International Journal of Machine Learning and Cybernetics (2023) 14:551–566
https://doi.org/10.1007/s13042-022-01649-w

ORIGINAL ARTICLE

A novel adaptive methodology for removing spurious components in a modified incremental Gaussian mixture model

Shuping Sun[1] · Yaonan Tong[1] · Biqiang Zhang[2] · Bowen Yang[3] · Long Yan[4] · Peiguang He[2] · Hong Xu[2]

Received: 1 August 2021 / Accepted: 31 August 2022 / Published online: 17 September 2022

Abstract

Regarding the computational complexity of the update procedure in the fast incremental Gaussian mixture model (FIGMM) and no efficiency for removing the spurious component in the incremental Gaussian mixture model (IGMM), this study proposes a novel algorithm called the modified incremental Gaussian mixture model (MIGMM) which is an improvement of FIGMM, and a novel adaptive methodology for removing spurious components in the MIGMM. The major contributions in this study are twofold. Firstly, a more simple and efficient prediction matrix update, which is the core of the update procedure in the MIGMM algorithm, is proposed compared to that described in FIGMM. Secondly, an effective exponential model (p_{Thv}) related to the number of output components generated in MIGMM, combined with the Mahalanobis distance-based logical matrix (LM), is proposed to remove spurious components and determine the correct components. Based on the highlighted contributions, regarding the removal of spurious components, comparative experiments studied on synthetic and real data sets show that the proposed framework performs robustly compared with other famous information criteria used to determine the number of components. The performance evaluation of IGMM compared with other efficient unsupervised algorithms is verified by conducting on both synthetic and real-world data sets.

Keywords MIGMM · FIGMM · Mahalanobis distance · *LM* · Number of components

1 Introduction

The Gaussian mixture model (GMM), a statistical modeling tool, has been successfully and widely used in pattern recognition and machine learning tasks [1–6] due to its powerful statistical characteristics. Typically, the parameters of GMM are learned in a weight space using the expectation maximization (EM) algorithm. However, since this approach assumes that all demonstrations are available at the learning time, and the number of components M remains fixed after learning. This means that the parameters of the GMM need to be computed using all the previous demonstrations in case a new demonstration is available. Moreover, a GMM with a fixed number of mixture components cannot cope with applications of online learning.

The incremental Gaussian mixture model (IGMM) [7, 8], an online incremental learning algorithm, is very similar to the EM algorithm for Gaussian mixture models [9], as discussed in [10]. IGMM adopts a GMM of distribution components that can be expanded to accommodate new information from an input data point, or reduced if spurious components are identified along the learning process. Based on the maximization of the likelihood of the data, each data point assimilated by the model contributes to the sequential update of the model parameters. The estimated parameters are updated through the accumulation of relevant information from each data point. New points are added directly to existing Gaussian components, or new components are created when necessary, avoiding merge and split operations, much like adaptive resonance theory (ART) [11]. Nevertheless, for N data samples, M Gaussian components and D-dimensions, IGMM has a time complexity of $O(NMD^3)$, which makes the algorithm prohibitively expensive for

✉ Shuping Sun
Shp_sun@yeah.net

1 School of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang 414006, China

2 Department of Information Engineering, Nanyang Institute of Technology, Nanyang 473004, China

3 School of Integrated Circuits, University of Chinese Academy of Science, Beijing 101400, China

4 College of Physics and Optoelectronic Engineering, Shenzhen University, Shenzhen 518061, China

high-dimensional tasks (such as visual tasks) and thus limits its use.

The fast incremental Gaussian mixture model (FIGMM), proposed in [12], is an improvement of the IGMM. To reduce the time complexity of IGMM to $O(NMD^2)$, FIGMM applies the Sherman–Morison formula $(A+xy^{\mathrm{T}})^{-1} = A^{-1} - \frac{A^{-1}xy^{\mathrm{T}}A^{-1}}{1+y^{\mathrm{T}}A^{-1}x}$ to update the precision matrix directly, and the matrix determination Lemma $|A+xy^{\mathrm{T}}| = |A|(1+y^{\mathrm{T}}A^{-1}x)$ to directly update the determination of the precision matrix. FIGMM has been successfully used to study the estimated cost of transport for legged robots [13], and traversability cost for aerial reconnaissance support to ground units [14]. However, the process for updating the precision matrix Σ is too complex to be applied to high-dimensional and large data sets. Moreover, the removing procedure of spurious component k based on the accumulator sp_k combined with the age v_k of component k is validated inefficient, due to not considering the smaller prior probability p_k which acts as the criterion to cancel this smaller effective component [15], and also failing to consider overlapping rates between components which violate the update process rule of incremental Gaussian mixture models [12].

