

OBTENCIÓN DE UN NÚMERO DIFUSO UTILIZABLE EN APLICACIONES A PARTIR DE DATOS ORDENADOS

<p>Juan Moreno-García E.U.I. Técnica Industrial D. Tecnologías y Sist. de Información Universidad de Castilla-La Mancha Toledo Juan.Moreno@uclm.es</p>	<p>Luis Jiménez, Luis Rodríguez Escuela Superior de Informática Dep. Tecn. y Sist. de Información Universidad de Castilla-La Mancha Ciudad Real {Luis.Jimenez,Luis.Rodriguez}@uclm.es</p>	<p>Fuensanta Andrés E.U.I. Técnica Industrial Dep. de Matemáticas Universidad de Castilla-La Mancha Toledo Fuensanta.Andres@uclm.es</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------

Resumen

Los datos imprecisos son tratados mediante números difusos en aplicaciones reales. El problema principal radica en la obtención de la función de pertenencia del número difuso a partir de un conjunto de datos. En este trabajo se trata el problema de la recapitulación de datos como número difuso, más concretamente, se presenta un método para la obtención de un número difuso trapezoidal a partir de un conjunto de datos ordenado. El método tiene un orden de complejidad $O(n^2)$.

Palabras Clave: números difusos, conjuntos de datos ordenados.

1. INTRODUCCIÓN

Una técnica para el aprendizaje de clasificadores es la estimación de la función $class(x)$, donde un dato $x \in X_1, X_2, \dots, X_n$ son los dominios de los n atributos de los datos, y $class(x) \in C = \{c_1, c_2, \dots, c_m\}$ es la clase asignada al elemento x siendo c_i cualquiera de las posibles clases. El uso de la distribución de frecuencias de los valores de los atributos de los datos de ejemplo es utilizado como aproximación a la función de clasificación $class(x)$. En este tipo de problemas es común representar una determinada clase a través de la frecuencia de aparición de los valores de determinados atributos en los ejemplos. Se establece de esta forma una correspondencia entre la distribución de frecuencias de los valores de dichos atributos con la asignación a una clase determinada. Considerando que los dominios sobre los que están definidos los datos son ordenados, una aproximación para la representación y manejo de estas distribuciones es el uso de conjuntos difusos, y más concretamente de los números difusos. La definición de la función de pertenencia de estos números

difusos mediante funciones trapezoidales facilitan su uso en aplicaciones.

Nuestro trabajo se centra en la obtención de la función de pertenencia del número difuso a partir de una distribución de frecuencias que determina la clase. Para simplificar la manipulación de dichos números generaremos funciones de pertenencia trapezoidales o triangulares. Por tanto se presenta un método para obtener la función de pertenencia del número difuso triangular o trapezoidal que representa al conjunto de datos ordenado.

Existen algunos trabajos previos, Grzegorzewski y Mrówka [3, 4] tratan el problema de obtener una aproximación trapezoidal a los números difusos. Para ello formulan una lista de requerimientos deseables para los operadores de aproximación, obteniendo un operador de aproximación satisfactorio. Delgado [1, 2] obtiene parámetros que capturan la información contenida en un número difuso discreto para simplificar su representación y tratamiento. Con éstos obtienen su representación canónica y posteriormente tratan con ellos en problemas de toma de decisión. Voxman [5] amplía los trabajos anteriores para números difusos no discretos.

En este trabajo se presenta un método para la obtención de un número difuso a partir de un conjunto de datos definido en un dominio ordenado. Está basado en el cálculo de las tres rectas que forman el trapecoide. Las rectas del lado izquierdo y derecho se calculan mediante rectas de regresión. Con la recta central, que representa los valores con pertenencia 1, se busca la mejor colocación de la misma en el eje Y. Las pruebas realizadas demuestran que el método obtiene buenos resultados y es computable.

En la Sección 2 se describe el método. Se muestra un ejemplo de su funcionamiento (Sección 3). En la Sección 4 se realizan las pruebas del método, y el trabajo finaliza exponiendo las conclusiones.

2. MÉTODO PROPUESTO

El método calcula el número difuso trapezoidal (NDT) que representa la función de distribución asignada a una determinada clase de los datos de un dominio ordenado. El caso del número difuso triangular se verá como un caso particular dentro del trapezoidal, el que sólo tiene un valor con pertenencia 1.

