

CO²RBFN frente a otros métodos de diseño de RNAs en problemas de clasificación no balanceada

M. D. Pérez-Godoy A. Fernández A. J. Rivera M. J. del Jesus

lperez@ujaen.es

ahilario@ujaen.es

arivera@ujaen.es

mjjesus@ujaen.es

Departamento de Informática

Universidad de Jaén

Resumen

En los problemas de clasificación reales ocurre con frecuencia que las clases no están equilibradas, de forma que aparecen muchos más ejemplos de una clase de que otra; y suele ocurrir que la clase minoritaria es la clase de interés. La mayoría de los métodos de clasificación, en su intento de obtener una buena precisión, se centran en la clase mayoritaria. Como soluciones al desbalanceo de clases se presentan las aproximaciones internas, a nivel de algoritmo, y las aproximaciones externas, a nivel de datos, que pre-procesan éstos para intentar equilibrar las clases. En este trabajo se va a comparar CO²RBFN, un algoritmo cooperativo-competitivo para el diseño de Redes de Funciones de Base Radial, con otros métodos de diseño de redes neuronales. La comparativa se va a realizar cuando los métodos se aplican sobre los datos originales y cuando se aplican con pre-procesamiento mediante la técnica SMOTE (*Synthetic Minority Over-sampling Technique*).

1. Introducción

En el mundo real son muchos los conjuntos de datos en los que las distribuciones de las diferentes clases no están balanceadas, por lo que este problema es muy significativo [28]. Este problema se presenta en aplicaciones tales como clasificación de imágenes de satélite [24], análisis de riesgos [14], clasificación de datos de tele-observación [7] y especialmente

en aplicaciones médicas [20]. Es importante resaltar que la clase minoritaria representa la mayoría de las veces el concepto de interés y por tanto es donde reside el conocimiento más novedoso.

El problema en la clasificación de bases de datos no balanceadas es que los algoritmos de clasificación tienen tendencia a describir la clase mayoritaria. Esto ocurre puesto que el clasificador intenta reducir el error global, y éste no tiene en cuenta la distribución de los datos.

Las Redes de Funciones de Base Radial (RBFNs) constituyen uno de los paradigmas más populares dentro del campo de las redes neuronales. Este modelo de cómputo ha demostrado su solvencia a la hora de abordar problemas como aproximación de funciones [13], clasificación [8] o predicción de series temporales [25].

Son muchas las características que despertan el interés por este tipo de redes entre las cuales cabe destacar: su topología simple con tan solo una capa oculta y su capacidad de ser un aproximador universal [18].

En el trabajo actual aplicaremos nuestro algoritmo, CO²RBFN, un algoritmo cooperativo-competitivo para diseño de RBFNs, a la clasificación en bases de datos no balanceadas y compararemos los resultados obtenidos con los de otros métodos de diseño de redes neuronales. La organización del resto de este trabajo se describe a continuación. En la Sección 2 se comentan los métodos que intentan solucionar el problema de desbalanceo de clases. En la Sección 3 se

describen las RBFNs y se explica cómo se estudian aplicaciones de éstas al problema de clasificación no balanceada. En la Sección 4 se presenta el algoritmo propuesto, CO²RBFN, y sus resultados se muestran en la Sección 5. Por último se explican las conclusiones alcanzadas y las líneas de trabajo futuro en la Sección 6.

2. Soluciones al desbalanceo

El problema del desbalanceo, considerando dos clases, ocurre cuando el número de instancias de una de las clases es mucho mayor que el de la otra.

En este trabajo nos vamos a centrar en bases de datos desbalanceadas binarias, formadas por ejemplos de dos clases: positiva y negativa. El nivel de desbalanceo se mide a través del indicador IR (*Imbalanced Ratio*), que se define como la proporción del número de instancias entre la clase mayoritaria y la minoritaria.

Existen distintas aproximaciones que intentan manejar los problemas asociados con las bases de datos no balanceadas. Dichos enfoques se pueden reunir en dos grupos según su relación con el clasificador [6]:

- aproximaciones internas, tratan de crear algoritmos de clasificación nuevos o modificar los existentes de forma que sean éstos los que tengan en cuenta el desbalanceo de clases [5, 27].
- aproximaciones externas al clasificador que pre-procesan los datos para poder disminuir el efecto causado por el desbalanceo de las clases [9, 6, 10]. Entre éstas se incluyen técnicas de bajo-muestreo, sobre-muestreo y mixtas.

