TRANSACTIONS OF THE INSTITUTE OF MEASUREMENT & CONTROL

Article

Transactions of the Institute of Measurement and Control
2020, Vol. 42(6) 1204–1214
© The Author(s) 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0142331219887827
journals.sagepub.com/home/tim
SAGE

Batch process modeling by using temporal feature and Gaussian mixture model

Wei Guo[1], Tianhong Pan[1,2], Zhengming Li[1] and Shan Chen[1]

Abstract

Multi-model/multi-phase modeling algorithm has been widely used to monitor the product quality in complicated batch processes. Most multi-model/multi-phase modeling methods hinge on the structure of a linearly separable space or a combination of different sub-spaces. However, it is impossible to accurately separate the overlapping region samples into different operating sub-spaces using unsupervised learning techniques. A Gaussian mixture model (GMM) using temporal features is proposed in the work. First, the number of sub-model is estimated by using the maximum interval process trend analysis algorithm. Then, the GMM parameters constrained with the temporal value are identified by using the expectation maximization (EM) algorithm, which minimizes confusion in overlapping regions of different Gaussian processes. A numerical example and a penicillin fermentation process demonstrate the effectiveness of the proposed algorithm.

Keywords

Data-driven, batch process, Gaussian mixture model, overlapping modeling, temporal feature

Introduction

It is necessary to monitor the key variable (quality variable or response variable) of a batch process in time. However, the key variable can only be measured at a very limited sampling frequency in many industrial processes. In order to overcome the difficulty of real-time measurement, soft sensors developed using knowledge-based or data-driven strategies have been widely used (Bakirov et al., 2017; Chen et al., 2015). Considerable time and effort are involved in establishing a mechanistic or physical model due to the increasing complexity of the batch process. Therefore, it is difficult to construct a knowledge model in industrial processes (He et al., 2015).

Compared with the knowledge-based model, data-driven method is much flexible and easy to be estimated by using statistical and machine learning techniques (such as principle component analysis (PCA) (Facco et al., 2014; Folch-Fortuny et al., 2015), partial least squares (PLS) (Bao et al., 2015; Luo et al., 2016), kernel PCA/PLS (Botre et al., 2016; Mori and Yu, 2014), probabilistic model (Zheng and Song, 2018; Zhou et al., 2015), independent component analysis (ICA) (Liu et al., 2019), neural networks (NNs) (Aslipour and Yazdizadeh, 2019), deep learning (Gopakumar et al., 2018) and support vector machines (SVMs) (Kaneko and Funatsu, 2013)). Although those methods work well, the overlapping regions on the operating space are always ignored in the aforementioned processes.

Examples of switched linear clustering models such as piecewise affine (PWA) systems and k-means/medians systems are commonly used. The PWA method computes an affine map that fits the training data including sub-model parameters identification and regressor space partition (Breschhi et al., 2016). The *k*-means method generates several mutually exclusive sub-sets with specified cluster number based on the Euclidean distance, Mahalanobis distance, or angle (Zhang et al., 2017). However, the aforementioned hard division strategies assume that the distribution of data can be partitioned into linearly separable regions. Therefore, it is impracticable for dealing with overlapped data of different sub-models. In order to smooth the hard division, fuzzy c-means (FCM) and Gustafson-Kessel (GK) clustering algorithms assign data into different clusters with the membership degree between 0 and 1 (Gautam et al., 2019; Kim et al., 2004). Additionally, Gaussian mixture model (GMM) considers sufficient linear combinations of a single multivariate Gaussian process (GP) (Liang et al., 2018) approximating to almost any continuous process (Grbić et al., 2013; Yuan et al., 2014). Thus, the aforementioned fuzzy division strategies convert the final output as a linear superposition with the weighted sub-output. Irrespective of hard or fuzzy divisions, it is unable to deal with overlapping regions in which data points lie in the proximity of the intersection of multiple regions.

[1] School of Electrical Information and Engineering, Jiangsu University, PR China
[2] School of Electrical Engineering and Automation, Anhui University, PR China

Corresponding author:
Tianhong Pan, School of Electrical Engineering and Automation, Anhui University, Hefei, Anhui 230601, PR China.
Email: thpan@ahu.edu.cn

The size of overlapping regions reflects the classification error rate that further influences the accuracy and generalization performance of inferential sub-models (Sun and Wang, 2011). Several modeling methods focused on this issue. The PWA model uses a greedy strategy and partitions infeasible system of linear inequalities into minimum number of feasible subsystems in which undecidable data points are discarded (Bemporad et al., 2003). This may lead to the loss of information and biased estimation when several identification data points exist. The robust PWARX model overcomes the pitfall of sensitiveness to the starting point with a pre-EM algorithm with certain a stopping condition, and a refinement procedure can classify undecidable data to their own clusters by using the information provided by their spatially closest data points (Jin and Huang, 2010). Although the size of the identification data set is preserved, the misclassified data points still decrease the performance of inferential sub-models. Bayesian framework model combines prior knowledge about work conditions and Bayesian inference model to accommodate overlapping regions (Khatibisepehr et al., 2012). Unfortunately, prior knowledge of the nominal operating condition is not always available without expertise or data analysis. Kernel fisher discriminant analysis (KFDA) is efficient in nonlinear classification problems. However, it is still difficult to accurately classify the data when edge classes and outliers coexist (Lü and Yang, 2014). The probabilistic neural network-based estimator searches an optimized overlapping region detection and a soft decision strategy (Tang et al., 2010). Also, in the two-step classification SVM (Fu et al., 2015), differences between adjacent samples are denoised by wavelet transform and magnified by a proper weighting function, after samples are sorted into correct groups in the first step. However, such two techniques belong to the supervised learning and need the labeling data. Thus, it is difficult to deal with the overlapping phenomenon without introducing other additional criteria (such as prior knowledge and labeled data). Moreover, the optimal number of mixture members selected by deterministic methods or stochastic and resampling methods (Figueiredo and Jain, 2002) involves considerable time. Motived by the above two considerations, a temporal feature GMM (TFGMM) algorithm is proposed in the study to deal with the overlapping regions in clustering process and reduce the computational burden of the optimal mixture number.

The remainder of the study is organized as follows. Section 2 provides a brief overview of the interval-halving framework for quality trend analysis (QTA) and GMM. Section 3 describes the number selection of the sub-Gaussian process by using QTA and TFGMM modeling. Section 4 applies the proposed model to a numerical example and penicillin production process. Section 5 concludes the study.

Preliminary

Interval-halving framework

Interval-halving framework for QTA is a useful approach to exploit the time-series function with respect to a process state (Zhao et al., 2014). Given a variable Y_q with N observations, any time-series function is approximated by temporal value T to any level of detail by using β with a polynomial of certain order τ, and this is as follows

$$\begin{cases} Y_q = T\beta \\ \beta = (T'T)^{-1}T'Y_q \end{cases} \tag{1}$$

where, $\beta = [\beta_0 \quad \beta_1 \quad \cdots \quad \beta_\tau]$, and β_0 is a constant.