Algoritmo 1 Proceso de obtención de NDT

1. Suavizado de valores de Y
{Obtener lista *ordenada* de valores y otros índices}
 2. $ordenada \leftarrow \text{ordenar}(Y)$
 3. $min \leftarrow \text{ordenado}[0]$
 4. $max \leftarrow \text{ordenado}[\text{ordenado}]$
{*corte* toma la posición del máximo valor en Y }
 5. $corte \leftarrow |\text{ordenado}|$
{Cálculo de los valores de los extremos del trapezoide a y b }
 6. $a \leftarrow$ Obtenido mediante Ecuación 1
 7. $d \leftarrow$ Obtenido mediante Ecuación 2
 - while** $ordenada[corte] > \text{media}(y_a, y_d)$ **do**
{Calcular posiciones derecha e izquierda de corte}
 8. $b \leftarrow$ Obtenido mediante Ecuación 3
 9. $c \leftarrow$ Obtenido mediante Ecuación 4
{Calcular rectas izquierda, centro y derecha del trapezoide}
 - {Recta de regresión para el intervalo $[a, b]$ }
 10. Calcular Recta Izquierda $r_{a,b} \leftarrow a_{a,b} + (b_{a,b} * x)$
{Recta de regresión para el intervalo $[c, d]$ }
 11. Calcular Recta Derecha $r_{c,d} \leftarrow a_{c,d} + (b_{c,d} * x)$
{Calcular Recta centro para intervalo $[b, c]$ }
 12. Calcular Recta Centro $r_{b,c}$
{Cálculo Bondad Trapecio Obtenido}
 13. $bd_{a,b} \leftarrow$ Bondad($r_{a,b}$) (Ecuación 5)
 14. $bd_{b,c} \leftarrow$ Bondad($r_{b,c}$) Ecuación 5)
 15. $bd_{c,d} \leftarrow$ Bondad($r_{c,d}$) Ecuación 5)
 16. Cálculo bondad del trapezoide (Ecuación 6)
{Seleccionar trapecio si obtenido es mejor}
 - if** $bd_{trap} < bd_{act}$ **then**
17. $bd_{act} \leftarrow bd_{trap}$
 18. Actualizar valores trapezoide con a, b, c, d .
 - end if**
{bajar línea de corte en *ordenada*}
 19. $corte \leftarrow corte - 1$
 20. Recalcular a y d para que $y_a = 0$ e $y_d = 0$
 - end while**
-

Sea una variable estadística unidimensional definida sobre el dominio X con una distribución de frecuencia $f(X) = Y$. Se representará de la siguiente forma:

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

La salida es el número difuso trapezoidal (NDT) que representa a la aproximación de la distribución

de frecuencias dada. El número difuso se representará con cuatro componentes, $NDT = \{a, b, c, d\}$ donde a, b, c, d son los cuatro valores del trapezoide.

La idea del método consiste en un proceso iterativo que va descendiendo y cortando el eje Y desde los valores más altos a los más bajos. Se calculan las rectas izquierda (desde a hasta b) y derecha (desde c hasta d) del trapezoide mediante las rectas de regresión de Y sobre X , y la recta paralela al eje X del trapezoide (pertenencias a 1) como la recta que une ambas (desde b hasta c). Después se calcula la bondad del trapezoide obtenido, y si es mejor que el obtenido hasta ese instante se selecciona. El proceso termina cuando el valor de corte en el eje Y llega a la media de los valores desde y_a a y_d .

El Algoritmo 1 muestra en detalle el proceso seguido para obtener el NDT. En determinados problemas, la distribución de frecuencias obtenida presenta saltos abruptos en sus valores que provocan grandes picos en su gráfica, por ejemplo ver Entrada R de la Figura 1. Para eliminar dichos picos se propone una fase inicial de suavizado (Sentencia 1). Ésta consiste en que cada valor y_i es sustituido por la media de los valores comprendidos entre y_{i-t} e y_{i+t} .

La Sentencia 2 crea una lista ordenada (llamada *ordenada*) a partir de los valores de Y . Esta lista es utilizada para calcular la ubicación de la recta con pertenencias a 1 en el eje Y en los trapezoides candidatos. En las Sentencias de la 3 a la 5 se obtienen las variables min , max y $corte$: min y max son los valores mínimos y máximos del soporte Y respectivamente, y $corte$ es la variable que almacena la posición donde se corta el eje Y para colocar la recta con las pertenencias a 1. Los cortes se van realizando por los valores de Y de mayor a menor, es decir, los valores de *ordenada* del último al primero (Sentencia 5).