En este trabajo nos vamos a centrar en el uso del método SMOTE como método de sobre-muestreo. SMOTE forma nuevos ejemplos de la clase minoritaria mediante interpolación de ejemplos de dicha clase que están próximos.

Una de las medidas más utilizadas en problemas de clasificación no balanceada para

evaluar la precisión de los métodos es la media geométrica (MG), definida como [5]:

$$MG = \sqrt{\frac{VP}{VP + FN} \frac{VN}{FP + VN}} \quad (1)$$

donde VP , VN , FP y FN representan los Verdaderos Positivos, Verdaderos Negativos, Falsos Positivos y Falsos Negativos respectivamente. Esta métrica trata de maximizar la precisión de ambas clases, la mayoritaria y la minoritaria.

3. RBFNs

Una Red de Funciones de Base Radial es un tipo de red neuronal hacia delante con tres capas: la capa de entrada con n nodos, una capa oculta con m neuronas o RBFs, y una capa de salida que en el caso de predicción presenta sólo un nodo (1).

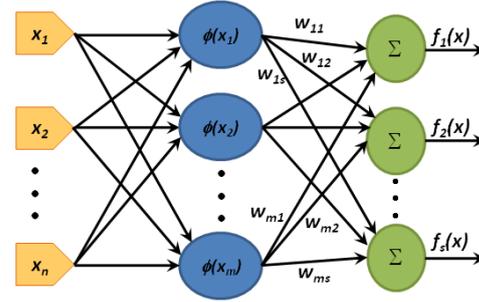


Figura 1: Topología de una Red de Funciones de Base Radial

Las m neuronas/RBFs de la capa oculta ofrecen una activación simétrica radial $\phi_i : R^n \rightarrow R$, que puede tomar diferentes formas aunque la más común es la función gaussiana que viene dada por la expresión:

$$\phi_i(\vec{x}) = \phi(\|\vec{x} - \vec{c}_i\| / r_i) \quad (2)$$

La función tiene una salida simétrica, y se incrementa (o decrecienta) de forma monótona con respecto a un centro, $\vec{c}_i \in R^n$. Tiene un factor de escala, $r_i \in R$, conocido como radio y el operador $\|\cdot\|$ se suele tomar como la distancia Euclídea en R^n .

Los nodos de la capa de salida implementan la ecuación 3:

$$f(\vec{x}) = \sum_{i=1}^m w_{ij} \phi_i(\vec{x}) \quad (3)$$

Las técnicas empleadas en el diseño de RBFNs son muy diversas. El algoritmo típico de diseño de RBFNs tiene dos etapas. En la primera etapa se determinan los centros y los radios de las RBFs, mientras que en la segunda etapa se calculan sus pesos. Para determinar los centros y los radios se pueden emplear técnicas de clustering [19]. En la segunda etapa y para calcular los pesos se pueden utilizar algoritmos como Least Mean Square (LMS) [26], Singular Value Decomposition (SVD) [12], etc.

El diseño de una RBFN también se puede abordar desde el punto de vista de la computación evolutiva [4]. En la mayoría de estas propuestas evolutivas, un individuo representa una RBFN completa. De esta manera, los operadores evolutivos actúan sobre los individuos añadiendo RBFs, eliminándolas o modificándolas.

Sin embargo y según el trabajo de Potter [22] la computación evolutiva tradicional presenta ciertos problemas relacionados con la evaluación de subcomponentes independientes en el diseño de ciertos modelos. Así se propone la coevolución cooperativa para extender el modelo evolutivo tradicional y conseguir un entorno de diseño en el que los individuos de la población representen una parte de la solución evolucionando en paralelo. Ahora los individuos no sólo compiten por sobrevivir sino que deben cooperar para alcanzar una solución.