The process trend of Y_q is characterized by seven minimal qualitative representations termed as primitives that are determined by the signs of the first and second derivatives of the time-series function (shown in Figure 1). A process is identified as a sequence of piecewise unimodals or quadratic segments by polynomial fitting. Thus, the polynomial is fitted by the least-order (including constant, first order, and quadratic) of T and β labeled with T_{τ_l} and β_{τ_l} as follows

$$Y_q = T_{\tau_l}\beta_{\tau_l} \tag{2}$$

The polynomial fit-error ε is tested by using static F to consider the significance of the error since the elements of the error are assumed as independent and identically distributed, and this is defined as follows

$$\begin{cases} H_0 = 0 : s_\varepsilon^2 = s_w^2 \\ H_0 = 1 : s_\varepsilon^2 > s_w^2 \end{cases} \tag{3}$$

where s_ε denotes the standard deviation of ε, and s_w denotes the standard deviation of wavelet denoising error. The test static F is given as follows

$$F = \frac{s_\varepsilon^2}{s_w^2} \tag{4}$$

which is assumed to follow the χ^2 distribution. An increase in the deviation of this ratio from 1 increases the variation in the evidence for unequal error. The length of the current segment is halved, and the test process is repeated with the first half segment until $H_0 = 0$. Finally, a constrained polynomial fit is used to ensure the continuity of fitted data between different

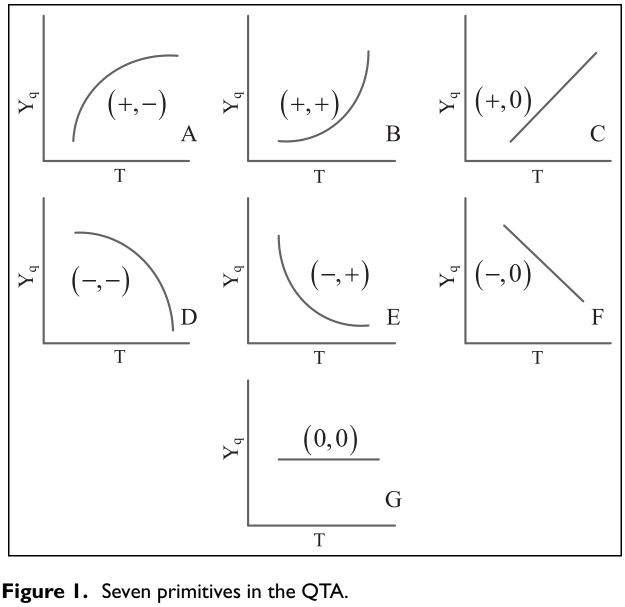


Figure 1. Seven primitives in the QTA.

segments. The details of the interval-halving QTA algorithm are specified in a previous study (Dash et al., 2004).

GMM

The GMM is summarized as the finite mixture density model that is frequently used for unsupervised probabilistic learning problems. It is assumed that the dataset $z \in R^D$ $(z = [(x)', (y)']')$ originates from a unimodal multivariate Gaussian distribution. The probability density function (PDF) is given as follows (Rasmussen and Williams, 2006)

$$f(z|\theta) = \frac{1}{\sqrt{(2\pi)^D |\sum|}} \exp\left\{-1/2(z-\mu)\sum^{-1}(z-\mu)'\right\} \quad (5)$$

where, μ and $\sum$ denote the mean vector and covariance matrix, respectively. Additionally, $\theta = \{\mu, \sum\}$ denotes the parameter to determine the Gaussian distribution. If z comes from a K mixture Gaussian model, the PDF is as follows

$$p(z|\Omega) = \sum_{i=1}^{K} \omega_i f\left(z|\theta_i\right) \quad (6)$$

where, ω_i denotes the probabilistic weight of the i^{th} Gaussian member that is subject to $\omega_i \geqslant 0$, $\sum_{i=1}^{K} \omega_i = 1$. Subsequently, the parameters of GMM Ω are redefined as $\Omega = \{\{\omega_1, \theta_1\}, \{\omega_2, \theta_2\}, \cdots, \{\omega_K, \theta_K\}\}$. Given N independent observations $Z \in R^{D \times N}$, the log-likelihood function of Z is formulated as follows

$$\log L(Z, \Omega) = \sum_{n=1}^{N} \log\left(\sum_{i=1}^{K} \omega_i f(z_n|\theta_i)\right) \quad (7)$$

Furthermore, Ω is estimated as follows

$$\hat{\Omega} = \arg\max_{\Omega}(\log L(Z, \Omega)) \quad (8)$$

and the expectation maximization (EM) algorithm is extensively used (Yuan et al., 2014). Given an initial parameter $\Omega^{(1)}$, the iteration of EM algorithm is performed as follows

E-step:

$$\gamma^{(m)}\left(c_{k,n}\middle|z_n, \Omega^{(m)}\right) = \frac{\omega_k^{(m)} f\left(z_n\middle|\mu_k^{(m)}, \sum_k^{(m)}\right)}{\sum_{i=1}^{K} \omega_i^{(m)} f\left(z_n\middle|\mu_i^{(m)}, \sum_i^{(m)}\right)} \quad (9)$$

M-step:

$$\begin{cases} \omega_k^{(m+1)} = \dfrac{\sum_{n=1}^{N} \gamma^{(m)}\left(c_{k,n}\middle|z_n, \Omega^{(m)}\right)}{N} \\ \mu_k^{(m+1)} = \dfrac{\sum_{n=1}^{N} \gamma^{(m)}\left(c_{k,n}\middle|z_n, \Omega^{(m)}\right) z_n}{\sum_{n=1}^{N} \gamma^{(m)}\left(c_{k,n}\middle|z_n, \Omega^{(m)}\right)} \\ \sum_k^{(m+1)} = \dfrac{\sum_{n=1}^{N} \gamma^{(m)}\left(c_{k,n}\middle|z_n, \Omega^{(m)}\right)\left(z_n - \mu_k^{(m+1)}\right)\left(z_n - \mu_k^{(m+1)}\right)'}{\sum_{n=1}^{N} \gamma^{(m)}\left(c_{k,n}\middle|z_n, \Omega^{(m)}\right)} \end{cases} \quad (10)$$

where, $k = 1, \cdots, K$, $\gamma^{(m)}\left(c_{k,n}\middle|z_n, \Omega^{(m)}\right)$ represents the posterior probability of the n^{th} observation in the k^{th} Gaussian process at the m^{th} iteration. Additionally, the probabilistic weight ω, mean vector μ, and covariance matrix $\sum$ in the k^{th} Gaussian process used for $(m+1)^{th}$ iteration are updated in the M-step. The iteration is terminated when the parameter converges or the iteration is achieved at the maxima steps. Then, the mean vector and covariance of GMM are obtained.

Temporal feature Gaussian mixture model (TFGMM)

The temporal feature used in the proposed TFGMM involves the following two purposes: (1) identify an appropriate number of sub-Gaussian processes for GMM by using QTA and (2) act as an additional criterion to separate the overlapping regions. Generally, the variable in the batch process either maintains a constant state or changes dynamically. In order to capture the dynamic property of the QTA process, the QTA variable Y_q is determined as follows

$$Y_q = \arg\max_{Z_i}(\text{var}(Z_i)) \quad (11)$$

where, $i = 1, \cdots, D$, Z_i denotes the non-pure Gaussian distribution variable that corresponds to phase diversity.

