

# Uso de aprendizaje computacional para extraer modelos del estudiante en el problema del péndulo invertido

Blanca Vargas Govea <sup>1</sup>, Eduardo Morales Manzanares <sup>2</sup> y Rafael Morales Gamboa <sup>3</sup>

<sup>1,2</sup> Instituto Tecnológico y de Estudios Superiores de Monterrey-Campus Cuernavaca, Av. Reforma 182-A, Col. Lomas de Cuernavaca, Temixco, Mor., 62589. México

<sup>1</sup> 00375963@academ01.mor.itesm.mx

<sup>2</sup> emorales@campus.mor.itesm.mx

<sup>3</sup> Instituto de Investigaciones Eléctricas, Av. Reforma 113, Col. Palmira, Temixco, Mor., 62370. México  
rmorales@iie.org.mx

**Resumen.** El inducir modelos que expresen habilidades como andar en bicicleta, pilotear un avión u operar una planta es un reto computacional que permite obtener descripciones explícitas sobre algo que por naturaleza no lo es. Esto se complica mas cuando se quiere inducir un modelo de un aprendiz que intenta adquirir una habilidad, debido a las inconsistencias presentes durante su aprendizaje. En un trabajo previo [7], se demostró que el hacer inspeccionables modelos explícitos de las habilidades de aprendices, mejora notablemente su entendimiento de la tarea y su capacidad de articular su conocimiento. En este trabajo se muestra como obtener modelos explícitos de habilidades de aprendices que son fácilmente interpretables. Para esto se siguen dos estrategias: (i) transformar los datos en descripciones cualitativas y utilizar un algoritmo de aprendizaje proposicional agregando nuevos atributos, y (ii) utilizar directamente un sistema de programación lógica inductiva con conocimiento del dominio. Se muestran los resultados y ventajas de ambos enfoques.

**Palabras clave:** aprendizaje inductivo, péndulo invertido, modelos de habilidades

## 1. Introducción

Cuando una persona aprende una materia como biología, requiere aprender conceptos que posteriormente aplica para resolver problemas. La persona es capaz de describir lo que sabe y puede transmitir su conocimiento, sea correcto o no, de forma explícita. Al resolver un problema, puede proporcionar un conjunto de pasos o reglas que describan la forma en que llegó a una solución y si ésta no es correcta, es posible revisar el proceso y encontrar el error. La descripción que la persona hizo puede ser inspeccionada y programada en un sistema.

Si lo que se trata de describir es una habilidad, como manejar una bicicleta o batear una pelota, la dificultad se incrementa ya que este tipo de aprendizaje se realiza a nivel inconsciente y es muy difícil que una persona pueda describir paso a paso lo que hizo.

Modelar este tipo de habilidades representa un reto, puesto que el objetivo es extraer descripciones explícitas sobre algo que por naturaleza no lo es.

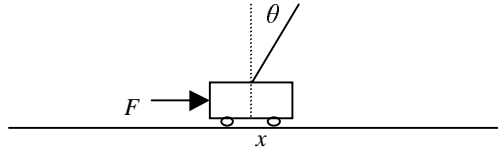
Una vez obtenido el modelo, éste puede tener diversas funciones: puede utilizarse como medio de comunicación, como herramienta de introspección, o de predicción y control, entre las más importantes. En el área de tutores inteligentes, por ejemplo, el modelo del estudiante es un componente fundamental para aplicar la estrategia pedagógica adecuada a cada individuo [12]. Modelar el comportamiento de personas que están aprendiendo presenta además otros problemas, entre ellos, la inconsistencia en sus respuestas o acciones. Si además lo que la persona está aprendiendo es una tarea de control entonces la dificultad se incrementa, pues generalmente este tipo de tareas involucran una gran cantidad de datos de tipo continuo y ruido [1].

Con el fin de comprender y modelar las habilidades humanas, se han estudiado sistemas simples de control dinámico, uno de ellos es el péndulo invertido. El problema consiste en mantener en equilibrio un péndulo que se encuentra sobre un carro el cual se desplaza sobre un riel de longitud limitada. El equilibrio se logra aplicando una fuerza al carro con una dirección determinada. Este sistema de control ha sido objeto de estudio de numerosas investigaciones [13]. En un trabajo previo [7], se aplicó un sistema de aprendizaje para inducir reglas de control, a partir de trazas de comportamiento de aprendices, para el péndulo invertido. Se mostraron diferentes representaciones de las reglas usando una interfaz gráfica. Una de las principales dificultades observadas fue la confusión de los participantes para entender la representación, por lo que después de presentarles explicaciones se facilitó su comprensión. Se demostró en [7, 8] que el usar esta representación y la participación activa del estudiante en la tarea de modelado incrementa su habilidad para articular su conocimiento y es capaz de formular mejor su estrategia de juego.