Therefore, to reduce the computational complexity, the precision matrix and its determinant update procedure are proposed to construct the MIGMM; to fully consider the importance of the prior probability p_k and the overlapping rates against the update process of the MIGMM, a novel adaptive methodology, which reflects in twofold: ① an effective exponential model (p_{Thv}) related to the number of output components, used as a threshold value, is proposed to remove component k when $p_k < p_{\mathrm{Thv}}$; ② the logical matrix LM, a novel efficient method used to measure the overlap rates between different components, is proposed to remove spurious components.

In summary, this study proposes a novel algorithm called the modified incremental Gaussian mixture model (MIGMM), which is an improvement of the fast incremental Gaussian mixture model (FIGMM), and a novel adaptive methodology for removing spurious components in the MIGMM. The major contributions in this study are (1) a more simple and efficient prediction matrix update, the core of the update procedure, is proposed compared to that described in FIGMM; (2) an effective exponential model (p_{Thv}) related to the number of output components, combined with the Mahalanobis distance-based logical matrix (LM), is proposed to remove spurious components. The performance compared to those of other efficient unsupervised algorithm is evaluated on both synthetic and real-world data sets.

The remainder of this paper is organized as follows. Section 2 presents the modified incremental Gaussian mixture model including update and create new procedures. Section 3 presents a novel adaptive methodology which includes the threshold value curve model p_{Thv} and logical matrix (LM) for removing spurious components. Section 4 presents an experiment-based evaluation of this proposed methodology compared with other state-of-the-art unsupervised algorithm. Finally, Sect. 5 concludes the paper.

2 Modified incremental Gaussian mixture model

As shown in Fig. 1, the block diagram of the proposed MIGMM algorithm is divided into three procedures: (1) a component update process, based on the Mahalanobis distance criterion; (2) a create new component creation process, which is based on the Mahalanobis distance criterion and uses input point $\mathbf{x}$ when creating a new component. The modified incremental Gaussian mixture model (MIGMM), expressed as $G(\mathbf{x}|\mu_m, \Sigma_m)$, is designed as follows:

$$G(\mathbf{x}|\mu_m, \Sigma_m) = \sum_{m=1}^{M} p_m p_{\mathbf{x},m}, \tag{1}$$

where M is the number of Gaussian mixture components; p_m is the prior probability corresponding to component m that satisfies condition $\sum_{m=1}^{M} p_m = 1$; and $p_{\mathbf{x},m} = \frac{1}{\sqrt{(2\pi)^D|\Sigma_m|}} e^{-\frac{1}{2}d_{\mathbf{x},m}}$ is the probability density function of component m, where $d_{\mathbf{x},m}$ is the so-called squared Mahalanobis distance, which is expressed as shown in Eq. (2):

$$d_{\mathbf{x},m} = \left(\mathbf{x}-\mu_m\right)' \Sigma_m^{-1} \left(\mathbf{x}-\mu_m\right), \tag{2}$$

where μ_m is the mth component mean, and Σ_m is its full covariance matrix. And $d_{\mathbf{x},m}$ is used as a novelty criterion applied when updating the current components (described in procedure 1) and in creating new components (described in procedure 2) as shown in Fig. 1. Furthermore, it is extended to generate the logical matrix (LM) used for removing spurious components (in Sect. 3).

2.1 Updating procedure

Given input $\mathbf{x}$ (a single instantaneous data point), the squared Mahalanobis distance $d_{\mathbf{x},m}$ for component m, $\forall m \in [1, M]$, is first computed by Eq. (2). If subset $K = \{k_i | i = 1, \ldots, \mathrm{I}\} \in [1, \mathrm{M}]$ consisting of the component k_i which satisfies with the inequation $d_{\mathbf{x},k_i} \le \chi^2_{D,1-\beta}$ (the $1-\beta$ percentile of a chi-squared distribution with D degrees-of-freedom, where D is the input dimensionality and β is a user-defined meta-parameter, and $\beta = 0.005$ is chosen in this study except for special instruction.) exists, the probability

