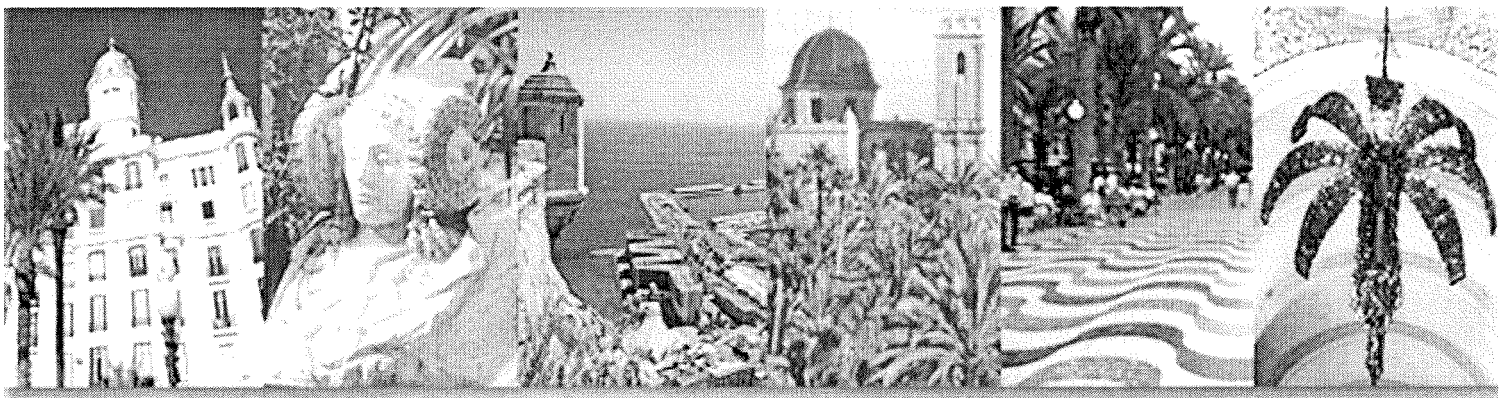

XXVI Jornadas de Automática

Universitat d'Alacant
Universidad de Alicante

UNIVERSITAS Miguel Hernández

ISBN: 84-689-0730-8
Editores: Fernando Torres, Oscar Reinoso

INTEGRACIÓN DE MATLAB Y WEKA PARA LA DOCENCIA EN ASIGNATURAS DE APRENDIZAJE AUTOMÁTICO (MACHINE LEARNING)

C. Fernández, M. A. Vicente, A. Gil, L. M. Jiménez, R. Ñeco
Universidad Miguel Hernández. Av. Universidad s/n. 03202 Elche (Alicante)
c.fernandez@umh.es

Resumen

Se presenta un interfaz entre las herramientas Matlab y Weka que puede ser de gran utilidad para la docencia en asignaturas de aprendizaje automático. El objetivo es utilizar los algoritmos del paquete Weka desde un entorno más familiar para el alumno, como es el software Matlab. El trabajo muestra cómo se puede conseguir este objetivo de una forma simple gracias a la posibilidad de ejecución de Weka desde la línea de comandos, invocando directamente la clase java correspondiente a cada algoritmo. Se explica el procedimiento a seguir para la interconexión de los dos paquetes software y se propone un programa práctico basado en este interfaz. Tal programa práctico se utiliza actualmente en una asignatura de Ingeniería de Telecomunicación de la Universidad Miguel Hernández de Elche.

Palabras clave: Docencia, Inteligencia Artificial, Aprendizaje Automático, Matlab, Weka.

1. INTRODUCCIÓN

Existen múltiples asignaturas en las que, en parte de su temario, se estudian técnicas de aprendizaje automático, generalmente aprendizaje inductivo. Tales asignaturas, en función de las Universidades, pueden denominarse Reconocimiento de Formas, Reconocimiento de Patrones, Inteligencia Artificial, Aprendizaje Automático, *Machine Learning*, *Soft Computing*, etc. La parte práctica de estas asignaturas requiere necesariamente el uso de un software específico, dado que el pretender que los alumnos desarrollen los algoritmos partiendo desde cero resultaría inviable.

