Preprints of Control Conference Africa 2024
Balaclava, Mauritius, 16-17 September 2024

Time-constrained Gaussian mixture model for clustering multi-modal chemical process data

J.F. Olivier* Prof. T.M. Louw**

** Chemical Engineering Department, Stellenbosch University, Stellenbosch, 7600, South Africa (e-mail: 22562206@sun.ac.za).*

*** Chemical Engineering Department, Stellenbosch University, Stellenbosch, 7600, South Africa (e-mail: tmlouw@sun.ac.za).*

Abstract: Two novel unsupervised learning algorithms were developed for improved clustering of multi-modal time series data that are not separable in feature space, which are common characteristics of chemical process data. The algorithms are extensions of the conventional Gaussian Mixture Model (GMM) and K-means clustering. Both algorithms were adapted to account for the time-dependent nature of chemical process data and thus are termed time-constrained GMM (TCGMM) and time-constrained K-means (TCK-means). The algorithms are evaluated using autoregressive time series data with small step changes in the means and variances; a problem that confounds conventional clustering algorithms. In Case Study 1, step changes in the means and variances are implemented at specific time intervals to create two modes. TCGMM outperforms the other algorithms by obtaining a minimum of 85% accuracy in correctly identifying the modes. The TCGMM algorithm is also tested in a second case study where combinations of mean- and variance-shifts are randomly instantiated based on a conditional probability table (CPT). TCGMM outperforms conventional GMM by an average accuracy of 65.4% versus 46.6% and learns the CPT with an average difference in the main diagonal entries (probabilities of remaining in the same mode) of 1.89% and an average difference in the off-diagonal entries (mode transition probabilities) of 0.664%.

Keywords: clustering, TCGMM, TCK-means

1. INTRODUCTION

The use of machine learning to support the monitoring and control of complex multi-modal chemical processes is becoming more common (Choi et al., 2023). Many process monitoring and control systems still rely on models of the process, but with the complexity of processes increasing, accurate process models are seldomly available. The availability of recorded sensor measurements and the subsequent development of data-driven methods in the computer science literature has led to increased interest in data-driven process monitoring and control algorithms (Thomas et al., 2018). According to Fredriksson et al. (2020), 95% of the machine learning algorithms implemented in various applications are supervised, which requires labelling of input data. Most of the data labelling is still performed manually by engineers and Fredriksson et al. (2020) approximates that 80% of the machine learning pipeline consists of data labelling as a preprocessing method for supervised machine learning.

Clustering is an unsupervised learning technique which generates groups of data using only the data itself (Thomas et al., 2018), thereby negating or at least alleviating the data labelling task. Process data are dynamic and autocorrelated; implementing clustering algorithms that assume time independence of data can fail because the data may not form dense, contiguous, separable clusters (Thomas et al., 2018).

Constrained clustering is a sub-field of clustering, where user-knowledge of the process is incorporated into the clustering algorithm to improve performance (Fredriksson et al., 2020). One of the most common methods of incorporating user-knowledge is in the form of introducing pair-wise constraints (Jain, 2010; Lu et al., 2008). Pair-wise constraints can be *must-link* constraints or *cannot-link* constraints (Basu et al., 2009). A *must-link* constraint dictates that pairs of observations connected by the constraint should belong to the same cluster, while *cannot-link* pairs should belong to different clusters, but there is limited work based on automatically learning pair-wise constraints from the data (Jain, 2010). Most approaches implement constrained clustering by modifying the objective function of existing algorithms to encourage the adherence to pair-wise constraints (Basu et al., 2009; Jain, 2010). Several methods have been developed that incorporate pair-wise constraints into the K-means algorithm: Wagstaff et al. (2001) developed a method based on K-means clustering that assigns an observation to its nearest mean that also adheres to the constraints, while Basu et al. (2004) incorporated must-link and cannot-link constraints into the K-means algorithm by adding penalty terms when these constraints are violated. Several methods have also been developed that incorporate pair-wise constraints into the Expectation-Maximation (EM) algorithm (Basu et al., 2009). Shental et al. (2003) constrain the EM algorithm to incorporate must-link constraints and cannot-link constraints by restricting the possible updates made at each iteration of the E-step of the EM algorithm by only summing over assignments which adhere to the given constraints, aiding the algorithm to converge to a solution that satisfies the constraints. However, the above algorithms require a substantial proportion of samples of pair-wise constraints to ensure accurate results. Further, if pair-wise constraints are fixed and observations are continuously added with no new pair-wise constraints, the algorithm reduces to the conventional unsupervised clustering (Basu et al., 2009).

