Mejoras en el uso de aprendizaje activo con árboles fuzzy: un ejemplo de su aplicación en la toma de decisiones de un sistema coordinado

Giovanni Acampora[1] **Giuseppe Fenza**[1] **Enrique Muñoz**[2] **Bernardino Romera**[3]

[1] Università degli Studi di Salerno, {gacampora,gfenza}@unisa.it
[2] Universidad de Murcia, enriquemuba@um.es
[3] University College London, bernard24@gmail.com

Resumen

El aprendizaje computacional se ha utilizado frecuentemente para ayudar en la toma de decisiones y al desarrollo de controladores, sin embargo, en ciertas ocasiones su elevado coste impide su uso. El aprendizaje activo trata de reducir este coste seleccionando aquellos ejemplos que aportan más información. Sin embargo cuando se aplica a los modelos basados en árboles de decisión los resultados no suelen ser competitivos debido a su inherente inestabilidad. En este artículo se propone una técnica que pretende paliar estos problemas consistente en combinar el aprendizaje semisupervisado con el descarte selectivo de aquellos ejemplos que una vez aprendidos empeoran el comportamiento del modelo. La eficacia de esta técnica se ha comprobado con distintas bases de datos UCI y con un problema real: el diseño del sistema de toma de decisiones en un sistema cooperativo para resolver problemas de optimización.

Palabras Clave: Aprendizaje activo, Toma decisiones, Arboles de decisión fuzzy, Sistema coordinado.

1 Introducción

Es frecuente el uso de técnicas de aprendizaje computacional para ayudar en la toma de decisiones o en el desarrollo de controladores. Pero en algunos casos la aplicación de este tipo de técnicas resulta complicada debido al alto coste que supone. Una de las razones por las que puede ser costoso es el etiquetado de los ejemplos que componen la base de datos de aprendizaje, entendiendo como etiquetado la asignación de la clase o el valor a inferir para un determinado ejemplo. Algunos casos concretos en los que se da este problema son aquellos en los que el etiquetado requiere la supervisión humana o aquellos que requieren la ejecución de algún sistema software que consuma muchos recursos.

El aprendizaje activo surge como respuesta a este tipo de problemas. La idea que utiliza es otorgar al modelo de aprendizaje la libertad de elegir qué ejemplo o ejemplos quiere aprender a continuación. Esto contrasta con el paradigma tradicional o pasivo en el que los ejemplos de entrenamiento son aprendidos de forma aleatoria. Empleando aprendizaje activo, se aprovecha el grado creciente de inteligencia adquirida por el modelo para seleccionar en cada momento aquel ejemplo o ejemplos de la base de datos, que pueden proporcionarle un mayor grado de información.

Dentro del aprendizaje activo el uso de árboles de decisión como modelo base no está muy extendido, debido a los problemas de inestabilidad [4] que muestran. En este artículo se propone una técnica de aprendizaje activo que, usando árboles de decisión fuzzy como técnica base, pretende paliar los problemas antes mencionados. Además esta técnica será aplicada al problema concreto del modelado del proceso de toma de decisiones del coordinador de un sistema cooperativo para así probar su efectividad. Para ello el artículo será estructurado de la siguiente forma, en la sección 2 se hará una pequeña introducción al aprendizaje activo, en la sección 3 se presentará la técnica propuesta, comprobándose su eficacia en la sección 4. Posteriormente, en la sección 5, se presentará un problema real, el modelado de la toma de decisiones de un sistema cooperativo de metaheurísticas, probándose la validez de la técnica propuesta en la sección 6. Finalmente en la sección 7 se presentan las conclusiones del trabajo.

