COMPARACIÓN POR SIMPLICIDAD DE MÉTODOS DE APRENDIZAJE EN ESTIMACIÓN DE FUNCIONES

COMPARISON BY SIMPLICITY OF LEARNING METHODS FOR FUNCTION ESTIMATION

Dora M. Ballesteros[1]
Andrés E. Gaona[2]
Luis F. Pedraza[3]

Recibido: Julio 2010
Aprobado: Octubre 2010

Resumen

uno de los problemas que resuelven los sistemas inteligentes consiste en la estimación de funciones, para lo cual se parte de un número finito de datos de un proceso y el objetivo es encontrar la función que mejor los modela. Dentro de los métodos de aprendizaje, los métodos de optimización, como el descenso de gradiente y el gradiente conjugado, han sido tradicionalmente utilizados en este tipo de problemas, con ventajas como la sencillez en los primeros y la rapidez de convergencia en los segundos. De acuerdo con el principio de simplicidad, se escoge el método que sea más sencillo, pero a la vez el más preciso, de tal forma que ninguno de los dos puede a priori considerarse rotundamente mejor que el otro, porque no satisface simultáneamente las dos condiciones. En este trabajo se evalúan los dos métodos en la estimación de funciones lineales y cuadráticas y se proponen mejoras con el objetivo de proporcionar un método que sea el mejor en términos de "simplicidad".

Palabras clave
gradiente descendente, gradiente conjugado, simplicidad, estimación de funciones

Abstract
One of the problems they solve Intelligent Systems, is the estimation of functions, for which part of a finite number of data from one process and the goal is to find better role models. Among the unsupervised learning methods, optimization methods like gradient descent and conjugate gradient have been traditionally used in such problems, with advantages such as simplicity

1. Ingeniera electrónica. Magíster en Ingeniería Electrónica y de Computadores. Docente, Universidad Militar Nueva Granada. Correo electrónico: dora.ballesteros@unimilitar.edu.co

2. Ingeniero electrónico. Magíster en Ingeniería Área Electrónica. Docente, Universidad Distrital Francisco José de Caldas. Correo electrónico: aegaona@udistrital.edu.co

3. Ingeniero electrónico. Magíster en Teleinformática. Docente, Universidad Distrital Francisco José de Caldas. Correo electrónico: lfpedrazam@udistrital.edu.co

in the first and the speed of convergence in the latter. According to the principle of simplicity, we choose the method that is simple yet the most accurate, so that neither of the two methods can strongly considered better than another, why not simultaneously satisfy both conditions. This paper evaluates the two methods in the estimation of linear and quadratic functions and suggests improvements in its definition with the objective of proportional method that is best in terms of "simplicity."

Key words
gradient descent, conjugate gradient, simplicity, function estimation1

INTRODUCCIÓN

Un sistema inteligente (SI) es un programa de computación que aprende y que tiene como su principal característica la adaptabilidad al medio [1]. Dentro de las tareas que se pueden "programar" en un SI con aprendizaje estadístico se encuentran la estimación, la clasificación y el reconocimiento de patrones [2]. Las primeras, se relacionan con la definición de funciones que se ajustan a un conjunto de datos conocido, con el propósito de determinar el comportamiento en momentos no proporcionados por el conjunto de datos, en un rango interno (interpolación), o en un rango externo (extrapolación); las segundas, consisten en la definición de reglas que separan datos; y las terceras, en la identificación y extracción de características del conjunto de datos conocido.

En este trabajo nos centraremos en el análisis de métodos de estimación, desde una perspectiva que permita la selección del mejor método, dentro del principio denominado "simplicidad", en el cual se escoge el método que sea más sencillo, pero a la vez el más preciso [3]. La estimación de una función puede verse como un problema de ajuste de curva [4] en el que se pueden calcular los errores entre la curva que se predice y los datos conocidos. De tal forma, al ajustarse igual de bien dos curvas a los datos (es decir, cuando se tienen los mismos errores), el principio de simplicidad dice que se debe favorecer la hipótesis más simple sobre la hipótesis menos simple.

