

The logo for TecnoLógicas features the word "TecnoLógicas" in a stylized, serif font. The "L" is particularly large and prominent. The logo is enclosed in a thin, rounded rectangular border.

Tecno Lógicas

ISSN: 0123-7799

tecnologicas@itm.edu.co

Instituto Tecnológico Metropolitano
Colombia

López-Sarmiento, Danilo A.; Manta-Caro, Héctor C.; Vera-Parra, Nelson E.
Clasificador Basado en una Máquina de Vectores de Soporte de Mínimos Cuadrados Frente a un
Clasificador por Regresión Logística ante el Reconocimiento de Dígitos Numéricos
Tecno Lógicas, núm. 31, diciembre, 2013, pp. 37-51
Instituto Tecnológico Metropolitano
Medellín, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=344234334003>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Artículo de Investigación/Research Article

Clasificador Basado en una Máquina de Vectores de Soporte de Mínimos Cuadrados Frente a un Clasificador por Regresión Logística ante el Reconocimiento de Dígitos Numéricos

Least Square Support Vector Machine Classifier vs a Logistic Regression Classifier on the Recognition of Numeric Digits

Danilo A. López-Sarmiento¹

Héctor C. Manta-Caro²

Nelson E. Vera-Parra³

Recibido: 30 de agosto de 2012

Aceptado: 31 de julio de 2013

1 Ph.D.(c) en Ingeniería,
Grupo Investigación Internet Inteligente
Universidad Distrital Francisco José de Caldas,
Bogotá-Colombia
dalopezs@udistrital.edu.co

2 Ph.D.(c) Ciencias de la Computación,
Universidad de Granada,
Granada-España
cristyan.manta@gmail.com

3 Ph.D.(c) en Ingeniería,
Grupo Investigación GICOGÉ,
Universidad Distrital Francisco José de Caldas,
Bogotá-Colombia
neverap@udistrital.edu.co

Resumen

En este artículo se compara el desempeño de una máquina de vectores de soporte de mínimos cuadrados multi-clase (multi-class Least Square Support Vector Machine mc-LSSVM) frente a un clasificador por regresión logística multi-clase, ante el problema del reconocimiento de dígitos numéricos (0-9) escritos a mano. Para desarrollar la comparación se usó un set de datos compuesto por 5000 imágenes de dígitos numéricos escritos a mano (500 imágenes por cada número del 0-9), cada imagen de 20 x 20 píxeles. La entrada a cada uno de los sistemas evaluados fueron vectores de dimensión 400, correspondientes a cada imagen (no se realizó extracción de características). Ambos clasificadores utilizan la estrategia Uno contra todos (OneVsAll) para habilitar la multi-clasificación y una función de validación cruzada aleatoria para el proceso de minimización de la función de costo. Las métricas de comparación fueron la precisión y el tiempo de entrenamiento bajo las mismas condiciones computacionales. Ambas técnicas evaluadas presentaron una precisión superior al 95 %, siendo LS-SVM ligeramente más precisa. Sin embargo, en el costo computacional sí se encontró una diferencia notoria: LS-SVM requiere un tiempo de entrenamiento 16,42 % inferior al requerido por el modelo basado en regresión logística bajo las mismas condiciones computacionales.

Palabras clave

Máquina de vectores de soporte, mínimos cuadrados, regresión logística, clasificador, dígitos numéricos.

Abstract

In this paper is compared the performance of a multi-class least squares support vector machine (LSSVM mc) versus a multi-class logistic regression classifier to problem of recognizing the numeric digits (0-9) handwritten. To develop the comparison was used a data set consisting of 5000 images of handwritten numeric digits (500 images for each number from 0-9), each image of 20 x 20 pixels. The inputs to each of the systems were vectors of 400 dimensions corresponding to each image (not done feature extraction). Both classifiers used OneVsAll strategy to enable multi-classification and a random cross-validation function for the process of minimizing the cost function. The metrics of comparison were precision and training time under the same computational conditions. Both techniques evaluated showed a precision above 95 %, with LS-SVM slightly more accurate. However the computational cost if we found a marked difference: LS-SVM training requires time 16.42 % less than that required by the logistic regression model based on the same low computational conditions.

