Conclusiones

En esta memoria hemos presentado el desarrollo en varias etapas de un algoritmo inductivo de aprendizaje basado en la metodología iterativa para la obtención de reglas difusas, llamado SLAVE. SLAVE usa algoritmos genéticos para determinar en cada iteración del proceso de aprendizaje la mejor regla que representa cada clase teniendo en cuenta la información existente en el conjunto de ejemplos. Una característica fundamental de SLAVE es que ha sido concebido para abordar problemas de clasificación tanto en sutuaciones en las que la variable de clasificación toma valores sobre un refencial crisp, como para aquellas en la que la variable de clasificación toma valores sobre un referencial continuo. Para este segundo tipo de problemas, la aplicación de SLAVE es directa, sin más que discretizar el referencial de la variable de clasificación usando conjuntos difusos.

Una de las características fundamentales de SLAVE es que obtiene la relevancia parcial de las variables de entrada, es decir, durante el proceso de aprendizaje el algoritmo selecciona las características relevantes para construir la descripción de la regla, no considerando para la descripción de la misma, las variables irrelevantes.

En la figura 5.14 se muestra la evolución del algoritmo SLAVE a lo largo de esta memoria. Así, en el capítulo 2 se expuso la metodología básica del algoritmo de aprendizaje. Los resultados obtenidos experimentalmente demuestran que SLAVE tiene un comportamiento para problemas con consecuente crisp, semejante a los resultados que proporcionan otros algoritmos de aprendizaje, aunque únicamente obtiene el mejor resultado para dos de las bases de datos propuestas. El comportamiento del algoritmo mejora con la inclusión de nuevos operadores genéticos definidos para el aprendizaje. Sin embargo, para problemas con consecuente difuso, los resultados de SLAVE no son tan buenos como cabría esperar. Esta deficiencia se debe fundamentalmente a que este segundo tipo de problemas, requieren la colaboración de las reglas extraídas para determinar la clasificación de los ejemplos. Este requisito no se contempla en esta primera versión de SLAVE, ya que las mejores descripciones para cada concepto se aprenden de forma aislada, sin tener en cuenta durante el proceso de aprendizaje, las reglas que se aprendieron para el resto de los conceptos.

Por esta razón, en el capítulo 3 expusimos dos posibles vías para mejorar el comportamiento del algoritmo mediante la inclusión de mecanismos que permitan la colaboración en unos casos, y la competición en otros, de las reglas aprendidas. La primera de las vías sugeridas consiste en la inclusión de un algoritmo de refinamiento de teorías como fase