Para nuestro ejercicio calificaremos la simplicidad de la hipótesis de acuerdo con el conjunto de operaciones que se deben realizar para encontrar la solución o los parámetros de las curvas de ajuste. Se evalúan dos escenarios: datos provenientes de funciones lineales y datos de funciones cuadráticas. Los datos son generados con adición de ruido gaussiano, con el propósito de hacer el ejercicio lo más próximo a situaciones reales, en las que se tienen pequeñas desviaciones producto de la calibración de instrumentos y las condiciones externas. Se comparan dos métodos clásicos de optimización, el método de descenso de gradiente y el gradiente conjugado, y se proponen modificaciones a cada uno de ellos, a fin de mejorar la sencillez y la convergencia.

2 MÉTODOS DE ESTIMACIÓN

Los modelos que se evalúan en el presente trabajo corresponden a una función lineal y a una cuadrática. La función lineal es

$$g_0(x) = w_o x + w_1 + \epsilon$$

y la cuadrática

$$g_1(x) = (w_o x + w_1)^2 + \epsilon,$$

donde ϵ corresponde al ruido gaussiano aditivo.

2.1. AJUSTE CURVA LINEAL

Se define una función lineal que modela el conjunto de datos dado como

$$f(x) = w_0 x + w_1,$$

de tal forma que el sistema debe estimar los parámetros w_0 y w_1. Este modelo no se ajusta al ruido, porque de ser así se podría tener un sobreajuste de curva [4].

Método descenso de gradiente: el primer método a estudiar es el método de descenso de gradiente, el cual permite estimar cada nuevo parámetro a partir del anterior, teniendo en cuenta la derivada de la función de coste, definida como el error cuadrático entre los datos conocidos y los datos encontrados con el modelo [5]. De esta manera, el parámetro actual se calcula como:

$$w[n+1] = w[n] - K\frac{\partial L(w[n])}{\partial w[n]}$$

con

$$K = \begin{pmatrix} k_0 \\ k_1 \end{pmatrix}$$

(1)

y $L(w[n]), L(w[n]),$ correspondiente a la función de coste. Los dos parámetros se estiman utilizando las ecuaciones recursivas:

$$w_0[n+1] = w_0[n] + k\sum_{i=1}^{m}\left(y_i - f(x;w)\right)$$

(2)

$$w_1[n+1] = w_1[n] + k\sum_{i=1}^{m}\left(y_i - f(x;w)\right) * x_i$$

(3)

donde k es la constante de paso, *yi* es el valor real del dato, y *f(x;w)* es el valor al reemplazar los parámetros en la función.

Método descenso de gradiente con tasa de aprendizaje variable: es una variante del método de descenso de gradiente, en la cual la constante de paso (tasa de aprendizaje), que en el algoritmo original es constante (k), se calcula en línea, para hacer el ajuste más fino y disminuir las oscilaciones. Dicha constante se determina a través de un problema de optimización unidimensional, utilizando ajuste cuadrático, ajuste cúbico o búsqueda dorada.

Método descenso de gradiente modificado: nuestra propuesta de modificación de este método tiene tres características que la diferencian del método de gradiente clásico y con tasa de aprendizaje variable:

- Se utilizan dos constantes de paso, una por cada parámetro que se estima.
- La constante de paso es negativa (k<0) en el cálculo del parámetro

$$w_1[n+1]w_1[n+1].$$

- La dependencia de la derivada de la función de coste se realiza en términos del otro parámetro, es decir, que para el cál-

culo de w0 se tiene en cuenta la derivada de L en función de w1, y viceversa.

