Revista Cubana de Ciencias Informáticas
Vol. 10, No. 4, Octubre-Diciembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 83-96

Tipo de artículo: Artículo original
Temática: Inteligencia Artificial
Recibido: 08/05/2016 | Aceptado : 22/10/2016

Efecto de la selección de rasgos en la clasificación basada en prototipos

Effect of the features selection in the classification based on prototypes

Yumilka Bárbara Fernández Hernández [1*], Rafael Bello Pérez [2], Yaima Filiberto Cabrera [1], Mabel Frías Dominguez[1], Yaile Caballero Mota [1]

[1*] Departamento de Computación. Universidad de Camagüey. "Ignacio Agramonte". Circunvalación Norte km 5 1/2, Camagüey, CP 70300. {yaima.filiberto, mabel.frias, yaile.caballero}@reduc.edu.cu
[2] Departamento de Ciencia de la Computación. Universidad Central de Las Villas, Carretera a Camajuaní, km 5, Santa Clara, CP 54838. rbellop@uclv.edu.cu

Autor para correspondencia: yumilka.fernandez@reduc.edu.cu

Resumen

La selección de atributos es una técnica de procesamiento de datos cuyo objetivo es buscar un subconjunto de atributos que mejore el rendimiento del clasificador. Teniendo en cuenta que en los problemas de clasificación, la generación de prototipos es de gran utilidad, el principal aporte de este trabajo es proponer un nuevo método que integre la construcción de prototipos en este tipo de problemas con el método NP-BASIR (utilizando las relaciones de similaridad para realizar la granulación del universo, esta genera clases de similitud de objetos del universo, y para cada clase de similitud se construye un prototipo) combinado con el método de selección de atributos basado en la medida calidad de la similaridad para el cálculo de reductos utilizando la técnica de optimización *Particle Swarm Optimization*. El principal aporte de esta investigación es demostrar la utilidad de combinar selección de atributos unido a la construcción de prototipos. El algoritmo propuesto fue probado en conjuntos de datos internacionales y se comparó con algoritmos conocidos para la generación de prototipos. Los resultados experimentales muestran que el método propuesto obtuvo resultados satisfactorios, siendo la principal ventaja que se logra reducir en el conjunto de datos la cantidad de objetos y la cantidad de atributos sin variar significativamente la calidad de la clasificación comparada con el conjunto de datos original.

Palabras clave: selección de atributos; generación de prototipos; relaciones de similaridad; clasificación

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 4, Octubre-Diciembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 83-96

Abstract

Feature selection is a preprocessing technique with the objective of finding a subset of attributes that improves the classifier performance. In this paper is proposed a new method for solving classification problems based on prototypes (NP-BASIR-Class method) using feature selection. When using similarity relations for the granulation of the universe, similarity classes are generated, and a prototype is constructed for each similarity class. The feature selection method used was REDUCT-SIM based in the technique of optimization PSO (Particle Swarm Optimization). The main contribution of this investigation is demonstrating the utility of combining feature selection together to the prototype generation. The proposed algorithm was proven in groups of international data set and it was compared with well-known algorithms for the generation of prototypes. The experimental results show that the proposed method obtained satisfactory results, being the main advantage that is possible to reduce in the data set, the quantity of objects and the quantity of features obtaining satisfactory results without varying significantly the quality of the classification compared with the original data set.

Keywords: *features selection; prototype generation; similarity relations; classification*

Introducción

En problemas de clasificación basados en objetos o ejemplos, los métodos de aprendizaje san un conjunto de entrenamiento para estimar la etiqueta de la clase. Estos algoritmos de aprendizaje presentan problemas de escalabilidad cuando el tamaño del conjunto de entrenamiento crece, siendo así que la cantidad de ejemplos de entrenamiento afecta el costo computacional del método (García-Duran, R., & Borrajo, 2010); como se puede ver en (Barandela, 2001) la regla del vecino más cercano es un ejemplo de un alto costo computacional cuando la cantidad de objetos ejemplos es mayor (Jiang, Pang, Wu, & Kuang, 2012).

Para disminuir este problema, una alternativa es la clasificación basada en Prototipo más Cercano (*Nearest Prototype, NP*) (Bezdek & Kuncheva, 2001). El mismo se refiere a determinar el valor del atributo decisión de un nuevo objeto analizando su similaridad con respecto a un conjunto de prototipos, seleccionados o generados a partir del conjunto inicial de objetos.

