Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 50-1 (2017) 8642–8647

A Novel Online LS-SVM Approach for Regression and Classification

Erdem Dilmen * **Selami Beyhan** **

* *Pamukkale University, Department of Mechatronics Engineering, Denizli 20020, Turkey (e-mail: edilmen@pau.edu.tr)*
** *Pamukkale University, Department of Electrical and Electronics Engineering, Denizli 20070, Turkey (e-mail: sbeyhan@pau.edu.tr)*

Abstract: In this paper, a novel online least squares support vector machine approach is proposed for classification and regression problems. Gaussian kernel function is used due to its strong generalization capability. The contribution of the paper is twofold. As the first novelty, all parameters of the SVM including the kernel width parameter σ are trained simultaneously when a new sample arrives. Unscented Kalman filter is adopted to train the parameters since it avoids the sub-optimal solutions caused by linearization in contrast to extended Kalman filter. The second novelty is the variable size moving window by an intelligent update strategy for the support vector set. This provides that SVM model captures the dynamics of data quickly while not letting it become clumsy due to the big amount of useless or out-of-date support vector data. Simultaneous training of the kernel parameter by unscented Kalman filter and intelligent update of support vector set provide significant performance using small amount of support vector data for both classification and system identification application results.

Keywords: Support vector machine, Online classification and regression, Adaptive kernel function, Real-time system identification, UKF.

1. INTRODUCTION

The conventional SVM model is constructed based on the quadratic programming (QP), Vapnik (1995), that minimizes a convex cost function with inequality constraints. As an alternative to conventional QP-based SVM, Suykens and Vandewalle (1999) proposed least squares SVM (LS-SVM) which is based on a set of linear equations with equality constraints. In this paper, LS-SVM is adopted to work online due to its straightforward structure with a rather simple algebraic expression which is solved optimally by least squares.

Choosing the optimal kernel parameter has been a critical problem since it has a huge effect on classification or regression performance of the SVM. Some authors developed solutions to this. Lin et al. (2006) proposed an adaptive fuzzy kernel function for offline classification. Training consists of three phases and after the initial fuzzy rules are derived, SVM is employed using that fuzzy kernel. Chapelle et al. (2002) proposed a minimax approach such that after the margin hyperplane is maximized by SVM training, an estimate of the generalization error is minimized over the set of kernel parameters which is performed using gradient descent optimization. Mu and Nandi (2006) tuned the kernel parameters by EKF with k-fold cross validation, which means k-sub-SVM classifiers. Once the pre-determined stopping criteria is met, final SVM model is employed for the classification task. Wang et al. (2003) calculated the optimal σ value for Gaussian kernel by pre-analysis of data. Then, that optimal parameter is adopted in the kernel function while SVM is trained. Different pre-analysis approaches were proposed for classification and regression separately. Once SVM is trained, kernel is modified and SVM is retrained using the modified kernel for classification. Kernels using local correlations were proposed to incorporate prior knowledge in SVM learning by Scholkopf et al. (1998). Forecasting performance of ε-insensitive support vector regressor (SVR) is improved by a hybrid algorithm called chaotic genetic algorithm (CGA) by Hong et al. (2013). C, σ and ε parameters are determined optimally via CGA. In addition, same parameters are optimally tuned via chaotic particle swarm optimization (CPSO) by Hong (2009).

The solutions mentioned above consist of pre-analysis of data, additional model for kernel evaluation or sequential optimization to improve the kernel function. No such study exists that it adjusts all the SVM parameters including the ones corresponding to the kernel function simultaneously. First novelty proposed in this paper arises as a solution to fullfill this need in the literature. $\boldsymbol{\alpha}$, b and the Gaussian kernel width parameter σ in the LS-SVM are trained by UKF as a multi-input multi-output (MIMO) optimization problem simultaneously at each time instant when a new sample arrives. Sparseness in the online LS-SVM is maintained by an intelligent incremental/decremental update of the support vector set, which is the second novelty proposed in this paper.

The remainder of paper is organized as follows. Section 2 reviews the LS-SVM model for classification and regression, unscented Kalman filter (UKF), UKF based SVM training and adaptive windowing algorithm. Section 3 applies online LS-SVM to classification and system identification. Finally, Section 4 summarizes this paper.

10.1016/j.ifacol.2017.08.1521

2. ONLINE SUPPORT VECTOR MACHINE

This section introduces the details of UKF based online LS-SVM model. In Section 2.1 batch LS-SVM is briefly presented for classification and regression. UKF algorithm is given in Section 2.2. The novel UKF training of LS-SVM for both classification and regression cases is given in Section 2.3. And novel intelligent update strategy for the support vector set is given in Section 2.4.

2.1 Batch LS-SVM

Batch LS-SVM is well-known in the literature. It will be briefly given for classification and regression from Suykens et al. (2002).

