



Letters

Fuzzy sigmoid kernel for support vector classifiers

G. Camps-Valls^{a,*}, J.D. Martín-Guerrero^a,
J.L. Rojo-Álvarez^b, E. Soria-Olivas^a

^a*Dept. Enginyeria Electrònica, Universitat de València, C/ Dr. Moliner, 50, 46100 Burjassot, València, Spain*

^b*Dept. Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Avda. Universidad, 30, 28911 Leganés, Madrid, Spain*

Communicated by R. Newcomb

Abstract

This Letter proposes the use of the fuzzy sigmoid function presented in (IEEE Trans. Neural Networks 14(6) (2003) 1576) as non-positive semi-definite kernel in the support vector machines framework. The fuzzy sigmoid kernel allows lower computational cost, and higher rate of positive eigenvalues of the kernel matrix, which alleviates current limitations of the sigmoid kernel.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Support vector machines; Positive semi-definite kernel; Sigmoid

1. Introduction

In the support vector machines (SVM) framework for pattern recognition, non-linear classifiers require the use of a symmetric and positive semi-definite (PSD) kernel, which transforms the input space samples into a high-dimensional (possibly infinite) feature space, in which a linear classification is performed [9]. Many non-linear kernels are available, such as linear, polynomial, Gaussian, or sigmoid-shaped

*Corresponding author. Tel.: +34-96-3160-197; fax: +34-96-3160-466.

E-mail address: gustavo.camps@uv.es (G. Camps-Valls).

functions. The sigmoid kernel has been proposed theoretically for SVM due to its origin from neural networks but, to date, it has not been extensively used in practice because it becomes a PSD kernel for only some combinations of its free parameters (slope and bias of the function) [2,7]. Given that sigmoidal activation functions have been widely proved to provide useful global classifiers in neural networks, its inclusion in the SVM framework is an interesting and open issue.

This Letter shows the advantages of using the fuzzy-based sigmoid function presented in [8] as a non-PSD kernel matrix for SVM. We benchmark our proposal to the standard sigmoid kernel in terms of (a) performance in benchmark data sets, (b) computational cost, and (c) positive definiteness properties.

2. SVM and non-PSD sigmoid kernels

Given a labelled training data set $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^N$ and $y_i \in \{-1, +1\}$, and a non-linear mapping $\phi(\cdot)$, usually to a higher dimensional space, $\mathbb{R}^N \xrightarrow{\phi(\cdot)} \mathbb{R}^H (H > N)$, the SVM method solves

$$\min_{\mathbf{w}, \xi_i, b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i \right\} \quad (1)$$

constrained to

$$y_i(\phi^T(\mathbf{x}_i)\mathbf{w} + b) \geq 1 - \xi_i \quad \forall i = 1, \dots, n, \quad (2)$$

$$\xi_i \geq 0 \quad \forall i = 1, \dots, n, \quad (3)$$

where \mathbf{w} and b define a linear classifier in the feature space. The non-linear mapping function ϕ is performed in accordance with Cover's theorem [4], which guarantees that the transformed samples are more likely to be linearly separable in the resulting feature space. The regularisation parameter C controls the generalisation capabilities of the classifier and it can be selected by the user, and ξ_i are positive slack variables enabling to deal with permitted errors.

Due to the high dimensionality of vector variable \mathbf{w} , primal function (1) is usually solved through its Lagrangian dual problem, which consists of maximising

$$L_d \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \quad (4)$$

constrained to $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i y_i = 0, \forall i = 1, \dots, n$, where auxiliary variables α_i are Lagrange multipliers corresponding to Eq. (2). It is worth noting that all ϕ mappings used in the SVM learning occur in the form of inner products. This allows us to define a kernel function K :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j), \quad (5)$$

and then a non-linear SVM can be constructed without considering the mapping ϕ explicitly, but only the kernel function. Some popular kernels are linear ($K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$), polynomial ($K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$) or Gaussian

($K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$). It can be noted that, though ϕ mapping can be explicitly expressed for the linear or polynomial kernel, there is no explicit form of the ϕ mapping corresponding to the Gaussian kernel. Moreover, it can be demonstrated that the expansion is an infinite-dimensional functional [7]. Mercer's condition [3] avoids to explicitly calculate ϕ in these cases, and then, by introducing (5) into (4), the dual problem can be finally stated as:

$$L_d \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j). \quad (6)$$

After the dual problem (4) is solved, $\mathbf{w} = \sum_{i=1}^n y_i \alpha_i \phi(\mathbf{x}_i)$, and the decision function implemented by the classifier for any test vector \mathbf{x} is given by

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^n y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \right), \quad (7)$$

where b can be easily computed from the α_i that are neither 0 nor C , as explained in [7].