Keywords

Support vector machine, least square, logistic regression, classifier, numeric digits.

1. INTRODUCCIÓN

Support Vector Machine (SVM) es un método de clasificación y regresión proveniente de la teoría de aprendizaje estadístico (Vapnik et al., 1971; Vapnik, 1998). La metodología base de SVM se puede resumir de la siguiente forma: Si se requiere clasificar un conjunto de datos (representados en un plano n -dimensional) no separables linealmente, se toma dicho conjunto de datos y se mapea a un espacio de mayor dimensión donde sí sea posible la separación lineal (esto se realiza mediante funciones llamadas Kernel). En este nuevo plano se busca un hiperplano que sea capaz de separar en 2 clases los datos de entrada; el plano debe tener la mayor distancia posible a los puntos de ambas clases (los puntos más cercanos a este hiperplano de separación son los vectores de soporte). Si lo que se requiere es hacer una regresión, se toma el conjunto de datos y se transforma a un espacio de mayor dimensión (donde sí se pueda hacer una regresión lineal) y en este nuevo espacio se realiza la regresión lineal pero sin penalizar errores pequeños.

La forma como se soluciona el problema de encontrar un hiperplano maximizando los márgenes entre este y los puntos de ambas clases (del conjunto de datos de entrada), define si la SVM es tradicional o es LS-SVM. SVM soluciona dicho problema mediante el principio de structural risk minimization, mientras que LS-SVM lo soluciona mediante un conjunto de ecuaciones lineales (Suykens et al., 1999). Mientras que en SVM tradicional muchos valores de soporte son cero (valores diferentes a cero corresponden a los vectores de soporte), en LS-SVM los valores de soporte son proporcionales a los errores.

LS-SVM ha sido propuesto y usado en el procesamiento de imágenes en diferentes campos. A continuación se nombran algunos ejemplos: análisis de imágenes médicas diagnósticas, tales como, MRI (Magnetic Resonance Imaging) y MRS (Magnetic Resonance Spectroscopy) (Luts, 2010; Laudadio et al., 2009); interpretación de imágenes provenientes de datos genéticos, tales como los microarray (Ojeda et al., 2011); tratamiento de imágenes en el ámbito de la seguridad, como la detección de objetos en imágenes infrarrojas (Danyan et al., 2010), la detección de rostros (Guang-

ying et al., 2008) y el reconocimiento de matrículas de vehículos (Xinming et al., 2008); tratamiento de imágenes espaciales, como RSI (Remote Sensing Image) (Sheng et al., 2008). En este trabajo se propone y se evalúa el uso de LS-SVM en el reconocimiento de patrones de escritura, específicamente en el reconocimiento automático de dígitos numéricos escritos a mano y se compara su desempeño frente a un sistema que se basa en regresión logística y optimización mediante el algoritmo del gradiente conjugado.

Este artículo se organiza de la siguiente forma: inicialmente se explica la metodología: conjunto de datos, modelos a evaluar y métricas, luego se exponen y analizan los resultados, y por último, se obtienen las conclusiones.

2. FUNDAMENTOS CONCEPTUALES

LS-SVM (Suykens et al., 1999; Suykens et al., 2002)

Dado un conjunto de entrenamiento (1):

$$\{y_k, x_k\}_{k=1}^N \text{ Input } x_k \in R^n, \text{ Output } y_k \in R^n \quad (1)$$

SVM tiene como objetivo construir un clasificador (2):

$$y(x) = \text{sign} \left[\sum_{k=1}^N \alpha_k y_k \Psi(x, x_k) + b \right] \quad (2)$$