Caben dos posibilidades: o bien se desarrolla un software propio, como en la Universidad de Granada [1] o bien se utiliza software externo, normalmente gratuito. Este software externo puede ser genérico (múltiples algoritmos en el mismo paquete, como es el caso de Weka [2][3] o XLMiner [4]) o bien específico para cada algoritmo a utilizar: C45 [5][6] para árboles de decisión; FOIL [7][6] para generar listas de reglas, etc.

La opción más comúnmente utilizada es la del software externo y genérico, dado que la mayor parte de las Universidades no están en disposición de desarrollar un software propio (que debe ser actualizado con frecuencia) y además el utilizar un software externo genérico permite trabajar con un mismo formato para los ficheros de datos, con independencia del algoritmo a utilizar. Este hecho simplifica las prácticas y permite llevar a cabo programas prácticos más extensos, en los que se estudia un mayor número de algoritmos. Entre los paquetes software genéricos, el de mayor aceptación entre las Universidades ([8][9][10][11][12]) es el software gratuito Weka.

2. CARACTERÍSTICAS BÁSICAS DE WEKA

Las iniciales WEKA responden a *Waikato Environment for Knowledge Analysis* [2][3]. Se trata de una herramienta de libre distribución desarrollada en la Universidad de Waikato (Nueva Zelanda), escrita en lenguaje java y que permite realizar multitud de análisis. La herramienta está aplicada a procesos de minería de datos, por lo que agrupa diferentes técnicas: preprocesado, agrupamiento o *clustering*, ajuste de clasificadores, generación de reglas de asociación, etc. También incluye facilidades para la visualización de los datos.

Existen múltiples modos de utilización de Weka. En primer lugar, la herramienta dispone de cuatro interfaces distintos:

1. Interfaz en modo texto: permite la introducción de todo tipo de comandos, pero no es posible realizar representaciones gráficas (realmente, el interfaz en modo texto permite instanciar las distintas clases java definidas en el programa Weka).
2. Interfaz *Explorer*: es el interfaz gráfico básico. En él se pueden mostrar gráficamente tanto las características de los datos de partida como los resultados de los análisis. Permite introducir los comandos con ayuda del ratón, seleccionando los operadores adecuados en menús desplegables.

3. Interfaz *Experimenter*: se trata de un interfaz gráfico más avanzado, en el que no sólo se pueden realizar análisis sobre los datos, sino que además es posible comparar el funcionamiento de diferentes algoritmos (por ejemplo, diferentes clasificadores) o bien comparar distintos ficheros de datos.

4. Interfaz *KnowledgeFlow*: este último interfaz permite representar como una red de operadores en cascada los procesos a realizar sobre los datos (preprocesado, selección de características, ajuste de un clasificador, evaluación de los porcentajes de acierto esperables, etc.)

La figura 1 muestra el aspecto de la pantalla inicial de Weka, desde la que es posible acceder a cada uno de los cuatro interfaces; la figura 2 muestra tales interfaces.

