INDUCCIÓN DIRECTA DE MODELOS DIFUSOS TEMPORALES

Juan Moreno-García
Universidad de Castilla-La Mancha
Departamento de Informática
Avda Carlos III s/n, Toledo
Juan.Moreno@uclm.es

Carlos González, Ramón Manjavacas
Universidad de Castilla-La Mancha
Departamento de Informática
Paseo de la Universidad nº4, Ciudad Real
{Carlos.Gonzalez,Ramon.Manjavacas}@uclm.es

Resumen

Los modelos difusos temporales (MDTs) se utilizan para representar de una manera lingüística los sistemas dinámicos (SDs). Están compuestos por reglas difusas temporales (RDTs) que están ordenadas. Cada RDT representa una zona temporal. Los MDTs se obtienen a partir de un modelo difuso inducido por otro algoritmo de inducción. Primeramente, se utiliza un algoritmo de inducción para obtener un modelo difuso, y a continuación, se transforma el modelo difuso obtenido en un MDT utilizando un algoritmo de transformación [6]. El objetivo de este trabajo es presentar un algoritmo directo de inducción de MDTs, es decir, un método que no necesita un MDT como entrada.

Palabras Clave: Inducción de modelos, modelado temporal, modelado lingüístico.

1 INTRODUCCIÓN

Un SD puede ser definido como un sistema que cambia en el tiempo. En un SD existen un conjunto de relaciones entre las variables de entrada y salida que representan las modificaciones de las variables de salida cuando las variables de entrada son modificadas. En los SDs, también se necesita conocer la evolución temporal de estas variables porque los valores de las variables en un tiempo dependen de los valores de las variables en los instantes anteriores. Los SDs con magnitudes físicas continuas verifican que la evolución de las variables es continua en el tiempo, esto es, en el instante $t+1$ el valor de la variable v_{t+1} es muy parecido al valor de la variable v_t en el instante t. Esta hipótesis se supone cuando se definen los MDTs.

Los SDs se modelan utilizando diferentes técnicas tradicionales [8, 5]. Se modelan para explicar y predecir su comportamiento, y para analizar sus propiedades dinámicas. Se pueden modelar utilizando de la lógica difusa [7, 1, 2, 3]. Los MDTs se utilizan para representar la evolución del SD. Esta estructura simula el comportamiento de un sistema en función de la entrada. Informalmente, los MDTs son una secuencia ordenada de reglas difusas, donde cada regla representa una zona temporal.

Una diferencia entre nuestro método y otros es que nosotros modelamos los SDs mediante sistemas ordenados de reglas, de esta forma, cada regla representa instantes consecutivos con la misma etiqueta de salida. Utilizando este método es posible detectar que variables cambian de una regla a otra, es decir, de un instante o otro. Otra diferencia es que utilizamos etiquetas lingüísticas [11] a lo largo de todo el proceso de inducción, así, los modelos obtenidos son más descriptivos. La última diferencia es que nuestro método agrupa utilizando la variable de salida tomando los ejemplos consecutivos que tienen la misma salida. En [6] se presenta un algoritmo para transformar un modelo difuso en un MDT. Este método es interesante cuando se dispone de un modelo difuso que es una buena aproximación al sistema que representa. El objetivo de este trabajo es presentar un método de inducción de MDT que no necesite otro modelo difuso como entrada.

En la sección 2 se presentan algunas consideraciones previas. Después, se expone el método de inducción de los MDTs. En la sección 4 se utiliza el algoritmo en un ejemplo, y se finaliza con las conclusiones.

2 CONSIDERACIONES PREVIAS

Antes de estudiar en detalle como se obtiene un MDT, se mostrará la estructura de nuestro método. Se utiliza el algoritmo 2 que tiene dos entradas: un conjunto de ejemplos E y $m+1$ conjuntos ordenados de etiquetas lingüísticas $SA_1 \dots SA_m, SC$.

En las subsecciones 2.1 y 2.2 se ofrece una descripción detallada de las dos entradas del algoritmo. La sección 2.3 define los MDTs.

2.1 CONJUNTO DE EJEMPLOS E

Una serie de tiempo es una secuencia de datos que es ordenada en el tiempo [9]. Un SD se puede representar por medio de una serie temporal. Cada ejemplo de la serie contiene los valores de las variables en un instante. Esta sección muestra formalmente la estructura de la serie de tiempo que se representa mediante un conjunto de ejemplos E.

