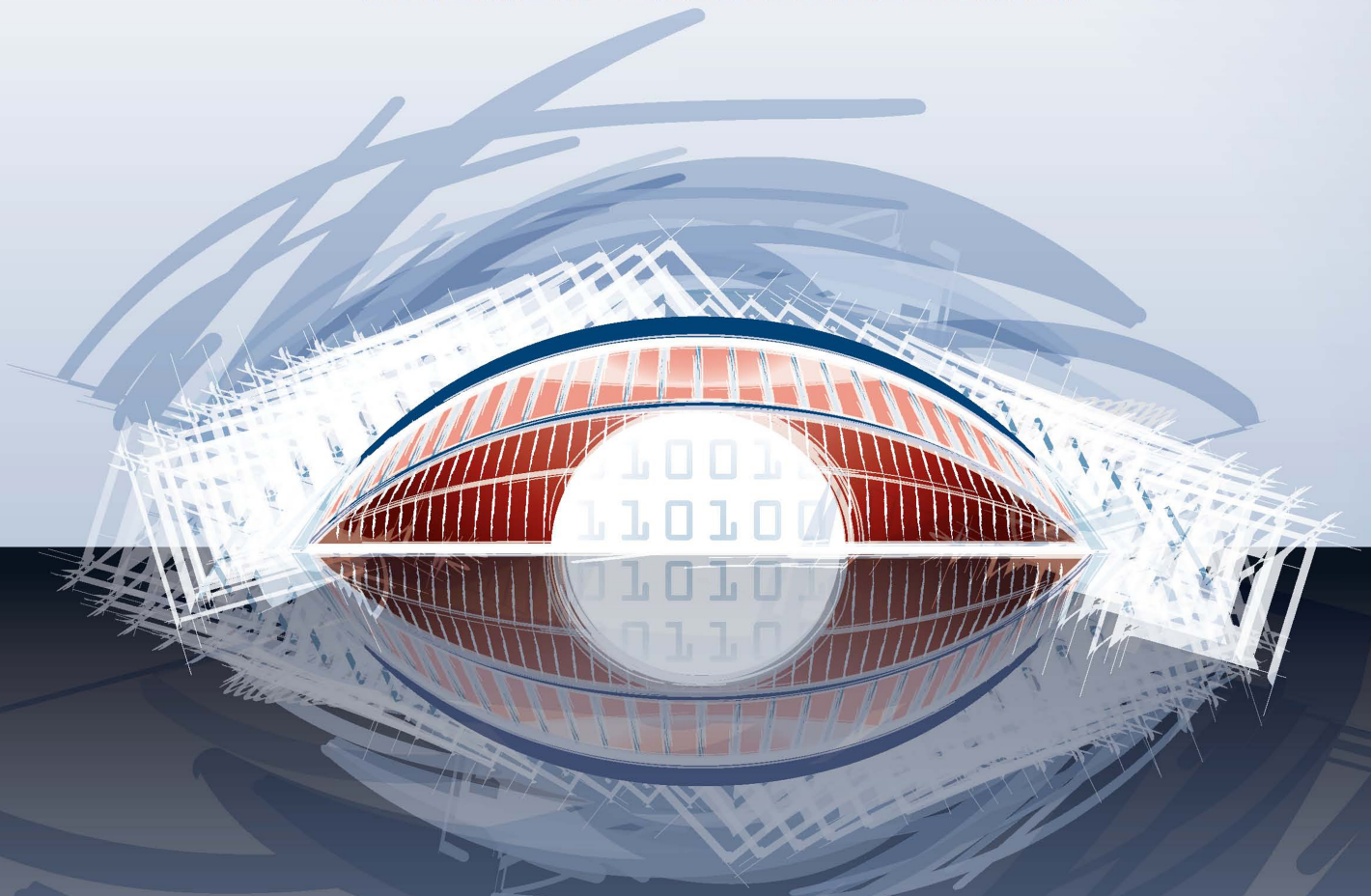
Garceta
grupo editorial

CEDI 2010
VALENCIA

7 A 10 DE SEPTIEMBRE DE 2010

III CONGRESO ESPAÑOL DE INFORMÁTICA

UNIVERSIDAD POLITÉCNICA DE VALENCIA



Actas del III Simposio sobre Lógica Fuzzy y Soft Computing

| LFSC2010 | (EUSFLAT)

EDITORES

Luis Martínez, Edurne Barrenechea, Macarena Espinilla, Jesús Alcalá, Victoria López, Manuel Mucientes, José Ángel Olivas, Rosa Mª Rodríguez

Nuevos retos científicos y tecnológicos en Ingeniería Informática

Actas del
III Simposio sobre Lógica Fuzzy y Soft Computing, LFSC2010
(EUSFLAT)

Editores:

Luis Martínez
Edurne Barrenechea
Macarena Espinilla
Jesús Alcalá
Victoria López
Manuel Mucientes
José Ángel Olivas
Rosa Mª Rodríguez

Garceta
grupo editorial

Actas del III Simposio sobre Lógica Fuzzy y Soft Computing, LFSC2010 (EUSFLAT)

Editores: Luis Martínez, Edurne Barrenechea, Macarena Espinilla Jesús Alcalá, Victoria López, Manuel Mucientes, José Ángel Olivas, Rosa Mª Rodríguez

ISBN: 978-84-92812-65-3
IBERGARCETA PUBLICACIONES, S.L., Madrid, 2010

Edición: 1ª
Impresión: 1ª
Nº de páginas: 494
Formato: 17 x 24
Materia CDU: 004 Ciencia y tecnología de los ordenadores. Informática

1ª Edición, 1ª Impresión

ISBN: 978-84-92812-65-3

Depósito legal: M-

Maquetación: Los Editores

Coordinación del proyecto: @LIBROTEX

Portada: Estudio Dixi

Impresión y encuadernación:

OI: 28/2010

PRINT HOUSE, S.A.

