Preprints, 10th IFAC International Symposium on
Advanced Control of Chemical Processes
Shenyang, Liaoning, China, July 25-27, 2018

Online Learning Algorithm for LSSVM Based Modeling with Time-varying Kernels

Weijian Kong*. Jinliang Ding**

* Department of Automation, Donghua University,
Shanghai, China, (e-mail: kongweijian@dhu.edu.cn)}
** State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University
Shenyang, China, (e-mail: jlding@mail. neu. edu.cn).

Abstract: Online learning based Least Squares Support Vector Machine (LSSVM) can address the modeling problems of a time-varying process, which has a few advantages such as low training time and good general. Nevertheless, many of online learning algorithms cannot adapt the kernel parameters for the time-varying characteristic, so the inferred LSSVM models are low-accuracy. An online learning algorithm with time-varying kernels is proposed to improve online training accuracy of LSSVM model. The kernel parameters are optimized along with time-varying process using updating samples data. To achieve reliable performance during online optimization, we propose a controllable metaheuristic algorithm that adopts a contracted particle swarm optimization with an elaborate chaotic operator. The proposed modeling approach is utilized in the energy efficiency prediction of the electrical smelting process, and the experimental results show that the proposed online learning algorithm can both improve the accuracy of LSSVM model and ensure low online training time.

Keywords: Time-varying process, online learning, LSSVM, online optimization, controllable metaheuristic algorithm

1. INTRODUCTION

The modeling of a time-varying process requires training model with sample data on line and in real time. As the standard Support Vector Machine (SVM) needs to solve a convex quadratic programming problem in training a model, it will take longer and longer time with the increase of samples. Furthermore, it has to retrain model once the samples change. To achieve training model on line and in real time, some online learning algorithms are suggested by means of reducing samples for training model (Alamdar, F. et al.,2016, Song, X. et al., 2017). He, Q. et al. (2011) developed an incremental learning algorithm for SVM, which can train some new coming data on line to update the existing model. Wang, H. et al. (2007) proposed a kernel cache-based method to accelerate the standard algorithm and obtained a new fast online SVR algorithm. Guo, L. et al. (2014) proposed an incremental extreme learning machine for online sequential learning problems. Maali, Y. and Jumaily, A. A. (2013) proposed a new approach to improve SVM performance in general, of which the idea is to transfer more information from the training phase to the testing phase. Agarwal, S. et al. (2008) applied kernel-based machine learning methods to online learning situations. A concept of span of support vectors was introduced into online SVMs that performs reasonably well in time-consuming. Wang, W. et al. (2008) adopted an online SVM model to the problem of air pollutant levels prediction. Manoel, C. N. et al. (2009) introduced an Online Support Vector machine for the prediction of short-term freeway traffic flow under both typical and atypical conditions.

LSSVM transforms the learning problem of SVM into solving the problem of linear equations, which reduces the computational complexity of the model training and therefore gains a faster computation speed (Langone, R. et al., 2014). Thus, LSSVM has an advantage of online modeling for a time-varying process. Zhang, W. et al. (2013) developed a heat rate forecasting method based on online LS-SVM that had possessed dynamic prediction functions.The current proposed online learnings cannot adapt the kernel parameters for the time-varying characteristic, so the inferred LSSVM models are with lower accuracy. Aiming at this problem, we will propose an online learning algorithm with time-varying kernels to improve the online training accuracy of LSSVM model. Once sample data updates, the associated kernel will be regulated optimally. Optimization algorithms include grid search, genetic algorithm, particle swarm optimization algorithm (PSO) (Long, B. et al., 2014, Guo, X. et al., 2008) for regulating the kernel parameter. Grid search is an exhaustive method, although the optimization accuracy is high but time-consuming. Genetic algorithm, PSO, and other metaheuristic algorithms have global search ability, so it is suitable for solving non-convex parameter optimization problem. However, these algorithms are random, and their performances lack reliability during online training of model. To achieve reliable optimization performance on line, we propose a controllable metaheuristic algorithm that adopts a contracted PSO with an ingeniously devised chaotic operator. To verify the proposed algorithm, we apply it in the energy efficiency prediction of electrical smelting process, and the experimental results show that the proposed online learning algorithm can both improve the accuracy of LSSVM model and ensure low online training time under the dynamic change of working condition.

