Adaptive Least Squares Support Vector Machine Predictor for Blast Furnace Ironmaking Process

Ling JIAN, Yunquan SONG, Shuqian SHEN, Yan WANG and Haiqing YIN*

College of Science, China University of Petroleum, Qingdao, Huangdao District, Qingdao, 266580 China.

(Received on October 11, 2014; accepted on December 18, 2014)

Blast furnace system is one of the most complex industrial systems and, as such, there are still many unsolved theoretical and experimental difficulties, such as silicon prediction. For this reason, based on recursive updating algorithm, an adaptive least squares support vector machine (LS-SVM) predictor is presented for prediction task of silicon content in blast furnace (BF) hot metal. The predictor employs recursive updating algorithm to get the precise solution of the latest LS-SVM model and avoid the long process of running through the whole model. Theoretically, the computational complexity is reduced significantly from $O(n^3 m^3)$ to $O(n^3 + m^3)$. Experiments on two different BF data sets demonstrate that the proposed adaptive LS-SVM predictor is suitable for the task of predicting BF ironmaking process for its high hitting percentage and time saving.

KEY WORDS: adaptive LS-SVM predictor; silicon content prediction; recursive updating algorithm; blast furnace.

1. Introduction

Blast furnace (BF) is a highly complex nonlinear system characterized by high temperature, high pressure, strong noise and distributed parameters. During the process of blast furnace ironmaking, the solid raw materials are charged layer by layer from the top of the BF. The compressed air and some auxiliary fuels are introduced through the tuyeres just equipped above the hearth. A large number of gas-solid, solid-solid and solid-liquid chemical reactions take place in different zones of the BF. Heat energy is produced to heat up the hearth to be nearly 2 000°C. As the operation proceeds, the final product, hot metal, continuously trickles down into the hearth and is then tapped from below at a regular interval for steelmaking. The whole ironmaking period lasts 6–8 hours. Being one of the most complex industrial reactors, BF has long received broad interests both theoretically and experimentally due to its complexity and the important role of iron and steel industry on the national economy.

There are some open problems, such as the prediction of silicon content in BF hot metal, BF dynamics characterization, BF automation, etc., in the contemporary metallurgical field. Thereinto, special attention is placed to the silicon prediction in the present work. The silicon content in hot metal, as a chief indicator of the internal thermal state of BF, determines the quality of hot metal and fuel consumption in hot metal production. Thus, it needs to be controlled within acceptable bounds. To achieve this, some mathematical models\(^1\)\(^,\)\(^2\) have been constructed in the hope of providing more possibilities for controlling the BF system in stable operation state and also achieving pig iron with high quality. Since the BF process is nonlinear and multivariate essential-ly, these mathematical models seldom yield successful application in practice due to lack of fully understanding on the physical and chemical reactions in the BF process and weak real-time services. It seems very difficult to meet the increasing demand of the BF process stability and hot metal quality just depending upon mathematical models. Recently, thanks to the fast development of measurement instruments and sensor techniques, one is able to get a large amount of process data from the ironmaking BF. Since the measured data contain rich dynamic information about the BF system, the so-called data-driven modeling methods can describe the complex BF process. Today, as a BF system is concerned, data-driven modeling is being broadly investigated and exhibits great potential in describing the complex behavior of the BF process. Candidate data-driven modeling tools include state space,\(^3\) neural networks,\(^4\) genetic programming,\(^5\) partial least squares,\(^6\) fuzzy set,\(^7\) chaos,\(^8\) support vector machine (SVM),\(^9\)\(^–\)\(^11\) generalized Gaussian regularization network,\(^12\) and generalized autoregressive conditional heteroskedastic model.\(^13\) These data-driven methods are expected to be able to improve the BF model accuracy and also to shed more light on how to optimize the BF process operation in the future. Among these data-driven methods, SVM and regularization network are equivalent under some mild assumption.\(^14\) They have gained much attention because their capability to identify nonlinear system and avoid the problem of over fitting as well as ensure good generalization ability.\(^15\) However, most of these models are constructed based on fixed learning samples, which are not adequate for online use. To overcome this limitation, a recursive updating algorithm based adaptive LS-SVM pred-
indicator is presented, and further employed to predict the silicon content in BF hot metal in real-time. Experimental results indicate that the established adaptive LS-SVM predictor is a good candidate to predict the silicon content in BF hot metal with high hitting rate and less time cost. The possible application of this work is to aid the BF operators to judge the inner state of BF getting hot or chilling in time properly, which provides a guide for them to determine the direction of controlling BF in advance.

