

Reconocimiento de firmas off-line mediante máquinas de vectores de soporte

Off-line signature recognition using support vector machines

Jabbour, Georges* ; Márquez, Renny y Ruiz, Lisbeth

Departamento de Investigación de Operaciones. Facultad de Ingeniería. ULA.
Mérida, 5101, Venezuela
*jabbour@ula.ve

Maldonado, Luciano

Facultad de Ciencias Económicas y Sociales. ULA.
Instituto de Estadística Aplicada y Computación. IEAC.
Mérida, 5101, Venezuela

Recibido: 22-07-2009

Revisado: 02-03-2010

Resumen

En este trabajo se evaluó el funcionamiento de las Máquinas de Vectores de Soporte (MVS) en el reconocimiento de firmas estáticas, con el uso de la extracción de características como método de construcción de patrones. Se utilizaron dos arquitecturas diferentes: MVS puras y MVS integradas con una Red Neuronal Artificial perceptrónica multicapa (MVS/RNA) para interpretar los resultados de las MVS. Los mejores resultados se observaron con el enfoque de clasificación multivaluada 1-v-r. En ambas arquitecturas se obtuvo una tasa de aciertos similar. La ventaja práctica de usar la MVS/RNA consistió en disminuir el error de confundir un firmante con otro: cuando el modelo clasifica erróneamente a un firmante, en lugar de reconocerlo como uno distinto, lo clasifica como desconocido.

Palabras clave: Clasificación, máquinas de vectores de soporte, redes neuronales artificiales, reconocimiento de firmas.

Abstract

This paper presents two approaches for static signature recognition using Support Vector Machines (SVM): pure SVM and SVM integrated with a multilayer perceptron Artificial Neural Network (SVM/ANN) to map the results of the SVM. Feature extraction as a pattern construction method was adopted. Best results were obtained with the 1-v-r multivaluated classification method. The rate of correctly identified signers was similar for both architectures. A practical advantage of SVM/ANN architecture was decreasing the error of confusing the actual signer with another one: when the model misclassified a signer, instead of classifying it as a wrong signer, the proposed architecture recognized it as unknown.

Key words: Classification, support vector machines, artificial neural networks, signature recognition.

1 Introducción

La identificación de firmas manuscritas es un área de investigación activa. Existen dos principales campos en esta área: verificación de firmas y reconocimiento de firmas. El problema del reconocimiento de firmas consiste en identificar al autor de una firma. Para ello, se consulta una base de datos de firmas para establecer la identidad del firmante da-

do. En cambio, en la verificación se examina una firma para decidir si realmente pertenece a una persona en particular. En este caso, la salida es la aceptación de la firma o el rechazo de la misma (Frias-Martínez et al., 2006).

Dentro del campo del reconocimiento de formas, el reconocimiento de firmas es una de las tareas más difíciles de abordar, no solo por la gran variabilidad asociada al estilo y la forma de la escritura manual de cada persona,

sino también por el solapamiento en la trayectoria del trazado, la interconexión de los caracteres que conforman las frases y las características del bolígrafo utilizado para realizar la escritura, entre otros (Rincón, 2008).

Por otro lado, el reconocimiento de firmas puede realizarse usando técnicas on-line o mediante el uso de técnicas off-line (Rohlík, 2003). Los sistemas on-line requieren la presencia física del firmante y utilizan características dinámicas para la verificación de la firma tales como la inclinación del bolígrafo, la presión ejercida en cada momento del trazo, la velocidad del trazo, la secuencia de direcciones tomadas por el trazo, etc. (Huang y Yan, 2002; Jain et al., 2002; Fang et al., 2003; Justino et al., 2004; Toscano et al., 2005; Frías-Martínez et al., 2006; Fierrez et al., 2007). Para obtener este tipo de características se necesitan dispositivos especiales, por ejemplo, tableros digitalizadores y bolígrafos electrónicos. Por otro lado, los sistemas off-line parten de la firma estampada en un documento, y tras escanear la imagen o tomarle una foto digital, se extraen características geométricas de la firma para verificar la identidad del firmante (Rohlík, 2003). Cada enfoque abarca un conjunto diferente de aplicaciones; este artículo se centra exclusivamente en el reconocimiento de firmas off-line.

En este sentido, han sido aplicadas diferentes técnicas a la verificación y reconocimiento de firmas, tanto off-line como on-line, entre las cuales se encuentran la lógica difusa (Hanmandlu et al., 2005), redes neuronales artificiales (Baltzakis y Papamarkos, 2001; Frías-Martínez et al., 2006), algoritmos genéticos (Xuhua et al., 1996; Ramesh y Narasimha, 1999), modelos ocultos de Markov (Justino et al., 2004; Fierrez et al., 2007; Rincón, 2008), máquinas de vectores de soporte (Justino et al., 2004; Lv et al., 2005; Frías-Martínez et al., 2006), modelos deformables activos (Pavlidis et al., 1994), redes bayesianas (Xiao y Leedham, 2002) y modelos híbridos (Quek y Zhou, 2002; Taylan y Canan, 2009). No obstante, las máquinas de vectores de soporte figuran como una herramienta prometedora en este tipo de aplicaciones, dado su buen desempeño en la solución de problemas complejos, incluyendo el reconocimiento de patrones en problemas de clasificación de alta dimensionalidad, como reconocimiento de imágenes y de señales de voz, entre otros (Silva et al., 2007; Liu et al., 2008). Por tal razón, el objetivo de este artículo es evaluar el funcionamiento de las máquinas de vectores de soporte (MVS) para la identificación de firmas, y adicionalmente, estudiar la ganancia de incorporar una red neuronal artificial (RNA) al proceso de identificación.

2 Máquinas de vectores de soporte

Las máquinas de vectores de soporte (MVS) surgieron como un método de clasificación basado en la teoría de minimización del riesgo estructural de Vapnik. En la actualidad, tienen numerosas aplicaciones debido a su versatilidad y a sus prestaciones (Gunn, 1998).

Para clasificar con las MVS, se inicia con una etapa de aprendizaje, la cual consiste en encontrar el hiperplano $h(x_i) = \omega^T x_i + b = 0$ que mejor separe un conjunto de datos $\{(x_1, y_1); (x_2, y_2); \dots; (x_n, y_n)\}$, en donde $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]^T$ es el i -ésimo patrón ($i=1, \dots, n$), caracterizado por d variables; y y_i es la clase a la cual pertenece el i -ésimo patrón ($y_i \in \{-1, 1\}$). Además, $\omega = [\omega_1, \omega_2, \dots, \omega_d]^T$ es el vector director de $h(x)$, y b es un escalar que permite ubicar a $h(x)$ en el sitio de mayor conveniencia, en donde se logre la mayor separación de las clases. Geométricamente, el hiperplano $h(x)$ debe tener una dirección y una posición tales que se maximice su distancia al punto más próximo de cada clase. Una consecuencia de lo antes expuesto es que $h(x)$ será equidistante a estos puntos (ver Fig. 1).