2 Aprendizaje activo

En el aprendizaje activo hacemos que sea la propia técnica de aprendizaje la encargada de seleccionar un

número limitado de ejemplos para pedir su etiqueta de modo que maximice su precisión en cada nueva iteración. Una de las primeras técnicas en aparecer [3] consistía en elegir ejemplos de la región de incertidumbre de modo que en cada iteración esta región fuera decreciendo. Sin embargo este enfoque resulta demasiado costoso, por lo que se han propuesto otros enfoques donde la región de incertidumbre no se calcula de forma explícita, como en Uncertainty Sampling [7], que consiste en elegir el ejemplo para el cuál existe una mayor incertidumbre. O Query By Committee [5], que consiste en mantener un comité de clasificadores para inferir la etiqueta de cada ejemplo candidato. Finalmente se elige aquel cuyo grado de desacuerdo entre el comité es mayor.

En el contexto de aprendizaje activo usando árboles de decisión, se han realizado algunos proyectos tales como [9]. Sin embargo el uso de este tipo de modelo no es muy común dentro del paradigma activo. De acuerdo a lo que se comenta en [4], las técnicas de construcción de árboles son inestables en el sentido de que se pueden construir árboles completamente distintos a partir de un conjunto de entrenamiento prácticamente igual. Esto se debe a la naturaleza recursiva del algoritmo de construcción del árbol, que tiene en cuenta todos los ejemplos de entrenamiento en cada momento.

Esta inestabilidad resulta especialmente importante si se usan árboles como modelos para realizar aprendizaje activo. En efecto, esto es así ya que el algoritmo de aprendizaje activo adquiere un ejemplo (o un conjunto limitado de ellos) en cada iteración basándose en el modelo aprendido hasta ese momento. Sabemos que un nuevo ejemplo puede hacer cambios bruscos en los árboles construidos haciendo que varíe su precisión de predicción, ya sea para bien o para mal. Cuando se da este último caso, los modelos se basan en mala información para poder elegir el siguiente ejemplo, con lo cuál se puede iniciar un ciclo que conlleva a que los malos resultados puedan fácilmente superar a los progresos hechos cuando los cambios bruscos de los árboles son positivos. La técnica propuesta en este artículo pretende paliar estos problemas, ayudando en la utilización de árboles de decisión en el aprendizaje activo, además el uso de árboles de decisión fuzzy, debido a su mayor robustez, también ayudará.

3 Combinando aprendizaje semisupervisado y arrepentimiento

3.1 Aprendizaje semisupervisado

La técnica del aprendizaje semisupervisado consiste en añadir a la base de datos de aprendizaje aquellos ejemplos, que aun no estando etiquetados se tiene una gran certeza acerca de la etiqueta que deberían tener, en este caso se utiliza la etiqueta esperada como si realmente se hubiera averiguado la misma.

En un esquema de aprendizaje activo típico, se eligen de entre todos los ejemplos disponibles, n ejemplos para ser evaluados por la técnica de aprendizaje. A continuación, se aplica, para cada uno, un criterio que mide la incertidumbre que se tiene sobre los resultados obtenidos. En función de este criterio, se seleccionará de los n ejemplos aquella con un mayor índice de incertidumbre, x_1.

Una vez que se ha elegido el ejemplo x_1, éste es inferido y seguidamente se dispone de su etiqueta correspondiente y_1, de modo que se añade $< x_1, y_1 >$ al conjunto de entrenamiento. Además se puede aprovechar la información aportada por la etiqueta para compararla con la predicción realizada por la técnica de aprendizaje. De esta manera se puede descubrir si se estaba realizando una predicción acertada o no. En caso de que la predicción fuera acertada, significa que, aun teniendo en cuenta que x_1 era el ejemplo con una mayor incertidumbre, el resultado podría haber sido predicho. Es en este momento donde se aplica la modalidad de aprendizaje semisupervisado, la cuál supone que, de la misma forma que el ejemplo con mayor incertidumbre pudo ser etiquetado correctamente, los ejemplos con menor incertidumbre también podrán ser etiquetados correctamente. De este modo, a estos ejemplos $x_2, x_3, \ldots, x_n$, le son asignados su correspondiente etiqueta en función de los resultados predichos $y'_2, y'_3 \ldots, y'_n$ y así los pares $< x_2, y'_2 >, < x_3, y'_3 > \ldots < x_n, y'_n >$ son añadidos al conjunto de entrenamiento.

