

Regresión logística multiclase utilizando funciones de base evolutivas de tipo proyección

C. Hervás, P.A. Gutiérrez, J.C. Fernández
Departamento de Informática y Análisis Numérico
Universidad de Córdoba
14071-Córdoba-España.
{ chervas@uco.es, zamarck@yahoo.es, i82fecaj@uco.es }

A.J. Tallón
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
41012-Sevilla-España
atallon@us.es

Resumen

Este trabajo aborda la resolución de problemas de clasificación multiclase utilizando una metodología que combina la Regresión Logística y modelos evolutivos de Redes Neuronales de unidades de tipo proyección, como son las unidades producto y las sigmoides. Para estimar los coeficientes del modelo se considerarán dos etapas; en la primera aprendemos los exponentes de las funciones sigmoidales y potenciales, entrenando los modelos de redes neuronales mediante computación evolutiva y, una vez estimados el número de funciones de base y los exponentes de estas funciones, se aplica el método de máxima verosimilitud al espacio de características formado por las covariables iniciales junto con las nuevas funciones de base obtenidas al entrenar los modelos. Esta metodología híbrida en el diseño del modelo y en la estimación de los coeficientes se aplica a un problema de determinación de clases de probabilidad de crecimiento de microorganismos. Los resultados obtenidos con este modelo mejoran los conseguidos con una Regresión Logística estándar en cuanto a porcentaje de patrones bien clasificados sobre el conjunto de generalización.

1. Introducción

Existen muchos campos de investigación donde es necesario resolver problemas de clasificación o reconocimiento de patrones, comprendiendo en la actualidad, entre otros, Medicina, Biología, Visión por Computador, Inteligencia Artificial, Teledetección, y adquiriendo recientemente un gran interés asociado a problemas de Minería de Datos, dado que nos encontramos con espacios de características de entrada de alta dimensionalidad

como las asociadas a la clasificación de documentos, de clientes *web*, de alumnos aprendiendo a través técnicas de *e-learning*, etc.

El modelo de Regresión Logística ha sido ampliamente empleado como método de regresión en Estadística desde hace muchos años y, recientemente, ha recibido una gran aceptación en la comunidad de *machine learning* debido a su cercana relación con las nuevas teorías asociadas a las máquinas de soporte vectorial SVM [16] y a las técnicas de remuestreo iterativo como puede ser AdaBoost [4]. La Regresión Logística, RL, es un modelo de regresión lineal generalizada que trata de predecir las probabilidades a posteriori de pertenencia de cada uno de los patrones de un conjunto de entrenamiento a uno de los valores que toma la variable dependiente mediante relaciones lineales con las variables predictoras [11]. Por otra parte, un modelo de RL multiclase se puede representar de forma equivalente a una estructura de grafo de tipo perceptron, con una función de activación logística, simbolizando un esquema de red neuronal lo más sencillo posible, como se observa en la Figura 1.

Además, en los últimos años, se está haciendo hincapié en la comparación de las nuevas técnicas y algoritmos de clasificación basados en modelos de Redes Neuronales y de Árboles de Decisión con modelos clásicos estadísticos como los de análisis discriminante lineal y cuadrático y los modelos de RL estándar o con restricciones [15].

El procedimiento habitual de estimación de los coeficientes de un modelo lineal de RL es el de máxima verosimilitud, en el que para obtener el óptimo de la función de verosimilitud se utiliza habitualmente un método de optimización local basado en un algoritmo iterativo de tipo Newton-Raphson o de mínimos cuadrados con reasignación de pesos (IRLS) [8].

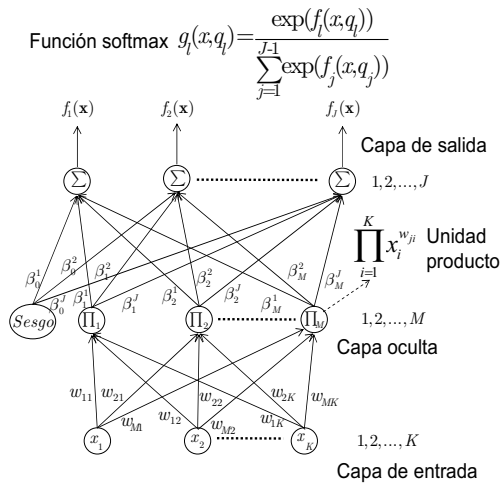


Figura 1. Esquema de red neuronal

Desde un punto de vista formal, la RL es un procedimiento sencillo y útil, pero no siempre se verifica que las probabilidades de pertenencia a cada clase transformadas mediante una transformación logarítmica presenten una relación lineal causa-efecto sobre las covariables.