Según la teoría de (Vapnik, 1995), el separador lineal $h(x)$, que maximiza el margen (dos veces la distancia al punto más próximo de cada clase) es el que da la mayor capacidad de generalización, es decir, la capacidad de clasificar correctamente a patrones que no forman parte del conjunto de entrenamiento (ver Fig. 1).

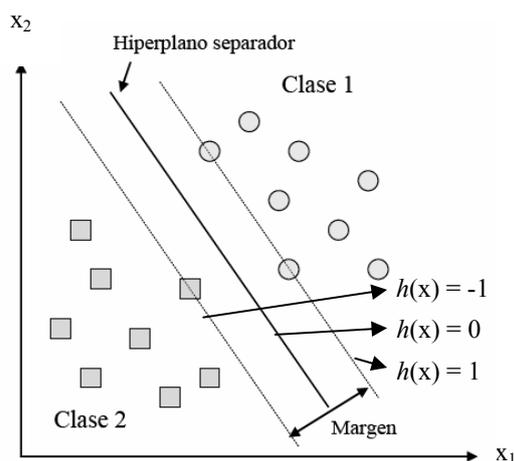


Fig. 1. Separación de un conjunto de datos mediante MVS

Para hallar ω y b , y por ende $h(x)$, es necesario resolver un problema de optimización, basado en un modelo de programación cuadrática. Los parámetros de este modelo se obtienen a partir de los patrones de entrenamiento. La solución del problema de optimización debe satisfacer las restricciones que se describen a continuación.

Los puntos más cercanos al hiperplano, denominados vectores de soporte, cumplen lo descrito en Ec. (1) y Ec. (2), mientras que el resto de los patrones, cumplen lo reflejado en Ec. (3) y Ec. (4), ya que no puede haber datos del conjunto de aprendizaje dentro del margen, es decir, no puede ocurrir que $|h(x_i)| < 1$ para estos datos.

$$h(x_i) = 1, \quad \text{si } y_i = 1 \quad (1)$$

$$h(x_i) = -1, \quad \text{si } y_i = -1 \quad (2)$$

$$h(x_i) > 1, \quad \text{si } y_i = 1 \quad (3)$$

$$h(x_i) < -1, \quad \text{si } y_i = -1 \quad (4)$$

Las condiciones descritas se implementan matemáticamente mediante la siguiente restricción:

$$y_i h(x_i) \geq 1 \tag{5}$$

Entonces, para cada patrón de entrenamiento, el modelo incluirá una restricción como la presentada en la Ec. (5).

Es importante destacar que el modelo de optimización que originalmente contiene restricciones se debe llevar a un modelo irrestricto, a través de multiplicadores de Lagrange.

Por lo tanto, para cada patrón de entrenamiento habrá un multiplicador de Lagrange ($\alpha_i, i=1, \dots, n$), que en la solución del problema indicará si el patrón está sobre el margen (y por ende, es un vector de soporte) o no. Aquellos patrones que se encuentren sobre los márgenes deben poseer un α distinto de cero, o equivalentemente, los puntos que no son vectores de soporte deben estar asociados a un $\alpha = 0$, ya que sus restricciones no se cumplen en la frontera. Al resolver el problema de optimización, se obtiene la siguiente solución:

$$w = \sum_{i=1}^n y_i \alpha_i x_i \tag{6}$$

$$h(x) = w^T x + b = \sum_{i=1}^n y_i \alpha_i x_i^T x + b \tag{7}$$

$$b = -\frac{1}{2} \left[\max_{y_i=-1} (w^T x_i) + \min_{y_i=1} (w^T x_i) \right] \tag{8}$$

En la Ec. (7), que refleja el hiperplano, se puede observar que éste solo depende de los vectores de soporte, ya que el resto de los puntos cumplen que $\alpha = 0$. Esto significa que si se eliminan todos los puntos que no son vectores de soporte y se vuelve a construir y resolver el modelo de optimización, se obtendría la misma solución.

Para un nuevo patrón x , la MVS proporcionará su clase según la regla de clasificación $\text{Clase}(x) = \text{signo}(h(x))$, es decir, se evalúa x en $h(x)$, y el signo del resultado es directamente la clase de x . Esta regla se utiliza para validar el modelo, evaluando en ella cada uno de los patrones de validación, que son datos que no fueron utilizados durante el entrenamiento. Para cada patrón de este conjunto, se compara la salida que genera la MVS con su verdadera clase, con lo cual se obtiene la tasa de aciertos del modelo.

Todo lo planteado hasta el momento es posible sólo si los patrones de entrenamiento son linealmente separables, ya que de lo contrario, el modelo de optimización tendrá un espacio de soluciones vacío. Sin embargo, en la práctica, es poco frecuente encontrar clases linealmente separables, y cuando no tengan esta característica, existen dos alternativas para resolver el problema: utilizando una transformación de los datos a un nuevo espacio de entrada y/o utilizando margen blando.

2.1 Función kernel

Cuando los datos no son linealmente separables, existe la posibilidad de transformarlos a un espacio τ de mayor dimensión (el espacio de características), en el que los puntos sí pueden ser separados por un hiperplano (Borges, 1998) (ver Fig. 2). Para ello, se utiliza una función $\phi \in \tau$, tal que:

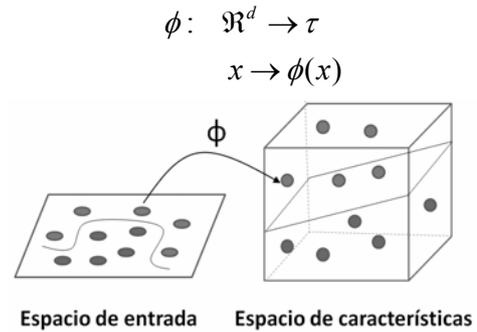


Fig. 2. Transformación de los datos de entrada a un espacio de mayor dimensión

La frontera de decisión resultante en el espacio de entrada ya no será lineal.

