

Learning Non-linear Time-scales with Kernel γ -Filters

Gustavo Camps-Valls ^{a,1}, Jordi Muñoz-Marí ^a,
Manel Martínez-Ramón ^b, Jesús Requena-Carrión ^c,
José Luis Rojo-Álvarez ^c

^a*Dept. Enginyeria Electrònica. Univ. València. València, Spain.*

^b*Dept. Señal y Comunicaciones, Univ. Carlos III. Madrid, Spain.*

^c*Dept. Señal y Comunicaciones, Univ. Rey Juan Carlos. Madrid, Spain.*

Abstract

A family of kernel methods, based on the γ -filter structure, is presented for non-linear system identification and time series prediction. The *kernel trick* allows us to develop the natural non-linear extension of the (linear) Support Vector Machine (SVM) γ -filter [1], but this approach yields a rigid system model without non-linear cross relation between time-scales. Several functional analysis properties allow us to develop a full, principled family of kernel γ -filters. The improved performance in several application examples suggest that a more appropriate representation of signal states is achieved.

Key words: Gamma filter, Support Vector Machine, Kernel, Non-linear System Identification.

¹ Correspondence address: Gustavo Camps-Valls. Escola Tècnica Superior d'Enginyeria. Dept. Enginyeria Electrònica. Universitat de València. C/ Dr. Moliner, 50. Burjassot (València). Spain. Tel.: +34-963544021 - Fax: +34-963544353. E-mail address: gustavo.camps@uv.es.

1 Introduction

Inclusion of short-term memory in learning machines is essential for processing time-varying information. A remarkable compromise between stability and simplicity of adaptation can be provided by the γ -filter proposed in [2]. Previous works on γ -filters [2, 3] claimed two main advantages: First, they provide stable models, and second, they allow us to study the memory depth of a model. However, the original γ -filter underwent some practical problems when dealing with low number of (potentially noisy) samples in the time series. These shortcomings were alleviated in [1] by introducing the use of Support Vector Machines (SVM) to estimate the γ -filter coefficients, yielding a robust and stable filter that inherits all the good properties of the SVM methodology.

The promising results we obtained with the linear SVM γ -filter [1] encouraged us to extend its capabilities by using non-linear kernels, as is usual in the SVM framework in general [4], and in the digital signal processing SVM framework in particular [5, 6]. Many neural network structures with a *linear memory* stage followed by a non-linear memoryless stage are commonly used in some signal processing tasks, such as the time delay neural network and the focused γ -network [7, 8]. These networks offer good performance at the expense of increased dimensionality of the state vector of the linear memory stage, and hence, training the memoryless stage involves both high computational burden and risk of overfitting. Kernel methods in general, and SVMs in particular, deal with this problem efficiently [4]. In this paper, we introduce a set of kernel methods to develop non-linear γ -filters in a straightforward, principled way.

2 Non-linear γ -Filter with the Kernel Trick

Assume a standard linear γ -filter structure given by $y_n = \sum_{i=1}^P w_i x_n^i$, where $x_n^i = x_n$ for $i = 1$, and $x_n^i = (1 - \mu)x_{n-1}^i + \mu x_{n-1}^{i-1}$ for $i = 2, \dots, P$; where y_n is the output signal, x_n is the input signal, x_n^i is the signal present at the input of the i -th γ -tap, and μ is a free parameter. For an observed time series d_n , error signal e_n is defined as $e_n = d_n - y_n$. For $\mu = 1$, this structure reduces

to Widrow’s adaline, whereas, for $\mu \neq 1$, it has an IIR transfer function. In comparison to general IIR filters, the feedback structure in the γ -filter presents two complementary conditions: (a) *Locality*, since the loops are kept local with respect to the taps; and (b) *globality*, since all the loops have the same loop-gain $1 - \mu$. The stability is trivially obtained with $0 < \mu < 1$ for low-pass transfer functions, and with $1 < \mu < 2$ for high-pass transfer functions. The SVM γ -filter algorithm proposed in [1] uses the robust cost function previously presented in [9], which allows to deal with different kinds of noise simultaneously. The SVM γ -filter is expressed as

$$y_n = \sum_{s=P+1}^N (\alpha_s - \alpha_s^*) \sum_{i=1}^P x_n^i x_s^i, \quad (1)$$