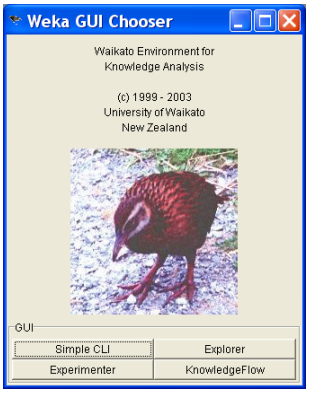

Figura 1: Pantalla inicial de Weka

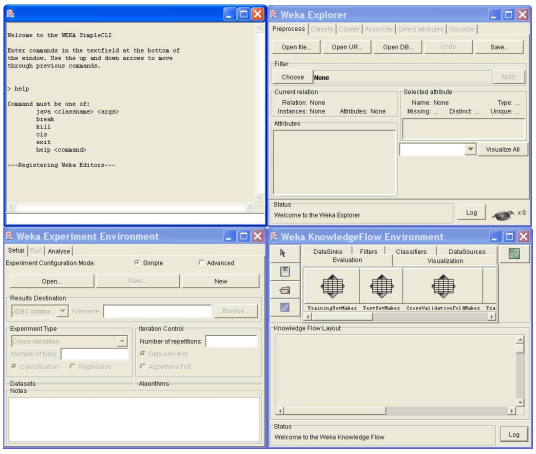

Figura 2: Interfaces disponibles en Weka

Además de estos cuatro interfaces, también caben otras dos alternativas: en primer lugar, es posible introducir el código java de Weka (que es accesible libremente) en una aplicación java propia, de modo que se puede crear un programa específico para cada asignatura, que haga uso de las utilidades de Weka que se desee. En segundo lugar y por último, también es posible instanciar las clases java desde la línea de comandos de MS-DOS, mediante la llamada al intérprete java de que disponga el sistema (sea este Windows, Linux o Macintosh). Esta alternativa será la que se utilice en el presente trabajo, donde se propone la integración de las herramientas Matlab y Weka para la realización de prácticas.

3. INTERFAZ DESARROLLADO ENTRE LOS PROGRAMAS WEKA Y MATLAB

En multitud de Universidades es común el uso del software Matlab para la realización de prácticas sobre equipos informáticos en muy diversas asignaturas, de primer y segundo ciclo. La experiencia de uso de Matlab en diferentes asignaturas a lo largo de la carrera hace que los alumnos adquieran grandes conocimientos de tal herramienta, y que sean capaces de utilizarla con gran soltura para la realización de análisis y preparación de gráficos con los que desarrollar informes de todo tipo.

En este trabajo se propone aprovechar esta capacidad adquirida a lo largo de la carrera para la realización de prácticas de aprendizaje automático con el software Weka. Para ello, se utiliza la capacidad que tiene Weka para ser ejecutado desde la línea de comandos y la capacidad que tiene Matlab (como casi cualquier lenguaje de programación) para realizar una llamada al sistema operativo y ejecutar cualquier programa externo. Desde un punto de vista funcional, se utiliza Weka como un servidor al que la aplicación cliente (Matlab) puede solicitar que realice procesamientos con los datos.

Para una utilización eficaz del programa Weka desde el entorno Matlab, se han desarrollado cuatro módulos diferentes, que se detallan a continuación.

3.1. MÓDULO DE PREPARACIÓN DE DATOS

Este módulo tiene por objetivo generar automáticamente ficheros de datos en el formato adecuado para Weka (formato ARFF o *Attribute-Relation File Format*) a partir de variables Matlab (matrices de datos). El formato ARFF es relativamente sencillo, con lo cual la generación automática de ficheros de datos no supone un problema. En la actualidad, el módulo de preparación de datos se proporciona a los alumnos como un conjunto de funciones Matlab que no tienen que programar, sino sólo comprender y utilizar. Partiendo de una

matriz de datos, las funciones desarrolladas permiten crear un fichero Weka con la totalidad de los datos o bien repartir aleatoriamente los datos en ficheros de entrenamiento y verificación (mediante selección estratificada, de modo que las diferentes clases queden representadas por igual en ambos ficheros).

3.2. MÓDULO DE INSTANCIACIÓN DE CLASES WEKA

El módulo de instanciación de clases Weka se utiliza para realizar llamadas desde Matlab al sistema operativo; llamadas en las cuales se invoca al interprete java para instanciar algunas de las clases de Weka. En la instanciación de estas clases, se realiza el intercambio de información mediante ficheros de entrada y salida. Los ficheros de entrada son los generados mediante el módulo de preparación de datos; y los ficheros de salida devueltos por Weka se almacenan para su posterior utilización. Asímismo, el módulo de instanciación de clases permite introducir todos los parámetros que afectan a cada algoritmo Weka (por ejemplo, umbrales de confianza para la poda en árboles de decisión, etc.)