La función que se emplea para realizar esta transformación se llaman función núcleo o kernel, y representa el producto vectorial de 2 patrones en el espacio de características. Matemáticamente, esta función se define como $K(x_i, x) = \langle \phi(x_i), \phi(x) \rangle$, donde $\phi(x) = \phi(X_1, \dots, X_d) = [f_1(X_1, \dots, X_d); \dots; f_h(X_1, \dots, X_d)]$, y $h > d$.

Entonces, para adaptar la Ec. (7) al caso no lineal, se sustituye el producto vectorial del espacio de entrada ($x_i^T x$) por el producto del espacio de características que se corresponde con la función núcleo, $K(x_i, x)$, obteniendo:

$$h(x) = \sum_{i=1}^n y_i \alpha_i K(x_i, x) + b \tag{9}$$

2.2 Margen blando

En general, puede que no sea posible lograr la separabilidad lineal de los datos, ni en el espacio de variables originales ni en un espacio de mayor dimensionalidad, inducido por alguna función núcleo.

Para solventar este tipo de inconvenientes, es necesario añadir nuevos elementos al modelo de optimización, con lo cual se "ablandan" los márgenes. Este artificio consiste en la introducción de variables de holgura no negativas (denotadas con la letra S), que permiten que las restricciones no se cumplan de manera estricta. Existirá una variable de holgura (S_i) asociada a cada patrón de entrenamiento (x_i), que indica si el patrón ha violado o no el margen, y la medida

en la que lo ha hecho.

Adicionalmente, se introduce un término de regularización en la función objetivo del modelo de optimización, el cual depende de las variables de holgura. Este término establece un compromiso entre el margen del hiperplano y la magnitud de las variables de holgura. El término de regularización también incluye una constante C , llamada coeficiente de penalización, que determina la flexibilidad del margen blando. Valores grandes de C obligan a tener pocas variables de holgura diferentes de 0 (o con valores grandes) en la solución óptima del modelo, y a medida que C toma valores más pequeños, entonces el margen blando se vuelve más flexible. Esta constante debe fijarse a priori para hacer uso del margen blando (Vapnik, 1995 y Gunn, 1998).

2.3 Clasificación multivaluada

Tal como se ha visto, una MVS es un clasificador binario. No obstante, la clasificación realizada en esta investigación involucra más de dos clases, puesto que cada firmante representa una categoría distinta, de manera que es necesario adaptar las MVS para que permitan realizar una clasificación multiclase.

(Bahón, 2001) explica que una de las soluciones para resolver esta clasificación multiclase es convertirla en varias clasificaciones binarias. Algunos de los esquemas que existen para descomponer el problema general en dicotomías son (Bahón, 2001; Lingras y Butz, 2007):

2.3.1 Clasificación 1-v-r

El término 1-v-r proviene del inglés *one-versus-rest*, que significa "uno contra el resto". Un k -ésimo nodo de dicotomía dentro de una arquitectura de clasificación multiclase se denomina 1-v-r si es una MVS entrenada con etiquetas positivas $y_i = 1$ para los ℓ_k patrones de entrenamiento de la k -ésima clase etiquetada θ_k , y con etiquetas negativas $y_i = -1$ para los $\ell - \ell_k$ patrones de las $K-1$ clases restantes (Bahón, 2001).

Un clasificador 1-v-r es entrenado para separar una clase de las $K-1$ restantes. El método estándar de descomposición de un problema general de clasificación multivaluada, a partir de ejemplos en dicotomías, consiste en situar $L = K$ clasificadores binarios, tantos como clases, del estilo 1-v-r en paralelo.

2.3.2 Clasificación 1-v-1

El término 1-v-1 proviene del inglés *one-versus-rest*, que significa "uno contra el resto". Un k -ésimo nodo de dicotomía dentro de una arquitectura de clasificación multiclase se denomina 1-v-1 si es una MVS entrenada con etiquetas positivas $y_i = 1$ para los ℓ_k patrones de entrenamiento de la k -ésima clase etiquetada θ_k , y con etiquetas negativas $y_i = -1$ para los $\ell - \ell_k$ patrones de las $K-1$ clases restantes (Bahón, 2001).

El método de descomposición asociado a este tipo de dicotomías consiste en situar $L = K(K-1)/2$ clasificadores binarios del estilo 1-v-1 en paralelo, cada nodo siendo entrenado por sólo dos de las K clases implicadas en la clasificación múltiple.

3 Redes neuronales artificiales

Una red neuronal artificial puede ser definida de múltiples maneras, una de ellas establece que una RNA es un modelo matemático cuya construcción se lleva a cabo mediante un proceso que, en parte, imita el funcionamiento de las redes neuronales biológicas (Gupta et al., 2003). Este modelo matemático consta de una o más variables de entrada (o variables independientes), y de una o más variables de salida (o variables dependientes), por lo tanto, una RNA puede ser usada para modelar fenómenos en los cuales existe alguna respuesta (variable explicada) que depende de algún conjunto de factores (variable(s) explicativa(s)). Esto implica que para modelar un fenómeno mediante una RNA, debe existir una relación matemática (lineal o no lineal) entre la(s) variable(s) explicativa(s) y la(s) variable(s) explicada(s), pues mientras mayor sea esta relación, mejor será el desempeño de la RNA (Gupta et al., 2003). En el caso bajo estudio, la salida de la RNA corresponde al identificador del firmante, mientras que las entradas son las características de la firma, que se describen en la siguiente sección.

Desde el punto de vista computacional, una RNA consiste en un conjunto de neuronas artificiales interconectadas entre sí. Además, las neuronas se encuentran, generalmente, agrupadas en capas o niveles. La manera como se pueden dar las conexiones entre las diversas neuronas de la red, es lo que comúnmente se denomina la topología de la red (Jain et al., 1996). Existen diversas topologías, sin embargo, sólo se hará referencia a aquella en la cual cada neurona de una capa dada, está conectada con todas las neuronas de la siguiente capa, tal como se representa en la Fig. 3, (Gupta et al., 2003).

