$N_ip_{1.5}$: UNA HERRAMIENTA SOFTWARE PARA LA GENERACIÓN DE CONJUNTOS DE DATOS CON IMPERFECCIÓN PARA MINERÍA DE DATOS

José M. Cadenas **Juan V. Carrillo** **M. Carmen Garrido** **Enrique Muñoz**
Departamento de Ingeniería de la Información y las Comunicaciones
Universidad de Murcia. Murcia. Spain
jcadenas@um.es jvcarrillo@um.es carmengarrido@um.es enriquemuba@um.es

Resumen

Este trabajo introduce una herramienta software llamada $N_ip_{1.5}$ la cual ayuda a generar y tratar conjuntos de datos con imperfección para el proceso de minería de datos y aprendizaje. Esta herramienta software ha sido concebida para su uso en entornos de investigación e incluye la posibilidad de distintos formatos de entrada/salida standard y definidos por el usuario; gestión de los atributos; generar e introducir valores desconocidos, ruido y etiquetas lingüísticas, etc.

Palabras Clave: Herramienta software, Datos con imperfección, Soft Computing.

1 INTRODUCCIÓN

En el proceso de descubrimiento de conocimiento en conjuntos de datos se debe tener en cuenta la imperfección de la información para poder extraer modelos más cercanos a la realidad. Sin embargo, a pesar de la gran cantidad de técnicas existentes para dicho proceso, sólo unas pocas tienen en cuenta la imperfección, entre otros motivos, por la falta de conjuntos de datos imperfectos con las que probar y comparar los algoritmos. Aunque el objetivo de las técnicas es ser usadas con conjuntos de datos con imperfección real, la manipulación de forma controlada de conjuntos de datos "perfectos" permite experimentar y comparar resultados de los algoritmos que gestionen datos imperfectos, ofreciendo un marco de trabajo común; este es el propósito de $N_ip_{1.5}$ [4].

Este trabajo presenta la herramienta $N_ip_{1.5}$, un software de tratamiento de datos que permite crear conjuntos de datos imperfectos, introduciendo ruido e imprecisión entre otras características. Además, $N_ip_{1.5}$ es capaz de generar etiquetas lingüísticas fuzzy mediante un algoritmo de particionamiento. Para ello el trabajo será estructurado de la siguiente forma: en la Sección 2 se hará una introducción a la imperfección que nos podemos encontrar en los datos, en la Sección 3 se presentarán algunas herramientas usadas para el tratameinto de datos. Posteriormente, en la Sección 4, se presentará la herramienta $N_ip_{1.5}$. Finalmente en la Sección 5 se presentan las conclusiones.

2 IMPERFECCIÓN EN CONJUNTOS DE DATOS

La información imperfecta aparece de manera general en dominios y situaciones reales. Los errores instrumentales o la corrupción debida al ruido durante la recogida de datos puede dar lugar a información incompleta para atributos específicos. En otros casos, la extracción de información exacta puede ser excesivamente costosa. En ese caso, puede ser útil aumentar el conjunto de datos con información adicional de un experto, que suele trabajar con términos lingüísticos fuzzy como *pequeño, más o menos, cercano a,* etc.

2.1 RUIDO Y VALORES DESCONOCIDOS

El ruido es un problema común en el análisis de datos, y como consecuencia debe ser gestionado y tratado debidamente. Este se define, [7], como cualquier cosa que oscurece la relación entre los atributos y la clase. De esta definición se obtienen las fuentes de ruido que aparecen en los datos: atributos erróneos, atributos desconocidos, atributos incompletos y atributos redundantes. El tratamiento de los atributos erróneos es complejo debido a la dificultad de distinguirlos de atributos correctos. Para gestionar los atributos desconocidos existen distintos enfoques: eliminar las instancias con atributos desconocidos, sustituir dichos atributos por su media o moda, usar fuzzy sets o intervalos, y otros métodos más complejos.

2.2 IMPRECISIÓN E INCERTIDUMBRE

Según la Teoría de la Posibilidad [6], un item de información puede ser impreciso e incierto. La imprecisión se manifiesta en el contenido de un ítem de información, mientras que la incertidumbre se refiere a su parecido con la realidad.