3.3. MÓDULO DE LECTURA DE RESULTADOS

De acuerdo con lo dicho anteriormente, el intercambio de información se realiza mediante ficheros de entrada y salida. En cuanto a los ficheros de salida, caben varias posibilidades:

- En primer lugar, el fichero de salida puede ser un fichero de datos modificados, en formato ARFF. Éste es el caso cuando se realiza un preprocesamiento de los datos de partida, o un proceso de selección o extracción de características. En estas circunstancias no es necesaria ninguna operación adicional para leer los resultados.
- En segundo lugar, es posible que el fichero de salida sea un modelo ajustado a los ejemplos de entrenamiento. Esta situación se produce cuando se utiliza uno de los clasificadores incorporados en el paquete Weka: el fichero de entrada contiene los ejemplos de entrenamiento y el fichero de salida contiene un modelo que es posible aplicar a nuevos ejemplos para obtener su clasificación. En esta situación tampoco es necesario ningún proceso de lectura de resultados.
- Por último, en algunos casos el fichero de salida muestra un conjunto de resultados en modo texto; y en estas circunstancias son necesarias funciones Matlab que permitan extraer los datos relevantes. Se proporciona a los alumnos un módulo de lectura de resultados básico, que permite extraer los datos más comunes (básicamente porcentajes de acierto en clasificación) y se propone al alumno completar este módulo con funciones que permitan acceder a datos más específicos (por ejemplo, relacionados con algoritmos concretos, como el número de nodos de un árbol de decisión).

3.4. MÓDULO GRÁFICO

Una de las razones por las que se realiza el interfaz entre Matlab y Weka es la posibilidad de realizar representaciones gráficas de los resultados de los análisis de un modo flexible y cómodo para los alumnos. El módulo gráfico contiene todas las funciones Matlab que permiten crear tales representaciones gráficas. Dado que los alumnos conocen el lenguaje, este módulo no se proporciona sino que es responsabilidad de los alumnos el desarrollo de las funciones adecuadas.

4. EJEMPLO DE PROGRAMA PRÁCTICO UTILIZANDO EL INTERFAZ

A continuación se muestra un ejemplo de programa práctico utilizando el interfaz desarrollado. Este programa se utiliza actualmente para cubrir la mitad de la asignatura Inteligencia Artificial y Reconocimiento de Patrones [13] que se imparte en quinto curso de Ingeniería de Telecomunicación en la Universidad Miguel Hernández de Elche (la restante mitad de la asignatura se centra fundamentalmente en la extracción de características a partir de imágenes).

El programa práctico se estructura en cinco sesiones, que son detalladas en los apartados siguientes. Existen dos sesiones adicionales, que no se detallan en este trabajo, y que se dedican a utilizar los conocimientos adquiridos sobre Weka en una aplicación de reconocimiento de caras a partir de imágenes capturadas por una cámara de video. Más información sobre estas últimas prácticas se puede encontrar en la página web de la asignatura [13] y en el trabajo que será próximamente presentado en [14].

4.1. Introducción a Weka

En esta primera sesión práctica se introduce el programa Weka, trabajando sobre el interfaz más sencillo e intuitivo, el interfaz *Explorer*. Los alumnos aprenden fundamentalmente tres aspectos del entorno Weka:

- En primer lugar, se aprende el formato ARFF para los ficheros de datos con el que se trabajará durante el resto de las prácticas. Los alumnos parten de un ejemplo y posteriormente deben generar un fichero de datos por ellos mismos. Cabe la posibilidad de generarlo manualmente o automáticamente haciendo uso de funciones Matlab.