En conclusión, las ecuaciones recursivas quedan como:

$$w_0[n+1] = w_0[n] + k_2 \sum_{i=1}^{m} (y_i - f(x;w)) *$$

para *k2*>0

(3)

$$w_1[n+1] = w_1[n] + k_3 \sum_{i=1}^{m} (y_i - f(x;w))$$

para *k3*<0

(4)

Método gradiente conjugado: utiliza el valor del gradiente anterior para calcular el paso total entre una iteración y otra. El pseudocódigo se presenta en la figura 1 con λ como tasa de aprendizaje variable.

2.2. AJUSTE CURVA CUADRÁTICA

Se define una función cuadrática

$$f(x) = (w_1 x + w_2)^2,$$

de tal forma que el sistema debe estimar los parámetros *w1* y *w2*.

Método descenso de gradiente: las ecuaciones recursivas quedan como:

$$w_1[n+1] = w_1[n] - k \frac{\sum_{i=1}^{m} d\left(y_i - (w_1 x + w_2)^2\right)^2}{d w_1}$$

Inicializar $\vec{w}$ *en un valor aleatorio y variables:*

$$d_0 = \nabla L\big|_{w_{k(0)}}$$

Realizar iterativamente:

Actualizar $\vec{w}$*:*

$$\vec{w}_{k+1} = \vec{w}_k + \lambda . \vec{d}_k .$$

Actualizar el $\vec{d}_{k+1}$*:*

$$\beta_k = \frac{\left(\nabla \cdot L\big|_{w_{k+1}}\right)^T \cdot \left(\nabla \cdot L\big|_{w_{k+1}}\right)}{\left(\nabla \cdot L\big|_{w_k}\right)^T \cdot \left(\nabla \cdot L\big|_{w_k}\right)}$$

$$d_k = -\left(\nabla \cdot L\big|_{w_{k+1}}\right) + \beta_k \cdot d_k .$$

Hasta convergencia

Figura 1. Pseudocódigo algoritmo de gradiente conjugado. Basado en [6]

$$w_1[n+1] = w_1[n] + 4k \sum_{i=1}^{m} (y_i - (w_1 x + w_2)^2)(w_1 x^2 + w_2 x)$$

$$w_1[n+1] = w_1[n] + 4k \sum_{i=1}^{m} (y_i - (w_1 x + w_2)^2)(w_1 x + w_2) x$$

(6)

Y para el caso de *w2*:

$$w_2[n+1] = w_2[n] - k \frac{d\left(\sum_{i=1}^{m} (y_i - (w_1 x + w_2)^2)^2\right)}{d w_2}$$

$$w_2[n+1] = w_2[n] - k \frac{\sum_{i=1}^{m} d\left(y_i - (w_1 x + w_2)^2\right)^2}{d w_2}$$

$$w_2[n+1] = w_2[n] + 4k \sum_{i=1}^{m} (y_i - (w_1 x + w_2)^2)(w_1 x + w_2)$$

(7)

Para los métodos descenso de gradiente con tasa de aprendizaje variable, gradiente conjugado y gradiente modificado, se utilizan las mismas consideraciones que las expuestas para la función lineal.

Para los métodos descenso de gradiente con tasa de aprendizaje variable, gradiente con-

jugado y gradiente modificado, se utilizan las mismas consideraciones que las expuestas para la función lineal.

3. RESULTADOS

Se generan dos conjuntos de datos, correspondientes a

$$g_0(x) = w_o x + w_1 + \epsilon$$

y a

$$g_1(x) = (w_o x + w_1)^2 + \epsilon$$

En ambos casos el ruido tiene media cero y varianza unitaria. Para comparar el desempeño de los tres modelos, se parten de los mismos parámetros iniciales y se cuentan las iteraciones que satisfacen un valor de función de coste o error, para lo cual se desarrollan ocho algoritmos en Matlab © (uno por cada modelo y para cada tipo de función).

El conjunto de datos de la función lineal se presenta en la figura 2.

