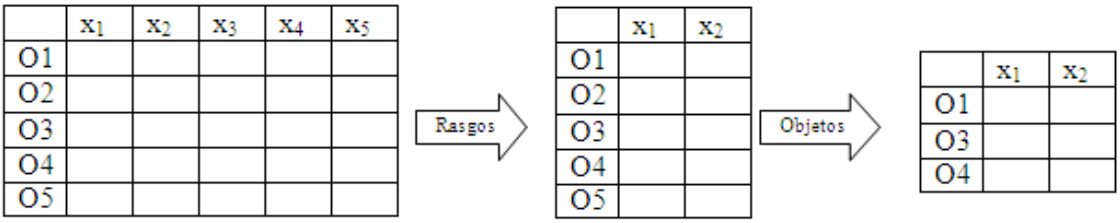


Fig. 1. Nótese que el método de selección de objetos, sólo tiene acceso a los resultados obtenidos por el método de selección de rasgos, no a la matriz original

Por otra parte, los datos en muchas aplicaciones criminalísticas, biológicas, médicas y de las geociencias, entre otras, pueden estar descritos en términos de rasgos de diferente naturaleza (Booleanos, k-valentes, ordinales, numéricos, etc.). Además, en algunos casos el valor de uno o varios rasgos puede ser desconocido. A este tipo de problemas se le conoce como problemas con datos mezclados e incompletos (MID, por sus siglas en inglés).

La selección simultánea o combinada de objetos y rasgos data de apenas una década, sin embargo, se han realizado numerosas propuestas en este sentido. En este reporte pretendemos realizar un análisis de las principales propuestas realizadas, a la luz de su aplicación a problemas con datos mezclados e incompletos.

El trabajo está estructurado de la siguiente manera: a continuación, se detalla un grupo de conceptos básicos, posteriormente, se realiza un análisis de los algoritmos reportados para la selección de rasgos y objetos, y una evaluación experimental de los mismos. Para finalizar, se brindan conclusiones y algunas líneas de trabajo futuro.

2 Conceptos básicos

2.1 Los objetos y su representación

El concepto de *objeto* es primario y por tanto no se puede definir en función de otros conceptos más simples. En este trabajo se considera como *objeto* una entidad entendible por el ser humano, que interviene en un problema de reconocimiento de patrones (RP). El conjunto de todos los

objetos se llamará *universo* y lo denotaremos aquí por U. Ejemplos de *objetos* pueden ser cosas tan disímiles como: embriones de zanahoria, pacientes, huellas dactilares, entre otros.

Usualmente los objetos están descritos por un conjunto de n *atributos* $R = \{r_1, r_2, \ldots, r_n\}$, que son extraídos del problema. Cada atributo está definido en un dominio $D_i = dom(r_i)$. La función $\begin{matrix} r_i : U \to D_i \\ x \to r_i(x) \end{matrix}$ asocia a cada objeto x el valor $r_i(x)$ de su atributo r_i. El *dominio* de definición de un atributo es su conjunto de valores admisibles. En dependencia de este conjunto se pueden tener atributos de diferentes tipos, por ejemplo: Booleanos ($D_i = \{0,1\}$); k-valentes ($D_i = \{0,1,\ldots,k-1\}, k > 2$); Reales ($D_i \subseteq \Re$); etc.

Usualmente, en un proceso de investigación, no se trabaja directamente con los objetos, sino con sus descripciones en función de los atributos seleccionados. En este documento utilizaremos el término *objeto* para referirnos tanto a los objetos, como a sus descripciones. En cada caso se puede saber a cuál de ellos se hace referencia a partir del contexto.

El desconocimiento del valor de un atributo r_i en un objeto x se denota con el símbolo "?", el que se adiciona al conjunto de valores admisibles de r_i. Es decir, si $? \in D_i$ puede haber desconocimiento del valor del atributo $r_i(x)$ en algún objeto x.

2.2 Funciones de analogía entre objetos

Las funciones de (di)similaridad son un componente importante de muchos métodos estadísticos, de aprendizaje y de reconocimiento de patrones. Son ampliamente utilizadas tanto en problemas de clasificación supervisada, como de agrupamiento y muchas veces determinan la calidad del resultado final.

La elección de la forma de comparar dos objetos debería ser extraída de la modelación del problema, según el concepto de similaridad (analogía) que utiliza el especialista en el dominio. Sin embargo, en muchos trabajos donde se utilizan repositorios estándar de bases de datos, esto no es posible. En estos casos se escoge entre un amplio repertorio de funciones, que ya han sido utilizadas con anterioridad, o se crean las propias. Es importante destacar que la representación de los objetos impone restricciones importantes en cuanto a las funciones que se pueden aplicar. Es por esto que existen varios trabajos dedicados a la creación de funciones para diferentes dominios [15, 16], así como estudios comparativos entre sus características [17].

Usualmente no se trabaja con los valores originales, sino que estos se normalizan primero. Esto se hace para que el resultado no sea afectado por el rango de definición de cada atributo. De no hacerse así, un atributo definido, por ejemplo, en el intervalo $[0,10000]$ puede hacer que su influencia anule totalmente a otro definido en el intervalo $[10,20]$. Otro aspecto que se incorpora en muchos sistemas es el *pesado* de los atributos, para reflejar el hecho que no todos tienen la misma importancia para un problema dado.

