JOURNAL OF CHEMOMETRICS
J. Chemometrics 2005; **19**: 427–438
Published online 5 January 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cem.945

Clustering multivariate time-series data

Ashish Singhal[1]* and Dale E. Seborg[2]

[1]Johnson Controls, Inc. 507 E. Michigan Street, Milwaukee, WI 53202, USA
[2]Department of Chemical Engineering, University of California, Santa Barbara, CA 93106, USA

Received 9 December 2004; Revised 22 August 2005; Accepted 25 August 2005

A new methodology for clustering multivariate time-series data is proposed. The new methodology is based on calculating the degree of similarity between multivariate time-series datasets using two similarity factors. One similarity factor is based on principal component analysis and the angles between the principal component subspaces while the other is based on the Mahalanobis distance between the datasets. The standard *K*-means clustering algorithm is modified to cluster multivariate time-series datasets using similarity factors. Simulation data from two nonlinear dynamic systems: a batch fermentation and a continuous exothermic chemical reactor, are clustered to demonstrate the effectiveness of the proposed technique. Comparisons with existing clustering methods show several advantages of the proposed method. Copyright © 2006 John Wiley & Sons, Ltd.

KEYWORDS: clustering; similarity factor; fault diagnosis; process monitoring

1. INTRODUCTION

Cluster analysis is the art of finding groups in data. Because living beings, object and events encountered in everyday life are too numerous for processing as individual entities, they are usually groped on the basis of the similarity of their features into categories. A related term, *classification*, is the process or act of assigning a new item or observation to its proper place in an established set of categories or classes [1,2].

In industrial plants, modern data recording systems collect large amounts of data that contain valuable information about normal and abnormal behavior of the process. It would be beneficial if these data could be categorized into groups of operating conditions so that the characteristics of these groups can be used for decision support in fault detection and diagnosis, gross error detection, etc. [3] There have been numerous textbooks [1,4] and publications on clustering of scientific data for a variety of areas such as taxonomy [5], agriculture [6], remote sensing [7], as well as process control [3,8].

Clustering of multivariate time-series data attempts to find the groups of datasets that have similar characteristics. These groups can then be further analyzed in detail to gain insight from the common characteristics of the datasets in each group. The process knowledge acquired from the clustering can be very valuable for activities such as process improvement or fault diagnosis, where each new operating condition could be classified as either an existing condition or a new condition.

*Correspondence to: Ashish Singhal, Johnson Controls, Inc., 507 E. Michigan Street, Milwaukee, WI 53202, USA.
E-mail: Ashish.Singhal@jci.com
Contract/grant sponsor: ChevronTexaco Research and Technology Co.
Contract/grant sponsor: OSI Software Inc.
Contract/grant sponsor: UCSB Process Control Consortium.

In this paper, a new clustering methodology for process data, particularly multivariate time-series data, is presented. It is assumed that the database contains sets of multivariate time-series data, which correspond to different periods of process operation, for example different batches produced by a batch process. The clustering methodology is based on calculating the degree of similarity using principal component analysis (PCA) and distance similarity factors.

We first review the previous research concerning clustering process data in Section 2 and then present our approach in Section 3. Sections 4 and 5 describe two simulation examples and results using the new methodology.

2. PREVIOUS WORK

Although clustering is a popular topic in the area of pattern recognition, relatively few applications have been reported in the process monitoring and chemometrics literature. Most reported chemometrics applications cluster objects that can be described by a set of *features* or *attributes* [9–11]. Clustering is then performed using widely available methodologies [1,4] or their modifications/extensions. Although there is a large amount of literature available concerning clustering methodologies and applications [12–18], only a few applications have been reported that cluster multivariate time-series data, such as data from process engineering and control applications.

Many researchers have used PCA with clustering to reduce the dimensionality of the feature space. The number of linearly dependent features are reduced and their scores are calculated. The scores are then used as 'new' uncorrelated features that are clustered [19–22]. Guthke and Schmidt-Heck [23] used clustering and PCA to estimate the phase of growth of microorganisms. Lin *et al*. [24] used PCA

to extract scores from 2-D images and then used clustering methods to estimate the motion characteristics to distinguish different moving objects. Rosen and Yuan [25] used dynamic PCA to extract the principal components that represent the underlying mechanisms of a wastewater treatment process for monitoring. Thus, they used cluster analysis on the principal components to determine the operational state of the process.

In analyses of forearm ischemia [26] and clustering gene expression data [27], researchers found that applying PCA to the features does not necessarily improve clustering performance and in some cases may even degrade performance [27]. Although these researchers do not recommend using PCA with clustering, their results appear to be application specific. The successful application of PCA to clustering reported by other researchers mentioned above suggests that PCA can be beneficially used for clustering.

Gaffney and Smyth [28], Smyth [29] and Gershenfeld *et al.* [30] proposed methodologies for clustering sequential data using general probabilistic models, but their approaches are restricted to clustering of univariate sequences, in contrast to the multivariate time series which are typical of process data. Smyth [29] clustered sequential data using polynomial regression models, but this approach has limited applications, given the nonlinear and diverse behavior of industrial time-series data.

Clustering applications have also appeared in the chemical engineering literature. Johnston and Kramer [3] clustered data using a probabilistic approach and the expectation-maximization algorithm. Their methodology involved estimating the probability distributions of the steady states of a system in the multidimensional space of process variables. But this approach is difficult to extend to dynamic systems (such as batch processes) because process dynamics blur the distinction between different operating conditions in the multidimensional space [32].

Huang *et al.* [33] used PCA models to cluster multivariate time-series data by splitting large clusters into smaller clusters based on the amount of variance explained by a number of principal components. This approach can be quite restrictive if the number of principal components for the entire dataset is not known *a priori*, and also because a pre-determined number of principal components may be inadequate for some of the operating conditions.

Wang and McGreavy [3] clustered multivariate time-series data for a simulated fluid catalytic cracking unit in order to classify different operating conditions. The process data were organized as an $m \times n$ matrix of m observations and n variables. The data were clustered by unfolding the multivariate dataset into a long row vector and then using the individual elements as features. Then the datasets were clustered using the Autoclass algorithm [34]. This methodology quickly becomes computationally prohibitive as the number of measurements and variables for each dataset increase. Also, this approach requires that each dataset contains the same number of observations; otherwise, different datasets will contain different number of features. This requirement is quite restrictive for process data where the duration of an operation (e.g. a batch), can vary from one dataset to another. These limitations of clustering unfolded data can be overcome by using PCA and distance similarity factors as measures of similarity between datasets. By contrast, in Wang and McGreavy's methodology, Euclidean distance between features was used as a dissimilarity measure. Our new approach is the focus of this paper.

Wang and Li [8] described another clustering methodology called 'conceptual clustering' for designing state-space-based monitoring systems. This approach generates 'conceptual knowledge' about the major variables, and projects the data to a specific operational state. The dynamic trends are represented using principal components of the data. The datasets are then clustered using simple 2-D plots of the first two principal components for each variable. Although this is an interesting technique, it requires user input and can become tedious for a large number of process variables.

Keogh *et al.* [32] considered clustering of streaming time-series data, and claim that clustering such data is 'meaningless'. We also found that it is difficult to cluster time-series data that includes transients between different plant operating conditions. The transients appear to blur the distinction between operating conditions and result in either too many or too few clusters. Thus, in this paper we assume that the steady-state plant operation periods have been identified for continuous plants and cluster the steady-state datasets, or batch datasets, for the case of batch processes.