El objetivo de este trabajo es obtener modelos de las habilidades de control del péndulo invertido de novatos que sean fáciles de interpretar, y por lo mismo que al ser inspeccionables por aprendices, les permita describir y formular mejor su estrategia. Para esto se siguen dos enfoques: (i) se utiliza un algoritmo de aprendizaje proposicional, como en [7], pero los datos son pre-procesados, transformándolos en una representación cualitativa y se añaden nuevos atributos a la descripción de los ejemplos, y (ii) se utiliza un sistema de aprendizaje relacional con conocimiento del dominio. Se presentan y comparan los resultados para ambos enfoques y se muestran sus ventajas y desventajas.

## **2. El péndulo invertido**

El objetivo de esta tarea consiste en mantener en equilibrio un péndulo que se encuentra colocado sobre un carro que rueda sobre un riel limitado. El péndulo está sujeto a la gravedad, originando que se mueva de un lado a otro y el objetivo es que el usuario aprenda a controlar el carro para evitar que el péndulo se caiga.



**Fig. 1.** El péndulo y el carro.  $\theta = 0$  es el ángulo de inclinación del péndulo en equilibrio. Para  $\theta < 0$ , el péndulo se inclina a la izquierda. Para  $\theta > 0$ , se inclina a la derecha.  $x$  es la posición del carro.  $x < 0$  significa que el carro está en la mitad izquierda del riel, mientras que  $x > 0$  significa que el carro está en la mitad derecha.

El estado del sistema está definido por cuatro elementos: posición del carro, velocidad del carro, ángulo del péndulo y velocidad angular del péndulo. A partir de esta información, se debe determinar la dirección en la cual se empuja el carrito, aplicando una fuerza constante a la derecha o izquierda. Es uno de los problemas de interés porque describe un sistema inherentemente inestable que es representativo de una amplia gama de problemas.

### 3. Datos originales

Los datos utilizados en este trabajo son las trazas correspondientes a 14 participantes que tomaron parte en sesiones de 5 minutos de aprendizaje de control del péndulo invertido practicando en el sistema PACMOD [7]. También se obtuvo el modelo de la traza de un experto [3, 5, 6]. Las trazas constan de 4 atributos: velocidad angular ( $\dot{a}$ ), ángulo  $a$ , velocidad del carro ( $\dot{x}$ ) y posición del carro ( $x$ ), todos con valores continuos. La clase puede tomar cualquier valor de las acciones *right* (ejercer una fuerza a la derecha), *left* (ejercer la fuerza a la izquierda) o *inertia* (no ejercer ninguna fuerza). La tabla 1 muestra tres ejemplos de una traza original.

$\dot{a}$	$a$	$\dot{x}$	$x$	clase
-1.32811	0.138182	2.4245	0.915903	<i>right</i>
-1.57452	0.11162	2.61763	0.964393	<i>left</i>
-2.37603	0.001579	3.20006	1.13308	<i>inertia</i>

**Tabla 1.** Taza original.

Las trazas de los participantes tienen entre 1320 y 1902 ejemplos mientras que la del experto tiene 3530.

### 4. Metodología

La metodología que se siguió para la construcción de los modelos consiste básicamente en los siguientes pasos:

1. Pre-procesar la traza original, transformando los valores continuos en valores cualitativos. Agregar nuevos atributos, obteniendo tendencias, aceleración y dirección. Inducir los modelos con RIPPER.
2. Plantear el conocimiento del dominio, expresar los ejemplos en forma de programas lógicos y finalmente inducir el modelo en Progol. La figura 2 esquematiza el proceso.