Existen diferentes tipos de funciones de base, entre las que destacamos las sigmoides que dan lugar a los modelos de redes neuronales multicapa (MLP) [2], las funciones de tipo *kernel* o ventana, donde las funciones más habituales son las gaussianas que dan lugar a los modelos de base radial (RBF) y aquellas que dan lugar a los modelos de aprendizaje basado en proyecciones (*projection pursuit learning*) [5]. También deben destacarse los modelos lineales generalizados [7] y los métodos adaptativos multivariantes de tipo *spline* como MARS [6] o la muy reciente aproximación de utilizar funciones recíprocas sigmoides [14]. Sin embargo, existen dificultades a la hora de su utilización debido a la complejidad de estos modelos derivada del diseño de la tipología y de la determinación del número de funciones de base necesarios para un problema concreto de clasificación.

Para introducir no linealidad en el modelo y poder abordar otras metodologías de clasificación utilizando técnicas heurísticas y técnicas estadísticas de optimización, proponemos en este trabajo un modelo de regresión logística basado en la hibridación del modelo lineal en las covariables

iniciales y en las funciones de base (no lineales) de tipo proyección. De esta forma, la parte no lineal del modelo propuesto se corresponde con funciones de tipo proyección como son las Redes Neuronales de Unidades Producto (PUNN) [3], o las Redes Neuronales de Unidades Sigmoides o redes MLP.

Desafortunadamente en nuestra aproximación no podemos garantizar la no existencia de óptimos locales en la superficie de log-verosimilitud como ocurre en los modelos estándar de regresión logística, dado que las superficies de error asociadas a las redes PUNN o redes MLP son complejas, con numerosos óptimos locales y superficies planas.

El aprendizaje de los coeficientes y del número de funciones de base se lleva a cabo en varias etapas. En la primera, se utiliza un algoritmo evolutivo (EA) para diseñar la estructura: número de funciones de base y el aprendizaje de los pesos asociados a los exponentes de las funciones de base; en la segunda etapa, consideramos lineal el modelo tanto en estas nuevas funciones de base como en las covariables iniciales y podemos obtener los estimadores de los coeficientes de regresión mediante un procedimiento de máxima verosimilitud asociado al modelo estándar de RL. Por último, aplicamos un método de selección por pasos de las covariables en función de su capacidad para explicar la variable de respuesta.

Para evaluar el rendimiento de nuestra metodología utilizamos un problema de clasificación multiclase asociado a la determinación de la probabilidad de crecimiento de microorganismos *S. aureus* (*Staphylococcus aureus*) en función de diferentes factores de almacenamiento y manipulación de alimentos como son la temperatura (T), el pH, y la sal a_w . Los resultados empíricos muestran que tanto la metodología como el modelo propuesto son muy prometedores tanto en su capacidad de predicción en la clasificación como en su simplicidad, dada por el relativamente pequeño número de coeficientes y de funciones de base utilizadas en los mejores modelos propuestos como clasificadores.

Es interesante apuntar que el modelo propuesto mejora los resultados obtenidos con el modelo de RL (denominado LR) utilizando las covariables iniciales. De esta forma, el modelo híbrido de RL con covariables iniciales y con

funciones de base sigmoides, LRLSU, determina un buen balance entre la estructura lineal y no lineal del modelo.

El trabajo está organizado de forma tal que en la sección II hacemos una revisión de trabajos relacionados con el tema que nos ocupa; en la sección III realizamos una breve introducción a la RL multiclase para, a continuación, presentar nuestro modelo en profundidad; en la sección IV describimos la metodología de estimación de los coeficientes del modelo basada en dos etapas; en la sección V planteamos el problema de la determinación de las probabilidades de crecimiento de microorganismos *S. aureus* mediante técnicas de clasificación; en la VI, mostramos los experimentos realizados y los resultados obtenidos para analizar el rendimiento de nuestro algoritmo y en la sección VII presentamos las conclusiones y propuestas de futuro.