In a recent paper, Lin *et al.* [35] considered clustering univariate time-series data using wavelets, expectation-maximization algorithm [36] and *K*-means clustering to group univariate time-series datasets. They decomposed each time series using the wavelet transform and then clustered the resulting wavelet coefficients. Although their approach is promising, the focus of this paper is clustering of *multivariate* time-series datasets.

Srinivasan *et al.* [37] used Euclidean distance to cluster different modes of operation for a fluidized catalytic cracker unit and the Tennessee Eastman challenge process [38]. They also used dynamic PCA-based similarity factors to determine the similarity of transitions between plant modes and proposed that two datasets were similar if the PCA similarity factor was larger than a specified threshold. The threshold could either be calculated from historical data or specified from *a priori* knowledge. This procedure defines similarity between datasets in absolute terms, whereas defining similarity as a relative measure is more appropriate for clustering. Another limitation of the dynamic PCA-based approach is that it is strongly influenced by the ratio of sampling period to the dominant time constant of the process. Two similar transitions that have different dynamics but the same sampling period will result in different autocorrelation functions and consequently different dynamic PCA models. This is another reason why clustering transient data is difficult.

Owsley *et al.* [39] clustered multivariate time-series using Hidden-Markov models. Hidden-Markov models (HMMs) are probabilistic models that are able to capture not only the dependencies between variables, but also the serial correlation in the measurements [40]. Thus, they are well suited for modeling multivariate time series. Each cluster center is represented by an HMM and datasets that can be described most accurately by an HMM are grouped in a cluster. Although the HMM approach is suitable for clustering multivariate time-series data, building HMMs for continuous data

may require either an assumed probability distribution or vector quantization [41]. In spite of these limitations, clustering of multivariate time-series data using HMMs is a promising approach.

3. CLUSTERING USING SIMILARITY FACTORS

Similarity factors can be used instead of Euclidean distance to measure similarity between two multivariate datasets. Krzanowski [42] developed a method for measuring the similarity of two datasets, X_1 and X_2, using a PCA similarity factor that is calculated using the k largest principal components (PCs) of each multivariate dataset. The principal components are also the eigenvectors of the covariance matrix of a multivariate dataset. The PCA similarity factor, S_{PCA} is defined as [42],

$$S_{\text{PCA}} \triangleq \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{k}\cos^2\theta_{ij} \tag{1}$$

where k is the number of selected PCs in both datasets, θ_{ij} is the angle between the ith PC of X_1 and the jth PC of X_2. The number of PCs, k, can be chosen such that k PCs describe at least 95% variance in each dataset.

Because S_{PCA} weights all PCs equally, it may not capture the degree of similarity between the datasets when only one or two PCs explain most of the variance. Thus, it is natural to define a modified PCA similarity factor, S^{λ}_{PCA} that weights each PC by its explained variance. The S^{λ}_{PCA} is defined as [43,44],

$$S^{\lambda}_{\text{PCA}} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k}\left(\lambda_i^{(1)}\lambda_j^{(2)}\right)\cos^2\theta_{ij}}{\sum_{i=1}^{k}\lambda_i^{(1)}\lambda_i^{(2)}} \tag{2}$$

where $\lambda_i^{(1)}$ and $\lambda_i^{(2)}$ are the ith eigenvalues of the first and second datasets respectively.

The distance similarity factor, S_{dist} [45], compares two datasets that may have similar spatial orientation but are located far apart. This similarity factor is particularly useful when two datasets have similar principal components but the values of the process variables may be different due to different operating conditions. The distance similarity factor can be used to distinguish between these cases. The distance similarity factor, S_{dist} is defined as,

$$\begin{aligned} S_{\text{dist}} &\triangleq 2\times\frac{1}{\sqrt{2\pi}}\int_{\Phi}^{\infty}\mathrm{e}^{-z^2/2}\mathrm{d}z \\ &= 2\times\left[1-\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\Phi}\mathrm{e}^{-z^2/2}\mathrm{d}z\right] \end{aligned} \tag{3}$$

where:

$$\Phi = \sqrt{(\bar{\boldsymbol{x}}_2-\bar{\boldsymbol{x}}_1)\,{\boldsymbol{\Sigma}_1}^{*-1}\,(\bar{\boldsymbol{x}}_2-\bar{\boldsymbol{x}}_1)^{\mathrm{T}}} \tag{4}$$

$\bar{\boldsymbol{x}}_1$ and $\bar{\boldsymbol{x}}_2$ are sample mean row vectors. $\boldsymbol{\Sigma}_1$ is the covariance matrix for dataset X_1, and ${\boldsymbol{\Sigma}_1}^{*-1}$ is the pseudo-inverse of X_1 calculated using singular value decomposition. k-singular values used to calculate the pseudo-inverse such that k PCs describe at least 95% of the variance in each dataset. In Equation (4), dataset X_1 is assumed to be the reference dataset.

Note that a one-sided Gaussian distribution is used in Equation (3) because $\Phi \geqslant 0$. The error function in Equation (3) can be evaluated using standard tables or software packages. Also note that the integration in Equation (3) normalizes S_{dist} between zero and one. Because the relative values of S_{dist} are used for clustering and pattern matching applications, any mapping from Φ to S_{dist} may be used that is monotonic and results in $0 \leqslant S_{\text{dist}} \leqslant 1$, and $\Phi_i > \Phi_j \Rightarrow S_{\text{dist,i}} < S_{\text{dist,j}}$.

The similarity factors do not depend on the number of observations in each dataset and can be calculated with a relatively small computational effort. Later, the standard K-means clustering algorithm [1,4] is modified for clustering using similarity factors.

3.1. Inclusion of product quality data

For many practical problems, each dataset includes a set of 'quality measurements' that are made infrequently, for example at the end of a batch or an 8-hour shift. Furthermore, important calculations are often made for each dataset such as a reaction yield, furnace efficiency or error of closure for a mass or energy balance. These infrequent measurements or calculated quantities will be referred to as *additional features* or the 'Y data' to distinguish them from the time-series (process) data, which are referred to as the 'X data' [46]. The dimensions of the **X** data matrix are $m \times n_x$ while the dimensions of the Y data are $1 \times n_y$, where n_y is the number of quality variables.

The similarity factor for the Y data is based on the Euclidean distance between two datasets being compared. The Euclidean distance between the Y data for two datasets, Y_1 and Y_2, is defined as,

$$\Phi_y \triangleq \|Y_1 - Y_2\| \tag{5}$$

where the notation $\|\cdot\|$ represents Euclidean norm. Assuming that the Y data have a Gaussian probability distribution, the distance similarity factor for the Y data, S^{y}_{dist}, is defined as,

$$S^{y}_{\text{dist}} \triangleq \sqrt{\frac{2}{\pi}}\int_{\Phi_y}^{\infty}\mathrm{e}^{-z^2/2}\mathrm{d}z \tag{6}$$

Note that both S_{dist} and S^{y}_{dist} lie between zero and one.

3.2. Combination of similarity factors

When more than one similarity factor is used to calculate the similarity between datasets, a key issue is to decide how should the similarity factors be combined to produce a single measure of the degree of similarity. It is convenient to combine S^{λ}_{PCA} and S_{dist} into a single similarity factor, SF, using a weighted average of the two quantities

$$\text{SF} \triangleq \alpha_1 S^{\lambda}_{\text{PCA}} + \alpha_2 S_{\text{dist}} \quad (\alpha_1+\alpha_2=1) \tag{7}$$

When Y data are included, the combined similarity factor SF, becomes

$$\text{SF} \triangleq \alpha_1 S^{\lambda}_{\text{PCA}} + \alpha_2 S_{\text{dist}} + \alpha_3 S^{y}_{\text{dist}} \quad (\alpha_1+\alpha_2+\alpha_3=1) \tag{8}$$

The weighted average SF can be used as a similarity measure between datasets and is also used for clustering. It is up to the user to choose the weighting factors $\{\alpha_i\}$ to give relative importance of each similarity factor for his/her application. Fortunately, experience has shown that good pattern matching can be obtained for a wide range of $\{\alpha_i\}$.