Cada ejemplo de E tiene un conjunto de m variables reales de entrada definidas en el dominio $X = X_1 \times \dots \times X_m \subset R^m$, y una variable real de salida $S \subset R$. $E = \{e_1 \dots e_n\}$ siendo $e_i = (x_1^i \dots x_m^i, s^i, t_i)$, donde $x_j^i \in X_j$, $s^i \in S$ y t_i es el tiempo en que ocurre e_i.

2.2 CTOS ORDENADOS DE ETIQUETAS

Los MDTs trabajan con variables lingüísticas que pueden tomar valores de un conjunto ordenado de etiquetas lingüísticas con algunas restricciones sobre su dominio. Estas variables se denominan *variables lingüísticas continuas* (desde ahora variables). Las etiquetas lingüísticas (desde ahora etiquetas) de las variables se definen antes de obtener el MDT.

Se define un conjunto ordenado de etiquetas para cada variable de entrada X_j. Su estructura es $SA_j = \{SA_j^1 \dots SA_j^{i_j}\}$. Para la variable de salida S se define un conjunto ordenado de etiquetas $SC = \{SC^1 \dots SC^{i_y}\}$. En estos conjuntos el superíndice i es la posición de la etiqueta, $i_j = |SA_j|$ e $i_y = |SC|$.

El orden de las etiquetas en los conjuntos de etiquetas se basa en el método de defuzzification llamado *media de máximos* (MOM) [4]. El MOM de una etiqueta SA_j^a lo representamos como $MOM(SA_j^a)$. Una etiqueta SA_j^a es menor que SA_j^b si ambas están definidas para X_j y el MOM de SA_j^a es menor que el MOM de SA_j^b. De igual forma se puede establecer cuando una etiqueta es menor o igual que otra.

X_j es una variable lingüística continua si tiene asociada una semántica a cada SA_j^i definida sobre ella que verifica: (1) Cada etiqueta SA_j^i se define para un dominio ordenado; (2) Las etiquetas definidas sobre X_j están ordenadas mediante el MOM; (3) La suma del grado de pertenencia de un valor x a todas las etiquetas definidas en SA_j debe ser 1 ($\sum_{SA_j^i \in SA_j} \mu_{SA_j^i}(x) = 1$); (4) Cada x_j en el dominio de X_j tiene 1 o 2 etiquetas con un grado de pertenencia mayor que 0. Así, una de estas expresiones se verifica: ($\mu_{SA_j^i}(x_j) = 1$) ó ($\mu_{SA_j^i}(x_j) = a$ y $\mu_{SA_j^{i+1}}(x_j) = 1 - a$). Si el grado de pertenencia de x_j a SA_j^i se incrementa o decrementa entonces el grado de pertenencia de x_j a SA_j^{i+1} se decrementará o incrementará en el mismo grado.

2.3 MODELOS DIFUSOS TEMPORALES

Los SDs estudiados tienen una variable de salida. Nuestro método de inducción agrupa utilizando la variable de salida, así, todos los ejemplos consecutivos con la misma etiqueta de salida (la etiqueta con más grado de pertenencia al valor de salida s^i del ejemplos e_i) se representan con la misma regla.

Un MDT está compuesto por **Reglas Difusas Temporales** ($RDTs$) que tienen dos componentes: un antecedente y una etiqueta de salida. El antecedente es una conjunción de disyunciones de expresiones X_j *is* SA_j^i. El antecedente tiene asociado un conjunto de ejemplos consecutivos $E_{p,u}$. Un **conjunto de ejemplos consecutivos** $E_{p,u}$ es un conjunto de ejemplos que verifica: (1) $p \leq u$ y $E_{p,u} \subset E$; (2) $\forall k \in [p,u], \exists e_k \in E_{p,u}$; (3) $\forall e_a, e_b \in E_{p,u}, a \neq b$, donde E es el conjunto de ejemplos; p, u, k, a y b son los índices a los ejemplos de E.

