Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 134-148

Tipo de artículo: Artículo original
Temática: Inteligencia Artificial
Recibido: 23/06/2016 | Aceptado: 23/01/2017

Estudio del comportamiento de métodos basados prototipos y en relaciones de similitud ante "hubness"

A study of the behavior of methods based on prototypes and similarity relations in the face of "hubness"

Yanela Rodríguez Alvarez [1*], Rafael Bello Pérez [2], Yailé Caballero Mota [1], Yaima Filiberto Cabrera[1], Yumilka Fernández Hernández[1], Mabel Frías Hernández[1]

[1] Departamento de Computación, Universidad de Camagüey, Circunvalación Norte Km 5 ½, Camagüey, Cuba. {yanela.rodriguez, yaile.caballero, yaima.filiberto, yumilka.fernandez, mabel.frias}@reduc.edu.cu
[2] Departamento de Ciencias de la Computación, Universidad Central "Marta Abreu" de las Villas. Carretera a Camajuaní Km. 5 y ½, Santa Clara, Villa Clara, Cuba. rbellop@uclv.edu.cu

[*] Autor para correspondencia: yanela.rodriguez@reduc.edu.cu

Resumen

El fenómeno *hubness*, es un aspecto del curso de la dimensionalidad descrito recientemente, que está relacionado con la disminución de la especificidad de las similitudes entre los puntos en un espacio de alta dimensión; lo cual va en detrimento de los métodos de aprendizaje automático. En este trabajo se evalúa el impacto del fenómeno *hubness* en la clasificación utilizando un enfoque basado en prototipos. El estudio experimental realizado demuestra que los métodos de generación y selección de prototipos estudiados ofrecen resultados comparables contra otros métodos basados en el enfoque kNN, encontrados en la literatura, los cuales son *hubness*-consientes y están diseñados específicamente para lidiar con este problema. Teniendo en cuenta los resultados alentadores de este estudio y las bondades de los métodos basados en prototipos es posible asegurar que la utilización de los mismos permitirá mejorar el desempeño de los sistemas que manejen datos de altas dimensiones y bajo la asunción de *hubness*.

Palabras Claves: *Hubness*, selección de prototipos, generación de prototipos, relaciones de similitud, clasificación.

Abstract

The hubness phenomenon, is an aspect of the curse of dimensionality recently described, that is related to the diminishment of specificity in similarities between points in a high-dimensional space; which is detrimental to the machine learning methods. This paper deals with evaluating the impact of hubness phenomenon on classification based

on the nearest prototype. Experimental results show that the studied methods of generation and selection of prototypes offer comparable results against others methods based on kNN approach, found in the literature, which are hubness aware and are specifically designed to deal with this problem. Based on these encouraging results and the extensibility of methods based on prototypes, it is possible argue that it might be beneficial to use them in order to improve system performance in high dimensional data under the assumption of hubness.

Keywords: *Hubness, prototype selection, prototype generation, similarity relations, classification.*

Introducción

Los datos de alta dimensión plantean muchos desafíos intrínsecos para los problemas de reconocimiento de patrones como por ejemplo la maldición de la dimensionalidad que incluye la escasez, la redundancia, la concentración de las distancias, las estimaciones de densidad problemáticas, los vecinos más próximos menos significativos (Bolei, Khosla, Lapedriza, Oliva, & Torralba, 2015; Gao, Song, Nie, et al., 2015; Nguyen, Yosinski, & Clune, 2015; B. Yao, Khosla, & Fei-Fei, 2011; Zhou et al., 2015); así como la disminución de la especificidad de las similitudes entre los puntos en un espacio de alta dimensión (Choi, Pantofaru, & Savarese, 2013; Khosla, An, Lim, & Torralba, 2014; Wiskott, 2013). Específicamente, la complejidad de muchos algoritmos de minería de datos existentes es exponencial con respecto al número de dimensiones (Gao, Song, Liu, et al., 2015). Con el aumento de la dimensionalidad, los algoritmos existentes pronto se convierten en computacionalmente intratables y por lo tanto inaplicables en muchas aplicaciones reales (Gao, Song, Liu, et al., 2015).