The remainder of the paper is organized as follows: Section 2 reviews the formulation of LS-SVM and presents the adaptive LS-SVM predictor, Section 3 performs the application and validation of the built adaptive LS-SVM predictor on two typical real BFs, together with the detailed analysis about the application results, Section 4 concludes this work.

2. Adaptive Least Squares Support Vector Machine Predictor

2.1. Least Squares Support Vector Machine Formulation

As an interesting variant of classical SVM, LS-SVM proposed by Suykens et al. is based on the margin-maximization principle has remarkable generalization ability. The classical SVM formulation is modified at two points: first, the inequality constraints with the slack variable are replaced by the equality constraints with error variable; second, substitute the squared loss function for ε-insensitive loss function in the objective function as it is frequently used in training of neural networks. This reformulation greatly simplifies the problem since the solution is characterized by linear equations instead of quadratic programming in the case of classical SVM. LS-SVM is formulated as follows

\[
\min_{w,b,e} \frac{1}{2} w^T w + \frac{1}{2\nu} \sum_{i=1}^{n} e_i^2 \\
\text{s.t. } y_i = w^T \phi(x_i) + b + e_i, i = 1, \ldots, n 
\]

In this formulation, \( x_1, \ldots, x_n \in \mathbb{R}^n \) are a set of input vectors, \( y_1, \ldots, y_n \in \mathbb{R} \) are the corresponding output, \( \phi(x) \) is the specified feature mapping from input space \( \mathbb{R}^n \) to feature space \( F \), \( w \) and \( b \) are the orthogonal vector and intercept of hyperplane \( w^T \phi(x) + b = 0 \) in the feature space \( F \), \( e_i \in \mathbb{R}, i = 1, \ldots, n \) are error variables and the model regularization parameter \( \nu \in (0, +\infty) \) is specified by the users.

The corresponding Lagrangian is given as

\[
L(w,b,e;\alpha) = \frac{1}{2} w^T w + \frac{1}{2\nu} \sum_{i=1}^{n} e_i^2 - \sum_{i=1}^{n} \alpha_i (w^T \phi(x_i) + b + e_i - y_i) 
\]

where \( \alpha_i \) is the \( i \)-th Lagrange multiplier. The optimal solution of Eq. (1) satisfies its Karush-Kuhn-Tucker system

\[
\begin{align*}
\frac{\partial L}{\partial w} &= 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i \phi(x_i), \\
\frac{\partial L}{\partial b} &= 0 \Rightarrow \sum_{i=1}^{n} \alpha_i = 0, \\
\frac{\partial L}{\partial e_i} &= 0 \Rightarrow e_i = \nu \alpha_i, i = 1, \ldots, n, \\
\frac{\partial L}{\partial \alpha_i} &= 0 \Rightarrow y_i = w^T \phi(x_i) + b + e_i, i = 1, \ldots, n.
\end{align*}
\]

After eliminating variables \( w \) and \( e_i \), the Karush-Kuhn-Tucker system is reformulated into the following linear equations

\[
\begin{bmatrix}
0 & I_n^T \\
I_n & K + \nu I_n
\end{bmatrix}
\begin{bmatrix}
b \\
\alpha
\end{bmatrix} =
\begin{bmatrix}
0 \\
y
\end{bmatrix}
\]

where \( K = k(x_i, x_j) = \phi(x_i)\phi(x_j) \), \( I_n \) denotes a unit matrix, \( 1_n \) is a n-dimensional vector of all ones, \( \alpha = (\alpha_1, \ldots, \alpha_n) \), and \( y = (y_1, \ldots, y_n)^T \). Once the Lagrange dual variables \( \alpha \) and bias term \( b \) are obtained, the corresponding output of any input \( x \) can subsequently be predicted by computing the decision function

\[ f(x) = w^T \phi(x) + b = \sum_{i=1}^{n} \alpha_i k(x_i, x) + b. \]

2.2. Adaptive LS-SVM Predictor Based on Recursive Updating Algorithm

To track the status of time varying system, adaptive predictor as well as incremental predictor updates the training set continuously by adding the latest sampling data timely. The schematic diagram of adaptive and incremental predictor is exhibited in Fig. 1.