IMPRESO EN ESPAÑA -PRINTED IN SPAIN

Last results on constrained microaggregation.....

Isaac Cano, Guillermo Navarro-Arribas, Vicenç Torra

Comparación estadística de algoritmos de aprendizaje estocásticos usando tests extendidos a datos intervalo-valorados y borrosos..... 427

José Otero, Luciano Sánchez, Inés Couso

Influencia de un aprendizaje basado en costes lingüísticos a partir de datos de baja calidad y no balanceados respecto al preprocesamiento de balanceado de los datos..... 435 443

Ana Palacios, Luciano Sánchez, Inés Couso

Linguistic local change comparison of time series..... 451

Rita Castillo-Ortega, Nicolás Marín, Daniel Sánchez

Un primer estudio sobre el uso de aprendizaje sensible al coste con sistemas de clasificación basados en reglas difusas para problemas no balanceados..... 459

Victoria López, Alberto Fernández, Francisco Herrera

Análisis del impacto del ruido en clases y atributos para Sistemas de Clasificación Basados en Reglas Difusas..... 467

José A. Sáez, Julián Luengo, Francisco Herrera

Are fuzzy systems as interpretable (readable and understandable) as the fuzzy community usually claims?.....

Jose M. Alonso, Luis Magdalena

475

Análisis del impacto del ruido en clases y atributos para Sistemas de Clasificación Basados en Reglas Difusas

José A. Sáez Julián Luengo Francisco Herrera

Dept. Ciencias de la Computación e Inteligencia Artificial
CITIC - Universidad de Granada, 18071 Granada
smja@correo.ugr.es, {julianlm, herrera}@decsai.ugr.es

Resumen

En cualquier base de datos real, es habitual la presencia de ruido, el cual puede afectar negativamente a la precisión del clasificador, su tiempo de construcción y complejidad. Los clasificadores construidos por los Sistemas de Clasificación Basados en Reglas Difusas destacan por su gran interpretabilidad, pero tradicionalmente estos métodos no han tenido en cuenta este ruido en los datos, por lo que será interesante cuantificar su efecto en los mismos.

El objetivo de esta contribución es estudiar el comportamiento y robustez de los Sistemas de Clasificación Basados en Reglas Difusas en presencia de ruido. Para ello se han creado 138 bases de datos sintéticas a partir de 23 bases de datos sin ruido del repositorio UCI, introduciendo distintos niveles de ruido en la clase y los atributos independientemente. Se han considerado los métodos de Chi et al. y PDFC como caso de estudio, analizando la precisión de los modelos creados, en los casos de ruido de clase o de atributos. A partir de los resultados obtenidos, es posible deducir que los Sistemas de Clasificación Basados en Reglas Difusas tienen una buena tolerancia al ruido.

1. Introducción

El problema de la clasificación [2] consiste en realizar generalizaciones a partir de un conjunto de ejemplos de entrenamiento, de forma que el conocimiento aprendido a partir de éstos pueda ser aplicado sobre un conjunto de ejemplos no observados para predecir la clase de los mismos, dadas sus características.

Los factores más determinantes en la precisión de un clasificador son la calidad de los datos de entrenamiento y la capacidad inductiva del algoritmo de aprendizaje. Así, dado un algoritmo de aprendizaje concreto, su precisión en la clasificación dependerá crucialmente de la calidad de los datos de entrenamiento, que a su vez está determinada por un gran número de componentes [16, 15]. Uno de ellos es la fuente de la que provienen los datos y la introducción de los mismos, que están inherentemente sujetas a errores. A pesar de los esfuerzos en solventar este problema, los errores en las grandes bases de datos son comunes y pueden ser graves, y a menos que se tomen medidas extremas para evitarlos, los porcentajes de error pueden alcanzar el 5 % o más [19, 13, 12].

Así pues, las bases de datos reales rara vez son perfectas y a menudo presentan este tipo de corrupciones, que llamamos ruido, y que pueden afectar en la interpretación de los datos, las decisiones tomadas y los modelos creados basados en los mismos, así como en el rendimiento del sistema.

Los Sistemas de Clasificación Basados en Reglas Difusas (SCBRDs) [8, 11] destacan por ser capaces de construir un modelo lingüístico interpretable por seres humanos. Aunque este tipo de sistemas están ampliamente estudiados en la literatura [11], aún no se han comprobado los efectos del ruido en los resultados obtenidos.

El objetivo principal de este estudio es analizar el comportamiento de los SCBRDs cuando los datos de entrenamiento presentan ruido, atendiendo a la precisión del clasificador construido.

Para alcanzar nuestro objetivo, se va a analizar el ruido en dos categorías diferenciadas: ruido de clase y ruido de atributos. Para ello se han creado 138 bases de datos a partir de otras 23 sin ruido ya existentes en el repositorio UCI [1], introduciendo 3 niveles de ruido de clase y atributos independientemente: 5 %, 10 % y 20 %. Se considerarán los SCBRDs de Chi et al. [5] y PDFC [4], estudiando distintas configuraciones, variando para el algoritmo de Chi et al. el número de etiquetas lingüísticas y para PDFC, el tipo de las etiquetas, y observando cómo influyen en la tolerancia al ruido. Por último, analizaremos el impacto del ruido en el rendimiento de los sistemas por separado.