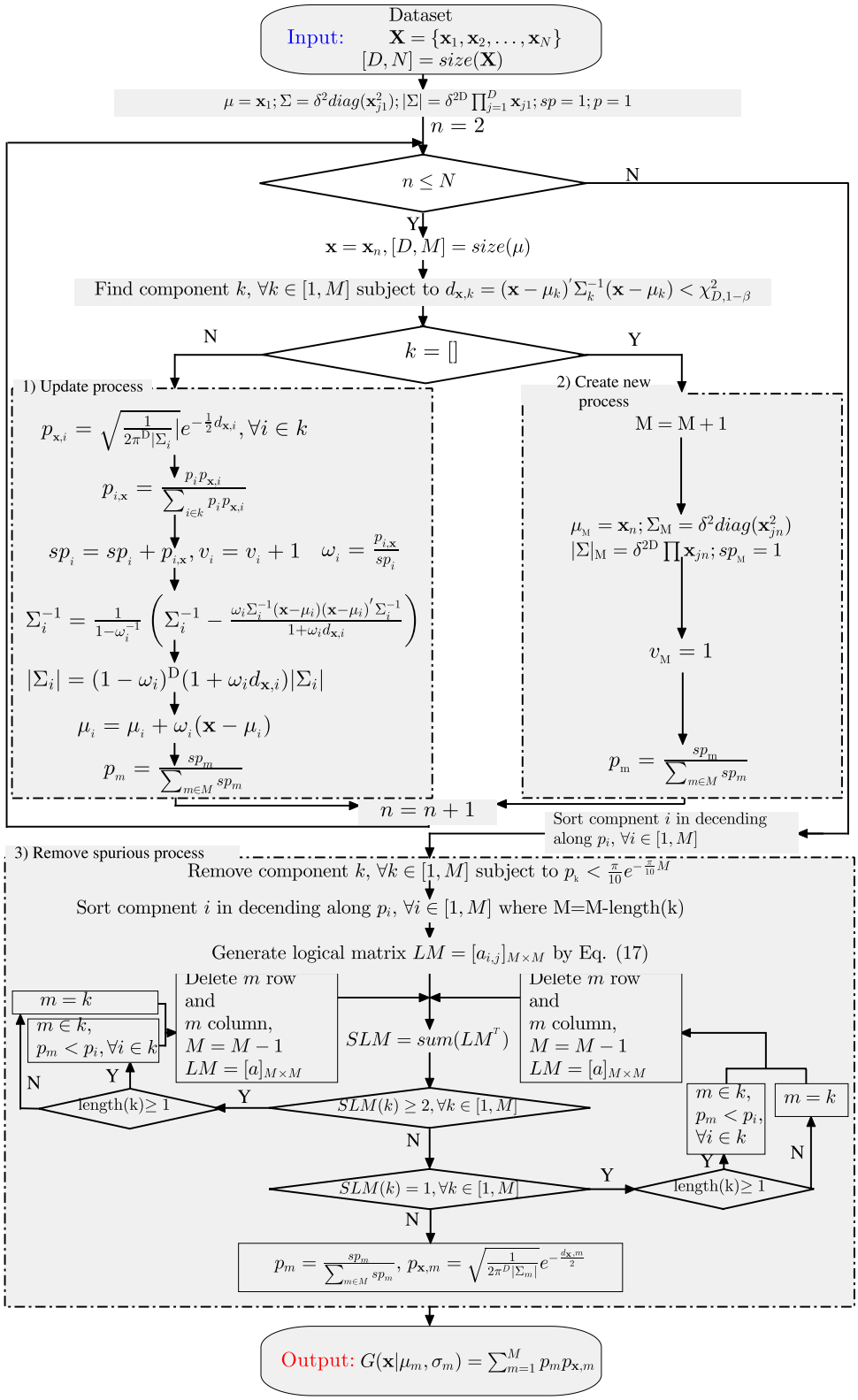


Fig. 1 Block diagram of the proposed modified incremental Gaussian mixture model algorithm

density for the component k_i, denoted as $p_{\mathbf{x},k_i}$, is calculated by

$$p_{\mathbf{x},k_i} = \frac{1}{\sqrt{(2\pi)^D |\Sigma_{k_i}|}} e^{-\frac{1}{2} d_{\mathbf{x},k_i}} \tag{3}$$

where $|\Sigma_{k_i}|$ is the determinant of the estimated covariance matrix Σ_{k_i} corresponding to component k_i. And the posterior probability of the component k_i is calculated by

$$p_{k_i,\mathbf{x}} = \frac{p_{k_i} p_{\mathbf{x},k_i}}{\sum_{k_i \in K} p_{k_i} p_{\mathbf{x},k_i}}, \tag{4}$$

where $K = \{k_i | i = 1, 2, \ldots, \mathrm{I}\}$, a subset of the set $\{1, 2, ..., M\}$, contains the component k_i which satisfies the inequation $d_{\mathbf{x},k_i} \le \chi^2_{D,1-\beta}$. Now, parameters of the algorithm must be updated according to the Eqs. (5–9).