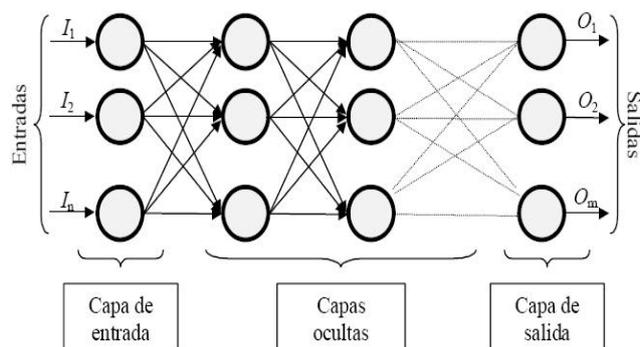


Fig. 3. Red perceptrónica multicapa

Este tipo de RNA se conoce como Redes Perceptrónicas Multicapa (RPM), y fue el tipo de red utilizado en esta investigación. Tal como se puede apreciar en la Fig. 3, las

neuronas (nodos) están agrupadas en capas verticales. Una RNA puede tener 2 o más capas, y cada capa puede tener cualquier cantidad de neuronas, dependiendo del problema que se esté tratando. La primera capa (de izquierda a derecha) es la capa de entrada, la última es la capa de salida, y las restantes son las capas ocultas.

A su vez, cada nodo de la RNA está formado por los elementos de la Fig. 4. (Gupta et al., 2003).

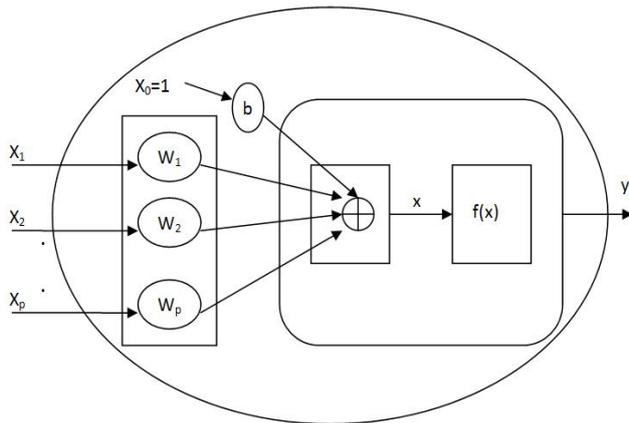


Fig. 4. Modelo de una Neurona Artificial

En la Fig. 4, X_i es la i -ésima entrada de la neurona, W_i es el peso sináptico asociado a la i -ésima entrada y b es el sesgo de la neurona (constante). La salida de la neurona es $y=f(x)$, siendo f una función lineal o no lineal denominada función de transferencia, y x la suma ponderada de las entradas y el sesgo, tal como se muestra a continuación:

$$x = \sum_{i=1}^p W_i X_i + b \quad (10)$$

Por lo tanto, puede observarse que una neurona artificial es un modelo matemático, en donde la salida no es más que una transformación matemática (o función) de las entradas. A su vez, esto implica que la RNA es también un modelo matemático, dado por las neuronas de la capa de salida.

Luego de plantear la arquitectura de la RNA (entradas, número de capas, número de neuronas para cada capa y la función de transferencia de cada capa), se procede con el entrenamiento de la misma. El entrenamiento de una RNA es un proceso mediante el cual se ajustan los pesos sinápticos y los sesgos de la red, con el propósito de que la RNA pueda llevar a cabo con mayor exactitud la tarea para la cual fue desarrollada. En esta investigación se utilizó el algoritmo de retropropagación del error, el cual ajusta (entrena) los pesos y sesgos de la red desplazándose en el sentido opuesto del gradiente de la función del error de entrenamiento con respecto a tales pesos y sesgos (Jain et al., 1996; Gupta et al., 2003).

4 Pre-procesamiento de firmas

Dado que se está trabajando con el enfoque de reconocimiento de firmas off-line, las firmas no son más que figuras plasmadas en un papel. Por lo tanto, para suministrarle firmas a una MVS es necesario seguir un proceso mediante el cual se transformen estos trazados en patrones compatibles con el formato del modelo de clasificación. Este proceso consta de 4 fases (Justino, 2004; Toscano et al., 2005): 1) digitalización de la firma mediante un escáner, para transformar el trazado en papel en una imagen digital (archivo en formato bmp); 2) adelgazamiento del trazado, para generar un nuevo trazado de 1 píxel de ancho, a lo que se denomina el esqueleto de la firma, que idealmente contiene toda la información relevante; 3) transformación de la imagen a una matriz de valores binarios; y 4) extracción de las características de la firma, a partir de la matriz obtenida en el paso anterior. El resultado de este proceso es un vector cuya cantidad de componentes coincide con la cantidad de entradas del modelo de clasificación.

Suponga que luego de digitalizar la firma, se obtiene una imagen rectangular de A píxeles de ancho y H píxeles de alto, en donde cada píxel puede tomar uno de los dos posibles colores: "blanco" o "negro". Luego, se adelgaza o esqueletiza la firma eliminando píxeles de la imagen hasta que quede un trazado de un píxel de grosor, es decir, se reducen todos los objetos en la imagen binaria a líneas, sin cambiar la estructura esencial de la imagen (Lv et al., 2005).

Para llevar a cabo la extracción de características de las firmas, éstas se dividen en cuadrículas y a cada una de ellas se le extraen ciertas características. La imagen se divide en $F \times C$ cuadrículas (F filas y C columnas). Cada cuadrícula contiene a su vez H/F filas y A/C columnas puesto que las firmas están almacenadas en una imagen de H píxeles de alto por A píxeles de ancho, como se muestra en la Fig. 5.

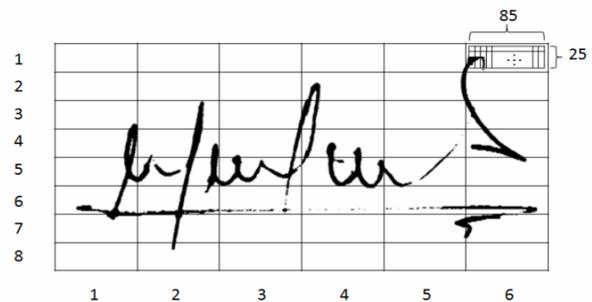


Fig. 5. Proceso de partición de una firma para $F=8$, $C=6$, $H=510$ y $A=200$

Luego de realizada la partición de la firma se proceden a extraer S características de cada celda, obteniéndose un vector de $F \times C$ valores por cada característica, para un total de S vectores. Finalmente, para cada firma se obtiene un vector de $S \times F \times C$ valores que corresponde al patrón asociado a la firma bajo análisis.