3.2 Corrigiendo las decisiones erróneas

Cuando un ente inteligente tiene cierto conocimiento sobre un tema y recibe nueva información sobre el mismo, el primer proceso que realiza es asimilar esta nueva información e intentar integrarla con el conocimiento previo. Si este ente se ve incapaz de encajar la nueva información con la base que tenía previamente, dudará acerca de la veracidad de la última información recibida. La idea clave tras la técnica propuesta se basa en esta idea: comprobar que el nuevo ejemplo no sólo aporta nueva información al sistema sino que refuerza lo que se había adquirido anteriormente. En caso contrario se duda de él y se descarta. A este enfoque le llamaremos aprendizaje con arrepentimiento.

A grandes rasgos, el aprendizaje con arrepentimiento se refiere a la modificación del esquema activo de modo que una vez elegido y etiquetado el nuevo ejemplo, el modelo comprueba si la incorporación de ese

nuevo ejemplo a su base de entrenamiento va a proporcionar una mejor o, por contra, una peor capacidad de predicción ante la inferencia de nuevos ejemplos.

Para dilucidar si el nuevo ejemplo compensa ser aprendido o no, lo ideal sería realizar una validación tomando como conjunto de prueba todos los ejemplos de la base de datos: en primer lugar se realiza un test utilizando el modelo actual y a continuación se realiza la misma prueba pero añadiendo al conjunto de entrenamiento el ejemplo recién etiquetado. Finalmente se comparan los resultados y se añade este último ejemplo en caso favorable. Obviamente esto no se puede hacer debido a que la mayoría de los ejemplos están sin etiquetar. De hecho los ejemplos que están etiquetados son los que ha aprendido el modelo. Por esto se debe hacer una estimación de esta medida. En concreto la implementación de esto ha sido realizar lo explicado anteriormente pero tomando el conjunto de entrenamiento como conjunto de prueba. De esta forma:

$ErrorAnterior = \frac{Error(M_T(T))}{|T|}$

$ErrorActual = \frac{Error(M_{T\cup\{n\}}(T\cup\{n\}))}{|T|+1}$

donde

- T es el conjunto de ejemplos que forman el conjunto de entrenamiento.
- n es el nuevo ejemplo recién etiquetado.
- $Error(M_E(T))$ es el error cometido al inferir los ejemplos del conjunto T a partir de un modelo entrenado con los ejemplos del conjunto E.

Además se introduce el concepto de margen de confianza como una variable de control que permite al algoritmo adaptarse a cualquier posible evolución del proceso de aprendizaje, de modo que nunca se rechazan o aceptan ejemplos en exceso.

De esta forma, una vez que se ha calculado ErrorAnterior y ErrorActual se comprueba si ($ErrorActual > ErrorAnterior * MargenConfianza$) en caso afirmativo se rechaza el nuevo ejemplo x_1 y se incrementa el margen de confianza, en caso nevativo se decrementa el margen de confianza.

Podría pensarse que este enfoque desaprovecha los recursos ya que éstos pueden consumirse para etiquetar un ejemplo que posteriormente no será utilizado. Sin embargo, su uso ha propiciado mejores resultados a los obtenidos anteriormente.

La técnica propuesta consiste en una mezcla del aprendizaje semisupervisado con el aprendizaje con arrepentimiento. En este artículo se probará esta solución sobre un esquema basado en comité, a pesar de ello este enfoque es extrapolable a otros esquemas de aprendizaje activo.

4 Probando la aproximación

En este apartado se probará la eficiencia de la aproximación propuesta, comparándola con el aprendizaje activo tradicional y el aprendizaje pasivo. Las pruebas se realizarán utilizando como modelo de aprendizaje básico un árbol de decisión fuzzy. Como base de datos de pruebas se usarán distintos problemas de aprendizaje obtenidos de la base de datos UCI [1]. Estas base de datos serán: Breast Cancer Wisconsin (BCW), German Credit (Ger), Ionosphere (Iono), Pima Indians Diabetes (Pima) y Image Segmentation (Seg). Todas estas bases de datos tienen en común el alto número de ejemplos que las componen, y por tanto podría ser conveniente aplicarles aprendizaje activo.