La manera de obtener este conjunto de prototipos está basada en seleccionarlos de un conjunto original de ejemplos etiquetados, o reemplazando el conjunto original por uno diferente y disminuido (Triguero I, Derrac J, Garc´ıa S., & F., 2011). Los métodos de reducción de datos, cuando se refieren a ejemplos se dividen en dos categorías: la Selección de Prototipos (SP) (BIEN & TIBSHIRANI, 2012) y la Generación de Prototipos (GP) (León, J., & Giraldo, 2012). Los

métodos de selección de prototipos conservan un subconjunto de objetos representativos de los datos de entrenamiento original desechando el ruido y los objetos redundantes. Los métodos de generación de prototipos, por el contrario, seleccionan los datos que puedan reemplazar los datos originales con nuevos datos artificiales (Espinilla, Quesada, Moya, Martinez, & Nugent., 2015). Este proceso permite llenar regiones del dominio del problema que no tiene ejemplos representativos en los datos originales.

La Generación de Prototipos (GP) que tiene como objetivo obtener un conjunto de entrenamiento lo más reducido posible que permita 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 costo computacional. Además, en ocasiones puede mejorar su precisión, mediante la eliminación de ruido. Las técnicas de generación de prototipo han demostrado ser muy competitivas mejorando el rendimiento del clasificador del vecino más cercano (Triguero, Derrac, García, & Herrera, 2012). Los clasificadores basados en prototipos permiten determinar la clase de un nuevo ejemplo basado en un conjunto reducido de prototipo en vez de usar un gran conjunto de ejemplos conocidos.

El acercamiento NP permite disminuir los costos de almacenamiento y procesamiento de las técnicas de aprendizaje basadas en ejemplos. En (S.W. Kim, 2003) se plantea que cuando la cantidad de información es muy grande, una posible solución es reducir la cantidad de vectores "ejemplos", siempre que la eficacia obtenida sea mejor o igual que cuando se utiliza el conjunto original de datos. Por lo tanto, un buen conjunto de prototipos tiene dos características importantes: cardinalidad mínima y precisión máxima en la solución de un problema. Se necesitan estrategias para reducir la cantidad de ejemplos de entrada a una cantidad pequeña representativa y su actuación debe evaluarse teniendo en cuenta la precisión de la clasificación y el coeficiente de la reducción.

Más de 25 métodos de GP han sido referidos en la literatura, en (Triguero I et al., 2011) se presenta un estudio experimental donde se identifican las principales características de estos métodos y se muestra su taxonomía siguiendo un orden jerárquico basado en los mecanismos de generación, en los conjuntos de generación resultantes, tipo de reducción y evaluación de la búsqueda. Los 5 métodos que ofrecen los mejores resultados según este artículo son: Generalized Editing using Nearest Neighbor GENN (Koplowitz & Brown, 1981), Evolutionary Nearest Prototype Classifier ENPC (F. Fernández & Isasi, 2004), Integrated Concept Prototype Learned 2, ICPL2 (W.LAM, C.K.KEUNG, & . Aug. 2002), Reduction by space partitioning 3 RSP3 (Sánchez, 2004), *Particle Swarm Optimization* PSO (Nanni & Lumini, 2008).

Existen varias aproximaciones basadas en técnicas de agrupamiento (Bermejo & Cabestany, 2000), (Patané & Russo, 2001), (León et al., 2012), las cuales se caracterizan por dos etapas principales, la primera es para agrupar un conjunto

de datos de entradas sin etiquetas para obtener un conjunto reducido de prototipos, la segunda es para poner etiquetas a estos prototipos basados en ejemplos ya etiquetados y en la regla del vecino más cercano (F. Fernández & Isasi, 2004).

En (Triguero et al., 2012) los autores plantean un enfoque híbrido para integrar un esquema de pesos con el fin de obtener la reducción de datos, aplicando una técnica auto-adaptativa de evolución diferencial con el fin de optimizar la ponderación dada a cada característica y la localización de los prototipos, obteniendo mejoras en el rendimiento del clasificador del vecino más cercano.