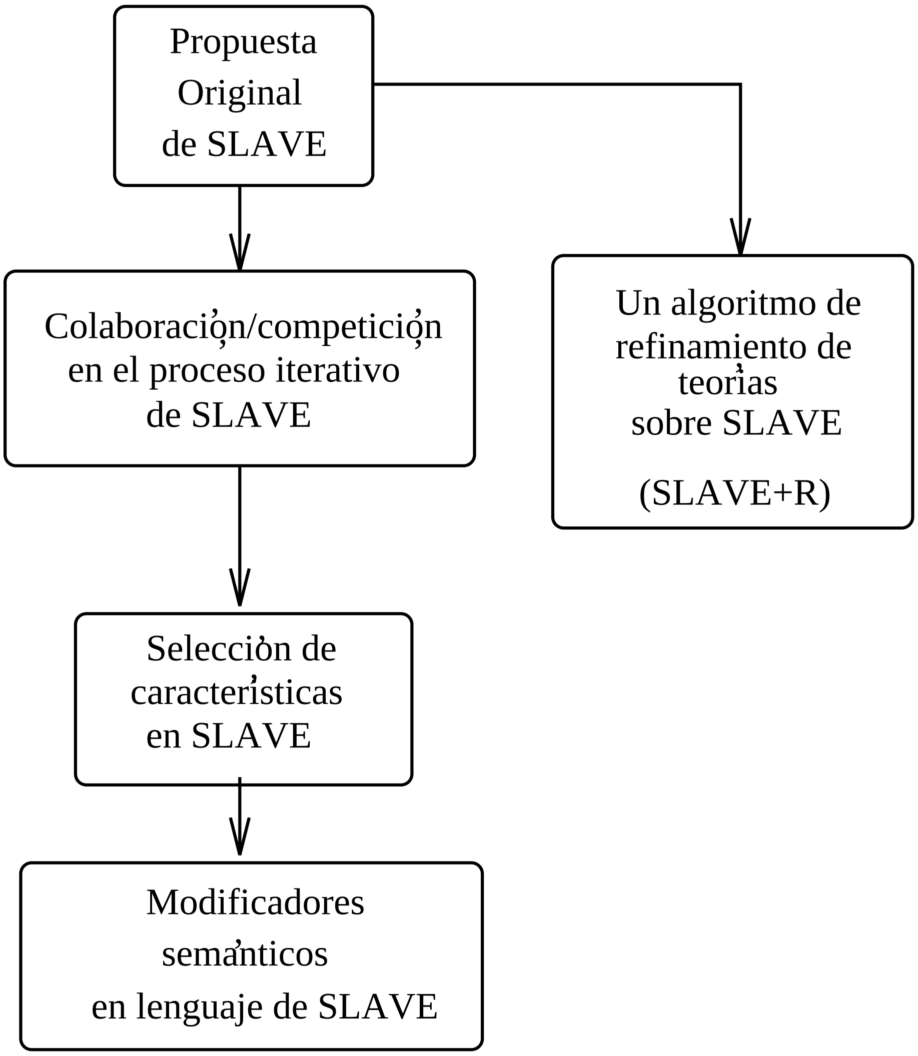

Figura 5.14: Evolución del algoritmo de aprendizaje SLAVE

final del algoritmo de aprendizaje. Este algoritmo de refinamiento será el encargado de establecer de forma conveniente la colaboración o competición entre las reglas, teniendo como objetivo no sólo mejorar la capacidad de predicción de las mismas, sino también mejorar en la medida de lo posible la comprensibilidad del conocimiento extraído.

Como segunda vía, se contempla la posibilidad de que sea el propio algoritmo el que incorpore la colaboración/competición entre las reglas durante el proceso de aprendizaje. Con tal fin, se propuso un nuevo mecanismo de evaluación de reglas, que contempla el conocimiento previamente extraído para los conceptos ya aprendidos, para determinar en que medida las nuevas reglas que se proponen afectarán a la clasificación de los conceptos.

Los resultados experimentales obtenidos en este capítulo ponen de manifiesto que ambas vías son válidas para resolver el problema anteriormente mencionado. Aunque estas dos vías presentan una capacidad de clasificación y un número de reglas parecidos, la segunda mejora sustancialmente el tiempo de respuesta del algoritmo de aprendizaje. Por este motivo, consideramos que la evolución de SLAVE debe seguir esta segunda vía, si se desea obtener un algoritmo que ofrezca un mejor tiempo de respuesta. En cualquier caso, el algoritmo de refinamiento de teorías, aquí propuesto sigue siendo una alternativa válida, no sólo para mejorar las bases de reglas obtenidas por SLAVE, sino para cualquier base de reglas (con un modelo de regla semenjante al aquí propuesto) que proporcione otro algoritmo de aprendizaje o un experto.

Además, de los resultados que se muestran en este capítulo, podemos deducir las siguientes conclusiones:

- La nueva versión de SLAVE mejora sustancialmente su comportamiento para los problemas con consecuente difuso, de tal manera, que mejora para algunas bases de datos los resultados obtenidos por el algoritmo de aprendizaje WM, un algoritmo que muestra tener un buen comportamiento para este tipo de problemas.
- Por el contrario, el algoritmo tiene problemas para seleccionar de forma correcta el conjunto de variables relevantes, cuando el conjunto de ejemplos tiene un número elevado de ellas. Por esta razón, la capacidad de predicción es menor sobre el conjunto de prueba y es elevado el número de reglas que presenta el conocimiento obtenido para la base de datos IONOSPHERE.

Esta última conclusión, nos llevó a proponer en el capítulo 4, una nueva modificación de SLAVE para aumentar su capacidad de detección de variables irrelevantes. Con este propósito, en el módulo de selección de reglas se establecen dos procesos de búsqueda diferentes, uno que trata de encontrar el subconjunto de variables relevantes que formarán parte de la descripción de la regla, y otro que intenta encontrar la mejor asignación de valores a estas variables. Estas dos búsquedas se llevan a cabo de forma simultánea mediante la ejecución de forma paralela de dos algoritmos genéticos. Estos algoritmos genéticos, mediante coevolución, tratan de encontrar la mejor descripción para un concepto. Así, se consigue no sólo agilizar la selección, sino que además, se aumenta el nivel de exploración del espacio de los subconjuntos de variables relevantes.