En la fase experimental se han detectado problemas con los valores de y cercanos a 0 para determinar el soporte $[a, d]$ del NDT buscado. El trapezoide cubre un soporte muy amplio y el intervalo $[a, d]$ engloba un número de datos superior al estrictamente necesario (Figura 4). Al calcular las rectas de regresión se cubren las zonas con valores bajos provocando rectas que no se ajustan correctamente a las zonas con valores más altos y más fiables. Por ello se considera que un valor pertenece al soporte si supera un porcentaje del valor máximo de la distribución de frecuencias. En este trabajo le llamaremos umbral de soporte y lo representaremos como U_α , siendo α el porcentaje de corte. En las pruebas realizadas el α se ha fijado al 10%.

A continuación se obtienen los valores que definen el valor mínimo y máximo del soporte del número difuso (a y d respectivamente). La posición de a , que se ob-

tiene mediante la sentencia 6 (Ecuación 1), se asigna al menor x_i desde 0 a x_n que verifica que $y_i > U_\alpha$.

$$\operatorname{argmin}_{x_i \in [0, x_n]} y_i > U_\alpha \quad (1)$$

La sentencia 7 fija la posición de d (Ecuación 2) al mayor x_i desde 0 a x_n que verifica que $y_i > U_\alpha$.

$$\operatorname{argmax}_{x_i \in [0, x_n]} y_i > U_\alpha \quad (2)$$

El bucle *while* calcula las tres rectas que forman el trapezoide, y posteriormente se obtiene la bondad del trapezoide obtenido. Primeramente se obtiene el valor de b y c del trapezoide (Sentencias 8 y 9 del bucle). b se asigna al menor x_i desde a a d que verifica que $y_i \geq \text{ordenada}[\text{corte}]$ (Ecuación 3).

$$\operatorname{argmin}_{x_i \in [a, d]} y_i \geq \text{ordenada}[\text{corte}] \quad (3)$$

De forma análoga, c toma el valor del mayor x_i que verifica que $y_i \geq \text{ordenada}[\text{corte}]$ (Ecuación 4).

$$\operatorname{argmax}_{x_i \in [a, d]} y_i \geq \text{ordenada}[\text{corte}] \quad (4)$$

Una vez obtenidos los puntos de trapezoide (a, b, c y d) se pueden obtener las rectas que forman el mismo. Para obtener las rectas $r_{a,b}$ y $r_{c,d}$ se hace uso de la regresión lineal. La Sentencia 10 calcula la recta de regresión de Y sobre el eje X comprendida en el intervalo $[a, b]$ ($r_{a,b}$). La Sentencia 11 calcula la recta de regresión de Y sobre el eje X en el intervalo $[c, d]$ ($r_{c,d}$). La Sentencia 12 calcula la recta central ($r_{b,c}$), la comprendida en el intervalo $[b, c]$. Es uno de los pasos críticos del método propuesto debido a que el valor y_b obtenido por $r_{a,b}$ no tiene porque ser cercano al valor y_c obtenido por $r_{c,d}$, provocando que haya escalones en el trapezoide (Figura 2). Para solucionar esto se debe recalculer el valor de b o de c mediante el Algoritmo 2 (Sección 2.1). Una vez obtenidos los valores definitivos de b y c , la recta paralela al eje Y comprendida entre b y c toma el valor de la Ecuación $r_{b,c} = a_{b,c}$.

A continuación se calcula la bondad del trapezoide obtenido, para ello se calcula la bondad de las rectas $r_{a,b}$, $r_{b,c}$ y $r_{c,d}$ (Sentencias 13, 14 y 15 respectivamente) mediante la Ecuación 5 obteniendo las bondades $bd_{a,b}$, $bd_{b,c}$ y $bd_{c,d}$ respectivamente.

$$bd_{j,k} = \frac{\sum_{i=j}^k (y_i - (a_{j,k} + (b_{j,k} * x_i)))^2}{|[j, k]|} \quad (5)$$

donde $i \in [j, k]$, $|[j, k]|$ es el número de elementos del intervalo $[j, k]$, $y_i \in Y$, y $a_{j,k}$ y $b_{j,k}$ definen la recta a calcular la bondad ($r_{j,k}$).

