Dynamic transfer partial least squares for domain adaptive regression

Zhao, Zhijun; Yan, Gaowei; Ren, Mifeng; Cheng, Lan; Zhu, Zhujun; Pang, Yusong

DOI
10.1016/j.jprocont.2022.08.011

Publication date
2022

Document Version
Final published version

Published in
Journal of Process Control

Journal of Process Control 118 (2022) 55–68

Contents lists available at ScienceDirect

Journal of Process Control

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

Dynamic transfer partial least squares for domain adaptive regression

Zhijun Zhao [a], Gaowei Yan [a,*], Mifeng Ren [a], Lan Cheng [a], Zhujun Zhu [b], Yusong Pang [c]

[a] College of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan, 030024, Shanxi, China
[b] Shanxi Gemeng Sino-US Clean Energy R & D Center Co., Ltd., Taiyuan, 030031, Shanxi, China
[c] Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, Delft, 2628CD, Netherlands

ARTICLE INFO

Article history:
Received 18 April 2022
Received in revised form 29 July 2022
Accepted 22 August 2022
Available online xxxx

Keywords:
Transfer learning
Dynamic partial least squares
Domain adaptive regression

ABSTRACT

The traditional soft sensor models are based on the independent and identical distribution assumption, which are difficult to adapt to changes in data distribution under multiple operating conditions, resulting in model performance deterioration. The domain adaptive transfer learning methods learn knowledge in different domains by means of distribution alignment, which can reduce the impact of data distribution differences, and effectively improve the generalization ability of the model. However, most of the existing models established by domain adaptation methods are static models, which cannot reflect the dynamic characteristics of the system, and have limited prediction accuracy when applied to dynamic system modeling under multiple operating conditions. The dynamic system modeling methods can effectively extract the dynamic characteristics of the data, but they cannot deal with the concept drift problem caused by the change of data distribution. This paper proposes a new dynamic transfer partial least squares method, which maps the high-dimensional process data into the low-dimensional latent variable subspace, establishes the dynamic regression relationship between the latent variables and the labels, and realizes the systematic dynamic modeling, at the same time, the model adds regular terms for distribution alignment and structure preservation, which realizes dynamic alignment of data distribution difference. The effectiveness of the proposed method is validated on three publicly available industrial process datasets.

1. Introduction

Data-driven soft-sensing modeling strategies based on machine learning and multivariate statistical methods have been successfully applied in the field of industrial process control [1]. However, machine learning modeling methods need to satisfy the assumption that the training data and prediction data are independent and identically distributed (iid) [2]. In the actual industrial process, the assumption of independent and identical distribution is difficult to be satisfied due to the rapid changes in production operating conditions and environment, resulting in model mismatch and a decrease in the prediction accuracy. This phenomenon of degrading model prediction performance due to changes in data statistical properties and underlying distributions is known as concept drift [3].

The essence of the concept drift problem is that the training data streams used in the traditional machine learning modeling process are static data streams, while the actual data streams are time-varying dynamic data streams [4]. Due to resource constraints, the static data used for modeling cannot cover all production conditions. When the operating conditions change, the existing models cannot reflect the characteristics of the new conditions, which will inevitably lead to model performance degradation. For example, in the process of chemical production, the performance of catalysts degrades and fails [5]; in the process of solid waste incineration, the types of raw materials and physical and chemical properties are different [6]; in the process of rolling steel production, the specifications and batches of products produced are different, and the required operating conditions are different [7].

In theory, re-acquiring data and training a new model can solve the above-mentioned concept drift problem, but in practical applications, it is not easy to obtain enough training data, especially labeled data, under a brand-new working condition, some quality index data needs to be analyzed and obtained under laboratory conditions, which is time-consuming, labor-intensive and expensive.

Transfer learning [8] allows the use of data from different domains and data distributions during training and testing, breaking the traditional machine learning assumption that training and testing data are independent and identically distributed. The data under the existing working conditions are regarded as the source domain, and the data under the unknown working conditions are regarded as the target domain, by learning the knowledge of the source domain and transferring the knowledge of the source

* Corresponding author.
E-mail address: yangaowei@tyut.edu.cn (G. Yan).

https://doi.org/10.1016/j.jprocont.2022.08.011
0959-1524/

domain to the target domain, the prediction accuracy on the target domain and the generalization ability of the model can be improved. Domain adaptation method is the most commonly used method in transfer learning, it can improve the generalization ability of the model by narrowing the data distribution gap between the source domain and the target domain.

Although related theories and methods of transfer learning are often found in the literature, most of them are based on classification problems, and many transfer methods do not work well for regression problems. The possible reason is that for classification problems, there are clear decision boundaries between different categories, through the transfer learning method, the decision boundaries of different categories on the target domain will be expanded, which will help to improve the generalization ability of the model at the target domain. But for regression problems, the label space is continuous and there is no obvious decision boundary [9]. For regression problems such as industrial soft sensing, the situation is more complicated, because the actual industrial processes are mostly non-stationary time series [10], and the data presents heteroscedasticity and dynamic characteristics, therefore, the construction of soft sensing model should not only consider the problem of data drift, but also consider the dynamic problem of data. Most of the existing regression models are static models, in order to effectively improve the prediction accuracy and the generalization of the model, it is necessary to study the dynamic transfer soft sensor model.

The data-driven soft sensor modeling method utilizes easily accessible auxiliary variables to establish a prediction model to realize the prediction of key quality indicators that are difficult to measure directly [11,12]. Common soft-sensor modeling methods include multivariate statistical methods represented by partial least squares (PLS) [13], canonical correlation analysis (CCA) [14], latent variable regression (LVR) [15], and machine learning methods represented by support vector machine (SVM) [16] and extreme learning machine (ELM) [17]. As a modeling method for supervised data analysis, partial least squares has been widely used in chemometrics and economics. The partial least squares method projects the high-dimensional process data into the mutually orthogonal low-dimensional latent variable space, which can effectively eliminate the influence of data multicollinearity while reducing the dimension. The partial least squares method extracts the latent variables in the input (feature) space and the output (label) space respectively, so that the covariance between the latent variables is maximized, and the quality variable is explained to the greatest extent while extracting the variation information.

Most of the actual industrial process data has dynamic characteristics, traditional multivariate statistical regression methods, such as principal component regression (PCR) [18], and partial least squares (PLS) are all static models that cannot reflect the dynamic characteristics of the system. To this end, a series of dynamically extended algorithms are proposed. Ku et al. proposed a DPCA [19] algorithm, which perform standard PCA analysis on time-lagged augmented data, this method fails to establish a dynamic relationship in the latent variable space, so it is still a static model in nature. Dong and Qin proposed the DiPCA [20] algorithm to extract dynamic latent variables with the largest variance relationship. Although this method establishes a dynamic model of latent variables, as an unsupervised algorithm, it cannot be used for supervised regression soft sensing applications. Li et al. proposed the DPLS [21] algorithm, which input the weighted combination of time-lagged data into the model to maximize the linear combination of the latent variables in the feature space and the variance of the latent variables in the label space. Dong and Qin further improved the model, to achieve a dynamic model (DiPLS) [22] whose outer model is consistent with the inner model, making the model more interpretable.

Inspired by the above dynamic expansion method of partial least squares, and considering the concept drift problem caused by the difference of data distribution in the industrial process of multiple working conditions, this paper proposes a dynamic transfer partial least squares method (DTPLS), which maps high-dimensional time-lagged augmented data into a low-dimensional latent variable space, a dynamic regression model between latent variables and labels is established, and the distribution dynamic alignment regular term is added to the model to realize the dynamic alignment of the distribution differences of the working conditions. The direction of the data mapping can be obtained by the Lagrange multiplier method to obtain a analytical solution.

The main contributions of this paper are as follows:

(1) This paper proposes a novel dynamic transfer partial least squares algorithm (DTPLS) by considering the process dynamics and distribution shift, which can solve the problem of concept drift caused by the difference of data distribution in the industrial process of multiple working conditions.

(2) Under the framework of structural risk minimization, distribution alignment regularization and structure preservation regularization are designed based on distribution differences and smoothness assumptions, which effectively improve the generalization ability of the model.

The rest of the paper is organized as follows: Section 2 gives a brief overview of related works, Section 3 reviews the existing methods including traditional PLS and dynamic inner PLS, Section 4 presents the proposed DTPLS algorithm and develops the theoretical basis of the algorithm, Section 5 compares our algorithm with existing methods on three public benchmark datasets from industrial process, Section 6 presents theoretical analysis and experimental verifications, Section 7 concludes the paper.

2. Related work

Complex industrial process data generally include: dynamic, nonlinear, multi-scale, multi-modal and other process characteristics [23–25]. Affected by the equipment inertia and closed-loop control characteristics of the actual industrial control system, the continuous observation values obtained by the monitoring system are often sequence-related in time. Such a system whose state variables change with time is called a dynamic system. The samples generated by the dynamic system are time series with autocorrelation, and the assumption of independent and identical distribution among the samples is no longer satisfied.

The multi-modal characteristics of the data are due to the continuous adjustment of production equipment, raw materials and environments with the change of production tasks, driven by the diverse product market demands, resulting in the multi-working conditions of the production process [26]. Multi-condition (multi-modal) characteristics are caused by process nonlinearity, that is, the process structure and parameters change due to the process system operating at different equilibrium points. It is generally considered that the working conditions are stable inside, and there are different model structures or data distributions between different working conditions.

Multi-condition soft sensor modeling methods can be divided into multi-model integration methods and local modeling methods. The multi-model integration methods divide the multi-condition process into multiple single-condition processes to establish models respectively, and use the corresponding single-condition sub-model to predict [27]. The local modeling methods are to select the samples with the highest similarity with the current working condition data from the historical working condition data for modeling, and the typical methods are the sliding window(SW) and the just-in-time-learning(JITL) method [28,29].

The dynamic problem and the concept drift problem caused by the change of data distribution are the two main problems in the

field of industrial multi-condition soft sensor, and there are relatively few studies focusing on these two aspects simultaneously. In order to achieve dynamic system modeling, Ku et al. Li et al. Lee et al. used direct data augmentation to extend the traditional multivariate statistical methods PCA, CCA, and PLS, respectively, and proposed DPCA [19], DCCA [30] and DPLS [31] method. However, this method of direct data augmentation cannot provide an explicit representation of the dynamic relationship of the system, and the extracted latent variables still have temporal correlations. To this end, Dong and Qin proposed the dynamic inner model modeling methods DiPCA [20], DiCCA [32], DiPLS [22], which established the autoregressive relationship of latent variables in the latent variable space, and expressed the dynamic relationship of the system explicitly, make the model more interpretable. In addition, Xu et al. [33], Gao et al. [34] used slow feature analysis (SFA) method to establish an autoregressive model of a dynamic system, providing a new idea for dynamic system modeling.

Aiming at the difference of working conditions caused by the change of data distribution under multiple working conditions, Liu et al. [35], Zhang et al. [36,37], Lou et al. [38] proposed common feature extraction and special feature extraction method, common features reflect the common information of the process of multiple working conditions, and special features reflect the unique information of each working condition. These methods divide the feature space into several orthogonal subspaces, and construct statistics in each subspace to evaluate the process. The above research work provides a new research idea for multi-condition soft sensor modeling: different working conditions with distribution differences have similar process mechanisms, using the common features of similar working conditions and narrowing the feature differences between working conditions can effectively improve the modeling accuracy.

Under the framework of deep learning, Jiang et al. [9] used an adversarial regressor to maximize the prediction error on the target domain, and at the same time used an optimized feature extractor to minimize the prediction error, realizing cross-domain keypoint domain adaptive regression. Chen et al. [39] used an represent subspace distance to narrow the distribution differences between different domains, and also proposed an unsupervised transferable domain adaptation method for regression problems under the framework of deep learning. Du et al. [40] used the time-domain distribution matching method to match the distribution of different periods to learn common knowledge in different periods, which minimized the distribution differences and effectively improved the accuracy of regression prediction. The transfer regression method based on deep learning has received extensive attention in recent years, and a large number of studies have shown that the modeling method based on deep learning can achieve better prediction results. However, deep learning methods generally have many model parameters, poor model interpretability, long training time, and often require GPU devices with large computing power.