The mapping function is a Mercer's kernel if $K(\cdot, \cdot)$ is symmetric and positive semi-definite (PSD) [3]. Nevertheless, some non-PSD matrices are used in practice. The sigmoid kernel ($K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(a \cdot \mathbf{x}_i \cdot \mathbf{x}_j + r)$) is quite popular for SVM due to its origin from neural networks. However, it has not been extensively used in practice because it becomes a PSD kernel for only some ranges of its free parameters (slope a and bias r) [2,7,9]. When K is not PSD, Eq. (5) cannot be satisfied for real ϕ and the primal–dual relationship does not exist. For $a > 0$ and $r < 0$ and small enough, it can be demonstrated that the sigmoid kernel matrix is conditionally positive definite (CPD), i.e. for all $\mathbf{x} \neq 0 \in \mathbb{R}^N$ then $\mathbf{x}^T K \mathbf{x} > 0$ [5]. We take advantage of this property to analyse the performance of the fuzzy-based sigmoid function proposed in [8] as a CPD kernel matrix. The good results obtained in the context of neural networks encourage its use in the SVM framework.

3. Fuzzy sigmoid kernel

The fuzzy-based sigmoid function models the hyperbolic tangent function by means of linguistic variables [8]. We can extend its description to the kernel framework as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} -1, & \mathbf{x}_i \cdot \mathbf{x}_j \text{ is low,} \\ +1, & \mathbf{x}_i \cdot \mathbf{x}_j \text{ is high,} \\ m \cdot \mathbf{x}_i \cdot \mathbf{x}_j, & \mathbf{x}_i \cdot \mathbf{x}_j \text{ is medium,} \end{cases} \quad (8)$$

where m is a constant factor representing the smoothness of the sigmoid tract. In the context of fuzzy logic, the sigmoid kernel can be defined by a series of membership functions. In this paper, we only consider three triangular and piece-wise linear functions due to their simplicity, as depicted in Fig. 1. Since the activation function must be continuous, the membership limits are given by $\gamma \pm 1/a$ where $\gamma = -r/a$.

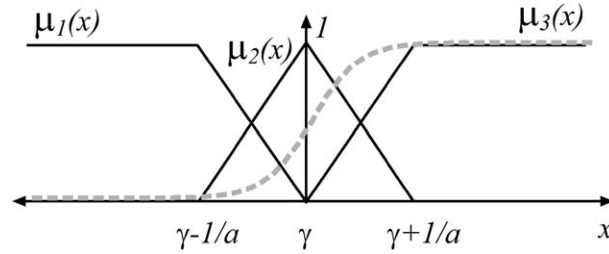


Fig. 1. Schematic of the three membership functions (two linear and a triangular one, solid black lines) used to model $\tanh(x)$ (dashed grey line). The values of γ and a of the resulting fuzzy tanh function can be easily related to the slope and bias of the tanh function [8].

Hence, expression (8) can be readily re-written as a function of a and r , as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} -1, & \mathbf{x}_i \cdot \mathbf{x}_j \leq \gamma - (1/a), \\ +1, & \mathbf{x}_i \cdot \mathbf{x}_j \geq \gamma + (1/a), \\ 2(\mathbf{x}_i \cdot \mathbf{x}_j - \gamma) - a^2(\mathbf{x}_i \cdot \mathbf{x}_j - \gamma) \cdot |(\mathbf{x}_i \cdot \mathbf{x}_j - \gamma)| & \text{otherwise} \end{cases} \quad (9)$$

which is the final form of the proposed fuzzy sigmoid (*fuzzy tanh*) kernel.