En [6] se define un ítem de información como una 4-tupla (atributo, objeto, valor, confianza) que representa el valor tomado por un objeto y la confianza sobre dicho valor. Un ítem de información es preciso cuando el conjunto correspondiente a su componente valor no se puede subdividir; en caso contrario hablamos de información imprecisa. Hay otros términos para referirse a la información imprecisa como vago, fuzzy, general o ambiguo. Un ítem de información es ambiguo cuando se refiere a contextos diferentes o conjuntos diferentes de referencia; es general si se nombra a una clase de objetos de los cuales se expresa una propiedad común; es vago si hay ausencia de límites claros en el conjuntos de valores asignados al objeto.

Algunos autores hablan de una "escala de conocimiento" que va desde la *certeza* absoluta a la *falta de conocimiento*, pasando por la incertidumbre [8], aunque no hay acuerdo sobre una terminología común sobre la incertidumbre.

2.3 REPRESENTAR LA INCERTIDUMBRE

Como se ha comentado, la incertidumbre y la imprecisión son conceptos complementarios, ya que la incertidumbre se debe normalmente a la imprecisión en los datos. Los datos pueden ser representados junto a su imprecisión y/o incertidumbre usando intervalos, modelos estocásticos o fuzzy sets, utilizando un conjunto de valores en lugar de uno único. La teoría de la probabilidad, el análisis de intervalos y la teoría de la posibilidad son los marcos matemáticos de trabajo más usados que nos permiten conceptualizar la incertidumbre y la imprecisión.

La primera aproximación para representar la incertidumbre es el análisis de intervalos, representando un valor con un intervalo. Se utilizan conjuntos clásicos en lugar de un número, pero el grado de pertenencia sigue siendo un valor binario. La teoría de la probabilidad es otra manera de representar la incertidumbre, haciendo un análisis probabilístico para representar la incertidumbre mediante la probabilidad asociada con los eventos. Las incertidumbre asociadas a las entradas del modelo se describen mediante distribuciones de probabilidad, y el objetivo es estimar la salida de dichas distribuciones. La teoría de los Fuzzy Sets y la teoría de la posibilidad [6], como extensión de la noción de conjuntos clásicos, reemplazan la función bivaluada de pertenencia al conjuntos por una función con valor real; esto es, un fuzzy set representa el concepto de número aproximados, y la pertenencia al conjunto viene dada por $\alpha \in [0, 1]$.

3 HERRAMIENTAS Y CONJUNTOS DE DATOS

3.1 CONJUNTOS DE DATOS

Para poder comprobar la eficacia de una técnica de aprendizaje se utilizan conjuntos de datos obtenidas de diferentes entornos reales, algunas de las cuales se pueden obtener de repositorios públicos [1, 2]. Estos conjuntos de datos no suelen contener información imperfecta, a excepción de algún valor desconocido o, algunas de ellas, valores en intervalos. Como se comentó anteriormente, los problemas reales suelen tener algún tipo de información imperfecta que, de alguna manera, se ignora al publicar los conjuntos de datos en los repositorios. Por otro lado, cuando se diseña una técnica capaz de gestionar información imperfecta, los autores tienen que crear sus propios conjuntos de datos para probar y ajustar su método. Con $\mathrm{N_i p_{1.5}}$, [4], se facilita dicho proceso de una manera más formal y rigurosa para simplificar asimismo la comparación de resultados con otros autores. En la siguiente sección veremos algunas de las herramientas software de minería de datos junto con sus principales características desde este punto de vista.

3.2 HERRAMIENTAS SOFTWARE

En esta subsección se verán algunas de las herramientas más usadas para el tratamiento de datos, especialmente aquellas destinadas a la minería de datos. Además, se comparan sus principales características con las de $\mathrm{N_i p_{1.5}}$.

- *Sodas2* [5]: Es una herramienta que soporta el análisis simbólico de datos. Trata de generalizar el proceso de minería de datos y estadística a un nivel superior, descrito por datos simbólicos. De esta manera, se transforman los datos en datos más manejables y más complejos. El funcionamiento de *Sodas2* consiste en construir un conjunto de datos simbólicos que resume la información del conjunto de datos inicial y, tras ello, realizar el *análisis simbólico*.

  El uso de datos simbólicos permite introducir diferentes tipos de imperfección en los datos: - Atributos multivaluados, - intervalos y - Atributos multivaluados con pesos. Este tipo de atributos permiten representar otros tipos de imperfección,

Tabla 1: Características de las herramienta Software con respecto al tratamiento de datos.

