Regularized Recurrent Least Squares Support Vector Machines

Hai-Ni Qu[1] ， Yacine Oussar[2]
[1]College of Electronics and Information Engineering
Tongji University
Shanghai , China;
e-mail: hattiequ@126.com

Gerard Dreyfus[2]， Weisheng Xu[1]
[2]ESPCI-Paristech, Paris 75005, France.

***Abstract*—Support vector machines are widely used for classification and regression tasks. They provide reliable static models, but their extension to the training of dynamic models is still an open problem. In the present paper, we describe Regularized Recurrent Support Vector Machines, which, in contrast to previous Recurrent Support Vector Machine, models, allow the design of dynamical models while retaining the built-in regularization mechanism present in Support Vector Machines. The principle is validated on academic examples; it is shown that the results compare favorably to those obtained by unregularized Recurrent Support Vector Machines and to regularized, partially recurrent Support Vector Machines.**

Keywords- recurrent least squares support vector machines; dynamic systems; machine learning; modeling

I. INTRODUCTION

Recurrent least squares support vector machines (RLSSVMs) were first described in [1], and were further discussed in [2]. As in standard least squares support vector machines, the constraints are equality constraints (instead of inequalities in standard SVMs), but, in addition, the recursion that is necessary to design dynamic models is taken into account. However, for simplification purposes, the regularization term was neglected altogether, thereby losing one of the attractive features of the SVM framework. As a result, the authors had to resort to conventional regularization tricks such as early stopping. In [3-4], different simplifying assumptions were also made.

In the present paper, we show that such simplifications are not necessary. We derive the exact equations that describe the problem and the constraints, and we show that satisfactory performances can be obtained while retaining the regularization mechanism of support vector machines.

This paper is organized as follows: recurrent least squares support vector machines are first recalled. The simplified versions of RLSSVMs are subsequently presented: unregularized recurrent LSSVMs (as derived in [1] and [2]) and regularized partially recurrent LSSVMs (as described in [3]-[4]). Regularized, fully recurrent LSSVMs are derived, and some illustrative examples are provided.

II. RECURRENT LEAST SQUARES SUPPORT VECTOR MACHINES

The present section reviews the basic concepts of recurrent least squares support vector machines (RLSSVMs) [1].

Given a deterministic nonlinear single-input single-output dynamical system with measured inputs $u_k \in R$ and measured outputs $y_k \in R$, where k is a positive integer, we consider a nonlinear model of the form:

$$\hat{\hat{y}_k} = f\left(y_{k-1},.....,y_{k-P},u_{k-1},.....,u_{k-M}\right) = f\left(\mathbf{z}_k\right) \quad (1)$$

where $\hat{y}_k$ is the process output at time k predicted by the model, P is the order of the model, and M is a positive integer. Therefore, vector $\mathbf{z}_k$ is of size $P + M$. Such a model is called Nonlinear Auto-Regressive with eXogenous inputs (NARX). Since relation (1) is a recurrent discrete-time equation, such a model belongs to the family of recurrent models.

In order to design such a model, a function f must be postulated. In the present framework, we postulate a parameterized function of the form:

$$f\left(\mathbf{x}\right) = \mathbf{w}^T \varphi\left(\mathbf{x}\right) + b \quad (2)$$

where $\mathbf{w}$ is a vector of parameters, b is a scalar parameter, and $\boldsymbol{\varphi}(\mathbf{x})$ is a "feature vector" derived from the primary variables $\mathbf{x}$. Relation (1) can be rewritten as:

$$\hat{y}_k = \mathbf{w}^T \varphi\left(\mathbf{z}_k\right) + b \quad (3)$$

Given a postulated function $\boldsymbol{\varphi}$, training is intended to provide a vector of parameters such that the model predicts correctly the data sequences used for training, and, in addition, provides satisfactory predictions for other sequences drawn from the same probability distributions.

In the framework of least squares support vector machines, training is cast into the form of a constrained optimization problem: minimize the cost function

$$J\left(\mathbf{w},b,\mathbf{e}\right) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{\gamma}{2}\sum_{k=1}^{N} e_k^2 \quad (4)$$

subject to the constraints

$$\hat{y}_k = y_k - e_k = \mathbf{w}^T \varphi\left(\mathbf{z}_k\right) + b \text{ , } k = 1, 2 \ldots \text{N} \quad (5)$$

where e_k (component k of vector $\mathbf{e}$) is the prediction error at time k, and N is the length of the training sequence. γ is a positive real constant that is intended to provide an

appropriate tradeoff between the regularization term $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ and the accuracy term $\frac{1}{2}\sum_{k=1}^{N} e_k^2$, thereby providing both accurate predictions on the training set and satisfactory generalization on test data. Thus, SVM training has a built-in regularization mechanism.