Se utilizan los intervalos lingüísticos para representar las disyunciones, y el conjunto de m intervalos lingüísticos para representar las conjunciones de disyunciones de una RDT. **Un intervalo lingüístico de longitud** c es una secuencia de c etiquetas consecutivas definidas en SA_j que comienzan en la etiqueta p. Se representa como $LI_{j,p}^c = \{SA_j^p \dots SA_j^{p+(c-1)}\}$, donde $SA_j = \{SA_j^1 \dots SA_j^{i_j}\}$ es un conjunto ordenado de i_j etiquetas en X_j, p es la posición en SA_j de la primera etiqueta del intervalo lingüístico y c es el número de etiquetas del intervalo. Por ejemplo, el intervalo $LI_{j,3}^4$ representa la disyunción ((x_j **is** FN) **OR** (x_j **is** NR) **OR** (x_j **is** FP) **OR** (x_j **is** P)) si se utiliza el conjunto de la figura 3.

Un **conjunto ordenado de** m **intervalos** SLI_m es una secuencia de m intervalos definida sobre m variables. Se representa como $SLI_m = \{LI_{1,p_1}^{c_1} \dots LI_{m,p_m}^{c_m}\}$ donde p_j es la posición en SA_j de la primera etiqueta del intervalo j y c_j es el número de etiquetas del intervalo j. Los SLI_m se utilizan para representar lingüísticamente el rango de valores de m variables de entrada. Por ejemplo, si para cada variable de entrada se utiliza un conjunto ordenado de 7 etiquetas representadas en la figura 3, entonces, $SLI_3 = \{LI_{1,5}^3, LI_{2,1}^2, LI_{3,2}^4\} = \{\{FP, P, VP\}, \{VN, N\}, \{N, FN, NR, FP\}\}$ representa la conjunción de disyunciones ((x_1 **is** FP) **OR** (x_1 **is** P) **OR** (x_1 **is** VP)) **AND** ((x_2 **is** VN) **OR** (x_2 **is** FN)) **AND** ((x_3 **is** N)

OR (x_3 **is** FN) **OR** (x_3 **is** NR) **OR** (x_3 **is** FP)).

Así, una **regla difusa temporal** $R_{p,u}$ es una regla difusa que verifica: (1) Su antecedente es un SLI_m; (2) Su consecuente es una etiqueta $SC^w \in SC$; (3) Su SLI_m tiene asociado un conjunto de ejemplos consecutivos $E_{p,u}$. $E_{p,u}$ indica los instantes que una RDT representa. Luego una RDT $R_{p,u}$ es una regla difusa que representa los instantes del p al u en un SD. Se pueden comparar dos RDTs, así, $R_{k,l}$ está incluida en $R_{i,j}$ si el índice i es menor o igual que el índice k y el índice j es mayor o igual que l. Una regla $R_{i,j}$ es menor que $R_{k,l}$ si $R_{k,l}$ no está incluida en $R_{i,j}$ y $i < k$. De la misma forma se define cuando una RDT es igual a otra RDT, o cuando una RDT es mayor que otra. Utilizando estos conceptos se pueden ordenar las RDTs. Como puede verse, el orden del cada regla viene dado por el conjunto de ejemplos asociado a cada SLI_m.

Para finalizar esta sección, el algoritmo 1 ofrece el método de inferencia de los MDTs. Para comprender su comportamiento se estudiará la función de pertenencia de $LI^c_{j,p}$. La ecuación 1 define la función de pertenencia de un $LI^c_{j,p}$.

$$\mu_{LI^c_{j,p}}(a_j) = \sum_{SA^z_j \epsilon LI^c_{j,p}} \mu_{SA^z_j}(a_j) \tag{1}$$

donde a_j es un valor discreto definido en el dominio de la variable continua X_j, $z \in [p..c-1]$.

La ecuación 2 define la función de pertenencia de un SLI_m.

$$\mu_{SLI_m}(e_i) = *(\mu_{LI^{c_j}_{j,p_j}}(x_j)) \tag{2}$$

donde e_i es un ejemplo del conjunto E, $j\epsilon[1..m]$ y * es una t-norma.