After the QTA variable selection, the initial window of segment $W_s = [T_a^{(1)}\ T_b^{(1)}]$ from 1 to N is normalized to $[0 \quad 1]$, the time-series of Y_q is fitted by equation (2) with the least-order $(\tau_l \leqslant 2)$, and the polynomial fit error is tested to verify whether the fit error is significant from the system noise (it is estimated by wavelet denoising). The result of m^{th} segment division based on the statistical hypothesis test is performed as follows

$$\begin{cases} T_a^{(m)} = T_b^{(m-1)}, \quad T_b^{(m)} = N & if\ H_0^{(m-1)} = 0 \\ T_a^{(m)} = T_a^{(m-1)}, \quad T_b^{(m)} = \frac{T_a^{(m-1)} + T_b^{(m-1)}}{2} & if\ H_0^{(m-1)} = 1 \end{cases} \quad (12)$$

In order to maximize the current segment size and simplify divisions in which the situation of the trend is changed in the range of $\left(T_b^{(m)}, T_b^{(m-1)}\right)$, several additional test procedures $(l = 1, 2, 3 \cdots)$ are given as follows

$$\begin{cases} T_a^{(m),l} = T_a^{(m)} \\ T_b^{(m),l} = T_b^{(m)} + \lambda l \end{cases} \quad if \quad H_0^{(m-1)} = 1, \quad H_0^{(m),l} = 0 \quad (13)$$

where λ denotes the minimum acceptable window size desired that reduces the steps of additional test while ensuring the correct segment division. Additionally, the segment division proceeds with $T_a^{(m+1)} = T_b^{(m),l}, T_b^{(m+1)} = N$ until there is no further need for a division. Figure 2 illustrates the idea of segment maximization QTA on Y_q.

During the division procedure, the regression parameter of different segments is updated by using the constrained least square model to construct a continuous connection, which is as follows

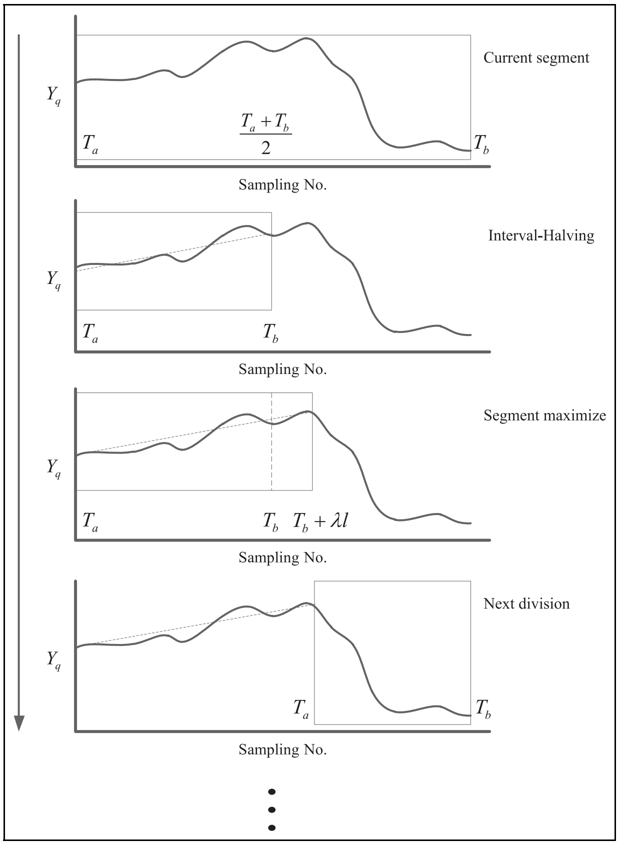


Figure 2. Segment maximization of the QTA.

$$L(\beta^1,\beta^2,\varsigma,\zeta) = \varepsilon^{1\prime}\varepsilon^1 + \varepsilon^{2\prime}\varepsilon^2 + \varsigma\left(r^{1\prime}\beta^1 - r^{2\prime}\beta^2\right) + \xi\left(r^{0\prime}\beta^1 - d^0\right) \tag{14}$$

where, β^1 and β^2 denote the fitting parameters of 2^{nd} and 3^{rd} segments, respectively, in three adjacent divisions; d^0 and r^1 denote the last regressors of 1^{st} and 2^{nd} segments, respectively; r^0 and r^2 denote the first regressors of 2^{nd} and 3^{rd} segments, respectively; ε^1 and ε^2 denote the fitting errors of 2^{nd} and 3^{rd} segments, respectively; and ς and ξ are Lagrange multipliers, respectively. After the identification of the primitives, the total number of segments K is used for the GMM algorithm (there is no need to essentially identify the primitives since the division number K is sufficient for the next procedure).

The GMM is feasible for clustering, and the clustering rule based on GMM involves determining the maximum posteriori probability of sample z as follows

$$c_k(z) = \arg\max(\gamma(c_i|z,\Omega)), \quad i = 1, \cdots, K \tag{15}$$

Figure 3 illustrates the overlap between c_1 and c_2 in which the overlap range (dark shadow) increases with increases in the dispersion and decreasing distance of centers. The overlap section may involve a misclassification given the determination based on equation (15) and exhibits a strong connection with clustering performance. Hence, a coefficient of overlap between c_1 and c_2 is formulated as follows (Sun and Wang, 2011)

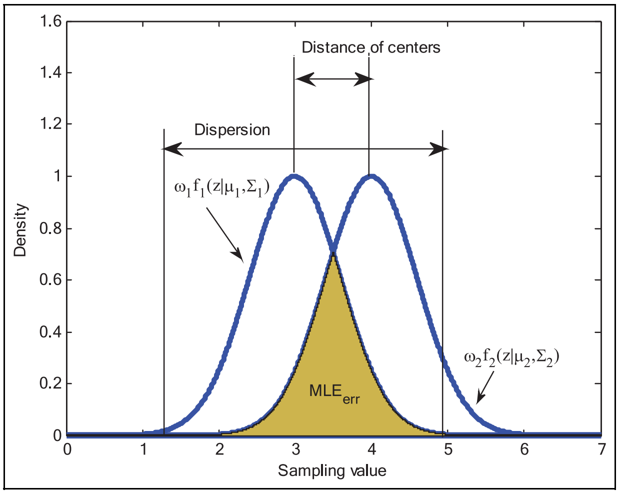


Figure 3. Overlap between the K=2 Gaussian mixture in one-dimension.

$$MLE_{err} = \int \min\left[\omega_1 f_1\left(z\middle|\mu_1, \sum\nolimits_1\right), \omega_2 f_2\left(z\middle|\mu_2, \sum\nolimits_2\right)\right] dz \tag{16}$$

The general form of the MLE_{err} in K mixture is as follows

$$MLE_{err} = 1 - \int \max\left[\omega_1 f_1\left(z\middle|\mu_1, \sum\nolimits_1\right), \cdots, \omega_K f_K\left(z\middle|\mu_K, \sum\nolimits_K\right)\right] dz \tag{17}$$

There is no reasonable behavior to decrease MLE_{err} in the original dimension due to the premise of maximum likelihood estimation with GMM. In the batch processes, each batch follows a confirmed temporal stage such as the initial step, transition, and stabilization. Once the overlap sample is labeled with an index of a different stage, the misclassification is less than unlabeled data. In order to distinguish the samples in the overlapping section, a punishment factor is added by introducing temporal features, and the log-likelihood function is expressed as follows

$$\log L(Z,\Omega) = \sum_{n=1}^{N} \log\left(\sum_{i=1}^{K} \omega_i f(z_n|\theta_i) p(t_n|T_i)\right) \tag{18}$$

where, t_n denotes the temporal value of n^{th} data, T_i denotes the temporal stage of i^{th} Gaussian process, and $p(t_n|T_i)$ represents the probability of sample n appearing in phase i. Unfortunately, $p(t_n|T_i)$ is not known in the absence of prior knowledge. We assume that the beginning and ending of T_i are T_i^a and T_i^b, respectively. Furthermore, $p(t_n|T_i)$ is inversely proportional to the distance between t_n and the center of T_i $(T_{c,i} = T_i^a + \frac{T_i^b - T_i^a}{2})$ with slope R as follows

$$p(t_n|T_i) \propto \frac{R}{\|t_n - T_{c,i}\|^2} \tag{19}$$

Given the stable work stages in batch processes, the center of T_i obeys a Gaussian distribution as $T_{c,i} \sim N\left(\mu_i^T, \sum_i^T\right)$. The sign of the first derivative of $p(t_n|T_i)$ is identical to that of the