(S: Sí, N: No, I:Medio)		Sodas2	Weka	Keel	RapidMiner	MiningMart	$N_{i}P_{1.5}$
Gestión de datos		S	S	S	S	S	S
Formato	Standard	I	S	S	S	I	I
	Usuario	N	N	N	N	N	S
Discretización	Crisp	S	S	S	S	S	S
	Fuzzy	N	N	N	N	N	S
Valores Desconocidos	Introducción	N	S	S	N	N	S
	Imputación	S	S	S	I	I	I
Ruido en atributo	Nominal	N	S	N	S	N	S
	Numérico	N	N	N	S	N	S

como datos fuzzy, imprecisos e inciertos. No obstante, *Sodas2* no permite el uso directo de tecnología fuzzy ni la introducción de valores desconocidos, ruido o la modificación de los datos para introducir ningún tipo de imperfección.

- *Weka* [13]: Proporciona un conjunto de herramientas para el preproceso de datos, clasificación, regresión, clustering, reglas de asociación y visualización. Para el preproceso de datos, Weka ofrece una amplia variedad de técnicas para preproceso de atributos e instancias. En el caso de datos con imperfección, el número de técnicas disminuye, ofreciendo pocas posibilidades. Weka únicamente es capaz de gestionar valores desconocidos, y proporciona herramientas para sustituir dichos valores por la moda o media. Además, permite la adición de ruido en atributos nominales.
- *Keel* [1]: Es un software para evaluar algoritmos evolutivos para problemas de minería de datos. Contiene una gran colección de algoritmos clásicos de extracción de conocimiento, técnicas de preproceso, algoritmos de aprendizaje basados en inteligencia computacional, modelos híbridos, y un módulo de tests estadísticos para comparación. KEEL permite el uso de diferentes formatos de entrada y salida como CSV, XML o ARFF.

  Su módulo de *Gestión de Datos* permite realizar la fase de preproceso proporcionando una amplia variedad de técnicas para selección de instancias, selección de características, discretización, etc. Una vez más, las técnicas proporcionadas para gestión de datos imperfectos no son muchas, ya que sólo se permite el uso de valores desconocidos mediante varios métodos de imputación (borrado de instancias con desconocidos, imputación de k-vecinos más cercanos, imputación de k-medias, etc).
- *Rapid miner* [10]: Antiguamente llamado YALE, es un entorno para aprendizaje computacional y minería de datos destinado a soportar el paradigma de prototipado rápido. Ofrece la posibilidad de usar diferentes formatos de entrada/salida. El número de técnicas de preproceso ofrecidas por la herramienta es grande pero, de nuevo, disminuye si nos centramos en técnicas que gestionen datos imperfectos. Rapid miner permite gestionar valores desconocidos usando varios métodos de imputación: reemplazo con la media, moda, máximo, mínimo o el valor predicho mediante una técnica de minería de datos definida por el usuario. También permite la introducción de ruido en atributos numéricos y/o nominales.
- *MiningMart* [11]: Se desarrolla con el propósito de reutilizar técnicas con éxito en el preproceso de grandes volúmenes de datos. MiningMart no se centra en todo el proceso de descubrimiento de conocimiento, sino que sólo trata el preproceso. Ofrece diversas técnicas de agregación, discretización, limpieza de datos, tratamiento de valores nulos y selección de atributos relevantes. El único soporte que ofrece para datos imperfectos es sobre los valores desconocidos, eliminando las instancias que los contienen o bien usando algún método de imputación, como reemplazar el valor por la moda, media, un valor estocástico, etc., o usando un algoritmo para predecir el valor.
- *Otras herramientas*: Existen otras herramientas centradas en el proceso de extracción de conocimiento, como *ADaM* [12], *D2K* [9] or *KNIME* [3], entre otras, pero no prestan mucha atención al tratamiento de datos imperfectos.

En la Tabla 1 podemos ver un resumen de las características principales de las herramientas examinadas. La mayoría centran su funcionalidad en el tratamiento de los datos, teniendo sólo en cuenta la imperfección de tipo valor desconocido. En cuanto a la introducción de imperfección, son sólo algunas herramientas las que permiten introducir valores desconocidos, menos las que introducen ruido, y no hay ninguna que trabaje con conjuntos fuzzy.