Entre las funciones de analogía que admiten datos mezclados e incompletos, cabe señalar la HEOM (Heterogeneous Euclidean-Overlap Metric). Esta función fue introducida por Wilson y Martínez [16] con el propósito de hacer comparaciones con funciones que mencionaremos a continuación. HEOM es similar a las usadas por IB1, IB2 and IB3, ver [15] entre otros. HEOM

usa la métrica *overlap* para los atributos nominales y la distancia normalizada de Euclides para los atributos numéricos y se define como:

$$HEOM(x,y) = \sqrt{\sum_{a=1}^{m} d_a(x_a, y_a)^2}$$

donde d_a depende del tipo del atributo a, siendo:

$$d_a(x,y) = \begin{cases} 1 & \text{si el valor del atributo } a \text{ en } x \text{ o } y \text{ es desconocido} \\ overlap(x,y) & \text{si el atributo } a \text{ es nominal} \\ rn\_diff_a(x,y) & \text{en otro caso} \end{cases}$$

donde

$$overlap(x,y) = \begin{cases} 0, \text{ si } x = y \\ 1, \text{ en otro caso} \end{cases}$$

y

$$rn\_diff_a(x,y) = \frac{|x-y|}{\max_a - \min_a}$$

Posteriormente Wilson y Martínez [16], propusieron varias extensiones, como por ejemplo la HVDM (Heterogeneous Value Difference Metric), donde la función de analogía de valores de un atributo d_a fue substituida por la utilizada en VDM [18]:

$$vdm_a(x,y) = \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^q$$

donde:

- $N_{a,x}$: Cantidad de objetos del conjunto de entrenamiento que tiene el valor x en el atributo a,
- $N_{a,x,c}$: Cantidad de objetos de clase c del conjunto de entrenamiento que tiene el valor x en el atributo a
- C: Cantidad de clases
- q: es una constante, usualmente 1 ó 2

Al utilizar vdm_a, dos valores son considerados como *cercanos* si tienen clasificaciones más similares (correlaciones más similares con el valor de clase), independientemente del posible orden de los valores.

También se ha utilizado VDM con datos mezclados, discretizando los valores numéricos [19]. Aunque discretizar tiene los inconvenientes ya señalados, puede dar resultados superiores en algunos problemas, por ejemplo, para seleccionar las personas que son buenos candidatos

para pilotear un avión de combate específico. En este caso las personas con alturas significativamente mayores o menores al tamaño óptimo, deben ser consideradas malos candidatos. Por tanto, a los efectos de la clasificación, estos valores pueden considerarse como similares, aunque numéricamente sean muy diferentes [16].

Hay que señalar que, aunque sus autores les dan el nombre de "distancias" heterogéneas, en caso de existir ausencia de información, la función no cumple la propiedad de ser definida positiva, por lo que no es una distancia. En otras palabras, no son distancias en el caso de datos mezclados e incompletos.

En este trabajo utilizaremos el término más general de disimilaridad para catalogar a las funciones de analogía entre objetos que devuelven valores más pequeños cuanto más similares son los objetos comparados. Utilizaremos el término de distancia para señalar el subconjunto de las disimilaridades que cumplen con las propiedades de: ser definida positiva, simetría y desigualdad triangular.

3 Métodos de selección de rasgos y objetos

Numerosos métodos de selección simultánea de rasgos y objetos se encuentran reportados en la literatura. En dependencia de las estrategias que guían el proceso de selección de rasgos y objetos, estos métodos pueden dividirse de manera general en métodos de *selección evolutiva*, métodos de *selección empotrada* y métodos de *fusión de submatrices*. Cada uno de estos enfoques tiene particularidades que lo diferencian del resto, y que determinan en gran medida el comportamiento de los métodos de selección.

3.1 Selección evolutiva

Los métodos de selección evolutiva, como su nombre lo indica, utilizan técnicas de la Computación Evolutiva [20] para obtener un subconjunto reducido de rasgos y objetos. De manera general estas técnicas poseen una gran capacidad de realizar búsquedas en espacios de grandes dimensiones, por lo que se han aplicado con éxito a la solución del problema que nos ocupa. Sin embargo, estos algoritmos tienen un componente estocástico elevado, y en su mayoría carecen de una semántica para especialistas de las áreas de aplicación como criminalistas, geólogos, entre otros, que les dificultad la mejor comprensión de los resultados alcanzados.

3.1.1 Selección mediante estrategias de *Ascenso de la Colina* (*Hill Climbing*)

En 1994 se realizó la primera propuesta de selección simultánea de rasgos y objetos [21]. El método, llamado RMHC-FP1, utiliza una estrategia de *Random Mutation Hill Climbing* para escoger el mejor subconjunto de rasgos y objetos.

El RMHC-FP1 funciona de la siguiente forma: inicialmente, se genera una cadena binaria de longitud igual a la cantidad de objetos más la cantidad de rasgos que representa la inclusión o exclusión de cada objeto y rasgo (ver Figura 2.) y es considerada como la solución actual. Posteriormente, esta solución es mutada aleatoriamente (ver Figura 3), y se calcula la eficacia del clasificador utilizando sólo estos rasgos y objetos. Si la eficacia es mejor que la de la

solución antes de mutar, se actualiza la solución, en caso contrario, se desecha. El proceso continúa durante un número de iteraciones definido por el usuario.

0	1	0	1	1	0

Fig. 2. Cadena binaria en el RMHC-FP1

0	1	0	1	1	0

⇩

0	1	0	1	**0**	0

Fig. 3. Mutación en el RMHC-FP1

A pesar de su simplicidad, este algoritmo en muchos casos obtiene resultados comparables con otros de su categoría, que utilizan estrategias más complejas.