This paper employs Sherman-Morrison-Woodbury formula to deduce the recursive updating method for LS-SVM regression based on which we design the adaptive LS-SVM predictor for the online prediction task. Indeed, similar work has been carried out by Mika et al. to develop efficient algorithm for kernel Fisher discriminants.

Lemma 1. Sherman-Morrison-Woodbury formula. Let \( M = \begin{bmatrix} A & u \\ u^T & c \end{bmatrix} \) be a symmetric invertible matrix given by

\[
M^{-1} = \begin{bmatrix} B & v \\ v^T & t \end{bmatrix} \]

where \( A \) is a symmetric positive-definite matrix, \( u \) a column vector and \( c \) a scalar, then the inverse of \( M \) is written as

\[
M^{-1} = \begin{bmatrix} B & v \\ v^T & t \end{bmatrix} = A^{-1} - \frac{1}{t} uu^T \]

The recursive update algorithm is used to incorporate new training sample, which is crucial for adaptive learning problem. Given a training set \( \{(x_i, y_i)\}_i \), we construct the saddle point matrix according to Eq. (4) as follows

\[
\begin{bmatrix}
0 & I_n^T \\
I_n & K + \nu I_n
\end{bmatrix} = A_{old} \]

As \( K \) is a positive semi-definite matrix and \( K + \nu I_n \) positive definite, the matrix \( A_{old} \) is invertible. We compute the
inverse matrix $A_{old}^{-1}$ in advance. Denote the right-side hand of Eq. (4) as $\begin{bmatrix} 0 \\ y_{old} \end{bmatrix}$ and the solution of Eq. (4) as $\begin{bmatrix} b_{old} \\ \alpha_{old} \end{bmatrix}$. Once a new sample $(x, y)$ is available, the linear equations Eq. (4) is updated to

$$\begin{bmatrix} 0 & 1 \\ 1 & k(x, x) \end{bmatrix} \begin{bmatrix} b_{new} \\ \alpha_{new} \end{bmatrix} = \begin{bmatrix} 0 \\ y_{new} \end{bmatrix} \quad (8)$$

where

$$k(x, x) = (k(x_1, x), \ldots, k(x_n, x))^T, \quad (9)$$

c = k(x, x) + v, \quad (10)

$$y_{new} = (y_{old}, y)^T. \quad (11)$$

Note that the saddle point matrix is partitioned into a bordered matrix as follows

$$\begin{bmatrix} 0 & 1 \\ 1 & k(x, x) \end{bmatrix} = \begin{bmatrix} \alpha_{old} \\ u^T \end{bmatrix} = \begin{bmatrix} A_{old}^{-1} u \\ u^T c \end{bmatrix} = A_{new}, \quad (12)$$

where

$$u = (1, k(x, x), \ldots, k(x_n, x))^T. \quad (13)$$

According to Lemma 1, the inverse of new saddle point matrix $A_{new}$ is computed in terms of $A_{old}$ and vector $u$, i.e.,

$$A_{new}^{-1} = \begin{bmatrix} B & v \\ v^T & t \end{bmatrix}, \quad (14)$$

where $t = 1/(c - u^T A_{old} u)$, $v = -t A_{old}^T u$, and $B = A_{old}^{-1} + t A_{old}^T u^T A_{old}$. Next, the corresponding multiplier vector $\alpha_{new}$ and intercept $b_{new}$ are deduced from

$$\begin{bmatrix} b_{new} \\ \alpha_{new} \end{bmatrix} = A_{new} \begin{bmatrix} 0 \\ y_{new} \end{bmatrix} = \begin{bmatrix} B & v \\ v^T & t \end{bmatrix} \begin{bmatrix} 0 \\ y_{new} \end{bmatrix}. \quad (15)$$

Recursive updating method is one type of direct methods. The algorithm is numerically stable provided that the saddle point matrix $A$ is well-conditioned. Generally, half of the machine precision, also called machine epsilon or unit roundoff, which gives an upper bound on the relative error due to rounding in floating point arithmetic, can be used as one standard to decide whether a matrix is well-conditioned or not. For example, IEEE double precision is $\varepsilon = 2^{-53} = 10^{-16}$. In fact, the condition numbers of saddle point matrices in this paper, ranging from 620 to 2,500, are much smaller than $10^5$. They are well-conditioned with respect to the machine precision. So theoretically the recursive algorithm is numerically stable.