2. ONLINE LEARNING ALGORITHM FOR LSSVM BASED MODELING

2.1 LSSVM based modeling

The training of the LSSVM model by the data set can be regarded as an optimization problem defined as follows:

$$\min_{w,\xi} J(w,\xi)=\frac{1}{2}\|w\|^2+\frac{1}{2}\gamma\sum_{i=1}^{l}\xi_i^2 \qquad (1)$$
$$s.t.\ \ y_i=\langle w,\phi(\mathbf{x}_i)\rangle+b+\xi_i,\quad i=1,\ldots,l$$

where γ and ξ_i represent the relative weight and regression error, respectively, and $\mathbf{x}_i$ represents input vector, and y_i represents output variable, and l is the number of samples. The function $J(\omega,\xi)$ represents the structural risk consisting of empirical risk and confidence range.

To solve this optimization problem, Lagrange function is constructed as follows:

$$L(w,\xi,b,\alpha)=\frac{1}{2}\|w\|^2+\frac{1}{2}\gamma\sum_{i=1}^{l}\xi_i^2- \sum_{i=1}^{l}\alpha_i[\langle w,\phi(\mathbf{x}_i)\rangle+b+\xi_i-y_i] \qquad (2)$$

where α_i is Lagrange multipliers. The solution of (1) can be obtained by Karush-Kuhn-Tucker (KKT) with respect to *w*, *b*, ξ_i and α_i. Eliminate ω and ξ_i, and then get linear equality:

$$\begin{bmatrix}0 & e\mathbf{1}^T\\ 1 & \mathbf{\Omega}+\frac{1}{\gamma}A_{l\times l}\end{bmatrix}\begin{bmatrix}b\\ \mathbf{\alpha}\end{bmatrix}=\begin{bmatrix}0\\ \mathbf{y}\end{bmatrix} \qquad (3)$$

where $\mathbf{y}=[y_1,\cdots,y_l]^T$, $\mathbf{\alpha}=[\alpha_1,\cdots,\alpha_l]^T$. $e\mathbf{1}$ is a column vector, each element of which is one. $A_{l\times l}$ is an identity matrix and $\mathbf{\Omega}=(\Omega_{ij})_{l\times l}$, $\Omega_{ij}=\langle\varphi(x_i)^T,\varphi(x_j)\rangle$.

Supposed $U=H^{-1}=(\mathbf{\Omega}+\gamma^{-1}I)^{-1}$, we can get

$$\mathbf{\alpha}=U\left(y-\frac{e\mathbf{1}e\mathbf{1}^TUy}{e\mathbf{1}^TUe\mathbf{1}}\right)\qquad b=\frac{e\mathbf{1}^TUy}{e\mathbf{1}^TUe\mathbf{1}} \qquad (4)$$

Apply Mercer's condition, we can get

$$\Omega_{ij}=\langle\varphi(x_i)^T,\varphi(x_j)\rangle=\kappa(x_i,x_j) \qquad (5)$$

In this paper, RBF kernel is adopted, which expresses:

$$K(x_i,x_j)=\exp\left(-\frac{\|x_i-x_j\|^2}{2\sigma^2}\right) \qquad (6)$$

Then it leads to the following LSSVM regression model:

$$y=L(\mathbf{x})=\sum_{i=1}^{l}\alpha_iK(\mathbf{x},\mathbf{x}_i)+b \qquad (7)$$

Therefore, the LSSVM based modeling need to determine the kernel parameter σ and relative weight γ by training samples set off line, and then calculate Lagrange multipliers $\mathbf{\alpha}$ and b by solving equation (4).

2.2 Online learning algorithm for LSSVM with time-varying kernels

LSSVM based online modeling is to retrain the LSSVM regression function once samples update, and thus it requires faster training on line.

Supposed new measured data is (x_k,y_k) in the *k*th sampling point, and the online LSSVM model should be adjusted as:

$$y(k)=\sum_{i=1}^{k}\alpha_i(k)\kappa_i(x,x_i)+b(k) \qquad (8)$$

where the parameters of $\{\kappa_i(x,x_i)\,|\,i=1,2,\cdots,k-1\}$ will inherit history value, respectively, and the parameter σ_k of $\kappa_k(x,x_k)$ associated with new support vector should be optimized by training updated samples.