2. Trabajos relacionados

En esta sección damos una breve reseña acerca de los diferentes métodos que utilizan funciones de base para ir más allá de la linealidad del modelo, entre los cuales citaremos trabajos recientes que muestran la proximidad existente entre modelos de Regresión Logística y métodos de *machine learning*.

Los modelos aditivos generalizados [7] comprenden métodos estadísticos automáticos y flexibles que se utilizan para identificar y caracterizar los efectos de la regresión no lineal. Para un problema de clasificación con dos clases, el modelo de Regresión Logística es un ejemplo de modelo aditivo generalizado donde se reemplaza cada término lineal por una forma funcional más general que aproxima funciones multidimensionales mediante la suma de funciones unidimensionales.

Uno de los modelos más populares de regresión no lineal es la regresión adaptativa multivariante de tipo *spline* (MARS) [6], donde las funciones base vienen determinadas por el producto de algún número de funciones unidimensionales de tipo *spline*. Las funciones de base se añaden incrementalmente durante el aprendizaje, utilizando una técnica constructiva de selección secuencial.

3. Regresión logística multiclase con funciones de base de tipo proyección

En un problema de clasificación, tomamos un conjunto de medidas, x_i , $i=1,2,\dots,k$, sobre los elementos de una población, los cuales están clasificados en una de las J clases. Si suponemos J finito, las medidas x_i son variables aleatorias de elementos de estas clases. Sea $D = \{(\mathbf{x}_n, \mathbf{y}_n); n=1,2,\dots,N\}$ una muestra extraída de la población que consideraremos como el conjunto de entrenamiento, donde $\mathbf{x}_n = (x_{1n}, \dots, x_{kn})$ es el vector de medidas que toma valores en $\Omega \subset R^k$ e \mathbf{y}_n es la etiqueta de la clase a la que pertenece el n -ésimo individuo de D . Adoptaremos la técnica bastante habitual "1-de- J " de situar las etiquetas de las clases codificando el vector $\mathbf{y} = (y^{(1)}, y^{(2)}, \dots, y^{(J)})$ de forma tal que

$y^{(l)} = 1$ si \mathbf{x} se corresponde con un elemento perteneciente a la clase l e $y^{(l)} = 0$ en otro caso. Basándonos en el conjunto de entrenamiento estamos interesados en encontrar una función de decisión $C: \Omega \rightarrow \{1,2,\dots,J\}$ que clasifique correctamente los elementos.

Para calcular el rendimiento de nuestro clasificador utilizaremos el porcentaje de patrones bien clasificados (CCR) en la forma,

$$CCR = \frac{1}{N} \sum_{n=1}^N I(C(\mathbf{x}_n) = \mathbf{y}_n), \text{ donde } I(.) \text{ es la función de pérdida cero-uno.}$$

Un buen clasificador intentará obtener para un problema dado el mayor valor posible de CCR. Habitualmente, se supone que las covariables de la muestra obtenida para formar el conjunto de entrenamiento son variables aleatorias idénticamente distribuidas que siguen una distribución de probabilidad desconocida. De esta forma, bajo la hipótesis de que la probabilidad condicionada de que \mathbf{x} pertenezca a la clase l verifica: $p(y^{(l)} = 1 | \mathbf{x}) > 0$, $l = 1,2,\dots,J$, $\mathbf{x} \in \Omega$

definimos la función:

$$f_l(\mathbf{x}, \boldsymbol{\theta}_l) = \log \frac{p(y^{(l)} = 1 | \mathbf{x})}{\sum_{j=1}^J p(y^{(j)} = 1 | \mathbf{x})}, \quad l = 1,2,\dots,J$$

donde θ_l es el vector de pesos correspondiente a la clase l y $f_j(\mathbf{x}, \theta_j) \equiv 0$. Si utilizamos un modelo de regresión logística multiclase, la probabilidad de que \mathbf{x} pertenezca a la clase l viene dada por:

$$p(y^{(l)} = 1 | \mathbf{x}, \theta) = \frac{\exp f_l(\mathbf{x}, \theta_l)}{\sum_{j=1}^J \exp f_j(\mathbf{x}, \theta_j)}, \quad l=1,2,\dots,J$$

donde $\theta = (\theta_1, \theta_2, \dots, \theta_{J-1})$.