Instead of providing pair-wise constraints to clustering algorithms, in some cases must-link constraints can be implemented indirectly. One example is to encourage temporal

continuity of clusters. When data is sequential and can be modelled as a Markov process, must-link constraints can be automatically obtained by encouraging observations that are temporally proximate to belong to the same cluster (Basu et al., 2009). This paper illustrates the preliminary results of two novel time-constrained clustering algorithms performed on a synthetic auto-regressive data set. The algorithms were developed to leverage the dynamic, autocorrelated nature of process data for improved clustering performance by favouring similar cluster assignments for consecutive data points. The performance of the novel clustering algorithms is evaluated using synthetic autoregressive time series data that pose a challenge to conventional clustering algorithms, namely timeseries data with constant means, but different variances and data with constant variances, but different means. Conventional algorithms fail to accurately cluster the data due to significant data overlap in the feature space as well as the implicit assumption of iid data, but the proposed time constrained algorithms utilise cluster autocorrelation to enhance clustering performance.

2. CLUSTER ANALYSIS AND ALGORITHMS

2.1 K-means algorithm

The K-means algorithm is one of the most widely used clustering algorithms due to its simplicity and efficiency (Thomas et al., 2018). The algorithm minimises an objective function J which consists of the sum of a distance metric d_{nk} between each observation $\boldsymbol{x}_n$ and the assigned cluster centroid $\boldsymbol{\mu}_k$ (1, 2), by varying the cluster assignment vector $\boldsymbol{r}_n$ for each observation. An observation belongs to set $\mathcal{C}_k$ if it is assigned to cluster k, i.e., $r_{nk} = 1$ if $\boldsymbol{x}_n \in \mathcal{C}_k$. The squared Euclidian distance $d_{nk} = \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$ is most commonly used. $J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} d_{nk}$ (1)

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} r_{nk}\boldsymbol{x}_n}{\sum_{n=1}^{N} r_{nk}} = \frac{1}{|\mathcal{C}_k|}\sum_{\boldsymbol{x}_n \in \mathcal{C}_k} \boldsymbol{x}_n \tag{2}$$

Where $|\mathcal{C}_k|$ represents the number of elements in the set. The objective function is minimised by iteratively updating $\boldsymbol{\mu}_k$ using (2) and r_{nk} using (3), until convergence is reached.

$$r_{nk} = \begin{cases} 1 & \text{where} \quad k = \arg\min_k \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

2.2 Gaussian mixture models

Gaussian mixture models assign a probability of each observation belonging to a cluster (Choi et al., 2004), and the observation is assigned to the cluster with the highest responsibility for that observation. The main assumption for GMM is that the clusters are normally distributed, and the feature space can be described by a mixture of normal distributions with the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k as parameters, as in (4). The GMM is learnt using the EM algorithm to iteratively estimate the parameters by maximising the likelihood of the observed data (Xie & Shi, 2012). Using Bayes' rule, the posterior probability of an observation assigned to a particular distribution $p(r_{nk} = 1|\boldsymbol{x}_n)$, also known as the responsibility $\gamma(r_{nk})$, is given by (5). The responsibilities are calculated with fixed means, covariances and mixing coefficients; this is the Expectation step of the EM algorithm. Subsequently fixing the responsibilities and taking the partial derivatives of the log of the likelihood function, the means (6), covariances (7) and mixing coefficients (8) can be updated during the Maximisation step of the EM algorithm. The algorithm repeats until convergence is reached.

$$p(\boldsymbol{x}_n) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4}$$

$$p(r_{nk} = 1|\boldsymbol{x}_n) = \gamma(r_{nk}) = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{5}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma(r_{nk})\boldsymbol{x}_n}{\sum_{n=1}^{N} \gamma(r_{nk})} \tag{6}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{N} \gamma(r_{nk})(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T}{\sum_{n=1}^{N} \gamma(r_{nk})} \tag{7}$$

