Proyecto KEEL: Desarrollo de una Herramienta para el Análisis e Implementación de Algoritmos de Extracción de Conocimiento Evolutivos*

Jesús Alcalá-Fdez[1], María José del Jesus[2], Josep M. Garrell[3], Francisco Herrera[1], Cesar Hervás[4], Luciano Sánchez[5]

[1] Universidad de Granada, Departamento de Ciencias de la Computación e Inteligencia Artificial, Escuela Técnica Superior de Ingeniería Informática, 18071, Granada, España
jalcala@decsai.ugr.es, herrera@decsai.ugr.es

[2] Universidad de Jaén, Departamento de Informática, Escuela Politécnica Superior, 23071, Jaén, España
mjjesus@ujaen.es

[3] Universitat Ramon Llull, Enginyeria i Arquitectura La Salle, 08022, Pg. Bonanova, Barcelona, España
josepmg@salleurl.edu

[4] Universidad de Córdoba, Departamento de Informática y Análisis Numérico, Campus Universitario de Rabanales, 14071, Córdoba, España
chervas@uco.es

[5] Universidad de Oviedo, Departamento de Informática, Campus de Viesques, 33204, Gijón, Oviedo, España
luciano@ccia.uniovi.es

Resumen En este trabajo se presenta el proyecto KEEL (Knowledge Extraction based on Evolutionary Learning) en el que se está desarrollando la herramienta software KEEL 1.0. Esta nos permite utilizar y construir diferentes modelos para Minería de Datos y tiene como característica importante la inclusión de una librería de algoritmos de aprendizaje evolutivo. También se está desarrollando una librería de problemas KEEL-dataset, una página WEB asociada que contiene las bases de datos de los problemas para aprendizaje no supervisado, clasificación y regresión, y que incluye las particiones utilizadas junto con los resultados obtenidos por los diferentes algoritmos implementados en KEEL.

1. Introducción

En la actualidad existen múltiples herramientas software de aprendizaje y minería de datos. Características que no tiene estas herramientas en muchos casos son: la inclusión de algoritmos de aprendizaje evolutivo, la disponibilidad del código abierto, la inclusión de herramientas estadísticas para el análisis de algoritmos.

* Financiado por el Ministerio de Ciencia y Tecnología y los Fondos FEDER bajo el proyecto TIC-2002-04036-C05.

En el proyecto coordinado "KEEL: Entorno para la Extracción de Conocimiento basado en Algoritmos de Aprendizaje Genético y Evolutivo", TIC-2002-04036-C05, nos planteamos como objetivo el desarrollo de una herramienta software que incluyese los requerimientos anteriormente indicados.

En el proyecto se desarrollan dos grandes líneas de trabajo:

- una línea de aportaciones científicas en la que se desarrollan diferentes propuestas de aprendizaje evolutivo entre los distintos miembros del proyecto. Las publicaciones pueden ser consultadas en la pagina web (http://sci2s.ugr.es/keel/publication.php y http://keel.ugr.es/publication.php)
- una línea de desarrollo de una herramienta software que nos permita utilizar y construir diferentes modelos para minería de datos, KEEL 1.0, junto con de una página WEB, que contiene una librería de problemas que pretende servir de soporte a la actividad científica, KEEL-dataset.

En el proyecto intervienen 5 grupos de investigación en Aprendizaje Evolutivo, que son:

- Soft Computing y Sistemas de Información Inteligentes (Universidad de Granada). Investigador principal: Francisco Herrera
- Aprendizaje y Redes Neuronales Artificiales (Universidad de Córdoba). Investigador principal: Cesar Hervás
- Grupo de Investigación en Sistemas Inteligentes (Universitat Ramon Llull). Investigador principal: Josep M. Garrell
- Sistemas Inteligentes (Universidad de Jaén). Investigador principal: María José del Jesus
- Metrología y Modelos (Universidad de Oviedo). Investigador principal: Luciano Sánchez

Este capítulo se centra en la presentación del desarrollo software del proyecto KEEL. En la sección 2 presentaremos la herramienta software de Minería de Datos KEEL 1.0. En la sección 3 describiremos el contenido de KEEL-dataset. Por último, en la sección 4 presentaremos unos comentarios finales.