En esta investigación se utilizaron 4 características grafométricas estáticas y pseudodinámicas (Justino et al., 2004):

- Densidad de los píxeles (X_{DP}): número de píxeles dentro de la cuadrícula.

$$X_{DP} = \frac{\text{Número de Píxeles Negros}}{\text{Número de Píxeles}} \quad (11)$$

- Centro de gravedad relativo (X_{CG}): definido por la Ec. (13):

$$X_{CG} = \frac{\sqrt{a^2 + b^2}}{D} \quad (12)$$

en donde a y b son las coordenadas del centro de gravedad de los píxeles negros, y D es la longitud de la diagonal de la cuadrícula.

- Curvatura promedio del segmento (X_{CS}): para la extracción de esta característica se recorre cada cuadrícula y en cada píxel negro del segmento se analizan sus ocho píxeles vecinos, dependiendo de la celda vecina que tenga un píxel negro se coloca un valor de ángulo como se muestra en la Fig. 6.

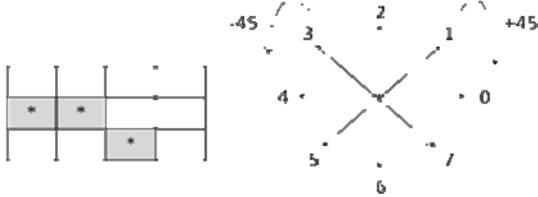


Fig. 6. Curvatura del segmento.

Así, la curvatura del segmento será la suma de todos los ángulos encontrados en el mismo entre el número de píxeles negros de la cuadrícula, como lo refleja la Ec. (13):

$$X_{CS} = \frac{\sum \text{Ángulos de los Píxeles Negros}}{\text{Número de Píxeles Negros}} \quad (13)$$

Por ejemplo, para el caso de la Fig. 6, y tomando en cuenta que las cuadrículas marcadas son píxeles negros, si se desea calcular la curvatura del segmento para el píxel ubicado en la fila 2 columna 2, se tiene que $X_{CS} = \frac{4 + 7}{3} = \frac{11}{3}$.

- Inclinación predominante (X_{IP}): la inclinación predominante es una medida de la orientación del segmento en la imagen. Por cada celda con píxel negro se analizan sus 8 vecinos, dicho píxel pertenece a un valor del conjunto de segmentos estructurales, como se muestra en la Tabla 1. A medida que se recorre la cuadrícula, se van incrementando los contadores correspondientes a dicho conjunto

de segmentos estructurales. Así, el símbolo con mayor ocurrencia será aquel que tenga el mayor contador. En las celdas vacías (píxeles blancos) todas las características entran en el contador 0 ya que existe una prioridad dentro del conjunto de elementos estructurales, clasificados como 1, 2, 3, 4 y 0.

Tabla 1. Elementos estructurales

Contador	Elementos estructurales
0	Otro caso
	1 0 1
1	1 x 1
	1 0 1
	1 1 1
2	0 x 0
	1 1 1
	1 1 0
3	1 x 1
	0 1 1
	0 1 1
4	1 x 1
	1 1 0

La x representa la celda en estudio. Los ceros y los unos indican los píxeles vecinos. El 1 representa un píxel en blanco. El 0 representa un píxel marcado.

5 Arquitectura de las MVS y RNA para el reconocimiento de firmas

5.1 Arquitectura MVS pura

El caso bajo estudio es un problema de clasificación multivaluada, en donde cada clase corresponde a un firmante. En este caso, la forma de utilizar las MVS depende del enfoque de clasificación con el que se esté trabajando (1-v-r ó 1-v-1).

5.1.1 Arquitectura MVS pura bajo el enfoque 1-v-r

Cuando se emplea el método 1-v-r, hay que crear K MVS, una para cada clase (una para cada firmante), tal como se muestra en la Fig. 7.

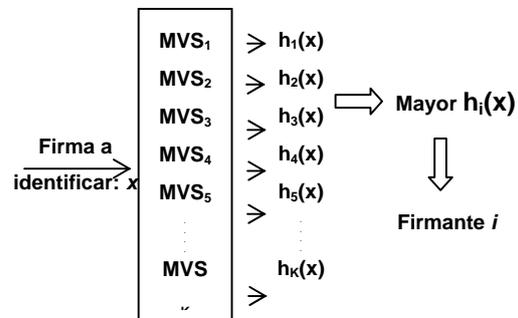


Fig. 7. Reconocimiento de firmas mediante MVS puras basadas en el método 1-v-r.

La i -ésima MVS se entrena con un conjunto de patrones dividido en 2 grupos: uno asociado al i -ésimo firmante (clase +1) y otro formado por patrones correspondientes a los $K-1$ firmantes restantes (clase -1). Para clasificar una nueva firma, se suministra el respectivo patrón a cada una de las K MVS (ya entrenadas), y luego sus resultados se interpretan de la siguiente manera:

- Firma correctamente clasificada: ocurre cuando la MVS que arroja el mayor valor de $h(x)$ corresponde a la verdadera clase de la firma. Una condición para aceptar este resultado es que $h(x)$ debe superar cierto umbral preestablecido.
- Firma mal clasificada: ocurre cuando la MVS que arroja el mayor valor de $h(x)$ (siendo $h(x)$ mayor que cierto umbral) no corresponde a la verdadera clase de la firma.
- Firma indeterminada: ocurre cuando se presenta una de las siguientes situaciones: 1) ninguna de las MVS ha generado un valor positivo para $h(x)$, es decir, la firma no se corresponde con ninguna clase; o 2) la firma ha sido clasificada como positiva por más de una MVS, en cuyo caso se selecciona la más positiva, pero ésta no supera el umbral preestablecido

5.1.2 Arquitectura MVS pura bajo el enfoque 1-v-1

Cuando se utiliza el método 1-v-1, y cuando se tienen K clases, la cantidad de MVS es igual a:

$$L = \binom{K}{2} = \frac{K(K-1)}{2} \quad (14)$$

Bajo este enfoque, cada MVS va a servir de modelo discriminante entre dos clases particulares, de manera que el resultado de una MVS siempre va a ser una de las 2 clases (a diferencia del enfoque anterior, en donde una MVS no necesariamente arroja una clase como resultado), ya que ahora no se emplea a $h(x)$ sino al signo de $h(x)$ como resultado de cada MVS. Cabe destacar, que cada clase va a estar representada en $K-1$ MVS, pues por cada posible pareja de clases debe haber una MVS. En definitiva, este enfoque funciona como un sistema de votación, en donde cada MVS es un votante, y cada clase es un candidato. Cuando se va a clasificar una nueva firma, ésta se debe someter a cada una de las L MVS, y por lo tanto, se obtienen L votos que dan lugar a los siguientes casos:

- Firma correctamente clasificada: cuando la clase que recibe la mayor votación coincide con la verdadera clase de la firma.
- Firma mal clasificada: cuando la clase que recibe la mayor votación no coincide con la verdadera clase de la firma.
- Firma indeterminada: cuando al menos dos clases están empatadas con la mayor votación.