- En segundo lugar, los alumnos pueden comprobar las capacidades gráficas del interfaz *Explorer*. Estas capacidades gráficas son las típicas de un entorno de minería de datos o *data mining*.

- Por último, se muestra la forma de utilizar clasificadores en Weka. En concreto, se trabaja sobre la generación de un árbol de decisión a partir de un fichero de datos.

A pesar de ser una práctica realizada sobre un interfaz gráfico, los alumnos también pueden apreciar las salidas que el programa Weka ofrece en modo texto, que serán las utilizadas en prácticas posteriores. Esta sesión introductoria se considera necesaria para facilitar la comprensión de las siguientes sesiones.

4.2. Utilización de Weka desde la línea de comandos

En esta segunda sesión práctica se muestra cómo los mismos resultados obtenidos desde el interfaz *Explorer* se pueden lograr lanzando el intérprete java desde la línea de comandos e invocando la clase adecuada del paquete Weka. Como primer paso, se ejecuta una función Weka desde MS-DOS; y posteriormente se realiza la misma operación desde Matlab, mediante una llamada al sistema. Estas llamadas al sistema para ejecutar Weka constituyen el módulo de instanciación al que se hace referencia en el apartado 3.2.

Para mostrar la potencialidad de este modo de operación, se proponen ejemplos en los cuales se realizan múltiples llamadas a Weka desde un bucle Matlab; tal operación no resultaría posible desde los entornos que proporciona Weka.

A continuación, se muestra la forma de leer los resultados ofrecidos por Weka y convertirlos en variables Matlab. Dada la estructura de los ficheros de salida de Weka, esta lectura se puede realizar mediante funciones Matlab de gran simplicidad, que constituyen el módulo al que hace referencia el apartado 3.3. Como se ha comentado anteriormente, es responsabilidad del alumno ampliar el módulo de lectura para permitir un mejor intercambio de resultados entre Weka y Matlab.

Como trabajo final de la práctica, el alumno debe realizar llamadas a Weka dentro de un bucle, variando los parámetros del clasificador empleado, y mostrar los gráficos resultantes sobre Matlab. En concreto, se utiliza un árbol de decisión y se debe mostrar tanto el porcentaje de aciertos en clasificación como el tamaño del árbol en función del umbral de confianza para la poda. La figura 3 muestra uno de los gráficos que deben obtener los alumnos.

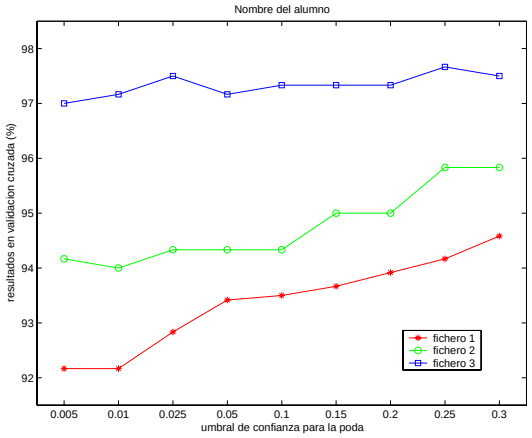

Figura 3: Resultados de clasificación sobre tres ficheros de datos distintos.

4.3. Almacenamiento y posterior utilización de modelos

La tercera sesión práctica aborda el almacenamiento de modelos con Weka. En las prácticas anteriores se ajustan clasificadores a partir de datos de ejemplo y se evalúa el porcentaje de aciertos en clasificación esperable mediante un experimento de validación cruzada realizado internamente en Weka. En esta sesión se almacena el modelo obtenido (por ahora se continúa con el ejemplo del árbol de decisión) para su utilización posterior. Esto requiere el manejo de ficheros distintos a los utilizados en las prácticas anteriores.