Combining cost function (4) and constraints (5), the Lagrange function is derived

$$\begin{aligned} L(\mathbf{w},b,\mathbf{e};\alpha) &= J(\mathbf{w},b,\mathbf{e}) + \sum_{i=1}^{N}\alpha_i\left[y_i - e_i - \mathbf{w}^T\varphi(\mathbf{z}_i) - b\right] \\ &= \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{\gamma}{2}\sum_{k=1}^{N} e_k^2 + \sum_{i=1}^{N}\alpha_i\left[y_i - e_i - \mathbf{w}^T\varphi(\mathbf{z}_i) - b\right] \end{aligned} \tag{6}$$

where $\boldsymbol{\alpha}$ is the vector of Lagrange multipliers, which can be either positive or negative.

Noting that $\mathbf{z}_i$ can be written as

$$\mathbf{z}_i = \left[y_{i-1} - e_{i-1},\ y_{i-2} - e_{i-2},\ \ldots, y_{i-P} - e_{i-P},\ u_{i-1},\ \ldots, u_{i-M}\right],$$

the stationarity conditions of the Lagrange function can be derived:

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N}\alpha_i\varphi(\mathbf{z}_i) = 0 \\ \dfrac{\partial L}{\partial b} = \sum_{i=1}^{N}\alpha_i = 0 \\ \dfrac{\partial L}{\partial e_k} = \gamma e_k - \alpha_k - \sum_{j=1}^{P}\alpha_{j+k}\dfrac{\partial}{\partial e_k}\left[\mathbf{w}^T\varphi(\mathbf{z}_{j+k}) - b\right] = 0,\ k = 1,\ldots,N \\ \dfrac{\partial L}{\partial \alpha_k} = y_k - e_k - \mathbf{w}^T\varphi(\mathbf{z}_k) - b = 0,\ k = 1,\ldots,N \end{cases} \tag{7}$$

Assuming that there exists a feasible solution $(\mathbf{w}^*, b^*, \mathbf{e}^*)$, the model can be written as

$$\hat{y}_k = \sum_{i=1}^{N}\alpha_i^* K(\mathbf{z}_i, \mathbf{z}_k) + b^*$$

where the kernel function K is defined by $K(\mathbf{x},\mathbf{y}) = \varphi(\mathbf{x})\cdot\varphi(\mathbf{y})$.

III. SOLVING THE NONLINEAR EQUATIONS

Clearly, the critical point is the computation of the partial derivatives in the third stationarity condition. Several approaches to that problem have been described in the literature.

A. Unregularized recurrent LSSVMs

In [1], the problem is simplified drastically by setting γ to infinity. The optimization problem reduces to minimizing

$$J(\mathbf{w},b,\mathbf{e}) = \frac{1}{2}\sum_{k=1}^{N} e_k^2 \tag{8}$$

under the constraints

$$\begin{cases} \sum_{i=1}^{N}\alpha_i = 0 \\ y_k - e_k - \mathbf{w}^T\varphi(\mathbf{z}_k) - b = 0 \end{cases} \tag{9}$$

Since the regularization mechanism of SVMs is deleted by setting γ to infinity, another regularization scheme, such as early stopping, must be implemented in order to avoid overfitting

B. Regularized partially recurrent LSSVMs

In [3], the recurrence is not taken into account in the computation of $\frac{\partial L}{\partial e_k}$, i.e. the summation term is ignored. Therefore, the nonlinear equations to be solved are:

$$\begin{cases} \dfrac{\partial L}{\partial b} = \sum_{i=1}^{N}\alpha_i = 0 \\ \dfrac{\partial L}{\partial e_k} = \gamma e_k - \alpha_k = 0,\ k = 1,2,\ldots\ldots N \\ \dfrac{\partial L}{\partial \alpha_k} = y_k - e_k - \mathbf{w}^T\varphi(\mathbf{z}_k) - b = 0,\ k = 1,\ldots,N \end{cases} \tag{10}$$