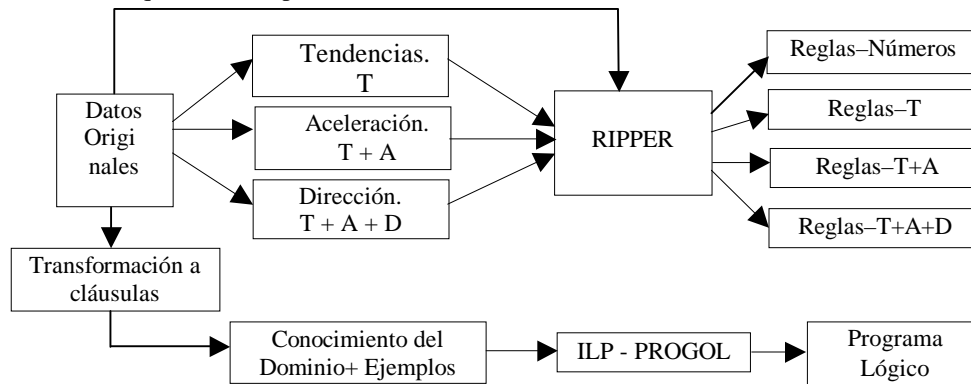


Fig. 2. Diagrama de la metodología seguida para la obtención de los modelos.

#### 4.1 Pre-procesamiento

A partir de la traza numérica se obtuvieron 3 trazas con valores cualitativos:

1. Tendencias. Los valores de los 4 atributos se transformaron de acuerdo a su comportamiento. Se utilizaron dos ejemplos para obtener su tendencia. Los posibles valores de los atributos son “aumenta”, “disminuye” o “noCambia”. De esta manera, la descripción cambia de numérica: “-1.57452, 0.11162, 2.61763, 0.964393, left” a cualitativa: “disminuye, noCambia, aumenta, aumenta, left”.
2. Aceleración. Se agregaron dos atributos a la traza: la aceleración angular y la aceleración del carro. Se utilizaron tres ejemplos para obtener la aceleración. Además, se cambió la descripción del ángulo del péndulo por los valores: “alaIzquierda” y “alaDerecha” de acuerdo a la dirección de caída del mismo. Un ejemplo pasaría del tipo: “-1.57452, 0.11162, 2.61763, 0.964393, left” a “constante, disminuye, noCambia, constante, aumenta, aumenta, left” donde el primero y cuarto atributo corresponden a la aceleración del péndulo y el carro respectivamente.
3. Dirección. En esta transformación se modificaron los valores del ángulo del péndulo por los valores “levantandoDerecha”, “cayendoIzquierda”, “levantandoIzquierda”, “cayendoDerecha” y “equilibrio”. También se modificaron los valores de la posición del carro por “alejaIzq”, “alejaDer”, “acercaIzq”, “acercaDer” y “noCambia”. Esto indica si el carro se aleja o acerca del centro y si se encuentra a la izquierda o a la derecha del centro o casi no se mueve. El ejemplo “-1.57452, 0.11162, 2.61763, 0.964393, left” se

transforma en “constante, disminuye, levantandoIzquierda, constante, aumenta, alejaDer, left”.

## 4.2 Inducción de modelos en RIPPER

Antes de obtener los modelos para las trazas pre-procesadas, se obtuvieron los modelos de la traza original con el fin de observar el porcentaje de error obtenido por el modelo, los números de reglas obtenidos y la claridad de las reglas. En la tabla 3 se concentran los resultados.

### 4.2.1 Definición de umbral

Para la transformación de los datos originales en descripciones cualitativas, es necesario definir un umbral que nos indique cuando un cambio en el valor de los datos continuos es suficientemente alto como para considerar que la tendencia está aumentando o disminuyendo. Este umbral se definió por prueba y error observando el porcentaje de error de los modelos obtenidos. Para esto, se tomaron como base los resultados de un participante cuyo porcentaje de error del modelo obtenido a partir de los datos originales fue intermedio con respecto al resto de los participantes.

Primero se obtuvo un umbral para tendencias variando los valores hasta obtener un modelo con un porcentaje de error aproximado al original o menor. Una vez determinado un valor adecuado, se procedió a obtener el umbral para aceleración, manteniendo fijo el umbral de tendencias obtenido previamente y siguiendo los mismos criterios de selección. No se hicieron pruebas variando simultáneamente los umbrales para tendencias y aceleración por lo que no puede garantizarse que los valores obtenidos sean los mejores. La tabla 2 muestra los valores probados para determinar el umbral de tendencias y aceleración para el participante 3.