Algoritmo 1 Método de Inferencia

```
j ← 1
for i = 1 to |E| do
   if μ_{SLI^j_m}(e_i) > μ_{SLI^{j+1}_m}(e_i) then
      s ← Cons(R_j)
   else
      s ← Cons(R_{j+1})
      j ← j + 1
   end if
end for
```

El algoritmo de inferencia necesita como entrada un conjunto de ejemplos. El ejemplo y regla actual se indican mediante los índices i y j respectivamente. El bucle *for* se utiliza para recorrer el conjunto de ejemplos E. Si el grado de pertenencia de e_i al SLI_m de R_j (SLI^j_m) es mayor que el grado de pertenencia de e_i al SLI_m de R_{j+1} (SLI^{j+1}_m) la salida es la etiqueta de salida de R_j ($s \leftarrow Cons(R_j)$), en otro caso, la salida es la etiqueta de salida de R_{j+1} ($s \leftarrow Cons(R_{j+1})$) y la nueva regla actual es R_{j+1} ($j \leftarrow j+1$). Se necesita un proceso de defuzzification [4, 10] para obtener un valor discreto de salida.

3 ALGORITMO DE INDUCCIÓN

Esta sección está dividida en dos subsecciones. La primera explica dos conceptos necesarios para comprender el algoritmo de inducción, y la segunda muestra el funcionamiento del mismo.

3.1 ALGUNAS DEFINICIONES

Primero se verá cómo obtener el SLI_m de una RDT que representa un conjunto de ejemplos consecutivos. Para hacer esto, se simplifica el SLI_m con todas las posibles etiquetas para cada uno de los m intervalos (todas las etiquetas de los conjuntos $SA_1 \ldots SA_m$ respectivamente). El método de simplificación consiste en que se cambian la primera y última etiquetas para cada intervalo del SLI_m, luego el número de etiquetas se reduce.

La primera y última etiqueta (SA^f_j y SA^l_j) de cada intervalo $LI^{c_j}_{j,p_j}$ son la primera y última que verifican las ecuaciones 3 y 4 respectivamente.

$$\exists e_i \epsilon E_{p,u} / \mu_{SA^f_j}(x^i_j) > 0 \tag{3}$$

$$\exists e_i \epsilon E_{p,u} / \mu_{SA^l_j}(x^i_j) > 0 \tag{4}$$

donde x^i_j es el valor real en la posición j del ejemplo e_i.

Así, las nuevas primera y última etiquetas son la primera SA^f_j y última SA^l_j en $LI^{c_j}_{j,p_j}$ tal que uno o más ejemplos en E_{SLI_i} tienen un grado de pertenencia a SA^f_j y SA^l_j mayor que cero respectivamente. El intervalo simplificado $LI^{l-f+1}_{j,f}$ tiene $l-f+1$ etiquetas.

El segundo concepto es la definición de unión total de RDTs. Primeramente, se mostrarán las definiciones de *unión de intervalos* y *unión de conjuntos ordenados de intervalos* (definiciones 3.1 y 3.2 respectivamente), y finalmente, se estudiará la definición de unión total de RDTs.

Definición 3.1 *Sean n $LI^{c_{j_1}}_{j,p_{j_1}} \ldots LI^{c_{j_n}}_{j,p_{j_n}}$ definidos sobre la variable X_j, su unión es el intervalo $LI^{u-p+1}_{j,p}$ donde: (1) La primera etiqueta SA^p_j de $LI^{u-p+1}_{j,p}$ es la menor de $LI^{c_{j_1}}_{j,p_{j_1}} \ldots LI^{c_{j_n}}_{j,p_{j_n}}$; (2) La última etiqueta SA^u_j de $LI^{u-p+1}_{j,p}$ es la mayor de $LI^{c_{j_1}}_{j,p_{j_1}} \ldots LI^{c_{j_n}}_{j,p_{j_n}}$; (3) El intervalo está compuesto por cada etiqueta desde SA^p_j a SA^u_j, esto es, $LI^{u-p+1}_{j,p} = \{SA^p_j \ldots SA^u_j\}$.*

Definición 3.2 *Sean n $SLI_m^1 \ldots SLI_m^n$, su unión se define como otro SLI_m, donde cada intervalo $LI_{j,p_j}^{c_j}$ de SLI_m se obtiene realizando la unión de los n intervalos $LI_{j,p_{j_1}}^{c_{j1}} \ldots LI_{j,p_{j_n}}^{c_{jn}}$ de $SLI_m^1 \ldots SLI_m^n$ (definición 3.1).*