where only samples with non-zero $\alpha_s^{(*)}$ count in the solution, and they are called *support vectors*. Note that one can identify the cross-correlation function of gamma-filtered signals $R_x^P(n, s) = \sum_{i=1}^P x_n^i x_s^i$, which can be conveniently expressed as a dot product involving the delayed and filtered versions of the input time-series. This leads to easily define the natural non-linear extension of the model by means of the “kernel trick”, which allows us to work in the mapped kernel space without knowing the mapping explicitly, but only the kernel matrix formed by the dot product of mapping functions. For this purpose, the delayed line outputs x_n^i are transformed into a high-dimensional feature space using a non-linear mapping ϕ . Here, $R_x^P(n, s) = \sum_{i=1}^P \phi(x_n^i)^\top \phi(x_s^i)$, and the solution becomes inherently non-linear:

$$y_n = \sum_{s=P+1}^N (\alpha_s - \alpha_s^*) \sum_{i=1}^P \phi(x_n^i)^\top \phi(x_s^i) = \sum_{s=P+1}^N (\alpha_s - \alpha_s^*) \sum_{i=1}^P K(x_n^i, x_s^i) \quad (2)$$

3 Non-linear γ -Filters with Composite Kernels

By following the previous approach, the underlying structure of the time series has been partially obviated, given that all the information among samples remains embedded into the kernel matrix. The theoretical properties of kernels are well known and extensively reviewed [10, 11, 12]. Note, however, that one should first analyze the linear structure of the signal problem, and then focus in

the application and design of the most suitable kernel functions [5]. In [6], non-linear SVM-ARMA system identification was addressed for both the statement of the autorregressive ARX equations in a reproducing kernel in Hilbert space RKHS, and the use of a set of composite kernels for improved flexibility. The same methodology was recently extended to a Bayesian framework in [13]. Here, we adapt this methodology for model design in order to develop a family of non-linear γ -filters.

Summation composite kernel. One can define the concatenation of non-linear transformations of the involved counterparts x_{n-1}^i and x_{n-1}^{i-1} as $\phi(\mathbf{x}_{n-1}) \equiv \{\varphi_1(x_{n-1}^i), \varphi_2(x_{n-1}^{i-1})\}$ where $\{\cdot, \cdot\}$ denotes concatenation of (weighted) column vectors, and $\varphi_1(\cdot)$, $\varphi_2(\cdot)$ are transformations into Hilbert spaces \mathcal{H}_1 and \mathcal{H}_2 . The corresponding dot product between vectors (or kernel) is simply

$$\begin{aligned} K(\mathbf{x}_{n-1}, \mathbf{x}_{k-1}) &= \phi(\mathbf{x}_{n-1})^\top \phi(\mathbf{x}_{k-1}) \\ &= \{\varphi_1(x_{n-1}^i), \varphi_2(x_{n-1}^{i-1})\}^\top \cdot \{\varphi_1(x_{k-1}^i), \varphi_2(x_{k-1}^{i-1})\} \\ &= K_1(x_{n-1}^i, x_{k-1}^i) + K_2(x_{n-1}^{i-1}, x_{k-1}^{i-1}) \end{aligned} \quad (3)$$

which leads to a sum of kernels, each of them being dedicated to measure the similarity among samples at different time scales. Clearly, if the chosen kernel is a linear transformation of concatenation $\mathbf{x}_{n-1} = \{x_{n-1}^i, x_{n-1}^{i-1}\}$, then the solution is equivalent to using the common kernel trick, which is a particular case of the proposed machine. The so-called *summation kernel* has been extensively used in the literature as a straightforward way of fusing heterogeneous information [14, 15].

Tensor product composite kernel. In the latter formulation, one only needs to make the summation of two matrices working independently with γ -filtered samples x_{n-1}^i and x_{n-1}^{i-1} . One can also design the non-linear SVM γ -based filters by considering the *tensor product kernel*, which can be expressed as:

$$K(\mathbf{x}_{n-1}, \mathbf{x}_{k-1}) = K_1(x_{n-1}^i, x_{k-1}^i) \otimes K_2(x_{n-1}^{i-1}, x_{k-1}^{i-1}) \quad (4)$$

This kernel offers an easy way to modulate the similarity at different tapped versions.