Como puede verse, la ecuación calcula el error cuadrático medio entre la recta obtenida y la distribución de frecuencias de entrada del eje Y del intervalo que cubre la recta.

La Sentencia 16 calcula la bondad del trapezoide (bd_{trap}) mediante la Ecuación 6 que realiza la media ponderada de las bondades de las tres rectas.

$$\frac{(bd_{a,b} * |[a, b]|) + (bd_{b,c} * |[b, c]|) + (bd_{c,d} * |[c, d]|)}{|[a, b]| + |[b, c]| + |[c, d]|} \quad (6)$$

Finalmente, si la bondad obtenida bd_{trap} es menor que la mejor hasta el momento (bd_{act}), ésta toma el valor de bd_{trap} (Sentencia 17), y se actualizan los valores que definen el trapezoide (Sentencia 18). La sentencia 19 decreuenta *corte* para calcular el trapezoide del siguiente punto de *ordenada*.

De esta forma el *NDT* queda determinado por los puntos ($a, a_{a,b} + (b_{a,b} * a)$), ($b, a_{b,c}$), ($c, a_{b,c}$), ($d, a_{c,d} + (b_{c,d} * d)$). El soporte de Y para estos puntos está comprendido entre 0 y $a_{b,c}$, por lo que se debe obtener su equivalente para valores de salida en $y_i \in [0, 1]$.

b y c mantienen su valor y representarán el comienzo y fin del intervalo de x que ofrece como salida $y = 1$, es decir, $r_{c,d} = 1$. Sin embargo, si el *NDT* está en los límites del dominio de X , los valores de a y d puede que no verifiquen que $y_a = 0$ e $y_d = 0$ (Figura 3). Por ello se debe recalculer el valor de a y d para que $y_a = 0$ e $y_d = 0$ (Sentencia 20). Para calcular el punto de corte de a con 0 se debe verificar la ecuación $r_{a,b} = a_{a,b} + (b_{a,b} * a) = 0$, y despejando se obtiene que $a \leftarrow \frac{-a_{a,b}}{b_{a,b}}$. El valor de d se obtiene de igual forma que el caso anterior ($d \leftarrow \frac{-a_{c,d}}{b_{c,d}}$).

Una vez obtenido el *NDT* su función de pertenencia queda determinada por la Ecuación 7. El método propuesto tiene una complejidad de orden $O(n^2)$. Ésta viene determinada por la complejidad del bucle, que tiene sentencias de orden $O(n)$, y la propia iteración del bucle ($O(n)$), lo que provoca que la complejidad total del método sea $O(n^2)$.

$$(x \leq a) \rightarrow 0$$

$$(x > a) \text{ and } (x < b) \rightarrow \frac{(x-a)}{b-a}$$

$$(x \geq b) \text{ and } (x \leq c) \rightarrow 1 \quad (7)$$

$$(x > c) \text{ and } (x < d) \rightarrow \frac{(d-x)}{d-c}$$

$$(x \geq d), \rightarrow 0$$

2.1. SELECCIÓN DE LOS VALORES DEFINITIVOS DE b y c

El Algoritmo 2 recalcula el valor de b o de c para que el máximo valor de $r_{a,b}$ (y_b) sea lo más parecido posi-

ble al máximo valor de $r_{c,d}(y_c)$, evitando así que haya escalones en la gráfica. La Figura 2 muestra un ejemplo, como puede verse el máximo valor de la recta $r_{a,b}$ es mucho mayor que el máximo valor de la recta $r_{c,d}$, lo que provoca que aparezca un escalón al colocar la recta $r_{b,c}$ paralela al eje X .