To differentiate these kernels, define $\sigma_{i,j}$ is the parameter of $\kappa_{i,j}(x_i,x_j)$, and supposed

$$\sigma_{k,1}=\cdots=\sigma_{k,k-1}=\sigma_{1,k}=\cdots=\sigma_{k-1,k}=\sigma_k \qquad (9)$$

Owing to the Symmetry of kernel functions, it can be got as:

$$\kappa_k(x_C,x_D)=\kappa_k(x_D,x_C) \qquad (10)$$

Hence, $H(k)$ can represent as following:

$$H(k)=Q(k)+A_\gamma(k)=\begin{pmatrix}\kappa_1(x_1,x_1)+1/\gamma_1 & \cdots & \kappa_k(x_1,x_k)\\ \vdots & \ddots & \vdots\\ \kappa_k(x_1,x_t) & \cdots & \kappa_k(x_k,x_k)+1/\gamma_k\end{pmatrix} \qquad (11)$$

Change $H(k)$ into block matrix：

$$H(k)=\begin{pmatrix}H(k-1), & V(k)\\ V(k)^T, & v(k)\end{pmatrix} \qquad (12)$$

$V(k)=[\kappa_k(x_1,x_k),\cdots,\kappa_k(x_{k-1},x_k)]^T$, $v(k)=\kappa_k(x_k,x_k)+1/\gamma_k$.

According to the matrix formula, the following equation will be established when both $H(k)^{-1}$ and $H(k-1)^{-1}$ are available:

$$U(k) = H(k)^{-1} = \begin{bmatrix} H(k-1) & V(k) \\ V(k)^T & v(k) \end{bmatrix}^{-1} = \begin{bmatrix} H(k-1)^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + r_1(k) r_1(k)^T z_1(k) \tag{13}$$

$$r_1(k) = \left(V(k)^T U(k-1), -1\right)^T,\ z_1(k) = \frac{1}{v(k) - V(k)^T U(k-1) V(k)}$$

According to equation (13), both $H(k-1)^{-1}$ and $U(k-1)$ can inherit the history results got in the *(k-1)th* sampling period. As long as the parameters σ_k and γ_k are determined, both $V(k)$ and $v(k)$ can be figured out. Thus, both Lagrange multipliers vector $\boldsymbol{\alpha}$ and b can be figured out easily by means of equation (4).

2.3 Online Controllable Optimization of Kernel parameters

The metaheuristic algorithms have global search ability, and are suitable for solving non-convex parameter optimization problem. However, these algorithms lack reliability due to stochastic behavior. To achieve reliable optimization performance on line, we propose a contracted PSO tuned by an elaborate chaotic operator (gPSO for short).

We devise two types of chaotic map to initialize population and replace the random numbers of PSO, respectively. The first one is a two-dimensional cat map that is area-preserving map and suitable for initializing population for PSO in two-dimensional decision area. The cat map is formulated as

$$\begin{bmatrix} \gamma_{0,i+1} \\ \sigma_{0,i+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \gamma_{0,i} \\ \sigma_{0,i} \end{bmatrix} \bmod 1 \tag{14}$$

The other one is designed by utilizing tent map perturbed by a Logistic map that is utilized to replace the random numbers of PSO. The slight disturbance of Logistic map can eliminate the fixed points of tent map and almost do not change the amplitude of tent map. The new chaotic map model is formulated as

$$\begin{aligned} \vartheta_{i+1} &= f(\vartheta_i) = 4 * \vartheta_i (1 - \vartheta_i) \qquad 0 \le \vartheta_i \le 1 \\ \varepsilon_{i+1} &= f(\varepsilon_i) = \begin{cases} \dfrac{1}{1.001} * (2 * \varepsilon_i + 0.001 * \vartheta_{i+1}) & 0 \le \varepsilon_i \le 1/2 \\ \dfrac{1}{1.001} * (2 * (1 - \varepsilon_i) + 0.001 * \vartheta_{i+1}) & 1/2 < \varepsilon_i \le 1 \end{cases} \end{aligned} \tag{15}$$

The detailed explanation for each step of gPSO is presented as follows.