En (BELLO-GARCIA, GARCIA, & BELLO, 2013) se presenta un método para la construcción de prototipos en problemas de aproximación de funciones utilizando la medida calidad de la similaridad (Filiberto, Bello, Caballero, & Larrua, 2010b) y la metaheuristica UMDA, mientras que en (Y. Fernández et al., 2015) se propone un algoritmo para construir prototipos en problemas de clasificación.

Selección de atributos

En los problemas de clasificación se puede pensar que lo más significativo es disponer de la máxima información posible. Por lo que puede parecer que cuanto mayor sea el número de atributos empleados mejor. El rendimiento de los algoritmos de aprendizaje se puede deteriorar ante la abundancia de información. Muchos atributos pueden ser completamente irrelevantes para el problema. Además, varios atributos redundantes pueden estar proporcionando la misma información (Chandrashekar & Sahin, 2014). Por tanto, podríamos prescindir perfectamente de algunos de ellos sin perjudicar nada la resolución del problema, y mejorar los resultados en diversos aspectos. De modo que resulta útil realizar una selección de atributos, antes de proceder al proceso de clasificación, pues es un hecho que el comportamiento de los clasificadores mejora cuando se eliminan los atributos no relevantes y redundantes (Araúzo, 2006).

En la selección de atributos se intenta escoger el subconjunto mínimo de atributos de acuerdo con dos criterios: que la tasa de aciertos no descienda significativamente; y que la distribución de clase resultante, sea lo más semejante posible a la distribución de clase original, dados todos los atributos. En general, la aplicación de la selección de características ayuda en todas las fases del proceso de descubrimiento de conocimiento. Los algoritmos de selección de atributos escogen un subconjunto mínimo de características que satisfaga un criterio de evaluación (Liu & Motoda, 2007) (Ch., D., R., & K., 2011). Cuando los algoritmos de selección de atributos se aplican antes de la clasificación, estamos interesados en aquellos atributos que clasifican mejor los datos desconocidos hasta el momento. Si el algoritmo proporciona un subconjunto de atributos, este subconjunto se utiliza para generar el modelo de conocimiento que clasificará los nuevos datos (Triguero et al., 2012).

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 4, Octubre-Diciembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 83-96

La utilización de métodos de selección de atributos en el desarrollo de clasificadores se aborda en (Araúzo, 2006), donde se expone las ventajas que puede aportar en cuanto a la eficiencia (en tiempo y/o en espacio) de la mayoría de los algoritmos de aprendizaje; ya que seleccionando un conjunto de características más pequeño el algoritmo funcionará más rápido y/o con menor consumo de memoria u otros recursos. También la mejora en los resultados obtenidos es otra ventaja puesto que algunos de los algoritmos de aprendizaje, que trabajan muy bien con pocas características relevantes, ante la abundancia de información pueden tratar de usar características irrelevantes y ser confundidos por las mismas, ofreciendo resultados peores. Así que la selección de atributos puede ayudar a obtener mejores resultados a un algoritmo indicándole en que características centrar la atención, puede lograr la reducción de los costes de adquisición de datos y la mejora en la interpretación de los resultados, al estar basada en un menor número de características.

En la sección 2 se describe el método REDUCT-SIM (Y. Fernández, Bello, Filiberto, Caballero, & Frías, 2013) que se utilizó para realizar la selección de atributos antes de aplicar el proceso de construcción de prototipos propuesto en (Y. Fernández et al., 2015), En la sección 3 se analiza el desempeño de la nueva propuesta y finalmente se presentan las conclusiones.

Selección de rasgos aplicado a la clasificación basada en prototipos.

Este epígrafe presenta las principales características que definen al método de selección de atributos (REDUCT-SIM) y al método de generación de prototipos NPBASIR-Class. Se describe la nueva propuesta que surge de la combinación de estas dos técnicas, NPBASIR-Class-RED, y se realizan los estudios experimentales con conjuntos de datos internacionales, incluyendo además el análisis estadístico basado en pruebas no paramétricos.

Método de selección de atributos (REDUCT-SIM)

Para el cálculo de los reductos se utilizan los pesos calculados para los rasgos al construir la relación de similaridad. El método propuesto en (Filiberto et al., 2010b) busca el conjunto de pesos $W=\{ w_1, \ldots, w_n \}$ asociado al conjunto de los rasgos de condición $C=\{ r_1, \ldots, r_n \}$ mediante una búsqueda heurística que maximice la medida calidad de la similaridad. El método REDUCT-SIM utiliza el conjunto de pesos W para construir reductos. La esencia de este método es eliminar los rasgos en orden inverso a su importancia, mientras se mantenga la condición de que el subconjunto resultante es un reducto.