PDF of $T_{c,i}$. Therefore, $p(t_n|T_i)$ in equation (18) is replaced by $f\left(t_n\middle|\mu_i^T, \sum_i^T\right)$ as follows

$$\log L(Z^*, \Omega^*) = \sum_{n=1}^{N} \log\left(\sum_{i=1}^{K} \omega_i f(z_n|\theta_i) f\left(t_n\middle|\mu_i^T, \sum\nolimits_i^T\right)\right) \tag{20}$$

where, Z^* includes Z and temporal value T, and Ω^* includes Ω, $\left\{\mu_i^T, \sum_i^T\right\}$. It is observed that each Gaussian process variable is independent of $T_{c,i}$, and the log-likelihood function is re-expressed as the format of a new joint probability density as follows

$$\log L(Z^*, \Omega^*) = \sum_{n=1}^{N} \log\left(\sum_{i=1}^{K} \omega_i f\left(\begin{bmatrix} \eta t_n \\ z_n \end{bmatrix} \middle| \begin{bmatrix} \eta\mu_i^T \\ \mu_i^Z \end{bmatrix}, \begin{bmatrix} \eta^2 \sum_i^{T,T} & \eta \sum_i^{T,Z} \\ \eta \sum_i^{Z,T} & \sum_i^{Z,Z} \end{bmatrix}\right)\right) \tag{21}$$

where, η is a constant that constrains the weight of t_n to avoid achieving N/K samples in each Gaussian process, and μ_i^T is an unbiased estimation of $T_{c,i}$. Thus, the GMM is estimated by the EM algorithm by considering the temporal value as an additional variable (Calinon et al., 2007).

The above formulated equations explicitly obtain the posterior probability in overlapping regions. Subsequently, the parameter associated with temporal value is discarded, and the key variable is predicted by using remaining parameters of mean and covariance. The mean vector μ_k^Z and the covariance matrix $\sum_k^Z$ of k^{th} Gaussian process on the original space are calculated as follows

$$\mu_k^Z = \begin{bmatrix} \mu_k^X \\ \mu_k^Y \end{bmatrix}, \quad \sum\nolimits_k^Z = \begin{bmatrix} \sum_k^{X,X} & \sum_k^{X,Y} \\ \sum_k^{Y,X} & \sum_k^{Y,Y} \end{bmatrix} \tag{22}$$

The estimation of the output related to test data x_{te} in the k^{th} Gaussian process is generalized as follows

$$\begin{cases} \hat{y}_k = \mu_k^Y + \bar{\sum}_k^{x_{te},X}\left(\bar{\sum}_k^{X,X}\right)^{-1}\left(Y - \mu_k^Y\right) \\ \hat{\sum}_k^{y_k} = \bar{\sum}_k^{x_{te},x_{te}} - \bar{\sum}_k^{x_{te},X}\left(\bar{\sum}_k^{X,X}\right)^{-1}\bar{\sum}_k^{X,x_{te}} \end{cases} \tag{23}$$

where, $\bar{\sum}_k^{(\bullet)}$ is the kernel function of k^{th} sub-model, in which the hyperparameters were optimized by using the conjugate gradient descent method. Additionally, the final output is formulated as follows

$$\begin{cases} \alpha_k = \dfrac{\omega_k f\left(x_{te}, T_{te}\middle|\mu_k^X, \sum_k^{X,X}, \mu_k^T, \sum_k^{T,T}\right)}{\sum\limits_{i=1}^{K} \omega_i f\left(x_{te}, T_{te}\middle|\mu_i^X, \sum_i^{X,X}, \mu_i^T, \sum_i^{T,T}\right)} \\ \hat{y} = \sum\limits_{k=1}^{K} \alpha_k \hat{y}_k \\ \hat{\sum}^y = \sum\limits_{k=1}^{K} \alpha_k^2 \hat{\sum}_k^{y_k} \end{cases} \tag{24}$$

The procedure of the TFGMM algorithm is given below.

Pseudo-code for training phase:

```
initialization: $T_a^{(m)} = 0$, $T_b^{(m)} = 1$, $\tau_l = 0$, $K = 1$, $m = 1$, $Y_q$,
$Z^* = \begin{bmatrix} \eta T \\ Z \end{bmatrix}$ and $\Omega^*$
while: $m = 1$, or $H_0 = 1$, or $T_b^{(m)} \neq 1$
    while: $\tau_l \leqslant 2$
        do $F$ test
        if: $H_0 = 1$
            $\tau_l = \tau_l + 1$
        else: repeat $T_b^{(m)} = T_b^{(m)} + \lambda$ until $H_0 = 1$ or $T_b^{(m)} = 1$
            $m = m + 1$, $K = K + 1$
            $T_a^{(m)} = T_b^{(m-1)}$, $T_b^{(m)} = 1$
        end
    end
    if:     $H_0 = 1$
            $m = m + 1$, $\tau_l = 0$
            $T_a^{(m)} = T_a^{(m-1)}$, $T_b^{(m)} = (T_a^{(m-1)} + T_b^{(m-1)})/2$
    else:   break
    end
end
output $K$
reset: $m = 1$
while: $\Omega^*$ is not convergence
    calculate $\gamma^{(m)}(c_{k,n}|z_n^*, \Omega^{*(m)})$, $\omega_k^{*(m+1)}$, $\mu_k^{*(m+1)}$, $\sum_k^{*(m+1)}$
    $m = m + 1$
end
```

Case studies

Numerical example

In this section, the proposed algorithm was tested by a numerical batch process that contains two modes as follows

$$\begin{cases} y_1 = 2 + \exp(3x_1 \sin(\pi x_2)) + \sin(\pi x_2) + \epsilon \\ y_2 = 5 + \exp(2x_1 \sin(\pi x_2)) + 2\sin(\pi x_2) + \epsilon \end{cases} \tag{25}$$

where, x_1 and x_2 are the input variables generated from Gaussian distribution, y_1 and y_2 are the outputs of mode 1 and mode 2, respectively, and ϵ is an independent random noise: $\epsilon \sim N(0,\ 0.01)$. In order to simulate a complex work condition and an overlap, the distribution of inputs were set as listed in Table 1. Two batches containing a total of 1000 samples were collected in which the training batch includes 250 samples of G_1 and 250 samples of G_2 (identical to the testing batch). The input of training data is shown in Figure 4 where the overlapping appears in the vicinity of point $(0.5, 0.5)$.

The parameter of the proposed method was selected as follows: $\eta = 2 \times 10^{-2}$. Additionally, $K = 2$ was obtained from the QTA result (shown in Figure 5). The posterior probability of the test data to G_1 and G_2 (shown in Figure 6 (b), (d)) was contrasted with the posterior probability by using traditional GMM (shown in Figure 6 (a), (c)). It is observed that the posterior probabilities in Figure 6 (b) and Figure 6 (d) were closer to that of the original case. The predictive result of testing data was shown in Figure 7. In order to evaluate the performance of the proposed method, two numerical criteria, namely the root mean square error (RMSE), the average

Table 1. Variables and working conditions.