Una vez terminada esta sesión, los alumnos son capaces de ajustar un clasificador a partir de un conjunto de datos de entrenamiento y utilizar posteriormente ese clasificador sobre un conjunto distinto de ejemplos; todo ello mediante llamadas desde el entorno Matlab.

4.4. Comparación de métodos de aprendizaje con Weka y Matlab

La cuarta sesión práctica aprovecha los conocimientos adquiridos sobre el interfaz para la realización de un comparación básica de métodos de aprendizaje inductivo. Se comparan los siguientes métodos:

- Árbol de decisión.
- Vecino más cercano.
- Lista de reglas.
- Naive Bayes.

Para realizar la comparación, se parte de un fichero de datos de entrenamiento (en concreto, un fichero con datos de una aplicación de agarre robótico basado en aprendizaje) y se evalúa el porcentaje de aciertos en clasificación esperable para cada algoritmo, mediante un experimento de validación cruzada. El resultado que obtienen los alumnos es similar al que se muestra en la figura 4, donde cada algoritmo se parametriza de forma distinta (5 ejecuciones del árbol de decisión con distintos umbrales de confianza para la poda, 5 ejecuciones del vecino más cercano con distinto número de vecinos, 2 ejecuciones de Naive Bayes con distinto modo de ajuste de las funciones de densidad de probabilidad y una única ejecución para la lista de reglas).

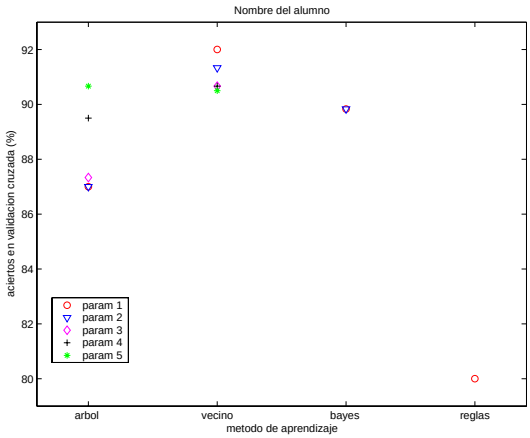

Figura 4: Comparación de algoritmos mediante validación cruzada.

4.5. Aspectos avanzados en la comparación de métodos de aprendizaje

En la quinta sesión práctica se realiza una comparación de métodos de aprendizaje más avanzada, en la que se estudia el efecto de la variación del número de ejemplos de entrenamiento sobre el número de clasificaciones correctas previsible para cada método de aprendizaje.

Se comparan todos los métodos de la práctica anterior y se añade el perceptrón multicapa. Este último método también se parametriza, de modo que se realizan experimentos con distinto número de neuronas en la capa oculta. Con este experimento, los alumnos pueden comprobar de un modo práctico otros aspectos relevantes como el tiempo de cómputo requerido por los distintos métodos (el ajuste de los pesos de un perceptrón multicapa resulta mucho más lento que la creación de modelos con el resto de métodos).

Se muestran dos ejemplos de los gráficos que los alumnos obtienen en esta práctica. En primer lugar, el gráfico que compara el funcionamiento de tres algoritmos distintos (árbol de decisión, vecino más cercano y Naive Bayes) en función del número de ejemplos de entrenamiento; tal gráfico se muestra en la figura 5. En segundo lugar, el que representa el comportamiento del perceptrón multicapa en función del número de neuronas en la capa oculta y del número de ejemplos de entrenamiento. Tal gráfico se muestra en la figura 6.

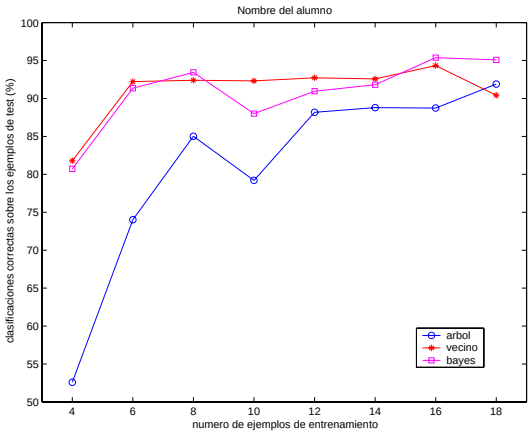

Figura 5: Comparación de algoritmos en función del número de ejemplos de entrenamiento.

