Chinese Journal of Chemical Engineering 22 (2014) 774–781

Contents lists available at ScienceDirect

Chinese Journal of Chemical Engineering

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

Process Control

Adaptive Nonlinear Model Predictive Control Using an On-line Support Vector Regression Updating Strategy ☆

Ping Wang [1,2], Chaohe Yang [1], Xuemin Tian [2,*], Dexian Huang [3]

[1] State Key Laboratory of Heavy Oil Processing, China University of Petroleum, Qingdao 266580, China
[2] College of Information and Control Engineering, China University of Petroleum, Qingdao 266580, China
[3] Department of Automation, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 24 June 2013
Received in revised form 14 October 2013
Accepted 16 November 2013
Available online 20 June 2014

Keywords:
Adaptive control
Support vector regression
Updating strategy
Model predictive control

ABSTRACT

The performance of data-driven models relies heavily on the amount and quality of training samples, so it might deteriorate significantly in the regions where samples are scarce. The objective of this paper is to develop an on-line SVR model updating strategy to track the change in the process characteristics efficiently with affordable computational burden. This is achieved by adding a new sample that violates the Karush–Kuhn–Tucker conditions of the existing SVR model and by deleting the old sample that has the maximum distance with respect to the newly added sample in feature space. The benefits offered by such an updating strategy are exploited to develop an adaptive model-based control scheme, where model updating and control task perform alternately. The effectiveness of the adaptive controller is demonstrated by simulation study on a continuous stirred tank reactor. The results reveal that the adaptive MPC scheme outperforms its non-adaptive counterpart for large-magnitude set point changes and variations in process parameters.

1. Introduction

Model predictive control (MPC) has been widely accepted as an advanced control strategy in process industry owing to its ability to handle complex control problems with constraints [1,2]. MPC uses a process model to predict future process behavior and future control actions are computed by minimizing a pre-specified cost function, so the effectiveness of MPC relies heavily on the availability of a reasonably accurate process model. Until recently, industrial applications of MPC have mainly based on linear models due to their inherent simplicity from conceptual and implementation points of view [1]. However, as many chemical plants exhibit highly nonlinear behavior when operated over a wide range, linear MPC often results in poor control performance, which motivates its extension to nonlinear MPC, with a more accurate nonlinear model used for prediction and optimization [3].

Among various nonlinear modeling methods, the support vector regression (SVR) method has been widely applied in data-driven modeling [4], since it not only shares many of its features with neural networks but also possesses some additional desirable characteristics. The advantage of SVR is that [4], for a given modeling problem with a finite set of samples, it can automatically derive the optimal network structure with respect to generalization error. Furthermore, current experience shows that SVRs work as well as, and in some cases, better than classical statistical approaches on noisy or imprecise data. Because of these advantages, SVR has been found increasing applications in chemical processes [5], especially when training data are insufficient or the process has strong nonlinearity [6–8]. Nevertheless, a basic limitation of all data-driven models is their inability to extrapolate accurately once the information is outside the range of data used to generate the model [9]. As the training samples available only describes a period of process historical behavior and might not represent complete characteristics of true dynamics, the performance of the model may deteriorate substantially with time as a consequence of changes in the dynamics of process [10–14]. Although the model can be retrained from scratch when the training set is modified, it is cumbersome and computationally inefficient. In this case, the accurate online SVR (AOSVR) technique [15] seems to be a better alternative to SVR because it uses an incremental algorithm, which updates SVR model efficiently and accurately when a new sample is added to the training set without retraining from scratch. Several model-based control schemes based on AOSVR have been developed in recent years. Iplikci has proposed an adaptive generalized predictive control (GPC) method by combining SVR-based GPC approach with AOSVR [16]. An adaptive inverse control algorithm, where SVR model is used to construct the inverse model of