3.1.2 Selección mediante *Algoritmos Genéticos*

Los Algoritmos Genéticos (GA) han sido extensamente utilizados para la selección simultánea de rasgos y objetos en aplicaciones reales [22]. Los métodos que utilizan GA se diferencian fundamentalmente en las funciones de ajuste *(fitness)*, los métodos de selección, mutación y cruzamiento que utilizan y en el valor de los parámetros del GA.

La primera en utilizar los GA para esta tarea fue Kuncheva, en 1999 [14]. En ese trabajo, se propone una estrategia de codificación utilizando cadenas binarias, de manera similar a las utilizadas por Skalak [21]. Inicialmente, se genera de forma aleatoria una población de individuos (cadenas binarias), que representan las posibles soluciones. Posteriormente, se aplica el operador de cruzamiento (en este caso cruzamiento simple con un punto de corte), y el operador de mutación. Los individuos que pasarán a la próxima generación se seleccionan de forma elitista, de acuerdo con siguiente función de ajuste:

$$fitness = A_{1-NN}(V) - \alpha\left(\frac{sf + so}{f + o}\right)$$

donde A_{1-NN} es la eficacia del 1-NN con respecto a un conjunto de validación V, sf es el número de rasgos seleccionados, so es el número de objetos seleccionados y α es un parámetro definido por el usuario. f y o representan la cantidad inicial de rasgos y objetos, respectivamente.

También en 1999, Ishibushi y Nakashima [23] propusieron el uso de GA, siguiendo una estrategia muy similar a la de Kuncheva. Su algoritmo sólo se diferencia en el uso del operador de mutación y en la función de ajuste utilizada. En este caso, se utilizan dos operadores de mutación, con diferentes probabilidades, uno con probabilidad más alta para las mutaciones que excluyan a un rasgo y objeto, y otro, con probabilidad muy baja, para la inclusión de rasgos u objetos. Esta estrategia favorece la obtención de soluciones con muy pocos rasgos y objetos. La función de ajuste utilizada está dada por:

$$fitness = w_{1-NN} * nwc(T) - w_f * sf - w_o * so$$

donde $nwc(T)$ es el número de objetos bien clasificados de la matriz de entrenamiento T, sf es el número de rasgos seleccionados , so es el número de objetos seleccionados y w_{1-NN}, w_f y w_o son pesos definidos por el usuario que están asociados a la efectividad del clasificador, la cantidad de rasgos seleccionados y la cantidad de objetos, respectivamente.

Otras propuestas que utilizan GA para la selección de rasgos y objetos son las de Rozypal y Kubat, y Ahn y colaboradores [22, 24]. En el primer caso, se utiliza una estrategia de codificación diferente, utilizando números reales, con el objetivo de obtener cadenas más cortas, y en el otro, se utiliza como función de ajuste la eficacia del 1-NN con respecto a un conjunto de prueba.

$$fitness = A_{1-NN}(Test)$$

Los parámetros de los diferentes algoritmos se muestran en la Tabla 1. Denotamos como KJ-GA, IN-GA y AKH-GA a los algoritmos propuestos por Kuncheva y Jain, Ishibushi y Nakashima y Ahn y colaboradores, respectivamente.

Tabla 1. Parámetros de los algoritmos

Parámetros	*Algoritmos*		
	KJ-GA	IN-GA	AKH-GA
Número de objetos	10	50	200
Número de generaciones	100	500	20
Probabilidad de mutación	0.1	0.1 - 0.01	0.1
Probabilidad de cruzamiento	1.0	1.0	0.7

Más recientemente, Ros y colaboradores [25] propusieron el uso de un Algoritmo Genético híbrido para la selección de rasgos y objetos. Esta propuesta se diferencia de las anteriores en varios aspectos:

1. Aplica operadores de cruzamiento y mutación de forma independiente en rasgos y objetos, es decir, cada cromosoma es tratado como dos subcromosomas (uno de rasgos y otro de objetos), utilizando el cruzamiento simple de un punto.
2. Impone restricciones a las soluciones en cuanto al número de objetos presentes de cada clase, que no puede ser menor que un porciento definido por el usuario. Esto garantiza la existencia de representantes de todas las clases en todos los cromosomas.
3. La generación de la población inicial no se realiza de forma totalmente aleatoria, sino que se limita la cantidad de bits activos, tanto para rasgos como para objetos.

Otra propuesta que utiliza Algoritmos Genéticos para la optimización de clasificadores NN es la de Ahn y colaboradores en el 2009 [26]. En este caso, no se seleccionan rasgos, sino que se le asignan pesos a cada uno de ellos. También se seleccionan objetos y se decide el número óptimo de vecinos a utilizar durante el proceso de clasificación.

3.1.3 Selección mediante *Algoritmos de Estimación de Distribución*

De manera similar a los Algoritmos Genéticos, los Algoritmos de Estimación de Distribución (EDA, por su sigla en inglés) también han sido utilizados para la selección simultánea de rasgos y objetos en clasificadores del vecino más cercano. Un ejemplo de ello lo constituye la propuesta de Sierra y colaboradores, el algoritmo PS-FSS-EDA[27].

En ese trabajo, se utiliza una estrategia de codificación binaria, de manera similar a la utilizada por Kuncheva y Jain. Es decir, cada individuo consiste en una cadena binaria, de longitud dada por la suma de la cantidad de rasgos y de objetos, donde 0 representa la exclusión de un determinado rasgo u objeto, y 1 su inclusión.