Donde σ y b son constantes y Ψ representa el kernel que para este clasificador se ha utilizado una función de base radial (RBF) (3):

$$\Psi(x, x_k) = e^{\left\{ \frac{-\|x - x_k\|_2^2}{\sigma^2} \right\}} \text{ RBF Kernel, } \sigma \text{ es constante} \quad (3)$$

El clasificador se construye de la siguiente forma (4):

$$\omega^T \varphi(x_k) + b \geq 1, \text{ if } y_k = +1 \quad (4)$$

$$\omega^T \varphi(x_k) + b \leq 1, \text{ if } y_k = -1$$

Que se puede expresar de forma compacta (5):

$$y_k[\omega^T \varphi(x_k) + b] \geq 1, k = 1, \dots, N. \quad (5)$$

Donde φ es la función que mapea los datos de entrada a un nuevo espacio de mayor dimensión donde exista el hiperplano separador. Si no se consigue obtener el hiperplano en el espacio de mayor dimensión se introduce un tipo de bias (6):

$$y_k[\omega^T \varphi(x_k) + b] \geq 1 - \varepsilon_k, \varepsilon_k \geq 0, k = 1, \dots, N \quad (6)$$

SVM tradicional soluciona el problema de encontrar el mejor hiperplano clasificador utilizando el principio *structural risk minimization* (7):

$$\min_{\omega, \varepsilon_k} f(\omega, \varepsilon_k) = \frac{1}{2} \omega^T \omega + c \sum_{k=1}^N \varepsilon_k \quad (7)$$

LS-SVM plantea una modificación a (7) para ser resuelta mediante un conjunto de ecuaciones lineales (Cristianini et al., 2003). La ecuación propuesta por LS-SVM es (8):

$$\min_{\omega, b, e} f(\omega, b, e) = \frac{1}{2} \omega^T \omega + \gamma \frac{1}{2} \sum_{k=1}^N e_k^2 \quad (8)$$

De acuerdo con (6) se construye el lagrangiano (9):

$$\mathcal{L}(\omega, b, e; \alpha) = f(\omega, b, e) - \sum_{k=1}^N \alpha_k \{y_k[\omega^T \varphi(x_k) + b] - 1 + e\} \quad (9)$$

Donde α_k son los multiplicadores de Lagrange. Para optimizar (10):

$$\begin{aligned}
 \frac{\partial \mathcal{L}}{\partial \omega} = 0 &\rightarrow \omega = \sum_{k=1}^N \alpha_k y_k \varphi(x_k) \\
 \frac{\partial \mathcal{L}}{\partial b} = 0 &\rightarrow \omega = \sum_{k=1}^N \alpha_k y_k = 0 \\
 \frac{\partial \mathcal{L}}{\partial e_k} = 0 &\rightarrow \alpha_k = \gamma e_k, k = 1, \dots, N \\
 \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 &\rightarrow \alpha_k = y_k [\omega^T \varphi(x_k) + b] - 1 + e_k = 0, k = 1, \dots, N
 \end{aligned} \tag{10}$$

Las condiciones para optimizar se escriben como la solución a un conjunto de ecuaciones lineales (11):

$$\begin{bmatrix} I & 0 & 0 & -Z^Y \\ 0 & 0 & 0 & -Y^T \\ 0 & 0 & \gamma^I & -I \\ Z & Y & I & 0 \end{bmatrix} \begin{bmatrix} \omega \\ b \\ e \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vec{1} \end{bmatrix} \tag{11}$$

$$Z = [\varphi(x_1)^T y_1; \dots; \varphi(x_N)^T y_N]$$

$$Y = [y_1; \dots; y_N]$$

$$\vec{1} = [1; \dots; 1]$$

$$e = [e_1; \dots; e_N]$$

$$\alpha = [\alpha_1; \dots; \alpha_N]$$

La solución se expresa también como (12):

$$\begin{bmatrix} 0 & -Y^T \\ Y & ZZ^T + \gamma^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \vec{1} \end{bmatrix} \tag{12}$$

y aplicando la condición de Mercer, se obtiene (13):

$$\Omega = ZZ^T, \Omega_{kl} = y_k y_l \varphi(x_k)^T \varphi(x_l) = y_k y_l \Psi(x_k, x_l) \tag{13}$$

Por último, el clasificador (2) es encontrado solucionando el conjunto de ecuaciones lineales (12)-(13). Los valores de soporte α_x son proporcionales a los errores (10).