Si utilizamos como regla de decisión la función softmax, ésta coincide con la regla óptima de Bayes. Así, dado el valor de \mathbf{x} , un individuo deberá asignarse a la clase que tenga el mayor valor de probabilidad, esto es:

$$C(\mathbf{x}) = \hat{l}, \quad \text{donde } \hat{l} = \arg \max_l f_l(\mathbf{x}, \hat{\theta}_l)$$

Por otra parte, debido a los axiomas del Cálculo de Probabilidades, tenemos que:

$$\sum_{l=1}^J p(y^{(l)} = 1 | \mathbf{x}, \theta) = 1,$$

y la probabilidad de una de las clases, en este caso consideraremos la última, no necesitará ser estimada, por lo que $f_j(\mathbf{x}, \theta_j) \equiv 0$.

Nuestro modelo de regresión logística está basado en la combinación de un modelo lineal en las covariables iniciales y en las nuevas covariables formadas por unidades de base $B_j(\mathbf{x}, \mathbf{w}_j)$ de tipo proyección en la forma:

$$f_l(\mathbf{x}, \theta_l) = \alpha_0^l + \sum_{i=1}^k \alpha_i^l x_i + \sum_{j=1}^m \beta_j^l B_j(\mathbf{x}, \mathbf{w}_j), \quad l=1,2,\dots,J-1$$

donde $\theta_l = (\alpha^l, \beta^l, \mathbf{W})$, $\alpha^l = (\alpha_0^l, \alpha_1^l, \dots, \alpha_k^l)$,

$\beta^l = (\beta_1^l, \dots, \beta_m^l)$ y $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m)$, con

$\mathbf{w}_j = (w_{j1}, w_{j2}, \dots, w_{jk})$, para $w_{ji} \in \mathbb{R}$. La parte no lineal de $f_l(\mathbf{x}, \theta_l)$ se corresponde con redes neuronales *feed-forward* con funciones de transferencia sigmoides o producto, estas últimas introducidas por Durbin y Rumelhart [3].

4. Estimación de coeficientes del modelo

En el contexto de un aprendizaje supervisado, las componentes de los vectores de pesos $\theta = (\theta_1, \theta_2, \dots, \theta_{J-1})$ se estiman a partir del conjunto de entrenamiento D mediante el método

de máxima verosimilitud (ML), construyendo para ello la función:

$$l(\theta) = -\frac{1}{N} \sum_{n=1}^N \log p(\mathbf{y}_n | \mathbf{x}_n, \theta) = \frac{1}{N} \sum_{n=1}^N \left[-\sum_{l=1}^J y_n^{(l)} f_l(\mathbf{x}_n, \theta_l) + \log \sum_{l=1}^J \exp f_l(\mathbf{x}_n, \theta_l) \right] \quad (1)$$

La superficie de error asociada con este modelo a tenor de la forma de $f_l(\mathbf{x}_n, \theta_l)$ es muy irregular con numerosos óptimos locales. Debido a la no linealidad del modelo con respecto a los parámetros θ_l y dado que la matriz Hessiana asociada es indefinida, no se puede garantizar la obtención del máximo global utilizando métodos de gradiente. Además, el número óptimo de funciones de base del modelo es desconocido (esto es, el número de nodos de la capa oculta de las redes), por lo que utilizaremos un algoritmo evolutivo (EA) como heurística de optimización de ese número y de los exponentes de las funciones de base, como paso previo a la estimación de los coeficientes del modelo de RL por el método de máxima verosimilitud. Las condiciones bajo las cuales existe un máximo global y las condiciones de convergencia de los estimadores obtenidos por el método de máxima verosimilitud se discuten en [12]. La estimación de los parámetros α y β habitualmente se realiza mediante un procedimiento iterativo como el algoritmo de Newton-Raphson o el algoritmo iterativo de mínimos cuadrados con reasignación de pesos (IRLS) [8].