Como es sabido, los Algoritmos de Estimación de Distribución realizan la estimación de las probabilidades de acuerdo a un modelo de probabilidad predefinido. En el caso del trabajo mencionado, no se ofrecen detalles de qué modelo probabilístico de estimación se utilizó en el estudio.

3.1.4 Selección con otras estrategias evolutivas

Los *Algoritmos Evolutivos de Múltiples Objetivos* (MOEA), y sus variantes inteligentes (IMOEA) también han sido aplicados en la selección simultánea de rasgos y objetos. A diferencia de otras estrategias evolutivas, los MOEA hacen uso de la frontera de Pareto, con el propósito de encontrar múltiples soluciones Pareto óptimas, sin necesidad de integrar todos los objetivos a optimizar en una única función.

Chen y colaboradores [28] utilizan una estrategia de codificación binaria similar a la descrita anteriormente. La función de optimización empleada está dada por:

$$fitness = p - q + c$$

Donde p es la cantidad de individuos dominados por la solución actual en términos de optimización de Pareto, q es la cantidad de individuos que dominan a la solución actual y c es una constante. Así, tendrán mayor valor los individuos más dominantes.

En este algoritmo, se utiliza una estrategia de cruzamiento inteligente (*intelligent crossover*) [29]. Esta estrategia trata de identificar segmentos de genes con buen comportamiento, y busca la permanencia y evolución de éstos dentro de la población.

Para la estrategia de selección, se eligen mediante torneo binario un grupo de individuos de la población, y el resto de forma aleatoria de un conjunto élite, que está formado solamente por los individuos que son Pareto óptimos.

3.2 Selección empotrada

Entre los métodos de selección de rasgos se pueden mencionar los llamados métodos *envoltorios* (*wrappers*). Estos métodos obtienen un conjunto reducido de rasgos tal que optimice una cierta función, como por ejemplo la eficacia de un determinado clasificador. Entre los métodos de selección de rasgos de la categoría envoltorio, se destacan por su simpleza y efectividad el SBS y SFS, propuestos por Kittler en 1978 [30].

Estos métodos de manera general evalúan un conjunto candidato de rasgos con respecto a una determina función que se desea optimizar, devolviendo el conjunto de rasgos que obtenga

mejores resultados. El SBS considera sólo cambios locales en el conjunto de rasgos actual, inicialmente todos los rasgos. Una vez que se ha efectuado un cambio este no vuelve a considerarse. Comienza eliminando el rasgo cuya eliminación maximice una cierta función de optimización, y luego elimina un rasgo en cada iteración de forma tal que el conjunto de rasgos que queda maximice la función objetivo. Así, Dasarathy [31] propuso la integración del proceso de selección de objetos para cada conjunto de rasgos candidato, empotrando la selección de objetos en el SBS, y usando como función de optimización una medida combinada de la eficacia del clasificador y la tasa de retención de objetos.

$$fitness = \sqrt{\left(A_{1-NN}(V)\right)^2 + \left(\frac{so}{o}\right)^2}$$

donde A_{1-NN} es la eficacia de un clasificador 1-NN con respecto a un conjunto de validación V, so es el número de objetos seleccionados y o representa la cantidad inicial de objetos.

Para seleccionar los objetos, Dasarathy realiza la aplicación secuencial del algoritmo de edición con grafos de vecinos relativos (RNG-E) [32] y del algoritmo MCS [33], que es la combinación de métodos que obtuvo mejores resultados en los estudios realizados por él [34].

Aunque en nuestro conocimiento este es el único método que integra el proceso de selección de objetos en un algoritmo de selección de rasgos, la idea es aplicable a otros métodos empotrados de selección de rasgos y es posible utilizar otros métodos de selección de objetos, lo que abre un posible espacio de investigación en este sentido.

De manera general, esta estrategia tiene un costo computacional elevado, y se hace impracticable en dominios con grandes cantidades de rasgos. Sin embargo, ha mostrado buenos resultados experimentales con bases de datos internacionales.

3.3 Fusión de submatrices

Los métodos de la familia de *fusión de submatrices*, como su nombre lo indica, se basan en la obtención de submatrices de la matriz de entrenamiento, proyectando ésta con los rasgos obtenidos mediante el cálculo de testores típicos [35], y aplicando métodos de selección de objetos a cada proyección. Posteriormente, las submatrices se unen utilizando un procedimiento de fusión.

De manera general, los métodos de fusión de submatrices operan de la siguiente forma:

1. Calcular los testores típicos. Los testores típicos son subconjuntos de rasgos que tienen dos propiedades básicas: son combinaciones irreducibles de rasgos y no confunden descripciones de objetos de diferentes clases. Es decir, un testor típico es un subconjunto de rasgos con un alto poder discriminativo.
2. Proyectar la matriz de entrenamiento utilizando los rasgos presentes en cada testor.
3. Aplicar un método de selección de objetos en cada proyección, obteniendo así una submatriz.
4. Ordenar las submatrices, siguiendo un cierto criterio.
5. Fusionar las submatrices hasta que se cumpla una determinada condición de parada. La submatriz fusionada tendrá todos los rasgos y objetos de las submatrices que le dieron origen.

Estos métodos son deterministas, y aunque el cálculo de todos los testores típicos es un problema NP completo, existen algoritmos eficientes para su cómputo, como por ejemplo el LEX [36]. Es de destacar que estos métodos están especialmente diseñados para el manejo de datos mezclados e incompletos.