Cross-information composite kernel. Further kernel algorithms can be developed exploiting properties of Hilbert spaces [16]. In particular, one can build composite kernels to account for the cross-relationship between different scales of the γ -filtered signals. Assume a non-linear mapping $\varphi(\cdot)$ into a Hilbert space \mathcal{H} and three linear transformations \mathbf{A}_k from \mathcal{H} to \mathcal{H}_k , for $k = 1, 2, 3$. Let us make the composite vector $\boldsymbol{\phi}(\mathbf{x}_{n-1}) \equiv \{\mathbf{A}_1\varphi(x_{n-1}^i), \mathbf{A}_2\varphi(x_{n-1}^{i-1}), \mathbf{A}_3(\varphi(x_{n-1}^i) + \varphi(x_{n-1}^{i-1}))\}$ and compute the dot product

$$\begin{aligned} K(\mathbf{x}_{n-1}, \mathbf{x}_{k-1}) &= \boldsymbol{\phi}(\mathbf{x}_{n-1})^\top \boldsymbol{\phi}(\mathbf{x}_{k-1}) \\ &= \boldsymbol{\phi}(x_{n-1}^i) \mathbf{R}_1 \boldsymbol{\phi}(x_{k-1}^i) + \boldsymbol{\phi}(x_{n-1}^{i-1}) \mathbf{R}_2 \boldsymbol{\phi}(x_{k-1}^{i-1}) \\ &\quad + \boldsymbol{\phi}(x_{n-1}^i) \mathbf{R}_3 \boldsymbol{\phi}(x_{k-1}^{i-1}) + \boldsymbol{\phi}(x_{n-1}^{i-1}) \mathbf{R}_3 \boldsymbol{\phi}(x_{k-1}^i), \end{aligned} \quad (5)$$

where $\mathbf{R}_1 = (\mathbf{A}_1^\top \mathbf{A}_1 + \mathbf{A}_3^\top \mathbf{A}_3)$, $\mathbf{R}_2 = (\mathbf{A}_2^\top \mathbf{A}_2 + \mathbf{A}_3^\top \mathbf{A}_3)$, and $\mathbf{R}_3 = \mathbf{A}_3^\top \mathbf{A}_3$ are three independent positive definite matrices. Similarly to the previous composite kernel, one can demonstrate that (5) can be expressed as a sum of positive definite matrices:

$$\begin{aligned} K(\mathbf{x}_{n-1}, \mathbf{x}_{k-1}) &= K_1(x_{n-1}^i, x_{k-1}^i) + K_2(x_{n-1}^{i-1}, x_{k-1}^{i-1}) \\ &\quad + K_3(x_{n-1}^i, x_{k-1}^{i-1}) + K_3(x_{n-1}^{i-1}, x_{k-1}^i) \end{aligned} \quad (6)$$

The only restriction for this formulation to be valid is that x_{n-1}^i and x_{n-1}^{i-1} need to have the same dimension, which can be trivially assured by zero-padding the training patterns.

Extended composite kernels. Instead of using the proposed algorithms for kernel γ -based system identification separately, one can think of using the aforementioned mapping functions in a collaborative way. Two main possibilities arise:

- (1) By using three concatenated transformations: one for x_{n-1}^i , one for x_{n-1}^{i-1} , and one for their concatenation $z_{n-1} = [x_{n-1}^i, x_{n-1}^{i-1}]^\top$. The result is a combination between the kernel trick and the summation composite kernels described in the preceding sections. It is straightforward to see that the corresponding kernel is

$$K(\mathbf{x}_{n-1}, \mathbf{x}_{k-1}) = K_{zz}(z_{n-1}, z_{k-1}) + K_1(x_{n-1}^i, x_{k-1}^i) + K_2(x_{n-1}^{i-1}, x_{k-1}^{i-1}) \quad (7)$$

- (2) By defining a mapping that leads to the summation of the cross-terms composite kernel and the kernel trick matrix:

$$\begin{aligned}
K(\mathbf{x}_{n-1}, \mathbf{x}_{k-1}) = & K_{zz}(z_{n-1}, z_{k-1}) + K_1(x_{n-1}^i, x_{k-1}^i) + K_2(x_{n-1}^{i-1}, x_{k-1}^{i-1}) \\
& + K_3(x_{n-1}^i, x_{k-1}^{i-1}) + K_3(x_{n-1}^{i-1}, x_{k-1}^i)
\end{aligned} \tag{8}$$

Note that these two models contain all the relevant model information that can be extracted from the data by each counterpart kernel.