Variable	G_1		G_2	
	Setting	MLE	Setting	MLE
x_1	(0.3, 0.06)	(0.225, 0.058)	(0.6, 0.08)	(0.598, 0.075)
x_2	(– 0.6, 0.2)	(– 0.566, 0.198)	(1.2, 0.15)	(1.205, 0.154)

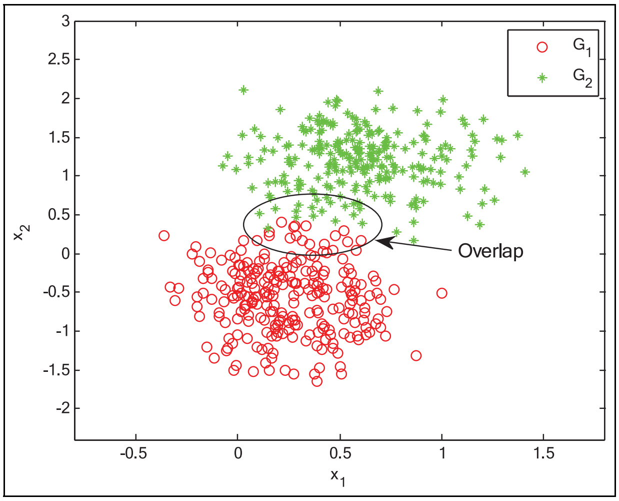


Figure 4. Scatter plot of the inputs.

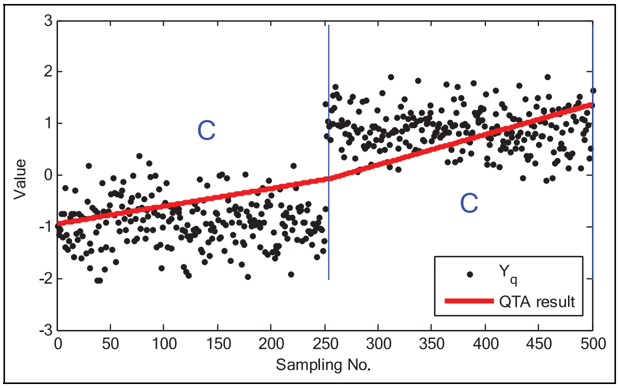


Figure 5. QTA with primitive judgment.

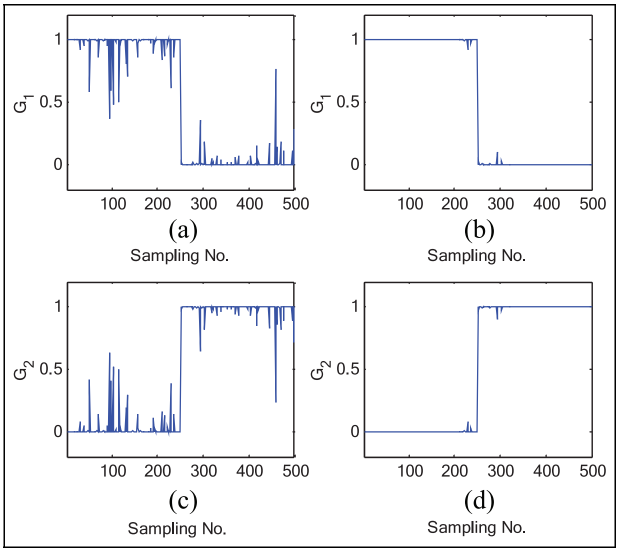


Figure 6. Contrast of posteriors between the GMM (as shown in (a) and (c)) and the proposed method (as shown in (b) and (d)).

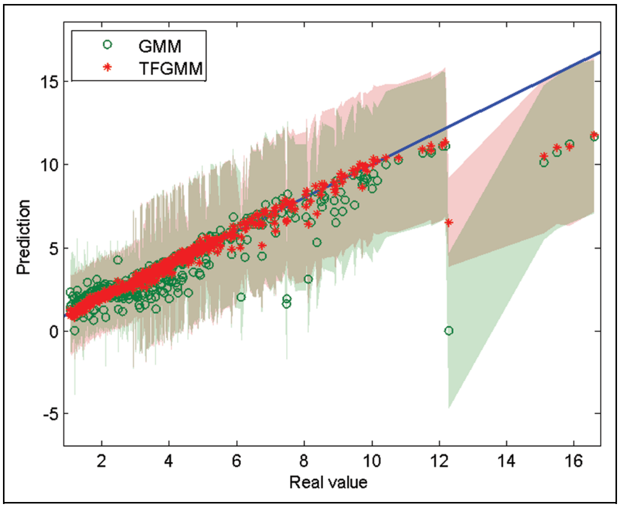


Figure 7. Prediction results of the GMM and the proposed method in 95% confidence.

relative error of prediction (AREP), and the adjusted R-square (R^2_{adj}) were selected

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \qquad (26)$$

$$AREP = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|y_i - \hat{y}_i|}{y_i}\right) \times 100\% \qquad (27)$$

$$R^2_{adj} = 1 - \frac{(1 - R^2)(N - 1)}{N - D - 1} \qquad (28)$$

Table 2. Criteria and parameter estimation of different methods.

Method	RMSE	AREP	R^2_{adj}	G_1	G_2
				$x_1(0.225, 0.058)$ $x_2(-0.566, 0.198)$	$x_1(0.598, 0.075)$ $x_2(1.205, 0.154)$
GMM	1.0447	14.2%	0.952	$(0.223, 0.058)$ $(-0.585, 0.182)$	$(0.592, 0.075)$ $(1.188, 0.165)$
TFGMM	0.5564	5.00%	0.986	$(0.225, 0.057)$ $(-0.567, 0.196)$	$(0.598, 0.075)$ $(1.205, 0.153)$