The main advantages of this function are that (a) it is differentiable at every point in its entire domain, (b) it induces faster trainings since the final solution will be expressed in a series of saturated samples (Eq. (8)), and (c) it permits to select different levels of non-linearity by choosing the amount and sophistication of the membership functions.

4. Results

We tested our proposal in ten databases from the UCI *Machine Learning* repository [1]. The selected databases have different number of samples, input dimension, and classes, which allows a proper comparison. In order to develop an SVM, penalization parameter C , and kernel parameters a and r must be tuned. We tried 30 exponentially spaced points for C ($\log_{10} C = [-2, \dots, 3]$) and 30 more equally spaced points for the variation of the bias term ($r = [-3, \dots, 0]$). Parameter a was fixed to $1/N$, where N is the input dimension. Therefore, 900 trainings were performed for each problem and the best free parameters were selected through the 8-fold cross-validation method. A multiclassification scheme was made in multiclass problems [10]. Models were trained using the iterated reweighted least squares (IRWLS) procedure [6], which has already been demonstrated to be more efficient than standard quadratic programming implementations in both time and memory requirements. The experiments ran on a Pentium IV (3.06 GHz) with 512 MB RAM. In order to analyse the performance of the SVM with both kernels, we took three

measures of quality: the cross-validation success rate (SR(%)), the average rate of positive eigenvalues ($\lambda > 0$), and the average CPU time (s).

Table 1 shows the best results obtained for each problem. It is worth noting that, in general, the computational effort and the rate of positive eigenvalues are better using our proposal (14.01% and 8.17% average improvement, respectively) at the expense of slightly lower results (2.6% average decrease) due to the simplicity of the fuzzy model considered. Differences are higher in data sets with a low number of features or classes, which encourages the use of the fuzzy sigmoid kernel to avoid numerical problems in those situations.

Table 1
Results on the selected UCI databases

| Name | #P. | #F. | #C. | Tanh | | | Fuzzy tanh | | |
|-----------------|-----|-----|-----|--------|---------------|---------|------------|---------------|---------|
| | | | | SR (%) | $\lambda > 0$ | CPU (s) | SR(%) | $\lambda > 0$ | CPU (s) |
| Wdbc | 569 | 30 | 2 | 97 | 73 | 28.54 | 98 | 78 | 27.53 |
| Glass | 214 | 10 | 7 | 72 | 44 | 10.00 | 69 | 48 | 9.83 |
| Ionosphere | 351 | 34 | 2 | 95 | 21 | 31.30 | 93 | 24 | 29.41 |
| Iris | 150 | 4 | 3 | 97 | 55 | 6.84 | 94 | 61 | 4.90 |
| Liver-disorders | 345 | 6 | 2 | 60 | 44 | 10.11 | 60 | 48 | 10.47 |
| Pima-diabetes | 768 | 8 | 2 | 71 | 88 | 22.60 | 73 | 81 | 18.66 |
| Sonar | 208 | 60 | 2 | 88 | 56 | 46.63 | 88 | 57 | 33.57 |
| Vehicle | 846 | 18 | 4 | 71 | 29 | 42.74 | 65 | 31 | 29.18 |
| Vowel | 990 | 10 | 11 | 77 | 45 | 270.78 | 71 | 51 | 270.21 |
| Wine | 178 | 13 | 3 | 95 | 33 | 3.94 | 92 | 44 | 2.89 |

From left to right columns: database name, number of patterns (#P), number of features (#F), number of classes (#C), success rate (SR (%)), rate of positive eigenvalues ($\lambda > 0$) and CPU time (s) for the sigmoid (tanh) and the fuzzy-based sigmoid (fuzzy tanh) kernels.