Tendencias		Aceleración	
Umbral	Error	Umbral	Error
0.001	34.04	0.001	17.61
0.005	24.68	0.005	17.62
0.009	23.92	0.01	17.8
0.01	22.81	0.021	17.1
0.015	20.23	0.025	17.21
0.02	19.82	0.03	17.62
0.025	19.42	0.04	17.33
0.021	19.12	0.045	17.86
0.03	17.89	0.05	17.51
0.035	20.88	0.08	17.33
0.04	17.37	0.1	17.51
0.045	22.28	0.15	18.15
0.05	22.11	0.2	17.74
0.06	25.5	0.17	17.08
0.065	24.5	0.18	18.33
0.07	28.19	0.21	17.45

**Tabla 2.** Selección de umbrales

#### 4.2.2 Resultados

Los modelos inducidos por RIPPER [2] se obtuvieron utilizando asumiendo el siguiente orden en las clases: *right*, *left*, *inertia*. La tabla 3 muestra los resultados, donde P es el número de participante, la columna “Cambios” indica el número de transiciones entre acciones, es decir, cuántos cambios *right-left*, *left-right*, *right-inertia*, *inertia-right*, *inertia-left*, *left-inertia* existen.

P	Cambios				Numéricos		Tendencias		Aceleración		Ace/Dirección	
		<i>right</i>	<i>left</i>	<i>inertia</i>	% Error	# Reglas	% Error	# Reglas	% Error	# Reglas	% Error	# Reglas
1	1500	894	897	1924	30.55	11	47.34	1	43.05	9	38.36	5
2	505	591	452	498	19.97	9	28.16	9	29.50	9	20.01	9
3	509	695	661	354	17.42	14	17.37	10	16.45	19	16.74	15
4	414	1027	819	234	10.81	13	10.24	10	8.95	19	9.10	16
5	427	572	639	391	21.21	14	23.60	11	20.19	17	20.69	16
6	348	416	534	376	22	11	33.33	8	34.44	8	30.97	10
7	440	949	896	212	13.56	16	11.58	12	9.69	18	10.61	16
8	461	360	482	545	19.96	10	23.58	10	20.58	16	20.58	16
9	387	570	660	312	21.97	9	29.77	11	26.36	14	25.52	9
10	444	453	571	398	22.14	15	32.91	9	32.32	13	30.99	11
11	453	892	790	282	13.28	16	15.17	11	14.42	14	12.69	17
12	499	729	550	318	24.28	12	34.31	8	33.48	13	30.53	16
13	515	687	613	381	22.06	11	22.07	11	20.01	17	18.64	16
14	510	717	515	381	25.03	9	28.77	10	26.75	13	26.44	17
Prom.					20.30	12.14	25.29	9.36	24.01	14.21	22.28	13.5
D.S.					5.19	2.54	10.10	2.68	9.99	3.68	8.60	3.88
Experto	61	224	215	1897	0.88	19	2.62	2	2.65	2	2.66	3

**Tabla 3.** Resultados para los modelos inducidos con RIPPER

Puede observarse que en el rango de 400-499 cambios se encuentran los menores porcentajes de error (inferiores a 15%) tomando en cuenta únicamente a los participantes. Para el caso del experto esto no es válido porque presenta 61 cambios únicamente a pesar de que su traza es la que contiene el mayor número de ejemplos. Las columnas “*right*”, “*left*” e “*inertia*” indican el número acciones para cada clase efectuadas. Se presentan primero los resultados de los modelos generados a partir de la traza numérica en términos de porcentaje de error y número de reglas. Después con tendencias, tendencias y aceleración, tendencias, aceleración e información sobre dirección de cambio. Finalmente se muestran los promedios, desviación estándar y los resultados para los modelos del experto. El conjunto de entrenamiento y prueba es el mismo, es decir, toda la traza.

#### 4.2.3 Observaciones

- Se observa que en el modelo para la traza de tendencias, el porcentaje de error tiende a aumentar con respecto al modelo obtenido a partir de la traza numérica. Para la traza numérica el error en promedio es de 20.30%; para la traza de

tendencias aumenta a 25.59%. Al agregar la aceleración, el porcentaje disminuye a 24.01% y cuando se incluye la dirección del péndulo y el carro, disminuye a 22.28%. La diferencia entre el porcentaje de la última transformación con respecto al porcentaje del modelo generado a partir de la traza original muestra un incremento de 1.98%. Sin embargo, transformar los datos y agregar información permite que el modelo se interprete más fácilmente.