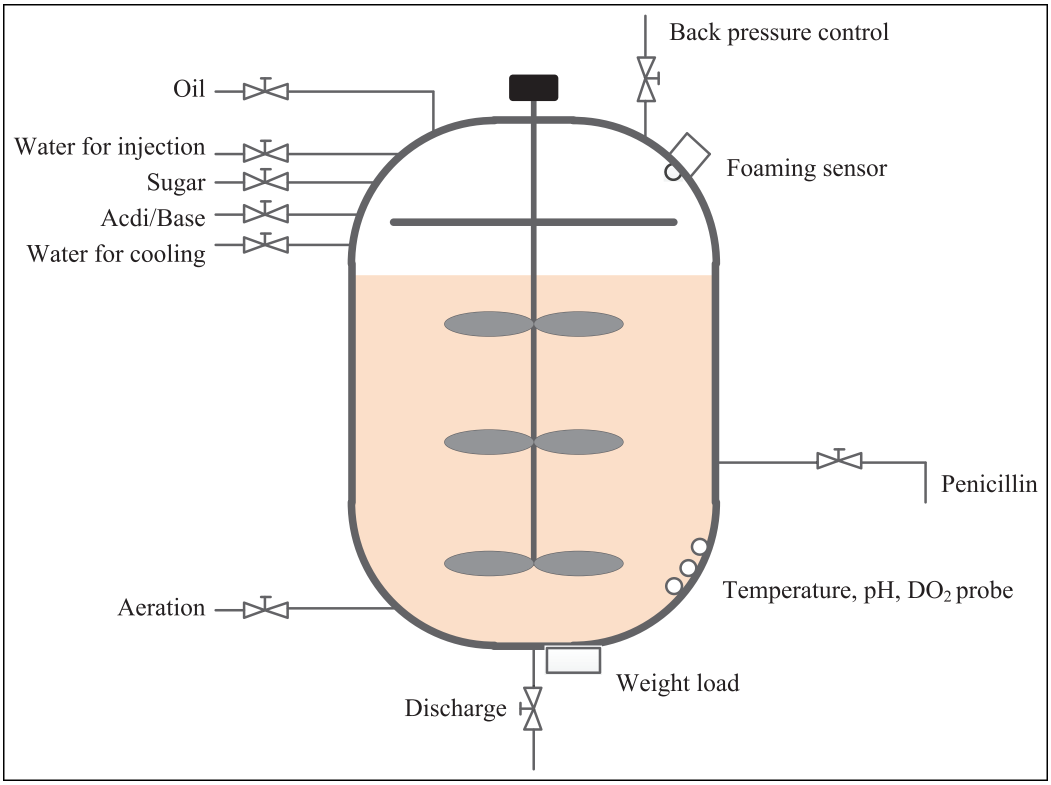


Figure 8. Flow sheet of the bioreactor for the penicillin fermentation process.

where, R^2 is the R-square, N is the number of samples, and D is the sample dimension. Comparison between GMM and proposed method was shown in Table 2. It can be seen that the proposed TFGMM method obtains a high accuracy in variable prediction and parameter estimation. Furthermore, the predictive performances of GMM and TFGMM were influenced by the parameter estimation using the EM algorithm (or the posteriori probability). Therefore, it is observed that the overlap section deteriorates the posterior probability and the estimation of parameters by using GMM. As a result, the predictive accuracy was also deteriorated.

Fermentation process for penicillin production

The penicillin fermentation process is a common biochemical batch benchmark that is widely used to evaluate soft sensor and fault diagnosis algorithms (Miao et al., 2015; Yuan et al., 2014). It is a typical nonlinear, multi-phase, dynamic, and non-unimodal process. Figure 8 shows the flow sheet of the bioreactor (Goldrick et al., 2015) with a radius of 2.1 m. Three Rushton impellers with an internal radius of 0.85 m were operated at a fixed agitation of 100 rpm. The vessel was equipped with pH, temperature, dissolved oxygen, foaming, and pressure sensors. Feed rates of soybean oil and substrate were controlled through sequential batch control by using a predetermined optimum profile that was manually adjusted throughout the batch by the operators in response to process behavior. Similarly, both the aeration rate and vessel back pressure were manipulated by using sequential batch control to maintain the desired dissolved oxygen concentration level as well to respond to process issues including foaming and high CO_2 levels. Therefore, soybean oil was utilized as a secondary carbon source and as an anti-foaming agent. Nitrogen was present in the starting medium of each batch and

Table 3. The parameters used in PID controllers for temperature and pH.

Controller output	Limitation	K_P	K_I	K_D
Acid flow rate (L/h)	[0, 225]	8×10^{-2}	12.5	0.125
Base flow rate (L/h)	[0, 225]	8×10^{-2}	4×10^{-5}	8
Cooling water flow rate (L/h)	$[0, 1.5\times10^3]$	−300	1.6	0.005
Heating water flow rate (L/h)	$[0, 1.5\times10^3]$	50	0.05	1

Table 4. Modeling variables in penicillin fermentation process.

Variable	Description	Unit
x_1	Aeration rate	L/h
x_2	Substrate feed rate	L/h
x_3	Cooling water flow rate	L/h
x_4	Dissolved oxygen concentration	g/L
x_5	Culture volume	L
x_6	pH	-
x_7	Fermentor temperature	K
x_8	Generated heat	kJ
x_9	Carbon dioxide concentration	%
y_1	Penicillin concentration	g/L
y_2	Biomass concentration	g/L
y_3	Substrate concentration	g/L

monitored throughout by using offline measurements. Low levels of nitrogen were rectified through the addition of ammonia sulphate shots. Temperature and pH were both critical process parameters recorded on the industrial-scale bioreactors. Coolant through the internal cooling coils and addition of acid/base solutions were used to control the temperature at 298 K and the pH at 6.5, respectively, by using PID controllers with the parameters (shown in Table 3). The vessel weight was recorded on-line by using a load cell that was used to schedule discharges to ensure that the capacity of the bioreactor did not exceed the limits, and the discharges also allow the achievement of longer batches.

For soft sensor model construction, nine measured variables related to product quality were considered as input and the penicillin, biomass, and substrate concentration were chosen as the output (shown in Table 4). The delay-free samples

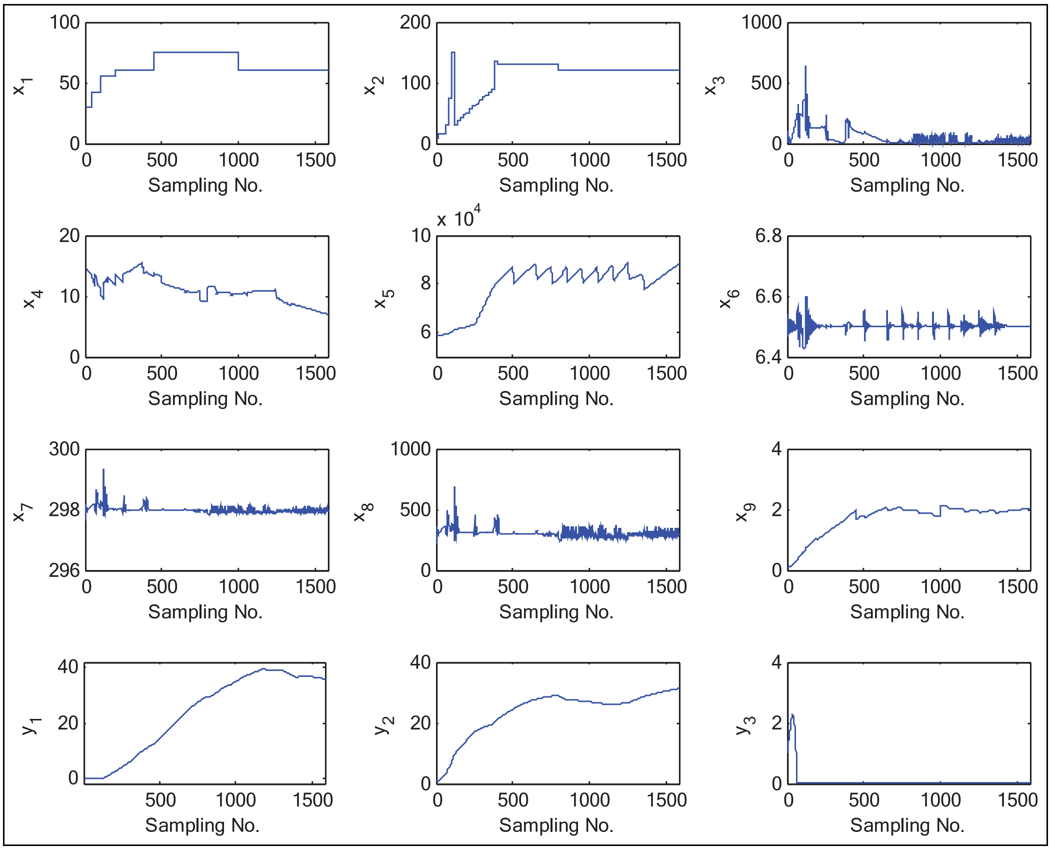


Figure 9. Input and output variables of the training data for the process.