Algorithm 1. *K* means clustering using similarity factors.

Given: Q datasets, $\{X_1, \ldots, X_q, \ldots, X_Q\}$, to be clustered into K clusters.

1. Let the jth dataset in the ith cluster be denoted by $X_j^{(i)}$. Compute the aggregate dataset χ_i $(i = 1,2,\ldots,K)$, for each of the K clusters as,

$$\chi_i = \left[\left(X_1^{(i)}\right)^T \ldots \left(X_j^{(i)}\right)^T \ldots \left(X_{Q_i}^{(i)}\right)^T\right]^T \tag{9}$$

   where, Q_i is the number of datasets in χ_i. Note that $\sum_{i=1}^{K} Q_i = Q$.
2. Calculate the dissimilarity between dataset X_q $(q = 1,2,\ldots, Q)$ and each of the K aggregate datasets χ_i, $i = 1, 2, \ldots, K$ as,

$$d_{i,q} = 1 - \mathrm{SF}_{i,q} \tag{10}$$

   where $\mathrm{SF}_{i,q}$ is the similarity factor between the qth dataset and the ith cluster described by Equations (7) or (8). Let the aggregate dataset χ_i in Equation (10) be the reference dataset. Dataset X_q is assigned to the cluster to which it is least dissimilar, that is to the cluster that has the smallest value of $d_{i,q}$. Repeat this step for all Q datasets.
3. Calculate the average dissimilarity of each dataset from its cluster as:

$$J(K) = \frac{1}{Q}\sum_{i=1}^{K}\sum_{X_{q\in\chi_i}} d_{i,q} \tag{11}$$

4. If the value of $J(K)$ has changed from a previous iteration, then go to Step 9. Otherwise stop.

3.3. Selection of the number of clusters

One of the key design parameters of the *K*-means clustering algorithm is the specification of the number of clusters, K. There are many methods available to estimate the optimum value of K [13,14]. Also, Rissanen [47,48] proposed model building and model order selection on the basis of model complexity. In Rissanen's approach, more complex models are penalized more than less complex ones. For the present clustering problem, a large number of clusters indicates a more complex model. Several methods that penalize high-model complexity, that is a large number of clusters, such as the Akaike Information Criterion (AIC) and Schwartz Information Criterion (SIC) [49], were evaluated for estimating the optimum number of clusters. However, preliminary results obtained using these methods were not promising and consequently a new method was developed.

Cross-validation is another approach for estimating the number of clusters in the data [50]. For cross-validation, the data are split into two or more parts. One part is used for clustering while the remaining parts are used for validation. However, Krieger and Green [51] have reported that such methods fail to identify the appropriate number of clusters, particularly for large amounts of data and for situations where the variables are highly correlated, which commonly occurs for process data.

We follow a different approach where data are clustered using the *K*-means algorithm for different values of K. We then analyze the sequence $J(K)$ in order to estimate the optimum number of clusters. The value of K is increased from 1 to Q, where Q is the number of datasets. Typically, $J(K)$ decreases with increasing K [2]. However, the optimum number of clusters can be estimated if the value of $J(K)$ changes significantly between consecutive values of K [2,13].

These considerations motivate a new method for estimating the optimum number of clusters. The values of K where the plot of $J(K)$ has a 'knee' are proposed as candidate values for the optimum number of clusters. The clustering algorithm is repeated for different values of K and the percentage change in $J(K)$, $\mathrm{d}J(K)$, is calculated as:

$$\mathrm{d}J(K) \triangleq \frac{|J(K+1) - J(K)|}{J(K)} \times 100\% \quad K = 1,2,3,\ldots \tag{12}$$

The value of $\mathrm{d}J(K)$ is plotted against the number of clusters K. The value of K for which $\mathrm{d}J(K)$ reaches a minimum or is close to zero, is a knee in the plot of $J(K)$. Thus, the sign of the difference of $\mathrm{d}J(K)$ is used to estimate the locations of these 'knees'

$$\psi(K) \triangleq \mathrm{Sign}[\mathrm{d}J(K+1) - \mathrm{d}J(K)] \quad K = 1,2,3,\ldots \tag{13}$$

The quantity, $\psi(K)$, is similar to the sign of the second derivative of $J(K)$. The values of K for which $\psi(K)$ changes from negative to positive as K increases, are selected to be the 'knees' in the $J(K)$ versus K plot. Usually, the location of the first knee is selected as the optimum number of clusters.

3.4. Clustering metrics

Some key definitions are introduced here in order to evaluate the performance of various clustering methodologies considered in this paper. Suppose that data contain N_{op} operating conditions and there are N_{DB_j} datasets of operating condition number j in the database $(j = 1,2,3,\ldots)$. Suppose that the data have been divided into K clusters; then a *cluster purity*, p, is defined to characterize the purity of each cluster in terms of how many datasets of a particular operating condition are present in that cluster. The cluster purity for the ith cluster is defined as,

$$p_i \triangleq \frac{(\max_j N_{i,j})}{N_{\mathrm{P}_i}} \times 100\% \tag{14}$$

where $N_{i,j}$ is the number of datasets of operating condition j in the ith cluster, and N_{P_i} is the number of datasets in the ith cluster. The dominant operating condition in a cluster is the operating condition, which occurs in the largest number in that cluster.

A second metric, the *clustering efficiency*, η, is defined to measure the extent to which an operating condition is distributed in different clusters. If there is perfect partitioning of data, then all datasets for a particular operating condition will be grouped in a single cluster. Thus, this measure is designed to penalize large values of K, when an operating condition is distributed in different clusters.

Table I. Relevant model variables and parameters for acetone-butanol fermentation example

Variable/ Parameter	Description	Sampling period
y	Dimensionless cellular RNA concentration	Not measured
X	Reactor cell concentration	30 minutes
S	Reactor substrate concentration	1 minute
BA	Reactor butyric acid concentration	1 minute
AA	Reactor acetic acid concentration	1 minute
B	Reactor butanol concentration	1 minute
A	Reactor acetone concentration	1 minute
E	Reactor ethanol concentration	1 minute
CO_2	CO_2 concentration	1 minute
H_2	H_2 concentration	1 minute
K_S	Substrate uptake saturation constant	Not applicable
K_I	Butanol inhibition constant	Not applicable

The clustering efficiency for the jth operating condition is defined as,

$$\eta_j \triangleq \frac{(\max_i N_{i,j})}{N_{\mathrm{DB}_j}} \times 100\% \tag{15}$$

where N_{DB_j} is the total number of datasets for operating condition j in the database. The p and η metrics provide a tradeoff between cluster purity and the concentration of operating conditions in separate clusters.

4. SIMULATION CASE STUDY: BATCH FERMENTATION

In order to evaluate different pattern matching techniques, a case study was performed based on a simulated database for batch fermentation. The dynamic model by Vortruba *et al.* [52] summarizes biochemical as well as physiological aspects of growth and metabolite synthesis of acetone-butanol-ethanol fermentation. The model consists of ten nonlinear ordinary differential equations with nine measured variables. A detailed description of the model and its parameters is provided by Vortruba *et al.* [52]. The relevant model variables and parameters are described in Table I while all other parameters are described by Vortruba *et al.* [52]. The measured variables are also shown in Table I while the quality variables calculated at the end of every batch are presented in Table II.

