Chemometrics and Intelligent Laboratory Systems 135 (2014) 133–140

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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

A unified recursive just-in-time approach with industrial near infrared spectroscopy application

Mulang Chen, Swanand Khare, Biao Huang *

Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB T6G 2G6, Canada

ARTICLE INFO

Article history:
Received 3 January 2014
Received in revised form 10 April 2014
Accepted 10 April 2014
Available online 22 April 2014

Keywords:
Recursive PLS
Locally weighted PLS
Just-in-time modeling
Near-infrared spectroscopy

ABSTRACT

Time varying and non-linearity issues are commonly seen in soft senso development. Recently, just-in-time approach has been widely used to address the non-linearity problem in near infrared (NIR) spectroscopy modeling. However, to the best of the authors' knowledge, the time varying problems in just-in-time (JIT) framework are rarely discussed and the adaptation strategy for the local models in JIT approach remains an open issue. In this paper, a new model updating approach is proposed which can adjust to process changes by merging the traditional recursive algorithm in the JIT framework. The advantage of the presented approach is that it can solve both time varying and non-linearity issues simultaneously under the JIT framework. The performance of the method has been tested on a spectroscopic dataset from an industrial process. By comparison with traditional PLS, locally weighted PLS and several other on-line model updating strategies, it is shown that the proposed method achieves good performance in the prediction of fuel properties.

1. Introduction

Soft sensors are often built in order to monitor the hard-to-measure target variables in process industry. An important step for building a successful soft sensor is the offline model identification using the recorded historical data. However, there are always factors that cannot be foreseen during the offline modeling phase. For example, catalyst activity can change, instrument can drift and external environment can also change (e.g. weather, temperature, season) [1,2]. As a result of these facts, it is commonly observed that the performance of time invariant models will deteriorate with time after they are put for online operation [3]. To sustain the performance as initially achieved, it often requires to adapt the offline soft sensor to the varying operation conditions during the online operation phase.

The procedure for the adaptation of soft sensors is summarized in Fig. 1. Initially, offline models are built using the historical input data **X** and lab reference data **y**. Usually, data preprocessing is applied before modeling in order to deal with issues like outliers, missing values, noise, etc., and expert knowledge about the process is often introduced to identify the important variables and time delays. As time progresses, the online input data **x** and corresponding lab reference data are accumulated and stored in historical database, which can be used for model updating. In the online updating phase, data preprocessing is also included in order to ensure that the model is adapted to the useful data preventing the model from absorbing disturbances or noises, especially when the hardware sensors deliver faulty measurements [4]. In the meantime, the available expert knowledge of the process is used to determine the window size or forgetting factor for adaptation [1]. Tsymbal have categorized the existing adaptation methods into three approaches [5]:

- Instance selection [6];
- Instance weighting [7]; and
- Ensemble methods [8].

The first two approaches have been named as moving window technique and recursive adaptation technique in [1], respectively.

Generally, linear models can be easily updated using the above mentioned adaptation approaches. For example, moving window PCR is performed by re-training the PCR model periodically after accumulating a certain number of new samples [9]. The latest N data points are used to re-train the model, where N is the window size. Wang et al. proposed an optimized version of the moving window PCR called fast moving window principal component analysis (FMWPCA) [10]. It was shown that the new algorithm is computationally more efficient than conventional block-wise moving window techniques, under the assumption that the window size is at last 3 times the number of variables. Recursive least squares (RLS) [1] and recursive partial least squares (RPLSs) [3] are also well-known adaptive algorithms for linear models, in which the past data are discounted gradually by a forgetting factor. In this way, the most recent samples are assigned higher weights than the distant past samples. The role of forgetting factor is similar to a control of the window size in the moving window approach, which quantifies the importance of the recent samples over the distant past samples.

Adaptation of non-linear models has also been reported in the literature. Liu et al. proposed a moving window kernel PCA (MWKPCA) [11],

* Corresponding author. Tel.: +1 780 492 9016; fax: +1 780 492 2881.
E-mail address: biao.huang@ualberta.ca (B. Huang).

http://dx.doi.org/10.1016/j.chemolab.2014.04.007
0169-7439/

Nomenclature

$\mathbf{X}$	Input matrix ($N \times M$)
$\mathbf{y}$	Output vector ($N \times 1$)
$\mathbf{x}_i$	The i-th sample in the database ($1 \times M$)
$\mathbf{x}_q$	The query sample ($1 \times M$)
$\hat{y}_q$	Predicted output for the query sample (1×1)
N	Number of samples in database $\mathbf{X}$
M	Number of variables in database $\mathbf{X}$
$\mathbf{u}_k$	Weight vector in k-th iteration ($M \times 1$)
$\mathbf{t}_k$	Score vector in k-th iteration ($N \times 1$)
$\mathbf{p}_k$	Loading vector in k-th iteration ($M \times 1$)
q_k	y-loading vector in k-th iteration (1×1)
$\mathbf{R}_x$	Covariance matrix of $\mathbf{X}$ ($M \times M$)
$\mathbf{R}_y$	Covariance matrix of $\mathbf{X}$ and $\mathbf{y}$ ($M \times 1$)
λ	Forgetting factor for updating ($0 < \lambda \leq 1$)
A	Number of latent variables in PLS calculation
d_i	Distance between query sample and i-th sample
w_i	Weight of i-th sample in the database
$\mathbf{W}$	Diagonal Weighting matrix ($N \times N$)
β	Regression coefficients
ρ	Balancing parameter