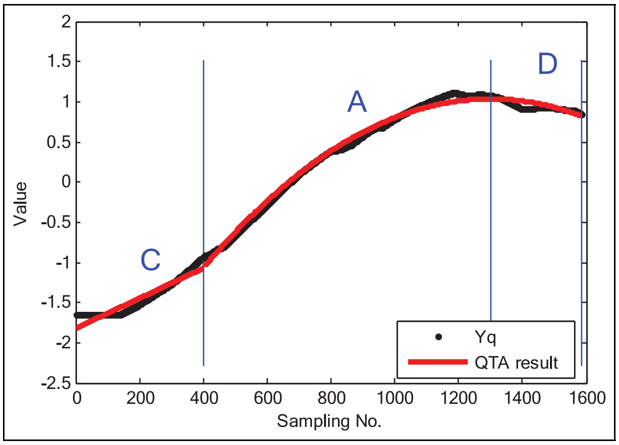
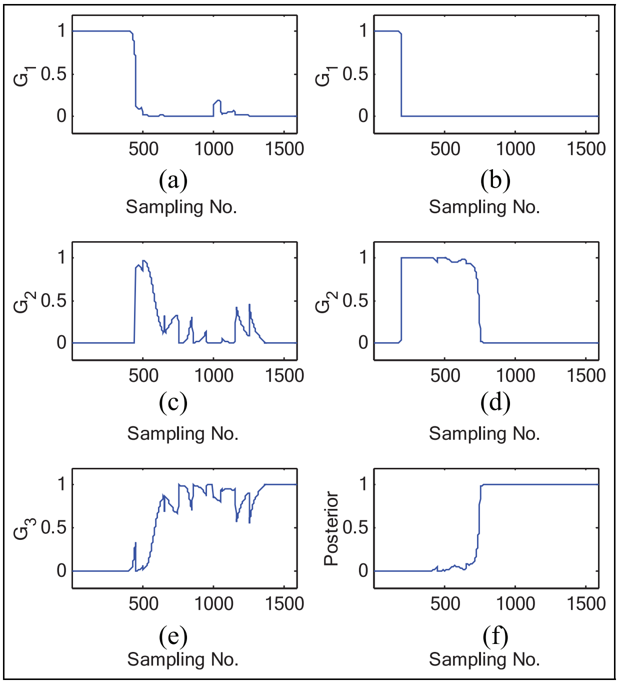


Figure 10. QTA with primitive judgment.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 11. Contrast of posteriors between the GMM (as shown in (a), (c), and (e)) and the proposed method (as shown in (b), (d), and (f)).

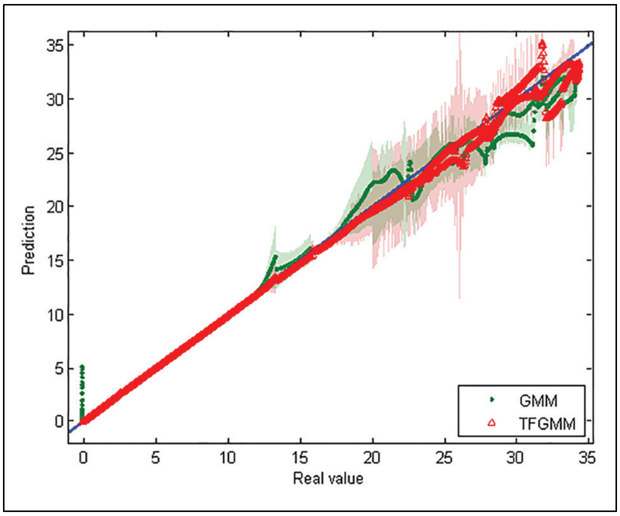


Figure 12. Penicillin concentration predictive results of the GMM and the proposed method in 95% confidence.

Table 5. Criteria of different methods on penicillin fermentation process.

Method	RMSE			R^2_{adj}		
	y_1	y_2	y_3	y_1	y_2	y_3
GMM	1.3051	0.7806	0.0615	0.9961	0.9988	0.9662
TFGMM	0.9591	0.3695	0.0585	0.9978	0.9997	0.9696

of each variable were generated from the simulator presented by Goldrick et al. (2015). Also, the initial conditions and the parameters were set by referencing the 3rd industrial-scale observation. Totally, three batches were used for training and three batches were used for testing. Also, the parameter used for modeling was obtained by the combination of each appropriative batch training result. One batch in the training set with the sampling rate of 0.2 h was shown in Figure 9. Specifically, η was selected as 1×10^{-3}, and $K = 3$ was obtained by using the proposed algorithm (shown in Figure 10). Although a delay occurs in QTA, there is no effect that reflects the trend number. The posterior probability of a test batch to modes #1-#3 using the GMM and the proposed method were shown in Figure 11. It is observed that the initial phase (mode #1) was too long for penicillin fermentation process and the rising phase (mode #2) was too short by using the GMM. Furthermore, the maturing phase (mode #3) was mixed with mode #2 (samples from 700 to 1400). The proposed method provides a more certain probability description and reasonable segment division than GMM on a fermentation process. The predictive result of penicillin concentration within 95% confidence was shown in Figure 12, which reflected the consequence of overlapping distribution on data set. And the overall RMSE, and R^2_{adj} of three batches outputs (including penicillin, biomass, and substrate concentration) using the proposed method were shown in Table 5. It can be seen that the proposed TFGMM provides a higher predictive accuracy than GMM.

Discussion

Samples on the overlapping region were both contained in the above two unsupervised cases. The GMM obtained the mixed posterior probabilities of test samples, which increased the weights of corresponding sub-models from other work conditions. As a result, the final prediction was deteriorated by the increased weights. The TFGMM obtained a more certain probability description about test data than the GMM, which improved the predictive accuracy. Although the GMM and TFGMM shared the same initial parameter in this work, such as the initial center, covariance, and the iterative termination condition, the performance of TFGMM has been improved by embedding the temporal feature into the GMM. Moreover, the proposed TFGMM algorithm can also be applied to the continuous process with cyclic stages by resetting the temporal value.

Conclusions

A TFGMM regression algorithm is proposed for a complicated batch process with an overlapping section in different regimes in this work. An EM algorithm is presented to estimate GMM that is punished by a probability of the temporal phase. The key variable is predicted by utilizing the property of detachable parameter in GMM. Conversely, a maximum interval QTA algorithm is applied to determine an optimal number of sub-Gaussian processes. The validity and effectiveness of the proposed method are demonstrated by applying the proposed method to a numerical example and the fermentation process for penicillin production.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported by the National Natural Science Foundation under Grant 61873113, the Key R&D Program of Jiangsu Province under Grant BE2018370 and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (under Grant KYCX17_1785).

ORCID iD

Tianhong Pan https://orcid.org/0000-0002-0993-3937

References

Aslipour Z and Yazdizadeh A (2019) Identification of Damavand tokamak using fractional order dynamic neural network. *Transactions of the Institute of Measurement and Control* 41(5): 1447–1457.

Bakirov R, Gabrys B and Fay D (2017) Multiple adaptive mechanisms for data-driven soft sensors. *Computers and Chemical Engineering* 96: 42–54.

Bao L, Yuan XF and Ge ZQ (2015) Co-training partial least squares model for semi-supervised soft sensor development. *Chemometrics and Intelligent Laboratory Systems* 147: 75–85.

Bemporad A, Garulli A, Paoletti S, et al. (2003) A greedy approach to identification of piecewise affine models. *Hybrid Systems: Computation and Control* 2623: 97–112.