Una alternativa para mitigar la maldición de la dimensionalidad, que implica la reducción del número de ejemplos a tener en cuenta, es la clasificación utilizando un enfoque basado en prototipos (*Nearest Prototype*, NP) (Bezdek & Kuncheva, 2001). La idea es determinar el valor del rasgo de decisión de un nuevo objeto analizando su similitud respecto a un conjunto de prototipos seleccionado o generado a partir del conjunto inicial de instancias. El propósito del enfoque NP es decrecer los costos de almacenamiento y procesamiento de las técnicas de aprendizaje basadas en instancias. Para reducir la cantidad de instancias, hay dos estrategias: Selección y Remplazo (Jin, Liu, & Hou, 2010). En el primer caso, se conserva una cantidad limitada de instancias del conjunto de datos original. En la segunda alternativa, se reemplaza el conjunto de datos originales por un número de prototipos que no necesariamente coincide con alguna instancia original.

El otro desafío que se deriva del aumento de la dimensionalidad de los datos y que está relacionado con la disminución de la especificidad de las similitudes entre los puntos en un espacio de alta dimensión es el conocido como: fenómeno *hubness*, y que se ha descrito recientemente. Este fenómeno consiste en la observación de que al aumentar la dimensionalidad de un conjunto de datos la distribución del número de veces que un punto de datos se encuentra entre los k vecinos más cercanos de otros puntos de datos se convierte en cada vez más sesgada a la derecha. Como consecuencia, los llamados "*hubs"* emergen, es decir, los puntos de datos que aparecen en las listas de los k vecinos más cercanos de otros puntos de datos con mucha mayor frecuencia que otros. (Low, Borgelt, Stober, & Nürnberger, 2013)

Los métodos de construcción y selección de prototipos no han sido estudiados bajo el supuesto de la presencia del fenómeno de *hubness* en los datos y su impacto en el aprendizaje de estos métodos. Por otro lado, El enfoque más intuitivo de la clasificación de patrones está basado en el concepto de similitud (Duda, Hart, & Stork, 2001; Weinberger & Saul, 2009); obviamente patrones que son similares en algunos sentidos tienen asignada la misma clase.

El objetivo de este trabajo es analizar experimentalmente cómo se comportan los métodos de construcción y selección de prototipos basados en relaciones de similitud **NP-BASIR-CLASS** y **NPBASIR SEL-CLASS** ante conjuntos de entrenamiento que tengan "*hubness*"; comparándolos con métodos basados en kNN, enfoque más estudiado hasta el momento, pero que son "*hubness*-consientes", es decir, que tienen en cuenta la presencia de los "malos *hubs*" y utilizan diferentes mecanismos para eliminarlos o mitigar sus efectos negativos en la clasificación.

El algoritmo **NP-BASIR-CLASS** (Fernández Hernández et al., 2015) es comparado en este propio articulo con 12 algoritmos mencionados en el artículo "*Una taxonomía y un estudio experimental sobre la Generación de Prototipos para la Clasificación del vecino más cercano*" (Triguero, Derrac, Garcia, & Herrera, 2012). Estos algoritmos son los que ofrecen mejores resultados según (Triguero et al., 2012). Los resultados experimentales muestran que el método NP-BASIR-CLASS tiene el mejor ranking y es estadísticamente superior a los otros en términos de precisión de la clasificación. El factor de reducción y el tiempo de ejecución alcanzado también fueron analizados y muestran resultados satisfactorios superiores en el caso del algoritmo NP-BASIR-CLASS (Fernández Hernández et al., 2015).

Por su parte, el algoritmo **NPBASIR SEL-CLASS** (Frias, Filiberto, Fernández, Caballero, & Bello, 2015) es comparado con los 4 algoritmos, mencionados en el artículo "Selección de Prototipos para la Clasificación del Vecino más Cercano: una Taxonomía y un Estudio Empírico" (Garcia, Derrac, Cano, & Herrera, 2012), que ofrecen mejores resultados. De acuerdo con los resultados presentados en este trabajo, la propuesta es estadísticamente significativa a los métodos de RMHC, SSMA, HMNEI RNG y ofrece resultados comparables con NPBASIR-CLASS en cuanto a la

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 134-148

precisión de la clasificación. También se analiza el factor de reducción lograda, mostrando resultados satisfactorios superiores para el método NPBASIR SEL-CLASS (Frias et al., 2015).

Materiales y Métodos

En la clasificación supervisada, la reducción de datos es una tarea importante, especialmente para clasificadores basados en instancias porque a través de ella los tiempos de ejecución se pueden reducir y obtener igual o mejor exactitud en la clasificación.

En (Fernández Hernández et al., 2015) y (Frias et al., 2015) se proponen dos métodos para la generación y selección de prototipos respectivamente que muestran buenos resultados. Estos trabajos proponen utilizar el método NPBASIR-CLASS y NPBASIR SEL-CLASS para construir y seleccionar los prototipos respectivamente para problemas de clasificación empleando los conceptos de Computación Granular (Y. Yao, 2000) basada en NPBASIR (Bello-García, García-Lorenzo, & Bello, 2012).