En la bibliografía especializada se pueden encontrar distintos métodos evolutivos [8, 15, 17] para el diseño de RBFNs. Sin embargo, la mayoría de las aproximaciones existentes trabajan con métodos evolutivos típicos donde un individuo representa una red completa, por lo que pueden tener problemas de alto coste computacional y convergencia prematura a mínimos locales. Como se ha comentado anteriormente estos problemas pueden mitigarse utilizando técnicas cooperativas-competitivas

donde un individuo representa una neurona o RBF tal y como se propone en este trabajo. Hasta ahora en la bibliografía no son muchos los trabajos [25, 23, 21] que implementan este tipo de paradigma, debido sobre todo a la dificultad que entraña definir los parámetros de cooperación y competición entre los individuos.

En la bibliografía, no son muchas las propuestas de diseño de RBFNs en las que se aborde el problema de clasificación en bases de datos no balanceadas, mediante RBFNs. En [3] se utiliza una función de coste en el proceso de entrenamiento para reducir el impacto del desbalanceo. Otras propuestas utilizan aproximaciones externas, en [1] se realiza un estudio para evaluar el efecto del tamaño del conjunto de datos de entrenamiento y del índice de desbalanceo, sobre la precisión en la clasificación de microalgas.

4. Descripción de CO²RBFN

CO²RBFN [21] es un método evolutivo cooperativo-competitivo para el diseño de RBFNs. En esta propuesta cada individuo de la población representa una función base y la población entera es la responsable de la solución final. Se presenta un entorno en el que los individuos cooperan para alcanzar la solución definitiva, no obstante, también compiten por la supervivencia, dado que si el trabajo de un individuo no es bueno dicho individuo será eliminado. En este entorno de cooperación-competición en el que la solución depende del comportamiento de muchos componentes, el fitness de cada individuo es conocido como asignación de crédito. Para medir la asignación de crédito de un individuo, se proponen tres factores que evalúan el papel de cada RBF en la red. Dichos factores tienen en cuenta la aportación de la RBF a la red, el error cometido por la RBF y la posible interacción de una RBF con otras. El algoritmo usa un Sistema Basado en Reglas Difusas (SBRD) para generar la probabilidad de aplicación de los operadores a las RBFs. Se utiliza un esquema de codificación real. Cada cromosoma representa

el centro (con un número de coordenadas igual al número de características de los ejemplos) y el radio de una RBF. Las etapas principales de CO²RBFN se explican en las siguientes subsecciones. A continuación se muestra su pseudocódigo (2):

```

Inicialización de la RBFN
Mientras (No Fin) Hacer
  Entrenamiento de la RBFN
  Evaluación de las RBFs
  Aplicación de los operadores
    a las RBFs
  Introducción de nuevas RBFs
Fin_Mientras

```

Figura 2: Principales pasos de CO²RBFN

Inicialización de la Red El proceso para definir la red inicial es simple. Las neuronas se colocan aleatoriamente sobre muestras del conjunto de entrenamiento. El número de RBFs está especificado como un parámetro (es el tamaño de la población, m). El centro de cada RBF, ϕ_i , se inicializa a partir de un patrón al azar del conjunto de entrenamiento. El radio, r_i , se inicializa a la mitad de la media de las distancias entre los centros. Finalmente los pesos, w_{ij} , se ponen a cero.

Entrenamiento de las RBFs Durante esta etapa se entrenan los pesos de las RBFs. El propósito de entrenar los pesos es el de explotar la información local extraída del comportamiento de las RBFs. Dicho entrenamiento se realiza mediante la técnica LMS [26].

Evaluación de las RBFs Se requiere un mecanismo de asignación de crédito para poder evaluar el papel de cada función base dentro del entorno cooperativo-competitivo. Para este propósito se consideran tres parámetros, a_i , e_i , s_i para cada RBF ϕ_i .

La contribución, a_i , de la RBF ϕ_i , $i = 1 \dots m$, a la salida de la RBFN, se determina considerando su peso, w_i , y el número de patrones de entrenamiento dentro de su radio, pi_i . Se penaliza una RBF con poco peso y pocos patrones dentro de su radio:

$$a_i = \begin{cases} |w_i| & \text{si } pi_i \geq q \\ |w_i| * (pi_i/q) & \text{en otro caso} \end{cases} \quad (4)$$

donde q es la media de los valores pi_i menos la desviación típica de dichos valores pi_i y w_i es la coordenada máxima de su vector de pesos, dado que es la que representa la clase que determina la RBF.