For simulation purposes, the cell inoculum, glucose and other nutrients are added to the reactor, and fermentation is allowed to proceed for a fixed period of time in order to produce acetone, butanol and ethanol.

Table II. Quality data calculated at the end of every batch for the acetone-butanol fermentation example

Variable/Parameter	Formula	Description
Y_X	$(X - X_0)/(S_0 - S)$	Cell yield
Y_{BA}	$(BA - BA_0)/(S_0 - S)$	Butyric acid yield
Y_B	$(B - B_0)/(S_0 - S)$	Butanol yield
Y_{AA}	$(AA - AA_0)/(S_0 - S)$	Acetic acid yield
Y_A	$(A - A_0)/(S_0 - S)$	Acetone yield
Y_E	$(E - E_0)/(S_0 - S)$	Ethanol yield

4.1. Generation of data

Model parameter values and initial conditions were varied from batch to batch in order to simulate abnormal operation. Each abnormal operating condition was characterized by an abnormal value of a cell physiology parameter. The magnitude of the abnormal parameter varied randomly from batch to batch. The duration of each batch was 30 hour. The five operating conditions and parameter ranges are shown in Table III. The operating conditions were simulated to provide a database of 100 batches. The number of batches of each operating condition in the database were different and are given in Table III. Gaussian measurement noise was added to the measured variables so that the signal-to-noise ratio for a normal batch run was approximately equal to 10.

4.2. Pre-processing of data

The reactor cell concentration was measured at 30-minute intervals while the other eight process variables in Table I were measured every minute during the 30-hour batch operation. In order to synchronize cell concentration measurement with the other eight process variables, linear interpolation was used to obtain values of cell concentration every minute between the 30-minute samples. Thus, the data consisted of a total of 180 000 measurements for each process variable. The data were then averaged every 5 minutes. Because each batch was of 30-hour duration, the averaging produced 360 measurements per variable for each batch.

Other simple methods such as zero-order hold or cubic spline interpolation could also have been used for reconstruction of missing values, but linear interpolation offered two advantages: (i) it provides more accurate information about the missing values (between the 30-minute samples) compared to a zero-order hold, and (ii) it is computationally less intensive than cubic spline interpolation. For these reasons, linear interpolation was used in this research.

4.3. Results for the batch fermentation example

The proposed clustering methodology was evaluated using data from the batch fermentation case study and a continuous reactor presented later in Section 5. Different combinations of similarity factors were used to characterize the similarity between datasets and the clustering results were compared for each case. Because the K-means algorithm can become trapped in local minima, for each value of K, clustering was repeated using ten independent and random initial guesses for the cluster memberships. After the convergence of the clustering procedure, the solution that resulted in the lowest value for $J(K)$ was considered as the best clustering solution.

We also compare the proposed clustering methodology with Wang and McGreavy's approach of clustering unfolded data to show the advantage of the new method. Because clustering unfolded data requires the datasets to have the same number of observations, every dataset was forced to be of the same duration for a fair comparison even though the new clustering method does not have this restriction.

The K-means clustering procedure was repeated for values of $K = 2$ through 10 for $\mathrm{SF} = 0.67S^{\lambda}_{\mathrm{PCA}} + 0.33S_{\mathrm{dist}}$

Table III. Operating modes for the acetone-butanol fermentation example

Op ID	Description	Nominal parameter values	Parameter ranges	N_{DB}
1	Normal batch operation	$y_0 = 1.0$ $X_0 = 0.03\,g/L$ $S_0 = 50\,g/L$	$0.9 \leqslant y_0 \leqslant 1.1$ $0.01 \leqslant X_0 \leqslant 0.05\,g/L$ $45 \leqslant S_0 \leqslant 55\,g/L$	13
2	Slow substrate utilization	$K_S = 40\,g/L$	$30 \leqslant K_S \leqslant 50\,g/L$	20
3	Increased cell sensitivity to butanol	$K_I = 0.425\,g/L$	$0.25 \leqslant K_I \leqslant 0.6\,g/L$	31
4	Decreased cell sensitivity to butanol	$K_I = 1.27\,g/L$	$1.11 \leqslant K_I \leqslant 1.42\,g/L$	24
5	Dead inoculum	$y_0 = 0.075\,g/L$ $X_0 = 0.003\,g/L$	$0.05 \leqslant y_0 \leqslant 0.1\,g/L$ $0.001 \leqslant X_0 \leqslant 0.005\,g/L$	12

Table IV. Results using similarity factors and X data only for the batch fermentation example

				Operating condition				
Cluster no.	N_P	p (%)	Dominant OpID	1	2	3	4	5
1	13	92	1	12	0	0	1	0
2	12	100	2	0	12	0	0	0
3	8	100	2	0	8	0	0	0
4	31	100	3	0	0	31	0	0
5	24	96	4	1	0	0	23	0
6	12	100	5	0	0	0	0	12
Average	17	98	NA	$\eta = 92$ $\eta_{av} = 90$	60	100	96	100

$SF = 0.67\, S^{\lambda}_{PCA} + 0.33\, S_{dist}$

Table V. Results using PCA scores of X data only for the batch fermentation example

				Operating condition				
Cluster no.	N_P	p (%)	OpID Dominant	1	2	3	4	5
1	20	100	2	0	20	0	0	0
2	18	100	3	0	0	18	0	0
3	23	57	3	10	0	13	0	0
4	27	89	4	3	0	0	24	0
5	12	100	5	0	0	0	0	12
Average	20	89	—	$\eta = 77$ $\eta_{av} = 87$	100	58	100	100

Table VI. Results using unfolded X data only for the batch fermentation example

				Operating condition				
Cluster no.	N_P	p (%)	Dominant OpID	1	2	3	4	5
1	18	50	1	9	0	8	1	0
2	13	100	2	0	13	0	0	0
3	7	100	2	0	7	0	0	0
4	23	100	3	0	0	23	0	0
5	27	85	4	4	0	0	23	0
6	12	100	5	0	0	0	0	12
Average	17	89	NA	$\eta = 69$ $\eta_{av} = 81$	65	74	96	100

and X data only. The simulation results are summarized in Figure 1. It is clear from Figure I(c) that $K = 6$ is the only knee in the plot of $J(K)$, and thus the optimum number of clusters is six. Tables IV–VI present clustering results using the similarity factor approach, PCA scores of unfolded data, and raw unfolded data respectively. Table IV shows that there is a significant improvement in the classification of the data into different clusters compared to the results obtained by clustering unfolded data in Tables V and VI.[1] Operating condition #1 is now present predominantly in a single cluster and operating condition #2 is split between clusters #2 and #3. Only two batches are misclassified out of a total 100. These results show that the use of similarity factors for

[1]The optimum numbers of clusters using the unfolded scores and raw unfolded data were found to be five and six respectively using the approach described in Subsection 3.3.