En la tabla 1 se muestran los resultados obtenidos de la siguiente manera: en la segunda columna se muestra el número de ejemplos que componen la base de datos, en la siguiente cual es el porcentaje de acierto obtenido por el aprendizaje pasivo después de haber aplicado sobre la base de datos completa usando validación cruzada de 10 conjuntos. En las siguientes se muestra el número de ejemplos que fue necesario utilizar por el aprendizaje activo (act.) y el aprendizaje activo semisupervisado con arrepentimiento (act. mod.) para obtener un porcentaje de acierto igual o mayor.

Tabla 1: Comparativa aprendizajes.

BD	**n. inst.**	**pas.**	**act.**	**act. mod.**
BCW	699	72.3%	218	78
Ger	1000	95.07%	272	77
Iono	351	91.8%	176	91
Pima	768	66.42%	329	78
Seg	2310	81.13%	416	139

Como puede observarse, el uso de cualquier técnica de aprendizaje activo supone una mejora sustancial en el número de ejemplos que se deben usar para alcanzar el porcentaje de acierto que alcanza el aprendizaje activo. De igual manera se puede ver como el uso de la técnica propuesta (act. mod.) también proporciona una mejora con respecto al aprendizaje activo clásico.

5 Aplicación al diseño de un sistema cooperativo inteligente

En esta sección pretendemos mostrar la validez del esquema propuesto para un problema real, como es el modelado de la toma de decisiones del coordinador de un sistema cooperativo de metaheurísticas para resolver problemas de optimización.

5.1 Un sistema híbrido de metaheurísticas

El sistema de metaheurísticas que centrará nuestra atención [2] está basado en un sistema multiagente, en el cual cada agente implementa una metaheurística y coopera con el resto de agentes para resolver una instancia del problema de optimización. Para controlar la cooperación entre los distintos agentes se hace uso de un agente coordinador, que tiene dos tareas principales: recolectar información acerca del rendimiento de cada agente y basándose en esta decidir como se debe modificar el comportamiento de cada agente de tal manera que se mejore el comportamiento global. La figura 1 muestra una imagen conceptual del sistema.

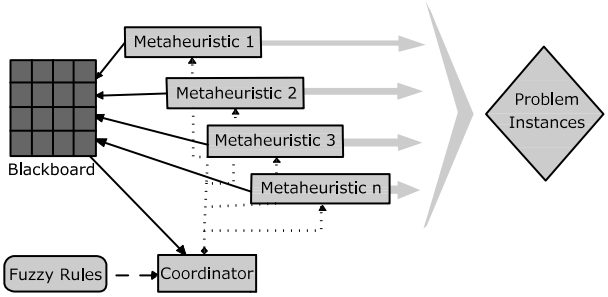

Figura 1: El sistema cooperativo de metaheurísticas.

Para que el coordinador pueda realizar las tareas antes mencionadas se le debe dotar de cierta "inteligencia", para hacer esto se realiza un proceso de aprendizaje supervisado con el objetivo de completar un conjunto de reglas fuzzy que controla el comportamiento del coordinador. Este conjunto de reglas se compone de la siguiente regla replicada una vez por cada metaheurística:

- SI $[(wm_1 * d_1$ O ... O $wm_n * d_n)$ ES *suficiente*] ENTONCES cambiar la solución actual de MH.

donde:

- n el el número de metaheurísticas.
- MH es la metaheurística evaluada por la regla.
- $d_i = (r_i - perf_{MH})/maximum(r_i, r_{MH})$, donde r es una medida de rendimiento, que en nuestro caso será el valor de la función objetivo obtenido por la solución actual de cada metaheurística.
- $wm_i \in [0,1]$ donde $\sum_{i=1}^{n} wm_i = 1$ y wm_i representa el peso de la metaheurística i (importancia de esta para resolver la instancia actual).
- $suficiente$ es un conjunto fuzzy con función de pertenencia trapezoidal con soporte contenido en $[0,1]$ definido por la cuadrupla (a,b,c,d). Su representación matemática es presentada en la ecuación (1):

$$\mu(x,a,b,c,d) = \begin{cases} 0 & x \leq a \ or \ x \geq d \\ (x-a)/(b-a) & x \in (a,b] \\ (d-x)/(d-c) & x \in [c,d) \\ 1 & x \in [b,c] \end{cases} \quad (1)$$

Esta regla cambia la posición en el espacio de búsqueda de la metaheurística que está mostrando un rendimiento malo por una posición cercana a la de otra metaheurística que está mostrando un comportamiento mejor.

Además de la anterior regla, el coordinador también dispone de un conjunto de reglas de inicialización de parámetros que deciden para cada metaheurística que valores de parámetros deben ser tomados dependiendo de la instancia del problema que se vaya a resolver.

Tanto los wm_i que controlan la regla como los valores de los parámetros de cada metaheurística que deberán tomarse son obtenidos del proceso de aprendizaje supervisado antes mencionado.

5.2 Adaptación del sistema

Para obtener los pesos, wm_i y los valores de los parámetros que serán usados por las reglas definidas anteriormente, se hace uso de un conjunto de árboles fuzzy obtenidos de la aplicación de un proceso de aprendizaje supervisado. Los modelos necesarios serían los siguientes: un árbol fuzzy para predecir los pesos de todas las metaheurísticas, y un árbol fuzzy para cada parámetro de cada metaheurística que predice el valor que obtendrá el mejor rendimiento.

5.3 Algunos conflictos

Para obtener estos modelos anteriormente se usaba un proceso de aprendizaje supervisado compuesto de dos fases, preparación de los datos y minería de datos. En la primera fase las diferentes metaheurísticas que componen el sistema son ejecutadas varias veces usando diferentes valores de los parámetros con el propósito de obtener una base de datos que resuma el comportamiento de cada metaheurística. En la segunda fase se aprenden los diferentes árboles fuzzy antes mencionados. Este proceso tiene un importante cuello de botella en la fase de preparación de los datos, puesto que la ejecución de las distintas metaheurísticas tiene un costo muy elevado, por ello es interesante reducir el número de ejemplos que se deben de usar para conseguir los modelos, lo cual se espera conseguir utilizando aprendizaje activo.

5.4 Aplicando aprendizaje activo

Al aplicar aprendizaje activo al sistema nos encontramos con dos situaciones distintas, el aprendizaje de los pesos y el de los valores de los parámetros. Para resolver ambos casos se ha utilizado el esquema antes descrito, pero con ciertas peculiaridades para cada uno.

En el caso de los pesos, al tratarse de una mezcla entre regresión y ranking se debe aplicar un esquema similar al propuesto en [9], en el cual para cada ejemplo de la base de datos se calcula la varianza de los pesos obtenidos por cada miembro del comité, siendo preferida aquella con una mayor varianza.

El caso de los parámetros es un problema de clasificación con la peculiaridad de que sea multiclase, en cuyo caso es interesante usar la aproximación propuesta en [6] que consiste en elegir aquel ejemplo que minimiza la diferencia entre los votos a la clase más popular y la segunda más popular.

6 Pruebas sobre un problema real

En este apartado se realizarán pruebas para comprobar la efectividad de la aproximación propuesta para el problema del modelado del coordinador de un sistema de metaheurísticas.

6.1 El problema de optimización

El sistema de metaheurísticas que se construirá tendrá como objetivo la resolución de instancias del problema de la mochila, que se puede definir formalmente de la siguiente forma:

$$\begin{aligned} Max \quad & \sum_{i=1}^{n} p_i \times x_i \\ s.a \quad & \\ & \sum_{i=1}^{n} w_i \times x_i \ \leq \ C \\ & x_i \in \{0,1\} \ , i = 1, ..., n \end{aligned}$$

donde

- n es el número de items.
- x_i indica si el item i está o no incluido en la mochila.
- p_i es el beneficio asociado al item i.
- $w_i \in [0, \ldots, r]$ es el peso del item i.
- C es la capacidad de la mochila.