El resto de la contribución está organizado como sigue. En la Sección 2 describimos los SCBRDs que hemos usado. A continuación, en la Sección 3 desarrollamos la descripción de los distintos tipos de ruido considerados en este estudio: el ruido de clase y de atributos. En la Sección 4, mostramos el marco experimental, y los resultados obtenidos junto con su análisis se muestran en la Sección 5. Finalmente, en la Sección 6, señalamos nuestras conclusiones sobre los SCBRDs estudiados cuando tratan con datos con ruido.

2. Sistemas de Clasificación Basados en Reglas Difusas

En esta sección introducimos los detalles específicos de los SCBRDs utilizados en la experimentación de este trabajo: el sistema de generación de reglas de Chi et al. en la Subsección 2.1 y el algoritmo PDFC en la Subsección 2.2.

Cualquier problema de clasificación se compone de m ejemplos de entrenamiento de la forma $x_p = (x_{p1}, \ldots, x_{pn})$, $p = 1, 2, \ldots, m$ donde x_{pi} es el valor del atributo i-ésimo $(i = 1, 2, \ldots, n)$ del p-ésimo ejemplo de entrenamiento. Cada ejemplo está etiquetado con una de M posibles clases.

2.1. Generación de Reglas de Chi et al.

Este método usa reglas difusas de la siguiente forma para el primer SCBRD:

$$\text{Rule } R_j: \text{ If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \text{ then Class} = C_j \text{ with } RW_j \quad (1)$$

donde R_j es la etiqueta de la j-ésima regla, $x = (x_1, \ldots, x_n)$ es un vector n-dimensional de un ejemplo, A_{ji} es un conjunto difuso antecedente, C_j es una etiqueta de clase, y RW_j es el peso de la regla [9].

Los métodos de aprendizaje difusos son la base para construir un SCBRD. El primer algoritmo que comentamos es el método propuesto por Chi et al. en [5, 18]. Para generar la base de reglas difusas, este método determina la relación entre las variables del problema y establece una asociación entre el espacio de características y el espacio de clases por medio de los siguientes pasos:

1. *Establecimiento de las particiones lingüísticas.* Una vez determinado el dominio de cada característica A_i, se calculan las particiones difusas.
2. *Generación de una regla difusa por cada ejemplo* $x_p = (x_{p1}, \ldots, x_{pn}, C_p)$. Para hacer esto es necesario:
   - 2.1 Calcular el grado de emparejamiento $\mu(x_p)$ del ejemplo con las diferentes regiones difusas usando un operador de conjunción.
   - 2.2 Asignar el ejemplo x_p a la región difusa con el mayor grado de pertenencia.
   - 2.3 Generar una regla para el ejemplo, cuyo antecedente está determinado por la región difusa seleccionada y cuyo consecuente es la etiqueta de clase del ejemplo.
   - 2.4 Calcular el peso de la regla.

Durante el aprendizaje, es posible que se generen reglas con el mismo antecedente. En este caso, si tienen la misma clase en el consecuente, entonces borramos una de las reglas duplicadas, pero si es distinta, sólo la regla con el mayor peso se mantiene en la base de reglas.

2.2. Método PDFC (Positive Definite Fuzzy Classifier)

El algoritmo PDFC [4] considera un modelo compuesto de m reglas difusas de la forma:

$$\text{Rule } j: \text{ If } A_j^1 \text{ AND } A_j^2 \text{ AND } \ldots \text{ AND } A_j^n \text{ THEN } b_j \tag{2}$$

donde A_j^k es un conjunto difuso con función de pertenencia $a_j^k : \mathbb{R} \to [0,1]$, $j = 1, \ldots, m$, $k = 1, \ldots, n$, $b_j \in \mathbb{R}$. La asignación de entrada-salida, $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}$ del modelo se define como

$$\mathcal{F}(x_p) = \frac{b_0 + \sum_{j=1}^{m} b_j \prod_{k=1}^{n} a_j^k(x_k)}{1 + \sum_{j=1}^{m} \prod_{k=1}^{n} a_j^k(x_k)}. \tag{3}$$

donde $b_0 \in \mathbb{R}$, las funciones de pertenencia son $a_0^k(x_k) \equiv 1$ para $k = 1, \ldots, n$ y cualquier $x_p \in \mathbb{R}^n$. Luego, el sistema produce un clasificador difuso binario, f, con la regla de decisión

$$f(x_p) = \text{sign}(\mathcal{F}(x_p) + t) \tag{4}$$

donde $t \in \mathbb{R}$ es un umbral. Nosotros asumimos $t = 0$, sin pérdida de generalidad.

Las funciones de pertenencia para el clasificador difuso binario definido arriba se generan a partir de una función de referencia (el tipo de las etiquetas) a^k a través de una transformación de posición [7], así como los clasificadores definidos en las mismas.