- El número de reglas por participante es entre 5 y 22, y aunque en algunos casos el número de reglas es mayor que con el modelo numérico original, la claridad parece aumentar. En promedio, para la traza numérica el número de reglas es de 12.14, para tendencias es de 9.36, para aceleración es de 14.21 y para aceleración más dirección es de 13.5.
- Cuando el participante muestra un porcentaje de error muy alto en el modelo original, el porcentaje obtenido con las trazas transformadas se incrementa mucho en comparación con el resto de los participantes. En el caso del participante 1 el error se incrementa de 30.55% a 38.36%, sin embargo, al analizar su desempeño vemos que tiene el número más alto de transiciones y es la traza con el mayor número de ejemplos. Esto indica que realizó muchas acciones en su sesión, su desempeño fue notablemente inconsistente y esto se refleja en el porcentaje de error tan alto.
- En general no se logró una disminución del porcentaje de error en los modelos. En el caso del valor para el ángulo, aunque los modelos nos indican la dirección del péndulo, no podemos saber con precisión en qué parte del cuadrante se encuentra. Podemos saber si se está cayendo hacia la derecha pero no sabemos si está casi en posición horizontal, si está en medio o cerca del equilibrio. Esta pérdida de información puede originar que el porcentaje de error no disminuya. Otro posible factor es el valor del umbral ya que no puede garantizarse que los obtenidos sean los que proporcionen los mejores resultados.
- El experto presenta un comportamiento muy distinto al de los participantes novatos. El porcentaje de error obtenido por el modelo numérico es muy bajo (0.88%). Al obtener los modelos a partir de las transformaciones, el número de reglas disminuye notablemente. Esto en realidad no es deseable ya que se esperaría que un modelo del experto proporcionara una buena estrategia, sin embargo no es así. Este comportamiento había sido observado en [11] con trazas de pilotos expertos. El porcentaje de error aumenta de 0.88% para la traza original, a 2.62% para tendencias, 2.65% para aceleración y 2.66% para aceleración más dirección. Parece que el conocimiento adicional no influye. Es notable que el número de reglas disminuye de 19 a 2, 2, y 3 respectivamente. Esto se explica porque el experto toma el control del dispositivo inmediatamente. No existe en su traza una gran diversidad de estados y acciones que originen que se obtengan reglas para diferentes estados del dispositivo.

#### 4.2.4 Ejemplos de reglas

A manera de ejemplo se muestran algunas reglas de los modelos del participante 13 obtenidos a partir de la traza original, agregando tendencias y aceleración más dirección. Se eligió a este participante porque en él se observa una disminución en el porcentaje de error. El par de números a la izquierda de la acción corresponden al

número de positivos y falsos positivos respectivamente para la regla. Se muestra una regla para cada acción.

**De la traza original.** Este modelo es difícil de interpretar. Se necesita conocer los posibles valores de los atributos y su significado.

*right* 181 9 IF anguloPendulo  $\geq$  0.031539 posicionCarro  $\leq$  -1.39321  
velocidadAngular  $\geq$  1.07392.  
*left* 232 30 IF anguloPendulo  $\leq$  -0.154342 velocidadCarro  $\leq$  -0.510581.  
*inertia* 226 192 IF.

**Tendencias.** Este modelo no resulta muy claro ya que no se puede determinar el estado del carro ni del péndulo.

*right* 256 21 IF velocidadCarro = aumenta posicionCarro = noCambia  
anguloPendulo = noCambia .  
*left* 331 57 IF velocidadCarro = disminuye anguloPendulo = noCambia .  
anguloPendulo = noCambia .  
*inertia* 214 156 IF .

**Aceleración.** Para este modelo se agregaron los atributos de aceleración angular y del carro.

*right* 288 15 IF velocidadCarro = aumenta aceAngular = constante  
anguloPendulo = noCambia .  
*left* 265 16 IF aceAngular = constante velocidadCarro = disminuye .  
velocidadAngular = noCambia .  
*inertia* 224 127 IF .

**Aceleración y dirección.** En este modelo se agrega la aceleración angular, la aceleración del carro y la dirección del péndulo. Esto hace más fácil su interpretación.