La propuesta se basa en la combinación de un EA (explorador global) y de un procedimiento de optimización local (explotador local) llevado a cabo mediante un procedimiento de maximización de la función de verosimilitud asociada a un modelo de RL. En una primera etapa, se aplica un EA para diseñar la estructura y el entrenamiento de los exponentes de las funciones de base de redes neuronales de unidades producto. El proceso evolutivo determina el número m de funciones de base sigmoides o producto de las covariables iniciales del problema, así como los vectores \mathbf{w}_j , correspondientes a los exponentes de cada una de estas m funciones. Una vez que tenemos determinadas las funciones base mediante el algoritmo evolutivo,

$$B(\mathbf{x}, \mathbf{W}) = (B_1(\mathbf{x}, \mathbf{w}_1), B_2(\mathbf{x}, \mathbf{w}_2), \dots, B_m(\mathbf{x}, \mathbf{w}_m)),$$

consideramos un aumento del espacio de entrada añadiendo a las covariables iniciales estas nuevas variables, transformaciones no lineales de dichas covariables iniciales. De esta forma, el modelo es lineal en este nuevo espacio de variables y sus coeficientes α y β , se calculan mediante un procedimiento de máxima verosimilitud condicional asociado a la RL estándar. Por último, utilizamos un método por etapas para eliminar variables de forma secuencial del modelo completo hasta que ninguna nueva eliminación mejore su capacidad de clasificación sobre el conjunto de entrenamiento.

En primer lugar, aplicamos un EA para encontrar las funciones de base, $B(\mathbf{x}, \mathbf{W}) = (B_1(\mathbf{x}, \mathbf{w}_1), B_2(\mathbf{x}, \mathbf{w}_2), \dots, B_m(\mathbf{x}, \mathbf{w}_m))$, o nuevas covariables, (z_1, z_2, \dots, z_m) , correspondientes a la parte no lineal de la función $f(\mathbf{x}, \theta)$. Debemos determinar el número de funciones m y la matriz de pesos formada por los exponentes estimados de las funciones sigmoideas o potenciales $\hat{\mathbf{W}} = (\hat{w}_1, \hat{w}_2, \dots, \hat{w}_m)$. Entre los diferentes paradigmas de la Computación Evolutiva, hemos elegido la Programación Evolutiva (EP) debido al hecho de que estamos evolucionando modelos cuya representación en grafo está asociada a un modelo de red neuronal artificial con números reales como pesos. Nuestro EA para diseñar la arquitectura del modelo y la estimación de los exponentes reales tiene puntos en común con otros algoritmos evolutivos [1], [17] y puede verse con detalle en [1]. La principal diferencia planteada en este trabajo está en la función de aptitud asociada a cada modelo, dado que abordamos un problema de clasificación. De este modo, consideraremos la ecuación (1) como la función de error de un individuo f de la población. Por tanto, definimos la medida de aptitud de un individuo mediante una función estrictamente decreciente de la función de error

$$l(\theta) \text{ en la forma } A(f) = \frac{1}{1 + l(\theta)}, \text{ donde}$$

$$0 < A(f) \leq 1.$$

La idea básica del algoritmo es la utilización de operadores de selección, replicación y mutación (paramétrica y estructural) en el proceso de

evolución. La estructura del algoritmo aparece en [10].

Una vez obtenidas las nuevas covariables del modelo, aplicamos el modelo de RL estándar a las variables $(x_1, x_2, \dots, x_k, z_1, \dots, z_m)$ en el nuevo espacio de características de entrada. De esta forma, calculamos el máximo de la función de verosimilitud condicional para las N observaciones del conjunto D .

Para obtener el modelo final utilizaremos un método de eliminación de variables no significativas del modelo, basado en la eliminación de una variable en cada paso, empezando con el modelo completo y eliminando variables secuencialmente hasta que no se mejore la capacidad de clasificación. Controlando de esta forma el número de coeficientes del modelo, podremos disminuir el riesgo de construir un modelo complejo que sobreaprenda los datos del conjunto de entrenamiento, con la consiguiente pérdida de capacidad de generalización sobre los datos del conjunto de test.

5. Probabilidad de crecimiento de microorganismos de *S. Aureus*

El envenenamiento de la comida causado por la contaminación con staphylococos es una de las causas más corrientes de la aparición de gastroenteritis en el mundo, producida por la ingestión de comida que contiene toxinas [9]. En especial el *S. aureus* produce con frecuencia las intoxicaciones más corrientes por ingestión de alimentos, [9]. El número de personas enfermas por la ingesta de comida con *S. aureus* referenciado por el Sistema Microbiológico Español de Información (*Spanish Microbiological Information System*, SIM), ha aumentado en los últimos cinco años llegando a 550 por año.

Debido a la gran presencia de *S. aureus* en diferentes tipos de comida, es importante saber su capacidad de crecimiento/supervivencia en la producción de alimentos y en los procesos de preservación y almacenamiento de los mismos. Los modelos probabilísticos se caracterizan por definir los límites de crecimiento / no crecimiento de un determinado microorganismo en un determinado medio como una función de ciertos factores del entorno en un muy limitado rango de condiciones.