La regla de decisión del clasificador binario difuso, entonces, podría escribirse de la forma:

$$f(x_p) = \text{sign}\left(\sum_{j=1}^{m} b_j K(x_p, z_j) + b_0\right) \tag{5}$$

donde $z_j = [z_j^1, z_j^2, \ldots, z_j^n]^T \in \mathbb{R}$ contiene los parámetros de posición de a_j^k. $K : \mathbb{R}^n \times \mathbb{R}^n \to [0,1]$ es una traducción del kernel invariante (kernel Mercer [6]) definido como

$$K(x_p, z_j) = \prod_{k=1}^{n} a^k(x_p^k - z_j^k) \tag{6}$$

Finalmente, la regla de decisión de un clasificador binario difuso es

$$f(x_p) = \text{sign}\left(b_0 + \sum_{j=1}^{m} b_j \prod_{k=1}^{n} a_j^k(x_p^k)\right) \tag{7}$$

Para encontrar las reglas difusas a partir del conjunto de entrenamiento, es necesario construir un kernel Mercer a partir de las funciones de referencia definidas positivas, como se indica en (Ecuación 6). El teorema 3.12 de [4] establece que la regla de decisión de un PDFC puede ser vista como un hiperplano en $\mathbb{F}$. Se emplea entonces el algoritmo SVM [17] para encontrar el hiperplano óptimo y, una vez que se obtiene, las reglas difusas se pueden extraer fácilmente a partir de la regla de decisión del SVM como sigue:

- b_0 es la constante b del hiperplano, $b_0 \leftarrow b$.
- Para cada vector soporte x_i, se crea una regla difusa z_j centrando las funciones de referencia en el vector soporte $z_j \leftarrow x_i$, y se asigna el consecuente de la regla $b_j \leftarrow y_i \alpha_i$, donde α_i es el multiplicador de Lagrange i-ésimo obtenido al resolver el problema de programación cuadrática con SVM, e y_i es la clase del vector soporte i-ésimo.

3. Presencia de Ruido en los Datos

El ruido es un problema que afecta a cualquier base de datos real [23] y puede influir en la precisión del clasificador construido y en el tiempo de construcción y tamaño del mismo, por lo que su interpretabilidad puede verse afectada también. Aunque existen métodos para el filtrado del ruido [3], éstos normalmente no pueden producir datos con iguales características a los originales [20] y, por este motivo, resulta interesante conocer la robustez que ofrecen los métodos de clasificación sin tratamientos externos de los datos ausentes o erróneos.

A continuación, se presentan brevemente los tipos de ruido analizados en esta contribución y los mecanismos usados para la introducción de ruido artificial.

3.1. Ruido de Clase

El ruido de clase se produce cuando la etiqueta de clase de una instancia no es la adecuada, lo que puede ocurrir por ejemplos mal etiquetados o contradictorios (instancias que aparecen

más de una vez con diferentes clases). Existen diversas aproximaciones para mitigar sus efectos [3], principalmente basadas en la eliminación de las instancias con ruido (que suelen mejorar la precisión del clasificador, como se indica en [23]) o incluso en la corrección de las etiquetas erróneas.

Para introducir ruido de clase en bases de datos limpias, adoptamos el esquema indicado en [21] y que se describe a continuación. Dados el par de clases (X, Y), con X la clase mayoritaria e Y la segunda clase con mayor número de instancias, y un nivel de ruido $x\,\%$, una instancia con su etiqueta X tiene una probabilidad de $x\,\%$ de ser incorrectamente etiquetada como Y. Como se indica en [23], este esquema es adecuado debido a que es más probable que sólo cierto tipo de clases sean mal etiquetadas.

3.2. Ruido de Atributos

El manejo de ruido de atributos es más complejo y ha sido menos estudiado en la literatura que el de clases, pero suele estar más presente en las bases de datos reales [20]. En este caso, a diferencia de cómo se suele tratar el ruido de clase, la eliminación de instancias con posible ruido no beneficia a la precisión del clasificador final, como se demostró en [14].

Para introducir el ruido de atributos en bases de datos limpias, seguimos el esquema de [22], en el que los valores erróneos se introducen en cada atributo A_i con un nivel de $x\,\%$, lo cual es consistente con la hipótesis de que las interacciones entre atributos son débiles [23]. Como consecuencia, el ruido introducido en cada atributo tiene una baja correlación con el ruido introducido en el resto. En esta contribución, tratamos únicamente atributos numéricos. Para corromper cada atributo A_i con un nivel de ruido del $x\,\%$, se eligen aproximadamente el $x\,\%$ de los ejemplos de la base de datos y, al valor de A_i de cada uno de esos ejemplos, se le asigna un valor aleatorio entre el máximo y el mínimo del dominio de ese atributo, siguiendo una distribución uniforme. Con este esquema, el porcentaje de ruido en la base de datos puede ser menor que el deseado, ya que algunas veces una asignación aleatoria puede elegir el valor original nuevamente.

4. Marco Experimental

En esta sección, comenzamos mostrando en la Subsección 4.1 las bases de datos reales escogidas para la experimentación. En la Subsección 4.2 damos los detalles correspondientes a la inserción de ruido tanto de clases como de atributos. En la Subsección 4.3, indicamos los algoritmos utilizados para el estudio junto con los parámetros utilizados en su ejecución.

4.1. Bases de Datos

En el Cuadro 1 mostramos las propiedades de las bases de datos seleccionadas para la experimentación. Para cada base de datos, representamos el número de ejemplos (#Ej.), el número de atributos numéricos (#Attrs.) y el número de clases (#Cl.).