In the field of actual industrial soft sensing, the economy and interpretability of the soft sensing model is the guarantee for the model to move from theory to practical application. For this reason, many researchers pay attention to the soft sensing transfer model under the shallow framework. Nikzad-Langerodi et al. [41–43] reduced the distribution differences in different domains by aligning the first and second moments of samples under the framework of nonlinear iterative partial least squares, learned a domain-invariant representation of transfer regression model, and applied to the calibration of the instrument. Similarly, Huang et al. [44] used the Hilbert–Schmidt criterion to narrow the distribution differences in different domains and proposed a domain-adaptive partial least squares algorithm, and the method is nonlinearly extended using the kernel trick.

The transfer learning methods of domain adaptation learn common knowledge in different domains by means of distribution alignment, which reduce the influence of data distribution differences and effectively improve the generalization ability of the model. However, most of the models established by the existing domain adaptation transfer methods are a static model, which cannot reflect the dynamic characteristics of the system. The dynamic system modeling methods can effectively extract the dynamic characteristics of the data, but they cannot deal with the concept drift problem caused by the change of data distribution. Based on this, this paper proposes a new dynamic transfer partial least squares modeling method, which maps the high-dimensional process data into the low-dimensional latent variable subspace, establishes the dynamic regression relationship between the latent variables and the labels, realizes the dynamic modeling of the system, at the same time, the model adds a regular term for the distribution difference to realize the dynamic alignment of the data distribution difference.

3. Existing methods

3.1. PLS

Given a feature input matrix $\mathbf{X} \in \mathrm{R}^{nxm}$, a label output matrix $\mathbf{Y} \in \mathrm{R}^{nxd}$, where n is the number of samples, m is the input feature dimension, and d is the output feature dimension. The multivariate partial least squares algorithm decomposes the input matrix $\mathbf{X}$ and the output matrix $\mathbf{Y}$ into the following bilinear terms:

$$\begin{aligned}\mathbf{X} &= \mathbf{T}\mathbf{P}^{\mathrm{T}}+\mathbf{E} = \sum_{a=1}^{A} \mathbf{t}_a \mathbf{p}_a{}^{\mathrm{T}} + \mathbf{E} \\ \mathbf{Y} &= \mathbf{U}\mathbf{Q}^{\mathrm{T}}+\mathbf{F} = \sum_{a=1}^{A} \mathbf{u}_a \mathbf{q}_a^{\mathrm{T}} + \mathbf{F}\end{aligned} \tag{1}$$

where $\mathbf{T} = [\ \mathbf{t}_1,\mathbf{t}_2 \cdots,\mathbf{t}_A\ ] \in \mathrm{R}^{nxA}$ and $\mathbf{U} = [\ \mathbf{u}_1,\mathbf{u}_2 \cdots \mathbf{u}_A,\ ] \in \mathrm{R}^{nxA}$ are the input and output score matrices, respectively, $\mathbf{P} \in \mathrm{R}^{mxA}$ and $\mathbf{Q} \in \mathrm{R}^{dxA}$ are the loading matrices, $\mathbf{E} \in \mathrm{R}^{nxm}$ and $\mathbf{F} \in \mathrm{R}^{nxd}$ are the residual matrix, and A is the latent variable dimension, which are generally obtained through cross-validation. PLS extracts the principal components $\mathbf{t}_a$ and $\mathbf{u}_a$ by projecting the input and output matrices into their respective latent variable spaces to maximize the covariance of the principal components $\mathbf{t}_a$ and $\mathbf{u}_a$. The regular mathematical expression is to solve the following optimization problem:

$$\begin{aligned}&\max \operatorname{cov}(\mathbf{t}_a, \mathbf{u}_a) = \mathbf{c}_a^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{X}\mathbf{w}_a \\ &s.t. \quad \| \mathbf{c}_a \| = 1, \| \mathbf{w}_a \| = 1\end{aligned} \tag{2}$$

where $\mathbf{X}\mathbf{w}_a = \mathbf{t}_a$, $\mathbf{Y}\mathbf{c}_a = \mathbf{u}_a$, $\mathbf{w}_a$ is the input weight vector, and $\mathbf{c}_a$ is the output weight vector. The above objective formula can be solved by the Lagrange multiplier method or the nonlinear iterative method (NIPALS) [45].

3.2. DiPLS

It can be seen from the PLS objective that the model established by the PLS algorithm is a static model. For the dynamic data modeling problem, Dong and Qin proposed a dynamic PLS algorithm DiPLS [22] as follows:

The sampled data obtained at time k are denoted as $\mathbf{x}_k$ and $\mathbf{y}_k$, and DiPLS establishes the following dynamic inner model in the latent variable space:

$$u_k = \beta_0 t_k + \beta_1 t_{k-1} + \cdots + \beta_\tau t_{k-\tau} + r_k \tag{3}$$

where $u_k = \mathbf{y}_k^{\mathrm{T}}\mathbf{c}$, $t_k = \mathbf{x}_k^{\mathrm{T}}\mathbf{w}$, $\boldsymbol{\beta} = (\beta_0, \beta_1 \cdots \beta_\tau)^{\mathrm{T}}$ is the vector of autoregressive coefficients, r_k is the residual, τ is the time-lagged coefficient. The estimation $\hat{u}_k$ of the output score u_k can be achieved by a dynamic inner model:

$$\begin{aligned}\hat{u}_k &= \beta_0 t_k + \beta_1 t_{k-1} + \cdots + \beta_\tau t_{k-\tau} \\ &= \mathbf{x}_k^{\mathrm{T}}\mathbf{w}\beta_0 + \mathbf{x}_{k-1}^{\mathrm{T}}\mathbf{w}\beta_1 + \cdots + \mathbf{x}_{k-\tau}^{\mathrm{T}}\mathbf{w}\beta_\tau \\ &= [\mathbf{x}_k^{\mathrm{T}}\mathbf{x}_{k-1}^{\mathrm{T}} \cdots \mathbf{x}_{k-\tau}^{\mathrm{T}}](\boldsymbol{\beta} \otimes \mathbf{w})\end{aligned} \tag{4}$$

where, $\otimes$ is the Kronecker product. Maximize the covariance between outer model u_k and inner model $\hat{u}_k$, the objective of DiPLS can be constructed as:

$$\max \sum_{k=\tau}^{N+\tau} \mathbf{c}^{\mathrm{T}}\mathbf{y}_k[\mathbf{x}_k^{\mathrm{T}}\mathbf{x}_{k-1}^{\mathrm{T}} \cdots \mathbf{x}_{k-\tau}^{\mathrm{T}}](\boldsymbol{\beta} \otimes \mathbf{w}) \tag{5}$$

$$s.t. \quad \| \mathbf{c} \| = 1, \| \mathbf{w} \| = 1, \| \boldsymbol{\beta} \| = 1$$

let $\mathbf{X}_i = \left[\mathbf{x}_i\mathbf{x}_{i+1} \cdots \mathbf{x}_{i+N}\right]^{\mathrm{T}}$, for $i = 0, 1, 2, \ldots, \tau$,

$\mathbf{Z}_\tau = [\mathbf{X}_\tau\mathbf{X}_{\tau-1} \cdots \mathbf{X}_0]$,

$\mathbf{Y}_\tau = \left[\mathbf{y}_\tau\mathbf{y}_{\tau+1} \cdots \mathbf{y}_{N+\tau}\right]^{\mathrm{T}}$

the above objective can be rewritten in matrix form:

$$\begin{aligned}&\max \mathbf{c}^{\mathrm{T}}\mathbf{Y}_\tau\mathbf{Z}_\tau(\boldsymbol{\beta} \otimes \mathbf{w}) \\ &s.t. \ \| \mathbf{c} \| = 1, \| \mathbf{w} \| = 1, \| \boldsymbol{\beta} \| = 1\end{aligned} \tag{6}$$

The Lagrange multiplier method can solve the above objective formula.

4. Proposed method

4.1. Problem formulation

Define the existing working condition data $\mathbf{X}^{\mathrm{s}}$ as the source domain $\mathcal{D}_{\mathrm{S}} = \{\mathbf{X}^{\mathrm{s}}, P\}$, and the unknown working condition data $\mathbf{X}^{\mathrm{t}}$ as the target domain $\mathcal{D}_{\mathrm{T}} = \{\mathbf{X}^{\mathrm{t}}, Q\}$. The data from the source domain obeys the probability distribution $P(\mathbf{x})$, and the data from the target domain obeys the probability distribution $Q(\mathbf{x})$, $P(\mathbf{x}) \neq Q(\mathbf{x})$. $h : \mathbf{X} \in \mathrm{R}^{nxm} \rightarrow \mathbf{Y} \in \mathrm{R}^{nxd}$ is the mapping function from feature space to label space. The purpose of multi-condition transfer soft sensing is to learn an empirical label function $\hat{h} : \mathbf{X} \in \mathrm{R}^{nxm} \rightarrow \mathbf{Y} \in \mathrm{R}^{nxd}$ through the knowledge of existing operating conditions (source domain), so that its expected error under unknown operating conditions (target domain) is minimized, which is:

$$\min \varepsilon_T(\hat{h}, h) = E_{X\sim Q}\left[|\hat{h} - h|\right] \tag{7}$$

According to the structural risk minimization (SRM) [46], $\hat{h} = \operatorname{argmin}\ loss(h(x), y) + R(h)$, where $loss(h(x), y)$ is empirical risk on data samples, $R(h)$ is the regularization term. For our task, the target domain samples are unlabeled, we can only perform the empirical risk minimization on the source domain. In order to minimize the distribution difference between the source and target domains, a distribution alignment regularization term is added. Moreover, another regularization term for structure preservation is added based on the smoothness assumption. We will describe our method in detail.

4.2. Latent space empirical risk minimization

We build an empirical loss function by minimizing the squared loss of the source domain output score u_k^{s} and its inner model estimation $\hat{u}_k^{\mathrm{s}}$:

$$\begin{aligned}\mathcal{L}_{L2} &= \frac{1}{N}\sum_{k=\tau}^{N+\tau}\left(u_k^{\mathrm{s}} - \hat{u}_k^{\mathrm{s}}\right)^2 \\ &= \frac{1}{N}\sum_{k=\tau}^{N+\tau}\left(\left(\mathbf{y}_k^{\mathrm{s}}\right)^{\mathrm{T}}\mathbf{c} - \left[\left(\mathbf{x}_k^{\mathrm{s}}\right)^{\mathrm{T}}\left(\mathbf{x}_{k-1}^{\mathrm{s}}\right)^{\mathrm{T}} \cdots \left(\mathbf{x}_{k-\tau}^{\mathrm{s}}\right)^{\mathrm{T}}\right](\boldsymbol{\beta} \otimes \mathbf{w})\right)^2 \\ &= \frac{1}{N}\| \mathbf{Y}_\tau^{\mathrm{s}}\mathbf{c} - \mathbf{Z}_\tau^{\mathrm{s}}(\boldsymbol{\beta} \otimes \mathbf{w}) \|^2\end{aligned} \tag{8}$$

where $\mathbf{Z}_\tau^{\mathrm{s}}$ and $\mathbf{Y}_\tau^{\mathrm{s}}$ are the time-lagged data augmentation matrices of the source domain data. Note that $\| (\boldsymbol{\beta} \otimes \mathbf{w}) \|^2 = (\boldsymbol{\beta} \otimes \mathbf{w})^{\mathrm{T}}(\boldsymbol{\beta} \otimes \mathbf{w}) = 1$, according to the triangle inequality, the above formula can be further derived as follows,

$$\begin{aligned}\| \mathbf{Y}_\tau^{\mathrm{s}}\mathbf{c} - \mathbf{Z}_\tau^{\mathrm{s}}(\boldsymbol{\beta} \otimes \mathbf{w}) \| &\leq \| \mathbf{Y}_\tau^{\mathrm{s}}\mathbf{c}(\boldsymbol{\beta} \otimes \mathbf{w})^{\mathrm{T}} - \mathbf{Z}_\tau^{\mathrm{s}} \| \| (\boldsymbol{\beta} \otimes \mathbf{w}) \| \\ &= \| \mathbf{Y}_\tau^{\mathrm{s}}\mathbf{c}(\boldsymbol{\beta} \otimes \mathbf{w})^{\mathrm{T}} - \mathbf{Z}_\tau^{\mathrm{s}} \|\end{aligned} \tag{9}$$

DiPLS builds a dynamic model by maximizing the covariance of the score $\hat{u}_k$ of the inner model and the score u_k of the outer model, different from this method, this paper constructs a dynamic model by empirical risk minimization. Although it can be proved that the maximum variance and the minimum error are equivalent, we can obtain an upper bound on the learning error by formula (9), thereby effectively reducing the risk of overfitting.