La granulación de un universo se realiza usando una relación de similitud que genera clases de similitud de objetos en el universo, y para cada clase de similitud se construye/selecciona un prototipo. Para construir la relación de similitud se utiliza el método propuesto en (Filiberto, Caballero, Larrua, & Bello, 2010).

Método de construcción de prototipos

El método propuesto en (Fernández Hernández et al., 2015) 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 se construye usando la relación de similitud R ($[Oi]R$) y para esta clase de similitud se construye un prototipo. 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. Se utiliza un arreglo de n componentes, 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 $X = \{X1, X2 \ldots Xn\}$ cada una de las cuales es descrita por un vector de a atributos descriptivos y pertenece a una de k clases $w = \{w1, w2, \ldots, wk\}$ y una relación de similitud R. La relación de similitud R es construida acorde al método propuesto en (Filiberto, Bello, Caballero, & Larrua, 2010; Filiberto, Caballero, et al., 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.

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 134-148

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.

$Usado\ [j]\ \leftarrow\ 0, para\ j = 1, \ldots, n$

$ProtoSet\ \leftarrow\ \emptyset$

$i\ \leftarrow\ 0$

P2: Comenzar procesamiento del objeto O_i.

$i\ \leftarrow\ índice\ del\ primer\ objeto\ no\ usado$

$S_i\ i = n\ entonces\ fin\ de\ la\ generación\ de\ prototipos.$

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 $[O_i]R$.

$P(i) \leftarrow\ f(V_i)$,

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.

$ProtoSet\ \leftarrow\ ProtoSet\ \cup\ P$

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

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

P6: Ir a P2 26

En el paso P4 la función f denota un operador de agregación, por ejemplo: Si el valor en V_i es real, se puede utilizar el promedio; si son discretos, el valor más común. El propósito es construir un prototipo o centróide 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.

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 para la selección de prototipos

El método propuesto en (Frias et al., 2015) es un proceso iterativo en el cual los prototipos son seleccionados 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 seleccionado para esta clase de similitud.

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 134-148

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

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

P1: Inicializar contador de objetos.

$Usado\ [j]\ \leftarrow\ 0, para\ j = 1, \ldots, n$

$ProtoSet\ \leftarrow\ \emptyset$

$i\ \leftarrow\ 0$

P2: Comenzar procesamiento del objeto O_i.

$i\ \leftarrow\ índice\ del\ primer\ objeto\ no\ usado$

$S_i\ i = n\ entonces\ Fin\ de\ la\ selección\ de\ prototipos.$

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: Calcular para cada O_i de $[O_i]R$ el grado de similitud con respecto al resto de O_J, tal como expresa la Ecuación 1:

$$S[O_i] = \frac{\sum_{j=1}^{n} F_1\left(O_i, O_j\right)}{n} \qquad i \neq j \tag{1}$$

Donde n denota la cantidad de objetos por clase de similitud, $S[O_i]$ es el grado de similitud para cada objeto O_i con respecto al resto de los objetos O_j para cada $[O_i]R$.

P5: Ordenar los objetos de mayor a menor $S[O_i]$ por cada $[O_i]R$.

Construir el conjunto de prototipos con el O_i de $+\ S[O_i]$.

$ProtoSet\ \leftarrow\ ProtoSet\ \cup Oi$

P6: Marcar como usado todos los objetos en la clase de similitud $[O_i]R$.

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

P7: Ir a P2

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

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

Dado un Nuevo objeto x y el conjunto ProtoSet.

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.

Clasificación *hubness*-consiente

La clasificación *hubness* consiente está apoyada en el modelo de aprendizaje basado en la ocurrencia de los vecinos más cercanos en los datos de entrenamiento.

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 134-148

Luego, $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ sea el conjunto de entrenamiento de puntos de datos etiquetados dibujado i.i.d[1]. a partir de una distribución conjunta $p(x, y) = p(x) \cdot p(y|x)$ sobre $X \times Y$, donde X es el espacio de rasgos y Y el espacio finito de etiquetas, $|Y| = C$.