$$\pi_k = \frac{\sum_{n=1}^{N} \gamma(r_{nk})}{N} \tag{8}$$

2.3 Unsupervised clustering metrics

Clustering metrics are used to assess the performance of clustering algorithms. Extrinsic clustering metrics are supervised because they require the ground truth (i.e., the correct class labels) to assess the accuracy of the clustering, whereas intrinsic clustering metrics are unsupervised because they only require the clustered data; only intrinsic methods are appropriate for hyperparameter optimisation. Popular intrinsic metrics are the Silhouette coefficient and the Davies-Bouldin index (Webb et al., 2022). The Silhouette coefficient and a new intrinsic metric – the sum of cluster transitions – were used in this study to tune the hyperparameters and obtain the best performance for the clustering algorithms, while the confusion matrix (an extrinsic method) was used to assess the quality of the clustering results. The Silhouette coefficient of an observation $\boldsymbol{x}_n$ assigned to cluster $\mathcal{C}_k$ is defined as in (9-11):

$$s(\boldsymbol{x}_n) = \frac{b(\boldsymbol{x}_n) - a(\boldsymbol{x}_n)}{\max\{a(\boldsymbol{x}_n), b(\boldsymbol{x}_n)\}} \tag{9}$$

$$a(\boldsymbol{x}_n) = \frac{\sum_{\boldsymbol{x}_m \in \mathcal{C}_k, \boldsymbol{x}_n \neq \boldsymbol{x}_m} \|\boldsymbol{x}_n - \boldsymbol{x}_m\|}{|\mathcal{C}_k| - 1} \tag{10}$$

$$b(\boldsymbol{x}_n) = \min_{\mathcal{C}_k} \left\{ \frac{\sum_{\boldsymbol{x}_m \in \mathcal{C}_k, \boldsymbol{x}_n \neq \boldsymbol{x}_m} \|\boldsymbol{x}_n - \boldsymbol{x}_m\|}{|\mathcal{C}_k|} \right\} \tag{11}$$

Parameter $a(\boldsymbol{x}_n)$ is the average distance between observation $\boldsymbol{x}_n$ and all other observations assigned to the same cluster (10). Thus, ideally $a(\boldsymbol{x}_n)$ should be as small as possible. Parameter $b(\boldsymbol{x}_n)$ is the minimum of the average distances between observation $\boldsymbol{x}_n$ and all other observations in every other cluster (11). Thus, ideally $b(\boldsymbol{x}_n)$ should be as large as possible (Han et al., 2011).

The sum of cluster transitions (SCT) metric calculates the number of times the cluster assignments change from one observation to the next (12-13). It is expected that mode transitions will be rare events in process data, thus SCT should be low for the current application.

$$\mathrm{SCT} = \sum_{n=1}^{N-1} q_n \tag{12}$$

$$q_n = \begin{cases} 0 & \text{if} \quad \boldsymbol{r}_n = \boldsymbol{r}_{n+1} \\ 1 & \text{if} \quad \boldsymbol{r}_n \neq \boldsymbol{r}_{n+1} \end{cases} \tag{13}$$

3. TIME-CONSTRAINED ALGORITHMS

The goal of the time-constrained algorithms is to discourage different cluster assignments for observations at time steps that are close together, whilst also allowing observations whose time steps are not close to be grouped together as well. To illustrate the point, consider a process which operates in one mode and transitions to another, but returns to the original mode at a later stage (Fig. 1a). The method must encourage clustering of observations close together in time *without unduly penalising* observations that are far apart in time, i.e., it must support clustering of observations into the original mode when the process returns to the original mode at a later time. At first glance, it seems that simply adding time as an additional feature to be clustered will serve as a solution to the problem. However, this approach will not allow observations that are far apart in time to be grouped together, since their difference in time steps will be large (Fig. 1b). The solution must be a constraint that encourages observations with similar time steps to be clustered together, but with an effect that diminishes as differences in time steps grow larger to not obstruct cluster assignments for observations with large differences in time steps.

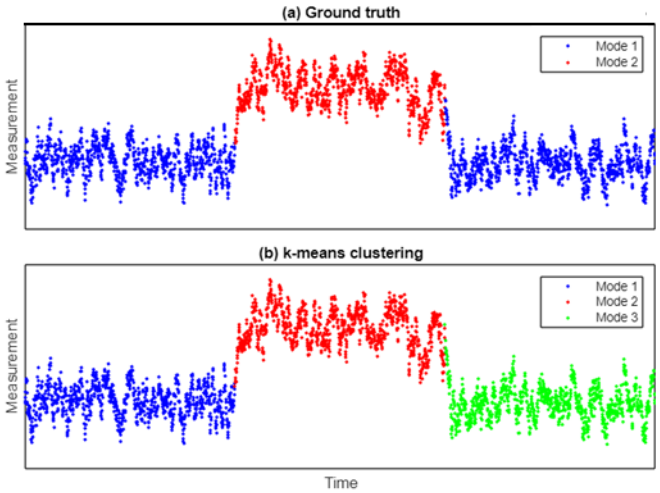

Figure 1: Illustration of incorrect clustering if time is introduced as a feature. a) Ground truth. b) k-means.