2. KEEL 1.0: Herramienta de Minería de Datos

KEEL 1.0 es una herramienta software desarrollada dentro del proyecto KEEL para utilizar y construir diferentes modelos de Minería de Datos. Destacar que es la primera herramienta software de este tipo que contiene una librería de algoritmos de aprendizaje evolutivo con código abierto en Java. Las características principales de KEEL 1.0 son las siguiente:

- Dispone de algoritmos de preprocesamiento: transformación, discretización, selección de instancias y selección de características.
- Contiene una librería de algoritmos de Extracción de Conocimiento: Supervisado y no Supervisado, destacando la incorporación de múltiples algoritmos de aprendizaje evolutivo.
- Dispone de una librería de herramientas estadísticas para el análisis de algoritmos.
- Incluye una Librería de Programación de Algoritmos Evolutivos en Java JCLEC (Java Class Library for Evolution Computation).
- Se desarrolla con un entorno de usuario de fácil uso y orientado hacia el análisis de los algoritmos.
- Aplicación para su uso vía WEB, que envía al usuario la información necesaria para realizar el experimento diseñado sobre la maquina que desee.

El software está implementado en Java, lo que permite:

- Una gran portabilidad del software generado, independientemente de la plataforma sobre la que se ejecute, y
- disponibilidad de una herramienta con acceso vía WEB tal como hemos indicado.

Dentro de KEEL 1.0 podemos distinguir 4 partes:

1. Experimental Setup
2. Statistical Analysis Tools
3. Data Preparation
4. Knowledge Extraction

A continuación describiremos brevemente cada una de estas partes. Además, presentaremos los formatos de los ficheros de Entrada/Salida de los algoritmos en KEEL 1.0 y el entorno gráfico suministrado al usuario para poder utilizar todo este software.

2.1. Experimental Setup

Existen muchas formas de configurar la experimentación que se va a realizar con los algoritmos de aprendizaje. En KEEL 1.0 se nos proporcionan las siguientes opciones:

- Hold Out:
  1. Indicamos manualmente los ficheros de las particiones que queremos utilizar
  2. Proporcionamos el porcentaje de las particiones y el fichero global, generando la herramienta software los ficheros sobre los que se ejecutará el algoritmo a partir del fichero global.

- Cross Validation:
  1. Realizar un k-fold cross-validation (por defecto 10-fold cross validation) [1]
  2. Realizar el 5x2 cross-validation de Dietterich [2]

2.2. Statistical Analysis Tools

Un soporte muy importante en KEEL 1.0 es la librería de herramientas estadísticas que estamos desarrollando para comparar los resultados obtenidos por los algoritmos. Esta librería está compuesta por:

- Test de Independencias: P-Pearson
- Test de Normalidad: Kolmogorov-Smirnov
- Test de Igualdad de Medias: ANOVA
- Test de Igualdad de Varianzas: Levene
- Comparación de dos Poblaciones: Test t-student, Test de Wilcoxon, Test Binomial
- Comparación de más de 2 poblaciones: Ajuste de Bonferroni y Ajuste de Tamhane.

2.3. Data Preparation

KEEL 1.0 nos proporciona una serie de herramientas para el preprocesamiento o preparación de datos:

- Discretizar del dominio de las variables continuas
- Selección de características
- Selección de instancias
- Transformación de los Datos:
  1. Escalado de los datos al intervalo [0,1]
  2. Tipificación de los datos
  3. Transformación de atributos nominales a atributos numéricos

- Herramientas de editado de Bases de Datos

2.4. Knowledge Extraction

KEEL 1.0 contiene una librería de algoritmos para Knowledge Extraction que abarca el siguiente abanico de técnicas:

- Árboles de Decisión
- Extracción de Reglas y Aprendizaje Supervisado
  - Aprendizaje de conceptos y de Reglas Intervalares para Clasificación
  - Sistemas Basados en Reglas Difusas para Clasificación
  - Sistemas Basados en Reglas Difusas para Regresión

- Extracción de Reglas e Inducción Descriptiva
  - Descubrimiento de Subgrupos
  - Reglas de Asociación

- Métodos Estadísticos para Clasificación
  - Técnicas Paramétricas

    - Técnicas No Paramétricas

- Métodos Estadísticos para Regresión
    - Técnicas Paramétricas
    - Técnicas No Paramétricas

- Otros Métodos Evolutivos para Clasificación
- Otros Métodos Evolutivos para Regresión
- Redes Neuronales
- Multiclasificadores - Combinación de Clasificadores
- Aprendizaje No Supervisado

2.5. Formatos de Entrada/Salida

Los algoritmos implementados en KEEL 1.0 utilizan el formato del fichero de datos presentado en el apéndice A.1. Este formato es una extensión del que utilizan los algoritmos implementados en WEKA [3] y permite al usuario tener una mayor flexibilidad para definir los ficheros.