Error, e_i , que comete la RBF. Este parámetro, se obtiene, como la proporción de patrones mal clasificados dentro del radio de la RBF. La ecuación 5 muestra el cálculo del error.

$$e_i = \frac{pimc_i}{pi_i} \quad (5)$$

El solapamiento de la RBF, ϕ_i , con otras RBFs se cuantifica usando el parámetro s_i . Este parámetro se calcula partiendo de la base de la metodología fitness sharing [11], que intenta mantener la diversidad en la población. El factor se expresa como:

$$s_i = \sum_{j=1}^m s_{ij}$$

$$s_{ij} = \begin{cases} (1 - \|\phi_i - \phi_j\|/r_i) & \text{si } \|\phi_i - \phi_j\| < r_i \\ 0 & \text{en otro caso} \end{cases} \quad (6)$$

donde s_{ij} mide el solapamiento entre las RBF ϕ_i y ϕ_j , $j = 1 \dots m$.

Aplicación de los operadores a las RBFs Se pueden aplicar cuatro operadores a las RBFs:

- Operador Elimina: es un operador que elimina una RBF.
- Operador Mutación Aleatoria: modifica de forma aleatoria el centro y el radio de la RBF. El radio se modifica con

una probabilidad inversamente proporcional al número de características en el problema (n). La modificación, en caso de llevarse a cabo, puede oscilar en un porcentaje entre un 5% y un 50% sobre el radio antiguo. En cuanto a la modificación del centro, el número de coordenadas que se van a mutar se obtiene aleatoriamente entre un 1% y un 25% del total de características del conjunto de entrenamiento. La coordenada a mutar sufre un incremento o decremento en una cantidad aleatoria comprendida entre un 5% y un 50% de valor del radio.

- **Operador Mutación Informada:** este operador modifica el radio y todas las coordenadas del centro de la RBF, usando información de su entorno. Las coordenadas del centro se modifican para intentar situarlo en el centro de los patrones que son de la clase que la RBF está representando. El objetivo de modificar el radio es intentar que la mayoría de los patrones que son de la clase que está representando la RBF, queden dentro de su radio.
- **Operador Nulo:** no se realiza ninguna acción sobre la RBF.

Estos operadores se aplican a la población total de RBFs. La probabilidad de elegir un operador para una RBF dada viene determinada por un sistema difuso tipo Mamdani [16], cuyas entradas son los parámetros a_i , e_i y s_i , usados para medir la asignación de crédito de cada RBF. Las salidas del sistema, $p_{elimina}$, p_{ma} , p_{mi} y p_{nulo} , representan la probabilidad de aplicación de los operadores Elimina, Mutación aleatoria, Mutación informada y Nulo, respectivamente.

La Tabla 1 muestra las reglas usadas para relacionar los antecedentes y los consecuentes. En la tabla cada fila representa una regla y así por ejemplo la interpretación de la primera regla es: Si la aportación de una RBF es baja Entonces la probabilidad de aplicar el operador elimina es medio-alta, la del operador de mutación aleatoria es medio-alta, la del operador mutación informada es

baja y la del operador nulo es baja. En el diseño de reglas se tiene en cuenta el hecho de que una RBF es peor si su contribución (a_i) es baja, su error (e_i) es alto y su solapamiento (s_i) también es alto. En el lado opuesto, una RBF es mejor cuando su contribución es alta y su error y solapamiento son bajos. De esta forma, la probabilidad de eliminar una RBF se incrementa cuando ésta tiene un mal comportamiento; la probabilidad de mutar va aumentando a medida que el comportamiento es mejor. El operador nulo se aplicará cuando el comportamiento de una RBF sea muy bueno.

Cuadro 1: Base de reglas difusas que representan conocimiento experto en el diseño de RBFNs

	Antecedentes			Consecuentes			
	v_a	v_e	v_s	$v_{p_{elimina}}$	$v_{p_{ma}}$	$v_{p_{mi}}$	$v_{p_{nulo}}$
R1	B			M-A	M-A	B	B
R2	M			M-B	M-A	M-B	M-B
R3	A			B	M-A	M-A	M-A
R4		B		B	M-A	M-A	M-A
R5		M		M-B	M-A	M-B	M-B
R6		A		M-A	M-A	B	B
R7			B	B	M-A	M-A	M-A
R8			M	M-B	M-A	M-B	M-B
R9			A	M-A	M-A	B	B

Introducción de nuevas RBFs En este punto el algoritmo sustituye las RBFs eliminadas por otras nuevas. Existen dos alternativas para situar las nuevas RBFs, ambas con probabilidad de 0.5: en un patrón dentro la zona donde se esté cometiendo el máximo error y que esté fuera de cualquier RBF o bien, en un patrón obtenido de forma aleatoria que no esté dentro del radio de ninguna RBF.