4.3. Distribution alignment and structure preservation

In order to align the distribution of the source domain and the target domain and reduce the distribution difference, on the basis of the above empirical risk minimization, a distribution alignment regular term is introduced, the source domain and the target domain are aligned with the dynamic variance of the latent variable space pivot to reduce distribution difference on domain and target domain:

$$\begin{aligned}\mathcal{L}_{DA} = |&\frac{1}{n_s}(\boldsymbol{\beta} \otimes \mathbf{w})^{\mathrm{T}}\left(\mathbf{Z}_\tau^{\mathrm{s}}\right)^{\mathrm{T}}\mathbf{Z}_\tau^{\mathrm{s}}(\boldsymbol{\beta} \otimes \mathbf{w}) \\ &- \frac{1}{n_t}(\boldsymbol{\beta} \otimes \mathbf{w})^{\mathrm{T}}\left(\mathbf{Z}_\tau^{\mathrm{t}}\right)^{\mathrm{T}}\mathbf{Z}_\tau^{\mathrm{t}}(\boldsymbol{\beta} \otimes \mathbf{w}) |\end{aligned} \tag{10}$$

where $\mathbf{Z}_\tau^{\mathrm{s}}$ and $\mathbf{Z}_\tau^{\mathrm{t}}$ are the time-lagged data augmentation matrices on the source and target domains, n_s and n_t are the number of samples, respectively.

In addition, considering that the industrial process data is constrained by the physical mechanism, the value cannot be changed suddenly, such as the collected liquid level, pressure, temperature and other signals should remain relatively stable in adjacent sampling periods, and the realistic factors of oversampling in the process of process data acquisition, this paper proposes a smoothness hypothesis: the difference between the data of adjacent sample points in the latent variable space should be as small as possible. The above smoothness hypothesis constitutes another regular term in this paper, structure-preserving regular terms:

$$\mathcal{L}_{SP} = \| \dot{\mathbf{Z}}_\tau^{\mathrm{s}}(\boldsymbol{\beta} \otimes \mathbf{w}) \|_2^2 + \| \dot{\mathbf{Z}}_\tau^{\mathrm{t}}(\boldsymbol{\beta} \otimes \mathbf{w}) \|_2^2 \tag{11}$$

where, $\dot{\mathbf{Z}}_\tau^{\mathrm{s}}$ and $\dot{\mathbf{Z}}_\tau^{\mathrm{t}}$ are the first-order differences of the time-lagged data augmentation matrices on the source and target domains.

4.4. Structural risk minimization

The above distribution alignment regular term, structure preservation regular term and empirical distribution minimization function together constitute the structural risk minimization objective of this paper:

$$\begin{aligned}\min J = &\| \mathbf{Y}_\tau^{\mathrm{s}}\mathbf{c}(\boldsymbol{\beta} \otimes \mathbf{w})^{\mathrm{T}} - \mathbf{Z}_\tau^{\mathrm{s}} \|_{\mathrm{F}}^2 + \lambda\mathcal{L}_{DA} + \gamma\mathcal{L}_{SP} \\ &s.t. \quad \| \mathbf{c} \| = 1, \| \mathbf{w} \| = 1, \| \boldsymbol{\beta} \| = 1\end{aligned} \tag{12}$$

where λ and γ are tradeoff coefficients.

The overall architecture of the proposed approach is shown in Fig. 1.

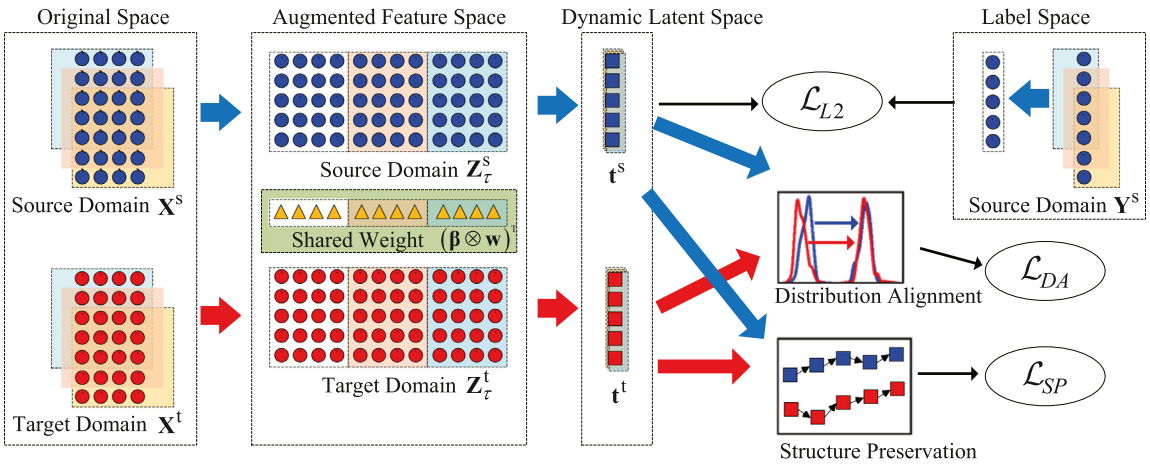


Fig. 1. The architecture of our method. The source domain data $\mathbf{X}^s$ and the target domain data $\mathbf{X}^t$ are augmented to form time-delay augmentation matrices $\mathbf{Z}_\tau^s$ and $\mathbf{Z}_\tau^t$, and then the augmented data are projected into their respective latent variable subspaces through the shared weight vector. The distribution alignment of data in different domains is achieved by aligning the variance of the latent variables, and the first-order difference of the augmented data in the source domain and the target domain is minimized to maintain the structure of the data. $\mathcal{L}_{DA}$ and $\mathcal{L}_{SP}$ are the losses of domain adaption and structure preservation, respectively.

Eq. (12) define an optimization problem for a multiple-input multiple-output system, for a multiple-input single-output system, $\mathbf{c} = 1$, let $\mathbf{Y}_\tau^s = \mathbf{y}_\tau^s \in \mathrm{R}^{n\times 1}$, $\mathbf{D} = \frac{1}{n_s}\left(\mathbf{Z}_\tau^s\right)^{\mathrm{T}}\mathbf{Z}_\tau^s - \frac{1}{n_t}\left(\mathbf{Z}_\tau^t\right)^{\mathrm{T}}\mathbf{Z}_\tau^t$, $\boldsymbol{\Gamma} = \left(\dot{\mathbf{Z}}_\tau^s\right)^{\mathrm{T}}\dot{\mathbf{Z}}_\tau^s + \left(\dot{\mathbf{Z}}_\tau^t\right)^{\mathrm{T}}\dot{\mathbf{Z}}_\tau^t$, Eq. (12) can be rewritten as:

$$\begin{aligned}\min J =& \| \mathbf{y}_\tau^s(\boldsymbol{\beta}\otimes\mathbf{w})^{\mathrm{T}} - \mathbf{Z}_\tau^s \|_{\mathrm{F}}^2 + \lambda|(\boldsymbol{\beta}\otimes\mathbf{w})^{\mathrm{T}}\mathbf{D}(\boldsymbol{\beta}\otimes\mathbf{w})| \\ &+\gamma((\boldsymbol{\beta}\otimes\mathbf{w})^{\mathrm{T}}\boldsymbol{\Gamma}(\boldsymbol{\beta}\otimes\mathbf{w})) \\ s.t.\quad & \| \mathbf{w} \| = 1, \| \boldsymbol{\beta} \| = 1\end{aligned} \tag{13}$$

due to $\mathbf{D}^{\mathrm{T}} = \mathbf{D}$, eigenvalue decomposition can be performed on it, $\mathbf{D} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}$, $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1\lambda_2\cdots\lambda_k)$, let $\mathbf{v} = [v_1 v_2\cdots v_k] = (\boldsymbol{\beta}\otimes\mathbf{w})^{\mathrm{T}}\mathbf{V}$,

$$\begin{aligned}|(\boldsymbol{\beta}\otimes\mathbf{w})^{\mathrm{T}}\mathbf{D}(\boldsymbol{\beta}\otimes\mathbf{w})| &= |(\boldsymbol{\beta}\otimes\mathbf{w})^{\mathrm{T}}\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}(\boldsymbol{\beta}\otimes\mathbf{w})| \\ &= |v_1^2\lambda_1 + v_2^2\lambda_2 + \cdots + v_k^2\lambda_k| \\ &\le |v_1^2\lambda_1| + |v_2^2\lambda_2| + \cdots + |v_k^2\lambda_k| \\ &= v_1^2|\lambda_1| + v_2^2|\lambda_2| + \cdots + v_k^2|\lambda_k| \\ &= (\boldsymbol{\beta}\otimes\mathbf{w})^{\mathrm{T}}\mathbf{V}|\boldsymbol{\Lambda}|\mathbf{V}^{\mathrm{T}}(\boldsymbol{\beta}\otimes\mathbf{w})\end{aligned} \tag{14}$$

where $|\boldsymbol{\Lambda}| = \mathrm{diag}(|\lambda|_1|\lambda_2|\cdots|\lambda_k|)$, let $\mathbf{D}' = \mathbf{V}|\boldsymbol{\Lambda}|\mathbf{V}^{\mathrm{T}}$, from Eqs. (13) and (14), an upper bound of empirical risk can be further determined, and the objective is:

$$\begin{aligned}\min J =& \| \mathbf{y}_\tau^s(\boldsymbol{\beta}\otimes\mathbf{w})^{\mathrm{T}} - \mathbf{Z}_\tau^s \|_{\mathrm{F}}^2 + (\boldsymbol{\beta}\otimes\mathbf{w})^{\mathrm{T}}(\lambda\mathbf{D}' + \gamma\boldsymbol{\Gamma})(\boldsymbol{\beta}\otimes\mathbf{w}) \\ s.t.\quad & \| \mathbf{w} \| = 1, \| \boldsymbol{\beta} \| = 1\end{aligned} \tag{15}$$

where $(\boldsymbol{\beta}\otimes\mathbf{w}) = (\boldsymbol{\beta}\otimes\mathbf{I}_{\mathrm{m}})\mathbf{w} = (\mathbf{I}_\tau\otimes\mathbf{w})\boldsymbol{\beta}$.

