Chemometrics and Intelligent Laboratory Systems 175 (2018) 104–115

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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

Highly-overlapped, recursive partial least squares soft sensor with state partitioning via local variable selection

Dominic V. Poerio, Steven D. Brown *

Department of Chemistry and Biochemistry, University of Delaware, 163 The Green, Newark, DE 19716, USA

ARTICLE INFO

Keywords:
Soft sensor
Process analysis
State partitioning
Online prediction
Recursive partial least squares regression

ABSTRACT

We report the use of a soft sensor ensemble based on recursive partial least squares with a large number of overlapping models. The proposed method uses process memory attenuation in the ensemble by varying the number of training samples included in each model, while always including the most recent samples, which are usually the most relevant for prediction of new samples, and also ensures that no local models are invalidated due to drift. To achieve state partitioning of the process data, covariance-based variable selection is performed on each of the model regions to ensure that only variables most relevant to the dominant process state are included in each of the local models. This approach yields a distribution of predictions from the models, permitting a prediction of the target property from the summary statistics of the observed prediction distribution. The effectiveness of the proposed method is demonstrated by testing against a conventional global soft sensor as well as a state-localized soft sensor, both with and without variable selection, on two soft sensing applications developed from real industrial processes employing various model updating frequencies. Results from the experiments demonstrate that the proposed method tends to outperform a global soft sensor in most cases, and is highly competitive with the compared state-localized soft sensor, indicating that the proposed method achieves accurate state partitioning.

1. Introduction

Soft sensors have become ubiquitous in the chemical industry due to their ability to allow real-time estimation and monitoring of certain process variables that have typically required an offline laboratory analysis by modeling those process variables in terms of more readily measured, common process diagnostic variables such as pressure, temperature and viscosity. A soft sensor is best suited for predictive control in situations where the analysis has a delay (e.g., with gas chromatography) or in other applications where the required offline analyses to determine the process variable cannot be completed at a frequency high enough to guide real-time process control decisions. For example, a soft sensor is suited to situations where a sample taken from a process needs time-consuming workup prior to a spectroscopic analysis. The models comprising soft sensors typically require regular maintenance and updating due to changing process dynamics and drifts which can quickly invalidate the underlying relationships [1]. Common strategies for dealing with these changes is to use a model that can be adapted to the changing process conditions by employing an autoregressive structure, for instance one using autoregression with exogenous inputs (ARX) [2,3], one based on a recursive regression approach [4,5], or by using various moving-window models [6,7].

Much of the recent research into soft sensors has focused on temporal-localization or spatial-localization of the process data, also referred to as state partitioning [5,8,9]. The goal of the localization is to either approximate local regions in the data as linear or to characterize local regions in the data as distinct process states. Kernel density estimation, distance and classification metrics, or model-based clustering methods can then be used to determine probabilities that the new sample belongs to a specific process state, followed by weighting the predictions from all of the models. A somewhat analogous approach is the use of just-in-time (JIT) learning [10], sometimes called lazy learning, in which a similarity metric is used to select historical process samples to create a model for a property describing the new sample. An issue with these localization approaches is that models representing older states often have very low predictive performance on new samples due to the changing dynamics of the process and drift in the process diagnostic sensors. Updating older models with information from the new samples will help ensure that no models become invalid, but this updating is contrary to the purpose of modeling the states separately because, with an update, information pertinent to the most recent state propagates across all of the process state models, which were specifically built to describe the property

* Corresponding author.
E-mail address: sdb@udel.edu (S.D. Brown).

https://doi.org/10.1016/j.chemolab.2018.02.006
Received 14 December 2017; Received in revised form 8 February 2018; Accepted 12 February 2018
Available online 15 February 2018

relationship in distinct states. JIT learning can have similar issues, in that the historical samples chosen for construction of the new model may belong to various process states, or, alternatively, the historical samples chosen may not fully describe the current process state.