Regresión Logística (Russell et al., 2005; Stephan et al., 2002)

En la versión regularizada de la regresión logística, la función de costo se define como (14):

$$J(\theta) = \frac{1}{2m} \left[\sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n \theta_j^2 \right] \quad (14)$$

En donde la función hipótesis está dada por (15):

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \quad (15)$$

Y en donde θ son los parámetros del modelo, λ es el parámetro de regularización, m el tamaño del conjunto de entrenamiento, n el número de características del modelo y g representa la función sigmoidea. A fin de minimizar la función costo y optimizar los parámetros theta del modelo por regresión logística se decide el uso del método del gradiente conjugado, por lo que se requiere en la implementación calcular las derivadas parciales de la función costo con respecto a los parámetros theta, de la forma (16):

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \right) + \frac{\lambda}{m} \theta_j \text{ para } j \geq 1 \quad (16)$$

Empleando el método del gradiente conjugado se encuentra la matriz de parámetros que optimizan la función costo y a fin de realizar predicciones se construye una matriz de parámetros de forma $\Theta \in \mathbb{R}^{K \times (N+1)}$ donde K es el número de clases.

Construyendo la función de predicción p como (17) y (18):

$$p = g(\Theta^T X) \quad (17)$$

$$c_p = \operatorname{argmax}_{c \in \mathcal{C}} (g(\Theta^T X)) \quad (18)$$

3. METODOLOGÍA

3.1 Conjunto de Datos

El conjunto de datos de entrenamiento se encuentra compuesto por 5000 imágenes de dígitos numéricos escritos a mano (de los cuales se utilizaron 4000 para entrenar y 1000 para validar). En la Fig. 1, se muestra un subconjunto de 20 imágenes del conjunto de entrenamiento.

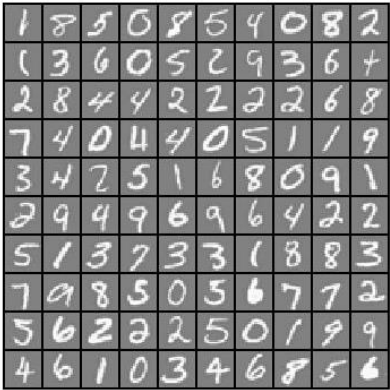


Fig. 1. Subconjunto de datos de entrenamiento: imágenes de escritura numérica de 40x40 píxeles. Fuente: <http://yann.lecun.com/exdb/mnist/>

Cada imagen se constituye en un vector de características para los modelos de predicción contruidos, así cada dato de entrenamiento es una imagen de 20x20 píxeles en escalas de grises. Cada pixel es representado por un número en punto flotante indicando en intensidad de escala de grises de cada posición. Cada imagen es entonces dispuesta como un vector de dimensión 400. De tal manera se construye una matriz de características de entrada de la forma:

$$X = \begin{bmatrix} -(x^{(1)})^T & - \\ -(x^{(2)})^T & - \\ \vdots & \\ -(x^{(m)})^T & - \end{bmatrix} \quad (19)$$

Adicionalmente, se parte de un vector y de dimensión 5000 que contiene las etiquetas del conjunto de datos de entrenamiento. Donde “10” es la etiqueta para el dígito cero (0) y los dígitos (0-9) se etiquetan “1” al “9” respectivamente. En la Fig. 2 se muestra un ejemplo de un dato de entrenamiento.