Denotamos por $D_k = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \dots, (x_{ik}, y_{ik})\}$ a los k-vecinos más cercanos de x_i. Para todo $x \in D_k(x_i)$ es un vecino de x_i y viceversa, x_i es vecino de todo $x \in D_k(x_i)$. Una ocurrencia de un elemento en algunos $D_k(x_i)$ se conoce como una k-ocurrencia. El número de k-ocurrencias de un punto x se denota por $N_k(x)$. Una k-ocurrencia se considera buena si la etiqueta de su vecino coincide con la etiqueta del punto de interés, por ejemplo $x_{ij} \in D_k(x_i)$ es una buena ocurrencia de x_{ij} si $y_{ij} = y_i$. Del mismo modo, los desajustes de etiquetas corresponden a las malas ocurrencias de puntos vecinos. El cómputo total de ocurrencias consiste en una suma de las buenas y malas ocurrencias, $N_k(x_i) = GN_k(x_i) + BN_k(x_i)$, donde $GN_k(x_i)$ y $BN_k(x_i)$ representa las buenas ocurrencias y las malas respectivamente. El conteo de las buenas y malas ocurrencias es también conocido como los buenos o malos *hubness* de un punto en particular. También es posible considerar la ocurrencia de la clase condicional que se denota como $N_{k,c}(x_i)$ y representa el número de k-ocurrencias de x_i en el vecindario de ejemplos que pertenecen a la clase c.

En grandes volúmenes de datos, la distribución $N_k(x)$ es altamente asimétrica, en el sentido de que es sesgada a la derecha. La asimetría de la distribución de la frecuencia de ocurrencia de los k-vecinos se define como se expresa en la Ecuación 2:

$$SN_k(x) = \frac{m_3(N_k(x))}{m_2^{3/2}(N_k(x))} = \frac{1/N \sum_{1=1}^{N}(N_k(x_i) - k)^3}{(1/N \sum_{1=1}^{N}(N_k(x_i) - k)^2)^{3/2}} \quad (2)$$

Formalmente, se dice que los *hubs* son puntos $x_h \in D$ tales que $N_k(x_h) > k + 2 \cdot \sigma_{N_K(x)}$. En otras palabras, sus frecuencias de ocurrencia exceden la media (k) por más del doble de la desviación estándar. Finalmente se denota el conjunto de todos los *hubs* en T por H_k^T.

Con el objetivo de aliviar la negativa influencia de los malos *hubs* en los datos y de obtener una robusta clasificación del k-vecino más cercano bajo el supuesto de *hubness*, recientemente se han propuesto varios métodos kNN-*hubness*-consientes: *hubness*-weighted kNN (hw-kNN) (Radovanović, Nanopoulos, & Ivanović, 2009), *hubness*-fuzzy kNN (h-FNN) (Tomašev, Radovanović, Mladenić, & Ivanović, 2014), *hubness*-information kNN (HIKNN) (Tomašev & Mladenić, 2012) y Naive *Hubness*-Bayesian kNN (NHBNN) (Tomasev, Radovanović, Mladenić, & Ivanović, 2011).

[1]Independientes e idénticamente distribuidos

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 134-148

hw-kNN: Este algoritmo de ponderación es la forma más sencilla de reducir la influencia de malos *hubs* donde simplemente se le asignan pesos de voto inferiores. El voto de cada vecino se pondera por $e^{-h_b(x_1)}$, donde $h_b(x_1)$ es la mala puntuación *hubness* estandarizada del vecino. Todos los vecinos votan por su propio sello (a diferencia de los algoritmos considerados más adelante), lo que podría resultar perjudicial en algunas ocasiones.

h-FNN: $\boldsymbol{u_c(x_i) = \frac{N_{k,c}(x_i)}{N_k(x_i)}}$ (clase *hubness* relativa) puede interpretarse como la fuzificación del evento de que $\boldsymbol{x_i}$ haya ocurrido como un vecino. Por esta razón, h-FNN integra la clase *hubness* en framework difuso de k-vecinos más cercanos de votación (Keller, Gray, & Givens, 1985). Esto significa que las probabilidades de la etiqueta en el punto de interés se calculan como se expresa en la Ecuación 3:

$$\boldsymbol{u_c(x_i) = \frac{\sum_{x_i \in D_k(x)} u_c(x_i)}{\sum_{x_i \in D_k(x)} \sum_{c \in C} u_c(x_i)}} \tag{3}$$

NHBNN: Cada k-ocurrencia puede ser tratada como un evento aleatorio. Lo que NHBNN hace esencialmente es que realiza una inferencia bayesiana (*Naive-Bayesian inference*) para estos *k* eventos tal como se expresa en la Ecuación 4.

$$p(y_i = c | D_K(x_i)) \propto p(y_i = c) \prod_{t=1}^{K} P(x_{it} \in D_k(x_i) | y_i = c) \tag{4}$$

Aun cuando las k-ocurrencias están altamente correlacionados, NHBNN todavía ofrece algunas mejoras sobre el kNN básico.