Conjunto	**#Ej.**	**#Attrs.**	**#Cl.**
banana	5300	2	2
contraceptive	1473	9	3
ecoli	336	7	8
glass	214	9	7
heart	270	13	2
ionosphere	351	33	2
iris	150	4	3
magic	19020	10	2
monk-2	432	6	2
page-blocks	5472	10	5
penbased	10992	16	10
phoneme	5404	5	2
pima	768	8	2
ring	7400	20	2
satimage	6435	36	7
segment	2310	19	7
sonar	208	60	2
spambase	4597	57	2
thyroid	7200	21	3
twonorm	7400	20	2
wdbc	569	30	2
wine	178	13	3
yeast	1484	8	10

Cuadro 1: Bases de datos utilizadas.

La estimación de la precisión de cada clasificador la obtenemos mediante una validación cruzada de 5 particiones. Dividimos la base de datos en 5 partes con igual número de ejemplos y manteniendo la proporción entre clases. Cada partición es usada como conjunto de test del modelo aprendido con las cuatro particio-

nes restantes. Usamos 5 particiones porque la mayoría de las bases de datos empleadas en la experimentación son pequeñas y así se tiene un número de instancias en test más significativo. Además, la diferencia de usar 5 particiones respecto a un número mayor es que, al tener más instancias cada partición, los efectos del ruido son también más notables en los resultados medios en test y facilitan el análisis de los mismos.

4.2. Perturbación de las Bases de Datos con Ruido

Siguiendo los esquemas de introducción del ruido de las Subsecciones 3.1 y 3.2, a partir de las 23 bases de datos originales sin ruido, se han creado 69 bases de datos con ruido de clase y 69 con ruido de atributos, con niveles de ruido del $x = 5\,\%$, $x = 10\,\%$ y $x = 20\,\%$.

En todas las bases de datos creadas, el ruido se introduce sólo en las particiones de entrenamiento, mientras que los conjuntos de test permanecen inalterados. Esto permite ver cómo afecta el ruido a la precisión de los clasificadores al entrenar con datos con ruido.

Para realizar una estimación de la pérdida (P_x) producida en un clasificador ante la presencia de ruido, utilizamos la siguiente medida:

$$P_x = \frac{Prec_{0\,\%} - Prec_{x\,\%}}{Prec_{0\,\%}} \tag{8}$$

donde $Prec_{x\,\%}$ es la precisión media del clasificador sobre las bases de datos con un nivel de ruido de $x\,\%$.

4.3. Configuración de Parámetros

Los algoritmos han sido ejecutados con la herramienta KEEL [1] [10] utilizando los parámetros mostrados en el Cuadro 2 siguiendo las configuraciones recomendadas por los autores.

Para el método de Chi et al., se ha considerado el uso de diferentes números de etiquetas para ver su influencia en la tolerancia al ruido. Para PDFC, se han utilizado diferentes tipos de etiquetas para ver cómo afecta el ruido al modelo obtenido, ya que este SCBRD ajusta el número de etiquetas automáticamente.

[1] www.keel.es

Algoritmo	*Parámetros*
Chi et al.	Número de etiquetas = 3, 5 y 7 T-norma para cálculo de compatibilidad = Producto Peso de reglas = Factor de certeza penalizado Método de razonamiento difuso = Regla ganadora
PDFC	C = 100 d = 0.25 Tolerancia = 0.001 epsilon = 1.0E-12 Tipo de las etiquetas = Gaussiana y triángulo simétrico Preprocesamiento = Normalización en [0,1]

Cuadro 2: Especificación de parámetros utilizados en la fase de aprendizaje.

5. Análisis de la influencia del ruido en los SCBRDs

En esta sección analizamos la robustez de los dos SCBRDs frente al ruido, estudiando dos medidas: la precisión clásica y la pérdida de precisión al introducir ruido (Ecuación 8). Se desea estudiar la robustez de los modelos inferidos por los SCBRDs considerando la inclusión de ruido en los datos de entrenamiento frente a datos de test sin alterar.

Las tablas de resultados mostradas reflejan los resultados medios en test para las bases de datos correspondientes (las 92 de ruido de clase y las 92 de ruido de atributos). El ruido de clase es analizado en la Subsección 5.1. El caso de presencia de ruido de atributos es analizado en la Subsección 5.2.

5.1. Ruido de Clase

En el Cuadro 3 se muestran los resultados de precisión medios en test obtenidos por el método de cada fila, según el porcentaje de ruido introducido en las bases de datos indicado en cada columna. La columna marcada como (Med.) muestra la media del método en todos los niveles de ruido. En la columna (Método) se indica el método a analizar y, entre paréntesis, el parámetro de la configuración que se varía: en el algoritmo de Chi et al. se indica entre paréntesis el número de etiquetas lingüísticas y, en el algoritmo PDFC, se indica el tipo de las etiquetas: (Tri.) para una función con triángulo simétrico y (Gau.) para una función gaussiana.

Método	Precisión				
	0 %	5 %	10 %	20 %	Med.
Chi (3 térm.)	69.82	69.90	70.18	68.88	69.69
Chi (5 térm.)	66.88	66.61	66.27	64.85	66.15
Chi (7 térm.)	58.40	57.97	57.49	55.90	57.44
PDFC (Gau.)	86.40	85.84	85.23	**83.40**	85.22
PDFC (Tri.)	**88.42**	**87.31**	**86.56**	83.39	**86.42**

Cuadro 3: Precisión en test con ruido en clase al 5 %, 10 % y 20 %

En el Cuadro 4 se muestra la pérdida de precisión para los diferentes niveles de ruido considerados.