Fig. 2. Ejemplo de dato de entrenamiento: imagen de escritura numérica de 40x40 píxeles. Fuente: <http://yann.lecun.com/exdb/mnist/>

3.2 Modelos a Evaluar

Clasificador LS-SVM. El clasificador diseñado y utilizado en este trabajo es multi-clase y utiliza la codificación uno-vs.-todos, en la cual se crean n_c clasificadores binarios que discriminan una específica clase contra una clase formada por el resto de clases ($c-1$ clases). Además utiliza una función de validación cruzada aleatoria (validación donde se divide aleatoriamente los datos que van a servir de entrenamiento y los que van a servir de prueba) para el proceso de minimización de la función de costo.

Clasificador Regresión Logística. Se usará un modelo vectorizado por regresión logística empleando la metodología uno-vs.-todos. Por tal forma, ya que el experimento supone clasificar 10 diferentes clases para reconocer cada uno de los 10 dígitos numéricos, se entrenaron 10 clasificadores por regresión logística de manera independiente.

3.3 Métricas

Precisión. Indica la capacidad del clasificador de dar el mismo resultado en mediciones diferentes realizadas en las mismas condiciones (20).

$$Precisión = \frac{VP + VN}{(VP + VN + FP + FN)} * 100 \quad (20)$$

Dada la tarea de clasificar las imágenes en dos conjuntos: las que corresponden a un número específico y las que no corresponden a dicho número, los elementos de (20) se definen así:

FP Falso Positivo: imagen clasificada erróneamente en el conjunto de las que corresponden al número específico.

FN Falso Negativo: imagen clasificada erróneamente en el conjunto de las que no corresponden al número específico.

VP Verdadero Positivo: imagen clasificada correctamente en el conjunto de las que corresponden al número específico.

VN Verdadero Negativo: imagen clasificada correctamente en el conjunto de las que no corresponden al número específico.

Tiempo de entrenamiento. La duración del proceso de entrenamiento bajo las mismas condiciones computacionales es utilizada como una medida indirecta del costo computacional que acarrea cada uno de los clasificadores.

3.4 Herramientas de Programación

Los script que implementan los clasificadores a evaluar se desarrollaron en MATLAB, haciendo uso de los Toolbox:

LS-SVM (De Brabanter et al., 2010) - Toolbox no oficial desarrollado por el Departamento de Electrónica de la Universidad ESAT, Bélgica.

STATISTICS – Toolbox Oficial.

4. RESULTADOS Y ANÁLISIS

En la Tabla 1 se exponen los resultados de precisión y tiempo de entrenamiento obtenidos por los dos modelos para cada uno de los dígitos.

Tabla 1 Resultados. Fuente: Autores

Dígito	Precisión (%)		Tiempo de entrenamiento promedio (s)	
	LS-SVM	Regresión Logística	LS-SVM	Regresión Logística
0	99,00	99,00		
1	94,50	95,10		
2	95,33	96,20		
3	95,50	94,20		
4	94,40	93,80		
5	94,50	93,80	280,0	335,0
6	94,86	95,30		
7	94,75	94,80		
8	94,33	92,91		
9	94,90	95,00		
Promedio	95,21	95,01		

4.1 Análisis de la Precisión

Ambos modelos presentan su mayor precisión para el número 0 (99 % para los dos modelos). De igual forma ambos modelos coinciden con su peor precisión para el número 8 (94,33 % para el modelo LS-SVM y 92,91 % para el modelo de regresión logística). Para los números 1, 2, 6, 7 y 9 el modelo basado en regresión logística presenta mejor precisión (aunque por muy poca diferencia), caso contrario para los números 3, 4, 5, 8 donde el modelo basado en LS-SVM es más preciso. En general el clasificador basado en

LS-SVM es más preciso con un promedio de 95,21 % frente a 95,01 % presentado por el clasificador basado en regresión logística. En la Fig. 3 se puede observar la precisión de cada uno de los clasificadores.