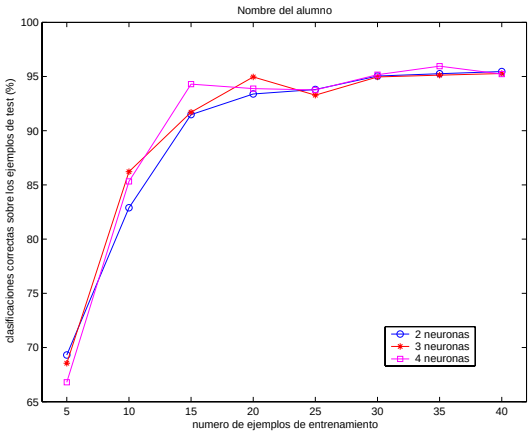

Figura 6: Comportamiento del perceptrón multicapa en función del número de neuronas de la capa oculta.

Los guiones de las prácticas, junto con los ficheros de datos y los programas Matlab necesarios para

cada una de ellas, están disponibles en la página web de la asignatura [13]. Se corresponden con las prácticas 5 a 9 del temario.

5. CONCLUSIONES

Las asignaturas de aprendizaje automático o *Machine Learning* requieren la realización de numerosas sesiones prácticas para un buen aprovechamiento. Por tanto, la elección del software con que se realizan estas prácticas es determinante para el éxito de la asignatura.

Dado que el software a emplear es muy específico, resulta siempre desconocido para los alumnos, que por tanto no son capaces de explotar las posibilidades del mismo.

En este trabajo se propone la utilización del paquete Weka a través de un interfaz con Matlab, de modo que se aprovecha la experiencia de los alumnos con este software, fundamentalmente en cuanto a las capacidades gráficas, pero también en cuanto a las capacidades de ejecución de bucles, trabajo con ficheros, etc.

El ejemplo de programa práctico presentado muestra cómo el interfaz entre Weka y Matlab permite llevar a cabo multitud de prácticas diferentes de un modo más intuitivo para los alumnos.

Referencias

[1] Software de Aprendizaje Automático de la Universidad de Granada
www-etsi2.ugr.es/depar/ccia/rf/software.html

[2] Witten, I. H. and Frank, E., (2000) *Data Mining.* Morgan Kaufmann Publishers.

[3] Programa Weka
www.cs.waikato.ac.nz/ml/weka

[4] Programa XLMiner
www.resample.com/xlminer/index.shtml

[5] Quinlan, J. R. (1993) *C4.5: Programs for machine learning.* Morgan Kaufmann.

[6] Programas C4.5 y FOIL
www.rulequest.com/Personal

[7] Quinlan, J.R., (1990) *Learning logical definitions from Relations.* Machine Learning, 5, pp. 239-266.

[8] Universidad Carlos III de Madrid
www.uc3m.es

[9] Universidad del País Vasco
www.ehu.es

[10] Universidad Politécnica de Madrid
www.upm.es

[11] Universidad de La Rioja
www.unirioja.es

[12] Universidad Miguel Hernández
www.umh.es

[13] Asignatura Inteligencia Artificial y Reconocimiento de Patrones
lorca.umh.es/isa/es/asignaturas/iarp

[14] Fernández, C., Vicente, M. A., Puerto, R., Coves, A. (2005) *Programa mixto presencial/no presencial de prácticas de visión por computador.* IV Jornadas de Enseñanza a través de Internet/Web de la Ingeniería de Sistemas y Automática. Pendiente de publicación.