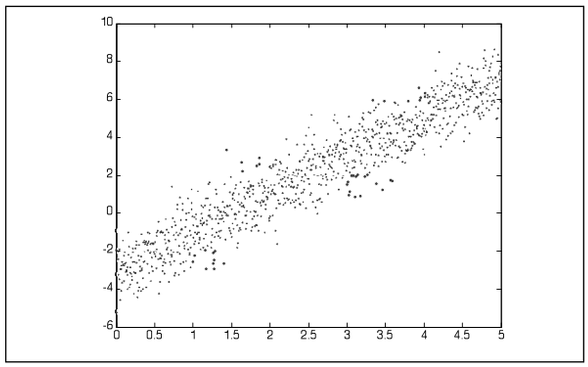

Figura 2. Datos de partida para estimar la función

La comparación de los cuatro modelos se presenta en la tabla 1.

Método				$\frac{\lvert g - f(x)\rvert}{M}$
Descenso de gradiente con tasa 1 … tasa 2 … tasa 3 … tasa 4	1	200 45 32 --	[1,9848-2.9370] [1,9901-2.9539] [1,9788-2.9181] No converge	0,0240 0,0204 0,0279 ---
Descenso modificado tasa 1 … tasa 2	1	40 19	[2,0120-2,9909] [1,9469 -2,9604]	0,0380 0,0940
D.G. con tasa de variable	2	5	[2,0288-3,0713]	0,00039
Gradiente conjugado con tasa variable	4	31	[1,9773 -2,9786]	0,0363

Los dos primeros métodos dependen de una o dos constantes que multiplican a la derivada de la función de coste. Para diferentes valores de constantes k, los modelos necesitan un número de iteraciones diferente para cumplir con un criterio de parada, y en algunos casos, pueden llegar a no converger (como en el método descenso de gradiente con tasa 4). Los dos últimos modelos tienen una tasa de aprendizaje que no es introducida por el usuario, sino que es calculada en línea por el algoritmo, por medio de un método de optimización.

En las figuras 3 y 4 se presenta la convergencia de los modelos de descenso de gradiente y gradiente modificado, para el caso de valores iniciales en [4 -0.5].