Método	Pérdida			
	5 %	10 %	20 %	Med.
Chi (3 térm.)	**0.00**	**-0.01**	**0.01**	**0.00**
Chi (5 térm.)	**0.00**	0.01	0.03	0.01
Chi (7 térm.)	0.01	0.02	0.04	0.02
PDFC (Gau.)	0.01	0.01	0.03	0.02
PDFC (Tri.)	0.01	0.02	0.06	0.03

Cuadro 4: Pérdida en test con ruido en clase al 5 %, 10 % y 20 %

Atendiendo a estos resultados de la experimentación con ruido de clase, se observa que el algoritmo PDFC, en sus dos configuraciones, funciona mejor que el algoritmo de Chi et al. ya que obtiene mejor precisión en todos los casos (con y sin ruido). El mejor método es PDFC empleando etiquetas triangulares.

En el algoritmo de Chi et al., con un menor número de etiquetas lingüísticas se obtiene una mejor toleracia al ruido, tal y como muestra el Cuadro 4, donde la pérdida se acentúa al introducir mayores niveles de ruido. Del mismo modo, en PDFC, el tipo de etiquetas también afecta a la sensibilidad al ruido del método. En algunos casos, como muestra el resultado negativo para el algoritmo Chi (3 térm.) con un 10 % de ruido en el Cuadro 4, algunos algoritmos pueden verse beneficiados al introducir bajos niveles de ruido, ya que puede suceder que los datos que antes podían considerarse como outliers, al variar sus características, caigan dentro de las fronteras de su clase.

Como se ha mencionado, el método PDFC obtiene una mayor precisión en test que el método de Chi et al., a la vez que tiene una mayor pérdida absoluta de precisión al ir incrementando los distintos niveles de ruido de clase. Esto se debe a que PDFC obtiene un modelo que se ajusta mejor a los datos de entrenamiento y el incremento de ruido perjudica más a dicho ajuste que en el caso del método de Chi et al. A pesar de tener más pérdida, PDFC sigue siendo superior en términos de precisión sobre Chi et al. frente al ruido.

En ambos casos, los porcentajes de pérdida de los algoritmos estudiados son muy reducidos y permiten comprobar la gran robustez de los SCBRDs frente al ruido de clase.

5.2. Ruido de Atributos

En el Cuadro 5 se muestran los resultados de precisión y, en el Cuadro 6, la pérdida de precisión al introducir ruido de atributos.

Método	Precisión				
	0 %	5 %	10 %	20 %	Med.
Chi (3 térm.)	69.82	68.83	67.41	63.27	67.33
Chi (5 térm.)	66.88	64.93	61.57	56.39	62.44
Chi (7 térm.)	58.40	55.30	51.07	44.08	52.21
PDFC (Gau.)	86.40	84.82	83.23	80.69	83.79
PDFC (Tri.)	**88.42**	**87.64**	**86.54**	**85.03**	**86.91**

Cuadro 5: Precisión en test con ruido en atributos al 5 %, 10 % y 20 %

Método	Pérdida			
	5 %	10 %	20 %	Med.
Chi (3 térm.)	**0.01**	0.03	0.09	0.05
Chi (5 térm.)	0.03	0.08	0.16	0.09
Chi (7 térm.)	0.05	0.13	0.25	0.14
PDFC (Gau.)	0.02	0.04	0.07	0.04
PDFC (Tri.)	**0.01**	**0.02**	**0.04**	**0.02**

Cuadro 6: Pérdida en test con ruido en clases al 5 %, 10 % y 20 %

En este caso, nuevamente, la configuración PDFC (Tri.) es la que mejores resultados obtiene, seguida por PDFC (Gau.) y las configuraciones de los algoritmos de Chi et al.

En este tipo de ruido, PDFC tiene una mayor robustez, ya que tiene un porcentaje de pérdidas similares con diferentes niveles de ruido. Por el contrario, el método de Chi et al. se ve más afectado respecto al ruido de clases,

con pérdidas de hasta el 0.25 en el caso de Chi (7 térm.), con un 20 % de ruido.

Como en el caso del ruido de clases, un mayor número de etiquetas lingüísticas en el algoritmo de Chi et al., produce una mayor sensibilidad al ruido de atributos y una reducción de la precisión con cada nivel de ruido. Para PDFC las etiquetas triangulares permiten obtener modelos más robustos al ruido que las gaussianas.

En el caso de ruido de atributos el método de Chi et al. se ve más afectado que PDFC. Dado que el método de Chi et al. realiza una partición difusa uniforme en el rango de cada atributo, mientras que PDFC ajusta las etiquetas donde necesita durante el proceso de aprendizaje, este último es más robusto ante este tipo de ruido. Así pues, podemos decir que el tipo de aprendizaje de las etiquetas a partir de los valores de los atributos es importante al tratar con este tipo de ruido.

6. Conclusiones

En esta contribución hemos estudiado la influencia del ruido en los SCBRDs. Para ello, hemos llevado a cabo un análisis empleando los Sistemas de Clasificación Basados en Reglas Difusas formalmente conocidos como de Chi et al. y PDFC. Hemos realizado un análisis desde el punto de vista del ruido de clase y de atributos, perturbando las bases de datos originales con niveles de ruido del $x = 5\,\%$, $x = 10\,\%$ y $x = 20\,\%$.