Differences with previous composite kernels. Though composite kernels have been paid attention in the kernel methods literature [10, 11, 12, 15], note that our emphasis is on the difference between applying the kernel trick and developing specific signal-based kernels. A proper design of composite kernels looking at the signal model structure allows us to develop adequate signal representations in the feature spaces, which may eventually lead to improved performance. Surprisingly, there are very few papers stating filter models and system identification machines from these principles, and basically, all the previous works [17, 18, 19] have directly applied the stacked (kernel trick) approach in these scenarios, ignoring the underlying signal relations in the model.

Remarks. In conclusion, composite kernels can be used to provide us with model flexibility in terms of emphasized consideration, if necessary, of the cross information among filtered signal states. Additional advantages are the robustness to high input dimension samples (high number of tapped delays), and the efficient processing of non-linear relations between input and output time discrete processes. It is straightforward to demonstrate that the proposed kernels are valid Mercer's kernels, constitute a convex cone, and as the Radial Basis Function (RBF) kernel is used in all cases, the composite kernels are isotropic and stationary. It should be also stressed that translation invariant kernels can be readily built for specific signal processing applications [20, 21].

4 Experimental Results

In this section, we compare the performance of the proposed kernel algorithms for non-linear system identification and time series prediction.

4.1 Model Development

Model building requires tuning different free parameters depending on the SVM formulation $(\sigma_{ker}, C, \varepsilon)$, and filter parameters (μ, P) . A non-exhaustive iterative search strategy (T iterations) was used here. At each iteration, a sequential search on each parameter domain is performed by splitting the range of the parameter in M linearly or logarithmically equally spaced points. Values of $T = 3$ and $M = 20$ exhibited good performance in our simulations. In each iteration we performed an 8-fold cross-validation and averaged the normalized MSE: $\text{nMSE} = \log_{10}(\sqrt{\frac{1}{N\hat{\sigma}^2} \sum_{i=1}^N (y_i - \hat{y}_i)^2})$, where the $\hat{\sigma}^2$ is the estimated variance of the data; and thus it removes the dependence on the dynamic range of the data. This normalization implies that if the estimated mean of the data is used as predictor, $\text{nMSE}=0$ is obtained. In all cases, we focus on the RBF kernel, which offers stable and fast solutions. The kernel is defined as $K(\mathbf{z}_i, \mathbf{z}_j) = \exp(-\|\mathbf{z}_i - \mathbf{z}_j\|^2 / (2\sigma_{ker}^2))$, where $\sigma_{ker}^2 \in \mathbb{R}^+$ represents the length scale or width of the kernel.

4.2 Numerical Comparison

Non-linear feedback system We first consider the following synthetic system. The input DTP is generated with Lorenz equations, given by $dx/dt = -\rho x + \rho y$, $dy/dt = -xz + rx - y$, and $dz/dt = xy - bz$, and using $\rho = 10$, $r = 28$, and $b = 8/3$. Only the x component is used as input signal to the system, and it goes forward through an 8th-order low-pass FIR filter, $H(z)$, with cutoff frequency $\omega_n = 0.5$ and normalized gain of -6dB at ω_n . No additive noise was introduced in this example. The output signal goes through a feedback loop consisting of a high-pass minimum-phase channel, $G(z) =$

$(1.00 + 2.01z^{-1} + 1.46z^{-2} + 0.39z^{-3})^{-1}$, with corresponding inverse time function $g(\cdot)$, and then distorted with non-linearity $f(\cdot) = \log(\cdot)$. This system was used to generate 10,000 input-output DTP samples $\{x_n, f(g(x_n))\}$, that were split into a training set (50) and a test set (following 500 samples). The experiment was repeated 100 times with randomly selected starting points in the DTP. Table 1 shows average results for all composite kernels. The best results are obtained with the summation kernel, followed by the kernel trick.

The Mackey-Glass time series Our second experiment deals with the standard Mackey-Glass time series prediction problem, which is well-known for its strong non-linearity. This classical high-dimensional chaotic system is generated by the delay differential equation: $dx/dt = -0.1x_n + 0.2x_{n-\Delta}/(1+x_{n-\Delta}^{10})$, with delays $\Delta = 17$ and $\Delta = 30$, thus yielding the time series **MG17** and **MG30**, respectively. We considered 500 training samples and used the next 1000 for free parameter selection (validation set), following the same approach as in [22]. Results are shown in Table 1 for both time series. The methods proposed here outperform the common kernel trick applied to the SVM linear filters, especially significant for the **MG17** time series. In the case of **MG30** differences are noticeable, and the summation kernel and the kernel trick combined with the cross-terms kernel performs best. This latter result suggests that a more complex model is necessary to obtain good results in the prediction of this time series, which holds more complex dynamics.