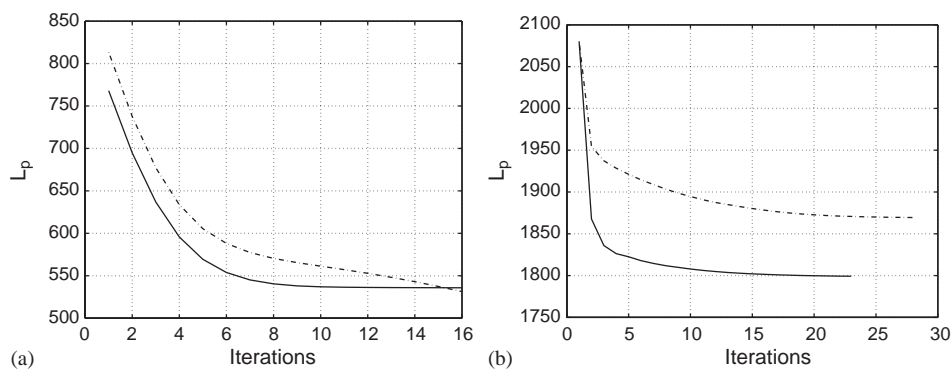


Fig. 2. Convergence of the averaged primal functional (Eq. (1)) for the tanh (dash-dotted) and fuzzy tanh (solid) kernel classifiers for the (a) wdbc and (b) vowel data sets.

It is also interesting to focus on the convergence speed of the classifiers. Fig. 2 shows the average convergence of the IRWLS algorithm for the tanh and the *fuzzy* tanh kernel classifiers for *wdbc* and *vowel* data sets. It can be observed that a lower number of iterations becomes necessary when using the fuzzy tanh, something that is specially significant in the case of high input dimension problems, Fig. 2(b).

5. Conclusions

This Letter has presented the use of a fuzzy-based sigmoid function in the form of a non-PSD kernel matrix for pattern recognition. The fuzzy sigmoid function allows lower computational cost and higher rate of positive eigenvalues of the kernel matrix than those from the standard sigmoid kernel. Moreover, the fuzzy kernel can be easily implemented in hardware. Results suggest that further and elaborated versions of the *fuzzy* tanh kernel could provide a SVM framework with a sigmoidal global kernel to overcome the current limitations of the sigmoid kernel.

References

- [1] C. Blake, C. Merz, UCI repository of machine learning databases, (<http://www.ics.uci.edu/~mllearn/MLRepository.html>). Last modification: June, 1st 2004. Last access: June, 1st 2004 (1998).
- [2] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, *Knowledge Discovery and Data Mining* 2 (2) (1998) 121–167.
- [3] R. Courant, D. Hilbert, *Methods of Mathematical Physics*, Interscience Publications. Wiley, New York, USA, 1953.
- [4] T.M. Cover, Geometrical and statistical properties of systems of linear inequalities with application in pattern recognition, *IEEE Trans. Electron. Comp.* 14 (1965) 326–334 (reprinted in: P. Mehra, B. Wah (Eds.), *Artificial Neural Networks: Concepts and Theory*, IEEE Computer Society Press, Los Alamitos, California, 1992).
- [5] H.-T. Lin, C.-J. Lin, A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods, Technical Report, Department of Computer Science and Information Engineering, National Taiwan University, available at (<http://www.csie.ntu.edu.tw/~cjlin/papers/tanh.pdf>). Last modification: April, 21st 2004. Last access: June, 1st 2004 (2003).
- [6] F. Pérez-Cruz, A. Artés-Rodríguez, A new optimizing procedure for ν -Support Vector Regressor, in: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, ICASSP'01*, vol. 2, Salt Lake City, UT, USA., 2001, pp. 1265–1268.
- [7] B. Schölkopf, A. Smola, *Learning with Kernels—Support Vector Machines, Regularisation, Optimization and Beyond*, The MIT Press Series, Cambridge, MA, 2001.
- [8] E. Soria, J. Martín, G. Camps, A.J. Serrano, J. Calpe, L. Chova, A low complexity fuzzy activation function for artificial neural networks, *IEEE Trans. Neural Networks* 14 (6) (2003) 1576–1579.
- [9] V.N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.
- [10] J. Weston, C. Watkins, Multi-class support vector machines, Technical Report CSD-TR-98-04, Dpt. of Computer Science. Royal Holloway—University of London, Egham, Surrey TW20 0EX, available at (<http://citeseer.ist.psu.edu/8884.html>). Last modification: June, 1st 2004. Last access: June, 1st 2004 (May 1998).