Para chequear la condición de reducto se utiliza la medida calidad de la clasificación, la cual se define por la expresión (1);

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 4, Octubre-Diciembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 83-96

$$\gamma_B(Y) = \frac{\sum_{i=1}^{m} card(B_* Y_i)}{N} \quad (1)$$

Donde *card(B*Yi)* denota la cardinalidad de la aproximación inferior de cada clase, *m* es la cantidad de clases y *N* es la cantidad de objetos del universo. Un reducto *B* cumple que $\gamma_B(Y)=\gamma_C(Y)$.

El cálculo de la medida calidad de la clasificación exige que los rasgos de condición tengan dominio discreto, en otro caso es necesario discretizar el dominio.

A continuación, se describe los pasos que son seguidos en el algoritmo REDUCT-SIM:

Paso 1: Calcular el conjunto de pesos *W* según el método propuesto en (Filiberto et al., 2010b).

Paso 2: Ordenar los rasgos en C según el valor de los pesos calculados de mayor a menor.

Paso 3: Discretizar los rasgos de dominio continuo.

Paso 4: Calcular la medida calidad de la clasificación del conjunto de entrenamiento discretizado.

Paso 5: Eliminar sucesivamente los rasgos de menores pesos mientras que se conserve el valor de la medida calidad de la clasificación.

Paso 6: Se considera como reducto el conjunto de rasgos resultantes.

Método de construcción de prototipos (NPBASIR-Class)

Este método es un proceso iterativo en el cual los prototipos son construidos de las clases de similitud de objetos en el universo: una clase de similitud es construida usando la relación de similitud R ([Oi]R) y un prototipo es construido para esta clase de similitud. Cuando un objeto es incluido en una clase de similitud, es marcado como usado y no se tiene en cuenta para construir otra clase de similitud; Un arreglo de n componentes es usado, llamado Usado [] donde en Usado [i] tiene valor 1 si el objeto fue utilizado o 0 en otro caso.

Este algoritmo utiliza un conjunto de instancias X = {X1, X2...Xn} cada una de las cuales es descrita por un vector de a atributos descriptivos y pertenece a una de k clases w = {w1, w2,..., wk} y una relación de similitud R. La relación de similitud R es construida acorde al método propuesto en (Filiberto et al., 2010b), (Filiberto, Bello, Caballero, & Larrua, 2010); que está basado en encontrar la relación que maximice la calidad de la medida de similitud. En este caso, la relación R genera una granulación considerando los a atributos descriptivos, tan similares como posibles para la granulación acorde a las clases.

El método propuesto consiste en un procedimiento iterativo en el cual se construyen prototipos a partir de las clases de similaridad de objetos del universo: se construye una clase de similaridad usando la relación de similaridad *R ([Oi] R)*

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 4, Octubre-Diciembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 83-96

y se construye un prototipo para esta clase de similaridad. Cada vez que un objeto se incluye en una clase de similaridad se marca como usado, y no se tiene en cuenta para construir a partir de una clase de similaridad; pero los objetos usados si pueden pertenecer a clases de similaridad que se construyan para otros objetos no usados. Se usa un arreglo de *n* componentes, denominado *Usado[]*, en el cual *Usado[i]* tiene un valor *1* si el objeto ya fue usado, ó *0* en otro caso.

Algoritmo: NPBASIR-C (entrada: X, salida: ProtoSet)

Dado un conjunto de n objetos con m atributos descriptivos y un atributo de decisión y una relación de similitud R

P1: Inicializar contador de objetos.

Usado [j] ← 0, para j=1,..., n

ProtoSet ← ∅

i ← 0

P2: Comenzar procesamiento del objeto O_i.

i ← índice del primer objeto no usado

Si i=n entonces fin de la generación de prototipos.

P3: Construir la clase de similitud de objetos O_i acorde a R.

$[O_i]R$ denota la clase de similitud del objeto O_i

P4: Construir un vector P con m componentes para todos los objetos en $[O_i]R$.