$N_{i}P_{1.5}$ surge para paliar esta falta en el estado del arte de herramientas de ayuda a la investigación en el

campo de descubrimiento de conocimiento. Este software es capaz de generar conjuntos de datos con esos tipos de imperfección que el resto de herramientas no soporta o soporta parcialmente.

4 NIP 1.5

NiP1.5 [4] es una herramienta que permite generar y gestionar conjuntos de datos con imperfección, concebida para su uso en entornos de investigación debido a la carencia de un software de características similares que ayude a establecer un marco común de trabajo con datos imperfectos. En esta sección veremos las principales funcionalidades del sistema, describiendo el uso de las diversas opciones. La figura 1 muestra, de manera muy general, las funcionalidades del software.

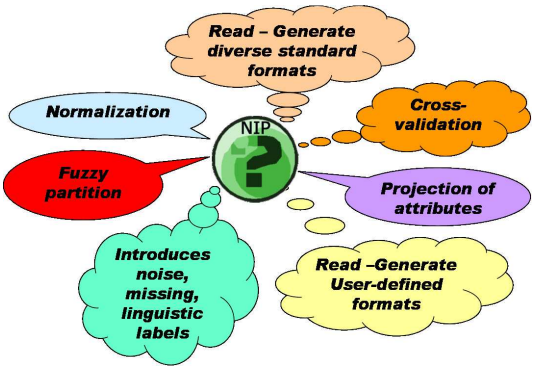

Figura 1: Funcionalidades de NiP1.5

4.1 FORMATOS DE ENTRADA/SALIDA

NiP1.5 tiene cuatro formatos de entrada y salida predefinidos o standard: WEKA, KEEL, UCI, CSV. Estos formatos son los más usados en la literatura y en los repositorios de conjuntos de datos [1, 2, 13]. Por otra parte, dado que se trata de una herramienta para la preparación de conjuntos de datos para su posterior uso con herramientas de tratamiento, el usuario puede definir su propio formato utilizando: separadores, representación de valores desconocidos, peso de los ejemplos, etc. En cuanto a los formatos de salida, también son personalizables. Además, NiP1.5 permite generar conjuntos de datos, según el formato elegido, para realizar una validación cruzada (Figura 2).

4.2 GESTIÓN DE ATRIBUTOS

Cuando NiP1.5 analiza el conjunto de datos, asigna a cada atributo un tipo, numérico o nominal . Así mismo, NiP1.5 ofrece información útil sobre las características del conjunto de datos, como el número de ejemplos, porcentaje de valores desconocidos, y valores estadísticos (Figura 3). Además, el usuario puede realizar las siguientes acciones:

Figura 2: Formato entrada/salida

1. Cambiar el nombre de un atributo.
2. Cambiar el orden de los atributos.
3. Borrar atributos.
4. Para atributos numéricos, normalizar a $[0, 1]$.
5. Cambiar el tipo del atributo de numérico a numérico y viceversa (siempre que el nominal tenga valores numéricos).

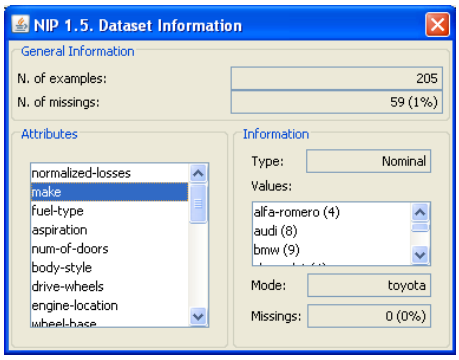

Figura 3: Información del conjunto de datos

4.3 INTRODUCCIÓN DE IMPERFECCIÓN

En esta subsección veremos las distintas opciones que ofrece NiP1.5 para introducir imperfección en los conjunto de datos.

4.3.1 Valores desconocidos

NiP1.5 permite introducir imperfección al conjunto de datos en forma de valores desconocidos. Dichos valores se pueden introducir para todos los atributos o a un subconjunto de ellos. El número de valores desconocidos a añadir depende de un porcentaje establecido por el usuario, que puede ser distinto para cada atributo (Figura 4). La introducción de valores desconocidos es útil si se quiere corromper el conjunto de datos de manera aleatoria; no obstante, hay que tener en cuenta