Todos los algoritmos implementados en KEEL 1.0 generan dos ficheros de salida en el formato que se describe en el apéndice A.2. Además, los algoritmos de preprocesamiento de datos también generan un fichero, en el formato del apéndice A.1, con el resultado de aplicar el algoritmo sobre el fichero de datos.

Estos formatos son los únicos requisitos que tienen que cumplir un algoritmo para poder ser integrado en KEEL 1.0.

2.6. Entorno Gráfico

En el entorno gráfico podemos distinguir 3 partes bien diferenciadas:

1. Preparación de las Bases de Datos
2. Diseño de Experimentos
3. Generación de algoritmos evolutivos con la librería JCLEC

La parte de *Preparación de las Bases de Datos* hace referencia a las herramientas descritas en la subsección 2.3. Esta parte del entorno permite al usuario crear sus propias particiones sobre bases de datos propios o disponibles en KEEL-dataset (ver sección 3), además de permitir realizar transformación sobre los datos para adaptarlos a la ejecuciones de nuestros algoritmos.

El *Diseño de Experimentos* del interfaz gráfico sirve para definir gráficamente experimentos. Su propósito es generar una estructura de directorios con los ficheros necesarios para poder ejecutar el experimiento diseñado en una máquina local que el usuario decida (ver figura 1). Una vez almacenada esta estructura en la máquina seleccionada el usuario podrá lanzar el experimento ejecutando una clase Java que se encargará de ejecutar toda la experimentación diseñada. Toda esta estructura será enviada al usuario en un fichero con extensión *.zip*.

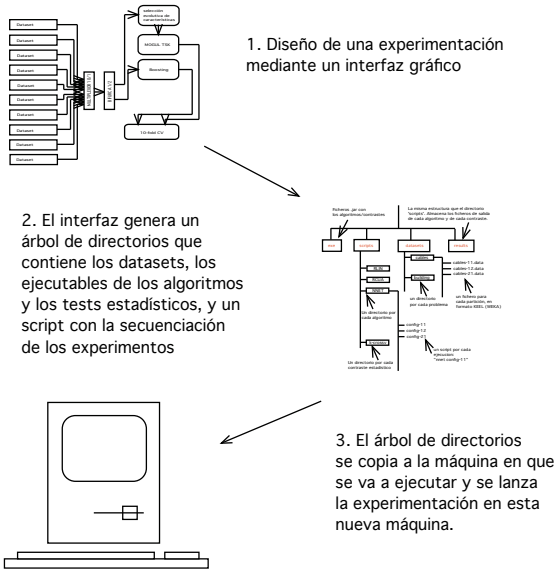

Figura 1. Diseño de Experimentos

Además, la interfaz permite al usuario añadir a la experimentación un algoritmo implementado por él. Para ello, el algoritmo tendrá que aceptar los formatos de los ficheros descritos en el apéndice A. Destacar que, como la interfaz solo genera los ficheros de ejecución los algoritmos el usuario no tendrá que implementar sus algoritmos en Java, puesto que sólo tendrá que colocar el fichero ejecutable en el directorio correspondiente (indicado por la interfaz). Esto proporciona una gran flexibilidad al usuario para poder comparar sus métodos con los que hay disponibles en KEEL 1.0.

Por último, la *Generación de algoritmos evolutivos con la librería JCLEC* permite al usuario crear sus propios algoritmos evolutivos mediante una interfaz gráfica. Para ello la interfaz utiliza un lenguaje de cajas y una librería de métodos/rutinas implementados con la librería JCLEC.