Definición 3.3 *Sean n RDTs $R_{p_1,u_1} \ldots R_{p_n,u_n}$, su unión se define como la regla $R_{p,u}$ donde: (1) Su SLI_m se obtiene realizando la unión de los n $SLI_m^1 \ldots SLI_m^n$ de $R_{p_1,u_1} \ldots R_{p_n,u_n}$ (definición 3.2); (2) El conjunto de ejemplos asociado es igual a $E_{SLI_m} = \bigcup E_{SLI_m^j}$ donde $j \in [1..n]$; (3) La etiqueta de salida se calcula utilizando la ecuación 5.*

3.2 INDUCCIÓN DE MDTs

El algoritmo 2, llamado Algoritmo Directo de Inducción de MDTs, se utiliza para inducir los MDTs. Sus entradas son: un conjunto de ejemplos E; los conjuntos $SA_1 \ldots SA_m$ para las m variables de entrada; y el conjunto SC para la variable de salida.

La idea del algoritmo consiste en recorrer los ejemplos de E, para cada ejemplo se calcula la RDT que lo representa y si la RDT obtenida tiene la misma etiqueta de salida que la RDT que representa sus ejemplos anteriores entonces se realiza la unión total de las dos RDTs, en otro caso, se añade la RDT que representa los ejemplos anteriores al MDT.

En este algoritmo, cada RDT se representa como una tupla con tres componentes: $R_i = \{SLI_m^i, E^i, SC^a\}$, donde SLI_m^i representa el antecedente de R_i, E^i es el conjunto de ejemplos consecutivos asociado al SLI_m^i, y finalmente, $SC^a \in SC$ es la etiqueta de salida.

Algoritmo 2 Algoritmo de inducción directa de MDT

$MDT \leftarrow \theta$
$E^{cur} \leftarrow \{e_1\}$
$SLI_m^{cur} \leftarrow Simplify(SA_1 \ldots SA_m)$
$SE^{cur} \leftarrow CalculateConsequent(E^{cur})$
for $i = 2$ to $|E|$ **do**
  $E^i \leftarrow \{e_i\}$
  $SLI_m^i \leftarrow Simplify(SA_1 \ldots SA_m)$
  $SE^i \leftarrow CalculateConsequent(E^i)$
  if $SE^{cur} = SE^i$ **then**
    $R_{cur} \leftarrow TotalUnion(R_{cur}, R_i)$
    $E^{cur} \leftarrow E^{cur} \bigcup E^i$
  else
    $MDT \leftarrow MDT + R_{cur}$
    $R_{cur} \leftarrow R_i$
  end if
end for
$MDT \leftarrow MDT + R_{cur}$

A continuación, se estudia el comportamiento del algoritmo 2. En dicho algoritmo, MDT es el modelo difuso temporal. R_{cur} es una RDT, denominada *regla actual*, que representa los ejemplos anteriores a la RDT que representa el ejemplo e_i (R_i). Sus tres componentes son: SLI_m^{cur}, E^{cur} y SE^{cur}. R_i es la RDT que representa el ejemplo número i en E, sus tres componentes son SLI_m^i, E^i y SE^i.

Primeramente, el MDT se inicializa a θ ($MDT \leftarrow \theta$) y se crea la primera regla actual R_{cur} que representa a e_1, proceso que se realiza en tres acciones: (1) E^{cur} es inicializado a $\{e_1\}$ ($E^{cur} \leftarrow \{e_1\}$); (2) Cada intervalo $LI_{j,p_j}^{c_j}$ del SLI_m^{cur} de R_{cur} se inicializa a la simplificación del conjunto ordenado de etiquetas SA_j utilizando como conjunto de ejemplos asociado $E^{cur} = \{e_1\}$ ($SLI_m^{cur} \leftarrow Simplify(SA_1 \ldots SA_m)$); (3) Se calcula la etiqueta de salida de la primera regla actual R_{cur} utilizando la ecuación 5 ($SE^{cur} \leftarrow CalculateConsequent(E^{cur})$):

$$max_{SC^w} \mu_{SC^w}(v) \tag{5}$$

donde w=1..i_y y $v = \frac{\sum_{i=1}^{|E^{cur}|} s^i}{|E^{cur}|}$, siendo $|E^{cur}|$ el número de ejemplos en E^{cur} y s^i el valor de salida de cada e_i que pertenece a E^{cur}.