P(i) ← $f(V_i)$, donde V_i es el conjunto de valores de los rasgos i en objetos en $[O_i]R$ y f es un operador de agregación.

ProtoSet ← ProtoSet ∪ P

P5: Marcar como usado todos los objetos en la clase de similitud $[O_i]R$.

Usado [j] ← 1 para todo $O_j \in [O_i]R$

P6: Ir a P2

En el paso P4 la función f denota un operador de agregación, por ejemplo: Si el valor en Vi es real, el promedio puede ser usado; si son discretos, el valor más común. El propósito es construir un prototipo o centroide para un conjunto de objetos similares.

El conjunto de prototipos ProtoSet es la salida del algoritmo NPBASIR-C. Este conjunto es usado por el clasificador para clasificar nuevas instancias:

Algoritmo NPBASIR-CLASS (entrada: ProtoSet, x, salida: class)

Dado un Nuevo objeto x y el conjunto ProtoSet.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 4, Octubre-Diciembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 83-96

P1: Calcular la similitud entre x y cada prototipo.

P2: Seleccionar los k prototipos más similar a x.

P3: Calcular la clase de x como la clase más probable en el conjunto de k prototipos más similares.

En el paso 3 la clase se calcula de igual forma que en k-vecinos más similares (k-SN) para la clasificación.

Método NPBASIR-Class-RED

El método NPBASIR-ClassRED es el resultado de realizar la selección de atributos utilizando el método REDUCT-SIM, explicado anteriormente y con el subconjunto de atributos obtenidos realizar la generación de prototipos utilizando NPBASIR-Class.

Con el fin de evaluar la exactitud de la propuesta fue realizada el siguiente estudio experimental. Se utilizaron 10 conjuntos de datos provenientes del depósito de datos para aprendizaje automatizado disponibles en el sitio ftp de la Universidad de Irvine, California[1], donde la mayoría de los atributos en A tienen dominio continuo y el rasgo d discreto. La tabla 1 resume las propiedades de los conjuntos de datos seleccionados.

Los conjuntos de datos considerados fueron particionados utilizando el procedimiento K-fold cross-validation. (Demsar, 2006). Este método, subdivide el conjunto de datos original en K (10) subconjuntos de igual tamaño, uno de los cuales se utiliza como conjunto de prueba mientras los demás forman el conjunto de entrenamiento. Luego la exactitud general del clasificador es calculada como el promedio de las precisiones obtenidas con todos los subconjuntos de prueba.

Para calcular la exactitud de los métodos se utilizó la métrica Accuracy, típicamente usada en estos casos por su simplicidad y aplicación exitosa (Witten & Frank, 2005).

Para el análisis estadístico de los resultados se utilizaron las técnicas de prueba de hipótesis (S. García & Herrera, 2009), (Sheskin, 2003). Para comparaciones múltiples, se utilizan las pruebas de Friedman y de Iman-Davenport (Iman & Davenport, 1980) para detectar diferencias estadísticamente significativas entre un grupo de resultados. Se emplea además la prueba de Holm (Holm, 1979) con el fin de encontrar los algoritmos significativamente superiores.

Estas pruebas son sugeridas en los estudios presentados en (Demsar, 2006), (S. García & Herrera, 2008), (S. García & Herrera, 2009) (S. García, et al., 2010), donde se afirma que el uso de estas pruebas es muy recomendable para la validación de resultados en el campo del aprendizaje automatizado.

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

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 4, Octubre-Diciembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 83-96

Tabla 1. Descripción de los conjuntos de datos

Conjuntos de datos	Cantidad de rasgos	Cantidad de objetos	Tipos de datos
Iris	4	150	Continuos
Pima	8	768	Continuos
Biomed	8	194	Continuos y discretos
Cleveland	13	297	Continuos
Mfeat-zernike	47	2000	Continuos
Mfeat-fourier	76	2000	Continuos
Mfeat-factor	216	2000	Continuos
Pendigitits	16	10992	Continuos
Wave form	40	5000	Continuos
Optdigits	64	5620	Continuos

Resultados y discusión

En la Tabla 2 se muestran los resultados de la comparación entre el algoritmo NPBASIR-Class, NPBASIR-ClassRED, y los algoritmos LVQ3, GENN, PSCSA, VQ, MSE,ENPC,AVQ, Chen e HYB, que según el estudio realizado en (Triguero I et al., 2011) obtienen los mejores resultados entre los clasificadores analizados. Estos algoritmos están implementados en la herramienta KEEL (Alcalá, 2008). Los conjuntos de datos utilizados se renombraron con la terminación –red, con el propósito de demostrar que fueron las bases reducidas las que se utilizaron en la experimentación, excepto para el algoritmo NPBASIR-Class, que fueron los conjuntos originales.