Para los datos (firmas) de los que se disponía en esta investigación, se realizaron pruebas piloto con el fin de determinar cuál de estas dos opciones ofrece los mejores resultados. En este sentido, fue seleccionado el método 1-v-r, tanto para los experimentos basados en MVS puras como para el modelo híbrido MVS/RNA, que se describe a continuación.

5.2 Modelo MVS/RNA

Este modelo también utiliza las MVS, cuyo entrenamiento y uso son idénticos a lo planteado en el método 1-v-r de la sección 5.1.1, no obstante, la identificación la hace una red neuronal artificial (RNA) en lugar de seleccionar la firma asociada a la mayor salida de las MVS. Esto se hace para evitar, al menos en la medida de lo posible, que algunas MVS respondan positivamente ante patrones que no les corresponden, como efectivamente suele ocurrir. Entonces, la RNA se encarga de aprender cómo responde cada MVS para cada firmante, y así corregir los errores que ellas presentan. Por lo tanto, las salidas de las MVS no son interpretadas por el usuario, sino que le son suministradas a la RNA, es decir, el patrón de entrada de la RNA es precisamente un vector formado por las salidas de las K MVS. A su vez, la capa de salida de la RNA está compuesta por K neuronas, cuyas salidas se interpretan de forma similar a como se hacía con las salidas de las MVS puras, es decir, si la i -ésima neurona es la que genera la mayor salida entonces el firmante que se está intentando reconocer es también el i -ésimo. Cabe destacar que la mayor de las salidas de la RNA debe superar cierto umbral preestablecido, de forma similar a como se hacía con las MVS. La Fig. 8 ilustra este proceso.

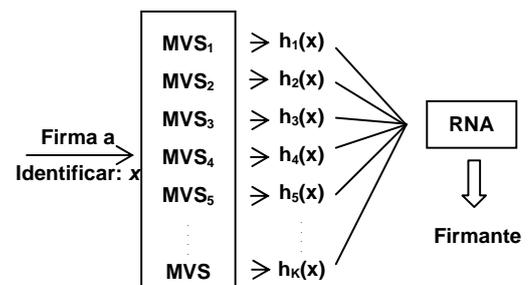


Fig. 8. Proceso de identificación del firmante en la arquitectura MVS-RNA

Lo antes expuesto implica que el entrenamiento del modelo requiere de dos fases, una para entrenar a las MVS y otra para entrenar a la RNA. Nótese que los parámetros de las MVS en este modelo coinciden con los parámetros obtenidos en el enfoque MVS puro. Entonces, los patrones de entrenamiento de la RNA se obtienen al evaluar cada firma en cada MVS, es decir, el i -ésimo patrón de entrenamiento de la RNA se obtiene al evaluar al i -ésimo firmante en las K MVS. Por otro lado, como la RNA es entrenada a través de un algoritmo de aprendizaje supervisado, es nece-

sario conocer la salida deseada para cada patrón. Tal como se mencionó en el párrafo anterior, la RNA posee K salidas, y por lo tanto, la salida deseada para cada patrón es un vector de tamaño K . Este vector debe poseer en $K-1$ componentes el valor cero, y el valor 1 uno en aquella posición que corresponda al firmante asociado al patrón de entrada.

6 Experimentos

Los datos que se describen a continuación fueron utilizados de la misma manera en las pruebas de ambos tipos de modelo (MVS puras y MVS/RNA). Se recopilieron firmas de 17 individuos, de los cuales, 10 de ellos (a lo que se llamará el grupo A) realizaron 22 réplicas cada uno, y los 7 restantes (grupo B) realizaron 3 réplicas cada uno, haciendo un total de 241 firmas. Las firmas del grupo A se utilizaron de la siguiente manera: se utilizaron 180 firmas (18 de cada firmante) para el entrenamiento de modelos y 40 para la validación de los mismos. Las 21 firmas del grupo B se utilizaron para evaluar la respuesta de los modelos cuando se les presentan firmas de autores desconocidos.

Las firmas digitalizadas se almacenaron en archivos de formato *bmp*, con un tamaño de 510 píxeles de ancho por 200 píxeles de alto, a las cuales se les aplicó el respectivo adelgazamiento, según lo descrito en la sección 4. Luego, las imágenes se dividieron en 48 cuadrículas (8 filas y 6 columnas), ya que un número mayor de cuadrículas incrementaría el costo computacional a un nivel difícil de manejar. Por lo tanto, cada cuadrícula contiene 25 filas y 85 columnas.

Finalmente, se extrajeron las cuatro características de cada cuadrícula, obteniéndose un vector de 48 valores por cada característica. Así, para cada firma se obtiene un vector de 192 valores que es el patrón de entrada del modelo clasificador.

6.1 Entrenamiento de la MVS pura

Los patrones de entrenamiento están clasificados en 10 categorías (10 firmantes), y por lo tanto, se entrenaron 10 MVS, una por cada firmante (dado que se empleó el método 1- v - r , descrito en la sección 5), de manera que para cada MVS, la clase +1 corresponde a un firmante en particular, y la clase -1 corresponde al resto de los firmantes. El tamaño del conjunto de entrenamiento se varió para comparar resultados: en un caso se utilizaron las 180 firmas del conjunto de entrenamiento y en otro caso sólo 99 firmas del mismo conjunto. En el primer caso se utilizaron 18 firmas de cada firmante, esto es, la clase +1 se entrenó con 18 firmas mientras que la clase -1 se entrenó con 162 firmas. En el otro caso, para cada MVS se utilizaron las 18 firmas correspondientes a ese firmante y sólo 9 firmas de cada uno de los otros firmantes, es decir, la clase +1 se entrenó con 18 firmas y la clase -1 con 81 firmas.

Se realizaron un conjunto de pruebas, para cada una de las 10 MVS, considerando las mismas características: tipo de función kernel (base radial, polinómica), parámetros de las funciones kernel (σ), coeficiente de penalización del margen blando (C) y umbral. Estas pruebas permitieron delimitar los posibles valores para cada uno de los parámetros de la arquitectura MVS pura.

Los modelos obtenidos consideran como función kernel, la polinomial o la de base radial; los posibles valores de σ , como 1, 2 o 3; los posibles valores de C , como 100 o 10000; y el umbral como -0,25 o -0,01.