3.1 TCK-means algorithm.

The Time-constrained K-means (TCK-means) algorithm solves the problem using an exponential decay function of time included as a weighting parameter α_{nk} defined by (14).

$$\alpha_{nk} = \frac{\sum_{m \in C_k} e^{-\left(\frac{t_n - t_m}{\tau}\right)^2}}{\sum_m e^{-\left(\frac{t_n - t_m}{\tau}\right)^2}}, m \neq n \tag{14}$$

Where τ is the time scale, a hyperparameter to be optimised. The numerator in (14) contains the sum of exponential squared differences in the time steps of observation $\boldsymbol{x}_n$ and all observations $\boldsymbol{x}_m$ assigned to the same cluster in the initial step (2). The denominator in (14) contains the sum of exponential absolute differences in the time steps of observation $\boldsymbol{x}_n$ and all other observation $\boldsymbol{x}_m$, regardless of their initial cluster assignment. The weighting parameter α_{nk} is restricted to $0 \leq a_{nk} \leq 1$, where observations that have many observations that are nearby in time and are assigned to the same cluster will return a value close to 1, but observations that have a small number of observations that are nearby in time and assigned to the same cluster will return a value close to 0. A new distance metric is defined where the squared Euclidean distances to the cluster centroid $\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$ are multiplied by $(1 - \alpha_{nk})$, as in (15). The new cluster centroids are calculated using (3) and the algorithm repeats until convergence is reached. The distance metric promotes clustering observations together if they are nearby in time.

$$d_{nk} = (1 - \alpha_{nk})\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \tag{15}$$

3.2 TCGMM algorithm

The Time-constrained Gaussian mixture model (TCGMM) algorithm does not rely on an alternative distance metric but rather adapts the posterior probabilities or responsibilities of the observations (5) to be conditioned on the cluster assignment of the previous time step, thereby incorporating correlation of cluster assignments between adjacent time steps. To understand the adaptations made to (5), the conditional probability table (CPT) summarised by matrix $\boldsymbol{A}$ is introduced, where the entries of the table contain the probabilities of transitioning between clusters during adjacent time steps (16). If the cluster transitions were represented using a weighted digraph, then $\boldsymbol{A}$ would correspond to the adjacency matrix. The entries of the matrix are the necessary hyperparameters required to calculate the adapted responsibilities, as will be seen in (20). The TCGMM algorithm adapts the responsibilities previously defined in (5) to now be conditioned on the observation at the current time step and the observations from all previous time steps, as in (17). Bayes' rule for expansion and marginalisation of the evidence yields (18).

$$A_{kl} = p\left(r_{n,k} = 1 \middle| r_{n-1,l}\right) \tag{16}$$

$$\gamma(r_{nk}) = p(r_{nk} = 1 | \boldsymbol{x}_n, \boldsymbol{x}_{n-1}, \dots) \tag{17}$$

$$\gamma(r_{nk}) = \frac{\mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) p(r_{nk} = 1 | \boldsymbol{x}_{n-1}, \dots)}{\sum_{j=1}^{K} \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) p(r_{nk} = 1 | \boldsymbol{x}_{n-1}, \dots)} \tag{18}$$

Where, by assuming conditional independence of $\boldsymbol{x}_n$ on $\boldsymbol{x}_{n-1}, \dots$ given r_{nk}, the likelihood $p(\boldsymbol{x}_n | r_{nk} = 1, \boldsymbol{x}_{n-1}, \dots)$ reduces to $p(\boldsymbol{x}_n | r_{nk}) = \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. The conditional independence assumption simply states that the statistical

properties of $\boldsymbol{x}_n$ are sufficiently specified by the cluster assignment, even though the cluster assignment itself may be dependent on the observations at previous time steps.