Tabla 2. Comparación del resultado de la clasificación para los métodos de generación de prototipos utilizando los conjuntos de datos reducidos.

Conjuntos de datos	NPBASIR-Class	NPBASIR-ClassRED	LVQ3	GENN	PSCSA	VQ	MSE	ENPC	AVQ	Chen	HYB
Iris-red	96.42	**96.50**	96.00	95.33	95.33	86.66	96.66	94.00	94.66	95.33	96.00
Pima-red	**75.38**	73.00	73.7	73.56	75.39	79.42	72.92	63.54	72.27	72.78	61.85
Biomed-red	**95.97**	91.03	82.5	81.55	82.52	79.42	80.94	81.55	82.52	78.31	77.84
Cleveland-red	**59.24**	54.13	43.41	49.85	50.16	42.08	48.51	45.11	30.93	48.49	33.94
Mfeat-zernike-red	**81.05**	78.04	70.51	68.97	69.03	73.84	70.8	67.85	70.89	70.31	69.54
Mfeat-fourier-red	**81.50**	79.07	69.8	70.78	69.34	68.75	69.97	67.97	70.80	71.65	69.23
Mfeat-factor-red	**96**	95.14	94.53	93.55	93.67	90.12	96.56	94.13	94.57	94.98	90.33

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 4, Octubre-Diciembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 83-96

Pendigits-red	**95.88**	93.34	93.02	92.08	93.45	89.76	93.02	92.56	92.55	93.30	92.68
Wave form-red	**83.78**	79.86	73.40	73.78	74.98	70.68	79.90	74.87	73.21	73.45	73.02
Optdigits-red	**97.47**	94.04	95.67	94.21	94.67	90.35	93.04	93.67	92.74	93.13	93.91

En la Tabla 3 se puede observar que el mejor ranking lo tiene el método NPBASIR-Class, seguido por el NPBASIR-ClassRED, demostrando que utilizando selección de atributos antes de la clasificación los resultados son satisfactorios, teniendo en cuenta que se disminuye el tiempo de ejecución (como se muestra en la Tabla 4) y la cantidad de prototipos encontrados es menor (como se muestra en la Tabla 5), sin variar notablemente la calidad de la clasificación. La cantidad de prototipos generados por los algoritmos NP-BASIR-Class y NP-BASIR-ClassRED fue calculada basada en el promedio de la cantidad de prototipos generados para cada partición.

El tiempo de ejecución que se muestra en la Tabla 4, en la columna NPBASIR-Class fue el tiempo de ejecución de los conjuntos de datos sin la selección de atributos y en la columna NPBASIR-ClassRED, el tiempo de ejecución de los conjuntos de datos con la selección de atributos, dado en segundos.

En la tabla 5 también se observa el coeficiente de reducción Re (·) (Bermejo & Cabestany, 2000), esta medida que se muestra en la expresión (2), indica la cantidad de objetos que fue reducida (relación entre la cantidad de prototipos determinados utilizando todos los atributos(P) y la cantidad de prototipos determinados utilizando selección de atributos(X)).

$$\mathrm{Re}(.) = \frac{|X| - |P|}{|X|} * 100 \qquad (2)$$

En la tabla 5, la quinta columna muestra el coeficiente de reducción del algoritmo NPBASIR-ClassRED respecto al conjunto original de datos y la columna 6 muestra el coeficiente de reducción del algoritmo NPBASIR-ClassRED respecto al algoritmo NPBASIR-Class.

Tabla 3. Resultados del Test de Friedman.

Algoritmos	**Ranking**
NPBASIR-Class	1.5
NPBASIR-ClassRED	3.2
LVQ3	5.45
GENN	6.35
VQ	6.2
MSE	8.6
ENPC	5.8
AVQ	8.05

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 4, Octubre-Diciembre, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 83-96

PSCSA	6.15
Chen	6.4
HYB	8.3

Tabla 4. Comparación entre los tiempos de ejecución de los algoritmos NPBASIR-Class y NPBASIR-ClassRED.