6.2 Entrenamiento del modelo MVS/RNA

La única diferencia entre las MVS obtenidas del enfoque puro y las utilizadas para este modelo es la variación del umbral, que para este caso toma los valores 0,35 ó 0,50.

En esta arquitectura se utilizó una red perceptrónica multicapa. Las características de esta red, obtenidas a partir de un conjunto de pruebas, incluyen 10 neuronas tanto en la capa oculta como en la capa de salida, como función de transferencia de la capa oculta la tangente hiperbólica, para la capa de salida la función lineal y como algoritmo de entrenamiento el backpropagation de regularización bayesiana.

7 Resultados

Para evaluar el desempeño de los modelos obtenidos durante la fase de entrenamiento, se utilizó el conjunto de validación, obteniendo así la salida de cada máquina o de cada red neuronal artificial, según corresponda, para la posterior identificación del firmante.

Tanto para la arquitectura RNA como para MVS/RNA, cuando el conjunto de firmas utilizadas en el entrenamiento fue de 180, se observó una mejora significativa, por lo cual, en las tablas de resultados, no se muestra lo correspondiente al conjunto de 99 firmas para el entrenamiento.

7.1 Validación de las MVS puras

Los resultados obtenidos al evaluar los modelos de las MVS puras, indican que el porcentaje de identificación correcta del firmante no varía con el cambio de C y es mayor en los modelos con una función kernel de base radial, cuya ecuación se muestra a continuación:

$$K(x, x') = \exp\left(\frac{-\|x - x'\|^2}{2\sigma^2}\right) \quad \sigma > 0 \quad (15)$$

Los resultados obtenidos al variar los demás parámetros se resumen en la Tabla 2. Las siglas FMC, FI y FBC, se refieren al número de firmas mal clasificadas, firmas indeterminadas y firmas bien clasificadas, respectivamente.

Las características de la MVS pura para la cual la tasa de aciertos es máxima (70,5%), son el uso de un umbral con -0,25 y un valor de σ de 2 ó 3. Sin embargo, con $\sigma = 2$, la tasa de firmas mal clasificadas es menor (23%).

Tabla 2. Resultados de las MVS puras

Entrena- miento	Umbral	σ	% de FMC	% de FI	% de FBC
180 firmas	-0,25	1	49,2	0,0	50,8
		2	23,0	6,6	70,5
		3	24,6	4,9	70,5
	-0,01	1	34,4	13,1	52,5
		2	9,8	27,9	62,3
		3	16,4	14,8	68,9

7.2 Validación de los modelos MVS/RNA

Al igual que para las MVS puras, los resultados obtenidos indican que el porcentaje de identificación correcta del firmante varía en forma insignificante con el cambio de C y es mayor en los modelos con una función kernel de base radial. Los resultados obtenidos al variar los demás parámetros se resumen en la Tabla 3.

Tabla 3. Resultados de los modelos MVS/RNA

Entrena- miento	Umbral	σ	% de FMC	% de FI	% de FBC
180 firmas	0,50	1	0,0	65,6	34,4
		2	4,9	27,9	67,2
		3	16,4	13,1	70,5
	0,35	1	0,0	65,6	34,4
		2	21,3	8,2	70,5
		3	29,5	4,9	65,6

El modelo MVS/RNA para el cual la tasa de aciertos es máxima (70,5%) corresponde a la asignación de un umbral de 0,50 con $\sigma = 3$, o un umbral de 0,35 con un $\sigma = 2$. Sin embargo, la primera combinación es mejor, ya que presenta una tasa menor de firmas mal clasificadas de 16,4%.

Otro resultado importante, es el hecho de que para un umbral de 0,50 y $\sigma = 2$, pese a que la tasa de aciertos es 67,2%, la tasa de firmas mal clasificadas se reduce drásticamente a un 4,9%, con una tasa de firmas indeterminadas de 27,9%.

En la literatura existen algunas referencias en donde se estudia el reconocimiento de firmas off-line que pueden ser utilizadas para comparar el trabajo realizado: (Pavlidis et al. 1994), utilizan modelos deformables activos para capturar características de la silueta de una curva, con una base de datos compuesta por 40 firmantes y 4 firmas por cada fir-

mante, obteniendo resultados que reflejan una tasa de firmas bien clasificadas de 70,8%, una tasa de firmas indeterminadas de 13,3% y una tasa de firmas mal clasificadas de 15,8%; (Baltzakis y Papamarkos 2001), presentan un sistema basado en dos etapas, con tres clasificadores de redes neuronales con la obtención de la distancia euclidiana para la primera etapa y un clasificador de base radial para una segunda etapa, para lo cual utilizaron una base de datos con 2000 firmas de 115 firmantes, con una tasa de acierto de 80,8%, con una tasa de firmas mal clasificadas de 18,2%, pero no consideran en ningún momento la posibilidad de hacer un estudio que incluya firmas indeterminadas; (Frias-Martínez et al. 2006), utilizaron datos de 38 firmantes y 6 firmas por firmante bajo el enfoque MVS, reportando una tasa de aciertos del 71,2%, con una tasa de firmas mal clasificadas de 28,8%, sin utilizar la concepción de firmas indeterminadas en el modelo.

Los resultados de esta investigación, similares a los de (Pavlidis et al, 1994), tienen la ventaja de que reportan tasas de indeterminación en el reconocimiento del firmante, a diferencia de lo realizado en (Baltzakis y Papamarkos 2001; Frías-Martínez et al. 2006), quienes sólo se enfocan en las firmas mal clasificadas. El aporte del presente trabajo es el uso de una RNA en conjunto con las MVS, lo cual produce una mejoría en el reconocimiento, reduciendo la tasa de firmas mal clasificadas.

8 Conclusiones y recomendaciones

Las máquinas de vectores de soporte (MVS) en el reconocimiento de firmas manuscritas proporcionaron resultados de identificación de hasta un 70,5% de exactitud.

Cuando se introduce una RNA al proceso de identificación, el tipo de error que se comete es diferente al que se obtiene utilizando sólo la MVS, como es el caso de la arquitectura MVS/RNA, en donde la tasa de firmas mal clasificadas se redujo, clasificando algunas como indeterminadas, observándose la ganancia de introducir la RNA en el proceso de identificación.

Para todos los modelos, se han utilizado los mismos parámetros función kernel, C , etc. Podría ocurrir que los parámetros que son adecuados para obtener el hiperplano de uno de esos modelos no lo sean para otro. Por esta razón, se podría estudiar si se pueden mejorar los resultados empleando distintos parámetros para cada modelo.