$$sp_{k_i} = sp_{k_i} + p_{k_i,\mathbf{x}} \tag{5}$$

$$\omega_{k_i} = \frac{p_{k_i,\mathbf{x}}}{sp_{k_i}} \tag{6}$$

$$\Sigma^{-1}_{k_i} = \frac{1}{1-\omega^{-1}_{k_i}} \left(\Sigma^{-1}_{k_i} - \frac{\omega_{k_i} \Sigma^{-1}_{k_i} (\mathbf{x} - \mu_{k_i})(\mathbf{x} - \mu_{k_i})' \Sigma^{-1}_{k_i}}{1 + \omega_{k_i} d_{\mathbf{x},k_i}} \right) \tag{7}$$

$$|\Sigma_{k_i}| = (1 - \omega_{k_i})^{\mathrm{D}} (1 + \omega_{k_i} d_{\mathbf{x},k_i}) |\Sigma_{k_i}| \tag{8}$$

$$\mu_{k_i} = \mu_{k_i} + \omega_{k_i} (\mathbf{x} - \mu_{k_i}) \tag{9}$$

$$p_m = \frac{sp_m}{\sum_{m \in [1,M]} sp_m} \tag{10}$$

where the parameters sp_m and p_m are respectively the accumulator and the prior probability of component m, $\forall m \in [1, M]$. The above equations are derived using the Robibins–Monro stochastic approximation [16] for maximum-likelihood of the model. The update procedure is summarized in Algorithm 1.

Algorithm 1: The squared Mahalanobis-based updating procedure

1: Input $\mathbf{x}$, μ, Σ, $|\Sigma|$, p, sp, and β
2: Determine the number of components M = length of sp and number of dimensions D = length of $\mathbf{x}$
3: Calculate the squared Mahalanobis distance $d_{\mathbf{x},m}$, $\forall m \in [1, M]$ by Eq. (2)
4: Find subset $K = \{k_i | i = 1, \ldots, I\}$ where $k_i = \underset{k_i}{arg} d_{\mathbf{x},k_i} \le \chi^2_{D,1-\beta}$
5: **For** each component k_i, $\forall i \in [1, I]$ **Do**
Calculate the Gaussian density for component k_i, $p_{\mathbf{x},k_i}$, by Eq. (3)
Calculate the posterior for component k_i, $p_{k_i,\mathbf{x}}$, by Eq. (4)
Calculate the accumulator posterior for component k_i, sp_{k_i}, by Eq. (5)
Calculate the ratio posterior for component k_i, ω_{k_i}, by Eq. (6)
Iterate
Update the reverse covariance matrix $\Sigma^{-1}_{k_i}$ by Eq. (7)
Update the determinant of covariance matrix Σ_{k_i}, $|\Sigma_{k_i}|$, by Eq. (8)
Update the mean μ_{k_i} by Eq. (9)
End for
6: Recompute the prior probability p_m, $\forall m \in [1, M]$, by Eq. (10)
7: Output μ, Σ, $|\Sigma|$, p, and sp

2.2 Creation new procedure

If the squared Mahalanobis distance $d_{\mathbf{x},m} > \chi^2_{\mathrm{D},1-\beta}$, $\forall m \in [1, M]$, then a new component M where $M = M + 1$ is created and initialized as follows:

$$\Sigma^{-1}_{\mathrm{M}} = \frac{1}{\delta^{2\mathrm{D}}} \begin{pmatrix} \mathbf{x}_{11} & 0 & \ldots & 0 \\ 0 & \mathbf{x}_{21} & \ldots & 0 \\ \vdots & 0 & \ldots & \mathbf{x}_{\mathrm{D1}} \end{pmatrix}^{-2} \tag{11}$$

$$|\Sigma_{\mathrm{M}}| = \delta^{2\mathrm{D}} \prod_{j=1}^{\mathrm{D}} \mathbf{x}^2_{j1} \tag{12}$$

$$\mu_{\mathrm{M}} = \mathbf{x} \tag{13}$$

$$sp_{\mathrm{M}} = 1 \tag{14}$$

$$p_m = \frac{sp_m}{\sum_{m \in [1,M]} sp_m}. \tag{15}$$

According to study [12], δ is set as 1 in this study. Next, the algorithm to create a new component is summarized in Algorithm 2.