Algoritmo 2 Cálculo de b y c definitivos

```

{Cálculo valores en posiciones  $b$  y  $c$  para el eje  $Y$ }
1.  $y_b \leftarrow a_{a,b} + (b_{a,b} * x_b)$ 
2.  $y_c \leftarrow a_{c,d} + (b_{c,d} * x_c)$ 
if  $a = b$  or  $c = d$  then
  { $b$  y  $c$  mantienen su valor en ambos casos}
  if  $a = b$  then
    {no existe  $r_{a,b}$ }
    {Pertenencias a 1 en el máximo valor de  $r_{c,d}$ }
    3.  $a_{b,c} \leftarrow y_c$ 
  else
    {no existe  $r_{c,d}$ }
    {Pertenencias a 1 en el máximo valor de  $r_{a,b}$ }
    4.  $a_{b,c} \leftarrow y_b$ 
  end if
else
  {Existen  $r_{a,b}$  y  $r_{c,d}$ }
  if  $y_c \leq y_b$  then
    {Se decreta  $b$  hasta que  $y_c \leq y_b$ }
    while  $y_c < y_b$  do
      5.  $b \leftarrow b - 1$ 
      6.  $y_b \leftarrow a_{a,b} + (b_{a,b} * x_b)$ 
    end while
    {corte a  $y_c$  ya que  $y_c \geq y_b$ }
    7.  $a_{b,c} \leftarrow y_c$ 
    { $b$  toma valor que cumple  $r_{a,b} = a_{b,c}$ }
    8.  $b \leftarrow \frac{a_{b,c} - a_{a,b}}{b_{a,b}}$ 
  else
    {Se decreta  $c$  hasta que  $y_b < y_c$ }
    while  $y_b < y_c$  do
      9.  $c \leftarrow c + 1$ 
      10.  $y_c \leftarrow a_{c,d} + (b_{c,d} * x_c)$ 
    end while
    {corte a  $y_b$  ya que  $y_b \geq y_c$ }
    11.  $a_{b,c} \leftarrow y_b$ 
    { $c$  toma valor que cumple  $r_{c,d} = a_{b,c}$ }
    12.  $c \leftarrow \frac{a_{b,c} - a_{c,d}}{b_{c,d}}$ 
  end if
end if

```

Las Sentencias 1 y 2 calculan el valor y_b e y_c que ofrecen las rectas $r_{a,b}$ y $r_{c,d}$ respectivamente. A continuación se comprueba si realmente se trabaja sobre un trapezoide o no, es decir, si el trapezoide está compuesto por tres rectas o no existe la recta izquierda o derecha del mismo ($r_{a,b}$ o $r_{c,d}$ respectivamente). La recta $r_{a,b}$ existe si $a \neq b$ (primer *if* dentro del primer *if*), y análogamente $r_{c,d}$ existe si $c \neq d$ (segundo *if*

dentro del primer *if*). En este caso, la recta de pertenencias a 1 ($r_{b,c}$) se fija en el máximo valor de la recta que existe: $a_{b,c} = y_c$ si no existe $r_{a,b}$ (Sentencia 3), o $a_{b,c} = y_b$ si no existe $r_{c,d}$ (Sentencia 4). El *else* del primer *if* trata el caso del trapezoide completo, es decir, existen $r_{a,b}$ y $r_{c,d}$. La idea consiste en bajar la ubicación de $r_{b,c}$ en el eje Y para que y_b e y_c estén lo más paralelos posible. Por ello, si y_b es mayor que y_c se va decrementando b (Sentencia 5) y calculando el nuevo valor de y_b (Sentencia 6) hasta que $y_c \geq y_b$. La línea de corte se coloca en y_c (Sentencia 7), y la Sentencia 8 asigna el valor de b al valor x que provoca el corte de las rectas $r_{a,b}$ y $r_{b,c}$, es decir, el que cumple que $y_c = r_{a,b}$, luego $b \leftarrow \frac{a_{b,c} - a_{a,b}}{b_{a,b}}$.

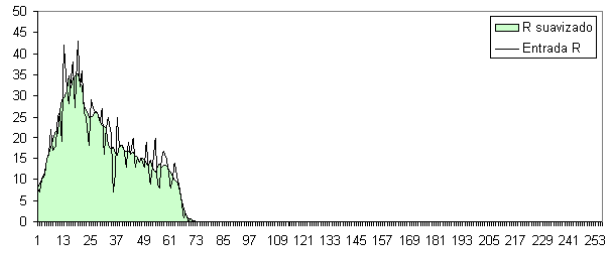


Figura 1: Proceso de suavizado de componente R

Si y_c es mayor que y_b se va incrementando c (Sentencia 9) y calculando el nuevo valor de y_c (Sentencia 10) hasta que $y_b \geq y_c$. La línea de corte se coloca en y_b (Sentencia 11), y análogamente al caso anterior, c toma el valor $b \leftarrow \frac{a_{b,c} - a_{c,d}}{b_{c,d}}$. La Figura 3 muestra cómo el algoritmo soluciona el problema del escalón mostrado en la Figura 2.