Selección de las mejores RBFs Después de aplicar los operadores de mutación aparecen nuevas RBFs y éstas son comparadas con sus padres para determinar cuáles tienen un mejor comportamiento en la red. Las mejores RBFs serán las elegidas para formar parte de la nueva población.

Cuadro 2: Resultados experimentales sin pre-procesamiento

IR	Base datos	CO ² RBFN	LVQ	MLP-Back	MLP-Grad	RBFN-Decr	RBFN-Incr
1.82	glass1	69.30 ± 6.16	62.27 ± 11.19	56.32 ± 10.53	71.28 ± 4.47	67.99 ± 10.07	73.23 ± 8.10
1.86	ecoli0vs1	97.03 ± 2.80	93.68 ± 3.87	97.59 ± 1.99	0.00 ± 0.00	94.71 ± 4.23	92.93 ± 8.40
1.86	wisconsin	97.29 ± 0.77	92.65 ± 5.78	96.23 ± 1.15	93.89 ± 2.23	95.31 ± 1.88	95.55 ± 1.89
6.38	glass6	87.07 ± 7.38	85.75 ± 8.04	90.40 ± 6.42	88.60 ± 11.01	88.12 ± 8.95	91.23 ± 6.60
8.11	yeast3	89.51 ± 2.58	67.78 ± 18.86	74.57 ± 5.51	81.82 ± 6.51	32.65 ± 24.84	67.40 ± 14.61
8.19	ecoli3	87.02 ± 7.65	74.24 ± 19.72	59.14 ± 33.33	54.32 ± 29.03	66.64 ± 23.66	68.43 ± 29.52
32.78	yeast5	94.12 ± 4.40	76.93 ± 23.64	59.79 ± 26.63	63.98 ± 14.45	35.99 ± 29.21	44.80 ± 23.77
39.15	ecoli0137vs26	70.50 ± 29.50	52.16 ± 46.90	61.92 ± 35.45	58.83 ± 48.37	68.70 ± 40.14	69.50 ± 40.62
39.15	yeast6	83.27 ± 10.45	64.85 ± 28.28	60.33 ± 16.06	53.23 ± 20.1	15.33 ± 25.68	28.34 ± 30.40
128.87	abalone19	50.12 ± 21.81	11.42 ± 20.60	49.88 ± 13.87	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	Media	82.52 ± 9.35	68.17 ± 18.69	70.62 ± 15.09	56.59 ± 13.62	56.54 ± 16.87	63.14 ± 16.39

Cuadro 3: Resultados experimentales con SMOTE

IR	Base datos	CO ² RBFN	LVQ	MLP-Back	MLP-Grad	RBFN-Decr	RBFN-Incr
1.82	glass1	69.86 ± 6.44	65.33 ± 7.63	57.53 ± 7.14	71.93 ± 5.98	67.95 ± 14.41	73.78 ± 8.91
1.86	ecoli0vs1	96.18 ± 2.96	93.05 ± 3.63	97.22 ± 1.70	0.00 ± 0.00	94.69 ± 4.71	95.13 ± 5.26
1.86	wisconsin	97.26 ± 0.85	94.02 ± 5.18	90.52 ± 17.72	95.31 ± 1.57	94.41 ± 4.95	94.16 ± 5.07
6.38	glass6	85.93 ± 8.39	85.86 ± 7.19	88.05 ± 7.17	85.19 ± 9.20	88.16 ± 8.22	85.52 ± 9.05
8.11	yeast3	91.11 ± 2.34	82.92 ± 4.88	82.96 ± 17.18	91.65 ± 2.47	56.92 ± 29.97	87.77 ± 4.56
8.19	ecoli3	85.72 ± 7.90	82.27 ± 6.06	86.65 ± 6.09	68.36 ± 34.94	78.96 ± 20.01	86.26 ± 9.65
32.78	yeast5	94.69 ± 3.60	94.86 ± 2.07	91.67 ± 6.02	93.17 ± 5.09	40.41 ± 43.18	94.54 ± 3.96
39.15	ecoli0137vs26	70.09 ± 36.30	66.41 ± 34.27	54.87 ± 45.05	57.69 ± 47.13	62.74 ± 38.02	61.16 ± 43.29
39.15	yeast6	86.57 ± 8.40	76.41 ± 10.18	85.78 ± 5.61	84.59 ± 7.71	34.09 ± 36.30	83.47 ± 9.04
128.87	abalone19	70.18 ± 11.77	52.86 ± 15.69	60.46 ± 14.83	51.87 ± 11.45	54.47 ± 14.09	63.32 ± 12.10
	Media	84.76 ± 8.90	79.40 ± 9.68	79.57 ± 12.85	69.98 ± 12.55	67.28 ± 21.39	82.51 ± 11.09