En determinados modelos, el alto porcentaje de error se debe a un número de firmas que han sido clasificadas como clase positiva en varios modelos. Se puede realizar una MVS específica para separar dos clases concretas entre las que el grado de confusión suele ser alto.

Las firmas utilizadas para el entrenamiento y la validación tienen el mismo tamaño. Se recomienda aumentar el número de firmantes y variar el número de firmas de cada uno, y estudiar si con estas modificaciones se afecta positiva o negativamente el modelo MVS/RNA. Asimismo, se recomienda aumentar la segmentación de las imágenes para

que cada celda o cuadrícula contenga más información sobre la firma en estudio.

Puede utilizarse en trabajos futuros la validación cruzada de varias particiones (k-fold cross validation) con el fin de obtener resultados robustos y más significativos.

Agradecimientos

Los autores de este artículo desean dar su agradecimiento al CDCHT de la Universidad de Los Andes Venezuela, por su apoyo a través de los proyectos E-252-06-02-B e I-1047-07-02-C.

Referencias

- Bahón C, 2001, Aprendizaje con máquinas núcleo en entornos de multclasificación Capítulo 4: Clasificación Multiclase con Maquinas de Soporte Vectorial, Tesis doctoral, Universidad Politécnica de Cataluña.
- Baltzakis H y Papamarkos N, 2001, A new signature verification technique based on a two-stage neural network classifier, *Engineering Applications of Artificial Intelligence*, Vol. 14, No. 1, pp. 95-103.
- Burges C, 1998, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, Vol. 2, No. 2, pp. 121-167.
- Fang B, Leung CH, Tang Y, Tse K, Kwok P y Wong YK, 2003, Off-line signature verification by the tracking of feature and stroke positions, *Pattern Recognition*, Vol. 36, No. 1, pp. 91-101.
- Fiérrez J, Ortega-García J, Ramos D y González-Rodríguez J, 2007, HMM-based on-line signature verification: Feature extraction and signature modeling, *Pattern Recognition Letters*, Vol. 28, No. 16, pp. 2325-2334.
- Frías-Martínez E, Sánchez A y Vélez J, 2006, Support vector machines versus multilayer perceptrons for efficient off-line signature recognition, *Engineering Applications of Artificial Intelligence*, Vol. 19, No. 6, pp. 693-704.
- Gunn S, 1998, Support vector machines for classification and regression, *Reporte Técnico*, Universidad de Southampton.
- Gupta M, Jin L y Homma N, 2003, *Static And Dynamic Neural Networks: From Fundamentals to Advanced Theory*, John Wiley & Sons, New York.
- Hanmandlu M, Yusof H y Krishna Madasu V, 2005, Off-line signature verification and forgery detection using fuzzy modelling, *Pattern Recognition: The Journal of the Pattern Recognition Society*, Vol. 38, No. 3, pp. 341-356.
- Huang K y Yan H, 2002, Off-line signature verification using structural feature correspondence, *Pattern Recognition*, Vol. 35, No. 11, pp. 2467-2477.
- Jain A, Mao J y Mohiuddin K, 1996, *Artificial Neural Networks: A Tutorial*, IEEE Computer, Vol. 29, No. 3, pp. 31-44.
- Jain A, Griess F y Connell S, 2002, On-line signature verification, *Pattern Recognition: The Journal of the Pattern Recognition Society*, Vol. 35, No. 12, pp. 2963-2972.
- Justino E, Bortolozzi F y Sabourin R, 2004, A comparison of svm and hmm classifiers in the off-line signature verification, *Pattern Recognition Letters*, Vol. 26, No. 9, pp. 1377-1385.
- Lingras P y Butz C, 2007, Rough set based 1-v-1 and 1-v-r approaches to support vector machine multi-classification, *Information Sciences: an International Journal*, Vol. 177, No. 18, pp. 3782-3798.
- Liu R, Wang Y, Baba T, Masumoto D y Nagata S, 2008, SVM-based active feedback in image retrieval using clustering and unlabeled data, *Pattern Recognition*, Vol. 41, No. 8, pp. 2645-2655.
- Lv H, Wang W, Wang C y Zhuo Q, 2005, Off-line Chinese signature verification based on support vector machines, *Pattern Recognition Letters*, Vol. 26, No. 15, pp. 2390-2399.
- Pavlidis I, Mavuduru R y Papanikolopoulos N, 1994, Off-line recognition of signatures using revolving active deformable models, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 771-776.
- Quek C y Zhou R, 2002, Antiforgery: a novel pseudo-outer product based fuzzy neural network driven signature verification system, *Pattern Recognition Letters*, Vol. 23, No. 14, pp. 1795-1816.
- Ramesh V y Narasimha M, 1999, Off-line signature verification using genetically optimized weighted features, *Pattern Recognition*, Vol. 32, No. 2, pp. 217-233.
- Rincón L, 2008, Reconocimiento de firmas mediante modelos ocultos de markov, Proyecto de grado, Facultad de Ingeniería, Universidad de Los Andes, Mérida, Venezuela.
- Rohlík O, 2003, *Handwritten text analysis*, Thesis, Faculty of Applied Science, University of west Bohemia, Laboratory of Intelligent Communication System, Pilsen.
- Silva E, Capobianco R, Scalassara P, Maciel C y Pereira J, 2007, Wavelet time-frequency analysis and least squares support vector machines for the identification of voice disorders, *Computers in Biology and Medicine*, Vol. 37, No. 4, pp. 571-578.
- Taylan M y Canan L, 2009, Signature verification (SV) toolbox: Application of PSO-NN, *Engineering Applications of Artificial Intelligence*, Vol. 22, No. 4-5, pp. 688-694.
- Toscano K, Nakano M, Sánchez G, Pérez H y Yasuhara H, 2005, Reconocimiento de caracteres manuscritos usando la función spline natural. *Científica*, Vol. 9, No. 3, pp. 143-154.
- Vapnik V, 1995, *The nature of statistical learning theory*, John Wiley & Sons, New York.
- Xiao X, y Leedham G, 2002, Signature verification using a modified Bayesian network, *Pattern Recognition*, Vol. 35, No. 5, pp. 983-995.
- Xuhua Y, Furuhashi T, Obata K y Uchikawa Y, 1996, Selection of features for signature verification using the genetic algorithm, *Computers & Industrial Engineering*, Vol. 30, No. 4, pp. 1037-1045.