HIKNN: Recientemente, la clase *hubness* también ha sido explorada con un enfoque de la teoría de la información aplicada a la clasificación de los k- vecinos más cercanos. Las ocurrencias raras tienen mayor auto-información (Ecuación 5) y son favorecidos por el algoritmo. Los *hubs*, por el contrario, se encuentran más cerca de los centros de grupos y llevan menos información local relevante para la consulta en particular.

$$pp(x_{it} \in D_k(x)) \approx \frac{N_k(x_{it})}{N}$$
$$I_{x_{it}} = \log \frac{1}{p(x_{it} \in D_k(x))} \tag{5}$$

La auto-información de la ocurrencia se utiliza para definir los factores de relevancia absolutos y relativos como se expresa en la Ecuación 6:

$$\alpha(x_{it}) = \frac{I_{x_{it}} - min_{x_j \in I_{x_j}}}{\log n - min_{x_j \in I_{x_j}}}, \beta(x_{it}) = \frac{I_{x_{it}}}{\log N} \tag{6}$$

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 134-148

La votación difusa final combina la información contenida en la etiqueta del vecino con la información contenida en su perfil de ocurrencia. El factor de relevancia relativa se utiliza para ponderar las dos fuentes de información. Esto se muestra en la Ecuación 7:

$$\bar{p}_k\left(y_i = c \middle| x_{it} \in D_k(x_i)\right) = \frac{N_{k,c}(x_{it})}{N_k(x_{it})} = \bar{p}_{k,c}(x_{it})$$
$$p_k(y_i = c|x_{it}) \approx \begin{cases} \alpha(x_{it}) + \left(1 - \alpha(x_{it})\right) \cdot \bar{p}_{k,c}(x_{it}),\ y_{it} = c \\ \left(1 - \alpha(x_{it})\right) \cdot \bar{p}_{k,c}(x_{it}), y_{it} \neq c \end{cases} \quad (7)$$

La asignación de la clase final está dada por la suma ponderada de estos votos difusos, como se muestra en la Ecuación 8. El factor de ponderación $d_w(x_{it})$ produce en su mayoría mejoras poco significativas y puede no tomarse en cuenta en la práctica.

$$u_c(x_i) \propto \sum_{t=1}^{k} \beta(x_{it}) \cdot d_w(x_{it}) \cdot p_k(y_i = c|x_{it}) \quad (8)$$

NHBNN, HIKNN y h-FNN utilizan estimaciones de frecuencia de la ocurrencia de la clase condicional para realizar una clasificación basada en los modelos de los vecinos más cercanos. En los datos de alta dimensión, esto podría ser un poco mejor que la votación por la etiqueta.

Resultados y Discusión

Para calcular la exactitud de los métodos se utilizó la métrica $Accuracy$, típicamente usada en estos casos por su simplicidad y aplicación exitosa y que se muestra en la expresión (9) dada en (Witten & Frank, 2005):

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (9)$$

Donde *TP* son los verdaderos positivos, *TN* los verdaderos negativos, *FP* son los falsos positivos, y *FN* son los falsos negativos.

Para el análisis estadístico de los resultados se utilizaron las técnicas de prueba de hipótesis (García & Herrera, 2009). Para comparaciones múltiples, se utilizan las pruebas de Friedman y de Iman-Davenport para detectar diferencias estadísticamente significativas entre un grupo de resultados. Se emplea además la prueba de Holm con el fin de encontrar los algoritmos significativamente superiores (Alcalá et al., 2010). Estas pruebas son sugeridas en los estudios

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 134-148

presentados en (Demšar, 2006; García, Fernández, Luengo, & Herrera, 2010), donde se afirma que el uso de estas pruebas es muy recomendable para la validación de resultados en el campo del aprendizaje automatizado.

La exactitud de la clasificación dada en la Tabla 2 para los métodos **kNN**, **hw-kNN**, **h-FNN**, **NHBNN** y **HIKNN** ha sido reportada en trabajos anteriores (Tomašev & Mladenić, 2012, 2014) y servirá como un punto de partida para su posterior análisis. Todos los algoritmos *hubness*-conscientes se probaron bajo sus configuraciones de parámetros por defecto, de acuerdo a lo especificado en sus respectivos artículos.