LS-SVM Classification Consider we have N data pairs $\{\mathbf{x}_k, y_k\}_{k=1}^N$ where $\mathbf{x}_k \in R^d$ and $y_k \in \{-1, +1\}$. Equality constraint based QP problem is given by

$$\begin{aligned} \min_{\mathbf{w},b,\mathbf{e}} \mathrm{L} = \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda\frac{1}{2}\sum_{k=1}^{N} e_k^2 \\ \text{Const.}: \quad & y_k[\mathbf{w}^T\varphi(\mathbf{x}_k) + b] = 1 - e_k, k = 1, \ldots, N. \end{aligned} \tag{1}$$

In (1), e_k is the error variable and λ is the regularization parameter which penalizes the error. $\varphi(.)$ is a nonlinear mapping from the input space to a higher dimensional feature space. $\mathbf{w}$ is the weighting vector in the dimension of feature space and b is the bias term. Lagrangian equation is obtained as follows.

$$\pounds(\mathbf{w}, b, \mathbf{e}, \boldsymbol{\alpha}) = L(\mathbf{w}, b, \mathbf{e}) - \sum_{k=1}^{N} \alpha_k\{y_k[\mathbf{w}^T\varphi(\mathbf{x}_k) + b] - 1 + e_k\} \tag{2}$$

In (2) α_k are the Lagrange multipliers. The Karush-Kuhn-Tucker conditions for optimality are as follows.

$$\begin{cases} \frac{\pounds}{\mathbf{w}} = 0, \mathbf{w} = \sum_{k=1}^{N} \alpha_k y_k \varphi(\mathbf{x}_k) \\ \frac{\pounds}{b} = 0, \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \frac{\pounds}{e_k} = 0, \alpha_k = \lambda e_k, k = 1, \ldots, N \\ \frac{\pounds}{\alpha_k} = 0, y_k[\mathbf{w}^T\varphi(\mathbf{x}_k) + b] - 1 + e_k = 0, k = 1, \ldots, N \end{cases} \tag{3}$$

Combining (2) and (3), a set of linear equations is obtained as in (4).

$$\begin{bmatrix} 0 & \mathbf{Y}^T \\ \mathbf{Y} & \boldsymbol{\Upsilon} + \lambda^{-1}\mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix} \tag{4}$$

where $\mathbf{Y} = [y_1 \quad y_2 \quad \ldots \quad y_N]^T$ and

$$\Upsilon_{kl} = y_k y_l K(\mathbf{x}_k, \mathbf{x}_l) \tag{5}$$

$$K(\mathbf{x}_k, \mathbf{x}_l) = \varphi(\mathbf{x}_k)^T\varphi(\mathbf{x}_l) \tag{6}$$

In (6), K(.,.) is a kernel function which is an alternative to the inner product of the mapping function $\varphi(.)$. It aviods the necessity of exact knowledge about $\varphi(.)$. Several kernel functions exist, e.g. Gauss, polynomial. They must satisfy the Mercer conditions, they must be positive semi-definite. Their success depend on the data processed. $\boldsymbol{\alpha}$ ve b, are the LS solution to (4) and LS-SVM classifier output is obtained as follows.

$$y(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k y_k K(\mathbf{x}_k, \mathbf{x}) + b \tag{7}$$

LS-SVM Regression For the regression case, equality constraint based QP problem is given by

$$\begin{aligned} \min_{\mathbf{w},b,\mathbf{e}} \mathrm{L} = \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \lambda\frac{1}{2}\sum_{k=1}^{N} e_k^2 \\ \text{Const.}: \quad & y_k = \mathbf{w}^T\varphi(\mathbf{x}_k) + b + e_k, k = 1, \ldots, N. \end{aligned} \tag{8}$$

e_k, λ, $\varphi(.)$, $\mathbf{w}$ and b are the same as given in (1). Lagrangian equation is obtained similar to (2). And the Karush-Kuhn-Tucker conditions for optimality are obtained similar to (3). When the Lagrangian equation and optimality conditions are combined, a set of linear equations is obtained as in (9).

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \boldsymbol{\Upsilon} + \lambda^{-1}\mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{Y} \end{bmatrix} \tag{9}$$

where $\mathbf{Y} = [y_1 \quad y_2 \quad \ldots \quad y_N]^T$ and

$$\Upsilon_{kl} = K(\mathbf{x}_k, \mathbf{x}_l) \tag{10}$$

K(.,.) is the kernel function in (6). $\boldsymbol{\alpha}$ and b are the LS solution to (9) and LS-SVM regressor output is obtained as follows.

$$y(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k K(\mathbf{x}_k, \mathbf{x}) + b \tag{11}$$

2.2 UKF

UKF provides a solution to the sub-optimal estimations of EKF due to linearization. For a random variable whose first two moments (expected value and covariance) of its probability distribution are known, sigma points generated around the excpected value with same covariance can yield the real values of first three moments via a nonlinear transformation. This is called unscented transformation (UT), Wan and Van Der Merwe (2000). Let us have a random variable $\mathbf{x} \in R^d$ with expected value and covariance $\hat{\mathbf{x}}$ and $\boldsymbol{P_x}$ respectively. It is transformed via a nonlinear transformation $\mathbf{y} = g(\mathbf{x}) \in R^m$. The statistics of $\mathbf{y}$ are calculated by generating a matrix $\mathbf{X} \in R^{2d+1}$ consisting of $\mathbf{X}_i$ sigma vectors.