Entre los algoritmos de esta familia se encuentran el SOFSA [37] y el TCCS [13]. El SOFSA funciona de la siguiente manera: primero, se calculan los testores típicos y se ordenan de acuerdo a su peso informacional [38] que es la suma de los pesos de los rasgos presentes en el testor.

$$\rho(TT) = \sum_{R_i \in TT} \rho(R_i)$$

Donde el peso de cada rasgo se calcula como:

$$\rho(R_i) = \alpha^* \frac{\tau_{R_i}}{\tau} + \beta^* \frac{\sum_{\Omega \in TT_{R_i}} \frac{1}{|\Omega|}}{\tau_{R_i}}$$

donde τ_{R_i} es la cantidad de testores típicos en los que R_i aparece, τ es el total de testores típicos y TT_{R_i} es la familia de todos los testores típicos en los que R_i aparece.

Después de que los testores típicos son ordenados, la matriz de entrenamiento es proyectada usando cada testor típico y se aplica el algoritmo CSE [8] a esta proyección. Si la eficacia del clasificador con respecto a una muestra de validación es mayor que la eficacia inicial utilizando todos los rasgos y objetos, se devuelve una solución. Si no, se considera el próximo testor típico. En cada paso, se obtiene una submatriz. Las submatrices obtenidas generan una nueva submatriz formada por la unión de los rasgos y objetos de cada una de ellas, como se explica en la Figura 4.

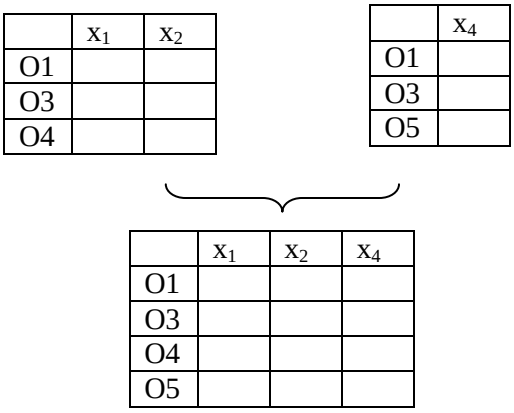


Fig. 4. Unión de submatrices en el SOFSA

Una de las deficiencias del SOFSA es que en ocasiones no logra reducir ni rasgos ni objetos, es por ello que en [13] se introduce un nuevo método capaz de suplir esa limitante.

El TCCS, al igual que el SOFSA, se basa en el uso de los testores típicos y el CSE, pero a diferencia de éste, pesa los testores típicos de acuerdo a la eficacia del clasificador con respecto a una matriz de validación, e incorpora una estrategia de doble edición (ver Figura 5).

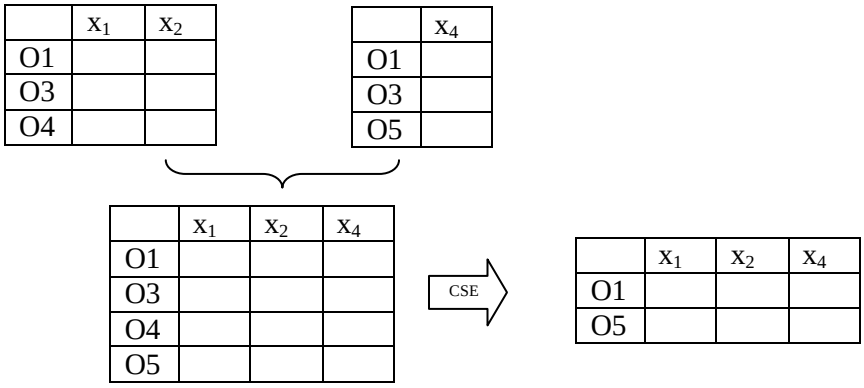


Fig. 5. Estrategia de mezcla y doble edición en el TCCS

Esta estrategia hace que el TCCS obtenga mayores reducciones en el número de objetos. Por otra parte, el uso de la eficacia del clasificador en el proceso de ordenamiento de la submatrices, agiliza éste, pues el peso informacional no está directamente relacionado con la eficacia de clasificación.

3.4 Otros algoritmos de selección de rasgos y objetos

En el contexto de la clasificación de texto, en 2002 Fragoudis y colaboradores propusieron un algoritmo para la selección de un subconjunto de rasgos y objetos de forma integrada [39]. En este caso, se considera que los objetos no están descritos por la misma cantidad de rasgos, sino que cada documento en particular está descrito por un conjunto de palabras de un vocabulario.

Dicho algoritmo asume un problema de clasificación binaria, es decir, sólo es aplicable a problemas de dos clases. Consiste en la selección de un subconjunto de rasgos (palabras) que más frecuentemente aparecen en la clase objetivo. En cada iteración, se añade el mejor rasgo y posteriormente todos los documentos que contienen dicha palabra. Finalmente, se obtiene un conjunto de documentos y rasgos relevantes. Como en este reporte se asume que todos los objetos están representados por un conjunto igual de rasgos, este algoritmo no lo hemos considerado para nuestro análisis.

Por otra parte, en 2008, Villegas y Paredes [40] introdujeron un algoritmo de selección de rasgos y construcción de objetos. Éste asume que los objetos están representados en un espacio vectorial, donde existe definida una suma y un producto por un escalar. El algoritmo (LDPP) va seleccionando un subconjunto de rasgos (proyecciones base), inicialmente obtenidas por Análisis de Componentes Principales (PCA) de dos componentes, y construye un conjunto de prototipos en cada paso. El conjunto de prototipos inicialmente se obtiene al calcular el objeto que representa la media de cada clase. Este método se basa en la construcción de objetos, es decir, objetos que no están presentes en la muestra de entrenamiento, y en este reporte nos

centramos en los métodos de selección de objetos, por lo que también queda fuera del alcance de nuestro análisis.

