Expert Systems with Applications 36 (2009) 11853–11857

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Prediction of silicon content in hot metal using support vector regression based on chaos particle swarm optimization

Xianlun Tang *, Ling Zhuang, Changjiang Jiang

Key Laboratory of Network Control & Intelligent Instrument (Chongqing University of Posts and Telecommunications), Ministry of Education, Chongqing, PR China

ARTICLE INFO

Keywords:
Support vector regression
Particle swarm optimization
Chaos
Silicon content in hot metal
Prediction

ABSTRACT

The prediction of silicon content in hot metal has been a major study subject as one of the most important means for the monitoring state in ferrous metallurgy industry. A prediction model of silicon content is established based on the support vector regression (SVR) whose optimal parameters are selected by chaos particle swarm optimization. The data of the model are collected from No. 3 BF in Panzhihua Iron and Steel Group Co. of China. The results show that the proposed prediction model has better prediction results than neural network trained by chaos particle swarm optimization and least squares support vector regression, the percentage of samples whose absolute prediction errors are less than 0.03 when predicting silicon content by the proposed model is higher than 90%, it indicates that the prediction precision can meet the requirement of practical production.

1. Introduction

Blast furnace is the governing process of ironmaking in the ferrous metallurgy industry. The complexity of the heat and mass transfer process coupling with a large number of gas–solid, solid–solid and solid–liquid reactions, combustion processes and interphase mass transfer makes the modeling of blast furnace an extremely difficult problem. The prediction of key operational parameters, such as silicon content in hot metal, has been a major research issue. Silicon content is one of the most important indices to represent the thermal state of the blast furnace, its accurate and advance prediction can greatly help in stabilizing the blast furnace operations (Zhou, 2007).

Numerous studies have focused on the accurate prediction of silicon content in hot metal by using statistical approaches and artificial intelligence approaches. However, the relationship among the operational parameters is complex and highly nonlinear, and the data collected from blast furnace are also quite noisy. Neural network modeling has been shown to reproduce nonlinear data very well while it was not very well explainable and often had the problem of overfitting leading to poor performance (Chen, 2001; Zhang & Jin, 2007). To obtain further useful information, many predictive systems have been developed by integrating the mathematical model with expert system. Successful expert systems and predictive systems have been gradually adopted by many blast furnaces in different countries for on-line or off-line processes; however, operational results showed that some of these models did not perform well (Liu & Liu, 2003). Generally, systems based on empirical knowledge have been developed mainly by programming based on rules, while they are characterized as serious limitation to inference method based on the empirical connection between the observable findings and the metallurgy criterion (Liu & Liu, 2003).

Growing efforts are made to explore innovative methods to increase prediction performance. Since the creation of the theory of support vector machines (SVM) (Vapnik, 1998, 1999), rapid development of SVM in statistical learning theory encourages researchers to actively focus on applying SVM to various research fields such as document classifications and pattern recognitions. SVM possesses a great potential and has shown a superior performance as is appeared in many previous researches. This is largely due to the structural risk minimization (SRM) principle in SVM which has greater generalization ability and is superior to the empirical risk minimization (ERM) principle as adopted in the neural networks. In SVM, the results guarantee global minima whereas ERM can only locate local minima. For example, in the training process in neural networks, the results give out any number of local minima that are not promised to include global minima. Furthermore, SVM is adaptive to complex systems and is robust in dealing with corrupted data. This feature offers SVM a greater generalization ability which is the bottleneck of its predecessor, the neural network approach (Deng & Tian, 2004).

On the other hand, applications of support vector regression (SVR) extended by SVM (Musicant & Alexander, 2004; Na & Upadhyaya, 2006; Quan, 2004; Shevade & Keerthi, 2000), such as forecasting of financial market, estimation of power consumption and prediction of highway traffic flow, have been also under development and have shown many breakthroughs and excellent performances (Ding &

* Corresponding author. Tel.: +86 23 62471349; fax: +86 21 62461535.
E-mail address: tangxlun@hotmail.com (X. Tang).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.04.015

Song, 2008; Van & Suykens, 2001; Shin, Lee, & Kim, 2005; Scholkopf, Smola, & Wilimson, 2000). The time-varying properties of SVR applications resemble the time dependency of silicon content prediction, combined with many successful results of SVR predictions encourage our research in using SVR for silicon content modeling.

Because the quality of SVR models depends on a proper setting of SVR meta-parameters, the main issue for practitioners trying to apply SVR is how to set these parameter values to ensure good generalization performance for a given dataset. Whereas existing sources on SVM regression (Deng & Tian, 2004; Kwok, 2001; Smola & Schölkopf, 1998) give some recommendations on appropriate setting of SVM parameters, there are clearly no consensus and contradictory opinions. Hence, resampling remains the method of choice for many applications. Unfortunately, using resampling for tuning several SVR parameters is very expensive in terms of computational costs and data requirements.

In this paper, we propose a new particle swarm optimization algorithm based on chaos searching (CPSO) to search the optimal parameters of SVR, then a prediction model of silicon content in hot metal (CPSO-SVR) was established by using SVR based on CPSO. The results show that this model is applicable to silicon content prediction and outperforms some previous methods.