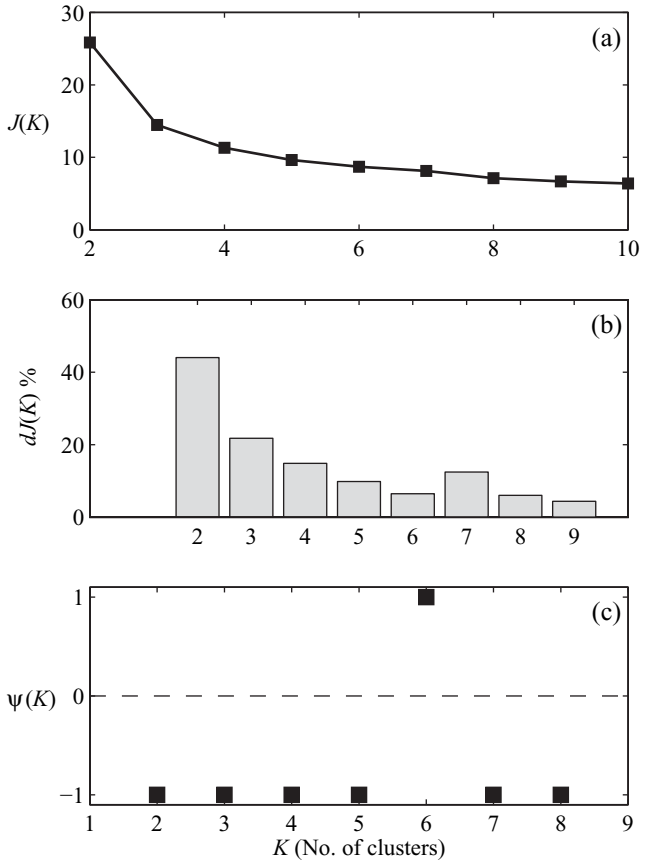

Figure 1. Clustering performance using similarity factors and X data only. SF $= 0.67\ S^{\lambda}_{\text{PCA}} + 0.33\ S_{\text{dist}}$.

clustering is more effective than clustering the scores of the unfolded data. Clustering using similarity factors requires an average of seven iterations with a computation time of 2.09 seconds per iteration, while clustering using unfolded data required seven iterations for convergence and 2.57 seconds per iteration on a Pentium 4/1.7 GHz/512 MB RDRAM computer and Matlab 6.0 for Windows XP.

When quality data are included in the clustering, S^{λ}_{PCA}, S_{dist} and S^{y}_{dist} are combined into a single similarity factor $\text{SF} = 0.5\,S^{\lambda}_{\text{PCA}} + 0.25\,S_{\text{dist}} + 0.25\,S^{y}_{\text{dist}}$. The clustering performance using this linear combination of similarity factors is presented in Figure 2. The optimum number of clusters is five as shown in Figure 2(c). This estimated optimum number of clusters is also the actual number of operating conditions in the data. Table VII indicates that there is only *one* misclassified dataset out of a total of 100. Thus, clustering using similarity factors produces very accurate results. Clustering based on similarity factors using both the X and Y data required an average of six iterations for convergence and each iteration required an average of 2.35 seconds of computer time.

Table VIII. Clustering performance using different combinations of similarity factors for the batch fermentation example

Similarity factor (SF)	Optimum K	Average p (%)	Average η (%)
S_{PCA}	N/A†	N/A	N/A
S^{λ}_{PCA}	6	98	92
S_{dist}	6	97	88
$0.67\ S_{\text{PCA}} + 0.33\ S_{\text{dist}}$	7	94	86
$0.5\ S_{\text{PCA}} + 0.5\ S_{\text{dist}}$	3	81	92
$0.67\ S^{\lambda}_{\text{PCA}} + 0.33\ S_{\text{dist}}$	5	98	90
$0.5\ S^{\lambda}_{\text{PCA}} + 0.5\ S_{\text{dist}}$	5	98	98
$0.5\ S_{\text{PCA}} + 0.5\ S^{y}_{\text{dist}}$	4	65	65
$0.5\ S_{\text{PCA}} + 0.25\ S_{\text{dist}} + 0.25\ S^{y}_{\text{dist}}$	6	98	92
$0.34\ S_{\text{PCA}} + 0.33\ S_{\text{dist}} + 0.33\ S^{y}_{\text{dist}}$	6	98	92
$0.5\ S^{\lambda}_{\text{PCA}} + 0.5\ S^{y}_{\text{dist}}$	4	86	93
$0.5\ S^{\lambda}_{\text{PCA}} + 0.25\ S_{\text{dist}} + 0.25\ S^{y}_{\text{dist}}$	5	99	99
$0.34\ S^{\lambda}_{\text{PCA}} + 0.33\ S_{\text{dist}} + 0.33\ S^{y}_{\text{dist}}$	5	99	99
$0.5\ S_{\text{dist}} + 0.5\ S^{y}_{\text{dist}}$	5	96	98

†Algorithm did not converge for any K

Clustering using different combinations of S^{λ}_{PCA}, S_{PCA}, S_{dist} and S^{y}_{dist} was also evaluated. These results are presented in Table VIII. Results for S_{PCA} alone or in combination with S_{dist} were less impressive than the corresponding cases where S^{λ}_{PCA} was used. The combination of S^{λ}_{PCA}, S_{dist} and S^{y}_{dist} provided the best clustering performance with 99% pure clusters. Also, the clustering performance was not sensitive to the choice of the weighting factors $\{\alpha_i\}$.

When only S^{λ}_{PCA} and S_{dist} were used, the performance was comparable to the situation where all three, S^{λ}_{PCA}, S_{dist} and S^{y}_{dist}, were used. The S^{λ}_{PCA}-S_{dist} combination produced superior results compared to the S^{λ}_{PCA}-S^{y}_{dist} combination. These results show that the X data play a stronger role than Y data in distinguishing between different operating conditions for this case study. This occurs because cell and product yields (Y data) alone are not sufficient to distinguish between batches. Interestingly enough, the cell and product yields are commonly used in the industry to label a batch as satisfactory or out of spec. The use of the standard PCA similarity factor, S_{PCA}, alone did not produce good results because the algorithm failed to converge for any value of K. This result was also observed when only S^{y}_{dist} was used to

Table VII. Clustering results using similarity factors and both X and Y data for the batch fermentation example

				Operating condition				
Cluster no.	N_{P}	p (%)	Dominant OpID	1	2	3	4	5
1	14	93	1	13	0	1	0	0
2	20	100	2	0	20	0	0	0
3	30	100	3	0	0	30	0	0
4	24	100	4	0	0	0	24	0
5	12	100	5	0	0	0	0	12
Average	20	99	NA	$\eta = 100$	100	97	100	100
				$\eta_{\text{av}} = 99$				

$\text{SF} = 0.5S^{\lambda}_{\text{PCA}} + 0.25S_{\text{dist}} + 0.25S^{y}_{\text{dist}}$

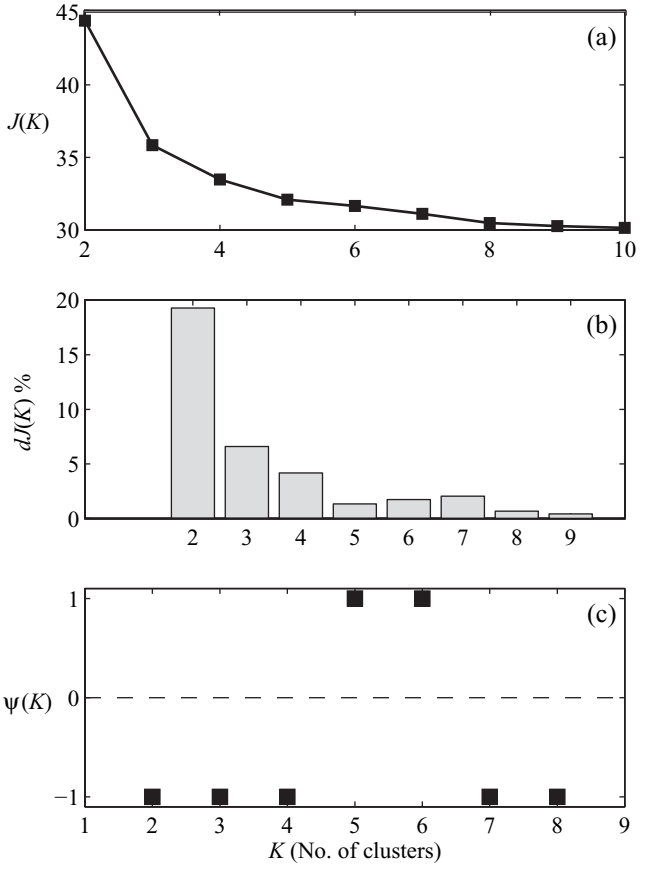

Figure 2. Clustering results using similarity factors with X and Y data for the batch fermentation example. $SF = 0.5S^{\lambda}_{PCA} + 0.25S_{dist} + 0.25S^{y}_{dist}$.