4 Evaluación experimental

Para la evaluación experimental de los diferentes algoritmos, utilizamos un grupo de bases de datos del repositorio de la Universidad de California en Irvine (UCI) [41]. En la sección 4.1 se brinda información de las bases de datos. Se utilizó la validación cruzada en 10 hojas (*10-fold cross validation*), por ser la más común en la comparación de clasificadores supervisados. Se utilizaron clasificadores k-NN con 1, 3 y 5 vecinos. Como funciones de analogía se utilizaron la HEOM y la HVDM [16], descritas en la sección 2.2. De esta forma, al realizar experimentaciones con varios clasificadores NN y varias funciones de analogía, se disminuye el riesgo de que las conclusiones de los experimentos sean dependientes de la cantidad de vecinos y la función utilizada. Se probaron cuatro algoritmos pertenecientes a la familia de Selección Evolutiva, el RMHC-FP, el KJ-GA, IN-GA y el AKH-GA, por ser ésta la más prolífera, así como el DS, perteneciente a la categoría de Selección Empotrada, y el SOFSA y el TCCS, de la familia de Fusión de Submatrices.

4.1 Bases de datos utilizadas

La descripción de las bases de datos utilizadas en los experimentos se muestra en la Tabla 2. Como se puede observar, se utilizaron bases de datos de diversa naturaleza, con un número de clases que varía entre 2 y 22.

Tabla 2. Descripción de las bases de datos utilizadas

Base de datos	Objetos	Rasgos numéricos	Rasgos no numéricos	Cantidad de clases	Valores desconocidos
autos	205	15	10	7	x
breast-w	699	9	0	2	
credit-a	690	6	9	2	x
diabetes	768	8	0	2	
heart-c	303	6	7	5	
hepatitis	155	6	13	2	x
iris	150	4	0	3	
labor	57	8	8	2	x
lymph	148	3	15	4	
postoperative	90	0	8	3	x
primary-tumor	339	1	16	22	x
vehicle	946	18	0	4	
vote	435	0	16	2	x
wine	178	13	0	3	
zoo	101	1	16	7	

4.2 Resultados experimentales

Para el análisis de los resultados, se utilizó la prueba T de Student con una significación de 0.05, con respecto al resultado de mejor eficacia del clasificador, incluyendo los resultados originales sin seleccionar rasgos ni objetos. De esta forma, se contó la cantidad de veces que cada método estuvo entre los de menor error de acuerdo a la prueba T de diferencia de medias.

En la Tabla 3 se muestran los resultados obtenidos para la función HEOM, y en la Tabla 4, para la HVDM. En negrita se muestran los mejores resultados.

Tabla 3. Cantidad de veces que se obtuvo el menor error con la función HEOM, de acuerdo a la prueba T

Métodos	Clasificadores		
	1-NN	3-NN	5-NN
AKH-GA	11	10	6
DS	11	7	8
IN-GA	3	4	3
KJ-GA	5	5	2
RMHC-FP1	8	7	5
SOFSA	**15**	**15**	**15**
TCCS	13	**15**	13
Original	**15**	**15**	**15**

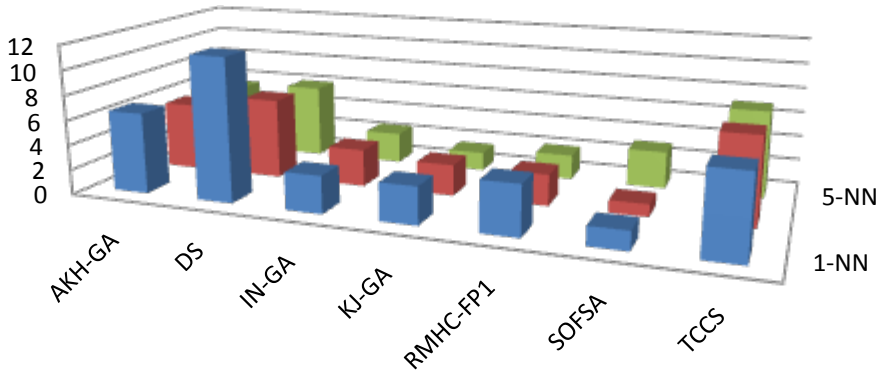
Para la función HVDM, en ninguna base de datos la eficacia original fue superada de forma significativa utilizando un número reducido de rasgos y objetos. Sin embargo, sí fue igualado para todos los clasificadores por el SOFSA, y para el 3-NN por el TCCS. Por otra parte, como puede observarse en la Tabla 4, para la función HVDM y el clasificador 3-NN, en dos bases de datos la eficacia original fue superada de forma significativa, utilizando un número reducido de rasgos y objetos. En el caso del 1-NN y el 5-NN, no fue posible igualar la eficacia original con un número reducido de rasgos y objetos, pues ningún método obtuvo el menor error en todos los casos. El método que más se acercó fue el SOFSA, pero falló en algunas bases de datos.