Step1. Adjust the initial value of two chaotic maps.

If the relative error of the previous training results is inacceptable, then adjust the initial value of chaotic maps to change the search trajectory of PSO until training error is acceptable, and then preserve the tuned initial value.

Step2. Initialize population by the iteration of two-dimensional cat map.

Since the iterative value is between 0 and 1, they need to be scaled up according to the ranges of σ and γ, respectively, for example, 1000 and 100. Supposed the initialized population is $\{\mathbf{px}_i | i = 1, \cdots, Pop\}$, and $\mathbf{px}_i = [\gamma_{0,i} \ \ \sigma_{0,i}]$ represents the ith particle.

Step3. Modify the guides.

Evaluate each individual of the population by calculating the root mean square error (RMSE) of entire testing set between trained model and real testing data, which is fitness value of optimization process, expressed as:

$$f_{eval}(px_i) = \sqrt{\frac{\sum_{i=1}^{lfold} \sum_{j=1}^{2} (f(px_{ij}) - y_i)^2}{l}}, i = 1, \cdots, Pop \tag{16}$$

where y_{ij} is the *jth* actual output of S_i, $f(x_{ij})$ is the *jth* predicting output of S_i.

To assess the performance of the optimization process comprehensively, we introduce the cross validation (CV) to calculate model error. The train set is split into lfold subsets equally.

The personal guide and global guide are modified by

$$\mathbf{pbest}_i \leftarrow \begin{cases} \mathbf{pbest}_i & \text{if } f_{eval}(\mathbf{px}_i) \ge f_{eval}(\mathbf{pbest}_i) \\ \mathbf{px}_i & \text{if } f_{eval}(\mathbf{px}_i) < f_{eval}(\mathbf{pbest}_i) \end{cases} \tag{17}$$

$$\mathbf{gbest} \leftarrow \begin{cases} \mathbf{gbest} & \text{if } f_{eval}(\mathbf{gbest}) \le \min\limits_{i=1}^{Pop_z + Pop_s} \{f_{eval}(\mathbf{px}_i)\} \\ \mathbf{px}_i & \text{if } f_{eval}(\mathbf{px}_i) = \min\limits_{j=1}^{Pop_z + Pop_s} \{f_{eval}(\mathbf{px}_j)\} \wedge \\ & \quad f_{eval}(\mathbf{px}_i) < f_{eval}(\mathbf{gbest}) \end{cases} \tag{18}$$

Step4. Compute the new positions of individuals. An individual's position is updated as follows:

$$\begin{cases} \mathbf{v}_i \leftarrow \chi[\mathbf{v}_i + c_1 c_3 (\mathbf{pbest}_i - \mathbf{px}_i) + \\ \qquad c_2 c_4 (\mathbf{gbest} - \mathbf{px}_i)] \\ \mathbf{px}_i \leftarrow \mathbf{v}_i + \mathbf{px}_i \end{cases} \tag{19}$$

Where, c_1 and c_2 are accelerating factors, and $c_1 + c_2 > 4$, $\chi = 2 / \left| 2 - (c_1 + c_2) - \sqrt{(c_1 + c_2)^2 - 4(c_1 + c_2)} \right|$. c_3 and c_4 generate by the tent map, and $c_3 = \varepsilon_j$, $c_4 = \varepsilon_{j+1}$.

Step5. Check for termination criteria. While running generations equal to the maximum generation, stop and output the found σ and γ. Otherwise, go to Step3 and perform the next generation search.

3. EXPERIMENTAL RESULTS

Firstly, the reliability of proposed gPSO is verified by optimizing an easy benchmark problem Rosenbrock function with gPSO, Clerc's Constriction PSO (cPSO), which has a known minimum 0. The population of both two algorithms is 30, and generations are 100. Figure 1 shows the results obtained by cPSO for 1000 runnings. The Y-axis is the target value obtained for each running. The smaller the value is, the better the algorithm is. Figure 2 shows the adjustment curve of the chaotic initial value for 50 runnings. Figure 3 shows the results obtained by cPSO for 50 runnings.