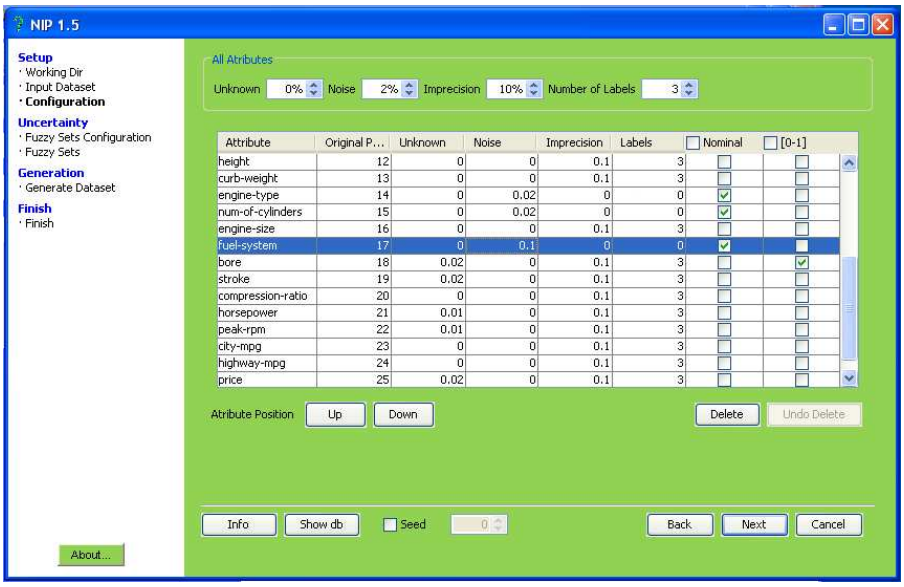

Figura 4: Pantalla de configuración

que en un conjunto de datos real con imperfección, los valores desconocidos pueden seguir un patrón concreto que no es posible simular.

Por otro lado, NiP1.5 no sólo introduce valores desconocidos en el conjunto de datos, sino que también aplica un tratamiento a valores desconocidos ya existentes. Si el algoritmo a usar no soporta valores desconocidos, NiP1.5 puede reemplazar dichos valores por la moda, media, intervalos o fuzzy sets (Figura 5).

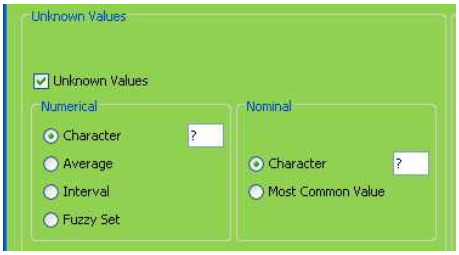

Figura 5: Gestión valores desconocidos

puede indicar el usuario, mediante un formato predefinido, o bien se puede dejar que NiP1.5 seleccione la mejor partición para el atributo basándose en un algoritmo de Árbol de Decisión Fuzzy (Figura 6). Esta imprecisión se puede representar de diversas formas:

- Con el nombre de la etiqueta lingüística.
- Con los valores del fuzzy set trapezoidal (V_1, V_2, V_3, V_4).
- Con los valores y el valor asociado $(V_1/\mu_1(V_1), V_2/\mu_2(V_2), V_3/\mu_3(V_3), V_4/\mu_4(V_4))$.

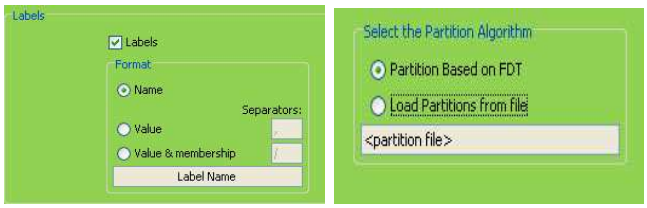

Figura 6: Introduciendo fuzzy sets

4.3.2 Imprecision

Como hemos visto en la sección 2.2, la teoría de la posibilidad [6] y la teoría de intervalos sirven para representar información imprecisa. NiP1.5 permite añadir imprecisión al conjunto de datos sustituyendo los valores de un atributo numérico por un fuzzy set que contenga dicho valor. El usuario decide qué atributos incluyen este tipo de imprecisión y en qué porcentaje (Figura 4). Para realizar esta transformación, NiP1.5 necesita de una partición fuzzy para cada atributo al que se vaya a aplicar dicha alteración. La partición la

4.3.3 Ruido

El ruido existente en los datos viene determinado tanto por el tipo de valores de los atributos como por la exactitud en la medida de dichos valores. Con NiP1.5 se puede introducir ruido para todos los atributos nominales o para un subconjunto de ellos. El porcentaje de ruido depende del valor indicado por el usuario, que puede ser distinto para cada atributo (Figura 4). Este tipo de ruido cambia el valor original del atributo por otro valor que esté en el conjunto de posibles valores de dicho atributo.