Botre C, Mansouri M, Nounou M, et al. (2016) Kernel PLS-based GLRT method for fault detection of chemical processes. *Journal of Loss Prevention in the Process Industries* 43: 212–224.

Breschhi V, Piga D and Bemporad A (2016) Piecewise affine regression via recursive multiple least squares and multicategory discrimination. *Automatica* 73: 155–162.

Calinon S, Guenter F and Billard A (2007) On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics* 37(2): 286–298.

Chen K, Castillo I, Chiang LH, et al. (2015) Soft sensor model maintenance: A case study in industrial processes. *IFAC-PapersOnLine* 48(8): 427–432.

Dash S, Maurya MR, Venkatasubramanian V, et al. (2004) A novel interval-halving framework for automated identification of process trends. *American Institute of Chemical Engineers* 50(1): 149–162.

Facco P, Largoni M, Tomba E, et al. (2014) Transfer of process monitoring models between plants: Batch systems. *Chemical Engineering Research and Design* 92(2): 273–284.

Figueiredo MAT and Jain AK (2002) Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(3): 381–396.

Folch-Fortuny A, Arteaga F and Ferrer A (2015) PCA model building with missing data: New proposals and a comparative study. *Chemometrics and Intelligent Laboratory Systems* 146: 77–88.

Fu MY, Tian Y and Wu F (2015) Step-wise support vector machines for classification of overlapping samples. *Neurocomputing* 155: 159–166.

Gautam RK, Singh N, Choudhary NK, et al. (2019) Model order reduction using factor division algorithm and fuzzy c-means clustering technique. *Transactions of the Institute of Measurement and Control* 41(2): 468–475.

Goldrick S, Ştefan A, Lovett D, et al. (2015) The development of an industrial-scale fed-batch fermentation simulation. *Journal of Biotechnology* 193: 70–82.

Gopakumar V, Tiwari S and Rahman I (2018) A deep learning based data driven soft sensor for bioprocesses. *Biochemical Engineering Journal* 136: 28–39.

Grbić R, Slišković D and Kadlec P (2013) Adaptive soft sensor for online prediction and process monitoring based on a mixture of Gaussian process models. *Computers and Chemical Engineering* 58: 84–97.

He YL, Ge ZQ and Zhu QX (2015) Data driven soft sensor development for complex chemical processes using extreme learning machine. *Chemical Engineering Research and Design* 102: 1–11.

Jin X and Huang B (2010) Robust identification of piecewise/switch ing autoregressive exogenous process. *American Institute of Chemical Engineers* 56(7): 1829–1844.

Kaneko H and Funatsu K (2013) Adaptive soft sensor model using online support vector regression with time variable and discussion of appropriate hyperparameter settings and window size. *Computers and Chemical Engineering* 58: 288–297.

Khatibisepehr S, Huang B, Xu FW, et al. (2012) A Bayesian approach to design of adaptive multi-model inferential sensors with application in oil sand industry. *Journal of Process Control* 22(10): 1913–1929.

Kim YI, Kim DW, Lee D, et al. (2004) A cluster validation index for GK cluster analysis based on relative degree of sharing. *Information Sciences* 168(1-4): 225–242.

Liang C, Cai CX and Xu J (2018) Minimum variance lower bound estimation with Gaussian Process models. *Transactions of the Institute of Measurement and Control* 40(6): 1799–1807.

Liu S, Gao XW, Qi WH, et al. (2019) Soft sensor modelling of propylene conversion based on a Takagi-Sugeno fuzzy neural network optimized with independent component analysis and mutual information. *Transactions of the Institute of Measurement and Control* 41(3): 737–748.

Luo LJ, Bao SY, Mao JF, et al. (2016) Quality prediction and quality-relevant monitoring with multilinear PLS for batch processes. *Chemometrics and Intelligent Laboratory Systems* 150: 9–22.

Lü Y and Yang HZ (2014) A multi-model approach for soft sensor development based on feature extraction using weighted kernel fisher criterion. *Chinese Journal of Chemical Engineering* 22(2): 146–152.

Miao A, Li P and Ye LJ (2015) Neighborhood preserving regression embedding based data regression and its applications on soft sensor modeling. *Chemometrics and Intelligent Laboratory Systems* 147: 86–94.

Mori J and Yu J (2014) Quality relevant nonlinear batch process performance monitoring using a kernel based multiway non-Gaussian latent subspace projection approach. *Journal of Process Control* 24(1): 57–71.

Rasmussen CE and Williams CKI (2006) *Gaussian Processes in Machine Learning*. Cambridge: The MIT Press.

Sun HJ and Wang SR (2011) Measuring the component overlapping in the Gaussian mixture model. *Data Mining and Knowledge Discovery* 23(3): 479–502.

Tang WY, Mao KZ, Mak LO, et al. (2010) Classification for overlapping classes using optimized overlapping region detection and soft decision. In: *13th International Conference on Information Fusion*, Edinburgh, UK, 26–29 July 2010, pp. 1–8. IEEE.

Yuan XF, Ge ZQ and Song ZH (2014) Soft sensor model development in multiphase/multimode processes based on Gaussian mixture regression. *Chemometrics and Intelligent Laboratory Systems* 138: 97–109.

Zhang XC, Liu H and Zhang XT (2017) Novel density-based and hierarchical density-based clustering algorithms for uncertain data. *Neural Networks* 93: 240–255.

Zhao LP, Zhao CH and Gao FR (2014) Regression modeling and quality prediction for multiphase batch processes with inner-phase analysis. *Chemometrics and Intelligent Laboratory Systems* 135: 1–16.

Zheng JH and Song ZH (2018) Semisupervised learning for probabilistic partial least squares regression model and soft sensor application. *Journal of Process Control* 64: 123–131.

Zhou L, Chen JH, Song ZH, et al. (2015) Semi-supervised PLVR models for process monitoring with unequal sample sizes of process variables and quality variables. *Journal of Process Control* 26: 1–16.

Appendix

Notations and descriptions

D	Measurement dimension
K	Number of classifications
L	Likelihood function
N	The number of observations
T	Temporal value of the time-series
T_a, T_b	Temporal bounds of window segment in QTA
T_i	Temporal stage in GMM
T_i^a, T_i^b	Beginning and ending of temporal stage in GMM
$T_{c,\,i}$	Center of temporal stage
W_s	Window of segment
X	Input samples
Y	Output samples
Y_q	Variable used for QTA
Z	Data set of all measurements
Z^*	Data set including measurements and temporal values
c	Cluster index
d^0, r^0, r^1, r^2	Designative segment regressors
m	Iteration times of EM algorithm
s	Standard deviation
t	Temporal value
w	Wavelet denoising errors
x_{te}	Test sample
$\widehat{y}$	Finial output prediction
$\widehat{y}_k$	Sub-model output prediction
z	The dataset of one sample
α_k	Weight of sub-model for prediction
β	Regression parameters of the time-series
β^1, β^2	Regression parameters of different segments
ε	Fit-errors of time-series
$\varepsilon^1, \varepsilon^2$	Fet-errors of two different segments
ς, ζ	Lagrange multipliers
η	Weight constraint
θ	Parameters of PDF
λ	Minimum acceptable window size
μ	Mean vector
$\sum$	Covariance matrix
τ, τ_l	Order of time-series polynomial
Ω	Parameters of GMM
Ω^*	Parameters of TFGMM
ω	Probabilistic weight of mixture model