5. Experimentación y Resultados

En este estudio CO²RBFN se compara con otros 5 métodos de diseño de redes neuronales: LVQ, MLP-Back, MLP-Grad, RBFN-Decr y RBFN-Incr, disponibles a través de la herramienta KEEL [2]. Los métodos se van a aplicar a 10 bases de datos con diferente nivel de desbalanceo. Para realizar el estudio comparativo se utiliza validación cruzada de orden 5 y se utiliza la media geométrica como medida para evaluar la precisión de los métodos en la clasificación.

El estudio se realiza desde dos perspectivas, primero se compararán los métodos cuando se aplican sobre las bases de datos originales y después se compararán cuando se aplican sobre las bases de datos pre-procesadas mediante SMOTE.

En la Tabla 2 se muestran los resultados obtenidos por los métodos cuando se han aplicado sobre los datos originales, sin haberles realizado ningún tipo de pre-procesamiento.

Como se puede observar CO²RBFN, obtiene los mejores resultados medios cuando los

métodos se han aplicado sobre los datos originales. Supera al resto de métodos en 7 bases de datos, sobre todo esta superioridad se observa en bases de datos con medio y alto IR.

En la Tabla 3 se muestran los resultados de los métodos aplicados sobre los datos pre-procesados mediante SMOTE. De nuevo CO²RBFN vuelve a obtener los mejores resultados medios. Supera al resto en 4 bases de datos de las cuales 3 tienen un valor alto de IR.

El comportamiento de CO²RBFN es robusto con cualquier IR mientras que en el resto de métodos el descenso de la precisión es más significativo a medida que aumenta el IR en las bases de datos.

También se puede observar que CO²RBFN tiene un buen comportamiento tanto cuando trabaja sobre los datos originales como sobre los datos pre-procesados. Aunque su eficiencia aumenta al pre-procesar los datos, dicho aumento es mucho mayor en el resto de métodos.

6. Conclusiones

En este trabajo se ha estudiado el comportamiento de CO²RBFN frente a otros modelos de diseño de redes neuronales, en la clasificación en bases de datos no balanceadas con diferente IR.

CO²RBFN es un método que obtiene buenos resultados cuando se aplica a bases de datos no balanceadas. Su comportamiento mejora cuando se aplica sobre los datos pre-procesados pero es menos sensible al pre-proceso de datos que el resto de métodos. En la comparativa con el resto de métodos obtiene los mejores resultados tanto en la aplicación sobre los datos originales como sobre los datos pre-procesados mediante SMOTE.

Como una línea de posible trabajo futuro está centrarnos en estudiar aproximaciones internas que aborden el desbalanceo desde dentro del algoritmo.

Agradecimientos

Este trabajo está soportado por la Comisión Interministerial de Ciencia y Tecnología (CI-CYT) con el proyecto TIN2008-06681-C06-02 y por el Plan de Investigación Andaluz TIC-3928.