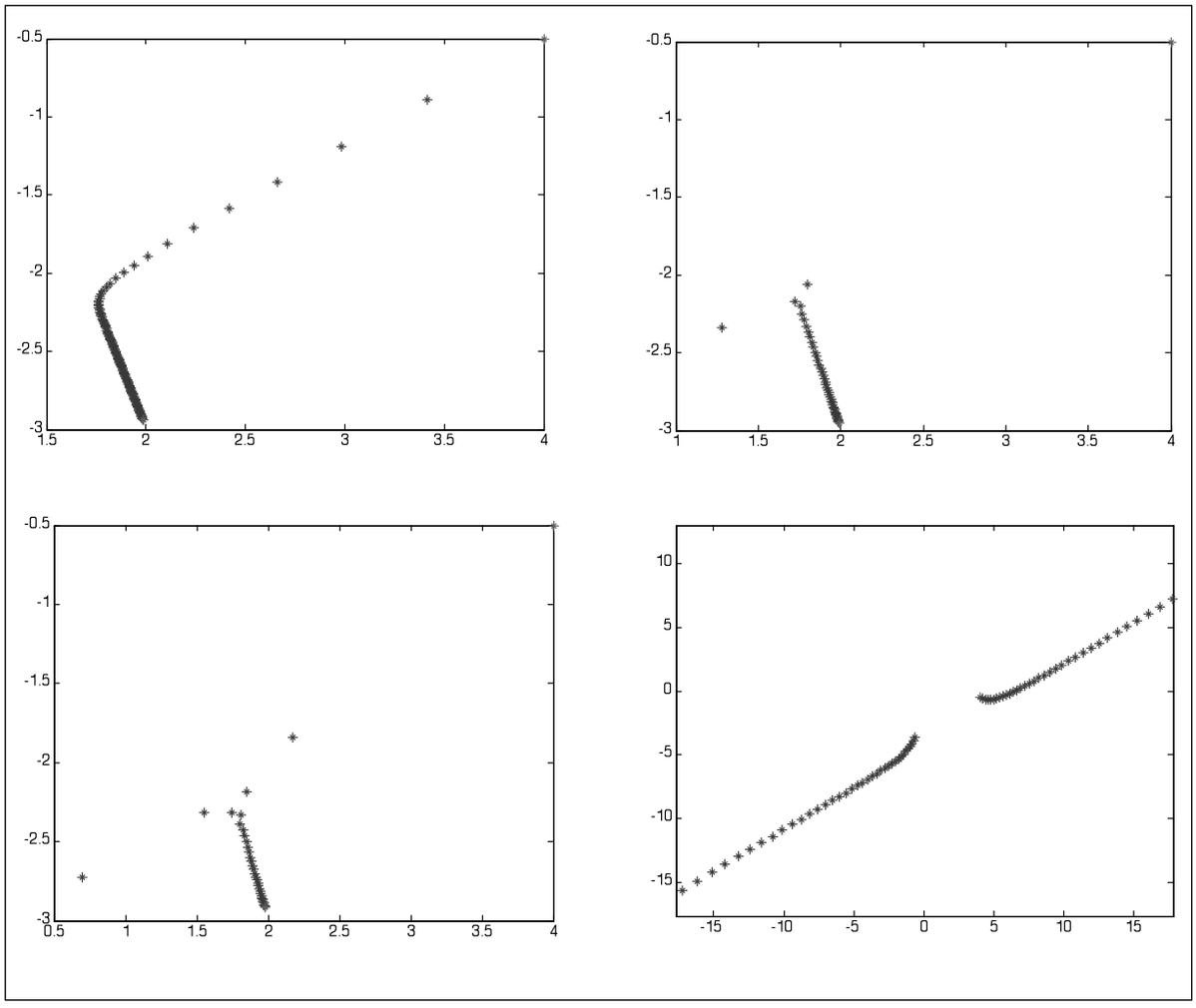

Figura 3. Convergencia descenso de gradiente $\boldsymbol{b}$ $\boldsymbol{d}$ **.**
De a) a d) la constante del modelo aumenta

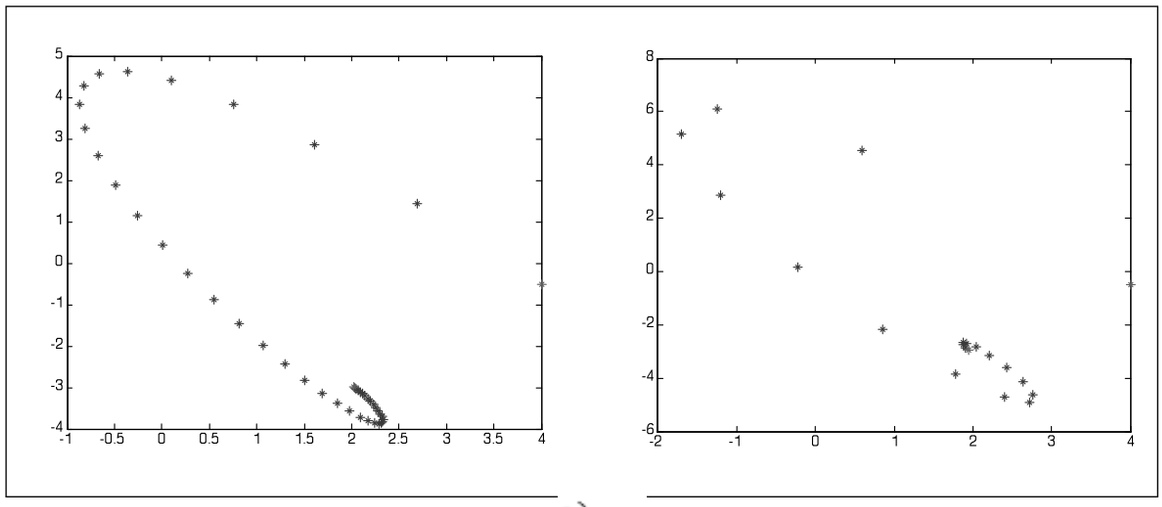

Figura 4. Convergencia descenso de gradiente modificado $b)$.
De a) a b) la constante del modelo aumenta.