The following adaptive LS-SVM predictor is presented based on recursive updating algorithm.

**Algorithm**

| Input | Training samples $(x_i, y_i)^{n-1}_{i=1}$ and pattern $x$ |
|-------|--------------------------------------------------------|
| Output| Prediction value $f(x) = \tilde{y}$ |

1: Denote the pattern and label matrices as $X_{old}$ and $y_{old}$, respectively.

2: Compute the inverse of saddle point matrix $A_{old}^{-1}$ and the corresponding solution in Eq. (4) denoted by $b_{old}$ and $\alpha_{old}$. In this way, get the LS-SVM predictor $f(\cdot) = b_{old} + \alpha_{old}^T k(X_{old}, \cdot)$.

3: While a new pattern $x$ is available, compute $u$, $c$ by Eqs. (13), (10) and the prediction result of pattern $x$ can be got by $\tilde{y} = b_{old} + \alpha_{old}^T k(X_{old}, x) = (b_{old}, \alpha_{old}^T) u$.

4: Update the inverse of saddle point matrix $A_{new}^{-1}$ according to Eq. (14). While the corresponding label $y$ is available, update $y_{new}$ according to Eq. (11), and compute the linear equations’ solution $b_{new}$, $\alpha_{new}$ by Eq. (15).

5: Set $A_{new}^{-1} = A_{new}^{-1}$, $X_{old} := (X_{old}, x^T)$, $y_{old} := (y_{old}, y)$, $b_{new} = b_{new}$, $\alpha_{new} = \alpha_{new}$.

The merit in speed is due to the efficient matrix inversion using Lemma 1. Whenever a new training sample is available, there is no need to solve the linear Eqs. (4) all over again. Direct computation of the inverse of saddle point matrix requires $O(n^3)$ operations. Suykens et al.\(^\text{16}\) proposed to use Conjugate Gradient method based iterative method for the solution of Eq. (4) which also requires $O(n^3)$ operations. Whereas adaptive LS-SVM predictor only needs $O(n^2)$ operations except for the first training process. It should be pointed out that the computation of the inverse of saddle point matrix requires $O(n^2)$ in the first training process. Once the initial inverse of saddle point matrix is given, the inverse of updated saddle point matrix, and subsequently, the solution of Eq. (4), i.e., $\alpha$ and $b$ can be computed efficiently using the adaptive LS-SVM predictor when a new training sample $(x, y)$ is available. Compared with the traditional algorithms, the total computational complex is reduced from $O(n^3m + m^3)$ to $O(n^2m + m^3)$ for $n$ training samples and $m$ test samples. The proposed algorithm is significantly efficient in the case of many test samples.

3. Application and Validation

BF is a typical multi-input and multi-output system. Since the silicon content in hot metal is very important, it is natural to take it as the BF output.\(^\text{3,4,8,9,11}\) Generally, the variables involved in the compressed air, the charged solid raw materials and some other hot metal components are thought to be the most relevant ones. They are selected as input variables. Here, to make the proposed adaptive LS-SVM predictor more convincing, two groups of experimental data sets about these related variables are collected from a medium-sized BF with the inner volume of about 2,500 m³ and a pint-sized BF with the inner volume of about 750 m³, labeled as BF(a) and (b), respectively. For the BF(a), totally 20 candidate variables listed in Table 1, including blast volume, blast temperature etc. are selected as input variables. While for the BF(b), there are only 6 relevant variables collected due to the weak measurement condition in this BF, which are selected as input variables. The 6 variables are also exhibited in Table 1 with marked * sign. It should be pointed out that too many input variables may increase the complexity of model while too few inputs will reduce the accuracy of model. A tradeoff should be taken between the model complexity and accuracy when we select the inputs. So there is much work worth investigating in the future, like input variables selection, to improve the performance of the model. The iron ore used for BF(a) is the Baiyunebo mineral.
3.1. Experimental Data and Preprocessing

There are 700 observed data points in all. The sampling interval is about 1.5 h for the BF(a) and about 2 h for the BF(b). In the field of machine learning, the number of training samples should be equal to or greater than the number of the learning parameters that specify a predictive model. On the other hand, evaluating results on large test set is more convincing than those on small test set. According to the demonstrated two variables, Due to the fact that the variable with a large magnitude has a stronger effect on the model parameters than the one with a small magnitude, it’s ill-considered to take the data of these variables as the model inputs for training. Thus, the following relationship is used to handle the input data to make them under the same magnitude

\[ z_i' = \frac{x_i' - m(x_i')}{\sigma(x_i')} \]

where \( z_i' \), \( m(x_i') \) and \( \sigma(x_i') \) stand for the actual \( i \)-th model input, the mean and the standard deviation of the variable \( x_i' \) respectively.