A partir de los resultados obtenidos y de su análisis, se observa que un mejor ajuste del modelo obtenido, y por tanto una mayor precisión en test, al entrenar con datos sin ruido, supone pérdidas ligeramente mayores en la precisión del método al introducir ruido en las clases. Los porcentajes de pérdida de precisión indican que los SCBRDs son robustos frente al ruido en clases, ya que a niveles altos de este tipo de ruido, este porcentaje no supera el 6 %. Por otro lado, el ruido de clase produce pérdidas de precisión menores que el ruido de atributos, mostrando en los SCBRDs estudiados una buena tolerancia al ruido de este tipo.

Cuando se introduce ruido en atributos, el aprendizaje uniforme de etiquetas por parte de Chi et al. se ve más penalizado que el ajuste individual de las mismas usado por PDFC, que obtiene mejores resultados. Los resultados indican que un aprendizaje más sofisticado de la Base de Conocimiento permite obtener una mayor tolerancia al ruido de atributos.

Se ha observado también la influencia del número de etiquetas lingüísticas en la sensibilidad a ambos tipos de ruido en el caso del algoritmo de Chi et al. y la mayor robustez frente ambos tipos de ruido de las etiquetas triangulares para el algoritmo PDFC. Finalmente, cabe indicar que los SCBRDs son bastante tolerantes a estos dos tipos de ruido según se desprende de los bajos resultados de pérdida de precisión observados.

Es necesario mencionar que éste es un estudio preliminar para dos SCBRD. Como trabajo futuro, es posible incorporar nuevos SCBRDs para hacer una mejor generalización de los comportamientos de los mismos frente al ruido, así como la aplicación de técnicas de filtrado o la inclusión de mecanismos para tratar el ruido en el propio SCBRD.

Agradecimientos

Este trabajo ha sido posible gracias a la subvención del Ministerio de Educación y Ciencia bajo los Proyectos TIN2007-65981 y TIN2008-06681-C06-01. J. Luengo está subvencionado por una beca FPU del Ministerio de Ciencia e Innovación.

Referencias

[1] A. Asuncion and D.J. Newman. *UCI Machine Learning Repository - http://mlr.cs.umass.edu/ml/index.html*. 2007.

[2] C. M. Bishop. *Pattern Recognition And Machine Learning (Information Science And Statistics)*. Springer, 2007.

[3] C.E. Brodley and M.A. Friedl. Identifying mislabeled training data. *Journal of Ar-*

tificial Intelligence Research, 11:131–167, 1999.

[4] Y. Chen and J. Z. Wang. Support vector learning for fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 11(6):716–728, 2003.

[5] Z. Chi, H. Yan, and T. Pham. *Fuzzy algorithms with applications to image processing and pattern recognition*. World Scientific, 1996.

[6] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge Univ. Press, Cambridge, U.K., 2000.

[7] D. Dubois and H. Prade. Operations on fuzzy numbers. *International Journal of Systems Science*, 9(6):613–626, 1978.

[8] T. Nakashima H. Ishibuchi and M. Nii. *Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining*. Springer–Verlag, 2004.

[9] H. Ishibuchi and T. Nakashima. Effect of rule weights in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 9(4):506–515, 2001.

[10] S. García M. J. Otero C. Romero J. Bacardit V. M. Rivas J. C. Fernández J. Alcalá-Fdez, L. Sánchez and F. Herrera. Keel: a software tool to assess evolutionary algorithms form data mining problems. *Soft Computing*, 13(3):307–318, 2008.

[11] L. Kuncheva. *Fuzzy classifier design*. Springer–Verlag, 2000.

[12] J. Maletic and A. Marcus. Data cleansing: Beyond integrity analysis. *Proceedings of the Conference on Information Quality*, 2000.

[13] K. Orr. Data quality and systems theory. *Communications of the ACM*, 41(2):66–71, 1998.

[14] J. R. Quinlan. *C4.5: Programs for Machine Learning*. San Mateo-California: Morgan Kauffman Publishers, 1st edition, 1993.

[15] D. Strong R. Wang and L. Guarascio. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12(4):5–34, 1996.

[16] V. Storey R. Wang and C. Firth. A framework for analysis of data quality research. *IEEE Transactions on Knowledge and Data Engineering*, 7(4):623–639, 1995.

[17] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, U.S.A., 1998.

[18] L. Wang and J. M. Mendel. Generating fuzzy rules by learning from examples. *In Proceedings of the 1991 IEEE International Symposium on Intelligent Control, Arlington, Virginia, U.S.A.*, pages 263–268, 1991.

[19] X. Wu. *Knowledge Acquisition from Databases*. Ablex Pulishing Corp., 1995.

[20] X. Wu and X. Zhu. Mining with noise knowledge: Error-aware data mining. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 38(4):917–932, 2008.

[21] X. Wu X. Zhu and S. Chen. Eliminating class noise in large datasets. *Proceedings of the 20th ICML International Conference on Machine Learning, Washington D.C.*, page 920–927, 2003.

[22] X. Wu X. Zhu and Y. Yang. Error detection and impactsensitive instance ranking in noisy datasets. *In Proceedings of 19th National conference on Artificial Intelligence (AAAI-2004), San Jose, CA.*, 2004.

[23] X. Zhu and X. Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210, 2004.