En la figura 5 se presenta la convergencia del algoritmo de gradiente con tasa de aprendizaje variable.

Al comparar la figura 5a con las figuras 3 y 4, se aprecia que las oscilaciones alrededor de la solución y el tamaño del paso se ajustan mejor. En la figura 5b se presentan los datos y la función estimada por el método de descenso de gradiente con tasa de aprendizaje variable.

En la figura 6 se presentan los resultados para la función cuadrática, con número de iteraciones fijas igual a 20.

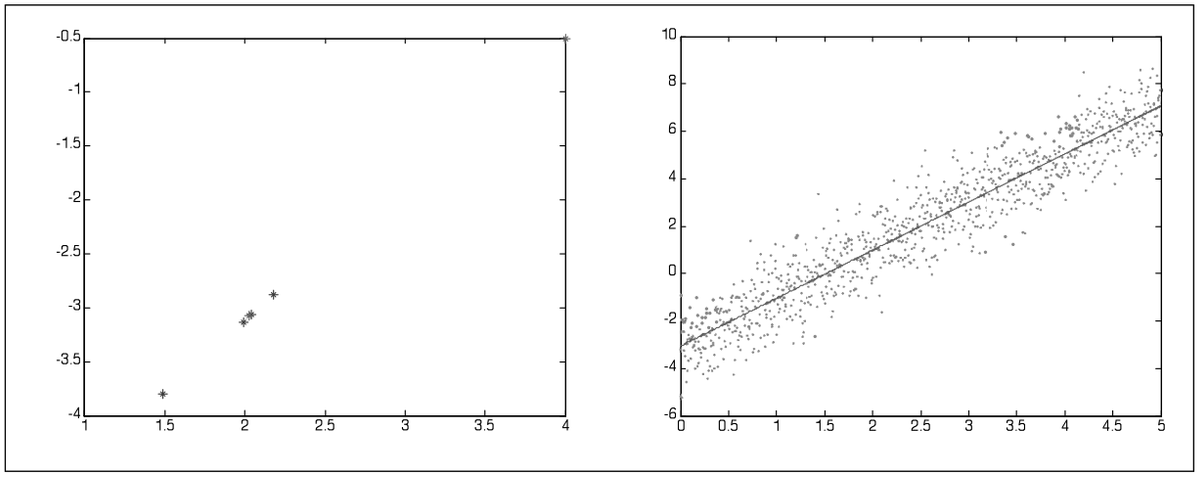

Figura 5 $b)$. Descenso de gradiente tasa de aprendizaje variable: a) convergencia, b) grafica de su solución en comparación con los datos de partida.