C. Regularized recurrent LSSVMs

In the present section, we relax the assumptions of the previous two approaches, and we derive the exact set of nonlinear equations that must be solved

$$\begin{aligned} \frac{\partial L}{\partial e_k} &= \gamma e_k - \alpha_k - \sum_{j=1}^{P}\alpha_{j+k}\frac{\partial}{\partial e_k}\left[\mathbf{w}^T\varphi(\mathbf{z}_{j+k}) - b\right] \\ &= \gamma e_k - \alpha_k - \sum_{j=1}^{P}\alpha_{j+k}\sum_{n=1}^{N}\alpha_n\left[\varphi(\mathbf{z}_n)\right]^T\frac{\partial}{\partial e_k}\varphi(\mathbf{z}_{j+k}) \\ &= \gamma e_k - \alpha_k - \sum_{j=1}^{P}\alpha_{j+k}\sum_{n=1}^{N}\alpha_n\left[\varphi(\mathbf{z}_n)\right]^T\frac{\partial\varphi(\mathbf{z}_{j+k})}{\partial \mathbf{z}_{j+k}}\frac{\partial \mathbf{z}_{j+k}}{\partial e_k} \\ &= \gamma e_k - \alpha_k - \sum_{j=1}^{P}\alpha_{j+k}\frac{\partial}{\partial \mathbf{z}_{j+k}}\sum_{n=1}^{N}\alpha_n\left[\varphi(\mathbf{z}_n)\cdot\varphi(\mathbf{z}_{j+k})\right]^T\frac{\partial \mathbf{z}_{j+k}}{\partial e_k} \\ &= \gamma e_k - \alpha_k - \sum_{j=1}^{P}\sum_{n=1}^{N}\alpha_{j+k}\alpha_n\frac{\partial}{\partial \mathbf{z}_{k+j}}K(\mathbf{z}_{j+k},\mathbf{z}_n)\frac{\partial \mathbf{z}_{j+k}}{\partial e_k} \end{aligned} \tag{11}$$

Different kernel functions lead to different forms for relation (11), which, together with the other three stationarity conditions in (7), must be solved numerically.

IV. NUMERICAL EXPERIMENTS

In this section, we illustrate various aspects of regularized recurrent LSSVMs on academic problems. Results on real-life problems will be described elsewhere. All simulations were implemented in the MATLAB environment running on HP Compaq dc7600. The nonlinear equations (7) were solved numerically using the 'fsolve' function in the Optimization Toolbox.

A. Noise-free process

In this section, two simulated processes, linear and nonlinear, without noise, are modeled.

Experiment 1: We consider a linear single-input-single-output simulated process. Data sequences are generated by

$$y(k) = 0.8y(k-1) - 0.5y(k-2) - 1.2u(k-1) \quad k = 3,\ 4,\ ...N \tag{12}$$

where $u(k)$ is the exogenous input, $y(k)$ the process output, and where $y(1) = y(2) = 0$. A training sequence of length N_T and a validation sequence of length N_V are generated. The input vector $\mathbf{z}_k$ has three components:

$$\mathbf{z}_k = \begin{bmatrix} y_{k-1} - e_{k-1} & y_{k-2} - e_{k-2} & u_{k-1} \end{bmatrix}^T .$$

Fig. 1 and Fig. 2 show the responses of the simulated process to a pseudo-random input signal.

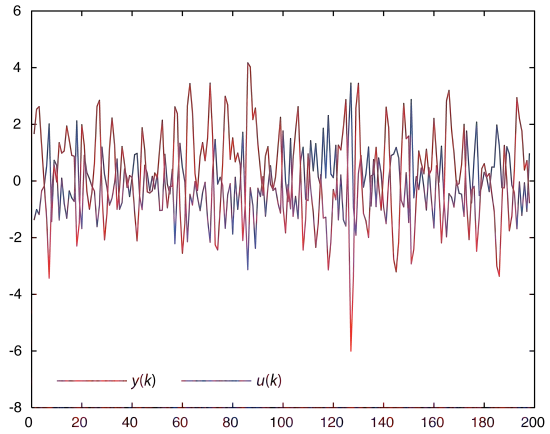

Figure 1. Training sequence.