Conjuntos de datos	**NPBASIR-Class/ Tiempo de ejecución**	**NPBASIR-ClassRED/ Tiempo de ejecución**
Iris-red	0.309	**0.240**
Pima-red	0.187	**0.141**
Biomed-red	0.436	**0.295**
Cleveland-red	0.888	**0.381**
Mfeat-zernike-red	28.173	**20.864**
Mfeat-fourier-red	41.367	**36.978**
Mfeat-factor-red	32.756	**26.7965**
Pendigits-red	110.219	**98.437**
Wave form-red	134.707	**125.075**
Optdigits-red	138.214	**129.492**

Tabla 5. Cantidad de prototipos encontrados por los algoritmos NPBASIR-Class y NPBASIR-ClassRED.

Conjuntos de datos	**Cantidad de objetos**	**Cantidad de prototipos por NPBASIR-Class**	**Cantidad de prototipos por NPBASIR-ClassRED**	**Coeficiente de reducción**	**Coeficiente de reducción**
Iris-red	150	**7**	9.2	93.87	31.42
Pima-red	768	146.3	**26**	96.61	82.22
Biomed-red	194	31.4	**6.6**	96.60	78.98
Cleveland-red	297	96.8	**27.9**	90.61	71.17
Mfeat-zernike-red	2000	355.9	**159.8**	92.01	55.09
Mfeat-fourier-red	2000	355.9	**157.4**	92.13	55.77
Mfeat-factor-red	2000	330.6	**214.9**	89.26	34.99
Pendigits-red	10992	355.8	**256.6**	97.67	27.88
Wave form-red	5000	41.7	**22.3**	99.55	46.52
Optdigits-red	5620	422.6	**321.8**	94.27	23.85

CONCLUSIONES

En este artículo, se analizaron los resultados obtenidos al comparar la propuesta con otras técnicas de generación de prototipos y se puede concluir que: La combinación de un método de selección de atributos con un método de generación de prototipos, favorece la obtención de mejores resultados. El método NPBASIR-ClassRED permite disminuir el

tiempo de ejecución y la cantidad de prototipos en comparación con otros métodos, sin variar notablemente la calidad de la clasificación.

Referencias

ALCALÁ, J., et al. (2008). KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems. *Soft Computing, 13*(3), 307-318.

ARAÚZO, A. (2006). *Un sistema inteligente para selección de características en clasificación.* (Tesis Doctoral), Universidad de Granada, Granada, España.

BARANDELA, R. (2001). *The nearest neighbour rule and the reduction of the training sample size* Paper presented at the 9th Symposium on Pattern Recognition and Image Analysis, Castellon, España.

BELLO-GARCIA, M., GARCIA, M. M., & BELLO, R. (2013). *A Method for Building Prototypes in the Nearest Prototype Approach Based on Similarity Relations for Problems of Function Approximation.* Paper presented at the MICAI.

BERMEJO, S., & CABESTANY, J. (2000). A Batch Learning Algorithm Vector Quantization Algorithm for Nearest Neighbour Classification. *Neural Processing Letters,, 11*, 173–184.

BEZDEK, J. C., & KUNCHEVA, L. I. (2001). Nearest Prototype classifiers design: an experimental study. *Int'l J. Intelligent Systems, 16*, 1445-1473.

BIEN, J., & TIBSHIRANI, R. (2012). Prototype selection for interpretable classification. *Annals of Applied Statistics, 5*, 2403-2424.

Ch., Y., D., M., R., W., & K., W. (2011). A rough set approach to feature selection based on power set tree. *Knowledge-Based Systems, 24*(2), 275–281.

CHANDRASHEKAR, G., & SAHIN, F. (2014). A survey on feature selection method. *Computers and Electrical Engineering, 40*, 16–28.

DEMSAR, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*(7), 1-30.

ESPINILLA, M., QUESADA, F. J., MOYA, F., MARTINEZ, L., & NUGENT. (2015). Reducing the response time for activity recognition through use of prototype generation algorithms. *LNCS,, 9102*, 313-318.

FERNÁNDEZ, F., & ISASI, P. (2004). Evolutionary design of nearest prototype classifiers. *J. Heurist, 10*(4), 431–454.