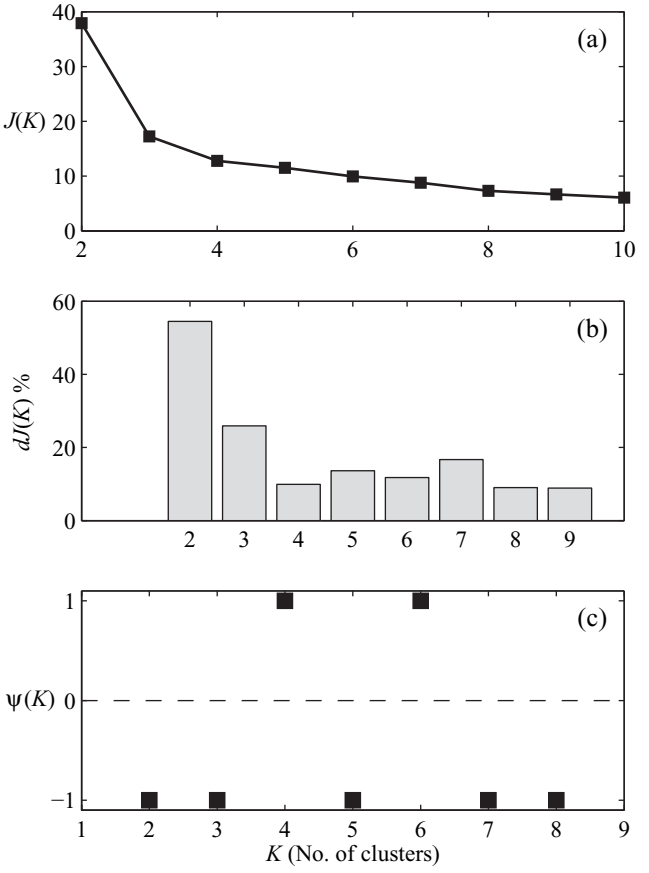

Figure 3. Clustering performance using similarity factors for the CSTR example. $SF = 0.67S_{PCA} + 0.33S_{dist}$.

Table IX. Comparison of clustering techniques for the fermentation example

Clustering method	No. of clusters	Average p (%)	Average η (%)
Unfolded data	6	89	81
PCA scores of X data	5	89	87
Y data and PCA scores of X data	6	89	80
$SF = 0.67\ S^{\lambda}_{PCA} + 0.33\ S_{dist}$	6	98	90
$SF = 0.5\ S^{\lambda}_{PCA} + 0.25\ S_{dist} + 0.25\ S^{y}_{dist}$	5	99	99

characterize the dissimilarity between datasets. Thus, the Y data alone are not sufficient to distinguish between different operating conditions and the X data play an important role in classification.

Table IX compares the results using different clustering techniques. Clearly, the similarity factor methods produce superior results.

5. SIMULATION CASE STUDY: CONTINUOUS REACTOR

A case study was performed for a simulated continuous chemical reactor to evaluate the proposed clustering methodology in addition to the batch fermentation example of Section 4. A nonlinear continuous stirred tank reactor (CSTR) with cooling jacket dynamics, variable liquid level and a first-order irreversible reaction, A → B was simulated. Operating conditions that included faults of varying magnitudes, and disturbances were simulated for the CSTR and 14 process variables for each operating condition were recorded. The details of the simulation study and six different operating conditions for the CSTR are available in previous publications [45,53,54].

The six operating conditions in Table X include a wide range of disturbance and fault types that can be encountered in a typical historical database. Each operating condition was simulated for a period of 85.3 minutes using a sampling period of 5 seconds for each variable. This approach produced 105 different data sets each having 1024 data points for each of the 14 measured variables.

5.1. Results for the CSTR example

The K-means clustering procedure was repeated for $K = 2$ through 10 using two similarity factor combinations: $SF = 0.67S_{PCA} + 0.33S_{dist}$ and $SF = 0.67S^{\lambda}_{PCA} + 0.33S_{dist}$. These results are summarized in Figures 3 and 4.

Figure 3(c) shows the presence of two 'knees' at $K = 4$ and 6 in the $J(K)$ versus K plot. The location of the first knee at $K = 4$ is the optimum number of clusters. The four clusters are analyzed in detail in Table XI. In cluster #1, the normal

Table X. Operating conditions for the CSTR case study

ID	Operating condition	Description	Nominal value
N	Normal operation	Operation at the nominal conditions. No disturbances.	N/A
F1	Catalyst deactivation	The activation energy ramps up.	The ramp rate for E/R is +3 K/minute
F2	Heat exchanger fouling	The heat transfer coefficient ramps down.	The ramp rate for UA_C is −125 (J/(minute(K))/minute
F5	Coolant valve stiction + F7	Dead band for stiction = 5% of the valve span.	N/A
F13	Autoregressive disturbance in feed flow rate	$Q_F(k) = 0.8 \times Q_F(k-1) + w(k).w(k) \sim \mathcal{N}(0,1)$	N/A
O3	Intermediate frequency oscillations in feed flow rate	Sinusoidal oscillations of frequency 0.5 cycles/minute	10 L/minute

Table XI. Results using S_{PCA} and S_{dist} similarity factors for the CSTR example

				Operating condition					
Cluster no.	N_P	p (%)	Dominant OpID	N	F1	F2	F5	F13	O3
1	51	53	N	27	0	0	0	12	12
2	23	100	F1	0	23	0	0	0	0
3	16	100	F2	0	0	16	0	0	0
4	15	100	F5	0	0	0	15	0	0
Average	26	88	NA	$\eta = 100$	100	100	100	100	100

$SF = 0.67S_{PCA} + 0.33S_{dist}$

operating condition is classified with autoregressive (F13) and oscillatory feed disturbances (O3). This result is also obtained when PCA scores of the unfolded data are clustered (Table XII).[2] The other three clusters are pure. Tables XI and XII suggest that it is difficult to distinguish between these types of feed disturbances and normal operation.

The clustering results using $SF = 0.67S^{\lambda}_{PCA} + 0.33S_{dist}$, are shown in Figure 4. The plot of $\psi(K)$ versus K in Figure 4(c) indicates the presence of a 'knee' at $K = 8$. Thus, the optimum number of clusters is eight for the $S^{\lambda}_{PCS} - S_{dist}$ method.

The eight clusters obtained using $SF = 0.67S^{\lambda}_{PCA} + 0.33S_{dist}$ are analyzed in detail in Table XIII. Again, normal operation is classified with the autoregressive feed disturbance in cluster #7. In cluster #6, the autoregressive and the sinusoidal disturbances in the feed are grouped together because both appear to cause similar effect on the reactor concentration and temperature variables. The remaining clusters are pure. The average purity for this clustering is 92%, however, the clustering efficiency drops to 71% due to the large number of identified clusters.