It can be seen from Eq. (15) that the objective function to be solved is a convex function, and the closed-form solution can be obtained by the Lagrange multiplier method as:

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} =& \left[(\boldsymbol{\beta}\otimes\mathbf{I}_{\mathrm{m}})^{\mathrm{T}}(\mathbf{I} + \frac{\lambda\mathbf{D}' + \gamma\boldsymbol{\Gamma}}{\left(\mathbf{y}_\tau^s\right)^{\mathrm{T}}\mathbf{y}_\tau^s})(\boldsymbol{\beta}\otimes\mathbf{I}_{\mathrm{m}})\right]^{-1} \\ &\times \frac{(\boldsymbol{\beta}\otimes\mathbf{I}_{\mathrm{m}})^{\mathrm{T}}\left(\mathbf{Z}_\tau^s\right)^{\mathrm{T}}\mathbf{y}_\tau^s}{\left(\mathbf{y}_\tau^s\right)^{\mathrm{T}}\mathbf{y}_\tau^s}\end{aligned} \tag{16}$$

$$\begin{aligned}\frac{\partial J}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \boldsymbol{\beta} =& \left[(\mathbf{I}_\tau\otimes\mathbf{w})^{\mathrm{T}}(\mathbf{I} + \frac{\lambda\mathbf{D}' + \gamma\boldsymbol{\Gamma}}{\left(\mathbf{y}_\tau^s\right)^{\mathrm{T}}\mathbf{y}_\tau^s})(\mathbf{I}_\tau\otimes\mathbf{w})\right]^{-1} \\ &\times \frac{(\mathbf{I}_\tau\otimes\mathbf{w})^{\mathrm{T}}\left(\mathbf{Z}_\tau^s\right)^{\mathrm{T}}\mathbf{y}_\tau^s}{\left(\mathbf{y}_\tau^s\right)^{\mathrm{T}}\mathbf{y}_\tau^s}\end{aligned} \tag{17}$$

By solving the above formula, the weighting vector $\mathbf{w}$ can be obtained. The scores of source domain can be computed by $\mathbf{t}^s = [t_0^s, t_1^s, \ldots, t_{N+\tau}^s]^{\mathrm{T}} = \mathbf{X}^s\mathbf{w}$. Let $\mathbf{t}_i^s = [t_i^s, t_{i+1}^s, \ldots, t_{N+i}^s]^{\mathrm{T}}$, we can construct the time-lagged score matrix $\mathbf{T}_\tau^s = [\mathbf{t}_\tau^s, \mathbf{t}_{\tau-1}^s, \ldots, \mathbf{t}_0^s]$. The scores of target domain can be computed by $\mathbf{t}^t = \mathbf{X}^t\mathbf{w}$.

The inner model estimation $\hat{\mathbf{u}}_\tau^s = \alpha_0\mathbf{t}_\tau^s + \alpha_1\mathbf{t}_{\tau-1}^s + \cdots + \alpha_\tau\mathbf{t}_0^s = \mathbf{T}_\tau^s\boldsymbol{\alpha}$, we can get the autoregressive coefficients vector $\boldsymbol{\alpha} = \left[\left(\mathbf{T}_\tau^s\right)^{\mathrm{T}}\mathbf{T}_\tau^s\right]^{-1}\left(\mathbf{T}_\tau^s\right)^{\mathrm{T}}\mathbf{y}_\tau^s$.

The loading vectors of source and target domains can be derived as

$\mathbf{p}^s = (\mathbf{X}^s)^{\mathrm{T}}\mathbf{t}^s/(\mathbf{t}^s)^{\mathrm{T}}\mathbf{t}^s$, $\mathbf{p}^t = \left(\mathbf{X}^t\right)^{\mathrm{T}}\mathbf{t}^t/\left(\mathbf{t}^t\right)^{\mathrm{T}}\mathbf{t}^t$.

Deflation of $\mathbf{X}^s$, $\mathbf{X}^t$ and $\mathbf{y}_\tau^s$ are performed as

$\mathbf{E}^s = \mathbf{X}^s - \mathbf{t}^s(\mathbf{p}^s)^{\mathrm{T}} \Rightarrow \mathbf{X}^s$, $\mathbf{E}^t = \mathbf{X}^t - \mathbf{t}^t\left(\mathbf{p}^t\right)^{\mathrm{T}} \Rightarrow \mathbf{X}^t$, $\mathbf{F}^s = \mathbf{y}_\tau^s - \hat{\mathbf{u}}_\tau^s \Rightarrow \mathbf{y}_\tau^s$.

Repeat the above steps until the required A latent variables are extracted, and we get the weighting matrix $\mathbf{W} = [\mathbf{W}, \mathbf{w}]$, the loading matrix $\mathbf{P} = [\mathbf{P}, \mathbf{p}^s]$, the regression matrix $\mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}$ and autoregressive coefficients matrix $\mathbf{B} = [\mathbf{B}, \boldsymbol{\alpha}]$.

The predict label for samples $\mathbf{x}_k^t$ in the target domain is calculated as

$$\mathbf{y}_k^t = \mathrm{Sum}([\mathbf{x}_k^t, \mathbf{x}_{k-1}^t, \ldots, \mathbf{x}_{k-\tau}^t]^{\mathrm{T}}R \odot B) \tag{18}$$

where $\odot$ is the Hadamard product, Sum is the matrix element-wise sum operator.

The steps of DTPLS algorithm are described in Algorithm 1.

4.5. Hyperparameter selection

The DTPLS algorithm has 4 hyperparameters: the latent variable dimension A, the time-lagged coefficient τ, the structure-preserving regular item tradeoff coefficient γ, and the domain adaptation regular item tradeoff coefficient λ. It can be seen from the algorithm that when γ and λ are both set to zero, the algorithm falls back to the DiPLS algorithm, When τ, γ and λ are all set to zero, the algorithm falls back to the PLS algorithm. In order to reduce the amount of calculation, the selection of hyperparameters in this paper adopts a two-step cross-validation method. First, set γ and λ to 0, and obtain the latent variable dimension A and time-lagged coefficient τ through cross-validation. Secondly, according to the obtained latent variable dimension A and time-lagged coefficient τ, use cross-validation again to obtain structure-preserving regular tradeoff coefficient γ and domain

Table 1
Comparison methods used in the paper.

Method	NIPALS	DIPALS	DiPLS	TCA+DiPLS	CORAL+DiPLS	DTPLS
Dynamic or static	Static	Static	Dynamic	Dynamic	Dynamic	Dynamic
Distribution alignment	No	Yes	No	Yes	Yes	Yes

Algorithm 1 DTPLS algorithm

Input: Source domain data $\mathbf{X}^s$, $\mathbf{y}^s$, target domain data $\mathbf{X}^t$, latent variable dimension A, time-lagged coefficient τ, tradeoff coefficient γ and λ
Output: Regression matrix $\mathbf{R}$, autoregressive coefficients matrix $\mathbf{B}$, target domain label $\mathbf{y}^t$
(**Initialization**):Data normalization and initialize $\boldsymbol{\beta} = [1, 0 \cdots, 0]^{\mathrm{T}}$
for i *in* $[1, A]$ **do**
  while $\boldsymbol{\beta}$ not converged **do**
    (**Projection**):
    Calculate the input weight vector $\mathbf{w}$ by formula (16) and normalize $\mathbf{w} = \mathbf{w}/\|\mathbf{w}\|$
    Calculate the autoregressive coefficient $\boldsymbol{\beta}$ by formula (17) and normalize $\boldsymbol{\beta} = \boldsymbol{\beta}/\|\boldsymbol{\beta}\|$
  end while
  (**Regression**):
  Calculate the score vectors for $\mathbf{X}^s$ and $\mathbf{X}^t$: $\mathbf{t}^s = \mathbf{X}^s\mathbf{w}$, $\mathbf{t}^t = \mathbf{X}^t\mathbf{w}$
  Constructing the time-lagged score matrix $\mathbf{T}_\tau^s = [\mathbf{t}_\tau^s, \mathbf{t}_{\tau-1}^s \cdots, \mathbf{t}_0^s]$
  Calculate autoregressive coefficients vector $\boldsymbol{\alpha} = \left(\left(\mathbf{T}_\tau^s\right)^{\mathrm{T}}\mathbf{T}_\tau^s\right)^{-1}\left(\mathbf{T}_\tau^s\right)^{\mathrm{T}}\mathbf{y}_\tau^s$
  (**Deflation**):
  Calculate inner model estimates $\mathbf{0}_\tau^s = \mathbf{T}_\tau^s\boldsymbol{\alpha}$
  Calculate the loading vectors for $\mathbf{X}^s$ and $\mathbf{X}^t$: $\mathbf{p}^s = \left(\mathbf{X}^s\right)^{\mathrm{T}}\mathbf{t}^s/\left(\mathbf{t}^s\right)^{\mathrm{T}}\mathbf{t}^s$, $\mathbf{p}^t = \left(\mathbf{X}^t\right)^{\mathrm{T}}\mathbf{t}^t/\left(\mathbf{t}^t\right)^{\mathrm{T}}\mathbf{t}^t$
  Calculate the residual matrix $\mathbf{E}^s = \mathbf{X}^s - \mathbf{t}^s\left(\mathbf{p}^s\right)^{\mathrm{T}} \Rightarrow \mathbf{X}^s$, $\mathbf{E}^t = \mathbf{X}^t - \mathbf{t}^t\left(\mathbf{p}^t\right)^{\mathrm{T}} \Rightarrow \mathbf{X}^t$, $\mathbf{F}^s = \mathbf{y}_\tau^s - \mathbf{0}_\tau^s \Rightarrow \mathbf{y}_\tau^s$
  $\mathbf{P} = [\mathbf{P}, \mathbf{p}^s]$, $\mathbf{W} = [\mathbf{W}, \mathbf{w}]$, $\mathbf{B} = [\mathbf{B}, \boldsymbol{\alpha}]$
end for
$\mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}$,
Calculate target domain label $\mathbf{y}^t$ by formula (18)

adaptation regular tradeoff coefficient λ. Since the target domain data is unlabeled, we use the labeled source domain data for cross-validation.

It is worth noting that there are many ways to optimize hyperparameters, such as random search, grid search, Bayesian optimization, heuristic algorithms, etc. The cross-validation method chosen here is to prove the effectiveness of the method proposed in the paper, and other more effective parameter optimization methods will be developed in the follow-up research.

5. Experiments

To verify the effectiveness of the proposed method, we conducted experiment on three publicly available industrial process datasets, the Tennessee Eastman (TE), the Debutanizer, and the Sulfur Recovery Unit (SRU) datasets, respectively. The comparison methods include the Non-linear Iterative Partial Least Squares (NIPALS) algorithm in [45], Domin-invariant Iterative Partial Least Squares (DIPALS) algorithm in [41], Transfer component analysis (TCA) in [47], Correlation alignment (CORAL) [48] and Dynamic inner PLS (DiPLS) algorithm in [22]. Among them, NIPALS is a traditional soft sensor modeling method, DiPLS is a classical dynamic latent variable modeling method, NIPALS and DiPLS are used as the baseline. The rest are soft sensor methods based on transfer learning. DIPALS aligns second-order statistics in the framework of nonlinear iterative least squares. TCA performs marginal distribution alignment and CORAL performs second-order subspace alignment. TCA and CORAL are unsupervised alignment methods, in order to consider the distribution differences and dynamic characteristics of the system at the same time, we first use the two methods for distribution alignment, and then use DiPLS to build a regression model on the aligned data. We conclude the comparison methods in Table 1.

Table 2
Prediction error of different methods on TEP dataset.

Method	NIPALS	DIPALS	DiPLS	TCA+DiPLS	CORAL+DiPLS	DTPLS
MSE	1.032	1.053	0.835	0.815	0.886	**0.667**
MAE	0.803	0.817	0.737	0.731	0.755	**0.655**
R2	0.296	0.282	0.431	0.445	0.396	**0.545**

5.1. TEP data set

TE dataset: The TE process dataset was created by Eastman Chemicals, a chemical model simulation platform based on real industrial processes [49]. The TE process consists of five main units: reactor, condenser, compressor, separator and stripper, with a total of 22 continuous process variables XMEAS(1–22), 12 manipulated variables XMV(1–12), 19 component variables XMEAS(23–41). Select 22 continuous process variables XMEAS(1–22) and 12 manipulated variables XMV(1–12) as input variables, component variables XMEAS(38) as output variables, 500 data as training set (source domain), 960 data is used as the test set (target domain), and the data is pre-normalized.