To address these issues, we propose a new paradigm for modeling relevant process states in soft sensing applications, as well as selecting samples that belong to a process state. To handle the general issue of models describing older process states and potentially not selecting the correct historical samples to describe the current process state, we first suggest a method that uses only those models that include the most recent samples, because these models generally have the highest predictive accuracy for the target (response) variable. Instead of performing a typical state partitioning of the data with some type of temporal or spatial localization, we construct an ensemble of models with varying window sizes, with the aim of establishing process memory attenuation in the ensemble. By varying the number of historical samples in each individual model, only the memory in each individual model is altered, so we also account for the differing covariance structure of the models defining an empirical process state by performing a covariance-based variable selection on each of the models using the CovSel algorithm [11]. Data collected over different time periods but describing the same process state should yield a selection of the same, or very similar, sets of variables. Again, this approach is distinct from typical state partitioning methods used elsewhere, since each process state is defined with respect to the most recent samples to give an ensemble of models that should all be suited for prediction on the incoming samples and are all simultaneously updated to account for drifts in the process, whether from sensor drifts or from dynamic changes in covariance structure. In this report, we have applied the framework described above to the recursive partial least squares approach for modeling dynamic processes, so we call this method highly-overlapped recursive partial least squares (HORPLS).

The remainder of this report is structured as follows: in Section 2, we review the methods utilized in this work, namely partial least squares, recursive partial least squares, the CovSel algorithm for variable selection, and we provide a description of our proposed method. We also briefly discuss other state partitioning methods applied to soft sensing. In Section 3, the experimental details, computational information, and descriptions of analyzed datasets are given. Section 4 compares the performance of HORPLS, RPLS, and a localized method based on clustering via a Gaussian mixture model paired with local RPLS, all tested with and without variable selection. The conclusions of the work are given in Section 5.

2. Background

2.1. Notation

We use bold capital letters (e.g., $\mathbf{X}_{(n\times m)}$) to denote a matrix, particularly a matrix of predictors, where the n rows represent samples and the m columns represent predictor variables defined on a set of process sensors. Column vectors are denoted by a lowercase, bold letter (e.g., $\mathbf{y}_{(n\times 1)}$), transposition is indicated by a superscript T (i.e. $\mathbf{X}^{\mathrm{T}}{}_{(m\times n)}$, $\mathbf{y}^{\mathrm{T}}{}_{(1\times n)}$), and matrix inversion is represented by a superscript -1 (i.e., $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}{}_{(m\times m)}$). Scalar quantities are denoted by an italicized letter (e.g., $a_{(1\times 1)}$ or $A_{(1\times 1)}$).

2.2. Recursive partial least squares

Partial least squares (PLS) regression is the basis for the modeling methods discussed below, but because the method is well-established, only a brief overview of the method is presented here. More detailed treatments of the subject, for instance the classic tutorial by Geladi and Kowalski [12], are available elsewhere.

In linear regression, a response $\mathbf{Y}_{(n\times p)}$ is modeled as a function g of $\mathbf{X}_{(n\times m)}$:

$$\mathbf{Y} = g(\mathbf{X}) = \mathbf{XB} + \mathbf{E} \tag{1}$$

where p is the number of response variables, $\mathbf{B}_{(m\times p)}$ is a matrix of regression coefficients and $\mathbf{E}_{(n\times p)}$ is a residual matrix. If the covariance matrix of $\mathbf{X}$ is ill-conditioned, this relation cannot be directly modeled with the ordinary least squares approach, and a regularized regression is required. Even if the covariance matrix of $\mathbf{X}$ is well-conditioned, modeling can often be improved by regularizing the regression. In the case of PLS, the regression is regularized by excluding specific sources of variation in the data by first performing a latent decomposition of the $\mathbf{X}$ and $\mathbf{Y}$ data:

$$\mathbf{X} = \mathbf{TP}^{\mathrm{T}} + \mathbf{E}_{\mathrm{X}} \tag{2}$$

$$\mathbf{Y} = \mathbf{UQ}^{\mathrm{T}} + \mathbf{E}_{\mathrm{Y}} \tag{3}$$

where $\mathbf{T}_{(n\times v)}$ and $\mathbf{U}_{(n\times v)}$ are, respectively, the $\mathbf{X}$ and $\mathbf{Y}$ scores, and $\mathbf{P}_{(m\times v)}$ and $\mathbf{Q}_{(p\times v)}$ are, respectively, the $\mathbf{X}$ and $\mathbf{Y}$ loadings, and $\mathbf{E}_{\mathrm{X}(n\times m)}$ and $\mathbf{E}_{\mathrm{Y}(n\times p)}$ are residual matrices. Here, v is the number of retained latent variables, such that $v \leq \min(n, m, r)$, where r is the rank of $\mathbf{X}$, thereby giving a low-rank representation of $\mathbf{X}$ and regularizing the regression solution. After the first vector of $\mathbf{X}$ and $\mathbf{Y}$ scores have been found, they are linked by the so-called inner PLS linear relation:

$$\mathbf{u}_1 = \beta_1 \mathbf{t}_1 + \mathbf{e}_{\mathrm{u}} \tag{4}$$

where β_1 is the linear regression coefficient minimizing the sum of the squared residuals $\mathbf{e}_{\mathrm{u}}$. Once the first set of scores and the inner relation are found, $\mathbf{X}$ and $\mathbf{Y}$ are deflated:

$$\mathbf{E}_{\mathrm{X},1} = \mathbf{X} - \mathbf{t}_1 \mathbf{p}_1{}^{\mathrm{T}} \tag{5}$$

$$\mathbf{E}_{\mathrm{Y},1} = \mathbf{Y} - \beta_1 \mathbf{t}_1 \mathbf{q}_1{}^{\mathrm{T}} \tag{6}$$

and the algorithm proceeds again with the residual matrices. For prediction, the new samples $\mathbf{X}_{\mathrm{new}}$ are projected into the score space to find $\mathbf{T}_{\mathrm{new}}$ and a rearranged form of equation (6) is used to estimate the response matrix $\widehat{\mathbf{Y}}$.

$$\widehat{\mathbf{Y}} = \sum_{i}^{v} \beta_i \mathbf{t}_{\mathrm{new},i} \mathbf{q}_i{}^{\mathrm{T}} \tag{7}$$

PLS regression is useful in situations where a regularized regression solution is required, but the method is not robust to dynamics in the covariance structure of the data. To deal with the dynamic changes in the relationship between predictor and response variables that commonly occur in time-varying processes, Helland et al. [13] reported a recursive version of the PLS algorithm to permit on-line updating of a PLS model. Qin [4] derived a computationally more efficient method for recursively updating a PLS model with a new sample $\mathbf{x}_{t+1}$ and a new response $\mathbf{y}_{t+1}$ by using the following equations:

$$\mathbf{X} = \begin{bmatrix} \lambda \mathbf{P}^{\mathbf{T}}{}_t \\ \mathbf{x}_{t+1} \end{bmatrix} \tag{8}$$

$$\mathbf{Y} = \begin{bmatrix} \lambda \mathbf{C}_t \mathbf{Q}^{\mathbf{T}}{}_t \\ \mathbf{y}_{t+1} \end{bmatrix} \tag{9}$$

where λ is a forgetting factor that determines the extent to which older samples influence the updated model, and $\mathbf{C} = diag(\beta_1, \ldots, \beta_r)$. Updating can also be done in batch-wise fashion if multiple new samples are received and are available for updating simultaneously.

Mean and variance updating of the data [14], which can be used for adaptive autoscaling of the predictors, can easily be combined with any updating approach to improve adaptation of the model. Assuming that the mean sample of the set of responses in $\mathbf{X}$ at time t is $\mathbf{d}_t$, the variance of those sensor responses at time t is $\boldsymbol{\sigma}_t$ [2], and that at time t we have