En resumen, la etiqueta seleccionada para la salida de la regla es la que tiene el máximo grado de pertenencia al valor medio de los valores de salida de los ejemplos de E^{cur}. Se pueden ver otras posibilidades en [4, 10].

Posteriormente, el bucle **for** se utiliza para examinar los ejemplos del e_2 al último ejemplo e_n de E. Para cada ejemplo e_i se calcula la regla R_i donde: SLI_m^i se asigna a la simplificación de $SA_1 \ldots SA_m$ utilizando $E^i = \{e_i\}$ como conjunto de ejemplos asociado ($SLI_m^i \leftarrow Simplify(SA_1 \ldots SA_m)$); y la etiqueta de salida SE^i se calcula utilizando la ecuación E^i ($SE^i \leftarrow CalculateConsequent(E^i)$).

Después, si la etiqueta de salida de R_{cur} (SE^{cur}) es igual a la etiqueta de salida de R_i (SE^i), significa que las dos reglas (que representan conjuntos de ejemplos consetivos en el tiempo) tienen la misma salida, así, se asigna R_{cur} a la unión total de las reglas R_{cur} y R_i (definición 3.3), porque representan la misma salida ($R_{cur} \leftarrow TotalUnion(R_{cur}, R_i)$); y E^{cur} se asigna a la unión de E^{cur} y E^i ($E^{cur} \leftarrow E^{cur} \bigcup E^i$). En otro caso, R_{cur} se añade a MDT ($MDT \leftarrow MDT + R_{cur}$) y R_{cur} se asigna a R_i ($R_{cur} \leftarrow R_i$). Cuando el bucle **for** termina, la última regla actual R_{cur} se añade a MDT ($MDT \leftarrow MDT + R_{cur}$).

4 DESCRICIPCIÓN DEL MÉTODO

Esta sección muestra el comportamiento del método. Para ello se obtiene un modelo del endeudamiento de

Andalucía desde 1987 a 2000. La variable de salida es el endeudamiento D/PIB, y las variables de entrada son los ingresos no financieros INF y el tipo de interés T. Estos datos de entrada al algoritmo han sido obtenidos de varias fuentes fiables como son el Banco Nacional de España, el Ministerio de Hacienda, y el Instituto Nacional de Estadística.

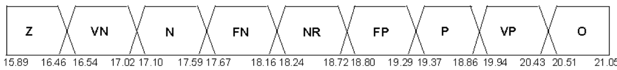

Figura 1: Conjunto de etiquetas de la variable INF

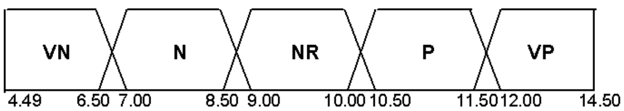

Figura 2: Conjunto de etiquetas de la variable T

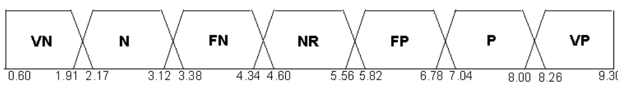

Figura 3: Conjunto de etiquetas de la variable D/PIB

Se utiliza un conjunto ordenado de 9 etiquetas para la variable INF (figura 1), un conjunto ordenado de 5 etiquetas para la variable T (figura 2), y finalmente, un conjunto ordenado de 7 etiquetas para la variable D/PIB (figura 3). En estas figuras Z es cero, VN is Muy Negativo, N es Negativo, FN es Poco Negativo, NR es Normal, FP is Poco Positivo, P es Positivo, VP es Muy Positivo y O es Uno.

La tabla 1 muestra el MDT obtenido. Las figuras 4, 5 y 6 muestra la zona temporal de cada intervalo y etiqueta de salida para las variables de entrada y salida respectivamente. Los recuadros con R_n representan la zona temporal de los intervalos y etiquetas de salida, donde n es el número de regla siendo $R_1, R_2 \dots R_7$ las reglas $R_{1,3}, R_{4,4} \dots R_{10,11}$ respectivamente. Como se puede ver, los intervalos y etiquetas de salida para INF, T y D/PIB son "continuos".