En el caso de los atributos numéricos, NiP1.5 permite introducir ruido gaussiano indicando la media μ y la desviación típica σ de la distribución para cada atributo. La representación en el conjunto de datos generada puede variar entre intervalos, fuzzy sets o funciones gaussianas (Figura 7).

Figura 7: Introduciendo ruido en atributo numérico

5 CONCLUSIONES

La falta de conjuntos de datos con representación de la imperfección inherente a cualquier recogida de datos hace difícil el desarrollo de técnicas de extracción de conocimiento que gestionen datos imperfectos. En este trabajo hemos descrito una herramienta software Java, llamada NiP1.5 , diseñada como una herramienta de ayuda a la generación y gestión de conjuntos de datos que sirva como marco común para la prueba y comparación de técnicas y algoritmos de minería de datos y aprendizaje.

Hemos presentado las principales características de esta herramienta software y hemos destacado que NiP1.5 es capaz de incluir en un conjunto de datos más tipos de imperfección que las herramientas actuales. Además NiP1.5 acepta formatos de entrada/salida predefinidos y personalizados, con lo cual la convierte en un software más flexible.

Agradecimientos

Los autores agradecen la financiación proporcionada por el proyecto TIN2008-06872-C04-03 del MICINN, y por el proyecto 04552/GERM/06 y al programa de FPI de la "Agencia de Ciencia y Tecnología" de la Región de Murcia, de España.

Referencias

[1] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández and F. Herrera, KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems. *Soft Computing* **13**(3) (2009) 307–318.

[2] A. Asuncion, and D.J. Newman, UCI Machine Learning Repository, http://www.ics.uci.edu/∼mlearn/MLRepository.html, Irvine, CA: University of California, School of Information and Computer Science, 2007.

[3] M.R Berthold, N. Cebron, F. Dill, T.R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel and B. Wiswedel, KNIME: The Konstanz Information Miner, in *Data Analysis, Machine Learning and Applications, eds. C. Preisach, H. Burkhardt, L. Schmidt-Thieme and R. Decker (Springer, Berlin, 2008)*, pp. 319–326.

[4] J.M. Cadenas, J.V. Carrillo, M.C. Garrido, R. Martínez-España, E. Muñoz, NIP 1.5 - A tool to handle imperfect information on datasets. Murcia University, 2008. *http://heurimind* .inf.um.es/NIP/index.htm

[5] E. Diday and M. Noirhomme-Fraiture, *Symbolic Data Analysis and the SODAS Software*, (Wiley-Interscience, New York, 2008).

[6] D. Dubois and H. Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty* (Plenium Press, Ney York, 1988)

[7] R. Hickey, Noise Modeling and Evaluating Learning from Examples, *Artificial Intelligence* **82**(1-2) (1996) 157–179.

[8] B. Klauer and J.D. Brown, Conceptualising imperfect knowledge in public decision making: ignorance, uncertainty, error and risk situations, *Environmental Research, Engineering and Management*, **27**(1) (2004) 124–128.

[9] X. Llorà, E2K: Evolution to knowledge, *SIGEVOlution* **1**(3) (2006) 10–16.

[10] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz and T. Euler, YALE: Rapid Prototyping for Complex Data Mining Tasks, in *Proc. 12th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining (KDD-06)* (2006) 935–940.

[11] K. Morik, M. Scholz, The MiningMart Approach to Knowledge Discovery in Databases, in *Intelligent Tech. for Information Analysis*, eds. N. Zhong, J. Liu (Springer, Berlin, 2004) pp. 47–65.

[12] J. Rushing, R. Ramachandran, U. Nair, S. Graves, R. Welch and H. Lin, ADaM: a data mining toolkit for scientists and engineers, *Computers & geosciences* **31** (5) (2005) 607–618.

[13] I.H. Witten, E. Frank *Data Mining: Practical machine learning tools and techniques, 2nd Edition*, (Morgan Kaufmann, San Francisco, 2005).