Para medir el desequilibrio de clases en un conjunto de datos concreto, se observan dos cantidades: $p(c_M)$, que es el tamaño relativo de la clase mayoritaria y el desequilibrio relativo de la distribución de las etiquetas ($RImb$), que se define como la desviación estándar normalizada de la probabilidad de la clase respecto al valor absoluto medio homogéneo de 1 / c para cada clase. En otras palabras, según la Ecuación 10:

$$RImb = \sqrt{\left(\sum_{c \in C} (p(c) - 1/C)^2\right) / ((C-1)/C))} \quad (10)$$

Los malos *hubs* no son causados directamente por el desbalance de clases, pero si son resultado de la interrelación de varios factores.

En la Tabla 1 se resumen las características de los conjuntos de datos del mundo real utilizados en la experimentación (Conjuntos de datos del Repositorio de Aprendizaje Automático de la UCI[2]). Cada conjunto de datos esta descrito por las siguientes propiedades: número de ejemplos (e), número de rasgos (d), número de clases (c), asimetría de la distribución de la frecuencia de ocurrencia de los k-vecinos (SN_5), porciento de las malas 5-ocurrencias (BN_5), el grado del punto *hub* más grande ($max\ N_5$), desequilibrio relativo de la distribución de las etiquetas ($RImb$) y el tamaño de la clase mayoritaria ($p(cM)$).

Tabla 1. Conjuntos de datos del mundo real utilizados en la experimentación

Conjuntos de Datos	e	d	c	S_{N5}	BN_5	max N_5	RImb	p(c_M)
ecoli	336	7	8	0.15	20.7%	13	0.41	42.6%
glass	214	9	6	0.26	25.0%	13	0.34	35.5%
iris	150	4	3	0.32	5.5%	13	0	33.3%
mfeat-factors	2010	216	10	0.83	7.8%	25	0	10%
mfeat-fourrier	2000	76	10	0.93	19.6%	27	0	10%
ovarian	2534	72	2	0.50	15.3%	16	0.28	64%
segment	2310	19	7	0.33	5.3%	15	0	14.3%
sonar	208	60	2	1.28	21.2%	22	0.07	53.4%
vehicle	846	18	4	0.64	35.9%	14	0.02	25.8%

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

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 134-148

En la Tabla 2 se muestra la exactitud promedio de la clasificación obtenida con los métodos: kNN, *hubness-weighted kNN* (hw-kNN), *hubness-based fuzzy nearest neighbor* (h-FNN), *naive hubness-Bayesian k-NN* (NHBNN), *hubness information k-NN* (HIKNN), construcción de prototipos NPBASIR-CLASS y selección de prototipos NPBASIR SEL-CLASS. Todos los experimentos fueron desarrollados para k=5.

La Tabla 3 muestra el ranking obtenido por la prueba de Friedman. En este caso el p-value asociado a dicha prueba es de 7.95, el cual no es lo suficientemente bajo como para rechazar la hipótesis de equivalencia, por lo que podemos concluir que no existen diferencias significativas entre los algoritmos comparados.

Tabla 2. Exactitud promedio de la Clasificación obtenida con los métodos del estado del arte

Conjuntos de Datos	kNN	hw-kNN	h-FNN	NHBNN	HIKNN	NPBASIR-CLASS	NPBASIR SEL-CLASS
diabetes	67.8	75.6	75.4	73.9	75.8	75.1	73.2
ecoli	82.7	86.9	87.6	86.5	87.0	83.3	84.2
glass	61.5	65.8	67.2	59.1	67.9	61.2	69.1
iris	95.3	95.8	95.3	95.6	95.4	96.7	96.7
mfeat-factors	94.7	96.1	95.9	95.7	96.2	92.2	94.7
mfeat-fourier	77.1	81.3	82.0	82.1	82.1	73.7	80.4
ovarian	91.4	92.5	93.2	93.5	93.8	82.6	81.8
segment	87.6	88.2	88.8	87.8	91.2	89.3	85.8
sonar	82.7	83.4	82.0	81.1	85.3	74.9	84.1
vehicle	62.5	65.9	64.9	63.7	67.2	64.7	70.7

Tabla 3. Resultados de la prueba estadística de Friedman para la exactitud de la clasificación general

Algoritmos	Ranking
kNN	5.9
hw-kNN	3.2
h-FNN	3.45
NHBNN	4.45
HIKNN	1.75
NPBASIR-CLASS	5.15
NPBASIR SEL-CLASS	4.1

La prueba de Holm (Tabla 4), respecto a la exactitud general de la clasificación haciendo un todo contra todos rechaza todas las hipótesis con valor $p \leq 0.002632$: solo en el caso HIKNN vs. NPBASIR-CLASS, HIKNN es ligeramente superior a NPBASIR-CLASS. Para el resto de las combinaciones la hipótesis nula no se rechaza, esto es equivalente a decir que no hay diferencias significativas entre sus comportamientos y por lo tanto se puede concluir que son igual de eficaces. Para la Tabla 4 solo se muestran las filas donde aparece uno de los métodos basados en prototipos por ser los de interés para este estudio.