Finalmente, se ha realizado un proceso de inferencia utilizando el algoritmo 1, el conjunto de ejemplos E se utiliza como entrada. El error obtenido es 0.2439. La figura 7 muestra gráficamente la comparación entre la salida real (línea negra) y la salida obtenida (línea gris). La línea obtenida es muy parecida a la real.

5 CONCLUSIONES

Se ha presentado un método para obtener directamente MDTs. Los MDTs son un nuevo método para representar lingüísticamente los SDs. Así, el modelo describe cualitativamente el rango de valores utilizando los intervalos lingüísticos. Se define el conjunto ordenado de etiquetas para cada variable antes de la utilización del algoritmo, como resultado de esto, el modelo obtenido es más comprensible porque el número de etiquetas y el dominio de cada etiqueta se define fuera del algoritmo, por ejemplo, por un profesional en el área de aplicación del algoritmo. Una buena área de aplicación de nuestro método es el deporte. Los MDTs pueden detectar las fases del movimiento mediante un conjunto de RDTs, y explicar cómo cada variable evoluciona a lo largo del tiempo.

Tabla 1: MDT Obtenido.

R.	INF	T	PIB	Ti.
$R_{1,3}$	$\{Z \dots N\}$	$\{P, VP\}$	VN	$\{e_1 \dots e_3\}$
$R_{4,4}$	$\{VN\}$	$\{VP\}$	N	$\{e_4\}$
$R_{5,5}$	$\{Z\}$	$\{VP\}$	FN	$\{e_5\}$
$R_{6,6}$	$\{NR\}$	$\{VP\}$	NR	$\{e_6\}$
$R_{7,7}$	$\{FP\}$	$\{NR, P\}$	FP	$\{e_7\}$
$R_{8,9}$	$\{NR, FP\}$	$\{NR, P\}$	P	$\{e_8, e_9\}$
$R_{10,11}$	$\{P \dots O\}$	$\{VN, N\}$	VP	$\{e_{10} \dots e_{14}\}$

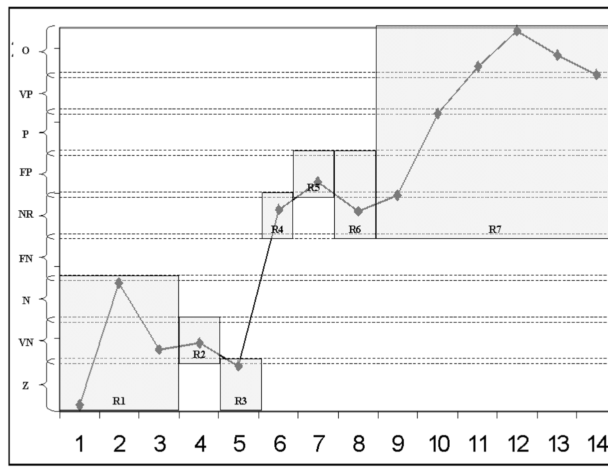

Figura 4: Variable INF

La evolución del tiempo se representa en el orden de las RDTs. Una RDT representa instantes consecutivos con la misma etiqueta de salida. Los MDTs reflejan las variables de entrada que cambian de un instante a los instantes siguientes. Nuestro método no tiene ningún problema cuando se obtienen diferentes valores de salida con los mismos valores de entrada debido al orden de las reglas. Así, es posible encontrar dos o más RDTs con el mismo antecedente y diferente consecuente.

Normalmente, los algoritmos de inducción agrupan mediante las variables de entrada. Nuestro método