Tabla 4. Cantidad de veces que se obtuvo el menor error con la función HVDM, de acuerdo a la prueba T

Métodos	Clasificadores		
	1-NN	3-NN	5-NN
AKH-GA	9	10	12
DS	12	6	4
IN-GA	3	2	3
KJ-GA	4	3	4
RMHC-FP1	8	2	3
SOFSA	14	**14**	12
TCCS	14	**14**	10
Original	**15**	13	**15**

Para analizar la eficacia en cuanto a la reducción de rasgos y objetos, se decidió utilizar el cálculo de la *frontera de Pareto* [42]. Este cálculo está diseñado para evaluar soluciones en problemas multiobjetivos. En este caso, se tienen dos objetivos: minimizar la cantidad de objetos y minimizar la cantidad de rasgos. Un resultado es *Pareto óptimo* si no existe ningún otro resultado que lo supere en todos los objetivos de forma simultánea. Así, estarán en la frontera de Pareto aquellos métodos que hayan logrado un menor error de forma significativa, y que no sean superados por ningún otro método en cuanto a la reducción de rasgos y objetos.

Como se utilizaron dos funciones de analogía, se promedió la cantidad de veces que cada método se encontró en la frontera de Pareto para cada función en cada uno de los tres clasificadores utilizados. Los resultados del promedio de aparición de cada método se muestran en la Figura 6. Como puede observarse, para el clasificador 1-NN el método con mejor eficacia fue el DS, pues tiene un promedio de aparición de 11.5, sin embargo, para los clasificadores 3-NN y 5-NN, el método con mejores resultados resultó ser el TCCS, con un promedio de 7 apariciones. El resto de los métodos, de manera general, presentan pocas apariciones en la frontera. A pesar de que el SOFSA obtiene los mejores resultados en cuanto a eficacia del clasificador, es superado en la reducción de rasgos y objetos por otros métodos.



	AKH-GA	DS	IN-GA	KJ-GA	RMHC-FP1	SOFSA	TCCS
1-NN	6.5	11.5	3	3	4	1.5	6.5
3-NN	5.5	6.5	3	2.5	2.5	1	7
5-NN	5.5	6	2.5	1.5	2	3	7

Fig. 6. Promedio de aparición en la frontera de Pareto

5 Conclusiones

En este reporte se analizaron los métodos de selección simultánea de rasgos y objetos para el mejoramiento de clasificadores de la familia del vecino más cercano. Se dividió el análisis de los diferentes métodos, agrupándolos de acuerdo a sus características en tres familias: la selección

evolutiva, la selección empotrada y la selección por fusión de submatrices. Se analizaron las ventajas y desventajas de cada una de las familias, y se realizó un análisis experimental de la eficacia de métodos representativos de cada una de ellas, utilizando tres clasificadores de la familia NN y dos funciones de analogía. Los resultados experimentales mostraron que en cuanto a la eficacia del clasificador, los métodos con mejor eficacia son los de selección por fusión de submatrices, sin embargo, en cuanto a la reducción de rasgos y objetos son en ocasiones superados por el método de selección empotrada. Los métodos de selección evolutiva, por su parte, en la mayoría de los casos degradan significativamente la eficacia del clasificador.

Referencias bibliográficas

1. Duda, R.O., P.E. Hart, and D.G. Stork, *Pattern Classification*. 2000: Wiley-Interscience.
2. Dasarathy, B.D., *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques.* 1991, Los Alamitos, California: IEEE Computer Society Press.
3. Kuncheva, L.I., *Combining pattern classifiers : methods and algorithms*. 2004, Hoboken, N.J.: Wiley-Interscience. xx, 350 p.
4. Cover, T. and P.E. Hart, *Nearest Neighbor pattern classification.* IEEE Trans. on Information Theory, 1967. **13**(1): p. 21-27.
5. Guyon, I. and A. Elisseeff, *An introduction to variable and feature selection.* Journal of Machine Learning Reserch, 2003. **3**: p. 1157-1182.
6. Morita, M., et al. *Unsupervised Feature Selection Using Multi-Objective Genetic Algorithms for Handwritten Word Recognition*. in *ICDAR03*. 2003.
7. Zhu, Z., Y.-S. Ong, and M. Dash, *Wrapper-Filter feature selection algorithm using a memetic framework.* IEEE Transactions on Systems, Man, and Cybernetics, 2007. **37**(1): p. 70-76.
8. García-Borroto, M. and J. Ruiz-Shulcloper, *Selecting Prototypes in Mixed Incomplete Data.* Lecture Notes in Computer Science, 2005. **3773**: p. 450-459.
9. Rodríguez-Colín, R., J.A. Carrasco-Ochoa, and J.F. Martínez-Trinidad, *Reward-Punishment Editing for Mixed Data.* Lecture Notes in Computer Science, 2005. **3773**: p. 481-488.
10. Caballero, Y., et al., *A method to edit training set based on rough sets.* International Journal of Computational Intelligence Research., 2007. **3**(3): p. 219-229.
11. Hao, X.-L., et al. *Efficient kNN text categorization based on Multiedit and condensing techniques*. in *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics*. 2007. Hong Kong.
12. García-Borroto, M., et al., *Selección y construcción de objetos para el mejoramiento de un clasificador supervisado: un análisis crítico*, in *Reconocimiento de Patrones*, CENATAV, Editor. 2008, Centro de Aplicaciones de Tecnologías de Avanzada: La Habana, Cuba.
13. Villuendas-Rey, Y., et al. *Selecting features and objects for mixed and incomplete data*. in *13th Iberoamerican Congress in Pattern Recognition, CIARP 2006*. 2008. La Habana, Cuba: LNCS 5197, Springer, Heidelberg.
14. Kuncheva, L.I. and L.C. Jain, *Nearest neighbor classifier: Simultaneous editing and feature selection.* Pattern Recognition Letters, 1999. **20**: p. 1149--1156.
15. Aha, D.W., D. Kibler, and M.K. Albert, *Instance-based learning algorithms.* Machine Learning, 1991. **6**: p. 37-66.
16. Wilson, R.D. and T.R. Martinez, *Improved Heterogeneous Distance Functions.* Journal of Artificial Intelligence Research, 1997. **6**: p. 1-34.
17. Lumijarvi, J., J. Laurikkala, and M. Juhola, *A comparison of different heterogeneous proximity functions and Euclidean distance.* Medinfo, 2004. **11**(2): p. 1362-6.
18. Stanfill, C. and D.L. Walts, *Toward memory-based reasoning*. Communications of the ACM, 1986. **29**: p. 1213-1228.