$$\begin{aligned} & \mathbf{X}_0 = \hat{\mathbf{x}} \\ & \mathbf{X}_i = \hat{\mathbf{x}} + (\sqrt{(d+\psi)\boldsymbol{P_x}})_i, \qquad i = 1, \ldots, d \\ & \mathbf{X}_i = \hat{\mathbf{x}} - (\sqrt{(d+\psi)\boldsymbol{P_x}})_i, \quad i = d+1, \ldots, 2d \\ & Wm_0 = \frac{\psi}{d+\psi} \\ & Wc_0 = \frac{\psi}{d+\psi} + 1 - \gamma^2 + \theta \\ & Wc_i = Wm_i = \frac{1}{2(d+\psi)}, \qquad i = 1, \ldots, 2d \end{aligned} \tag{12}$$

In (12) $\psi = \gamma^2(d + \kappa) - d$ is a scaling parameter. γ determines the prorogation of sigma points around $\hat{\mathbf{x}}$ and is usually set to a small number. κ is the second scaling parameter and is usually set to zero. θ is the a priori information about distribution of random variable $\mathbf{x}$ and its optimal value for Gaussian distribution is 2. $(\sqrt{(d+\psi)\boldsymbol{P_x}})_i$ is the i_{th} row of the matrix square root (Cholesky factorization can be employed). Process and observation covariance matrices, $\mathbf{P}_w$ and $\mathbf{P}_v$, must be involved to advance from UT to UKF as a recursive filter. UKF equations are given in Algorithm 1, Jiang et al. (2013). $\mathbf{F}$ and $\mathbf{G}$ are process and measurement functions respectively. $\boldsymbol{P_w}$ and $\boldsymbol{P_v}$ process and measurement noise covariance matrices.

Algorithm 1. UKF as a recursive filter.

% Intialization:
$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$
$\mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]$
% Sigma point generation:
$\mathbf{X}_{k-1} = [\hat{\mathbf{x}}_{k-1} \quad \hat{\mathbf{x}}_{k-1} \pm \sqrt{(d+\psi)\mathbf{P}_{k-1}}]$
% Time update:
$\mathbf{X}_{k|k-1} = \mathbf{F}(\mathbf{X}_{k-1})$
$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2d} Wm_i \mathbf{X}_{i,k|k-1}$
$\mathbf{P}_k^- = \sum_{i=0}^{2d} Wc_i (\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-)(\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-)^T + \mathbf{P}_w$
$\mathbf{y}_{k|k-1} = \mathbf{G}(\mathbf{X}_{k|k-1})$
$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2d} Wm_i \mathbf{y}_{i,k|k-1}$
% Measurement update:
$\boldsymbol{P}_{\boldsymbol{y_k y_k}} = \sum_{i=0}^{2d} Wc_i (\mathbf{y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-)(\mathbf{y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-)^T + \mathbf{P}_v$
$\boldsymbol{P}_{\boldsymbol{x_k y_k}} = \sum_{i=0}^{2d} Wc_i (\mathbf{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-)(\mathbf{y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-)^T$
$\mathbf{K} = \boldsymbol{P}_{\boldsymbol{x_k y_k}} \boldsymbol{P}_{\boldsymbol{y_k y_k}}^{-1}$
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$
$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}\boldsymbol{P}_{\boldsymbol{y_k y_k}}\mathbf{K}^T$

2.3 UKF-Based SVM

UKF-based SVM will be detailed for both classification and regression.

UKF-Based SVM Classifier When we rearrange (4) and write explicitly, we obtain the corresponding measurement function of SVM.

$$\mathbf{Y}_{SVM} = \mathbf{G}_{SVM}(\mathbf{X}_{SV}, \mathbf{Y}_{SV}, b, \boldsymbol{\alpha}, \sigma)$$

$$\begin{pmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^{n} \alpha_k y_k \\ y_1 b + y_1^2 K(\mathbf{x}_1, \mathbf{x}_1)\alpha_1 + \ldots + y_1 y_n K(\mathbf{x}_1, \mathbf{x}_n)\alpha_n + \lambda^{-1}\alpha_1 \\ y_2 b + y_2 y_1 K(\mathbf{x}_2, \mathbf{x}_1)\alpha_1 + \ldots + y_2 y_n K(\mathbf{x}_2, \mathbf{x}_n)\alpha_n + \lambda^{-1}\alpha_2 \\ \ddots \\ y_n b + y_n y_1 K(\mathbf{x}_n, \mathbf{x}_1)\alpha_1 + \ldots + y_n^2 K(\mathbf{x}_n, \mathbf{x}_n)\alpha_n + \lambda^{-1}\alpha_n \end{pmatrix} \tag{13}$$