Se asume que $w_i < C, \ \forall i$ y $\sum_{i=1}^{n} w_i > C$.

Este problema es NP-completo y ha sido ampliamente estudiado. Para realizar las pruebas se han utilizado distintos tipos de instancias generadas siguiendo el método propuesto en [8] y que han demostrado ser difíciles para los mejores algoritmos exactos. Las instancias generadas pueden tener un tamaño de 500, 1000, 1500 o 2000 objetos y son de los siguientes tipos:

- Spanner: Estas instancias se han construido de forma que sus items son múltiplos de un conjunto pequeño de items llamados key. Esos key fueron generados usando tres distribuciones: Uncorrelated, Weakly correlated, Strongly correlated.
- Profit ceiling: En estas instancias todos los beneficios son multiplos de una parámetro dado d.
- Circle: Estas instancias se generan de forma que los beneficios son una función de los pesos teniendo una representación elíptica.

6.2 El sistema implementado

Para resolver este problema se ha desarrollado un sistema cooperativo compuesto de tres metaheurísticas, un algoritmo genético, una búsqueda tabú y un temple simulado. Además el conjunto fuzzy $suficiente$ se define con los siguientes puntos $[0, 0.1, 1, 1]$ y el $\alpha - corte$ usado es 0.05. Para el aprendizaje previo se usaron 500 instancias de las cuales el aprendizaje pasivo usa todas y la técnica propuesta 20.

6.3 Metodología

Para realizar las pruebas se resolvieron 100 instancias distintas, ejecutándose cada estrategia durante 10 segundos parando cada 250 milisegundos para realizar la coordinación. Cada instancia fue resuelta 10 veces, y los resultados muestran los promedios. Los tests se desarrollaron en AMD Opteron Dual Core con 8 GB RAM pertenecientes al cluster UGRGrid[1].

6.4 Resultados

La tabla 2 muestra los resultados obtenidos por cada estrategia, agrupados por tipo de instancia y tamaño.

Como se ha comentado anteriormente los modelos obtenidos por el aprendizaje activo se construyeron usando 20 instancias, mientras que los obtenidos usando aprendizaje pasivo utilizaron 500 instancias. Como se puede observar los resultados obtenidos son muy similares, no pudiéndose observar diferencias significativas con el test estadístico no paramétrico de wilcoxon.

Estos resultados son los que esperábamos. Las técnicas de aprendizaje activo obtienen modelos similares a los

[1] http://ugrgrid.ugr.es/

Tabla 2: Comparativa aprendizajes.

tipo	tam.	pasivo	semisuperv. arr.
unc span	500	0,01% (0,05%)	0,00% (0,00%)
	1000	0,02% (0,07%)	0,02% (0,09%)
	1500	0,10% (0,22%)	0,08% (0,15%)
	2000	0,08% (0,22%)	0,16% (0,54%)
wea span	500	0,68% (0,71%)	0,69% (0,67%)
	1000	1,22% (1,29%)	1,20% (1,26%)
	1500	1,29% (1,35%)	1,29% (1,33%)
	2000	1,47% (1,44%)	1,49% (1,47%)
str span	500	1,24% (1,35%)	1,27% (1,33%)
	1000	1,48% (2,36%)	1,45% (2,16%)
	1500	1,69% (2,36%)	1,64% (2,31%)
	2000	2,20% (3,07%)	2,20% (3,02%)
pceil	500	0,20% (0,23%)	0,21% (0,26%)
	1000	0,32% (0,45%)	0,31% (0,42%)
	1500	0,52% (0,58%)	0,51% (0,57%)
	2000	0,37% (0,52%)	0,35% (0,49%)
circle	500	3,24% (2,36%)	3,28% (2,35%)
	1000	5,83% (3,46%)	5,62% (3,32%)
	1500	8,62% (5,26%)	8,66% (5,37%)
	2000	11,06% (6,64%)	10,77% (6,42%)