Hemos realizado un diseño factorial para determinar los límites de crecimiento de *S. aureus* bajo diferentes condiciones de temperatura (8-19°C), pH (4.5-7.5) y a_w (0.856-0.999). Las diferentes condiciones medioambientales se dividieron siguiendo una técnica de validación cruzada de tipo holdout, pero con un diseño de experimentos donde se eligieron 146 condiciones para el conjunto de entrenamiento y que cubrieran las regiones extremas del espacio tridimensional formado por los factores definidos anteriormente. Para construir el conjunto de generalización, se seleccionaron 141 condiciones de diferentes niveles pertenecientes al interior del dominio de los factores. El propósito de dicha selección fue poder definir el conjunto de entrenamiento utilizando zonas que representen realmente los bordes de las zonas de crecimiento / no crecimiento para poder obtener un mejor resultado sobre el conjunto de generalización. El crecimiento fue monitorizado mediante medidas de absorbancia realizadas con un Bioscreen C (Labsystems, Finland) durante 36 días para todas las condiciones.

Los experimentos han sido planteados con el objetivo de contrastar los resultados de nuestros modelos con un procedimiento de RL sólo con covariables iniciales. Los valores de los parámetros utilizados en el EA para el aprendizaje de los modelos de redes neuronales son: i) w_{ji} , los exponentes se inicializan en el intervalo $[-5,5]$, ii) β_j , los coeficientes se inicializan en el intervalo $[-5,5]$, iii) $m = 6$, el máximo número de nodos en la capa oculta de los modelos, iv) $N_r = 1000$, el tamaño de la población y v) el número de nodos que pueden ser añadidos o eliminados mediante una mutación estructural está entre 1 y 2, mientras que el número de conexiones que se pueden añadir o eliminar en una mutación estructural es un número entero del conjunto $\{1,3\}$. Las condiciones propuestas para el criterio de parada son: i) o bien que en 20 generaciones no se mejore ni el rendimiento medio del 20% de los mejores individuos de la población, ni la aptitud del mejor individuo, ii) o bien que el algoritmo alcance 80 generaciones.

6. Resultados

Para validar el rendimiento del modelo utilizamos el porcentaje de patrones de los datos correctamente clasificados para el conjunto de generalización, CCR_G . Para poder seleccionar las variables más significativas del modelo de regresión logística, utilizamos un método de eliminación de variables por pasos, utilizando el software SPSS 13.0 [13].

En la Tabla 1 presentamos, en primer lugar, los resultados estadísticos obtenidos para valores de CCR con modelos de red con funciones de base de tipo sigmoide (SU) y de tipo producto (PU), donde el número de generaciones es 80 y la topología es la misma: 3 características de entrada, de 2 a 5 nodos en la capa oculta y 4 nodos en la capa de salida. Observamos que los modelos utilizando unidades PU son mejores tanto en eficacia para CCR_G (Media = 78,21; SD = 1,65) como en complejidad asociada al número de coeficientes (Media = 17,50; SD = 3,42). Una vez obtenidos los mejores modelos con las redes MLP y PUNN, tendremos, por tanto, el número de funciones de base como los exponentes de esas funciones. De esta manera, podemos construir el modelo híbrido de regresión logística con covariables iniciales y con funciones de base sigmoideas, LRLSU, y el modelo correspondiente con unidades producto, LRLPU, los cuales se muestran en las Tablas 2 y 3, tras haber aplicado la eliminación de variables no significativas de los modelos. En estas tablas presentamos las funciones discriminantes donde el número de coeficientes de los mejores modelos son muy similares. En la Tabla 4 presentamos la matriz de contingencia con los valores CCR_T y CCR_G para los modelos LR, LRLSU y LRLPU. Observamos que el mejor modelo es LRLSU seguido de LRLPU con porcentajes en el CCR_G del 83,8 y del 81,7 respectivamente, valores muy superiores a los obtenidos con una RL multiclasa (76,1), con 7,7 y 5,6 puntos por encima en cuanto a CCR_G , lo que indica que nuestra metodología, al menos en este problema supera ampliamente a la RL estándar.