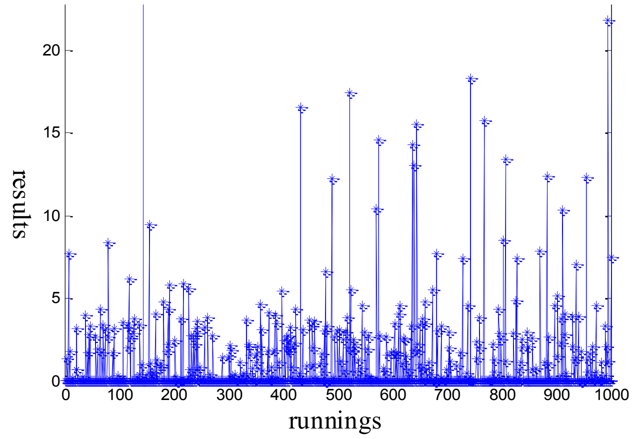

Fig. 1. Results found by cPSO in 1000 runnings

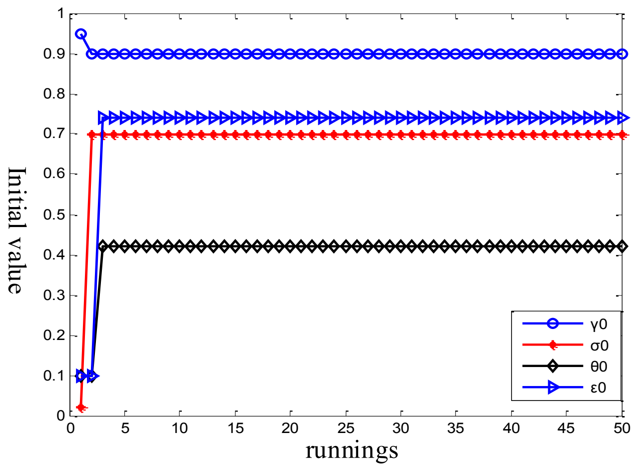

Fig. 2. Parameters adjusting of gPSO in 1000 runnings

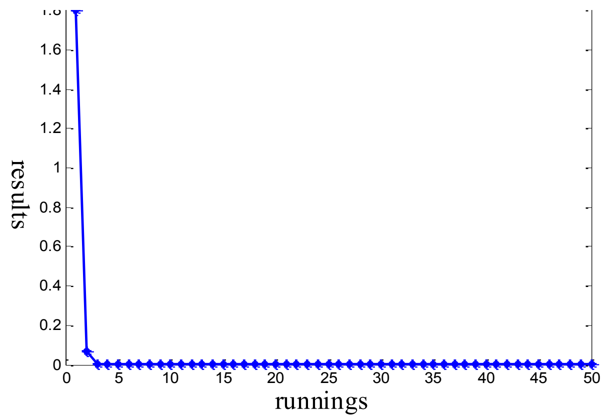

Fig. 3. Results found by gPSO in 1000 runnings

In Fig. 1, it shows the performance of cPSO is fluctuating in different runnings due to its stochastic behavior.

In Fig.2, the initial value of the two-dimensional Cat map $\gamma_{0,0}=0.95$, $\sigma_{0,0}=0.02$,and the initial value of the Tent map $\vartheta_0=0.1$, $\varepsilon_0=0.1$ in the first running of gPSO. At this time, the optimization result is 1.801, which cannot satisfy the requirement. Hence, the initial value of the chaotic map is tuned into $\gamma_{0,0}=0.9$, $\sigma^2_{0,0}=0.7$ in the second running. Then, the optimization result is improved to 0.068. The initial values of the cat map are kept, and the initial values of the tent map are tuned into $\vartheta_0=0.42$, $\varepsilon_0=0.74$ in the third running. The optimization result equal to 0.000049 and its accuracy is satisfactory. In the subsequent running of gPSO, it will not change the initial values until its performance deteriorates.

The simulation results, in Fig. 3, show that gPSO can change the search trajectory by adaptively adjusting the chaotic initial value. Therefore, once the appropriate initial value can be determined, the optimization performance of the algorithm can be ensured without fluctuation.

In view of the dynamic uncertainty existing in the electric smelting process, the proposed online LSSVM modeling with controllable optimization of kernels (gLSSVM for short) is applied to predict its energy efficiency to optimize its power supply. It verifies its effectiveness by the comparison among the standard LSSVM and the traditional online LSSVM and gLSSVM without chaotic operator and gLSSVM. The comparison results are shown in Table 1.