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 134-148

Tabla 4. Prueba de Holm para α=0.05 para la exactitud de la clasificación general, haciendo un todos contra todos

i	Algoritmos	$z = (R_0 - R_i)/SE$	p	Holm	Hipótesis
20	HIKNN vs. NPBASIR-CLASS	3.519334	0.000433	0.0025	Rechaza
16	HIKNN vs. NPBASIR SEL-CLASS	2.432481	0.014996	0.003125	No Rechaza
15	hw-kNN vs. NPBASIR-CLASS	2.018442	0.043545	0.003333	No Rechaza
14	kNN vs. NPBASIR SEL-CLASS	1.863177	0.062437	0.003571	No Rechaza
13	h-FNN vs. NPBASIR-CLASS	1.759667	0.078464	0.003846	No Rechaza
8	NPBASIR-CLASS vs. NPBASIR SEL-CLASS	1.086853	0.277102	0.00625	No Rechaza
6	hw-kNN vs. NPBASIR SEL-CLASS	0.931589	0.351549	0.008333	No Rechaza
5	kNN vs. NPBASIR-CLASS	0.776324	0.437558	0.01	No Rechaza
4	NHBNN vs. NPBASIR-CLASS	0.724569	0.468717	0.0125	No Rechaza
3	h-FNN vs. NPBASIR SEL-CLASS	0.672814	0.501066	0.016667	No Rechaza
2	NHBNN vs. NPBASIR SEL-CLASS	0.362284	0.71714	0.025	No Rechaza

Este trabajo es el primer intento de relacionar el fenómeno de *hubness*, un aspecto del curso de la dimensionalidad, con los métodos de aprendizaje automático basados en prototipos. Estos métodos permiten obtener un conjunto de entrenamiento representativo con menor tamaño comparado con el original y con similar o incluso mayor exactitud en la clasificación de los nuevos datos recibidos.

El estudio experimental realizado demuestra que los métodos de generación y selección de prototipos estudiados ofrecen resultados comparables con los obtenidos en la literatura con los métodos basados en el enfoque kNN, los cuales son *hubness*-consientes y están diseñados específicamente para lidiar con este problema.

Teniendo en cuenta los resultados alentadores de este estudio y las bondades de los métodos basados en prototipos es posible asegurar que la utilización de los mismos permitirá mejorar el desempeño de los sistemas que manejen datos altas dimensiones y bajo la asunción de *hubness*.

Conclusiones

El fenómeno de *hubness* en los datos es un desafío importante del curso de la dimensionalidad pues se conoce por ser perjudicial para el aprendizaje automatizado y para las tareas de minería de datos. Los resultados de este trabajo sugieren que los acercamientos probados exhiben niveles prometedores de robustez y tolerancia ante el problema tratado. Los métodos estudiados de selección y construcción de prototipos **NPBASIR SEL-CLASS** y **NPBASIR-CLASS** ofrecen resultados comparables con los obtenidos con los métodos HIKNN, h-FNN y hw-kNN y NHBNN, los cuales son *hubness*-consientes y están diseñados específicamente para lidiar con este problema.

El diseño de estos métodos debe extenderse para reducir el impacto negativo de los malos *hubs* en los datos. Una manera de hacer esto sería emplear los métodos basados en prototipos combinándolos con los acercamientos *hubness*-conscientes existentes. En trabajos futuros se intentará probar que, si se modifican los métodos de construcción y selección de prototipos basados en relaciones de similitud para el caso de la presencia de "*hubness*", identificando los "malos *hubs*" y no teniéndolos en cuenta a la hora de seleccionar y construir los prototipos, se obtendrán resultados que superen a los encontrados en la bibliografía hasta el momento. Además, se pretende extender el enfoque anterior para dar tratamiento también a los "anti-*hubness*".

Referencias

Alcalá, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. (2010). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing, 17*(255-287), 11.

Bello-García, M., García-Lorenzo, M. M., & Bello, R. (2012). A method for building prototypes in the nearest prototype approach based on similarity relations for problems of function approximation *Advances in Artificial Intelligence* (pp. 39-50): Springer.

Bezdek, J. C., & Kuncheva, L. I. (2001). Nearest prototype classifier designs: An experimental study. *International Journal of Intelligent Systems, 16*(12), 1445-1473.

Bolei, Z., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2015). Object detectors emerge in Deep Scene CNNs.