First, the baseline model DiPLS latent variable dimension $A = 5$, time-lagged coefficient $\tau = 3$ is obtained by cross-validation. In order to make a fair comparison and eliminate the influence of the latent variable dimension and the time-lagged coefficient on the model, fix the latent variable dimension A and the time-lagged coefficient τ, and obtain the domain adaptation regular tradeoff coefficient λ and structure-preserving regular tradeoff coefficient γ through cross-validation again. The final predictions of NIPALS, DIPALS, DiPLS, TCA, CORAL and our method (DTPLS) under TEP data are shown in Fig. 2. Because of the lower sampling rate, there is only one measured value out of every five data points for the component variables XMEAS(38), the rest are artificial, so only 1/5 of samples are compared. For the NIPALS algorithm, the dimension of the latent variable is $A = 5$. For the DIPALS algorithm, the dimension of the latent variable is $A = 5$, the regularization parameter $\lambda = 2670$. For the DiPLS algorithm, the dimension of the latent variable is $A = 5$, the delay coefficient $\tau = 3$. For the DTPLS algorithm, the dimension of the latent variable is $A = 5$, the time-lagged coefficient $\tau = 3$, $\lambda = 780$, $\gamma = 430$. For TCA and CORAL, the parameters are the same as those of DiPLS, we only perform spatial transformation without dimensionality reduction. The predicted error indicators are shown in Table 2.

5.2. Debutanizer data set

The debutanizer dataset [50] comes from a real industrial continuous distillation process, which contains a total of 7 input variables, respectively x1 top temperature, x2 top pressure, x3 reflux flow, x4 flow to next process, x5 VI tray temperature, x6 bottom temperature 1, x7 bottom temperature 2, 1 output variable is the bottoms butane concentration. The entire dataset has a total of 2394 samples. This paper selects the first 1100 samples as the training set (source domain) and the rest 1394 samples as the test set (target domain). The final predictions is shown in Fig. 3. Due to the lower sampling rate, there is only one measured value out of every ten data points for the output variable, the

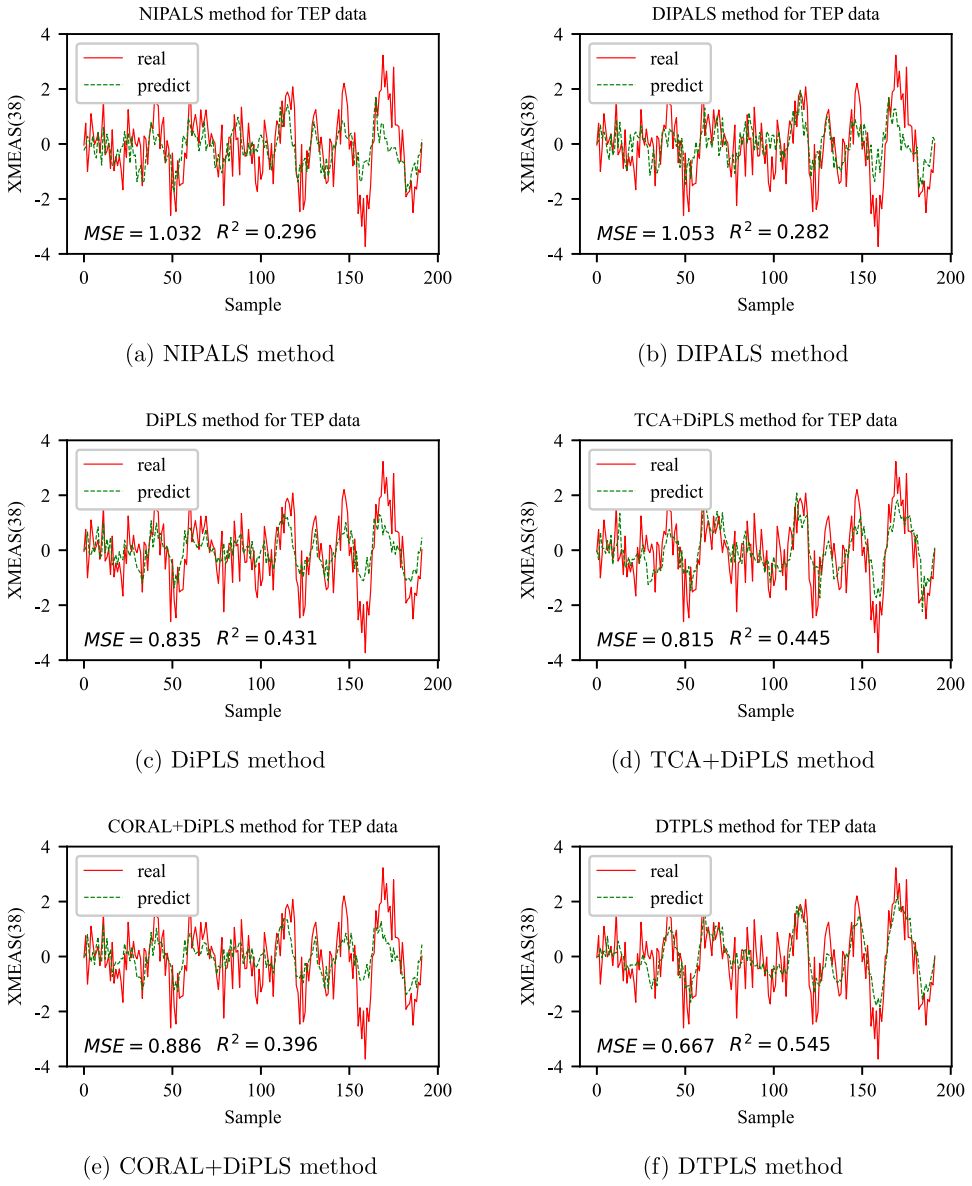


Fig. 2. Prediction results of different methods on the TEP dataset.

Table 3
Prediction errors of different methods on the Debutanizer dataset.

Method	NIPALS	DIPALS	DiPLS	TCA+DiPLS	CORAL+DiPLS	DTPLS
MSE	0.034	0.032	0.019	0.035	0.019	**0.012**
MAE	0.146	0.135	0.010	0.122	0.100	**0.074**
R2	−0.057	0.004	0.401	−0.093	0.400	**0.626**

rest are artificial, so only 1/10 of samples are compared. For the NIPALS algorithm, the dimension of the latent variable is $A = 6$. For the DIPALS algorithm, the dimension of the latent variable is $A = 6$, the regularization parameter $\lambda = 420$. For the DiPLS algorithm, the dimension of the latent variable is $A = 6$, the time-lagged coefficient $\tau = 19$. For the DTPLS algorithm, the dimension of the hidden variable is $A = 6$, the delay coefficient $\tau = 19$, $\lambda = 480$, $\gamma = 5$. For TCA and CORAL, the parameters are the same as DiPLS, and we also only perform spatial transformation without dimensionality reduction. The predicted error indicators are shown in Table 3.

5.3. SRU data set

The sulfur recovery unit dataset [51] is the upstream unit of the distillation column dataset, and the data contains 5 input variables and 2 output variables, input variables are x1 MEA gas flow, x2 first air flow, x3 s air flow, x4 SWS zone gas flow, x5 SWS zone air flow, the output variables is H2S and SO2 concentration. The entire data set contains 10 081 data samples. In this paper, the first 5000 samples are used as the training set (source domain), and the remaining samples are used as the test set (target domain) to predict the SO2 concentration. The final predictions is shown in Fig. 4. Due to the lower sampling rate, there is only one measured value out of every ten data points for the output variable, the rest are artificial, so only 1/10 of samples are compared. For the NIPALS algorithm, the dimension of the latent variable is $A = 3$. For the DIPALS algorithm, the dimension of the latent variable is $A = 3$, the regularization parameter $\lambda = 67700$. For the DiPLS algorithm, the dimension of the latent variable is $A = 3$, the delay coefficient $\tau = 13$. For TCA and CORAL, the parameters are the same as those of DiPLS. For the DTPLS algorithm, the dimension of the hidden variable is $A = 3$,

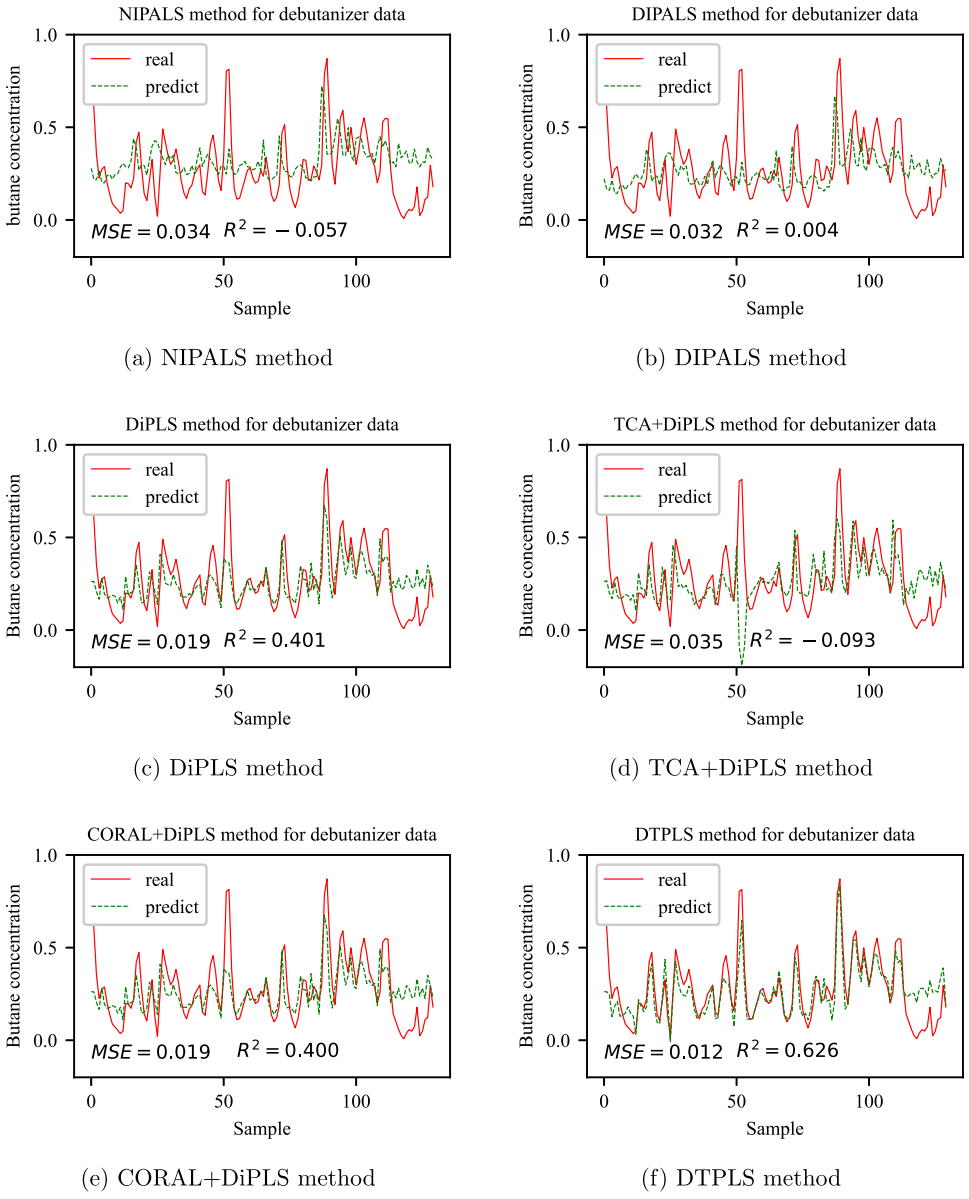


Fig. 3. Prediction results of different methods on the Debutanizer dataset.

Table 4
Prediction errors of different methods on the SRU dataset.

Method	NIPALS	DIPALS	DiPLS	TCA+DiPLS	CORAL+DiPLS	DTPLS
MSE	2.52×10^{-3}	2.46×10^{-3}	1.42×10^{-3}	2.33×10^{-3}	1.44×10^{-3}	$\mathbf{0.59 \times 10^{-3}}$
MAE	0.037	0.037	0.029	0.037	0.029	**0.018**
R2	−0.007	0.015	0.433	0.070	0.425	**0.766**

the delay coefficient $\tau = 13$, $\lambda = 1900$, $\gamma = 700$. The predicted error indicators are shown in Table 4.