Algorithm 2: The squared Mahalanobis-based creating procedure

Input $\mathbf{x}$, μ, Σ, $|\Sigma|$, p, sp, and δ
Redetermine the number of components $M = M + 1$
Create new component M
Specify the reverse covariance matrix Σ_M^{-1} by Eq. (11)
Specify the the determinant of covariance matrix Σ_{M}, $|\Sigma_{\mathrm{M}}|$, by Eq. (12)
Specify the mean μ_{M} by Eq. (13)
Specify the accumulator posterior for component M, sp_{M}, by Eq. (14)
Recompute the prior probability p_m, $\forall m \in [1, M]$, by Eq. (15)
Output μ, Σ, $|\Sigma|$, p, sp

3 Removing spurious components

Regarding the methods to remove spurious components, the prior studies can be summarized as following two branches: one is to remove a spurious component k proposed in FIGMM algorithm [17], by following the criteria $v_k > 5$ and $sp_k < 3$, where sp_k, the accumulator of component k, is calculated by

$$sp_k = sp_k + p_{k,\mathbf{x}}, \tag{16}$$

where $p_{k,\mathbf{x}}$ is the posterior probability of the component k, and v_k is actually the number of samples in component k, respectively, in FIGMM algorithm [17–19]. However, a large number of experimental results show the spurious component k is not removed efficiently by using $v_k > 5$ & sp_k. The other branch uses the Mahalanobis distance to remove the spurious components, as shown in [20] or uses the distance $d_{i,k} = \frac{p_i p_k}{p_i + p_k}(\mu_i - \mu_k)' \Sigma_k^{-1}(\mu_i - \mu_k)$ with the threshold value criterion $\chi^2_{D,0.99}$ as in [21, 22]. To explain the disadvantage of these methods and to discover a more efficient way to remove spurious components, a typical example is illustrated in Fig. 2. Figure 2A shows scatter plots of the input data sets generated from a mixture of nine bivariate Gaussian distributions using the parameters listed in Table 1. The bars of prior probability p_k plotted in Fig. 2B show the results of applying the update and creation procedures of the MIGMM, and these results are sorted in descending order. The mesh and contour plots generated by the updating procedure and the creation procedure of MIGMM are shown in Fig. 2C, D. Moreover, to highlight the superiority of the