3. EJEMPLO DE FUNCIONAMIENTO

Para ver el funcionamiento del método propuesto se verá un ejemplo de su uso en una aplicación real. La aplicación procesa imágenes satélite en color en formato RGB para detectar olivos. Dado que la información de entrada es imprecisa, se ha decidido utilizar números difusos para representar los píxeles de los olivos. Por otro lado, se tiene una imagen satélite RGB de un campo de olivos en la que los píxeles están correctamente etiquetados como olivo, suelo u otro caso. A partir de esta imagen se puede calcular el NDT de cada una de sus componentes utilizando como distribución de frecuencias el histograma de cada una de las componentes RGB de los píxeles que representan a los olivos. La Figura 1, mediante la gráfica etiquetada como *Entrada R*, muestra el histograma de la componente R que será tomado como entrada al método en el ejemplo.

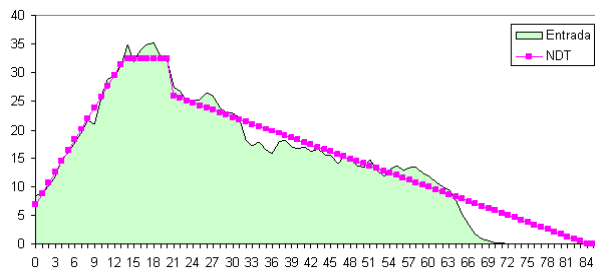


Figura 2: NDT con escalón

El primer paso es el suavizado con $t = 2$ que obtiene el resultado de la Figura 1 mediante la gráfica con la etiqueta *suavizado R*. A continuación se calculan las variables *ordenada*, *min*, *max* y *corte* (Sentencias de la 2 a la 5). *ordenada* es la lista de valores del histograma ordenados, *min* toma el valor 0, *max* es igual a 35,20 y el corte inicial en el eje Y es igual a *max* (*ordenada*[*corte*] = 35,20). Las Sentencias 6 y 7 calculan los valores de *a* y *d* con un $U_\alpha = 10\%$, *a* y *d* toman los valores 0 y 65 respectivamente.

Ya dentro del bucle *while*, las Sentencias 8 y 9 calculan los valores de *b* y *c* para la primera iteración, tomando ambos el valor 18. Con estos valores, el *NDT* inicial propuesto es $\{0, 18, 18, 65\}$, luego existen las rectas $r_{a,b}$ ($0 \neq 18$) y $r_{c,d}$ ($18 \neq 65$). La Sentencia 10 calcula la recta de regresión $r_{a,b}$, quedando definida por los valores $a_{a,b} = 7,9091$ y $b_{a,b} = 1,6705$. La recta $r_{c,d}$ toma el valor $36,3153 + (-0,4464 * x_i)$ (Sentencia 11). La Sentencia 12 calcula la recta central, al no estar correctamente alineadas las dos rectas de regresión (Figura 2) se recalcula el valor de *b* o *c* (Algoritmo 2), manteniendo *c* su valor y tomando *b* el valor 12,1949, el corte definitivo ($a_{b,c}$) pasa a valer 28,2809 y la recta $r_{b,c} = 28,2809$.

A continuación se calculan las bondades de cada una de las rectas mediante la Ecuación 5 obteniendo $bd_{a,b} = 1,3269$, $bd_{b,c} = 23,1222$ y $bd_{c,d} = 5,3426$ (Sentencias 13, 14 y 15). La Sentencia 16 calcula la bondad del trapecio obtenido aplicando la Ecuación 6, obteniendo que $bd_{trap} = 6,4338$. Las Sentencias 17 y 18 actualizan los valores de bd_{act} y del trapecio al ser la primera ejecución y finalmente se baja la línea de corte.

La siguiente ejecución establece el corte inicial en 34,80 y el *NDT* inicial a $\{0, 14, 18, 65\}$. Las rectas de los lados del trapecioide son $r_{a,b} = 6,7972 + (1,8874 * x_i)$ y $r_{c,d} = 36,3153 - (0,4464 * x_i)$. El cálculo de la recta $r_{b,c}$ provoca que *b* cambie de valor y se recoloque $a_{b,c}$, quedando el corte definitivo en 28,2809 y el *NDT* definitivo como $\{0, 11,3828, 18, 65\}$, luego $r_{b,c} = 28,2809$. Las bondades de las rectas son $bd_{a,b} = 1,0970$, $bd_{b,c} = 19,8575$ y $bd_{c,d} = 5,3426$. La bondad

del trapecioide es $bd_{trap} = 6,2856$. Esta bondad mejora la bd_{act} con lo que se actualizan los valores de bd_{act} y del trapecio.