7. Conclusiones

En este trabajo nos hemos centrado en la presentación de una nueva metodología donde se combinan modelos de regresión lineal generalizada y de RL multiclase con modelos de redes neuronales con funciones de base de tipo proyección como las funciones sigmoides y producto. Para estimar el número de estas funciones y los coeficientes asociados a los exponentes de las mismas, hemos utilizado un algoritmo heurístico basado en el paradigma de la EP.

Hemos aplicado el modelo propuesto a un problema difícil de determinación de clases asociadas a intervalos de probabilidad de crecimiento del microorganismo *S. aureus*. Los mejores resultados obtenidos de CCR_G para los modelos híbridos mejoran los obtenidos utilizando RL o modelos de unidades sigmoides o de unidades producto como clasificadores. De esta forma, el modelo híbrido determina un buen balance entre considerar sólo un modelo lineal o sólo un modelo no lineal.

Agradecimientos

Este trabajo ha sido financiado en parte por el proyecto TIN2005-08386-C05-02 de la Comisión Interministerial de Ciencia y Tecnología y fondos FEDER.

Referencias

[1] Angeline, P.J. Saunders, G.M., Pollack, J.B.: An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5 (1), 54-65, 1994.

[2] Bishop, M.: *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[3] Durbin, R., Rumelhart, D.: Products Units: A computationally powerful and biologically plausible extension to back propagation networks. *Neural Computation*, 1, 133-142, 1989.

[4] Freund, Y., Shapire, R.: Experiments with a new boosting algorithm. *Machine Learning: Proc. of the 13th Int. Conf.*, 1996.

[5] Friedman, J., Stuetzle, W.: Projection pursuit regression. *J. Amer. Stat. Assoc.*, 76, 817-823, 1981.

[6] Friedman, J.: Multivariate adaptive regression splines. *Ann. Stat.*, 19, 1-141, 1991.

[7] Hastie, T.J., Tibshirani, R.J.: *Generalized additive models*. Chapman & Hall, 1990.

[8] Hastie, T., Tibshirani, R.J., Friedman, J.: *The Elements of Statistical Learning. Data mining, Inference and Prediction*, Springer, 2001.

[9] Jablonski, L.M., Bohach, G.A.: *Staphylococcus aureus*. In: L Beuchat, M Doyle, and T Montville (eds), *Fundamentals of Food Microbiology*. ASM Press. Washington, D.C., 410-434, 2001.

[10] Martínez-Estudillo, A.C., Martínez-Estudillo, F. J., Hervás-Martínez, C., et al.: Evolutionary Product Unit based Neural Networks for Regression, *Neural Networks*, 19 (4), 477-486, 2006.

[11] McCullagh, P., Nelder, J.A.: *Generalized Linear Models*, 2nd edn., London, 1989.

[12] McLachlan, G.: *Discriminant analysis and statistical pattern recognition*. N. York, 1992.

[13] SPSS ©: SPSS ©13.0 advanced models. Inc, S., Ed. Chicago, IL, 1999.

[14] Toh, K.A.: Training a reciprocal-sigmoid classifier by feature scaling-space. *Machine Learning*, 65(1), 273-308, 2006.

[15] Vach, W., Robner, R., Schumacher, M.: *Neural Networks and logistic regression: Part I & II. Comp. Stat. & Data Analysis*, 21, 661-701, 1996.

[16] Vapnik, V.: *The nature of Statistical Learning Theory*, Springer, 1999.

[17] Yao, X., Liu, Y.: A new evolutionary system for evolving artificial neural networks. *IEEE Trans. on Neu. Net.*, 8 (3), 694-713, 1997.

Tabla 1. Resultados estadísticos de los valores CCR y número de conexiones obtenidos por los mejores individuos en 30 ejecuciones del algoritmo evolutivo para redes MLP y PUNN

Gen. 80	CCR _T		CCR _G				n° de conexiones			
3:2-5:4	Media	SD	Mejor	Peor	Media	SD	Mejor	Peor	Media	SD
MLP	79,90	2,02	83,56	73,28	77,15	2,17	80,98	71,12	25,33	2,72
PUNN	80,73	2,11	84,93	76,71	78,21	1,65	80,98	75,35	17,50	3,42