ÍNDICE DE AUTORES

Acampora, G., 231
Albusac, Javier, 3
Alcalá, Jesús, 425
Aldámiz-Echevarría, L., 107
Alonso, Jose M., 475
Alonso, S., 51
Alonso Fernández, Daniel, 273
Alonso Martínez, Israel, 401
Álvarez, Marcos, 327
de Andrés Rocío, 49
De Baets, B., 157
Barrena, R., 107
Barrenechea E., 131, 165
Berlanga, Antonio, 123
Brodeala, Luchiana C., 393
Buenestado, D., 107
Bustince, H., 157
Caballero, Jorge, 319
Cabrerizo, F. J., 51
Cadenas, J. M., 231, 343
Cano, Isaac, 427
Caro, Raquel, 287
Casado, F., 165
Castillo-Ortega, Rita, 451
Castiñeira, Elena E., 189
Castro, J. L., 27
Castro-Schez, J. J., 3
Cobo, M.J., 409
Conci, A., 173
Contreras, David, 377
Couso, Inés, 435, 443
Cubillo, Susana, 189
Delgado, M., 27
Díaz, Luis, 19
Díaz Agrela, Diana, 311
Díaz Vicedo, Concepción, 247
Doña, J. M, 35
Espinilla, Macarena, 49, 67
Esteve, Pablo, 303
Falcó Díaz de Cerio, Edurne, 83
Fernández, Alberto, 459
Fernández, Francisco Javier, 131
Fernández, J., 157
Fernández. Llamas, Camino, 11
Ferreira-Satler, Mateus, 295, 385
Fuentes-González, Ramón, 149
García, J. J., 107
García, Juan F., 11
García Lapresta, José Luis, 83, 115
Garmendia, Alfonso, 287, 311
Garmendia, Luis, 271, 281, 311
Garrido, M. C., 343
Godofredo, José Luis, 255
Gómez Ruiz, J, 35
Gómez-Romero, Juan, 123
González-Hidalgo, M., 141
Gramajo, Sergio, 59
Herrera, Francisco, 459, 467
Herrera-Viedma, Enrique, 51, 361, 409
Iruetaguena, A., 107
Jacas Moral, Joan, 239
Jiménez, L, 41, 133
Jiménez-Linares, L., 3
Jurio, A., 157
Lamata, M. T., 335
de Lara, Jesús, 327
León, Teresa, 263
Liern, Vicente, 229, 255, 263
Liu, J., 91
Loia, V., 231
López, Antonio Gabriel, 351
López, Victoria, 271, 273, 303, 319, 327, 459
López-Herrera, A.G., 409
López-Juárez, Pedro, 417
López-Molina, C., 157
Luengo, Julián, 467
Llamazares, Bonifacio, 115, 205
Magdalena, Luis, 475
Marín, Nicolás, 451
Martín Bonilla, Raúl, 281
Martin-Bautista, Maria J., 393
Martínez, Francisco J., 67
Martínez, Juan Carlos, 75
Martínez, Luis, 59, 67,179
Martínez, R., 343
Mas, M., 221
Massanet, Sebastià, 141, 213
Mata, Francisco, 75
Matellán, Vicente, 11
Medina, J., 27, 197
Medina, Rafael, 19
de Mendoza, Juan C., 303
Menéndez, Víctor H., 295, 369
Merlo, Roberto, 353, 377
de Miguel D., 173
Molina, José M., 123
Monreal Pujadas, Amadeo, 239

Monserrat, M., 221
Montero, Javier, 183
Montilla, Wilmer, 189
Moreno-García, Juan, 41, 99, 133
Mucientes Manuel, 1
Muñoz, E., 231
Muñoz, Rafael, 19
Navarro-Arribas, Guillermo, 427
Nunes, E., 173
Olivas, José Ángel, 295, 353, 385, 417
Ortega, Raúl, 319
Otero, José, 435
Pablo Jimeno, 319
Padellano, Celia, 327
Pagola, M., 131, 165
Palacios, Ana, 443
Pascual Romero, Francisco, 401
Paternain, D., 165
Peláez, J. I., 35
Pelta, D. A., 335
Peralta, Arturo 99
Pérez Asurmendi, Patrizia, 115
Pérez, I. J., 51
Pikatza, J. M., 107
Porcel, Carlos, 351, 361
Prieto, Manuel E., 295, 369
De Prisco, R., 231
Puente, Cristina, 353, 377
Ramos Gómez, Ángel, 123
Rdez. Lera, Francisco J., 11
Rodríguez, R.M., 91
Rodriguez-Benítez, L., 41, 133
Romero, Francisco P., 99, 351, 385
Ruan, D., 91
Ruiz-Lozano, M.D., 27
Sáez, José A., 467
Salvador, Adela, 287
Sánchez, A., 173
Sánchez, Daniel, 425, 451
Sánchez, Luciano, 435, 443
Sanjurjo, P., 107
Santos, Matilde, 273
Segundo, U., 107
Serrano-Guerrero, Jesús, 351, 385
Sobrino, Alejandro, 353
Solana-Cipres, C., 41, 133
Tejeda, Álvaro, 361
Tenorio, E., 35
Tinguaro Rodriguez, J., 183
Torra, Vicenç, 427
Torrens, Joan, 141, 221, 213
Vallejo, David, 3
Verdegay, J. L., 335
Vidal, Christian L., 295
Vitoriano, Begoña, 183
Yeguas, Enrique, 19
Zaccagnino, R., 231
Zapata, Alfredo, 295
Zuccarello, Pedro, 247