In (13), $\mathbf{X}_{SV}$ and $\mathbf{Y}_{SV}$ are input and output sample pairs in the support vector set. They are observed variables. b, $\boldsymbol{\alpha}$ and σ parameters constitute a multi-dimensional parameter vector, $\mathbf{p}_{SVM} = [b \quad \alpha_1 \quad \alpha_2 \quad \ldots \quad \alpha_n \quad \sigma]^T \in R^{n+2}$. Output $\mathbf{Y}_{SVM}$ is also multi-dimensional so system is MIMO (Multi-input Muti-output) type. (13) presents the measurement function of SVM. Process function is needed to estimate the parameters optimally by UKF and it is identity transition matrix.

$$\begin{aligned} \mathbf{p}_{SVM,k|k-1} &= \mathbf{F}_{SVM}(\mathbf{p}_{SVM,k-1}) \\ \mathbf{F}_{SVM} &= \mathbf{I}_{n+2\times n+2} \end{aligned} \tag{14}$$

As the process and measurement noises are $\mathbf{w}$ and $\mathbf{v}$, their corresponding covariance matrices will be $\mathbf{Q}$ and $\mathbf{R}$. They have small value (e.g., 1e-6). (13) and (14) can be combined implicitly considering the noises.

$$\begin{aligned} \mathbf{p}_{SVM,k|k-1} &= \mathbf{F}_{SVM}(\mathbf{p}_{SVM,k-1}) + \mathbf{w}_k \\ \mathbf{Y}_{SVM,k} &= \mathbf{G}_{SVM}(\mathbf{X}_{SV}, \mathbf{Y}_{SV}, b_k, \boldsymbol{\alpha}_k, \sigma_k) + \mathbf{v}_k \end{aligned} \tag{15}$$

Now it turned to be a parameter estimation problem and after the substitutions in (16) are done, parameter estimation can be performed by UKF.

$$\begin{aligned} &\mathbf{F} \leftarrow \mathbf{F}_{SVM}, \quad \mathbf{G} \leftarrow \mathbf{G}_{SVM}, \quad \mathbf{x} \leftarrow \mathbf{p}_{SVM} \\ &\mathbf{y} \leftarrow \mathbf{Y}_{SVM}, \quad \mathbf{P}_w \leftarrow \mathbf{Q}, \quad \mathbf{P}_v \leftarrow \mathbf{R} \end{aligned} \tag{16}$$

UKF-Based SVM Regressor When we rearrange (9) and write explicitly, we obtain the corresponding measurement function of SVM.

$$\mathbf{Y}_{SVM} = \mathbf{G}_{SVM}(\mathbf{X}_{SV}, \mathbf{Y}_{SV}, b, \boldsymbol{\alpha}, \sigma)$$

$$\begin{pmatrix} 0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^{n} \alpha_k \\ b + K(\mathbf{x}_1, \mathbf{x}_1)\alpha_1 + \ldots + K(\mathbf{x}_1, \mathbf{x}_n)\alpha_n + \lambda^{-1}\alpha_1 \\ b + K(\mathbf{x}_2, \mathbf{x}_1)\alpha_1 + \ldots + K(\mathbf{x}_2, \mathbf{x}_n)\alpha_n + \lambda^{-1}\alpha_2 \\ \ddots \\ b + K(\mathbf{x}_n, \mathbf{x}_1)\alpha_1 + \ldots + K(\mathbf{x}_n, \mathbf{x}_n)\alpha_n + \lambda^{-1}\alpha_n \end{pmatrix} \tag{17}$$

In (17), $\mathbf{X}_{SV}$, $\mathbf{Y}_{SV}$ and $\mathbf{p}_{SVM} = [b \quad \alpha_1 \quad \alpha_2 \quad \ldots \quad \alpha_n \quad \sigma]^T \in R^{n+2}$ are the same as in classification case. Necessary process function for UKF parameter estimation is the identity transition matrix as in (14). As the process and measurement noises are $\mathbf{w}$ and $\mathbf{v}$, their corresponding covariance matrices will be $\mathbf{Q}$ and $\mathbf{R}$. They have small value (e.g., 1e-6). (17) and (14) can be combined and written implicitly as in (15). After the substitutions in (16) are done, parameter estimation can be performed by UKF.

2.4 Adaptive Update Strategy for the Support Vector Set

A strategy which is fast enough to capture the changing dynamics of data while not getting clumsy due to useless or out-of-date information is proposed. Generally in the literature, first an incremental update in the support vector set is done and then a decremental update if necessary or vice versa. Such updates are sequential. In this case, the set has an increasing or decreasing profile, Yang et al. (2010); Tang et al. (2006); Liu et al. (2009). But at some instants, it may be better to do only one of these updates. The proposed strategy provides both single update (only decremental or incremental) and sequential updates. It can also determine whether there is no need for any of these updates. Algorithm 2 presents the proposed strategy for the classification case. Regression case is different in two ways; first, the term e is determined by $e = y_k - fSVM(r_k, SV, \mathbf{p}_{SVM,k})$. And the second difference is, the term e is checked by $if \mid e \mid > eps$ where eps is a pre-set variable which determines the sensitivity of regression. SV denotes the support vector set, n_{max} is the maximum number of support vectors allowed, *incupd* and *decupd* are incremental and decremental updates of the parameter vector $\mathbf{p}_{SVM,k}$ and parameter estimation error covariance matrix $\mathbf{P}_k$ while the k_{th} sample is being processed. In incremental update of the SV set, new sample is added to the top. How $\mathbf{p}_{SVM}$ and $\mathbf{P}$ are updated incrementally/decrementally are explained as follows.