3.2. Predictor Implementation and Performance Evaluation

To construct the LS-SVM predictor, it is necessary to determine the type of kernel function, the related kernel parameters, and the parameter \( \nu \) in Eq. (4) in advance. In this current work, the Gaussian radial basis function is specified as the kernel function for its extensive applications and good performance in the BF modeling, which implies that it is an arduous task to predict hot metal silicon through silicon historical information. Multivariate model is preferred to deal with this issue. Figure 2 demonstrates the time series of the silicon content in hot metal and the blast volume measured from these two BFs. Clearly, there is a great difference in the magnitude of the demonstrated two variables. Due to the fact that the variable with a large magnitude has a stronger effect on the model parameters than the one with a small magnitude, it’s ill-considered to take the data of these variables as the model inputs for training. Thus, the following relationship is used to handle the input data to make them under the same magnitude

\[ z_i' = \frac{x_i' - m(x_i')}{\sigma(x_i')} \]

where \( z_i' \), \( m(x_i') \) and \( \sigma(x_i') \) stand for the actual \( i \)-th model input, the mean and the standard deviation of the variable \( x_i' \) respectively.

Table 2. Statistical properties of silicon sequences of the selected BFs in terms of divided training and test sets.

| BF Set | Max  | Min  | Mean  | Std  | PSD | CV  | Skew | Kurtosis |
|--------|------|------|-------|------|-----|-----|------|----------|
| Train  | 1.080| 0.280| 0.567 | 0.161| 0.089 | 0.252| 1.609| 3.328    |
| Test   | 1.110| 0.130| 0.590 | 0.185| 0.185 | 0.314| 0.512| 2.780    |
| Train  | 1.060| 0.240| 0.455 | 0.111| 0.089 | 0.251| 1.609| 8.132    |
| Test   | 0.960| 0.220| 0.441 | 0.111| 0.089 | 0.251| 1.298| 5.413    |

Std: standard deviation; PSD: pseudo standard deviation; CV: coefficient of variation.

in Baotou, which contains materials such as rare earth elements, niobium and other metals. The complex ingredients of the iron ore make the ironmaking process extremely unstable. The prediction of silicon content in BF hot metal also becomes more difficult. The latest silicon content in BF hot metal through silicon historical information. It shows the extent of variability in relation to the mean and standard deviation of random variable \( x \). Skew tests the skewness of the data with \( Skew >, = \) and \( < 0 \) being right or positive skewness, symmetry and left or negative skewness. Kurtosis tests the relative peakness or flatness of the data with \( Kurtosis >, = \) and \( < 0 \) being peaked distribution, normal distribution and flat distribution. Based on the statistics of these series, it is concluded that the studied data sets have the characteristics of fluctuating tempestuously, heavy tailed distribution, right skewness distribution and peaked distribution. These indicate that the studied data is not normally distributed and cannot act as a good sample for model training, which implies that it is an arduous task to predict hot metal silicon through silicon historical information. Multivariate model is preferred to deal with this issue. Figure 2 demonstrates the time series of the silicon content in hot metal and the blast volume measured from these two BFs. Clearly, there is a great difference in the magnitude of the demonstrated two variables. Due to the fact that the variable with a large magnitude has a stronger effect on the model parameters than the one with a small magnitude, it’s ill-considered to take the data of these variables as the model inputs for training. Thus, the following relationship is used to handle the input data to make them under the same magnitude

\[ z_i' = \frac{x_i' - m(x_i')}{\sigma(x_i')} \]