*right* 231 16 IF anguloPendulo = cayendoDerecha velocidadAngular = disminuye .  
velocidadCarro = aumenta .  
*left* 213 12 IF anguloPendulo = cayendoIzquierda acelAngular = constante .  
*inertia* 275 168 IF .

**Modelo del experto para aceleración más dirección.** Los modelos del experto no proporcionan información significativa sobre su desempeño al controlar el péndulo.

*right* 195 6 IF velocidadAngular = disminuye acelCarro = aumenta .  
*right* 240 30 IF velocidadAngular = disminuye velocidadCarro = aumenta .  
*left* 425 40 IF velocidadAngular = aumenta .  
*inertia* 4010 57 IF .

### 4.3 Inducción de modelos con Progol

Para construir los modelos, Progol requiere como entrada un conjunto de ejemplos, conocimiento del dominio y definiciones de modo [9, 10]. Estos tres elementos deben ser expresados en forma de programa lógico, en este caso Prolog.

El conjunto de ejemplos se dividió en dos grupos. Se tomó la traza de un participante aleatoriamente y se hicieron pruebas con 100, 300 y 500 ejemplos de



entrenamiento. Posteriormente el modelo se probó con la traza completa y el modelo inducido con 300 ejemplos fue el que obtuvo menor porcentaje de error (1.19%) en comparación con los obtenidos para 100 (6.36%) y 500 (30.84%). Por lo tanto, para la inducción del modelo se utilizó un conjunto de 300 ejemplos de entrenamiento y como ejemplos de prueba se usó la traza completa.

### 4.3.1 Predicados definidos y determinación del umbral

Se agregaron los predicados para obtener tendencias y aceleraciones a partir de los datos numéricos, como conocimiento del dominio. En las pruebas iniciales se utilizaron los mismos valores de umbral que para RIPPER pero los resultados no fueron buenos. El porcentaje de error fue alto. Se observó que Progol redondea los valores al realizar el proceso de inducción. Esto influye en que el mismo valor de umbral no sea útil. Mientras que en RIPPER se utilizó un mismo umbral para todos los valores de atributo en Progol no fue así. Se realizaron pruebas con el participante 3 y se seleccionó el umbral que generara un modelo con el menor número de cláusulas sin generalizar, de esta forma se determinaron los siguientes valores: para las tendencias de todos los valores de atributo se utilizó un umbral de 0.01, para la aceleración angular y del carro el umbral es de 0.11 y para la posición del carro el umbral es 0.1.

### 4.3.2 Resultados

Una vez planteadas las definiciones de modo, conocimiento del dominio, formato de ejemplos y umbral, se indujeron los modelos para 14 participantes. Los resultados del porcentaje de error y el número de reglas se muestran en la tabla 4.

P	<i>Cambios</i>	<i>right</i>	<i>left</i>	<i>inertia</i>	Tendencias		Ace/Dirección	
					%Error	# Reglas	%Error	# Reglas
1	1500	894	897	1924	3.68	15(10)	1.00	19 (0)
2	505	591	452	498	10.06	13(11)	2.21	21 (0)
3	509	695	661	354	18.36	11 (23)	1.87	17 (0)
4	414	1027	819	234	8.56	11 (17)	5.68	16 (1)
5	427	572	639	391	13.55	9 (25)	0.69	22 (0)
6	348	416	534	376	6.49	14 (14)	1.28	17 (0)
7	440	949	896	212	13.76	12 (13 )	2.19	9 (1)
8	461	360	482	545	5.05	12 (16 )	15.37	10 (1)
9	387	570	660	312	23.28	13 (18 )	8.57	19 (1)
10	444	453	571	398	7.24	10 (35 )	1.13	16 (0)
11	453	892	790	282	3.82	10 (12 )	11.21	16 (1)
12	499	729	550	318	10.9	10 (25 )	12.28	17 (0)
13	515	687	613	381	11.15	14 ( 12)	4.88	12 (0)
14	510	717	515	381	11.16	13 (13 )	9.06	17 (0)
Prom.					10.5	11.93	5.53	16.29
D.S.					5.53	1.82	4.91	3.75
Experto	61	224	215	1897	1.19	11 (0)	0	18(0)

**Tabla 4.** Resultados de los modelos inducidos con Progol

Para la columna de reglas, el número entre paréntesis indica el número de cláusulas que no fueron generalizadas. Cuando Progol trata de encontrar una generalización de la cláusula aterrizada (ejemplo) y no tiene éxito, lo incluye en el modelo resultante. Lo que se espera es que el modelo no tenga cláusulas aterrizadas.