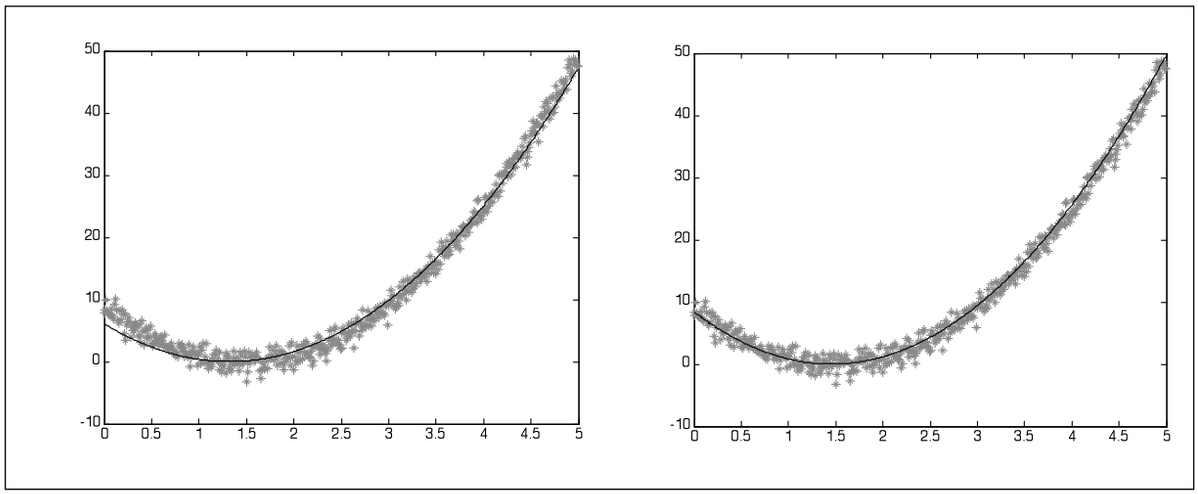

Figura 6 $\boldsymbol{b}$ $\boldsymbol{d}$ .Datos de partida y estimación: a) descenso de gradiente, b) gradiente modificado, c) descenso con tasa variable, d) gradiente conjugado

4. Conclusiones

Aunque el método de descenso de gradiente es el más sencillo en términos de programación, resulta ser el menos inteligente a la hora de adaptarse al problema, ya que su convergencia depende en gran medida de la configuración de su constante de paso. El valor de la constante cambia significativamente la convergencia del algoritmo y lo hace poco atractivo a la hora de seleccionar un modelo que sea auto-configurable.

La modificación del método de descenso de gradiente, basada en el intercambio de las derivadas que se utilizan en el cálculo del paso, produce mejores resultados que el método de descenso de gradiente tradicional, pero tiene la misma desventaja de su predecesor en cuanto a la definición de sus constantes.

En el caso de la función lineal, los resultados obtenidos con el método de gradiente conjugado son ligeramente inferiores (en términos de convergencia) al método de descenso con tasa de aprendizaje, pero son superiores a los métodos de descenso de gradiente tradicional y su versión modificada. En el caso de la función cuadrática, superan a todos los métodos con igual número de iteraciones, pero son muy similares al descenso de gradiente con tasa variable.

El método de descenso con tasa de aprendizaje variable se constituye en el mejor en términos de la simplicidad, ya que puede arrojar resultados muy parecidos e incluso mejores que el del método de descenso de gradiente, pero es más sencillo a la hora de programarlo. Por esta razón, a partir de los resultados de este trabajo, se recomienda este método a la hora de estimar funciones a partir de un conjunto de datos dispersos.

REFERENCIAS

[1] W. Fritz, "Sistemas inteligentes artificiales", en Sistemas inteligentes y sus sociedades, 2010. Consultado en: http://intelligent-systems.com.ar/intsyst/indexSp.htm#Back3

[2] S. Bermejo, "Aprendizaje", en Desarrollo de robots basados en el comportamiento, 1a ed., UPC, cap. 4, pp. 147-52, 2003.

[3] G. Harman y S. Kulkarni, "Induction and Simplicity," en Reliable Reasoning, MIT Press, cap. 3, pp. 55-76, 2007.

[4] P. Grunwald, J. Myung y M. Pitt, Advances in Minimum Description Length: Theory and applications, MIT Press, cap. 1, pp. 3-40, 2005.

[5] G. Hernández, "Métodos clásicos de optimización para problemas no-lineales sin restricciones", 2006, pp. 1-14. Consultado en https://www.u-cursos.cl/ingenieria/2010/2/MA3701/1/material_docente/bajar?id_material=329674.

[6] D. Ginestar, Métodos de gradiente: precondicionadores, Universidad Politécnica de Valencia, 2009.