- *incupd*: Let us have the current parameter vector as

$$\mathbf{p}_{SVM} = \begin{bmatrix} b \\ \boldsymbol{\alpha} \\ \sigma \end{bmatrix}_{n+2\times 1}, \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}_{n\times 1} \tag{18}$$

When a new sample is added, corresponding α parameter (initially 0) will be added to the top of the parameters $\boldsymbol{\alpha}$.

$$\boldsymbol{\alpha}_+ = \begin{bmatrix} \alpha_{new} = 0 \\ \boldsymbol{\alpha} \end{bmatrix}_{n+1\times 1}, \mathbf{p}_{SVM_+} = \begin{bmatrix} b \\ \boldsymbol{\alpha}_+ \\ \sigma \end{bmatrix}_{n+3\times 1} \tag{19}$$

Let the current parameter estimation error covariance matrix $\mathbf{P} \in R^{n+2\times n+2}$ be as follows.

Algorithm 2. Pseudo code of proposed adaptive SV approach

```
for k=1:N do
    Get the new sample r_k = {x_k, y_k}
    e = sign(y_k) − sign(fSVM(r_k, SV, p_SVM,k))
    % If classification is not correct
    if e ≠ 0 then
        e_1 = 1
        p_temp1 = p_SVM,k
        P_temp1 = P_k
        SV_temp1 = SV
        % Scenario 1 - decremental + (if necessary) incremental update
        % first decremental update
        if #SV > 1 then
            Select a sample r from SV by FLOO cross validation
            SV_temp1 = SV − r
            [p_temp1, P_temp1] = decupd(p_SVM,k, P_k)
            Perform decremental learning by UKF
            e_1 = sign(y_k) − sign(fSVM(r_k, SV_temp1, p_temp1))
        end
        % then check whether incremental update is needed
        if e_1 ≠ 0 && #SV_temp1 < n_max then
            SV_temp1 = SV_temp1 + r_k
            [p_temp1, P_temp1] = incupd(p_temp1, P_temp1)
            Perform incremental learning by UKF
            e_1 = sign(y_k) − sign(fSVM(r_k, SV_temp1, p_temp1))
        end
        e_2 = 1
        p_temp2 = p_SVM,k
        P_temp2 = P_k
        SV_temp2 = SV
        % Scenario 2 - incremental + (if necessary) decremental update
        % first incremental update
        if #SV < n_max then
            SV_temp2 = SV + r_k
            [p_temp2, P_temp2] = incupd(p_SVM,k, P_k)
            Perform incremental learning by UKF
            e_2 = sign(y_k) − sign(fSVM(r_k, SV_temp2, p_temp2))
        end
        % then check whether decremental update is needed
        if e_2 ≠ 0 && #SV_temp2 > 1 then
            Select a sample r from SV_temp2 by FLOO cross validation
            SV_temp2 = SV_temp2 − r
            [p_temp2, P_temp2] = decupd(p_temp2, P_temp2)
            Perform decremental learning by UKF
            e_2 = sign(y_k) − sign(fSVM(r_k, SV_temp2, p_temp2))
        end
        % pick the set with the smallest error
        if abs(e_1) < abs(e_2) then
            p_SVM,k+1 = p_temp1
            P_k+1 = P_temp1
            SV = SV_temp1
        else
            p_SVM,k+1 = p_temp2
            P_k+1 = P_temp2
            SV = SV_temp2
        end
    end
end
```

$$\mathbf{P} = \begin{bmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,n+1} & P_{1,n+2} \\ P_{2,1} & P_{2,2} & \dots & P_{2,n+1} & P_{2,n+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ P_{n+1,1} & P_{n+1,2} & \dots & P_{n+1,n+1} & P_{n+1,n+2} \\ P_{n+2,1} & P_{n+2,2} & \dots & P_{n+2,n+1} & P_{n+2,n+2} \end{bmatrix} \quad (20)$$

Corresponding rows and columns to the new α parameter will be added (initially 1 on the diagonal and 0 other) and $\mathbf{P}_+ \in R^{n+3 \times n+3}$ will be obtained.