EEG prediction The third (and real-life) experiment deals with the EEG signal prediction 4-samples ahead. This is a very challenging non-linear problem with high levels of noise and uncertainty. We used file ‘SLP01A’ from the MIT-BIH Polysomnographic Database². The file contains 10000 samples. We used 100 training samples, the next 1000 samples were used for free parameter selection (validation set), and the rest for testing. The experiment was repeated at 100 randomly selected starting points. Average test results are shown in Table 1. It is observed that the kernel trick combined with the summation or cross-terms kernel perform best, thus suggesting that the high complexity of the underlying signal model has been captured by the model.

² Data available at <http://www.physionet.org/physiobank/database/slpdb/slpdb.shtml>

Table 1

nMSE for the kernel γ -filters in non-linear feedback system identification (NLSYS), Mackey-Glass time series prediction with $\Delta = 17$ and $\Delta = 30$, and EEG prediction. Bold and italics indicate the best and second best results for each problem, respectively. Results in [23] are also included for comparison.

Method	Poly	Rat	Loc1	Loc2	MLP	KT	SK	TP	CT	K+S	K+T
Eq						(2)	(3)	(4)	(6)	(7)	(8)
NLSYS	-0.04	-0.11	-0.12	-0.71	-0.78	<i>-1.23</i>	-1.26	-1.08	-1.005	-0.72	-1.06
MG17	-1.95	-1.14	-1.48	-1.89	-2.00	-2.33	-2.35	-2.33	-2.34	<i>-2.35</i>	-2.35
MG30	-1.40	-1.33	-1.24	-1.42	-1.50	-1.64	-1.75	-1.68	<i>-1.75</i>	-1.72	-1.69
EEG	-0.05	-0.13	-0.33	-0.32	-0.46	-0.49	-0.66	-0.68	<i>-0.73</i>	<i>-0.73</i>	-0.77

4.3 On Model Complexity and Non-linear Time-Scales

Attending to the numerical results (nMSE) in Table 1, one could identify EEG and NLSYS as high complexity problems, and MG17/30 as moderate complexity problems. However, different kernel structures may accommodate problem difficulty better than others. Certainly, characterizing model complexity and versatility is an important aspect for time series analysis. In this sense, Fig. 1 reports the results for the four non-linear time-series problems in terms of machine complexity (SVs [%]), needed tap delays (P), memory requirements (μ), and its attendant the memory depth ($M = P/\mu$), which quantifies the past information retained and it has units of time samples [2]. Problems NLSYS and MG30 need $> 75\%$ of SVs to attain significant results, while $< 40\%$ of SVs are needed for the MG17 and EEG problems. Besides, it is observed a clear trade-off between SVs and P . Interestingly, in the most complex problem (EEG), the kernel trick (KT) needs almost twice the number of tap delays than the best model in terms of accuracy (K+T). A similar balance is also observed between SVs and μ . This suggests that kernels model different time-scales either with more SVs or by reducing the value of μ .

The memory depth M serves to uncover the efficiency in modeling the (non-linear) time-scales. On the one hand, it is worth to note that in complex