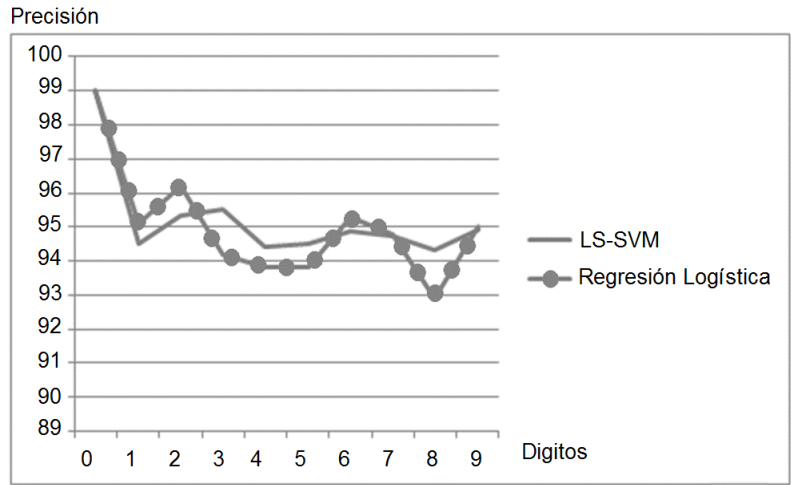


Fig. 3. Comparación de la precisión de los dos modelos para cada uno de los dígitos.
Fuente: Autores

Observando las confusiones (falsos positivos y falsos verdaderos), se puede notar que el criterio de selección de los clasificadores se basa en los patrones geométricos generales de los números tales como: circular doble (8, 3, 5, 6, 9), rectas horizontales, verticales y diagonales (1, 7, 4, 2) y circular sencillo (0). Desde esta perspectiva, la baja precisión para el número 8 se debe a la confusión de los clasificadores con los números 3, 5, 6 y 9 ya que comparten un patrón similar circular doble.

4.2 Análisis del Costo Computacional

Bajo las mismas condiciones computacionales, el clasificador basado en LS-SVM requiere un tiempo de entrenamiento de 280 segundos, 16,42 % menor a lo que requiere el clasificador basado en regresión logística, el cual empleó 335 segundos en su proceso de entrenamiento. También se comparó el proceso de minimización

de la función de costo con respecto al número de iteraciones, observándose que el clasificador LS-SVM solo en 19 iteraciones ya alcanza el valor alcanzado por el clasificador de regresión logística en 50 iteraciones. En la Fig. 4 se pudo observar dicha comparación.

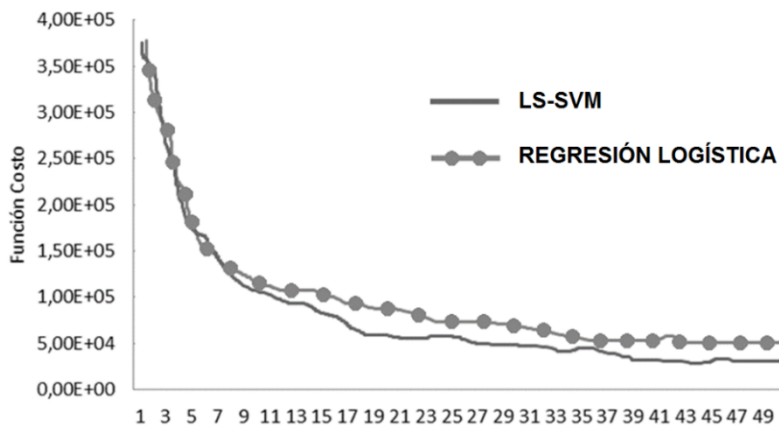


Fig. 4. Minimización iterativa de la función de costo vs. número de iteraciones para cada uno de los clasificadores. Fuente: Autores

5. CONCLUSIONES

Al evaluar dos técnicas de machine learning: LS-SVM y Regresión Logística como clasificadores de dígitos numéricos manuscritos, se llegó a las siguientes conclusiones:

Ambas técnicas evaluadas presentaron una precisión superior al 95 %, siendo LS-SVM ligeramente más precisa. Ambas técnicas presentaron mayor precisión para el número 0 y menor precisión para el número 8. LS-SVM presenta mayor precisión para los números 3, 4, 5, 8, caso contrario para los números 1, 2, 6, 7, 9 donde Regresión Logística presenta mayor precisión.