$$\mathbf{P}_+ = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & P_{1,1} & P_{1,2} & \dots & P_{1,n+1} & P_{1,n+2} \\ 0 & P_{2,1} & P_{2,2} & \dots & P_{2,n+1} & P_{2,n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & P_{n+1,1} & P_{n+1,2} & \dots & P_{n+1,n+1} & P_{n+1,n+2} \\ 0 & P_{n+2,1} & P_{n+2,2} & \dots & P_{n+2,n+1} & P_{n+2,n+2} \end{bmatrix} \quad (21)$$

- *decupd*: Removing the support vector with the smallest α value or the oldest one may not yield good results in every case. On the other hand, leave-one-out (LOO) cross validation is proven to be a standard criterion for comparing the generalization power of the statistical models. Therefore, LOO is used to determine which support vector to be removed from the set. It is aimed to choose the support vector which will provide the SVM model with the smallest approximation error after its removal. Let us assume the l_{th} vector has been determined to be removed. It will be pushed to the end of the SV set and then will be deleted. Corresponding α_l parameter will be pushed to the end of $\mathbf{p}_{SVM}$ and then will be deleted. Corresponding row and column to the parameter α_l will be pushed to the last row and column in the matrix $\mathbf{P}$ and then will be deleted.

3. SIMULATION RESULTS

Simulation results of online classification and regression are presented.

3.1 Classification

Two data sets from UCI repository, uci (2016), are used for online classification by UKF-based SVM. The first one is the Iris data set and the second one is the heart disease data set $n_{max} = 5$ is set as the maximum number of SV allowed in both experiments. Table 1 shows the online classification results. Figure 1 and 2 present change of the parameters and #SV in the online classification of Iris and heart disease data by UKF-based SVM model. Table 1 shows that due to the proposed SV set update strategy, SV set can be kept small while successful UKF training provides the SVM model with the performance of 0 misclassified sample in online classification. Due to

Table 1. Online classification results of Iris and heart disease data by UKF-based SVM model.

Data (#Samples/#Attributes)	#Error	Elapsed time (s)
Iris data set (150/4)	0	0.2041
Heart disease data set (297/14)	0	2.3411

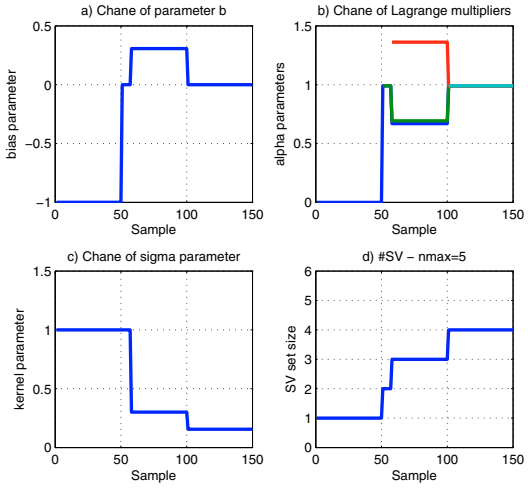

Fig. 1. Parameters a) b b) $\boldsymbol{\alpha}$ c) σ and d) #SV in online Iris data classification by UKF-based SVM model.

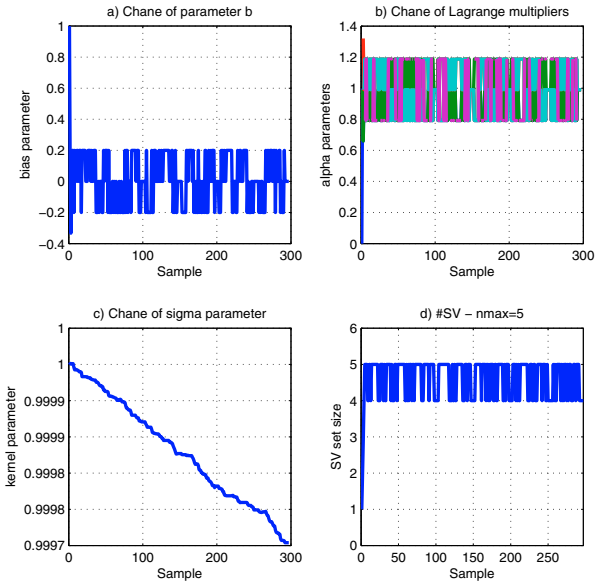

Fig. 2. Parameters a) b b) $\boldsymbol{\alpha}$ c) σ and d) #SV in online heart disease data classification by UKF-based SVM model.

the posteriori update of the UKF based SVM, the output shifts above or below the zero decision boundary, hence leading to the true classification of the given inputs. Results demonstrate that proposed online LS-SVM approach is both fast and have excellent classification performance. Especially, Iris flower classification (see Figure 1) points out the necessity to update kernel parameter simultaneously. This parameter has changed significantly over time, that the initial value would not yield such a performance.