The rest of this paper is organized as follows. Section 2 provides a brief analysis of support vector regression. In Section 3, particle swarm optimization, chaos particle swarm optimization and prediction model based on CPSO-SVR are presented. Experiment and analysis are discussed in Section 4. Finally, Section 5 provides a summary and a conclusion.

2. Support vector regression

Considering a set of training data $\{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$, where each $x_i \subset R^n$ denotes the input space of the sample and has a corresponding target value $y_i \subset R$ for $i = 1, \ldots, l$, where l corresponds to the size of the training data. The idea of the regression problem is to determine a function that can approximate future values accurately.

The generic SVR estimating function takes the form:

$$f(x) = (w \cdot \Phi(x)) + b \tag{1}$$

where $w \subset R^n, b \subset R$ and Φ denote a non-linear transformation from R^n to high-dimensional space. Our goal is to find the values of w and b such that the values of x can be determined by minimizing the regression risk:

$$R_{reg}(f) = C\sum_{i=0}^{\ell} \Gamma(f(xi) - yi) + \frac{1}{2}\|w\|^2 \tag{2}$$

where $\Gamma(\cdot)$ is a cost function, C is a constant that determines penalties to estimation errors and vector w can be written in terms of data points as:

$$w = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*)\Phi(x_i) \tag{3}$$

By substituting Eq. (3) into Eq. (1), the generic equation can be rewritten as:

$$f(x) = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*)(\Phi(x_i) \cdot \Phi(x)) + b = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*)k(x_i, x) + b \tag{4}$$

In Eq. (4) the dot product can be replaced with function $k(x_i, x)$, which is known as the kernel function. Kernel functions enable dot product to be performed in high-dimensional feature space using low-dimensional space data input without knowing the transformation Φ. All kernel functions must satisfy Mercer's condition that corresponds to the inner product of some feature space.

The ε-insensitive loss function is the most widely used cost function. The function is in the form:

$$\Gamma(f(x) - y) = \begin{cases} |f(x) - y| - \varepsilon & \text{for } |f(x) - y| \geqslant \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

By solving the quadratic optimization problem in (6), the regression risk in Eq. (2) and the ε-insensitive loss function (5) can be minimized:

$$\frac{1}{2}\sum_{i,j=1}^{\ell} (\alpha_i^* - \alpha_i)\left(\alpha_j^* - \alpha_j\right)k(x_i, x_j) - \sum_{i=1}^{\ell} \alpha_i^*(y_i - \varepsilon) - \alpha_i(y_i + \varepsilon)$$

subjected to

$$\sum_{i=1}^{\ell} \alpha_i - \alpha_i^* = 0, \quad \alpha_i, \alpha_i^* \in [0, C] \tag{6}$$

The Lagrange multipliers, α_i and α_i^*, represent solutions to the above quadratic problem that acts as forces pushing predictions towards target value y_i. Only the non-zero values of the Lagrange multipliers in Eq. (6) are useful in forecasting the regression line and are known as support vectors. For all points inside the ε-tube, the Lagrange multipliers equal to zero do not contribute to the regression function. Only if the requirement $|f(x) - y| \geqslant \varepsilon$ is fulfilled, Lagrange multipliers may be non-zero values and used as support vectors.

Now, we have solved the value of w in terms of the Lagrange multipliers. For the variable b, it can be computed by applying Karush-Kuhn-Tucker (KKT) conditions which, in this case, implies that the product of the Lagrange multipliers and constrains has to equal zero:

$$\alpha_i(\varepsilon + \zeta_i - y_i + (w, x_i) + b) = 0$$
$$\alpha_i^*(\varepsilon + \zeta_i^* + y_i - (w, x_i) - b) = 0$$

and

$$(C - \alpha_i)\zeta_i = 0$$
$$(C - \alpha_i^*)\zeta_i^* = 0$$

where ζ_i and ζ_i^* are slack variables used to measure errors outside the ε-tube. Since $\alpha_i, \alpha_i^* = 0$ and $\zeta_i^* = 0$ for $\alpha_i^* \in (0, C)$, b can be computed as follows:

$$b = y_i - (w, x_i) - \varepsilon \quad \text{for } \alpha_i \in (0, C)$$
$$b = y_i - (w, x_i) + \varepsilon \quad \text{for } \alpha_i^* \in (0, C)$$

Putting it all together, we can use SVM and SVR without knowing the transformation.

3. The prediction model based on CPSO-SVR

3.1. Particle swarm optimization

PSO is initialized with a swarm including N random particles. Each particle is treated as a point in a D-dimensional space. The ith particle is represented as $x_i = (x_{i1}, x_{i2}, \cdots, x_{iD}), x_{ij}$ is set in the range $[a_j, b_j]$. The best previous position of the ith particle is represented as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$. The best particle among all the particles in the population is represented by $P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$. The velocity of particle i is represented as $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. After finding the aforementioned two best values, the particle updates its velocity and position according to the following equations:

$$v_{id} = \omega v_{id} + c_1 r_1(p_{id} - x_{id}) + c_2 r_2(p_{gd} - x_{id}) \tag{7}$$

$$x_{id} = x_{id} + v_{id} \tag{8}$$

where ω is the inertia weight, c_1 and $c_2$are two positive constants called learning factors, and r_1 and $r_2$are random numbers in the range of $[0, 1]$ (Eberhart & Shi, 2000; Kennedy & Eberhart, 1995).