### 4.3.3 Observaciones

- Los modelos de tendencias tienen cláusulas que no pudieron ser generalizadas, esto se explica porque los participantes son novatos y no presentan consistencia en su desempeño, su traza tiene mucho ruido y el conocimiento del dominio que se agrega no es suficiente para que el modelo mejore su exactitud.
- Para las pruebas de aceleración más dirección no se incluye la velocidad. Se realizó una prueba previa para el participante 3 y se obtuvo un porcentaje de error alto (36.43%), aunque no puede afirmarse que el agregar velocidad reste exactitud debido a que no se hicieron pruebas exhaustivas con valores de umbral distintos para este caso.
- Al agregar más conocimiento del dominio, en este caso la aceleración más la dirección, el número de cláusulas sin generalizar prácticamente desapareció. El porcentaje de error obtenido también disminuye. De 10.50% que obtiene en tendencias se reduce a 5.53% en promedio para aceleración más dirección.
- En cuanto al número de reglas se observa un incremento que en promedio sube de 11.93 para tendencias a 16.29 para aceleración más dirección.
- A diferencia de los participantes, el modelo del experto nunca presenta cláusulas sin generalizar. Su porcentaje de error es el más bajo; en el modelo para aceleración más dirección, su porcentaje de error es de 0%.
- Conforme se va agregando conocimiento del dominio y el predicado a aprender incluye más variables, el tiempo que tarda Progol en obtener los modelos se incrementa considerablemente. La ventaja es que el modelo presenta mayor exactitud.
- Se observa que la incorporación del conocimiento del dominio directamente en la definición, sin necesidad de pre-procesar la traza mejora los resultados.

### 4.3.4 Características de los modelos

Los modelos obtenidos permiten ver las relaciones entre las variables. Para el caso de las tendencias, el predicado tiene como entrada dos estados con sus respectivas variables : velocidad angular, ángulo, velocidad del carro y posición. En el cuerpo del predicado se observa la forma en que estas variables se relacionan y su efecto. Por ejemplo, un predicado para tendencias:

$$\text{ej}(\text{dato1}(A,B,C,D),\text{dato2}(E,F,G,H),\text{right}) :- \text{tendApunto}(A,E,\text{aumenta}), \\ \text{tendA}(B,F,\text{aumenta}), \\ \text{tendXpunto}(C,G,\text{disminuye}).$$

“Si la velocidad angular aumenta, el ángulo se hace más grande y la velocidad del carro disminuye, entonces empuja a la derecha”.

Se observan las tendencias de las variables y la acción efectuada. A y E representan la velocidad angular, B y F representan el ángulo y C y G la velocidad del carro. Esta característica resulta favorable si las relaciones que se obtienen tienen un significado en el modelo. En este ejemplo, cada variable se relaciona con otra del mismo tipo en el segundo estado por lo que no existe confusión pero esto no siempre es así. En ocasiones, el modelo muestra relaciones de variables que no tienen sentido para el usuario, aunque un especialista si sea capaz de interpretarlas.

### **Ejemplo de un modelo obtenido con Progol**

La interpretación es similar a la del modelo de tendencias, sin embargo, al predicado se le ha agregado un nuevo estado y al cuerpo se le incorporó dirección del péndulo y el carro así como también la aceleración. Veamos el siguiente ejemplo:

```
ej(dato1(A,B,C,D),dato2(E,F,G,H),dato3(I,J,K,L),left) :-  
    pendulo(F, J,cayendoIzquierda),  
    aceAngulo(E,E,I,constante),  
    aceCarro(C,G,C,aumenta).
```

Se interpreta como: “ Si el péndulo está cayendo a la izquierda, la aceleración del ángulo del péndulo es constante y la aceleración del carro aumenta, empuja a la izquierda”.

## **5. Comparación de resultados: RIPPER/Progol**

Los resultados obtenidos por RIPPER son adecuados aunque el porcentaje de error obtenido en general no disminuye. No permite incorporar conocimiento del dominio y es necesario pre-procesar la traza original. Los modelos se construyen de forma rápida y el umbral se puede determinar de forma relativamente simple.