3.2 Regression

Two real time data sets are used for online system identification by UKF-based SVM. First one is the Box-Jenkins gas furnace data widely used in the literature. It consists of 296 input-output pairs. NARX (**N**onlinear **A**uto-**R**egressive e**X**ogenous) model is constructed.

$$\hat{y}_k = f(u_k, \ldots, u_{k-n_u}, y_{k-1}, \ldots, y_{k-n_y}) \qquad (22)$$

u_k and y_k are the control input applied to the system and corresponding system output at the time index k respectively. n_u and n_y denote the past input and output samples in the NARX model. Nonlinear system function f is unknown. Second data set is collected from the real time inverted pendulum system totally 700 samples. It is a highly nonlinear and originally unstable system, Feedback (2006). Its mathematical model is derived as follows.

$$\begin{aligned}
\dot{x}_1(t) =& x_2(t) \\
\dot{x}_2(t) =& \frac{1}{(m+M)}[F - bx_2(t) - ml\dot{x}_4(t)\cos x_3(t) \\
&+ mlx_4^2(t)\sin x_3(t)] \\
\dot{x}_3(t) =& x_4(t) \\
\dot{x}_4(t) =& \frac{1}{(I+ml^2)}[mgl\sin x_3(t) - ml\dot{x}_2(t)\cos x_3(t) - dx_4(t)].
\end{aligned} \qquad (23)$$

x_1 is the cart position, x_2 is the cart velocity, x_3 is the rod angular position (output) and x_4 is the rod angular velocity. Control input and applied force are constrained as $u(t) \in [-2.5V, +2.5V]$ Volt and $F \in [-20, +20]$ Newton in real time application. In both identification experiments, $n_u = 5$ and $n_y = 5$ are set in the NARX model and $n_{max} = 5$ is set as the maximum number of SV allowed. Table 2 shows the online identification results in terms of root-mean-squarred-error (RMSE). Figure 3 and 4 present

Table 2. Online identification RMSE results of Box-Jenkins and inverted pendulum systems by UKF-based SVM model.

System (#Samples)	RMSE	Elapsed time (s)
Box-Jenkins (296)	0.0120	2.7309
Inverted pendulum (700)	0.0595	10.0870

change of the parameters and #SV in the online identification of Box-Jenkins and inverted pendulum system by UKF-based SVM model. In Fgure 5, observed outputs

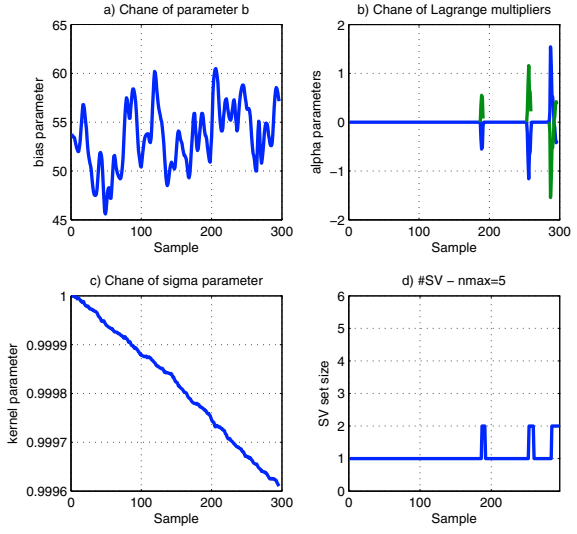

Fig. 3. Parameters a) b b) $\boldsymbol{\alpha}$ c) σ and d) #SV in online Box-Jenkins system identification by UKF-based SVM model.

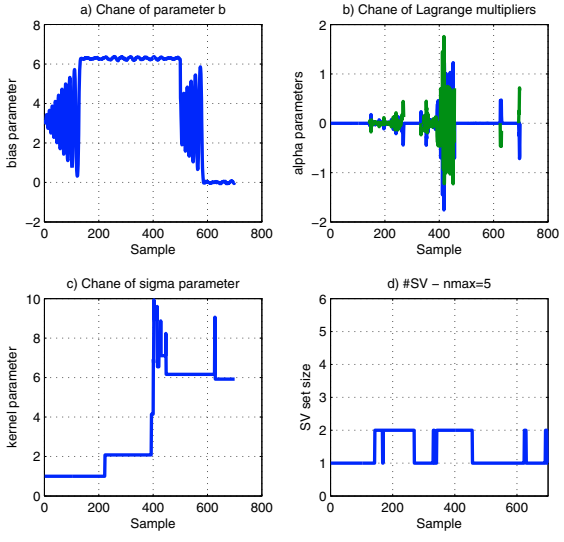

Fig. 4. Parameters a) b b) $\boldsymbol{\alpha}$ c) σ and d) #SV in online inverted pendulum system identification by UKF-based SVM model.

and onse-step-ahead predictions by the online UKF-based SVM model are compared for both systems. Figure 5