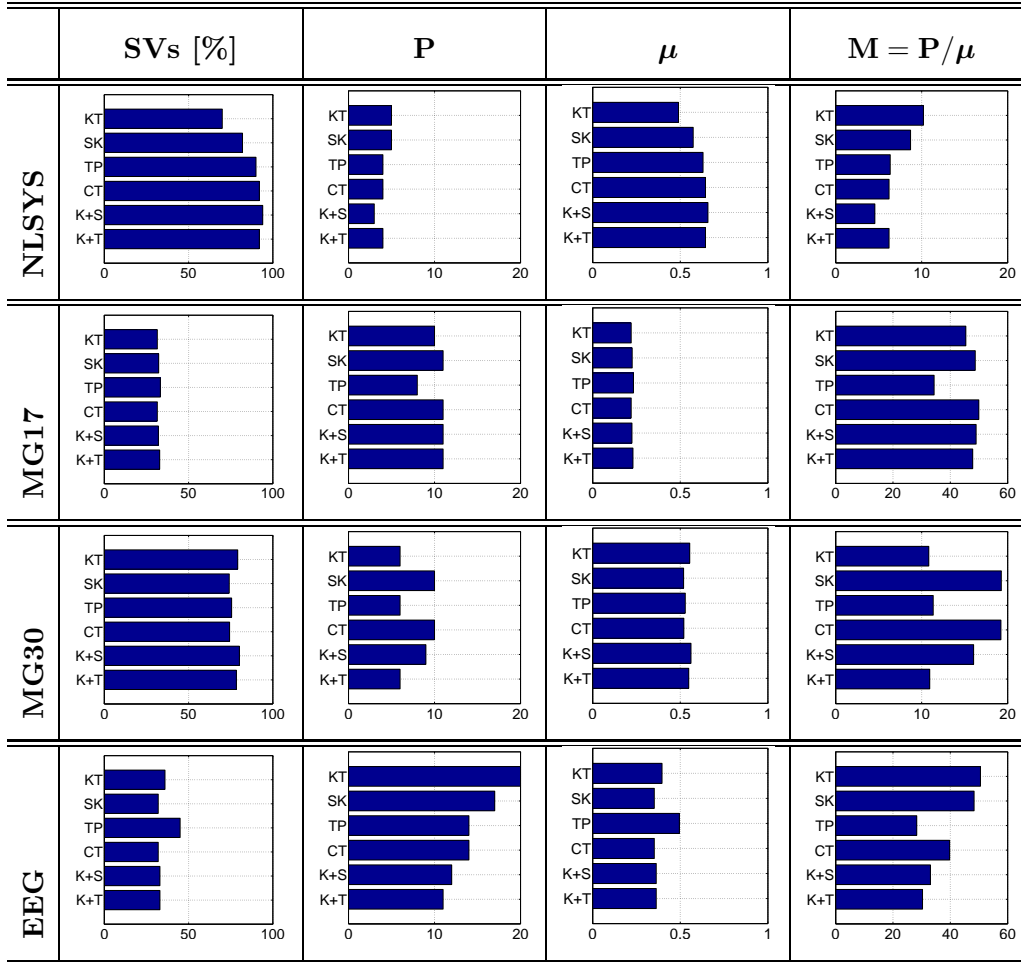


Fig. 1. Machine complexity (SVs [%]), tap delays (P), memory requirements (μ), and memory depth ($M = P/\mu$) for all kernel methods and problems.

problems (NLSYS and EEG) the kernel trick (KT) yields slightly higher values of M at the expense of poor numerical results (cf. Table 1). In these cases, a much better tradeoff is reported by using the proposed summation kernel (SK). On the other hand, in problems of moderate complexity (MG17 and MG30), the summation and the cross-terms kernels clearly yield better time-scales modeling along with improved accuracies. We should finally state that the tensor product offers, in general, uneven results, probably due to a poor modeling of the problem non-linear time-scales (either high μ or low P values are selected). Similarly, the use of highly complex models (K+S and K+T) do not necessarily imply improved performance or higher memory capabilities in the tested problems.

5 Conclusions

In this paper, we introduced a general class of non-linear γ -based filters based on kernel methods for non-linear system identification, which include the SVR case as a particular case. Given that the robustness capabilities for the linear SVM γ -filter in [1] are a direct consequence of the use of the SVM methodology (regularization and loss function), they also hold in the non-linear versions. This is why we focused on how the proper design of the kernel leads to an improved modeling of the non-linear time-scales. Experiments show that the application of the introduced composite kernels provide a substantial improvement with respect to the straightforward extension of the linear γ -filter based on the kernel trick, not only in terms of prediction accuracies but also in better compromise with memory depth and model complexity. The results confirm that the proposed methods fit the non-linear time-scales of the problem much better than by using the kernel trick, which merely operates with local (consecutive) samples. These results encourage the development of kernel versions of other filter structures.