FERNÁNDEZ, Y., BELLO, R., FILIBERTO, Y., CABALLERO, Y., & FRÍAS, M. (2013). Effects of using reducts in the performanceof the IRBASIR algoritmth. *Revista DYNA, 182*, 182-190.

FERNÁNDEZ, Y., BELLO, R., FILIBERTO, Y., FRÍAS, M., COELLO, L., & CABALLERO, Y. (2015). An Approach For Prototype Generation Based On Similarity Relations For Problems Of Classification. *Computación y Sistemas, 19*(1), 109–118.

FILIBERTO, Y., BELLO, R., CABALLERO, Y., & LARRUA, R. (2010). Using PSO and RST to Predict the Resistant Capacity of Connections in Composite Structures. *In International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO 2010) Springer*, 359-370. doi: 10.1007/978-3-642-12538-6_30

FILIBERTO, Y., BELLO, R., CABALLERO, Y., & LARRUA, R. (2010b, Nov. 19-Dec. 1). *A method to built similarity relations into extended Rough set theory.* Paper presented at the Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA2010), Cairo, Egipto.

GARCÍA-DURAN, R., F., F., & BORRAJO, D. (2010). A prototype-based method for classification with time constraints: a case study on automated planning. . *Pattern Anal. Applic*. doi: 10.1007/s10044-010-0194-6

GARCÍA, S., et al. (2010). 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.

GARCÍA, S., & HERRERA, F. (2008). An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research, 9*, 2677-2694.

GARCÍA, S., & HERRERA, F. (2009). Evolutionary under-sampling for classification with imbalanced data sets: Proposals and taxonomy. *Evolutionary Computation, 17*(3), 275-306.

HOLM, S. (1979). A simple sequentially rejective multiple test procedure. *Journal of Statistics, 6*, 65-70.

IMAN, R., & DAVENPORT, J. (1980). Approximations of the critical region of the friedman statistic. *Communications in Statistics, Part A Theory Methods, 9*, 571-595.

JIANG, S., PANG, G., WU, M., & KUANG, L. (2012). An improved K-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications, 39*(1), 1503–1509.

KOPLOWITZ, J., & BROWN, T. A. (1981). On the relation of performance to editing in nearest neighbor rule. *Pattern Recognition, 13*(3), 251-255.

LEÓN, E., J., G., & GIRALDO, F. (2012). *Online Cluster Prototype Generation for the Gravitational Clustering Algorithm Advances in Artificial Intelligence.* . Paper presented at the IBERAMIA 2012.

LIU, H., & MOTODA, H. (2007). Computational Methods of Feature Selection. *Data Mining and Knowledge Discovery Series*.

NANNI, L., & LUMINI, A. (2008). Particle swarm optimization for prototype reduction. *Neurocomputing, 72*(4–6), 1092–1097.

PATANÉ, G., & RUSSO, M. (2001). The Enhanced LBG Algorithm. *Neural Networks, 14*, 1219–1237.

S.W. KIM, J. O. (2003). A Brief Taxonomy and Ranking of Creative Prototype Reduction Schemes. *Pattern Analysis and Applications, 6*, 232-244.

SÁNCHEZ, J. S. (2004). High training set size reduction by space partitioning and prototype abstraction. *Pattern Recognit., 37*(7), 1561–1564.

SHESKIN, D. (2003). *Handbook of parametric and nonparametric statistical procedures, chapman & hall*. Paper presented at the *CRC*.

TRIGUERO I, DERRAC J, GARCIA S., & F., H. (2011). A Taxonomy and Experimental Study on Prototype Generation for Nearest Neighbor Classification. *EEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, do i: 0.1109/TSMCC.2010.2103939, in press*.

TRIGUERO, I., DERRAC, J., GARCÍA, S., & HERRERA, F. (2012). Integrating a Differential Evolution Feature Weighting scheme into Prototype Generation. *Neurocomputing, 97*, 332-343.

W.LAM, C.K.KEUNG, & ., D. L. (Aug. 2002). Discovering useful concept prototypes for classification based on filtering and abstraction. *Pattern Anal. Mach. Intell, 14*, 1075–1090.

WITTEN, I., & FRANK, E. (Eds.). (2005). *Data Mining. Practical Machine Learning Tools and Techniques* (Second Edition ed.): Department of Computer Science. University of Waikato.