where \( z_i' \), \( m(x_i') \) and \( \sigma(x_i') \) stand for the actual \( i \)-th model input, the mean and the standard deviation of the variable \( x_i' \) respectively.
used as the training set while the residual one fold is used as the validation set. The optimal parameter pair corresponds to those whose mean training performance, i.e., the MSE (mean squared error), is the lowest. In this work, the state-of-the-art LS-SVM software LS-SVMlab 1.5 package is employed to search the optimal parameter pair. The optimal parameter pairs of LS-SVM for BF(a) and BF(b) are (0.083, 19.264) and (0.191, 57.241) respectively. In this paper, we compare the present method with two LS-SVM methods, i.e., fixed LS-SVM predictor with fixed training set and incremental LS-SVM predictor which reTRAIN the LS-SVM all over again by adding one training sample to the training set once a new sample is available, and the online support vector regression (AOSVR) method. The model parameter ε of AOSVR is set as 0.1, and for fair comparison the parameter pair, (v, σ²) is set as (0.2, 20) for BF(a) and (0.1, 7) for BF(b) by five-fold cross-validation. To evaluate the performance of the adaptive LS-SVM predictor quantitatively, the following frequently used criteria in metallurgical field as well as the time cost are listed in Table 3.

- the percentage of hitting the target

\[ H = \frac{\sum_{i=1}^{m} H_i}{100\%}, \quad \text{(18)} \]

here \( m \) is the size of test set, and \( H_i \) is defined as

\[ H_i = \begin{cases} 1 & \text{if } |\hat{y}_i - y_i| \leq 0.1, \\ 0 & \text{otherwise.} \end{cases} \quad \text{(19)} \]

- the percentage of prediction success

\[ S = \frac{\sum_{i=1}^{m} \text{max}(H_i, T_i)}{100\%}, \quad \text{............. (20)} \]

here \( T_i \) is defined as

\[ T_i = \begin{cases} 1 & (\hat{y}_i - y_{i-1})(y_i - y_{i-1}) \geq 0, \\ 0 & \text{otherwise.} \end{cases} \quad \text{(21)} \]

We say that the prediction is successful if \( \text{max}(H_i, T_i) \) equal to one.

- the root mean square error

\[ \text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2}. \quad \text{............. (22)} \]

- the correlation coefficient between the actual values and the predicted values

\[ r = \frac{\sum_{i=1}^{m} (\hat{y}_i - \bar{y})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m} (\hat{y}_i - \bar{y})^2 \sum_{i=1}^{m} (y_i - \bar{y})^2}}. \quad \text{............. (23)} \]

\[ r = \frac{\sum_{i=1}^{m} (\hat{y}_i - \bar{y})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m} (\hat{y}_i - \bar{y})^2 \sum_{i=1}^{m} (y_i - \bar{y})^2}}. \quad \text{............. (23)} \]

Table 3. Results comparison of the four methods on test set.

| BF | Methods          | H (%) | S (%) | R    | RMSE | CPU (sec) |
|----|------------------|-------|-------|------|------|-----------|
| (a) Adaptive predictor | 81.4 | 90.8  | 0.7312 | 0.1267 | 7.7500 |
| Incremental predictor  | 81.4 | 90.8  | 0.7312 | 0.1267 | 397   |
| Fixed predictor       | 70.8 | 84.2  | 0.5653 | 0.1545 | 0.1563 |
| AOSVR predictor       | 71.0 | 86.4  | 0.6124 | 0.1495 | 171.3125|
| (b) Adaptive predictor | 84.4 | 92.6  | 0.4244 | 0.1033 | 12.5625 |
| Incremental predictor  | 84.4 | 92.6  | 0.4244 | 0.1033 | 361.0156|
| Fixed predictor       | 79.6 | 89.4  | 0.2564 | 0.1204 | 0.0938 |
| AOSVR predictor       | 86.2 | 92.2  | 0.1241 | 0.1257 | 472.0313 |

* The MATLAB code of AOSVR supplied by F. Parrella can be downloaded from http://www.onlinesvr.altervista.org.
\[
R = \frac{1}{m} \sum_{i=1}^{m} \frac{(\hat{y}_i - m(y))(y_i - m(y))}{\sigma(y)\sigma(y)}, \quad \text{(23)}
\]

where \( m(y) \) is the mean of the predicted values, \( m(y) \) is the mean of the actual values, \( \sigma(y) \) and \( \sigma(y) \) are the standard deviation of predicted and actual values respectively.