A summary of clustering results obtained using other similarity factor combinations is presented in Table XIV. These results are presented in Table XIV. For the CSTR example the combination $SF = 0.67S_{PCA} + 0.33S_{dist}$ provides the best clustering performance. Different combinations of S_{PCA} with S_{dist} produce different results, while different combinations of S^{λ}_{PCA} and S_{dist} produce results that are very close to each other. Table XIV suggests that the clustering performance is sensitive to the combination of S_{PCA} and S_{dist}, but not to the combination of S^{λ}_{PCA} and S_{dist}.

[2]The optimum number was found to be six.

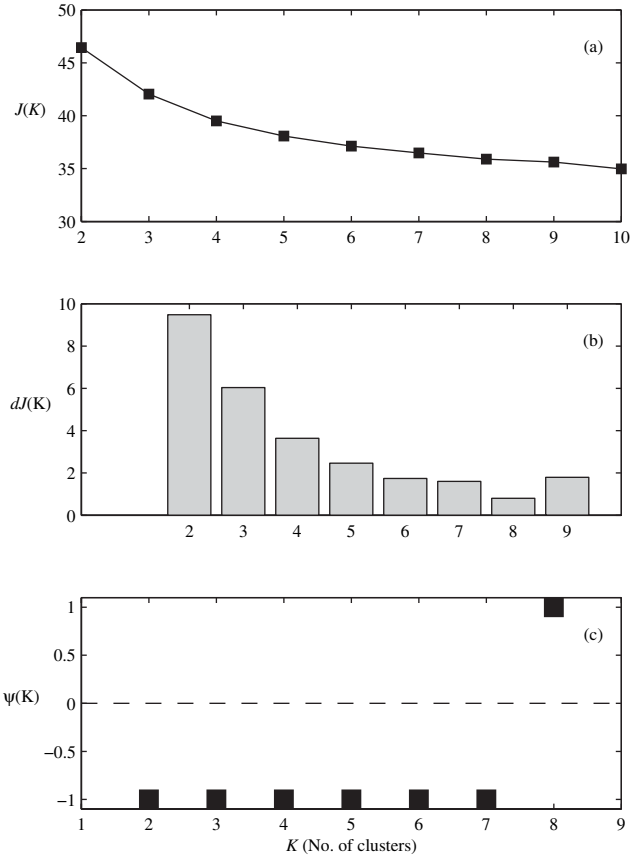

Figure 4. Clustering performance using similarity factor for the CSTR example. $SF = 0.67S^{\lambda}_{PCA} + 0.33S_{dist}$.

Table XII. Results using PCA scores of X-data for the CSTR example

				Operating condition					
Cluster no.	N_P	p (%)	Dominant OpID	N	F1	F2	F5	F13	O3
1	51	53	N	27	5	4	0	12	3
2	18	100	F1	0	18	0	0	0	0
3	12	100	F2	0	0	12	0	0	0
4	15	100	F5	0	0	0	15	0	0
5	5	100	O3	0	0	0	0	0	5
6	4	100	O3	0	0	0	0	0	4
Average	18	92	NA	$\eta = 100$	78	75	100	100	42
				$\eta_{av} = 83$					

Table XIII. Results using similarity factors for the CSTR example

				Operating condition					
Cluster no.	N_P	p (%)	Dominant OpID	N	F1	F2	F5	F13	O3
1	10	100	F1	0	10	0	0	0	0
2	10	100	F2	0	0	10	0	0	0
3	15	100	F5	0	0	0	15	0	0
4	6	100	O3	0	0	0	0	0	6
5	13	100	F1	0	13	0	0	0	0
6	11	55	O3	0	0	0	0	5	6
7	34	79	N	27	0	0	0	7	0
8	6	100	F2	0	0	6	0	0	0
Average	13	92	NA	$\eta = 100$	57	100	100	58	50
				$\eta_{av} = 71$					

Table XIV. Clustering performance using different combinations of similarity factors for the CSTR example

Similarity factor (SF)	Optimum K	Average p (%)	Average η (%)
S_{PCA}	5	90	93
S^{λ}_{PCA}	7	85	76
S_{dist}	7	80	68
$0.5\ S_{PCA} + 0.5 S_{dist}$	5	82	84
$0.67\ S_{PCA} + 0.33 S_{dist}$	4	88	100
$0.5\ S^{\lambda}_{PCA} + 0.5 S_{dist}$	6	89	83
$0.67\ S^{\lambda}_{PCA} + 0.33 S_{dist}$	8	92	71

Table XV. Comparison of clustering techniques for the CSTR example

Clustering method	No. of clusters (K)	Average p (%)	Average η (%)
PCA scores of X data	6	92	82
$SF = 0.67\ S_{PCA} + 0.33\ S_{dist}$	4	88	100

Table XV compares the results obtained by clustering PCA scores of X data and the results obtained by clustering using similarity factors. Clearly, clustering using similarity factors produces superior results because η is larger for the similarity factor method while the p values are close to each other.

6. CONCLUSIONS

A new methodology for clustering of multivariate time-series datasets has been presented and evaluated for two simulation case studies. The proposed methodology uses similarity factors to characterize the degree of dissimilarity between datasets. A new similarity factor to compare product quality data for different datasets has also been presented. The clustering algorithm can group datasets based on both frequently measured process data and additional attributes such as product quality data. A novel, simple procedure is proposed for estimating the optimum number of clusters in the data. Two case studies for a simulated nonlinear batch fermenter and a nonlinear exothermic chemical reactor have shown that the new proposed methodology using similarity factors is very effective in clustering multivariate time-series datasets and is superior to existing methodologies.

REFERENCES

1. Kaufman L, Rousseeuw PR. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley: NY, 1990.
2. Duda RO, Hart PE, Stork DG. *Pattern Classification*, 2nd edn., John Wiley: NY, 2001.
3. Wang XZ, McGreavy C. Automatic classification for mining process operational data. *Ind. Eng. Chem. Res.* 1998; **37**: 2215–2222.
4. Anderberg MR. *Cluster Analysis for Applications*. Academic Press: NY, 1973.
5. Fisher RA. The use of multiple measurements in taxonomic problems. *Ann. Eugenics* 1936; **3**: 179–188.
6. Ruiz-Garćia N, González-Cossío FV, Castillo-Morales A, Castillo-González F. Optimization and validation of cluster analysis applied to classification of Mexican races of maize. *Agrociencia* 2000; **35**: 65–77.
7. Talbot LM, Talbot BG, Peterson RE, Tolley HD, Mecham HD. Application of fuzzy grade-of-membership clustering to analysis of remote sensing data. *J. Climate* 1999; **12**: 200–219.