obtenidos por el aprendizaje pasivo, ya que obtienen los valores de error muy similares (a veces mejores), pero se debe notar el tiempo usado para generar los modelos que controlan la ejecución de los sistemas. Como se ha comentado anteriormente para clasificar una instancia se deben ejecutar muchas veces todas las metaheurísticas utilizando diferentes valores para sus parámetros, en este caso se ha ejecutado cada instancia durante 5 segundos con 3 metaheurísticas distintas, usando 8 combinaciones diferentes de parámetros y repitiendo cada ejecución 10 veces, esto significa que para cada instancia aprendida se necesitan 1200 segundos, 20 minutos. Por tanto para 20 instancias se han necesitado 400 minutos, aproximadamente 7 horas, mientras que para 500 instancias ha sido necesario aproximadamente 7 días. Se puede ver claramente que, aunque los resultados sean similares es preferible utilizar el aprendizaje activo debido a la reducción de tiempo que supone.

7 Conclusiones

En este trabajo se presenta una estrategia de aprendizaje activo desarrollada para paliar los problemas que presenta cuando se aplica a árboles de decisión. Esta estrategia se basa en la combinación de dos técnicas, aprendizaje semisupervisado que trata de etiquetar aquellos ejemplos para los que se tiene una menor incertidumbre con la etiqueta que propone el modelo de aprendizaje, y aprendizaje con arrepentimiento que se basa en ignorar aquellos ejemplos que han reportado un empeoramiento en el rendimiento del modelo. Los resultados obtenidos sobre determinadas bases de datos del repositorio UCI arrojan resultados prometedores para esta técnica.

Asimismo, también se ha probado la validez de la aproximación con un problema real, el modelado de la toma de decisiones de un coordinador de un sistema metaheurístico cooperativo. Para este caso se han realizado pruebas comparando los resultados que obtiene el sistema cooperativo cuando se ha entrenado usando aprendizaje pasivo con 500 ejemplos y el esquema propuesto con 20 ejemplos. Los resultados obtenidos muestran que estadísticamente ambos sistemas son equivalentes, sin embargo la reducción en tiempo obtenida (de 7 días a 7 horas) hace que la utilización del esquema propuesto sea preferible a la utilización del aprendizaje pasivo.

Referencias

[1] A. Asuncion, D.J. Newman. UCI Machine Learning Repository [http://www.ics.uci.edu/ m̃learn/MLRepository.html] 2007.

[2] J.M. Cadenas, M.C. Garrido, E. Muñoz. Using machine learning in a cooperative hybrid parallel strategy of metaheuristics. *Information Sciences*. 179: 3255–3267. 2009.

[3] D. Cohn, L. Atlas, R. Ladner. Improving Generalization with Active Learning. *Machine Learning* 15(2): 201–221. 1994.

[4] K. Dwyer, R. Holte. *Decision Tree Instability and Active Learning.* 18th European Conference on Machine Learning, 4701:128-139, año 2007.

[5] Y. Freund, H. S. Seung, E. Shamir, N. Tishby. Selective Sampling Using the Query by Committee Algorithm. *Machine Learning*, 28, 133–168 1997.

[6] J. Huang, S. Ertekin, Y. Song, H. Zha, C. L. Giles. Efficient Multiclass Boosting Classification with Active Learning. *Proceedings of The 7th SIAM Conference*, 297–308. 2007.

[7] D. D. Lewis, W. A. Gale.A sequential algorithm for training text classifiers. *Proceedings of the 17th international ACM SIGIR conference*. 3–12. 1994.

[8] D. Pisinger, Where are the hard knapsack problems?. *Computer and Operations Research*, 32(9):2271–2284, 2005.

[9] M. Saar-Tsechansky, F. Provost. Active Sampling for Class Probability Estimation. *Machine Learning* 54(2) 153–178. 2004.