We can define the time-dependent mixing coefficient $\pi_{nk} = p(r_{nk} = 1|\boldsymbol{x}_{n-1}, \dots)$ to yield (19). The mixing coefficient can be calculated by marginalising over the cluster assignment at the previous timestep, (20).

$$\gamma(r_{nk}) = \frac{\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\pi_{nk}}{\sum_{j=1}^{K}\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\pi_{nj}} \tag{19}$$

$$\begin{aligned}\pi_{nk} &= p(r_{nk} = 1|\boldsymbol{x}_{n-1}, \dots) \\ &= \sum_l p\left(r_{nk} = 1, r_{n-1,l}\middle|\boldsymbol{x}_{n-1}, \dots\right) \\ &= \sum_l p\left(r_{nk} = 1\middle|r_{n-1,l}, \boldsymbol{x}_{n-1}, \dots\right)p\left(r_{n-1,l}\middle|\boldsymbol{x}_{n-1}, \dots\right) \\ &= \sum_l p\left(r_{nk} = 1\middle|r_{n-1,l}\right)\gamma\left(r_{n-1,l}\right)\end{aligned} \tag{20}$$

. The prior or mixing coefficient is now calculated using the responsibilities from the previous time step and the transition probabilities hyperparameters from the CPT, as in (20). The responsibility of cluster 1 at the first time step is initialised as 1 ($\gamma_{1,1} = 1$) with no loss of generality.

$$\pi_{nk} = \sum_l A_{kl}\gamma\left(r_{n-1,l}\right) \tag{20}$$

The CPT is initialised at the start of the algorithm, but can be learned after every iteration of the EM algorithm by calculating the expected value of transitioning from clusters between adjacent time steps (21), which approximates to (22) by applying the law of large numbers.

$$\begin{aligned}A_{kl} &= p\left(r_{ik} = 1\middle|r_{i-1,l} = 1\right) \\ &= \int p(r_{ik} = 1, \boldsymbol{x}|r_{i-1,l})d\boldsymbol{x} \\ &= \int p(r_{ik} = 1|r_{i-1,l}, \boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \\ &= \mathbb{E}_x\left\{p\left(r_{ik} = 1\middle|r_{i-1,l}, \boldsymbol{x}\right)\right\} \qquad (21) \\ &\approx \frac{1}{N-1}\sum_n p(r_{nk} = 1|r_{n-1,l}, \boldsymbol{x}_n) \\ &= \frac{1}{N-1}\sum_n \frac{\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)A_{kl}}{\sum_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)A_{jl}} \qquad (22)\end{aligned}$$

4. CASE STUDIES

4.1 Case Study 1: Isolated mean and variance changes

Synthetic data exhibiting two operating modes were generated using a first order auto-regressive function (23).

$$x_{n+1} = \phi x_n + (1-\phi)\mu_x + \sigma\left[\sqrt{1-\phi^2}\right]\varepsilon \tag{23}$$

Where, ϕ, μ_x, σ and ε are the autocorrelations, means, standard deviations and innovation, respectively. Two data sets consisting of two arbitrary autocorrelated features were simulated for 1000 time steps. In the first data set, the mean was kept constant as a 10% step change in the variance was implemented to migrate from mode 1 to mode 2 at $n = 301$, and back to mode 1 at $n = 601$. In the second data set, the variance was kept constant as a 10% step change in the mean was implemented at $n = 301$ to migrate from mode 1 to mode 2. The data migrated back to mode 1 at $n = 601$ by decreasing the mean to the original value. The magnitudes of the variance and the change in the mean were chosen to exhibit data overlap in the two-dimensional feature space. See Fig. 2.

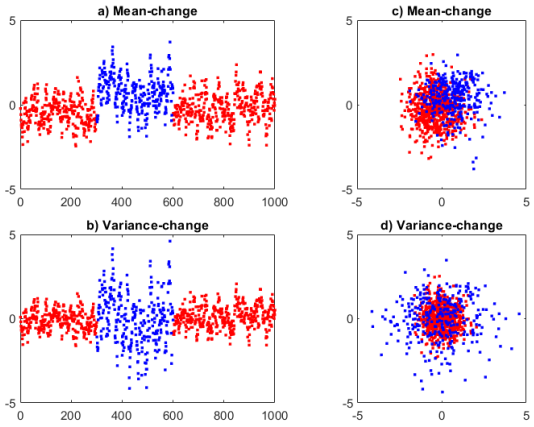

Figure 2: Case Study 1 ground truth: a, b) Time series, c, d) Feature space. (Legend: Mode 1: Red, Mode 2: Blue)

4.2 Case Study 2: Randomised mean and variance changes

The second case study used the same equation (23) to generate the autocorrelated data set, but in this case the mean and variance changes were not implemented during specific intervals, but four combinations of the mean and variance changes were implemented at random time steps. The four combinations or modes are given in Table 1.