☆ Supported by the National Basic Research Program of China (2012CB720500), Postdoctoral Science Foundation of China (2013M541964) and Fundamental Research Funds for the Central Universities (13CX05021A).
* Corresponding author.
E-mail address: tianxm@upc.edu.cn (X. Tian).

http://dx.doi.org/10.1016/j.cjche.2014.05.004
1004-9541/

the process to be controlled online, has been studied [17]. More recently, AOSVR is employed to capture the abrupt and incremental faults in the framework of SVR model based fault tolerance predictive control scheme [18]. However, in these applications the AOSVR technique is used without taking into account the differences among the newly incoming samples and lacking mechanisms to prune redundant samples efficiently [10]. This may result in unreliable predictions and heavy computation burden as the number of training data increases, limiting the applications of recursive SVR algorithm to a long-term online modeling and control task.

In our recent work, an efficient on-line model updating strategy based on AOSVR has been developed and shown good performance for predicting the melt index of an industrial polypropylene process [19]. This strategy allows us to improve the estimation performance with affordable computation burden by updating the existing SVR model based on the novelty of new samples that arrive sequentially. In this paper, the benefits offered by such an updating strategy are further exploited to design an adaptive MPC controller. Several enhancements are also developed to make the model updating strategy capture the current behavior of process more effectively. Specifically, only those samples that violate the Karush–Kuhn–Tucker (KKT) conditions of existing SVR model are added to improve the estimation accuracy for new operating region where samples may not be scarce or nonexistent. As far as the accumulation of obsolete data is concerned, the old training sample with the maximum distance to the newly added sample in feature space will be recognized as the redundant sample, which will be removed from training database subsequently to enhance the estimation accuracy for the current behavior of process. The effectiveness of the adaptive MPC controller is illustrated by a simulation study on a benchmark continuous stirred tank reactor (CSTR) [20]. The adaptive MPC controller is able to achieve a smooth transition for large magnitude set point changes and maintain the process at an unstable operating point in the presence of unmeasured disturbances and random noise.

2. On-line Support Vector Regression

Given a training set $\boldsymbol{T}=\{(\boldsymbol{x}_i, y_i), i=1,2,\ldots,N\}$, where N is the total number of training samples, $\boldsymbol{x}_i\in R^n$ is the n-dimension input vector, and $y_i\in R$ is the output variable. In the feature space F, SVR builds a linear regression function in the following form [4]

$$\hat{y}(k)=\langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle + b \tag{1}$$

where $\boldsymbol{w}$ is a vector in F, $\Phi(\cdot)$ is a mapping from the input space to the feature space, b is the bias term, and $\langle\cdot,\cdot\rangle$ stands for the inner product operation in F. The SVR algorithms regard the regression problem as an optimization problem in dual space with the model given by

$$\hat{y}(k)=\sum_{j=1}^{N} a_j K_{ij} + b \tag{2}$$

where a_j is the coefficient of each sample and K_{ij} denotes the kernel function. Sample $\boldsymbol{x}_j$ corresponding to a non-zero a_j value is referred to as the support vector (SV). By using Vapnik's ε-insensitive loss function, the dual form of the optimization problem becomes a quadratic programming (QP) problem

$$\min_{a,a^*} D_\varepsilon=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}K_{ij}(a_i-a_i^*)\left(a_j-a_j^*\right)+\varepsilon\sum_{i=1}^{N}(a_i+a_i^*)-\sum_{i=1}^{N}y_i(a_i-a_i^*) \tag{3}$$

subject to constraints

$$0\le a_i, a_i^*\le C, \sum_{i=1}^{N}(a_i-a_i^*)=0, i=1,\cdots,N \tag{4}$$

where ε is the maximum value of tolerable error and C is a regularization parameter that represents a trade-off between model complexity and effect of tolerance to the error larger than ε. The Lagrange formulation of the QP problem can be further represented as