Para esta nueva ejecución del bucle el corte inicial toma el valor 34,00 y el *NDT* inicial es $\{0, 14, 18, 65\}$. Las rectas del trapecioide son $r_{a,b} = 6,7972 + (1,8874 * x_i)$ y $r_{c,d} = 36,3153 - (0,4464 * x_i)$. Se calcula la recta $r_{b,c}$ tomando ésta el valor 28,2809, y siendo el resto de valores los siguientes: $a_{b,c} = 28,2809$, $NDT = \{0, 11,3828, 18, 65\}$. Las bondades son $bd_{a,b} = 1,0970$, $bd_{b,c} = 19,8575$ y $bd_{c,d} = 5,3426$, lo que implica que la bondad del trapecioide es $bd_{trap} = 6,2856$. Al no ser mejor que bd_{act} no se modifica.

Para la siguiente ejecución el corte inicial se establece en 32,80 y el *NDT* inicial es $NDT = \{0, 14, 19, 65\}$. Las rectas del trapecioide son $r_{a,b} = 6,7972 + (1,8874 * x_i)$, $r_{b,c} = 23,2986$ y $r_{c,d} = 35,3616 - (0,4272 * x_i)$, quedando el corte definitivo en 27,2457 y el *NDT* definitivo en $\{0, 10,8343, 19, 65\}$. Las bondades de las rectas son $bd_{a,b} = 1,0566$, $bd_{b,c} = 29,6076$ y $bd_{c,d} = 4,3482$, y la del trapecioide es $bd_{trap} = 7,3157$. No se mejora bd_{act} .

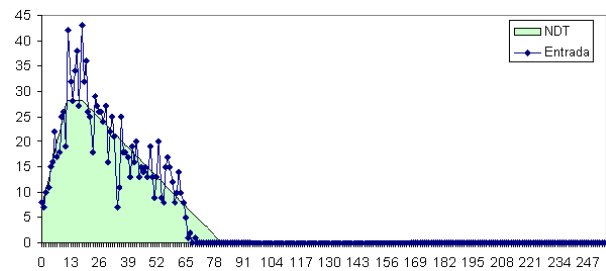


Figura 3: NDT obtenido para el ejemplo

En el resto de iteraciones del bucle la bondad del trapecioide se va incrementando iteración a iteración hasta llegar a un valor máximo de 33,8867.

De todos los trapecioide anteriores se ha seleccionado el de mejor bondad, es decir, el que obtiene la bondad $bd_{trap} = 6,2856$. Como puede verse, hay dos casos consecutivos, el algoritmo selecciona el primero de ellos. Luego el trapecioide final se muestra en la Figura 3. Se concluye que el error es pequeño y se observa un buen ajuste.

4. PRUEBA DEL MÉTODO

Para probar el método propuesto se ha utilizado función continua y otra discreta. Como función continua se ha seleccionado la función gaussiana y como función discreta se ha probado tomando como entrada el histograma de la componente B de los píxeles del olivo de la aplicación de la Sección 3.

Para poder realizar las pruebas de la función continua se ha definido un conjunto de datos con los valores que toma la función con un incremento fijo entre cada valor x_i a muestrear. Para la función gaussiana se ha utilizado la Ecuación $e^{-k*(x_i-m)^2}$ con $k = 3$, $m = 8$ e *incremento* = 0,05. La primera prueba ha consistido en obtener el *NDT* de la función gaussiana con $U_\alpha = 0\%$. La Figura 4 muestra el resultado obtenido. Se obtiene una bondad de 0,0473 y un resultado que no es del todo satisfactorio, ya que intenta cubrir todos los valores del dominio (incluidos los bajos). El *NDT* obtenido es igual a $\{43,00, 160,00, 178,54, 220,00\}$. La siguiente prueba se ha realizado con $U_\alpha = 10\%$, y se obtiene una bondad de 0,0006 aplicando la Ecuación 6, consiguiendo un ajuste perfecto y cubriendo perfectamente el dominio (Figura 5). El *NDT* obtenido es igual a $\{143,00, 158,00, 162,00, 177,00\}$, como puede verse el soporte cubierto por el *NDT* es bastante inferior al caso anterior y se ajusta mejor. Esto muestra que un $U_\alpha \neq 0\%$ es necesario en muchas ocasiones.