Referencias

- [1] L. Al-Haddad, C.W. Morris, and L. Boddy. Training radial basis function neural networks: effects of training set size and imbalanced training sets. *Journal of Microbiological Methods*, 43:33–44, 2000.
- [2] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. Del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J.C. Fernández, and F. Herrera. Keel: A software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13(3):307–318, 2009.
- [3] R. Alejo, V. García, J.M. Sotoca, R.A. Mollineda, and J.S. Sánchez. Improving the performance of the rbf neural networks trained with imbalanced samples. In *Proceedings of the 9th International Work-Conference on Artificial Neural Networks (IWANN'07)*, volume 4507, pages 162–169, 2007.
- [4] T. Bäck, U. Hammel, and H. Schwefel. Evolutionary computation: comments on the history and current state. *IEEE Transactions Evolutionary Computation*, 1(1):3–17, 1997.
- [5] R. Barandela, J.S. Sánchez, V. García, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851, 2003.
- [6] G.E.A.P.A. Batista, R.C. Prati, and M.C. Monard. A study of the behaviour of several methods for balancing machine learning training data. *SIGKDD Explorations*, 6(1):20–29, 2004.
- [7] L. Bruzzone and S.B. Serpico. Classification of imbalanced remote-sensing data by neural networks. *Patterns Recognition Letters*, 18(11–13):1323–1328, 1997.
- [8] O. Buchtala, M. Klimek, and B. Sick. Evolutionary optimization of radial basis function classifiers for data mining applications. *IEEE Transactions on System, Man, and Cybernetics, B*, 35(5):928–947, 2005.
- [9] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligent Research*, 16:321–357, 2002.
- [10] A. Estabrooks, T. Jo, and N. Japkowicz. A multiple resampling method for learning from imbalanced data-sets. *Computational Intelligence*, 20(1):18–36, 2004.
- [11] D. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49, 1987.

- [12] G. Golub and C. Van Loan. *Matrix computations*. Hopkins University Press. 3rd edition, 1996.
- [13] J. González, I. Rojas, J. Ortega, H. Pomares, J. Fernández, and A.F. Díaz. Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. *IEEE Transactions on Neural Networks*, 14(6):1478–1495, 2003.
- [14] Y. M. Huang, C. M. Hung, and H. C. Jiau. Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem. *Nonlinear Analysis: Real World Applications*, 7(4):720–747, 2006.
- [15] E. Lacerda, A. Carvalho, A. Braga, and T. Ludermir. Evolutionary radial functions for credit assessment. *Applied Intelligence*, 22:167–181, 2005.
- [16] E. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13, 1975.
- [17] R. Neruda and P. Kudová. Learning methods for radial basis function networks. *Future Generation Computer Systems*, 21(7):1131–1142, 2005.
- [18] J. Park and I. Sandberg. Universal approximation using radial-basis function networks. *Neural Computation*, 3(2):246–257, 1991.
- [19] W. Pedrycz. Conditional fuzzy clustering in the design of radial basis function neural networks. *IEEE Transactions on Neural Networks*, 9(4):601–612, 1998.
- [20] X. Peng and I. King. Robust BMPM training based on second-order cone programming and its application in medical diagnosis. *Neural Networks*, 21(2–3):450–457, 2008.
- [21] M. D. Pérez-Godoy, A. J. Rivera, F. J. Berlanga, and M. J. del Jesus. Co²rbfn: An evolutionary cooperative-competitive rbf design algorithm for classification problems. *Soft Computing*, 14(9):953–971, 2010.
- [22] M. Potter and K. De Jong. Cooperative coevolution: an architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.
- [23] A.J. Rivera, I. Rojas, J. Ortega, and M.J. del Jesus. A new hybrid methodology for cooperative-coevolutionary optimization of radial basis function networks. *Soft Computing*, 11(7):655–668, 2007.
- [24] S. Suresh, Narasimhan Sundararajan, and Paramasivan Saratchandran. Risk-sensitive loss functions for sparse multi-category classification problems. *Information Sciences*, 178(12):2621–2638, 2008.
- [25] B. Whitehead and T. Choate. Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction. *IEEE Transactions on Neural Networks*, 7(4):869–880, 1996.
- [26] B. Widrow and M.A. Lehr. 30 years of adaptive neural networks: perceptron, madaline and backpropagation. In *Proceedings of the IEEE*, volume 78, pages 1415–1442, 1990.
- [27] L. Xu, M.Y. Chow, and L.S. Taylor. Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification e-algorithm. *IEEE Transactions on Power Systems*, 22(1):164–171, 2007.
- [28] Q. Yang and X. Wu. 10 challenging problems in data mining research. *International Journal of Information Technology and Decision Making*, 5(4):597–604, 2006.