5.4. Experimental results

Through the comparative experiments on the above three industrial process data sets, it can be seen that for the dynamic industrial process data, the traditional static modeling method (NIPALS) has limited modeling accuracy, while the domain adaptation method based on the static model (DIPALS) cannot further improve the accuracy of the model. The dynamic modeling method (DiPLS) and the dynamic transfer method (DTPLS) based on the dynamic model can effectively improve the prediction accuracy of the model. This is because the static transfer method (DIPALS) only aligns the latent variable variance of the feature space. When the concept drift occurs in the process, the latent variable method that only aligns the feature space cannot directly reflect the drift characteristics of labels. It is worth noting that although TCA and CORAL are also modeling methods based on dynamic transfer, they have not been able to further improve the prediction accuracy, and even the phenomenon of negative transfer has occurred. This is because TCA and CORAL do not consider the mapping relationship between features and labels

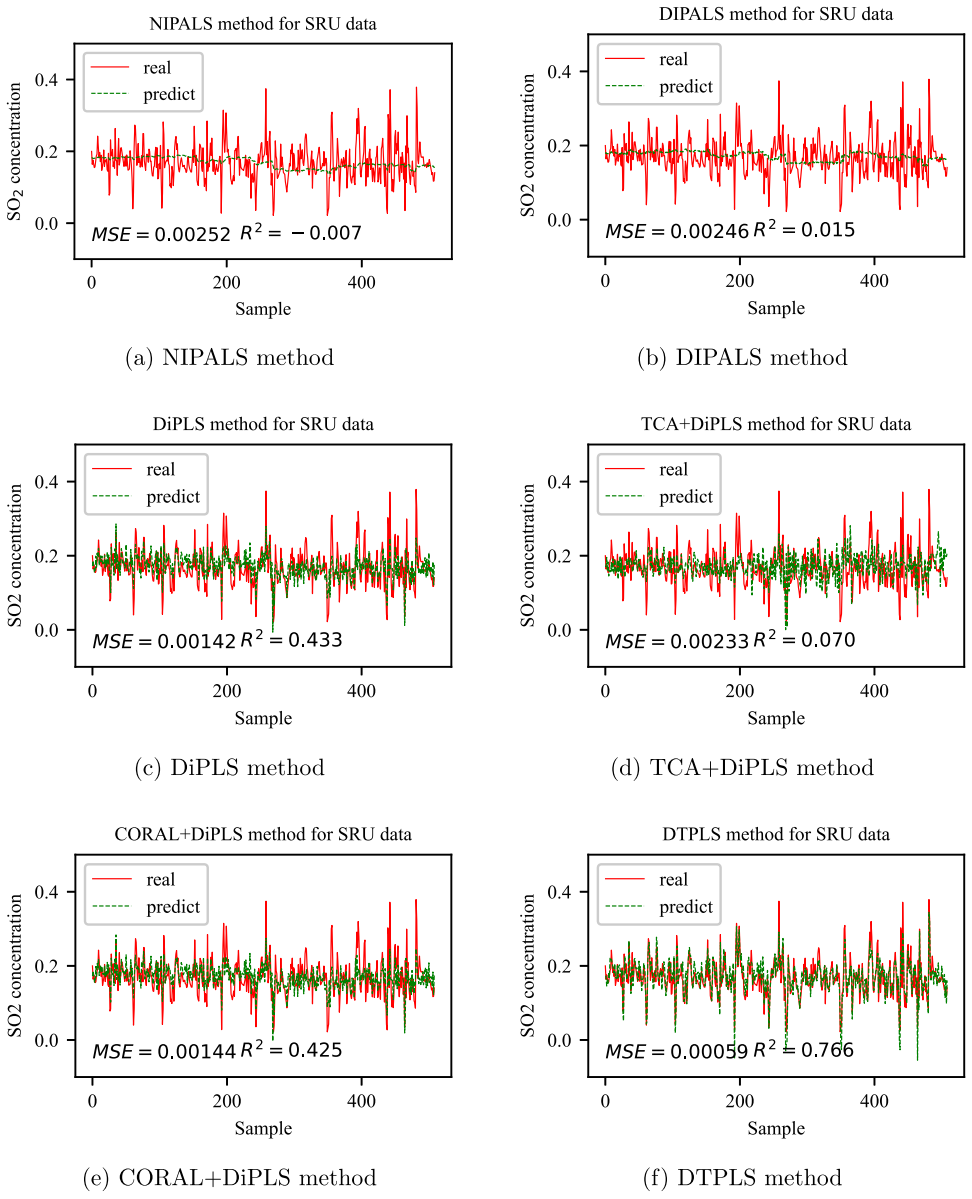


Fig. 4. Prediction results of different methods on the SRU dataset.

during the transfer process, and the transfer process may destroy the mapping structure of features and labels. By contrast, the dynamic transfer method (DTPLS), on the basis of dynamic modeling, aligns the variance of the latent variables in the label space, and introduces the assumption of smoothness to maintain the structure of the latent variables in the label space, which can effectively deal with the problem of label concept drift.

5.5. Hyperparameter analysis

The dimension of the latent variable space is an important parameter that constitutes the latent variable space, and the latent variable dimension ($1 \leq A \leq$ m) is generally obtained through cross-validation. In order to observe the influence of the latent variable dimension on the final prediction effect, different dimensions were selected for experiments on different data sets. Fig. 5 shows the variation of the final predicted mean square error (MSE) with the dimension of the latent variable. It can be seen from this figure that with the increase of the dimension of the latent variable, the mean square error generally shows a downward trend. When the dimension increases to a certain value, the mean square error no longer decreases, forming the so-called "elbow point". This is because when the latent variable dimension is low, the model residual contains useful knowledge related to labels. As the dimension gradually increases, the useful knowledge learned by the model also increases gradually, when the residual only contains noise, increasing the dimension at this time will not only fail to improve the performance of the model, but will tend to deteriorate the model. This feature is consistent with the latent variable characteristics of PCA and PLS algorithms, further indicating that the dynamic augmentation of data and dynamic variance alignment do not affect the spatial structure of latent variables.

The time-lagged coefficient is another important parameter of the algorithm in this paper. For the actual industrial process, the inherent time-lagged coefficient of the system can be obtained by the method of system identification. For the data-driven modeling method, only the data is obtained, so the time-lagged coefficient can only be obtained by the method of cross-validation. In view of the influence of the time-lagged coefficient on the system prediction error, different time-lagged coefficients between

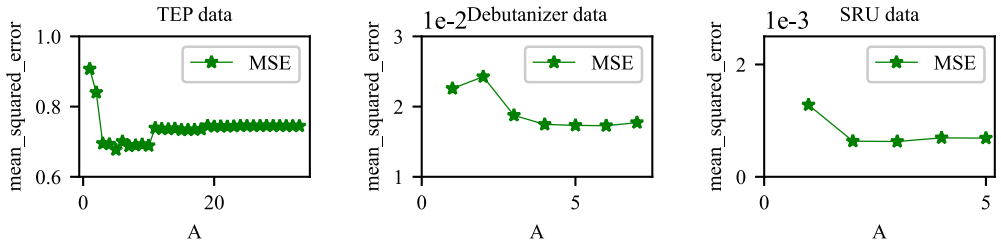


Fig. 5. The influence of latent variable space dimension on prediction results.

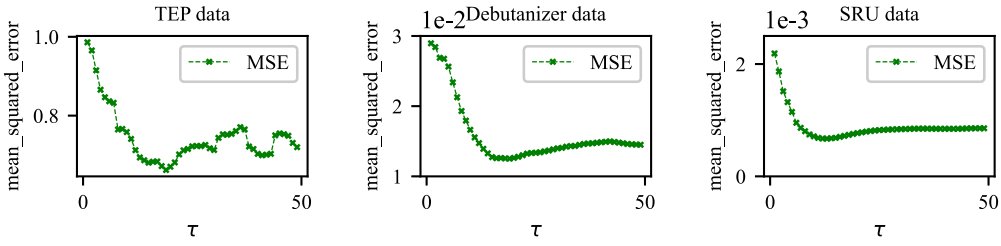


Fig. 6. The influence of the time-lagged coefficient on the forecast results.

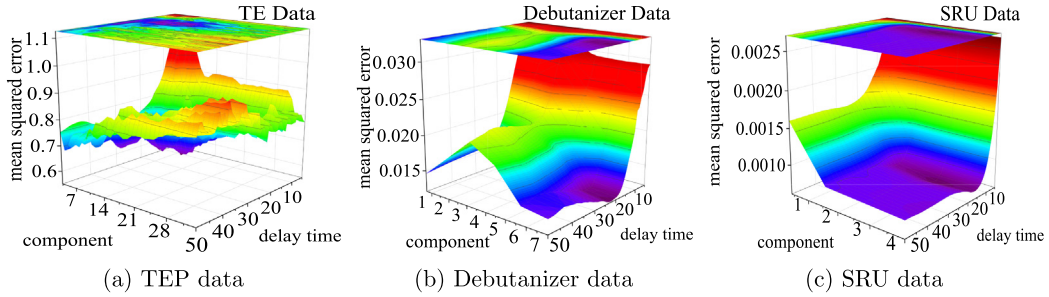


Fig. 7. The influence of latent variable space dimension and time-lagged coefficient on prediction results .

0 and 50 are selected, and the curve of the mean square error changing with the time-lagged coefficient is obtained as shown in Fig. 6. It can be seen from this figure that when the selected time-lagged coefficient is smaller than the inherent time delay coefficient of the system, the prediction error of the model gradually decreases with the increase of the time-lagged coefficient. When the time-lagged coefficient is larger than the inherent time-lagged coefficient of the system, the model prediction error starts to increase. A large time-lagged coefficient will lead to a multiple expansion of the data, increasing the amount of data operations, while a large time-lagged coefficient will lead to a smaller training error, which brings the risk of overfitting.

When the latent variable dimension and the delay time change at the same time, we can further obtain the 3D mapping surface graph of the mean square error with the change of these two parameters, as shown in Fig. 7, the dark part of the figure is the minimum point of the mean square error. It can be seen from this figure that with the increase of the latent variable dimension and the delay time, the mean square error shows a downward trend. When it exceeds a certain threshold, the mean square error begins to rise, which is consistent with the two-dimensional analysis process. In addition, it can be seen from the 3D surface map that on the SRU dataset, the overall model error function is smooth and convex, and the system has a global optimal solution.

In order to verify the effectiveness of the domain adaptation regularization term proposed in this paper, the latent variable dimension and the time-lagged coefficient are fixed on the three data sets respectively, and structure-preserving regular tradeoff coefficient is set to 0. The selection range of the domain adaptation regular term is: [0, 20000], and the curve of the obtained predicted label mean square error with the parameter of the domain adaptation regular term is shown in Fig. 8. It can be seen from this figure that for the TEP dataset and the Debutanizer dataset, the mean square error of prediction can be effectively reduced by giving a larger domain adaptation regular parameter. For the SRU dataset, an appropriate domain-adaptive regular parameter must be selected, and an excessively large domain-adaptive regular parameter will cause the model performance to deteriorate, resulting in negative transfer.

At the same time, in order to verify the effectiveness of the structure-maintaining regular term proposed in this paper, the dimension of the latent variable and the time-lagged coefficient are fixed, and the domain adaptation regular tradeoff coefficient is set to 0. The selection range of the structure-preserving regular term is: [0, 20000]. The obtained curve of the predicted label mean square error with the structure-maintaining regularization parameter is shown in Fig. 9. It can be seen from this figure that by giving a larger structure-maintaining regularization parameter, the mean square error of the prediction can be effectively reduced, and when the value of the regularization parameter exceeds a certain threshold, its numerical value has little influence on the prediction result, so that the algorithm is not sensitive to the selection of this parameter.

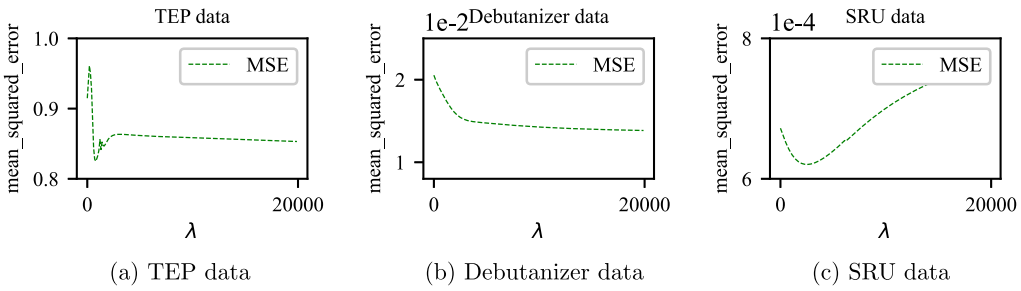


Fig. 8. The influence of the domain adaptation regular term on the prediction results .