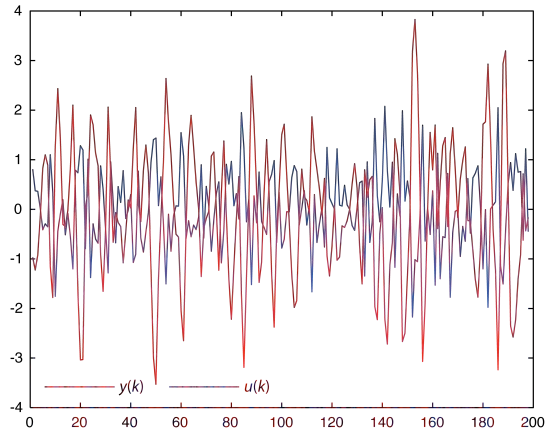

Figure 2. Validation sequence.

The performance of the model is assessed by the by root-mean-square-errors on the training sequence (TMSE) and on the validation sequence (VMSE),

$$TMSE = \sqrt{\frac{\sum_{k=1}^{N_T} (y_k - \hat{y}_k)^2}{N_T}};\ VMSE = \sqrt{\frac{\sum_{k=1}^{N_V} (y_k - y_k)^2}{N_V}} \tag{13}$$

Since a linear model is sought, the linear kernel function $K(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$ is used. The value of the regularization constant γ is varied from 0.1 ("strong" regularization) to 300 ("weak" regularization). The results are shown in Table 1.

TABLE I. PERFORMANCE OF REGULARIZED RLSSVMS FOR DIFFERENT VALUES OF γ

γ	TMSE	VMSE
0.1	0.0422	0.037
10	0.0004	0.0004
300	0	0

As expected, the training error decreases as regularization decreases. Similarly, the validation error decreases with decreasing regularization: since the generating model is linear and the predictive model is also linear, no overfitting occurs. For γ = 300, the optimal parameters of the model are equal to the parameters of the generating process (within roundoff errors). Therefore, the regularized recurrent LSSVM provides the best achievable result.

Experiment 2: We consider the following nonlinear data generating process [5]:

$$y(k) = \left(1 - \frac{T}{a + by(k-1)}\right) y(k-1) + T\frac{c + dy(k-1)}{a + by(k-1)} u(k-1) \tag{14}$$

with a = –0.139, b = 1.2, c = 5.633, d = –0.326, T = 0.1 sec. As in the previous example, two data sequences are generated for training and validation, as shown in Fig. 3 and Fig. 4.

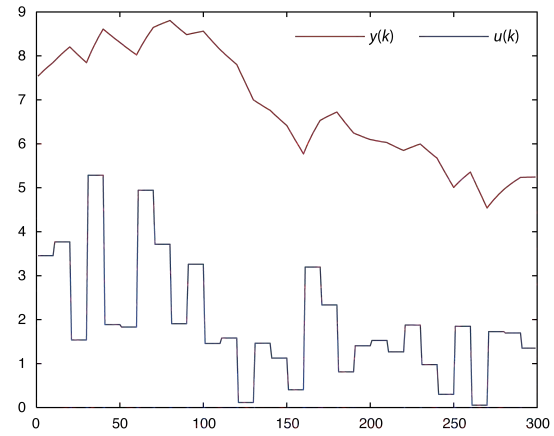

Figure 3. Training sequence

In this experiment, the predictive model is sought with the polynomial kernel function $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^n$ (n=2). In that case, the regression function (a rational fraction) does not belong to the family of polynomials. As a result, the influence of the regularization parameter γ is different from the previous example: as regularization decreases, i.e. as the effective complexity of the predictive model increases, the model tends to overfit the data, as evidenced by the increase of the VMSE (Table II).