Table 1: Randomised modes for Case Study 2.

Modes	**Mean value**	**Variance value**
Mode 1: No change	μ_x	σ^2
Mode 2: Shifted variance	μ_x	$1.1\sigma^2$
Mode 3: Shifted mean	$1.1\mu_x$	σ^2
Mode 4: Shifted mean and variance	$1.1\mu_x$	$1.1\sigma^2$

The modes were randomly simulated by defining a CPT that contained the probabilities of either remaining in the current mode or transitioning to a different mode. The probability of a variance change was denoted as $\alpha_\sigma = 0.01$ and the probability of a mean change was denoted as $\alpha_\mu = 0.01$. The probability of a simultaneous mean and variance change is thus $\alpha_\mu\alpha_\sigma$ and the probability of remaining in the same mode is $\alpha_0 = 1 - \alpha_\mu - \alpha_\sigma - \alpha_\mu\alpha_\sigma$. The CPT is summarised using the network model (Fig. 3) with adjacency matrix $\boldsymbol{A}$ (24). The data set was generated over a period of 10000 time steps to ensure sufficient occurrences of all the modes.

$$\boldsymbol{A} = \begin{bmatrix} \alpha_0 & \alpha_\sigma & \alpha_\mu & \alpha_\sigma\alpha_\mu \\ \alpha_\sigma & \alpha_0 & \alpha_\sigma\alpha_\mu & \alpha_\mu \\ \alpha_\mu & \alpha_\sigma\alpha_\mu & \alpha_0 & \alpha_\sigma \\ \alpha_\sigma\alpha_\mu & \alpha_\mu & \alpha_\sigma & \alpha_0 \end{bmatrix} \tag{24}$$

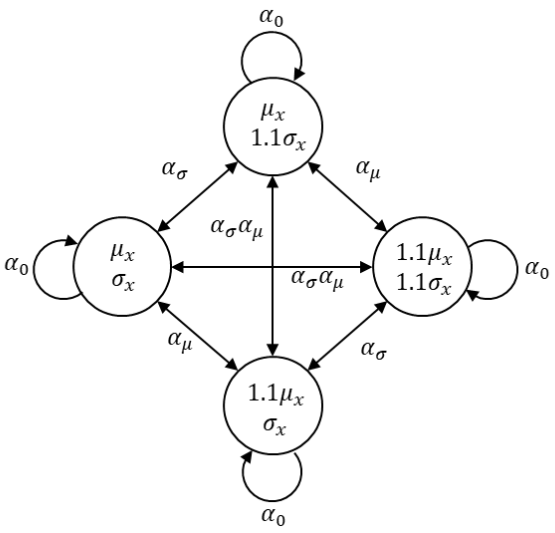

Figure 3: Case Study 2 network model of mode shifts.

4.2 Optimisation of algorithms

It was assumed the number of modes was known *a priori*; thus, it was not necessary to optimise the number of clusters. The optimisation process for both K-means and GMM only consisted of 100 random initialisations. The K-means plus-plus algorithm extension was also used to increase the chance of obtaining the global optimum clustering (Vouros et al., 2021).

The optimisation of TCGMM also consisted of 100 random initialisations. The main diagonal entries of the adjacency matrix were chosen to be $1 - \tilde{\alpha}$ while the off-diagonal entries were chosen to be $\tilde{\alpha} = 10^{-3}$. The matrix was then normalised to ensure that the columns sum to 1. To ensure convergence was reached, the sum of the absolute difference between the means and adjacency matrices between successive loops were required to be smaller than the tolerance value of $\delta = 10^{-10}$. The optimisation of the TCK-means algorithm consisted of 100 random initialisations and optimisation of the time scale τ using the built-in MATLAB® constrained optimisation routine fminbnd. Reasonably upper and lower bounds of the time scale τ used as inputs for fminbnd were manually estimated beforehand. For all algorithms, the Silhouette coefficient and SCT metric were used independently as the objective functions for hyperparameter optimisation and choosing the "best" clustering for each algorithm. After optimising each algorithm, the results were compared to the ground truth by using the Confusion Matrix.

5. RESULTS

5.1 Case Study 1: specified mean and variance changes

The TCK-means algorithm performed slightly better than K-means in identifying mode 1 with 74.7%, but 36.3% for mode 2, whilst K-means obtains 64.7% for mode 1 and 45.7% for mode 2. TCGMM does extremely well, obtaining 97.6% accuracy for mode 1 and 92.7% accuracy mode 2. GMM obtained an accuracy of 97.3% and 44.0% for modes 1 and 2, respectively. The clustering results as time series plots are shown in Fig. 4.