El criterio de selección de los clasificadores se basa en los patrones geométricos generales de los números tales como: circular doble (8, 3, 5, 6, 9), rectas horizontales, verticales y diagonales (1, 7, 4, 2) y circular sencillo (0).

En el costo computacional se encontró una diferencia notoria: LS-SVM requiere un tiempo de entrenamiento 16,42 % inferior al requerido por el modelo basado en regresión logística bajo las mismas condiciones computacionales y esto esencialmente es debido a que el clasificador basado en Regresión Logística requiere 50 iteraciones para alcanzar el nivel de aprendizaje que alcanza el clasificador LS-SVM únicamente con 19 iteraciones.

6. REFERENCIAS

- Cristianini, S. T. (2003). An introduction to support vector machines and other Kernel-Based Learning Methods. *Cambridge University Press*, 254-272.
- Danyan Y., Yiquan W., (2010). *Detection of small target in infrared image based on KFCM and LS-SVM*. Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2nd International Conference. Vol. 1, 309-312
- De Brabanter K., Karsmakers P., Ojeda F., Alzate C., De Brabanter J., Pelckmans K., De Moor B., Vandewalle J., Suykens J.A.K. (2010). LS-SVMLab Toolbox User's guide version 1.8. Internal report, ESAT-SISTA, K.U. Leuven (Leuven, Belgium), 10-146.
- Guangying, G., Xinzong, B., Jing G. (2008). Study on automatic detection and recognition algorithms for vehicles and license plates using LS-SVM. Intelligent Control and Automation. WCICA, 3760-3765.
- Laudadio, T., Luts, J., Martinez, M.C., Van, C., S., Molla, E., Piquer, J., Suykens J., Himmelreich, U., Celda, B., Van, H. (2009). Differentiation between brain metastases and glioblastoma multiforme based on MRI, MRS and MRSI, Proc. of the 22nd IEEE International Symposium on Computer-Based Medical Systems (CBMS), Albuquerque, New Mexico, 1-8.
- Luts J. (2010). *Classification of brain tumors based on magnetic resonance spectroscopy*. PhD thesis. Faculty of Engineering, K.U. Leuven, Leuven, Belgium.
- Ojeda F. (2011). *Kernel based methods for Microarray and Mass Spectrometry Data Analysis*. PhD thesis, Faculty of Engineering, K.U. Leuven, Leuven, Belgium.
- Russell, G., Xiaoyuan, S., Bin, S., Wei, Z. (2005). Structural extension to Logistic Regression: Discriminative parameter learning of belief net classifiers. *Machine Learning*, 59, 297-322.

- Sheng, Z., Wen, S., Jian, L., Jinwen, T. (2008). Remote Sensing Image Fusion Using Multiscale Mapped LS-SVM. *IEEE Transactions On Geoscience And Remote Sensing*, 46, 126-133.
- Stephan. D., Lucila, M. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of Bio-medical Informatic*, 35, 352-359.
- Suykens, J., Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, 9, 293-300.
- Suykens, J., Van Gestel, J., De Brabanter, B., De Moor, J. Vandewalle, (2002). *Least Squares Support Vector Machines*. Singapore, *World Scientific*, 35-123.
- Vapnik, A., Chervonenkis (1971). *On the uniform convergence of relative frequencies of events to their probabilities*. Theory of Probability and its Applications, 264-280.
- Xinming, Z., Jian, Z. (2008). Face recognition based on sub-image feature extraction and LS-SVM. *Computer Science and Software Engineering, International Conference*, 1, 772-77.