Table 1. Comparison results

	Training time (s)	Prediction deviation
standard LSSVM	85	3.3%
traditional online LSSVM	19	5.6%
gLSSVM without chaotic operator	32	3.4%
gLSSVM	31	3.1%

According to the experimental results, it can be found that both online LSSVM and two types of gLSSVM are shorter than standard LSSVM in training model, which thanks to the online learning algorithm to save matrix inversion calculation. The traditional online LSSVM uses a fixed kernel function so that its modeling is lower in accuracy than the two types of gLSSVM. The gLSSVM with chaotic operator perform reliably for online modeling than one without chaotic operator, so the former obtains a better prediction accuracy than the latter.

4. CONCLUSION

This paper proposes an online learning algorithm for LSSVM based modeling with controllable optimization of kernels to address the modeling problems of a time-varying process. Although the current online learnings for LSSVM can low training time, it cannot adapt the kernel parameters for the time-varying characteristic, so the accuracy of the LSSVM models are low. Aiming at this problem, we propose an online learning algorithm with time-varying kernels to improve the online training accuracy of LSSVM model. Once sample data updates, the associated kernel will be regulated optimally. Considering that genetic algorithm and PSO and other metaheuristic algorithms have global search ability, they are suitable for solving non-convex kernel-parameter optimization problem. However, these algorithms are random, and their performances lack reliability during online training. To achieve reliable optimization performance on line, we propose a controllable metaheuristic algorithm that adopts a contracted PSO with an ingeniously devised chaotic operator. To verify the proposed algorithm, we apply it in the energy efficiency prediction of an electrical smelting process, and the experimental results show that the proposed online learning algorithm can both improve the accuracy of LSSVM model and ensure low online training time under the dynamic change of working condition.

REFERENCES

Agarwal, S., Saradhi, V. V., Karnick H. (2008). Kernel-based online machine learning and support vector reduction. Neurocomputing, 71(7–9), 1230-1237.

Alamdar, F., Ghane, S., Amiri, A. (2016). On-line twin independent support vector machines. Neurocomputing, 186(19), 8-21.

He, Q., Du, C., Wang, Q., Zhuang, F., Shi, Z. (2011). A parallel incremental extreme SVM classifier. Neurocomputing, 74(16), 2532-2540.

Guo, L., Hao, J., Liu, M. (2014). An incremental extreme learning machine for online sequential learning problems. Neurocomputing, 128(27), 50-58.

Guo, X., Yang, J., Wu, C., Wang, C., Liang, Y. (2008). A novel LS-SVMs hyper-parameter selection based on particle swarm optimization. Neurocomputing, 71(16–18) , 3211-3215.

Langone, R., Agudelo, O. M., Moor, B. D., Suykensn, J. A. K. (2014). Incremental kernel spectral clustering for online learning of non-stationary data. Neurocomputing, 139(2), 246-260.

Long, B., Xian, W., Li, M., Wang, H. (2014). Improved diagnostics for the incipient faults in analog circuits using LSSVM based on PSO algorithm with Mahalanobis distance. Neurocomputing, 133(10), 237-248.

Maali, Y. and Jumaily, A. A. (2013). Self-advising support vector machine. Knowledge-Based Systems, 52, 214-222.

Neto, M. C., Jeong, Y. S., Jeong, M. K., Han, L. D. (2009). Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Systems with Applications, 36(3), 6164-6173.

Song, X., Jian, L., Song, Y. (2017). A chunk updating LS-SVMs based on block Gaussian elimination method. Applied Soft Computing, 51(2), 96-104.

Wang, H., Pi, D., Sun, Y. (2007). Online SVM regression algorithm-based adaptive inverse control. Neurocomputing, 70(4–6), 952-959.

Wang, W., Men, C., Lu, W. (2008). Online prediction model based on support vector machine. Neurocomputing, 71(4–6), 550-558.

Zhang, W., Niu, P., Li, G., Li, P. (2013). Forecasting of turbine heat rate with online least squares support vector machine based on gravitational search algorithm. Knowledge-Based Systems, 39(2), 34-44