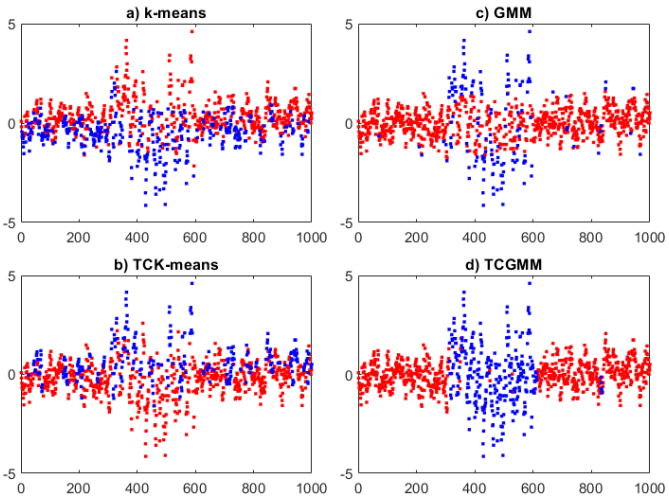

Figure 4: Variance-change clustering: a) K-means, b) TCK-means, c) GMM, d) TCGMM. (Legend: Mode 1: Red, Mode 2: Blue)

TCK-means overall outperforms K-means on the mean-change data set with 72.1% accuracy for mode 1 and 74.7% for mode 2, compared to 64.6% and 77.7% accuracy obtained by K-means for modes 1 and 2, respectively. K-means struggled to correctly cluster some of the noisier data that have values close to the data in the mean-change, whereas the time-dependent nature of TCK-means improved on those clustering. TCGMM once again performs very well on the mean-change data set obtaining 89.3% accuracy for mode 1 and 85.0% for mode 2, while conventional GMM obtained any accuracy of 89.0% and 58.7% for modes 1 and 2, respectively. The clustering results in the feature space are shown in Fig. 5.

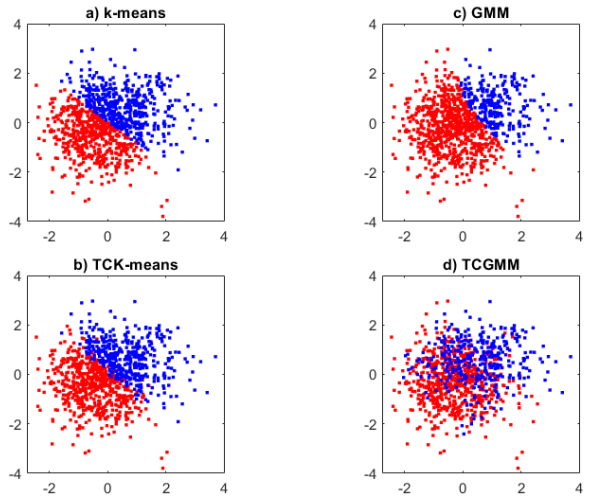

Figure 5: Mean-change clustering: a) K-means, b) TCK-means, c) GMM, d) TCGMM. (Legend: Mode 1: Red, Mode 2: Blue)

5.3 Case Study 2: Randomised mean and variance changes

TCGMM does not obtain quite the same level of accuracies it obtained in Case Study 1, but still outperforms GMM in general by obtaining the following accuracies for modes 1 to 4 respectively: 68.3%, 63.6%, 75.7% and 56.6% for an average accuracy of 65.4%. GMM obtains the following accuracies for modes 1 to 4 respectively: 83.4%, 37.7%, 3.1% and 49.0% for an average accuracy of 46.6%. The average difference in the main diagonal probabilities of the learned CPT from TCGMM

(Fig. 6) and the true CPT is 1.89% and the average difference in the off-diagonal probabilities is 0.664%.

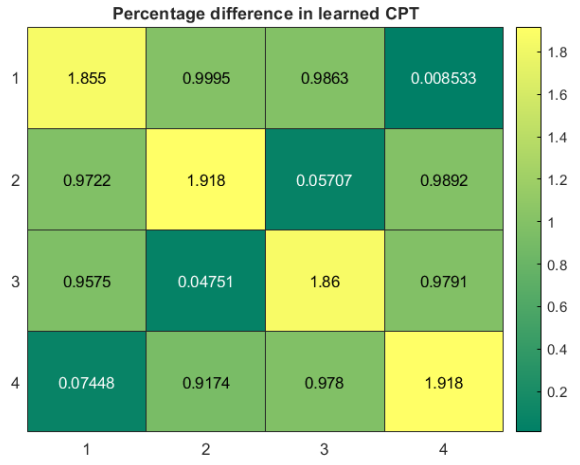

Figure 6: Heatmap of percentage differences between true CPT and the CPT entries learned via TCGMM.