19. Ventura, D. and T.R. Martínez. *An empirical comparison of discretization methods*. in *Tenth International Symposium on Computer and Information Sciences*. 1995.
20. Baeck, T., D.B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. 1997: CRC Press.
21. Skalak, D.B. *Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms*. in *Eleventh International Conference on Machine Learning*. 1994.
22. Ahn, H., K.J. Kim, and I. Han, *A case-based reasoning system with the two-dimensional reduction technique for customer classification.* Expert Systems with Applications: An International Journal, 2007. **32**: p. 1011-1019.
23. Ishibushi, H. and T. Nakashima, *Evolution of reference sets in nearest neighbor classification.* LNCS, 1999. **1585**: p. 82-89.
24. Rozsypal, A. and M. Kubat, *Selecting representative examples and attributes by a genetic algorithm.* Intelligent Data Analysis, 2003. **7**: p. 291-304.
25. Ros, F., et al., *Hybrid genetic algorithm for dual selection.* Pattern Analysis and Applications, 2008. **11**: p. 179-198.
26. Ahn, H. and K.J. Kim, *Global optimization of case-based reasoning for breast cytology diagnosis.* Expert Systems with Applications, 2009. **36**: p. 724-734.
27. Sierra, B., et al., *Prototype Selection and Feature Subset Selection by Estimation of Distribution Algorithms: A case study in the survival of cirrhotic patients treated with TIPS.*
28. Chen, J.H., H.M. Chen, and S.Y. Ho, *Design of nearest neighbor classifiers: multi-objective approach.* International Journal of Approximate Reasoning, 2005: p. 3-22.
29. Ho, S.-Y., L.-S. Shu, and J.-H. Chen, *Intelligent evolutionary algorithms for large parameter optimization problems.* IEEE Transaction on Evolutionary Computation, 2004. **8**(6): p. 522-541.
30. Kittler, J., *Feature set search algorithms*, in *Pattern recognition and signal proccessing*, C.H. Chen, Editor. 1978, Sijthoff and Noordhoff: The Netherlands.
31. Dasarathy, B.V. *Concurrent Feature and Prototype Selection in the Nearest Neighbor Decision Process*. in *4th World Multiconference on ystemics, Cybernetics and Informatics*. 2000. Orlando, USA.
32. Toussaint, G. *Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress*. in *34 Symposium on Computing and Statistics INTERFACE-2002*. 2002. Montreal, Canada.
33. Dasarathy, B.D., *Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design.* IEEE Transactions on Systems, Man and Cybernetics, 1994. **24**: p. 511-517.
34. Dasarathy, B.V., J.S. Sanchez, and S. Townsend, *Nearest Neighbour Editing and Condensing Tools - Synergy Exploitation.* Pattern Analysis & Applications, 2000.
35. Lazo-Cortés, M., J. Ruiz-Shulcloper, and E. Alba-Cabrera, *An overview of the evolution of the concept of testor.* Pattern Recognition, 2001. **34**(4): p. 753-762.
36. Santiesteban, Y. and A. Pons-Porrata, *LEX: A new algorithm to calculate typical testors. .* Mathematics Sciences Journal, 2003. **21**.
37. Villuendas Rey, Y., et al., *Simultaneous Features and Objects Selection for Mixed and Incomplete Data.* Lecture Notes on Computer Science, 2006. **4225**: p. 597-605.
38. Lazo-Cortés, M. and J. Ruiz-Shulcloper, *Determining the feature informational weight for non-classical described objects and new algorithm to calculate fuzzy testors.* Pattern Recognition Letters, 1995. **16**: p. 1259-1265.
39. Fragoudis, D., D. Meretakis, and S. Likothanassis. *Integrating Feature and Instance Selection for Text Classification*. in *SIGKDD '02*. 2002. Edmonton, Alberta, Canada: ACM Press.
40. Villegas, M. and R. Paredes, *Simultaneous Learning of a Discriminative Projection and Prototypes for Nearest-Neighbor Classification.* 2008.
41. Merz, C.J. and P.M. Murphy, *UCI Repository of Machine Learning Databases*. 1998, University of California at Irvine, Department of Information and Computer Science: Irvine.
42. Cohon, J., *Multiobjective Programming and Planning*. 1978: John Wiley, New York.

RT_022, enero 2010
Aprobado por el Consejo Científico CENATAV

Editor: Lic. Lucía González Bayona
Diseño de Portada: DCG Matilde Galindo Sánchez
RNPS No. 2142
ISSN 2072-6287
Indicaciones para los Autores:
Seguir la plantilla que aparece en www.cenatav.co.cu
C E N A T A V
7ma. No. 21812 e/218 y 222, Rpto. Siboney, Playa;
Ciudad de La Habana. Cuba. C.P. 12200
Impreso en Cuba