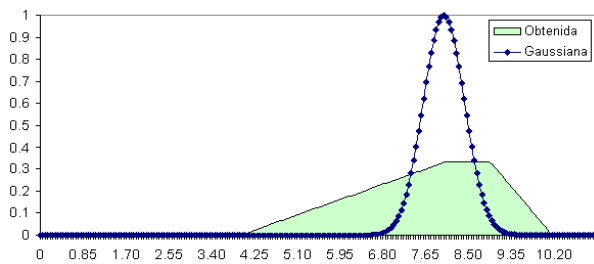


Figura 4: Resultado gaussiana con $U_\alpha = 0\%$.

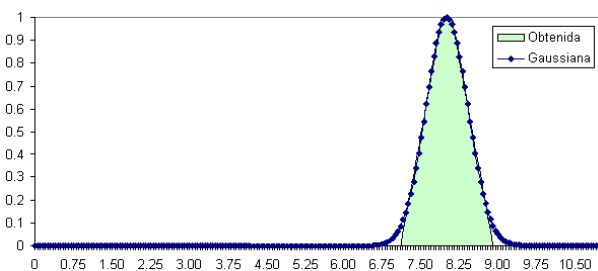


Figura 5: Resultado gaussiana con $U_\alpha = 10\%$.

La última prueba realizada ha tomado como entrada el histograma de la componente B de los píxeles del olivo de la aplicación de la Sección 3. La Figura 6 muestra tanto los datos de entrada como el resultado obtenido. Como puede verse se obtiene un buen ajuste en la recta $r_{c,d}$, y el error obtenido es de 375,3799. El *NDT* obtenido es igual a $\{0,00, 0,00, 1,00, 37,00\}$.

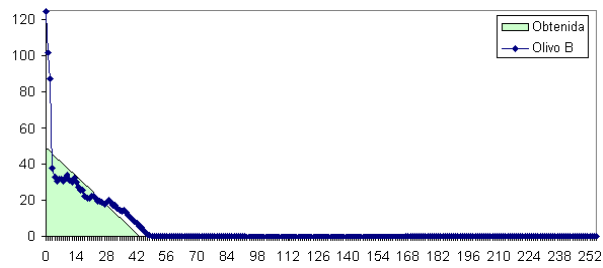


Figura 6: Resultado componente B de olivo.

5. CONCLUSIONES Y TRABAJOS FUTUROS

Se ha presentado un método novedoso para la obtención del número difuso trapezoidal a partir de un conjunto de datos ordenado. El método se basa en el cálculo de las rectas de regresión de los dos lados del trapecio. Es un método computable de orden $O(n^2)$ que puede ser utilizado en aplicaciones reales. Los resultados presentados pueden ser considerados como válidos, al obtener poco error y ajustarse el resultado a las gráficas de los datos de entrada. Basándonos en las pruebas podemos afirmar que los resultados son mejores si se aplica un $U_\alpha \neq 0\%$ a las frecuencias de distribución de entrada para eliminar los valores bajos.

Agradecimientos

Este trabajo ha sido financiado por la Consejería de Educación y Ciencia de Castilla-La Mancha a través de los proyectos PII2I09-0052-3440 y PII1C09-0137-6488, y por el Ministerio de Ciencia y Tecnología a través de los proyectos TIN2009-14538-C02-02.

Referencias

- [1] M. Delgado, M.A. Vila, W. Voxman. On a canonical representation of fuzzy numbers. *Fuzzy Sets and Systems*, Vol. 93, Pág. 125-135, 1998.
- [2] M. Delgado, M.A. Vila, W. Voxman. A fuzziness measure for fuzzy numbers: applications. *Fuzzy Sets and Systems*, Vol. 94, Pág. 205-216, 1998.
- [3] P. Grzegorzewski, E. Mrówka. Trapezoidal approximations of fuzzy numbers. *Fuzzy Sets and Systems*, Vol. 153, Pág. 115-135, 2005.
- [4] P. Grzegorzewski, E. Mrówka. Trapezoidal approximations of fuzzy numbers—revisited. *Fuzzy Sets and Systems*, Vol. 158, Pág. 757-768, 2007.
- [5] W. Voxman. Canonical representations of discrete fuzzy numbers. *Fuzzy Sets and Systems*, Vol. 118, Pág. 457-466, 2001.