6. CONCLUSIONS

Two novel time-constrained clustering algorithms were developed. Two case studies that exhibit the required behaviour and are difficult to distinguish using conventional clustering methods were simulated. The studies consisted of constant mean, variance-change and constant variance, mean-change data sets and a data set where the periods of the mean and variance changes were chosen randomly, which allowed both changes to be implemented at the same time. The clustering algorithms hyperparameters were optimised using two unsupervised clustering metrics as objective functions, namely the Silhouette coefficient and the sum of cluster transitions. The clustering accuracy was assessed by comparing the results with the ground truth.

The TCGMM algorithm's performance was superior to the GMM algorithm in Case Study 1, never achieving below 85% clustering accuracy. The TCK-means algorithm outperformed the K-means algorithm in the mean-change data set but achieved similar clustering performance in the variance-change data set. The TCGMM algorithm was also superior in Case Study 2, although achieving a lower accuracy of 65.4%. The TCGMM algorithm learned the conditional probability table used to randomly choose the modes; it achieved 1.89% error in the main diagonal entries and 0.664% error in the off-diagonal entries.

Overall, the TCGMM algorithm outperformed all the other algorithms and can learn the probabilities of transitioning between modes. TCGMM is therefore a very promising clustering method for applications in exploratory data analysis and process monitoring. In the future, it will be useful to investigate the performance of this algorithm on a more realistic chemical process simulation such as a CSTR. It will also be useful to include more than two process variables to incorporate dimensionality reduction and to investigate the performance when the number of clusters are not known beforehand.

REFERENCES

Basu, S., Banerjee, A., & Mooney, R. J. (2004). Active semi-supervision for pairwise constrained clustering. *Proc SIAM Intern Conf Data Mining*, 333–344.

Basu, S., Davidson, I., & Wagstaff, K. L. (2009). *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. CRC Press, New York, NY.

Choi, S. W., Park, J. H., & Lee, I. B. (2004). Process monitoring using a Gaussian mixture model via principal component analysis and discriminant analysis. *Comp and Chem Eng*, 28(8), 1377–1387.

Choi, Y., Bhadriaju, B., Cho, H., Lim, J., Han, I. S., Moon, I., Kwon, J. S. Il, & Kim, J. (2023). Data-driven modeling of multimode chemical process: Validation with a real-world distillation column. *Chem Eng J*, 457.

Fredriksson, T., Mattos, D. I., Bosch, J., & Olsson, H. H. (2020). Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies. Lecture Notes in Computer Science, 12562 LNCS, 202–216.

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining. Concepts and Techniques*, Morgan Kaufmann, Waltham, MA

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8), 651–666.

Lu, Z., & Leen, T. K. (2008). Semi-supervised Clustering with Pairwise Constraints: A Discriminative Approach, *PMLR* 2:299-306

Shental, N., Bar-Hillel, A., Hertz, T., & Weinshall, D. (2003). Computing Gaussian Mixture Models with EM Using Equivalence Constraints. Computing Gaussian Mixture Models with EM using Equivalence Constraints. *NIPS*.

Thomas, M. C., Zhu, W., & Romagnoli, J. A. (2018). Data mining and clustering in chemical process databases for monitoring and knowledge discovery. *J Proc Con*, 67, 160–175.

Vouros, A., Langdell, S., Croucher, M., & Vasilaki, E. (2021). An empirical comparison between stochastic and deterministic centroid initialisation for K-means variations. *Machine Learning*, 110(8), 1975–2003.

Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001). Constrained K-means clustering with background knowlegde. *Proc 8th Int Con Machine Learning*, 577–584.

Webb, Z. T., Nnadili, M., Seghers, E. E., Briceno-Mena, L. A., & Romagnoli, J. A. (2022). Optimization of multi-mode classification for process monitoring. *Front chem eng,* 4

Xie, X., & Shi, H. (2012). Dynamic multimode process modeling and monitoring using adaptive gaussian mixture models. *Ind Eng Chem Res*, 51(15), 5497–5505.