Tabla 2. Funciones discriminantes obtenidas con LRLSU

LRLSU Variables	Función 1							
	Sesgo	X_1^*	X_2^*	X_3^*	S_1	S_2	S_3	S_4
Coef.	6,13	-8,15	10,63	-30,39		19,73		314,62
Std Error	7,14	4,83	5,64	6,88		7,62		134,86
p-value	,39	,09	,05	,00		,01		,02
Función 2								
Coef.	8534,82	-12215,15	-4539,99	932,35		-7767,08		-17428,67
Std Error	196539,70	312634,95	82734,39	13806,76		196295,57		,00
p-value	,96	,96	,95	,94		,96		-
Función 3								
Coef.	11753,59	-16801,88	-4846,41	363,15		-12559,32		-7573,14
Std Error	143664,20	212493,40	54672,96	11367,93		167573,26		,00
p-value	,93	,93	,92	,97		,94		-

$$S_1 = \frac{1}{1 + \exp\{-2,06 - 6,14 \times X_3^*\}}; S_2 = \frac{1}{1 + \exp\{-2,32 - 6,90 \times X_1^* - 1,79 \times X_2^* + 0,01 \times X_3^*\}}$$

$$S_3 = \frac{1}{1 + \exp\{-2,11 + 1,58 \times X_1^* + 1,26 \times X_3^*\}}; S_4 = \frac{1}{1 + \exp\{-1,70 - 1,57 \times X_1^* - 2,83 \times X_2^*\}}$$

nº de funciones de base= 4, nº de coeficientes= 25

Tabla 3. Funciones discriminantes obtenidas con LRLPU

LRLPU Variables	Función 1							
	Sesgo	X_1^*	X_2^*	X_3^*	B_1	B_2	B_3	B_4
Coef.	21,71	-4,59	3,37	-1,92	-29,26	-20,12	,22	
Std Error	8,34	8,47	4,91	29,16	15,23	36,45	,26	
p-value	,00	,58	,49	,94	,05	,58	,40	
Función 2								
Coef.	923,11	-21402,46	-4082,79	-1821,28	24838,44	965,64	-65,63	
Std Error	715381,83	,000	83885,58	942273,70	516541,99	818414,38	123157,77	
p-value	,99	-	,96	,99	,96	,99	1,00	
Función 3								
Coef.	73,90	-3,06	16,18	-106,22	-66,73	79,96	-9,72	
Std Error	57,26	25,35	7,94	69,12	28,97	77,56	10,66	
p-value	,19	,90	,04	,12	,02	,30	,36	

$$B_1 = (X_1^*)^{0,67} (X_2^*)^{0,29} (X_3^*)^{0,19}; B_2 = (X_1^*)^{0,30} (X_3^*)^{1,23}; B_3 = (X_3^*)^{-1,87}; B_4 = (X_1^*)^{1,09}$$

nº de funciones de base= 4, nº de coeficientes= 27

Tabla 4. Matrices de contingencia para entrenamiento y generalización utilizando: Regresión Logística, LR, Regresión Logística con covariables iniciales y SU, LRLSU, (entre paréntesis) y Regresión Logística con covariables iniciales y PU, LRLPU (entre corchetes).

LR, LRLSU, LRLPU	Entrenamiento					Generalización				
	G=1	G=2	G=3	G=4	FC %	G=1	G=2	G=3	G=4	FC %
Clase Prevista/ Clase Real										
G= 1	83 (90) [88]	3 (0) [0]	0 (1) [1]	7 (3) [4]	89,2 (96,8) [94,6]	81 (81) [81]	2 (2) [2]	0 (0) [1]	8 (6) [6]	89,1 (89,1) [89,1]
G= 2	5 (0) [0]	1 (8) [8]	0 (0) [0]	2 (0) [0]	12,5 (100) [100]	5 (1) [1]	2 (4) [5]	0 (3) [2]	1 (0) [0]	25,0 (50,0) [62,5]
G= 3	3 (0) [1]	0 (0) [0]	0 (7) [5]	4 (0) [1]	0,0 (100) [71,4]	3 (1) [2]	0 (2) [1]	0 (1) [0]	4 (3) [4]	0,0 (14,3) [0,0]
G=4	9 (4) [4]	0 (0) [0]	0 (0) [0]	29 (34) [34]	76,3 (89,5) [89,5]	11 (4) [7]	0 (0) [0]	0 (1) [0]	25 (31) [29]	69,4 (86,1) 80,6]
CCR					77,4 (95,2) [92,5]					76,1 (83,8) [81,7]