All simulations are implemented in MATLAB 7.8 environment on a PC with Intel Core2Duo processor 2.1 GHz and 2 GB RAM. When an observed sample is available the adaptive LS-SVM predicator simply updates the previous solution to the new one to adapt to the change of BF iron-making system, so it is much faster than incremental LS-SVM predicator. In terms of other quality index, adaptive LS-SVM predicator and incremental LS-SVM predicator, which employs iteration method, \textit{i.e.}, Conjugate Gradient method to solve the saddle point system Eq. (8), are always the same. This indicates that the proposed adaptive LS-SVM predicator is numerically stable and robust from the perspective of experiment. The adaptive LS-SVM always outperforms AOSVR predictor except the percentage of hitting the target, \textit{i.e.}, H(\%) on BF(b). Furthermore, the performance of adaptive LS-SVM predicator is much better than fixed LS-SVM predicator on all criteria except the time cost.

4. Conclusions

The work presents an effective adaptive LS-SVM predicator with high hitting percentage and time saving for the task of BF silicon content prediction. Experiments demonstrate that adaptive LS-SVM predicator can be used for real-time prediction application. It is worth pointing out that as more and more observed data are available, the scale of training set will increase accordingly, and so is the training time. To overcome the above problem, we may use some decremental techniques such as sliding window etc., which will be our future investigation.

Acknowledgements

The authors would like to thank Dr. Zhaozhu Li for helpful discussions about this paper. This work was supported by National Natural Science Foundation of China under Grant No. 11326203, 61201455, 61403419, Natural Science Foundation of Shandong Province under Grant No. ZR2013FQ034 and Fundamental Research Funds for the Central Universities under Grant No. 13CX02008A and 14CX02017A.

REFERENCES

1) H. Nogami, M. S. Chu and J. Yaji: Comput. Chem. Eng., 29 (2005), 2438.
2) V. R. Radhakrishnan and K. M. Ram: J. Process. Contr., 11 (2011), 565.
3) J. H. Cai, J. S. Zeng and S. H. Luo: ISIJ Int., 52 (2012), 2194.
4) H. Saxén and F. Pettersson: ISIJ Int., 47 (2007), 1732.
5) F. Pettersson, H. Saxén and K. Deb: Mater. Manuf. Process., 24 (2009), 343.
6) T. Bhattacharya: ISIJ Int., 45 (2005), 1943.
7) R. D. Martín, F. Obeso, J. Mochon, R. Barea and J. Jiménez: Ironmaking Steelmaking, 34 (2007), 241.
8) C. H. Gao, J. M. Chen, J. S. Zeng, X. Y. Liu and Y. X. Sun: AIChE J., 55 (2009), 947.
9) L. Jian, C. H. Gao, L. Li and J. S. Zeng: ISIJ Int., 10 (2008), 1659.
10) X. L. Tang, L. Zhang and C. J. Jian: Expert. Syst. Appl., 48 (2009), 11853.
11) L. Jian, C. H. Gao and Z. Q. Xia: Steel. Res. Int., 82 (2011), 169.
12) M. Miyano, S. Kimoto, H. Shibuta, K. Nakashima, Y. Ikenaga and K. Aihara: Physica D, 135 (2000), 305.
13) J. S. Zeng, C. H. Gao, X. G. Liu, K. P. Yang and S. H. Luo: Asian J. Control., 10 (2008), 632.
14) F. Girosi: Neural Comput., 7 (1998), 1455.
15) V. N. Hien and P. Fatih: IEEE Trans. Pattern Anal. Mach Intell., 25 (2013), 970.
16) J. A. K. Suykens and J. Vandewalle: Neural Process. Lett., 9 (1999), 293.
17) J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor and J. Vandewalle: Least Squares Support Vector Machines, World Scientific, Singapore, (2002), 136.
18) S. Mika, A. J. Smola and B. Schölkopf: Proc. AISTATS, Morgan Kaufmann, San Francisco, (2001), 98.
19) J. S. Zeng, X. G. Liu, C. H. Gao, S. H. Luo and L. Jian: ISIJ Int., 48 (2008), 1734.
20) Z. H. Tang and Y. Yang: ISIJ Int., 54 (2014), 1836.
21) C. H. Gao, L. Jian and S. H. Luo: IEEE Trans. Ind. Electron., 59 (2012), 1134.
22) J. Ma, J. Theiler and S. Perkins: Neural Comput., 15 (2003), 2683.