Con Progol los resultados son mejores puesto que los modelos presentan un porcentaje de error bajo. El conocimiento del dominio se integra en la misma definición por lo que no se requiere pre-procesamiento. La obtención del umbral se complica. El tiempo de construcción de los modelos se incrementa conforme se agrega conocimiento del dominio y aumenta el número de ejemplos.

## **6. Conclusiones y trabajo futuro**

En este trabajo se obtuvieron modelos de habilidades humanas de novatos que aprenden a controlar un péndulo invertido. Esta tarea de control implica variables continuas, por lo que los modelos obtenidos a partir de las trazas originales resultan difíciles de interpretar. Para solucionar este problema, se siguieron dos estrategias: (i) las trazas se pre-procesaron, transformando los valores cuantitativos a cualitativos a fin de obtener reglas que resultaran más fáciles de entender. Se realizó la inducción con RIPPER y (ii) se obtuvieron los modelos utilizando Progol, sistema de programación lógica inductiva que permite la incorporación directa del conocimiento del dominio. El

proceso también se hizo con la traza de un experto y se compararon sus resultados con los novatos.

Los resultados obtenidos en este trabajo sugieren que el incorporar conocimiento del dominio permite construir modelos más fáciles de interpretar. Se observa además que el uso de programación lógica inductiva disminuye notablemente el porcentaje de error de los modelos.

Como trabajo futuro se considera la evaluación de los modelos como controladores. Es conveniente probar los modelos obtenidos en el simulador del que se tomaron las trazas para evaluar su comportamiento [7]. Otro aspecto es determinar una forma de obtener el umbral para la transformación de manera que garantice que el error en el modelo será el mínimo. Se pueden utilizar puntos de inflexión y podría probarse un mismo umbral para todos los atributos o un umbral diferente por cada uno. También se considera el realizar el mapeo de los valores numéricos de las trazas a valores cualitativos utilizando otras técnicas, como por ejemplo lógica difusa y finalmente mejorar la calidad descriptiva de los modelos agregando otros valores nominales que describan mejor el estado del dispositivo.

## Referencias

- [1] Camacho, R. *Inducing Models of Human Control Skills*. En 10th European Conference on Machine Learning (ECML-98), Chemnitz, Germany, 1998.
- [2] Cohen, W. *Fast Effective Rule Induction*. En Proceedings of the Twelfth International Conference, 1995.
- [3] Freyre, J. *Clonación conductista de habilidades de control humano: un enfoque bayesiano*. ITESM, 2000.
- [4] Hume, D. y Sammut, C. *Applying Inductive Logic Programming in Reactive Environments*. En Proceedings of the 1st International Workshop on Inductive Logic Programming, 1991.
- [5] Michie, D., Camacho, R. *Building Symbolic Representations of Intuitive Real-Time Skills from Performance Data*. En the Machine Intelligence 13 Workshop, Glasgow, U.K., 1992.
- [6] Michie, D. y Sammut, C. *Behavioural Clones and Cognitive Skill Models*. En K. Furukawa, Machine Intelligence 14. Oxford University Press, 1995.
- [7] Morales, R. *Exploring Participative Learner Modelling and its Effects on Learner Behaviour*. Ph.D. Thesis University of Edinburgh, 2000.
- [8] Morales, R., Pain, H., Conlon, T. *Effects of Inspecting Learner Models on Learners' Abilities*. Proc. AIED 2001, pp. 434-445. 13, 2001.
- [9] Muggleton, S. *Inductive Logic Programming*. New Generation Computing, 8(4):295-318, 1991.
- [10] Muggleton, S. y J. Firth. *CPROGOLA.4: A Tutorial Introduction*. En S. Dzeroski y N. Lavrac, editors, Relational Data Mining, pp. 160-188. Springer-Verlag, 2001.
- [11] Sammut, C., Hurst, S., Kedzier, D. & Michie, D. *Learning to Fly*. En Proceedings of the Ninth International Conference on Machine Learning, Aberdeen: Morgan Kaufmann, pp. 385-393, 1992.
- [12] Wenger, E. *Artificial intelligence and Tutoring Systems*. En Computational approaches to the communication of knowledge. Los Altos, Morgan Kaufmann, 1987.
- [13] Wieland, A. P. *Evolving Neural Network Controllers for Unstable Systems*. En Proceedings of the International Joint Conference on Neural Networks, pp. 2,667-2,673, 1991.