3. KEEL-dataset: Librería de problemas

Otra herramienta desarrollada en el proyecto es KEEL-dataset. Como hemos indicado, es una librería de problemas para aprendizaje no supervisado (clustering, reglas de asociación, etc), clasificación y regresión. Estamos diseñando una WEB (http://sci2s.ugr.es/keel-dataset ó http://keel.ugr.es/keel-dataset) que contiene la librería de problemas.

Cada uno de los problemas disponibles en la librería contiene:

- una descripción del problema

- una serie de particiones asociadas al problema:
  1. particiones al $90\,\% - 10\,\%$, $80\,\% - 20\,\%$ ó $75\,\% - 25\,\%$ para entrenamiento y prueba respectivamente, para realizar el 10 fold cross-validation [1].
  2. Cinco particiones al $50\,\%$ para realizar el 5x2 cross-validation de Dietterich [2]

- y un conjunto de tablas con los resultados obtenidos de ejecutar varios algoritmos sobre las particiones proporcionadas.

Se invita a cualquier usuario de la misma a ejecutar sus métodos sobre las particiones asociadas a estos problemas y que nos envíe los resultados obtenidos para ser añadidos en las tablas. Esta página WEB será un soporte importante para el análisis de algoritmos de aprendizaje, tanto los implementados dentro de KEEL como cualquier otro algoritmo ejecutados sobre estas particiones.

Los problemas que hay disponibles en esta página son los siguientes:

- Aprendizaje No Supervisado
  1. Fraccion de energía eléctrica generada cada hora del día en el 2003

- Regresión
  1. Series Temporales
     - Sunspot (Manchas Solares) [4]
     - Laser (Competición de Santa Fé) [5]
  2. Datos de Consumo Eléctrico
     - Estimación de la longitud de la línea de baja tensión en nucleos rurales de Asturias [6]
     - Precio medio diario de la energia eléctrica en España en el 2003
  3. Benchmark de Friedman [7]

- Clasificación
  1. Bases de Datos Pequeñas:
     - Iris[1]
     - Pima[1]
     - Cleveland [1]
     - Glass[1]
     - Led de 7 dígitos[1]
     - Led de 24 dígitos[1]
     - Wine[1]
     - Wisconsin[1]
     - Lung cancer[1]
     - Breast cancer[1]
     - Lymphography cancer[1]
     - Primary tumor[1]

[1] Base de Datos obtenida de la UCI (http://kdd.ics.uci.edu)

2. Bases de Datos Medianos:
   - Pen-based[1]
   - Satimage[1]
   - Thyroid[1]

3. Bases de Datos Grandes:
   - Adult[1]
   - Kddcup99[1]

4. Comentarios Finales

Una vez terminado KEEL 1.0 (tenemos previsto el tener un prototipo funcionando para Febrero de 2005) se dispondrá de una herramienta software que nos permitirá realizar análisis de comportamiento de los diferentes algoritmos de minería de datos disponibles.

KEEL-dataset permitirá contrastar los resultados de nuevas propuestas de algoritmos con los resultados obtenidos por otros algoritmos y puede ser una herramienta muy útil para todos los investigadores interesados en el desarrollo y análisis del comportamiento del algoritmos de minería de datos.

Actualmente este proyecto se encuentra en fase de desarrollo. Invitamos a cualquier lector a colaborar con nosotros, y a enviarnos algoritmos en Java que puedan integrarse en KEEL. Para ello pueden contactar con nosotros a través de la pagina web del proyecto (http://sci2s.ugr.es/keel y http://keel.ugr.es).

A. Formatos de Entrada/Salida KEEL

A.1. Fichero de Datos KEEL

El formato de los ficheros de datos para los algoritmos implementados en KEEL es una extensión del formato de datos presentado en WEKA [3], permitiendo una mayor flexibilidad al usuario. El formato es el siguiente:

- Nombre del conjunto de datos (obligatorio):

  @relation $< nombre >$

- Atributos (obligatorio):

  @attribute $< nombre >$ integer $[min, max]$
  @attribute $< nombre >$ real $[min, max]$

[1] Base de Datos obtenida de la UCI (http://kdd.ics.uci.edu)

@attribute $< nombre >$ $\{< valor_1 >, < valor_2 >, ... , < valor_N >\}$

(Si no se indican los intervalos o la lista de valores se extraerán de la lista de datos)