which is similar to the FMWPCA method discussed above. The idea was to transfer the original variables into the feature space using nonlinear mapping. Then a two step adaptation scheme to remove the oldest sample and add the newest sample was performed using the moving window approach. Li et al. proposed a recursive non-linear PLS algorithm to solve both non-linearity and time varying issues encountered in process modeling [12]. The work was based on the recursive PLS algorithm proposed by Qin [13]. In order to handle the non-linearity issue, the radial basis function (RBF) network was integrated with PLS regression, where the regression was performed between the output space and the extended input space. The extended input was constructed by the original input variables, the hidden nodes outputs of the RBF network and a constant column with all elements being 1. Liu et al. [14] proposed the adaptive version of support vector machine (SVM) [15] named online kernel learning algorithm (OKL), which was based on the Adaptive Kernel Learning framework proposed by Wang et al. [16]. The proposed OKL algorithm can adaptively learn the process dynamics using relatively small samples.

Recently, JIT modeling approach has been proposed to address the non-linearity problem in soft sensor development [17]. Under the JIT modeling framework, a local model is built after the query sample is available, using the historical samples most relevant to the query sample. The local model is used to estimate the output value; after which the local model is discarded. Although just-in-time approach is a local learning method, the time varying issue cannot be solved unless adaptive mechanism is introduced into the JIT framework [18]. Chen et al. proposed a new local modeling algorithm named adaptive local kernel-based learning (ALKL) scheme, where an adaptive weighted least squares support vector regression (AW-LSSVR) was employed to establish the local model [18]. In ALKL, for each query data, the trade-off parameters of LSSVR were adjusted iteratively along the gradient descending direction so that the local model can be updated recursively. The self-tuning of trade-off parameters was achieved by performing the fast leave-one-out cross-validation (FLOO-CV) criterion [19] on the most relevant l samples in the database. Similarly, the FLOO-CV strategy was adopted by Liu et al. [20] for adaptive selection of kernel parameters in a JIT modeling framework. In the above two adaptive JIT algorithms, only the local kernel parameters are updated recursively while the database is always fixed. In this regard, these two methods are quite different from traditional recursive methods where the most recent sample pair ($\mathbf{x}_t$,y_t) is merged into the database for model update. Under this circumstance, our motivation is to update the JIT models by adopting and weighting recent samples instead of refreshing the local tuning parameters. In this paper, an adaptive algorithm which prioritizes the most recent samples as well as nearby samples is proposed to solve both time varying and non-linearity problems in soft sensor development. The usefulness of this algorithm is demonstrated by an industrial case study.

2. Theory and algorithm

2.1. Just-in-time framework

The just-in-time (JIT) learning method [21,22,20], which is also called 'lazy learning' or 'locally weighted modeling' method, has been proposed in the machine learning literature to solve the nonlinear modeling problem. There are four key components in just-in-time approach, namely, historical database, similarity measurement, weight function and modeling technique. When the query sample arrives, the JIT method builds a local model online using the most relevant data. It should be noted that JIT itself is not a regression technique, but it can be considered as a framework under which data-driven modeling techniques can be applied in a more effective way. For example, partial least squares (PLSs) regression can be applied within JIT framework, which results in the well-known method called locally weighted PLS (LW-PLS) [23]. Also, nonlinear regression methods like artificial neural network (ANN) [24], least square support vector machines (LS-SVM) [15] and other kernel learning methods have also been used within the JIT framework [20]. Let $\mathbf{X} \in \Re^{N \times M}$ and $\mathbf{y} \in \Re^{N \times 1}$ be the input and output matrices, respectively. N is the number of samples and $\mathbf{x}_i$ denotes the input of the i-th sample. The widely-used LW-PLS algorithm is described as below:

1. When the query sample $\mathbf{x}_q$ arrives, calculate the distance d_i between $\mathbf{x}_q$ and the historical data point $\mathbf{x}_i$.
2. Calculate the weight of each sample using weight function $w_i = f(d_i)$.

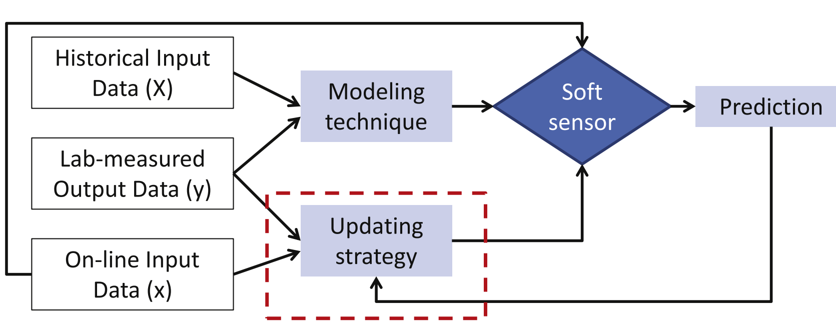

Fig. 1. Online adaptation of soft sensors.