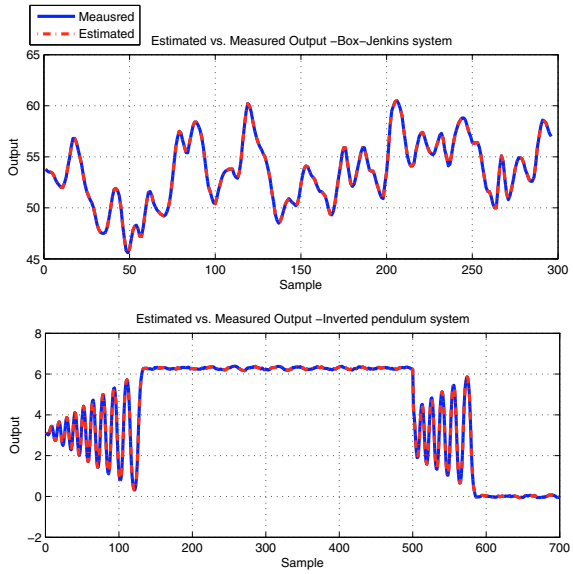

Fig. 5. Measured output and one-step-ahead predictions by the online UKF-based SVM model for Box-Jenkins and inverted pendulum systems.

demonstrates the good performance of the proposed online LS-SVM approach in regression where UKF plays an important role. Also, simultaneous kernel parameter adaptability has a significant effect in this performance, easily seen by the change of the kernel parameter over time in Figure 3 and 4. As Table 2 tells us, small identification RMSE is obtained using a small SV set which is maintained by the proposed adaptive SV set windowing approach.

4. CONLCUSION

In this paper, two novelty is introduced for online SVM classification and regression. First, all SVM parameters are trained simultaneously including the kernel parameter. Neither additional model for kernel evaluation is used nor sequential optimization is performed. Proposed approach, which is based on the LS-SVM model, has a simple framework indeed. Training is performed by UKF which has excellent performance in parameter estimation due to the unscented transformation adopted. And as the second novelty, small SV set is maintained by an intelligent variable size moving window strategy. Considering the simulations performed, as the kernel parameter varies over time, it is proved that there has been a realistic need for a simultaneous kernel parameter adaptation in the literature. And this need is full-filled by the proposed, simple and fast, online LS-SVM approach.

ACKNOWLEDGEMENTS

This paper is partly supported by Pamukkale University Scientific Research Projects Council (BAP).

REFERENCES

(2016). URL `http://archive.ics.uci.edu/ml/`.

Chapelle, O., Vapnik, V., Bousquet, O., and Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Mach. Learn.*, 46(1-3), 131–159.

Feedback (2006). *Digital Pendulum Control Experiments, 33-936s, Feedback Instruments lnc.*

Hong, W.C. (2009). Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model. *Energy Conversion and Management*, 50(1), 105 – 117.

Hong, W.C., Dong, Y., Zhang, W.Y., Chen, L.Y., and Panigrahi, B.K. (2013). Cyclic electric load forecasting by seasonal SVR with chaotic genetic algorithm. *International Journal of Electrical Power & Energy Systems*, 44(1), 604 – 614.

Jiang, Z., Liu, C., Zhang, G., Wang, Y., Huang, C., and Liang, J. (2013). GPS/INS integrated navigation based on UKF and simulated annealing optimized SVM. In *Proceedings of the 78th IEEE Vehicular Technology Conference, VTC Fall 2013, Las Vegas, NV, USA*, 1–5.

Lin, C.T., Yeh, C.M., Liang, S.F., Chung, J.F., and Kumar, N. (2006). Support-vector-based fuzzy neural network for pattern classification. *IEEE Transactions on Fuzzy Systems*, 14(1), 31–41.

Liu, Y., Hu, N., Wang, H., and Li, P. (2009). Soft chemical analyzer development using adaptive least-squares support vector regression with selective pruning and variable moving window size. *Industrial & Engineering Chemistry Research*, 48(12), 5731–5741.

Mu, T. and Nandi, A.K. (2006). Ekf based multiple parameter tuning system for a l2-svm classifier. In *2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, 229–233. IEEE.

Scholkopf, B., Simard, P., Smola, A., and Vapnik, V. (1998). Prior knowledge in support vector kernels. In *Advances in Neural Information Processing Systems 10*, 640–646. Max-Planck-Gesellschaft, MIT Press, Cambridge, MA, USA.

Suykens, J.A.K. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Process. Lett.*, 9(3), 293–300.

Suykens, J., Van Gestel, T., and De Brabanter, J. (2002). *Least Squares Support Vector Machines*. World Scientific.

Tang, H.S., Xue, S.T., Chen, R., and Sato, T. (2006). Online weighted ls-svm for hysteretic structural system identification. *Engineering Structures*, 28(12), 1728–1735.

Vapnik, V.N. (1995). *The Nature of Statistical Learning Theory*. Springer.

Wan, E.A. and Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications and Control Symposium 2000, AS-SPCC, IEEE*, 153–158.

Wang, W., Xu, Z., Lu, W., and Zhang, X. (2003). Determination of the Spread Parameter in the Gaussian Kernel for Classification and Regression. *Neurocomputing*, 55(3-4), 643–663.

Yang, X., Lu, J., and Zhang, G. (2010). Adaptive pruning algorithm for least squares support vector machine classifier. *Soft Comput.*, 14(7), 667–680.