- Entrada / salida (opcional): Sólo si es necesario en el problema

  @inputs $< nombre_1 >, < nombre_2 >, ... , < nombre_X >$
  @outputs $< nombre_1 >, < nombre_2, ... , < nombre_Y >$

- Datos (obligatorio): Se sigue el orden en el que se definieron los atributos

  @data
  x_{11}, x_{12}, ... , x_{1N}
  x_{21}, x_{22}, ... , x_{2N}

Si necesitamos indicar un valor nulo en los datos utilizaremos $< null >$. Por último destacar que los identificadores pueden contener espacios y que su tamaño máximo es de 12 caracteres.

Estos ficheros se almacenarán, por defecto, con la extensión "dat". Un ejemplo de este tipo de ficheros puede ser el siguiente:

@relacion paint
@attribute colour {yellow, white, black}
@attribute amount integer [1, 10]
@attribute density real [0.1, 2.5]
@inputs colour, amount
@outputs density
@data
yellow, 4, 1.2
black, 1, 2.1
yellow, 2, 0.8
white, 8, 0.9

A.2. Ficheros de Salida KEEL

Todos los métodos implementados en KEEL tienen que generar dos ficheros de salida: uno para entrenamiento (training) y otro para test. El formato de estos ficheros es el mismo que el de los ficheros de datos exceptuando que a continuación de @data se indica:

$< salida_{11} > ... < salida_{1N} > < salida_{11}\ metodo > ... < salida_{1N} metodo >$
$< salida_{21} > ... < salida_{2N} > < salida_{21}\ metodo > ... < salida_{2N} metodo >$

en lugar de los datos de entrada, es decir, las columnas de salida de los datos de entrada y a continuación las salidas que ha generado nuestro método (separada cada columna por un espacio en blanco). Esta información la puede generar todos los métodos que tenemos que implementar y a partir de estos se puede obtener toda la información que necesitamos de los métodos para realizar los test estadísticos.

Como cada método tendrá como entrada un fichero de entrenamiento y otro de test habrá que crear dos ficheros de salida con este formato, uno con extensión .tra y el otro .tst. Por lo tanto, habrán el doble de ficheros de salida que de parámetros.

Un ejemplo del tipo de fichero que tenemos que generar es el siguiente:

Fichero de entrada de entrenamiento:

@relacion furniture
@attribute height real [1, 10]
@attribute width real [1, 10]
@attribute profundity real [1, 10]
@attribute amount real [1, 10]
@attribute import real [3, 10]
@inputs height, width, profundity, amount
@outputs import
@data
3, 4, 5, 5.6, 7.5
1, 2.8, 3, 4.1, 3.9

Fichero result.tra generado:

Cuadro 1. Fichero de Datos

Fichero de Datos		**Salida del Método**	**Fichero de Salida**	
Entradas	**Salida**			
3 4 5 5.6	7.5	8.2	7.5	8.2
1 2.8 3 4.1	3.9	3.1	3.9	3.1

@relacion furniture
@attribute height real [1, 10]
@attribute width real [1, 10]
@attribute profundity real [1, 10]
@attribute amount real [1, 10]
@attribute import real [3, 10]

@inputs height, width, profundity, amount
@outputs import
@data
7.5 8.2
3.9 3.1

Referencias

1. Stone, M.: Cross-validatory choice and assessment of statistical predictions (with discussion). Journal of the Royal Statistical Society B **36** (1974) 111–147
2. T.G.Dietterich: Approximate statistica tests for comparing supervised classification learning algorithms. Neural Computation 10:7 (1999) 1895–1924
3. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann (1999)
4. Marple, S.: Digital Spectral Analysis with Applications. Prentice-Hall (1987)
5. U. Huebner, N.B. Abraham, C.W.: Dimension and antropies of a chaotic intensity pulsations in a single-mode far-infrared NH3 laser. Physics review, A 40 (1989)
6. O. Cordón, F. Herrera, L.S.: Solving electrical distribution problems using hybrid evolutionary data analysis techniques. Applied Intelligence **10** (1999) 5–24
7. N. Friedman, D. Geiger, M.G.: Bayesian network classifiers. Machine Learning **29** (1997) 131–163