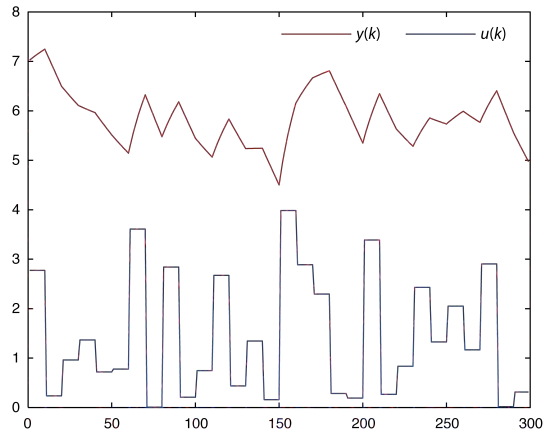

Figure 4. Validation sequence

TABLE II. PERFORMANCE OF REGULARIZED RLSSVMS FOR DIFFERENT VALUES OF γ

γ	TMSE	VMSE
0.001	0.5193	0.3821
0.01	0.1496	0.2222
0.1	0.0173	0.0197
1	0.0132	0.0132
10	0.0131	0.0143
100	0.0130	0.0145
1000	0.0101	0.0154

Clearly, the presence of the regularization term in RLSSVMs provides an important means of controlling the effective complexity of the machine learning model.

B. Process with output noise

We consider a nonlinear data generating process whose deterministic part is the same as in the previous example, but with additive output noise:

$$x(k) = \varphi\left[x(k-1), u(k-1)\right]$$
$$y(k) = x(k) + w(k) \quad (15)$$

where $w(k)$ is white noise with standard deviation 0.1. As usual (see for instance [6]), the purpose of training is to find a model with minimal prediction error, i.e. a model whose prediction error is as close as possible to the variance of the noise present in the training data. As in the previous case, the polynomial kernel $K\left(\mathbf{x},\mathbf{y}\right)=\left(\mathbf{x}\cdot\mathbf{y}+1\right)^n$, with $n = 2$, is used. Table III shows the results obtained, by regularized partially recurrent LSSVMs and regularized recurrent LSSVMs. Regularized fully recurrent LSSVMs provide a predictive model whose VMSE is closer to the standard deviation of the noise than the model obtained by partially recurrent LSSVMs, albeit at the expense of a longer computation time.

TABLE III. COMPARISON OF REGULARIZED FULLY RECURRENT LLSSVMS AND REGULARIZED PARTIALLY RECURRENT LSSVMS (γ= 1)

	TMSE	VMSE
Noise	0.101 (standard deviation)	0.103 (standard deviation)
Regularized fully recurrent LSSVM	0.098	0.109
Regularized partially recurrent LSSVM	0.109	0.115

V. CONCLUSIONS

In the present paper, regularized recurrent LSSVMs have been presented, and compared both to unregularized recurrent LSSVMs and to regularized partially recurrent LSSVMs. We have shown that the computations involved in RLSSVMs are tractable analytically without having to resort to approximations. As a result, capacity control can be more effectively achieved than with previous RLSSVM models.

However, it should be noticed that the uniqueness of the solution, which is an attractive feature of support vector machines, is lost due to the recurrence that is necessary for dynamic modeling. Therefore, several parameter initializations are required. In addition, the approach requires large computations times that do not scale nicely with the length of the training sequence. Future work will be directed towards gaining computational efficiency without losing the elegance and accuracy of the method.

ACKNOWLEDGMENT

The authors would like to thank Prof. Guo-Zheng Li for his valuable advices. This work was supported by the Natural Science Foundation of China under grant no. 70871091, the STCSM "Innovation Action Plan" Project of China under grant no. 07DZ19726.

REFERENCES

[1] J.A.K. Suykens and J. Vandewalle, "Recurrent least squares support vector machines", *IEEE Transactions on Circuits and Systems-Fundamental Theory and Applications*, 47, 1109-1114 (2000).

[2] Suykens JAK "Support Vector Machines: A nonlinear modelling and control perspective" *European Journal of Control* 7, 311-327 (2001).

[3] Sun JC, Zhou YT, Bai YH, et al "Nonlinear noise reduction of chaotic time series based on multi-dimensional recurrent least squares support vector machines" *Neural Information Processing, Lecture Notes in Computer Science* 4232, 900-906 (2006)

[4] Sun JC, Yu L, Yang G, et al. "Modelling of Chaotic Systems with recurrent least squares support vector machines combined with stationary wavelet transform" *Advances in Neural Networks, Lecture Notes in Computer Science* 3497, 424-429 (2005).

[5] O. Nerrand, D. Urbani, P. Roussel-Ragot, L. Personnaz, G. Dreyfus, "Training Recurrent Neural Networks: Why and How? An Illustration in Process Modeling", IEEE Transactions on Neural Networks, 178-184 (1994).

[6] Gérard Dreyfus, Neural networks: methodology and applications, Springer (2005).