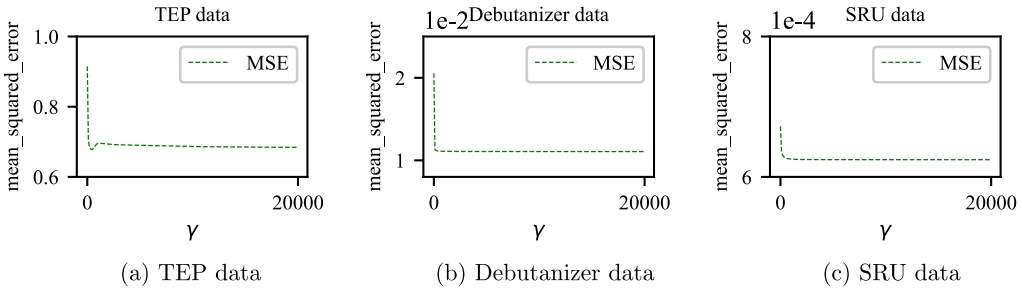


Fig. 9. The influence of structure-preserving regularization term on prediction results .

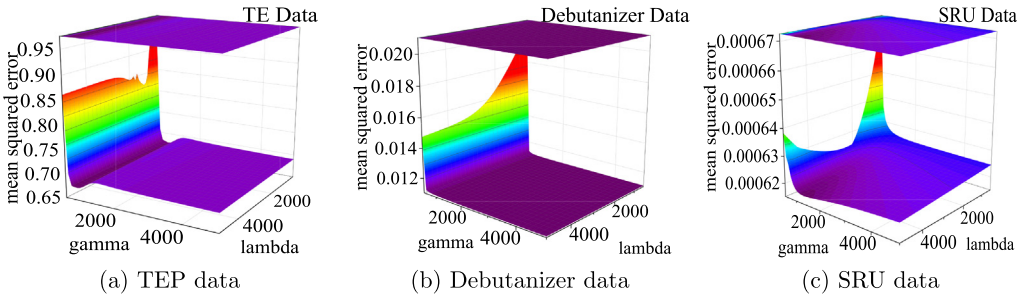


Fig. 10. The influence of structure-preserving coefficient and the domain adaptation coefficient on prediction results.

When the structure-preserving regular coefficient and the domain adaptation regular coefficient change at the same time, we can further obtain the 3D mapping surface graph of the mean square error with the change of these two parameters, as shown in Fig. 10. We can see that when both parameters are small, the domain adaptation regularization parameter plays a dominant role, and when both parameters are large, the structure-preserving parameter plays the dominant role. However, the overall trends tend to be flat, indicating that forecast errors are not sensitive to these two parameters.

6. Theoretical analysis and experimental verification

6.1. Theoretical analysis

Most transfer learning theories stem from Ben-David's pioneering work [52], for a binary classification problem, the target domain error can be bounded by the following three terms:

$$\varepsilon_T(h) \le \varepsilon_S(h) + d_1(D_S, D_T) + \min\left\{E_{D_S}\left[|f_S(x) - f_T(x)|\right], E_{D_T}\left[|f_S(x) - f_T(x)|\right]\right\} \tag{19}$$

where $\varepsilon_T(h)$ is the target error; the first term $\varepsilon_S(h)$ is the source error, which a training algorithm might seek to minimize; the second term $d_1(D_S, D_T)$ is the Total Variation Distance which often used to express the difference between two distributions; the third is the difference in labeling functions across the two domains, which is expected to be small. In above bound, the distribution distance is the main problem.

Nikzad-Langerodi et al. [41] further extended the above theory to take into account the latent variable space, proved that the upper bound of the error is:

$$\begin{aligned} &E_{X\sim Q}\left[|\hat{h} - h|\right] \le E_{X\sim P}\left[|\hat{h} - h|\right] + \sqrt{2D(N_{\tilde{P}} \parallel N_{\tilde{Q}})} + \eta(g) + \sqrt{8\zeta} \\ &\eta(g) = \inf(E_P\left[|f \circ g - h|\right] + E_Q\left[|f \circ g - h|\right]) \\ &\zeta = \max\{D(N_{\tilde{P}} \parallel \tilde{P}), D(N_{\tilde{Q}} \parallel \tilde{Q})\} \end{aligned} \tag{20}$$

where, $\hat{h} = f \circ g$, g is the mapping function from feature space to latent variable space, $g : \mathbf{X} \in \mathrm{R}^{nxm} \to \mathbf{T} \in \mathrm{R}^{nxA}$, f is the mapping function from the latent variable space to the label space, $f : \mathbf{T} \in \mathrm{R}^{nxA} \to \mathbf{Y} \in \mathrm{R}^{nxd}$, $\circ$ is the compound operation of the function. $\tilde{P}$ and $\tilde{Q}$ are the corresponding distributions of latent variables in the source and target domains, $N_{\tilde{P}}$ and $N_{\tilde{Q}}$ are the normal distribution approximation of distribution $\tilde{P}$ and distribution $\tilde{Q}$, $D(\cdot \parallel \cdot)$ is the corresponding KL divergence. The above content can be visually represented by Fig. 11.

The formula (20) shows that the upper bound of the target domain error $E_{X\sim Q}\left[|\hat{h} - h|\right]$ can be determined by the source domain error $E_{X\sim P}\left[|\hat{h} - h|\right]$, the distribution difference

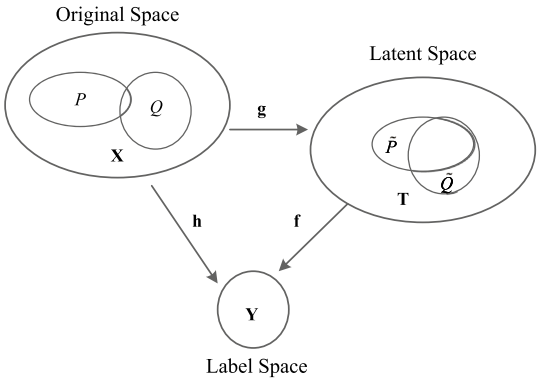


Fig. 11. Latent space representation.

$\sqrt{2D(N_{\tilde{P}} \parallel N_{\tilde{Q}})}$ between the source domain and the target domain, the mapping error $\eta(g)$ and the distribution approximation error ζ.

For dynamic systems with distribution differences, the dynamic modeling method presents higher modeling accuracy than the static modeling method, and the dynamic modeling method can effectively fit the data, so the source domain error $E_{X\sim P}\left[|\hat{h}-h|\right]$ and the mapping error $\eta(g)$ would be small.

According to the law of large numbers, when the sample size of the source domain and the target domain is large enough, the distribution can be effectively approximated by the normal distribution, so the distribution approximation error ζ in the latent space would be small too. Therefore, reducing the distribution difference between the source domain and the target domain becomes an effective means to reduce the error of the target domain.

However, reducing the distribution difference will have an impact on the source domain error. For some domain adaptation models, such as pre-train fine-tuning models, the data are first trained on the source domain, and then the models are fine-tuned on the target domain. This type of model reduces the distribution difference at the expense of increasing the error in the source domain. In that situation, source domain error and distribution difference cannot be analyzed separately as they are related to each other.

In contrast, our model introduces distribution alignment regularization and structure preserving regularization in the framework of structural risk minimization, which is to optimize the objective $J = \parallel \mathbf{Y}_\tau^s \mathbf{c}(\boldsymbol{\beta} \otimes \mathbf{w})^{\mathrm{T}} - \mathbf{Z}_\tau^s \parallel_{\mathrm{F}}^2 + \lambda\mathcal{L}_{DA} + \gamma\mathcal{L}_{SP}$, where the first term is to minimize the source domain empirical error, the second term is to penalize data with large distribution differences, the third term is to penalize unsmoothed data(such as mutation data, outliers). We achieve distribution alignment by aligning the variances of the source and target domains in the latent variable space, this approach is to learn domain-invariant representations of the source and target domains data. This domain-invariant feature is a direct 'cause' of the dependent variable. The introduction of data from different distributions will help to learn the domain-invariant features, thus improving model prediction accuracy and making the model more robust.

To verify the above analysis, we add error experiments on source domain. The experimental results show that when the distribution alignment regularization and structure preservation regularization are introduced, the error on the source domain can be effectively reduced. The experimental results are presented in the next subsection.

6.2. Experimental verification

We can see from objective function (12) that when the distribution difference regularization term λ and structure-preserving regularization term γ are both set to zero, the algorithm falls back to the DiPLS algorithm. So it is possible to observe whether the distribution difference regular term works by giving a larger value of λ. Fig. 12 shows the effect of the distribution difference alignment term. We can see from Fig. 12 that the distribution difference of the source domain and the target domain in the latent variable space can be effectively narrowed by introducing the distribution difference regular term.

When considering the source domain error, we use the same model that predicts the target domain data to predict the source domain data. Fig. 13 shows the effect of the distribution alignment regularization term parameter on the source domain error. As the figure shows that when $\lambda = 0$, model is trained using only source domain data. when $\lambda > 0$, model is trained using both source and target domain data. We can see that the introduction of distribution alignment regularization can effectively reduce the source domain error.

Fig. 14 shows the effect of the structure preserving regularization term parameter on the source domain error. From a similar analysis above, we can conclude that the introduction of structure preserving regularization can effectively reduce the source domain error.

7. Conclusion

In this paper, a dynamic tranfer partial least squares domain adaptive regression algorithm is proposed, which projects high-dimensional process data into a low-dimensional latent variable space, and establishes a dynamic regression relationship between latent variables and labels in the latent variable space. At the same time, based on the distribution difference and smoothness assumption, the domain adaptation regular term and the structure preservation regular term are introduced, which effectively reduces the influence of the data distribution difference and improves the modeling accuracy of the system. Comparative experiments on public datasets show that by dynamically learning the features of the source and target domains in the latent variable space, narrowing the distribution differences between the source and target domains, and maintaining their respective data structures can effectively improve predictive modeling precision.

CRediT authorship contribution statement

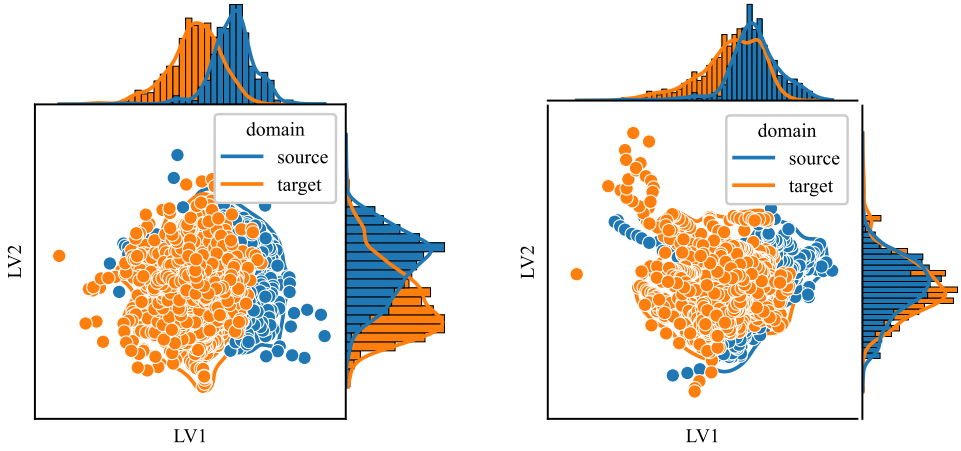
Zhijun Zhao: Conceptualization, Methodology, Software, Writing – original draft. **Gaowei Yan:** Methodology, Writing – review & editing, Supervision. **Mifeng Ren:** Methodology, Visualization, Investigation. **Lan Cheng:** Conceptualization, Methodology. **Zhujun Zhu:** Project administration. **Yusong Pang:** Validation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

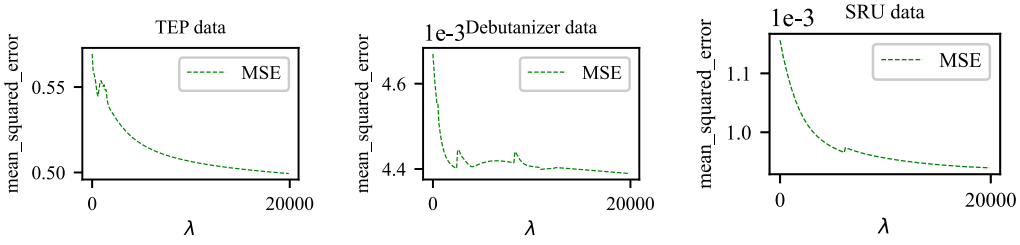
Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61973226, 62073232, 62003233), the Natural Science Foundation of Shanxi Province (20210302123189), and the Shanxi Provincial Key Research and Development Project (201903D121143).