Se propusieron en este capítulo tres formas posibles para llevar a cabo esta modificación. En la primera de ellas se propone un algoritmo genético para la selección de características que representa para cada variable de entrada, su presencia o ausencia en la descripción de la regla para cada individuo de la población, dejando que la evolución del sistema determine la configuración más adecuada para el concepto. Las otras dos propuestas, para cada variable en el algoritmo genético de selección de características, establecen una medida de información que es interpretada como la probabilidad de que dicha variable pertenezca a la descripción de la regla. Esta medida de probabilidad se obtiene a partir de la información que aporta la distribución de los ejemplos sobre cada uno de los conceptos que se desean aprender. Por consiguiente, el uso de esta información en el algoritmo genético, permite acercar las descripciones que se generan inicialmente a las descripciones que se desean obtener. La diferencia entre estas dos últimas propuestas está en la interpretación que se hace de las medidas de probabilidad: en la primera se utiliza directamente la probabilidad asociada para determinar su pertenencia o no a la descripción de la regla que define cada individuo (modelo por copia). En la segunda, se establece un valor de probabilidad diferente para cada individuo que se interpreta como un umbral de activación, de manera que toda variable que tenga un valor de probabilidad superior al umbral se considera en la descripción de la regla (modelo por umbral de activación). En ambos casos, la población inicial contiene las medidas de información obtenidas del conjunto de ejemplos, y será el proceso evolutivo el que, alterando mediante operadores genéticos dichas medidas, obtenga la configuración más adecuada para la descripción de la regla.

Los resultados experimentales reflejaron, que si bien todas ellas conseguían mantener o mejorar los resultados de las versiones anteriores, el modelo por umbral de activación presentaba los mejores resultados, disminuyendo significativamente el tiempo de respuesta del algoritmo. Por consiguiente, el algoritmo SLAVE dos niveles siguiendo el modelo por umbral de activación demuestra ser el algoritmo más apropiado, tanto para problemas con consecuente crisp como con consecuente difuso.

Aunque SLAVE dos niveles siguiendo el modelo por umbral de activación es un algoritmo de aprendizaje que produce muy buenos resultados, en el capítulo 5 nos planteamos la posibilidad de mejorar aún más los resultados obtenidos previamente. Para mejorar estos resultados era necesario dotar al algoritmo de la capacidad de alterar la semántica de sus etiquetas iniciales, de manera que éstas respondieran lo mejor posible a la información que contienen los ejemplos. Con esta idea, se propuso enriquecer el lenguaje de representación de SLAVE mediante la inclusión de modificadores semánticos. Las pruebas realizadas sobre este nuevo algoritmo ponen de manifiesto que si el conjunto de entrenamiento contiene un número suficiente de ejemplos que representen ampliamente el comportamiento del sistema, los resultados son muy buenos. En este sentido, cabe destacar que para el caso difuso, se mejora para todas las bases de datos los resultados obtenidos por el algoritmo WM. Por el contrario, se observa un sobreaprendizaje si la base de datos contiene pocos ejemplos.

Como conclusión final, resaltamos que se han conseguido los objetivos propuestos, proporcionando un algoritmo de aprendizaje que presenta un buen comportamiento en los dos tipos de problemas de clasificación planteados, dotando de un alto grado de comprensibilidad desde el punto de vista humano al conocimiento extraído.

Las líneas futuras de trabajo de esta memoria se pueden resumir en:

1. Profundizar en el problema planteado en el capítulo5 para tratar de reducir la tendencia al sobre-aprendizaje que provoca la inclusión de los modificadores semánticos en la descripción de las reglas.

2. La obtención de un algoritmo más eficiente sobre el modelo presentado en la memoria considerando fundamentalmente dos alternativas: la primera consiste en dotar

al algoritmo de la capacidad de autoadaptación de los parámetros de entrada del mismo. En las últimas propuestas de SLAVE se requieren muchos parámetros de entrada de los cuales depende la bondad de los resultados obtenidos. La idea consiste en introducir algoritmos genéticos autoadaptativos que permitan, durante el proceso de búsqueda, encontrar estimaciones correctas a estos parámetros que mejoren el tiempo de respuesta de SLAVE. La segunda de las alternativas, es sustituir los algoritmos genéticos del algoritmo de aprendizaje, por otros algoritmos de búsqueda que permitan obtener más rápidamente las soluciones deseables.

3. Dotar a SLAVE de la capacidad de readaptar su conocimiento ante la entrada de nuevos ejemplos, es decir, convertir a SLAVE en un algoritmo de aprendizaje incremental.

4. Aumentar la capacidad de descripción de SLAVE, incluyendo la posibilidad de determinar relaciones entre las variables de entrada para generar nuevos atributos. En este sentido, SLAVE se convertiría en un algoritmo que aprende relaciones entre los atributos iniciales.