References

- [1] G. Camps-Valls, M. Martínez-Ramón, J. L. Rojo-Álvarez, and E. Soria-Olivas. Robust γ -filter using support vector machines. *Neurocomputing Journal*, 62(12):493–499, Dec 2004.
- [2] J. C. Principe, B. deVries, and P. G. deOliveira. The gamma filter – A new class of adaptive IIR filters with restricted feedback. *IEEE Transactions on Signal Processing*, 41(2):649–656, Feb 1993.
- [3] M. Palkar and J. Principe. Echo cancellation with the gamma filter. In *Proc. International Conference on Acoustics, Speech and Signal Processing, ICASSP94*, volume 3, pages 369–372, Adelaide, Australia., 1994.
- [4] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.
- [5] J. L. Rojo-Álvarez, G. Camps-Valls, M. Martínez-Ramón, E. Soria-Olivas, A. Navia-Vázquez, and A.R. Figueiras-Vidal. Support vector machines

- framework for linear signal processing. *Signal Processing*, 85(12), Dec 2005.
- [6] M. Martínez-Ramón, J. L. Rojo-Álvarez, G. Camps-Valls, A. Navia-Vázquez, E. Soria-Olivas, and A. R. Figueiras-Vidal. Support vector machines for nonlinear kernel ARMA system identification. *IEEE Transactions on Neural Networks*, 2007.
- [7] J. Principe, L. Giles, N. Morgan, E. Wilson, S. Lawrence, A. D. Back, A. C. Tsoi, and C. L. Giles. The Gamma MLP using multiple temporal resolutions for improved classification. In *IEEE Workshop on Neural Networks for Signal Processing VII*, pages 362–367, 1997.
- [8] B. de Vries and J. C. Principe. The Gamma model – a new neural model for temporal processing. *Neural Networks*, 5(4):565–576, 1992.
- [9] J. L. Rojo-Álvarez, M. Martínez-Ramón, A. R. Figueiras-Vidal, A. García-Armada, and A. Artés-Rodríguez. A robust support vector algorithm for non-parametric spectral analysis. *IEEE Signal Processing Letters*, 10(11):320 – 323, Nov 2003.
- [10] M. G. Genton. Class of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2, Dec 2001.
- [11] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *24th International Conference on Machine Learning*, pages 775–782, Corvallis, Oregon, 2007. ACM.
- [12] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7, Jul 2006.
- [13] G. Camps-Valls, M. Martínez-Ramón, J. L. Rojo-Álvarez, and J. Muñoz-Marí. Non-linear system identification with composite relevance vector machines. *IEEE Signal Processing Letters*, 14(4):279–282, April 2007.
- [14] B. Mak, J.T. Kwok, and S. Ho. A study of various composite kernels for kernel eigenvoice speaker adaptation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP04*, volume 1, pages 325–8. IEEE, May 2004.
- [15] C. Longworth and M.J.F. Gales. Multiple kernel learning for speaker verification. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 1581–1584. IEEE, 2008.

- [16] M. C. Reed and B. Simon. *Functional Analysis*, volume I of *Methods of Modern Mathematical Physics*. Academic Press, 1980.
- [17] P. M. L. Drezet and R. F. Harrison. Support vector machines for system identification. In *UKACC International Conference on Control'98*, volume 1, pages 688–692, Swansea, UK, Sept 1998.
- [18] A. Gretton, A. Doucet, R. Herbrich, P.J.W. Rayner, and B. Schölkopf. Support Vector Regression for Black-Box System Identification. In *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, pages 341–4, 2001.
- [19] J. A. K. Suykens. Support vector machines: a nonlinear modelling and control perspective. *European Journal of Control, Special Issue on fundamental issues in control*, 7(2-3):311–327, Aug. 2001.
- [20] J. L. Rojo-Álvarez, C. Figuera, C. Martínez, G. Camps-Valls, F. Alonso-Atienza, and M. Martínez-Ramón. Nonuniform interpolation of noisy signals using support vector machines. *IEEE Transactions on Signal Processing*, 55(8), Aug 2007.
- [21] J. L. Rojo-Álvarez, M. Martínez-Ramón, C. Martínez, J. Muñoz Marí, G. Camps-Valls, and A. R. Figueiras-Vidal. Sparse deconvolution using support vector machines. *EURASIP JASP*, 2008.
- [22] S. Mukherjee, E. Osuna, and F. Girosi. Nonlinear prediction of chaotic time series using a support vector machine. In J. Principe, L. Gile, N. Morgan, and E. Wilson, editors, *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop*, New York, 1997. IEEE.
- [23] M. Casdagli and S. Eubank. Nonlinear modeling and forecasting. In *Proceedings of Santa Fe Institute Studies in the Science of Complexity*, volume XII. Addison–Wesley, Reading, MA, November 1992.