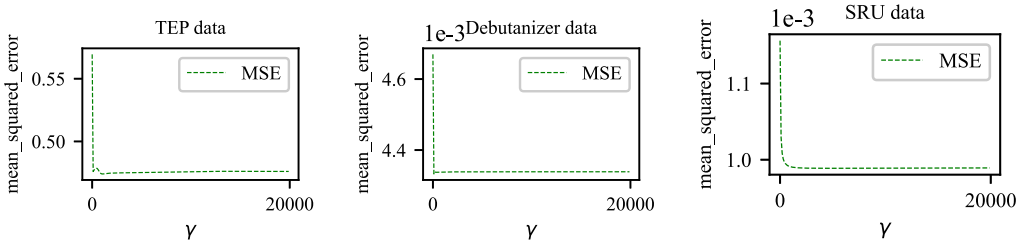
(a) DiPLS method (b) DTPLS method

Fig. 12. Projections of source and target domain data on the first two latent variables of a DiPLS and a DTPLS model, respectively, on the Debutanizer dataset. The middle part of the graph is a scatter plot, the upper and right parts of the figure are the frequency histogram and its corresponding kernel density estimate. It can be seen from the figure that the DTPLS method can effectively reduce the distribution difference of latent variables, which is consistent with the theoretical analysis.



(a) TEP data (b) Debutanizer data (c) SRU data

Fig. 13. The influence of distribution alignment regularization term on source domain error.



(a) TEP data (b) Debutanizer data (c) SRU data

Fig. 14. The influence of structure preserving regularization term on source domain error.

References

[1] Y.A. Shardt, X. Yang, K. Brooks, A. Torgashov, Data quality assessment for system identification in the age of big data and industry 4.0, IFAC-PapersOnLine 53 (2) (2020) 104–113.

[2] F. Curreri, L. Patanè, M.G. Xibilia, Soft sensor transferability: A survey, Appl. Sci. 11 (16) (2021) 7710.

[3] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under concept drift: A review, IEEE Trans. Knowl. Data Eng. 31 (12) (2018) 2346–2363.

[4] P.B. Dongre, L.G. Malik, A review on real time data stream classification and adapting to various concept drift scenarios, in: 2014 IEEE International Advance Computing Conference (IACC), IEEE, 2014, pp. 533–537.

[5] C. Zhao, B. Huang, A full-condition monitoring method for nonstationary dynamic chemical processes with cointegration and slow feature analysis, AIChE J. 64 (5) (2018) 1662–1681.

[6] Q. Jun-Fei, G. Zi-Hao, T. Jian, Dioxin emission concentration measurement approaches for municipal solid wastes incineration process: a survey, Acta Automat. Sinica 46 (6) (2020) 1063–1089.

[7] K. Peng, K. Zhang, B. You, J. Dong, Quality-related prediction and monitoring of multi-mode processes using multiple PLS with application to an industrial hot strip mill, Neurocomputing 168 (2015) 1094–1103.

[8] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proc. IEEE 109 (1) (2020) 43–76.

[9] J. Jiang, Y. Ji, X. Wang, Y. Liu, J. Wang, M. Long, Regressive domain adaptation for unsupervised keypoint detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 6780–6789.

[10] J. Wang, C. Zhao, Mode-cloud data analytics based transfer learning for soft sensor of manufacturing industry with incremental learning ability, Control Eng. Pract. 98 (2020) 104392.

[11] P. Kadlec, B. Gabrys, S. Strandt, Data-driven soft sensors in the process industry, Comput. Chem. Eng. 33 (4) (2009) 795–814.

[12] P. Kadlec, R. Grbić, B. Gabrys, Review of adaptation mechanisms for data-driven soft sensors, Comput. Chem. Eng. 35 (1) (2011) 1–24.

[13] P. Geladi, B.R. Kowalski, Partial least-squares regression: a tutorial, Anal. Chim. Acta 185 (1986) 1–17.

[14] A. Lindquist, G. Picci, Canonical correlation analysis, approximate covariance extension, and identification of stationary time series, Automatica 32 (5) (1996) 709–733.
[15] Y. Dong, S.J. Qin, New dynamic predictive monitoring schemes based on dynamic latent variable models, Ind. Eng. Chem. Res. 59 (6) (2020) 2353–2365.
[16] P.-H. Chen, C.-J. Lin, B. Schölkopf, A tutorial on ν-support vector machines, Appl. Stoch. Models Bus. Ind. 21 (2) (2005) 111–136.
[17] Y. Liu, C. Yang, K. Liu, B. Chen, Y. Yao, Domain adaptation transfer learning soft sensor for product quality prediction, Chemometr. Intell. Lab. Syst. 192 (2019) 103813.
[18] H. Abdi, L.J. Williams, Principal component analysis, Wiley Interdiscip. Rev. Comput. Stat. 2 (4) (2010) 433–459.
[19] W. Ku, R.H. Storer, C. Georgakis, Disturbance detection and isolation by dynamic principal component analysis, Chemometr. Intell. Lab. Syst. 30 (1) (1995) 179–196.
[20] Y. Dong, S.J. Qin, A novel dynamic PCA algorithm for dynamic data modeling and process monitoring, J. Process Control 67 (2018) 1–11.
[21] G. Li, S.J. Qin, D. Zhou, A new method of dynamic latent-variable modeling for process monitoring, IEEE Trans. Ind. Electron. 61 (11) (2014) 6438–6445.
[22] Y. Dong, S.J. Qin, Regression on dynamic PLS structures for supervised learning of dynamic data, J. Process Control 68 (2018) 64–72.
[23] Q. Liu, S.J. Qin, Perspectives on big data modeling of process industries, Acta Automat. Sinica 42 (2) (2016) 161–171.
[24] Q. Dai, C. Zhao, B. Huang, Incremental variational Bayesian Gaussian mixture model with decremental optimization for distribution accommodation and fine-scale adaptive process monitoring, IEEE Trans. Cybern. (2022) 1–14.
[25] C. Zhao, Perspectives on nonstationary process monitoring in the era of industrial artificial intelligence, J. Process Control 116 (2022) 255–272.
[26] Z. Ge, Review on data-driven modeling and monitoring for plant-wide industrial processes, Chemometr. Intell. Lab. Syst. 171 (2017) 16–25.
[27] Z. Ge, T. Chen, Z. Song, Quality prediction for polypropylene production process based on CLGPR model, Control Eng. Pract. 19 (5) (2011) 423–432.
[28] B. Alakent, Soft sensor design using transductive moving window learner, Comput. Chem. Eng. 140 (2020) 106941.
[29] B. Alakent, Soft-sensor design via task transferred just-in-time-learning coupled transductive moving window learner, J. Process Control 101 (2021) 52–67.
[30] G. Li, B. Liu, S.J. Qin, D. Zhou, Dynamic latent variable modeling for statistical process monitoring, IFAC Proc. Vol. 44 (1) (2011) 12886–12891.
[31] J.-M. Lee, C. Yoo, I.-B. Lee, Statistical monitoring of dynamic processes based on dynamic independent component analysis, Chem. Eng. Sci. 59 (14) (2004) 2995–3006.
[32] Y. Dong, S.J. Qin, Dynamic-inner canonical correlation and causality analysis for high dimensional time series data, IFAC-PapersOnLine 51 (18) (2018) 476–481.
[33] Y. Xu, M. Jia, Z. Mao, A novel auto-regressive dynamic slow feature analysis method for dynamic chemical process monitoring, Chem. Eng. Sci. 248 (2022) 117236.
[34] X. Gao, Y.A. Shardt, Dynamic system modelling and process monitoring based on long-term dependency slow feature analysis, J. Process Control 105 (2021) 27–47.
[35] J. Liu, T. Liu, J. Chen, P. Qin, Novel common and special features extraction for monitoring multi-grade processes, J. Process Control 66 (2018) 98–107.
[36] K. Zhang, K. Peng, J. Dong, A common and individual feature extraction-based multimode process monitoring method with application to the finishing mill process, IEEE Trans. Ind. Inf. 14 (11) (2018) 4841–4850.
[37] K. Zhang, K. Peng, S. Zhao, Z. Chen, A novel common and specific features extraction-based process monitoring approach with application to a hot rolling mill process, Control Eng. Pract. 104 (2020) 104628.
[38] Z. Lou, Y. Wang, Y. Si, S. Lu, A novel multivariate statistical process monitoring algorithm: Orthonormal subspace analysis, Automatica 138 (2022) 110148.
[39] X. Chen, S. Wang, J. Wang, M. Long, Representation subspace distance for domain adaptation regression, in: International Conference on Machine Learning, PMLR, 2021, pp. 1749–1759.
[40] Y. Du, J. Wang, W. Feng, S. Pan, T. Qin, R. Xu, C. Wang, Adarnn: Adaptive learning and forecasting of time series, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 402–411.
[41] R. Nikzad-Langerodi, W. Zellinger, S. Saminger-Platz, B.A. Moser, Domain adaptation for regression under Beer–Lambert's law, Knowl.-Based Syst. 210 (2020) 106447.
[42] R. Nikzad-Langerodi, F. Sobieczky, Graph-based calibration transfer, J. Chemometr. 35 (4) (2021) e3319.
[43] R. Nikzad-Langerodi, W. Zellinger, E. Lughofer, S. Saminger-Platz, Domain-invariant partial-least-squares regression, Anal. Chem. 90 (11) (2018) 6693–6701.
[44] G. Huang, X. Chen, L. Li, X. Chen, L. Yuan, W. Shi, Domain adaptive partial least squares regression, Chemometr. Intell. Lab. Syst. 201 (2020) 103986.
[45] H. Wold, Soft modelling by latent variables: the non-linear iterative partial least squares (NIPALS) approach, J. Appl. Probab. 12 (S1) (1975) 117–142.
[46] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, P.S. Yu, Visual domain adaptation with manifold embedded distribution alignment, in: Proceedings of the 26th ACM International Conference on Multimedia, 2018, pp. 402–410.
[47] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Trans. Neural Netw. 22 (2) (2010) 199–210.
[48] B. Sun, J. Feng, K. Saenko, Return of frustratingly easy domain adaptation, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI '16, AAAI Press, 2016, pp. 2058–2065.
[49] J.J. Downs, E.F. Vogel, A plant-wide industrial process control problem, Comput. Chem. Eng. 17 (3) (1993) 245–255.
[50] L. Fortuna, S. Graziani, M.G. Xibilia, Soft sensors for product quality monitoring in debutanizer distillation columns, Control Eng. Pract. 13 (4) (2005) 499–508.
[51] L. Fortuna, S. Graziani, M. Xibilia, G. Napoli, Comparing regressors selection methods for the soft sensor design of a sulfur recovery unit, in: 2006 14th Mediterranean Conference on Control and Automation, IEEE, 2006, pp. 1–6.
[52] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J.W. Vaughan, A theory of learning from different domains, Mach. Learn. 79 (1) (2010) 151–175.