$$\begin{aligned}L_D=&\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}K_{ij}(a_i-a_i^*)\left(a_j-a_j^*\right)+\varepsilon\sum_{i=1}^{N}(a_i+a_i^*)-\sum_{i=1}^{N}y_i(a_i-a_i^*)\\&-\sum_{i=1}^{N}(\delta_i a_i+\delta_i^* a_i^*)+\varsigma\sum_{i=1}^{N}(a_i-a_i^*)+\sum_{i=1}^{N}[\xi_i(a_i-C)+\xi_i^*(a_i^*-C)]\end{aligned} \tag{5}$$

where $\delta_i, \delta_i^*, \varsigma, \xi_i, \xi_i^*$ are the Lagrange multipliers. The KKT conditions of Eq. (5) are

$$\begin{aligned}&\frac{\partial L_D}{\partial a_i}=\sum_{j=1}^{N}K_{ij}\left(a_j-a_j^*\right)+\varepsilon-y_i-\delta_i+\varsigma+\xi_i=0\\&\frac{\partial L_D}{\partial a_i^*}=-\sum_{j=1}^{N}K_{ij}\left(a_j-a_j^*\right)+\varepsilon+y_i-\delta_i^*-\varsigma+\xi_i^*=0\\&\delta_i,\delta_i^*\ge 0, \delta_i a_i=0, \delta_i^* a_i^*=0\\&\xi_i,\xi_i^*\ge 0, \xi_i(a_i-C)=0, \xi_i^*(a_i^*-C)=0.\end{aligned} \tag{6}$$

According to Eq. (6), at most one of a_i and a_i^* will be nonzero and both are nonnegative. We define coefficient θ_i and margin function $h(\boldsymbol{x}_i)$ as

$$\theta_i=a_i-a_i^* \tag{7}$$

$$h(\boldsymbol{x}_i)\equiv f(\boldsymbol{x}_i)-y_i=\sum_{j=1}^{N}K_{ij}\theta_j+b-y_i. \tag{8}$$

Combining Eqs. (6)–(8), the KKT conditions can be rewritten as

$$\begin{cases} h(\boldsymbol{x}_i)\ge\varepsilon, & \theta_i=-C\\ h(\boldsymbol{x}_i)=\varepsilon, & -C<\theta_i<0\\ -\varepsilon\le h(\boldsymbol{x}_i)\le\varepsilon, & \theta_i=0\\ h(\boldsymbol{x}_i)=-\varepsilon, & 0<\theta_i<C\\ h(\boldsymbol{x}_i)\le-\varepsilon, & \theta_i=C.\end{cases} \tag{9}$$

Based on Eq. (9), the training samples in $\boldsymbol{T}$ can be divided into three subsets as follows [15]. Support set:

$$S=\left\{\begin{array}{l} i|(\theta_i\in(-C,0)\wedge h(x_i)=\varepsilon)\\ \vee(\theta_i\in(0,C)\wedge h(x_i)=-\varepsilon)\end{array}\right\}.$$

Error set:

$$E=\{i|(\theta_i=-C\wedge h(x_i)\ge\varepsilon)\vee(\theta_i=C\wedge h(x_i)\le-\varepsilon)\}.$$

Remaining set:

$$R=\{i|\theta_i=0\wedge|h(x_i)|\le\varepsilon\}.$$

The AOSVR algorithm consists of incremental algorithm and decremental algorithm [15]. The basic idea of incremental algorithm is to find a way to add a new sample to one of the three sets maintaining KKT conditions consistent. When a new sample $\boldsymbol{x}_c$ is received, its corresponding θ_c value is initially set to zero and then gradually changes under the KKT conditions. The relationship between $h(\boldsymbol{x}_i)$, $\Delta\theta_i$ and Δb is given by

$$\Delta h(\boldsymbol{x}_i)=\sum_{j=1}^{N}K_{ij}\Delta\theta_j+K_{ic}\Delta\theta_c+\Delta b. \tag{10}$$