8. Wang XZ, Li RF. Combining conceptual clustering and principal component analysis for state space based process monitoring. *Ind. Eng. Chem. Res.* 1999; **38**: 4345–4358.
9. Duflou H, Maenhaut W, De Reuck J. Application of principal component and cluster analysis to the study of the distribution of minor and trace elements in normal human brain. *Chemometrics Intel. Lab. Syst.* 1990; **9**: 273–86.
10. Marengo E, Todeschini R. Linear discriminant hierarchical clustering: a modeling and cross-validable divisive clustering method. *Chemometrics Intel. Lab. Syst.* 1993; **19**: 43–51.
11. Chtioui Y, Bertrand D, Barba D, Dattee Y. Application of fuzzy c-means clustering for seed discrimination by artificial vision. *Chemometrics Intel. Lab. Syst.* 1997; **38**: 75–87.
12. Wold SC. Pattern recognition by means of disjoint principal component models. *Pattern Recognition* 1976; **8**: 127–139.
13. Milligan GW, Cooper MC. An examination of procedures for determining the number of clusters in a data set. *Psychometrika* 1985; **50**: 159–179.
14. Dubes RC. How many clusters are best?—An experiment. *Pattern Recognition* 1987; **20**: 645–663.
15. Agrawal R, Gehrke J, Gunopulos D, Raghavan P. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, Seattle, WA, 1998; 94–105.
16. Cadez I, Gaffney S, Smyth P. A general probabilistic framework for clustering individuals. In *Proc. ACM-SIGKDD 2000*, Boston, MA, 2000; 140–149.
17. Maulik U, Bandyopadhyay S. Genetic algorithm-based clustering technique. *Pattern Recognition* 2000; **33**: 1455–1465.
18. Tseng LY, Yang SB. A genetic clustering algorithm for data with non-spherical-shape clusters. *Pattern Recognition* 2000; **33**: 1251–1259.
19. Sudjianto A, Wasserman GS. A nonlinear extension of principal component analysis for clustering and spatial differentiation. *IIE Trans*. 1996; **28**: 1023–1028.
20. Allgood GO, Upadhyaya BR. A model-based high-frequency matched filter arcing diagnostic system based on principal component analysis. In *Proc. of the SPIE—The Intl. Soc. for Optical Engr.*, vol. 4055, Orlando, FL, 2000; 430–440.
21. Jun BS, Ghosh TK, Loyalka SK. Determination of CHF pattern using principal component analysis and the hierarchical clustering method (critical heat Flux in reactors). In *Trans. Amer. Nuclear Soc.*, San Diego, CA, 2000; 250–251.
22. Trouve A, Yu Y. Unsupervised clustering trees by nonlinear principal component analysis. In *Proc. 5th Intl. Conf. on Pat. Rec. and Image Analysis: New Info. Tech.*, Samara, Russia, 2000; 110–114.
23. Guthke R, Schmidt-Heck W. Bioprocess phase detection by fuzzy c-means clustering and principal component analysis. In *Proc. 6th European Congress on Interl. Techniques and Soft Comput., EUFIT '98*, vol. 2, Aachen, Germany, 1998; 1294–1298.
24. Lin YT, Chen YK, Kung SY. A principal component clustering approach to object-oriented motion segmentation and estimation. *J. VLSI Signal Proc. Syst. for Signal, Image and Video Tech.* 1997; **17**: 163–187.
25. Rosen C, Yuan Z. Supervisory control of wastewater treatment plants by combining principal component analysis and fuzzy c-means clustering. *Water Sci. and Tech.* 2001; **43**: 147–156.
26. Mansfield MG J Rand Sowa, Scarth GB, Somorjai RL, Mantsch HH. Fuzzy c-means clustering and principal component analysis of time series from near-infrared imaging of forearm ischemia. *Comput. Med. Imag. Graphics* 1997; **21**: 299–308.
27. Yeung KY, Russo WL. Principal component analysis for clustering gene expression data. *Comput. Sci. & Engr.* 2001; **17**: 763–774.
28. Gaffney S, Smyth P. Trajectory clustering using mixtures of regression models. In *Proc. of ACM SIGKDD Intl. Conf. on Knowl. Discovery and Data Mining*, 1999; 63–72.
29. Smyth P. Probabilistic model-based clustering of multivariate and sequential data. In *Proc. Seventh Intl. Workshop on AI and Statistics*, D Heckermann and J Whittaker (eds). Morgan Kaufmann: Los Gatos, CA, 1999.
30. Gershenfeld N, Schoner B, Metois E. Cluster-weighted modelling for time-series analysis. *Nature* 1999; **397**: 329–332.
31. Johnston LPM, Kramer MA. Estimating state probability functions from noisy and corrupted data. *AIChE J.* 1998; **44**: 591–602.
32. Keogh E, Lin J, Truppel W. Clustering of time series subsequences is meaningless: implications for past and future research. In *Proceedings of 3rd IEEE International Conference. On Data Mining*, Melbourne, FL, 2003; 115–122.
33. Huang Y, McAvoy TJ, Gertler J. Fault isolation in nonlinear systems with structured partial principal component analysis and clustering analysis. Can. *J. Chem. Engr.* 2000; **78**: 569–577.
34. Cheeseman P, Stutz J. Bayesian classification (Autoclass): theory and results. In *Advanced Knowledge of Discrete Data Mining*, UM Fayyad, G Piatetsky-Shapiro, P Smyth, R Uthurusamy (eds). AAAI Press: MIT, 1996.
35. Lin J, Vlachos M, Keogh E, Gunopulos D. Iterative incremental clustering of time series. In *Proceedings of IX Conference on Extending Database Technology*, Crete, Greece, 2004.
36. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the *EM* algorithm. *J. Royal Stat. Soc., Ser. B* 1977; **39**: 1–38.
37. Srinivasan R, Wang C, Ho WK, Lim KW. Dynamic principal component analysis based methodology for clustering process states in agile chemical plants. *Ind. Eng. Chem. Res.* 2004; **43**: 2123–2139.
38. Downs JJ, Vogel EF. A plant-wide industrial process control problem. *Comput. Chem. Eng.* 1993; **17**: 245–255.
39. Owsley L, Atlas L, Bernard G. Automatic clustering of vector time-series for manufacturing machine monitoring. In *Proceeding IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany, 1997; 3393–3396.
40. Rabiner L, Juang BH. *Fundamentals of Speech Recognition*, Prentice Hall: NJ, 1993.
41. Huo QA, Chan CK. Contextual vector quantization for speech recognition with discrete hidden markov model. *Pattern Recognition* 1995; **28**: 513–517.
42. Krzanowski WJ. Between-groups comparison of principal components. *J. Amer. Stat. Assoc.* 1979; **74**: 703–707.
43. Singhal A, Seborg DE. Pattern matching in historical batch data using PCA. *IEEE Control Systems Mag*. 2002; **22**: 53–63.
44. Johannesmeyer MC. *Abnormal Situation Analysis Using Pattern Recognition Techniques and Historical Data*, M.Sc. Thesis, University of California: Santa Barbara, CA, 1999.
45. Singhal A, Seborg DE. Pattern matching in multivariate time series databases using a moving window approach. *Ind. Eng. Chem. Res.* 2002; **41**: 3822–3838.
46. Louwerse DJ, Tates AA, Smilde AK, Koot GLM, Berndt H. PLS discriminant analysis with contribution plots to determine differences between parallel batch reactors in process industry. *Chemometrics Intel. Lab. Syst.* 1999; **46**: 197–206.
47. Rissanen J. Modeling by shortest data description. *Automatica* 1978; **14**: 465–471.
48. Rissanen J. Complexity and information in data. In *Proc. IFAC Conf. System Identification, SYSID 2000*, Santa Barbara, CA, 2000.

49. Diebold FX. *Elements of Forecasting*. South-Western College Publishing: Cincinnati, OH, 1998.
50. Smyth P. Clustering using Monte Carlo cross-validation. In *Proc. 2nd Intl. Conf. Knowl. Discovery & Data Mining (KDD-96)*, Portland, OR, 1996; 126–133.
51. Krieger AM, Green PE. A cautionary note on using internal cross validation to select the number of Clusters. *Psychometrika* 1999; **64**: 341–353.
52. Vortruba J, Volesky B, Yerushalmi L. Mathematical model of a batch acetone-butanol fermentation. *Biotechnol. Bioeng*. 1986; **28**: 247–255.
53. Singhal A. *Pattern Matching in Multivariate Time-Series Data*, Ph.D. thesis, University of California, Santa Barbara, CA, 2002.
54. Johannesmeyer MC, Singhal A, Seborg DE. Pattern matching in historical data. *AIChE J*. 2002; **48**: 2022–2038.