Choi, W., Pantofaru, C., & Savarese, S. (2013). A general framework for tracking multiple people from a moving camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35*(7), 1577-1591.

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

Duda, R., Hart, P., & Stork, D. (2001). Pattern classification. *International Journal of Computational Intelligence and Applications, 1*, 335-339.

Fernández Hernández, Yumilka, B., Bello, R., Filiberto, Y., Frías, M., Coello Blanco, 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 *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)* (pp. 359-370): Springer.

Filiberto, Y., Caballero, Y., Larrua, R., & Bello, R. (2010). *A method to build similarity relations into extended rough set theory.* Paper presented at the Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on.

Frias, M., Filiberto, Y., Fernández, Y., Caballero, Y., & Bello, R. (2015). *Prototypes selection based on similarity relations for classification problems* Paper presented at the Engineering Applications - International Congress on Engineering (WEA), Bogota

Gao, L., Song, J., Liu, X., Shao, J., Liu, J., & Shao, J. (2015). Learning in high-dimensional multimedia data: the state of the art. *Multimedia Systems*, 1-11.

Gao, L., Song, J., Nie, F., Yan, Y., Sebe, N., & Tao Shen, H. (2015). *Optimal Graph Learning with Partial Tags and Multiple Features for Image and Video Annotation.* Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Garcia, S., Derrac, J., Cano, J. R., & Herrera, F. (2012). Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34*(3), 417-435.

García, S., Fernández, A., Luengo, J., & Herrera, F. (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*(10), 2044-2064.

García, S., & Herrera, F. (2009). Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evolutionary computation, 17*(3), 275-306.

Jin, X.-B., Liu, C.-L., & Hou, X. (2010). Regularized margin-based conditional log-likelihood loss for prototype learning. *Pattern Recognition, 43*(7), 2428-2438.

Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. *Systems, Man and Cybernetics, IEEE Transactions on*(4), 580-585.

Khosla, A., An, B., Lim, J. J., & Torralba, A. (2014). *Looking beyond the visible scene.* Paper presented at the Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on.

Low, T., Borgelt, C., Stober, S., & Nürnberger, A. (2013). The Hubness Phenomenon: Fact or Artifact? *Towards Advanced Data Analysis by Combining Soft Computing and Statistics* (pp. 267-278): Springer.

Nguyen, A., Yosinski, J., & Clune, J. (2015). *Deep neural networks are easily fooled: High confidence predictions for unrecognizable images.* Paper presented at the Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on.

Radovanović, M., Nanopoulos, A., & Ivanović, M. (2009). *Nearest neighbors in high-dimensional data: The emergence and influence of hubs.* Paper presented at the Proceedings of the 26th Annual International Conference on Machine Learning.

Tomašev, N., & Mladenić, D. (2012). Nearest neighbor voting in high dimensional data: Learning from past occurrences. *Computer Science and Information Systems, 9*, 691-712.

Tomašev, N., & Mladenić, D. (2014). Hubness-aware shared neighbor distances for high-dimensional k-nearest neighbor classification. *Knowledge and information systems, 39*(1), 89-122.

Tomasev, N., Radovanović, M., Mladenić, D., & Ivanović, M. (2011). *A probabilistic approach to nearest-neighbor classification: naive hubness bayesian kNN.* Paper presented at the Proceedings of the 20th ACM international conference on Information and knowledge management.

Tomašev, N., Radovanović, M., Mladenić, D., & Ivanović, M. (2014). Hubness-based fuzzy measures for high-dimensional k-nearest neighbor classification. *International Journal of Machine Learning and Cybernetics, 5*(3), 445-458.

Triguero, I., Derrac, J., Garcia, S., & Herrera, F. (2012). A taxonomy and experimental study on prototype generation for nearest neighbor classification. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 42*(1), 86-100.

Weinberger, K. Q., & Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research, 10*, 207-244.

Wiskott, L. (2013). How to solve classification and regression problems on high-dimensional data with a supervised extension of slow feature analysis. *The Journal of Machine Learning Research, 14*(1), 3683-3719.

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann.

Yao, B., Khosla, A., & Fei-Fei, L. (2011). Classifying actions and measuring action similarity by modeling the mutual context of objects and human poses. *a) A, 1*(D2), D3.

Yao, Y. (2000). *Granular computing: basic issues and possible solutions.* Paper presented at the Proceedings of the 5th joint conference on information sciences.

Zhou, X., Chen, L., Zhang, Y., Cao, L., Huang, G., & Wang, C. (2015). *Online video recommendation in sharing community.* Paper presented at the Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data.