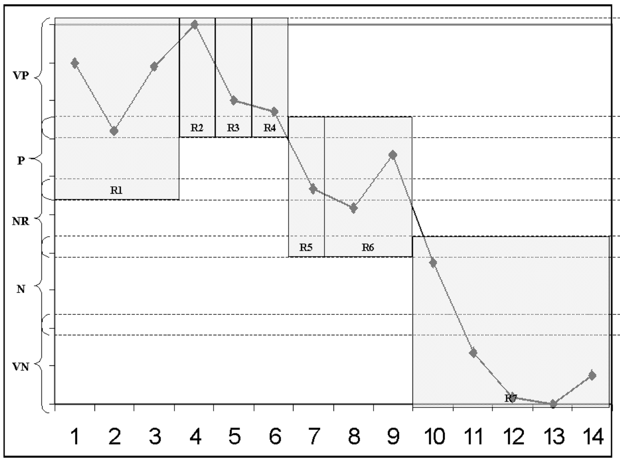

Figura 5: Variable T

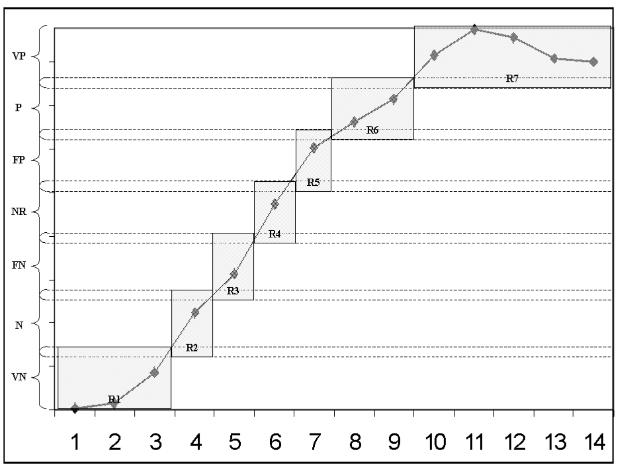

Figura 6: Variable D/PIB

agrupa utilizando la variable de salida, esto es una forma diferente de trabajar si lo comparamos con otros algoritmos.

Los resultados obtenidos confirman que el error obtenido con nuestro método es pequeño, así, los MDTs son una buena aproximación al SD que representan.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Ciencia y Tecnología y la Junta de Comunidades de Castilla-La Mancha mediante los proyectos DIMOCLUST TIC2003-08807-C02-02 y PREDACOM PBC-03-004.

Referencias

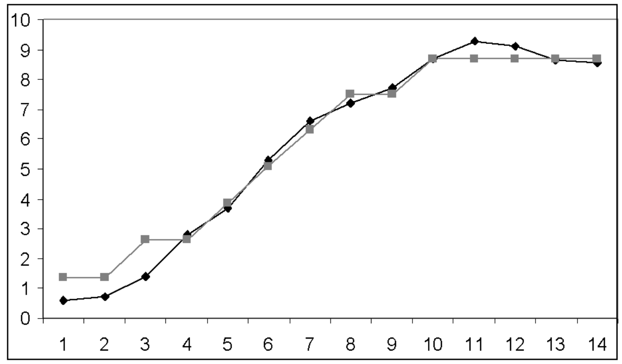

Figura 7: Proceso de inferencia

[1] J.L. Castro, J.J. Castro-Schez, J.M. Zurita. Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. *Fuzzy Sets and Systems* vol. 101, pag. 331-342, 1999.

[2] M.Delgado and A. Gonzalez. An inductive learning procedure to identify fuzzy system. *Fuzzy Sets and Systems*, vol. 55, pag. 121-132, 1993.

[3] A. Gonzalez. A learning methodology in uncertain and impresise enviroments. *International Journal of Intelligence Systems*, vol 10, pag. 357-371, 1995.

[4] W. V. Leekwijck, E. E. Kerre. Defuzzification: criteria and classification. *Fuzzy Sets and Systems*, vol. 108 n.2, pag. 159-178, 1999.

[5] D.G. Luenberger. Introduction to Dynamic Systems: Theory, Models and Applications. *John Wiley and sons*. 1979.

[6] J. Moreno-Garcia, L. Jimenez, J.J. Castro-Schez, L. Rodriguez. A linguistic modelling approach for dynamic systems by using Temporal Fuzzy Models. *Subm. to Int. Journal of Approximate Reasoning*, 2004.

[7] J. Moreno-Garcia, L. Jimenez, J.J. Castro-Schez, L. Rodriguez. A direct linguistic induction method for systems. *Fuzzy Sets and Systems*. Por aparecer. 2004.

[8] K. Ogata. Ingenieria de Control Moderna. *Prentice-Hall HispanoAmericana SA*. 1998.

[9] S.M. Paudit, S.M. Wu. Time Series and Systems Analysis, with applications. New York, Wiley, 1983.

[10] K. Tanaka. An introduction to fuzzy logic for practical aplications. *Springer*. 1998.

[11] L. Zadeh. The concept of a linguistic variable and its applications to Approximate reasoning. Part I,II,III. *Information Science*. 1975.