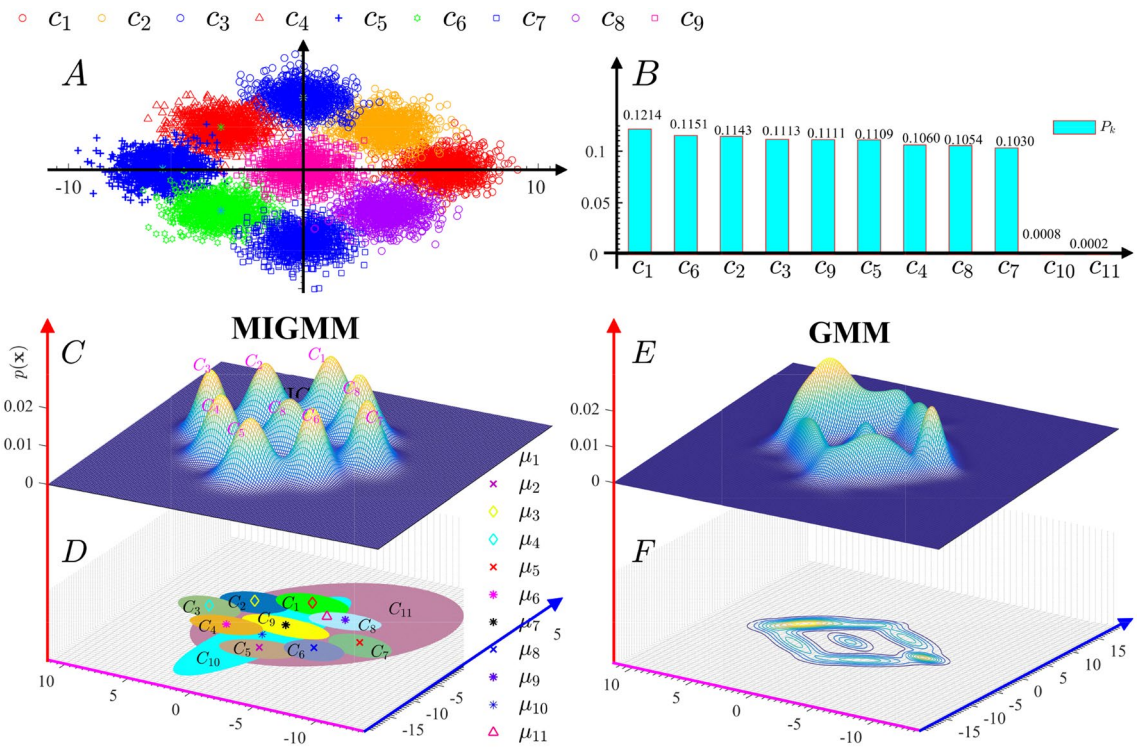


Fig. 2 Typical example for one component including multi-components. **A** represents the scatter plots for the data sets; **B** is the plot of prior probabilities achieved by MIGMM; **C** and **D** are the mesh plots for the results and the confidence intervals corresponding to each component, respectively; and **E** and **F** are the results achieved by the function *fitgmdist* in Matlab 2018b

Table 1 The specified parameters of the nine mixture Gaussian components depicted in Fig. 2

Input a mixture of 9						Output components by GMM and MIGMM							
Data	p	N_k	μ	Σ		p		μ		Σ			
						GMM	MIGMM	GMM	MIGMM	GMM		MIGMM	
1	$\frac{1}{9}$	1000	6	1	0	0.08576	0.1214	6.0540	5.8719	0.8543	0.0256	0.9127	− 0.0131
			0	0	1			0.1663	0.0192	0.0256	0.7472	− 0.0131	0.7932
2	$\frac{1}{9}$	1000	3.5	1	0	0.0933	0.1144	3.4455	3.2783	1.0239	0.0832	0.8077	0.0368
			3.5	0	1			3.6221	3.6189	0.0832	0.9165	0.0368	0.9141
3	$\frac{1}{9}$	1000	0	1	0	0.1119	0.1113	− 0.0394	− 0.2189	0.9977	− 0.0304	0.9113	− 0.1575
			6	0	1			5.9300	5.9052	− 0.0304	0.9728	− 0.1575	1.0347
4	$\frac{1}{9}$	1000	− 3.5	1	0	0.1098	0.1109	− 3.4647	− 3.5835	0.9636	− 0.0258	1.0996	0.0378
			3.5	0	1			3.4795	3.2820	− 0.0258	0.9801	0.0378	0.8016
5	$\frac{1}{9}$	1000	− 6	1	0	0.09194	0.1152	− 6.0056	− 5.9566	0.9441	0.0316	0.9807	0.2296
			0	0	1			0.2178	− 0.2328	0.0316	0.8025	0.2296	0.8169
6	$\frac{1}{9}$	1000	− 3.5	1	0	0.1838	0.1055	− 2.9220	− 3.4044	4.1171	− 2.7625	0.9181	− 0.0711
			− 3.5	0	1			− 3.9476	− 3.6089	− 2.7625	3.2500	− 0.0711	1.1693
7	$\frac{1}{9}$	1000	0	1	0	0.09194	0.1111	4.1492	0.1409	1.7740	− 1.2519	0.9085	− 0.1949
			− 6	0	1			2.2722	− 5.9456	− 1.2519	1.9294	− 0.1949	0.9623
8	$\frac{1}{9}$	1000	3.5	1	0	0.1857	0.1031	2.7867	3.5400	4.0507	2.6981	1.2570	0.1705
			− 3.5	0	1			− 4.0374	− 3.4747	2.6981	3.1476	− 0.0711	1.1693
9	$\frac{1}{9}$	1000	0	1	0	0.1115	0.1061	− 0.0013	− 0.0697	1.0789	− 0.0527	1.1750	− 0.0860
			0	0	1			− 0.0220	− 0.1111	− 0.0527	1.0631	− 0.0860	0.5103
–	–	–	–	–	–	–	0.0008	–	− 2.7007	—		0.0426	0.0931
								–	0.7675			0.0931	0.8763
–	–	–	–	–	–	–	0.0002	–	3.3983	—		0.0366	0.0183
								–	− 2.0686			0.0183	0.0982

MIGMM method proposed in this study, the *fitgmdist* function in MATLAB 2018b is used to return a GMM with $K = 9$ components fitted to the data sets in Table 1. The mesh plots and related contour plots are shown in Fig. 2E, F, respectively, and the analysis results of the parameters generated from MIGMM and GMM are listed in Table 1. Clearly, both Table 1 and Fig. 2C–F) show that the results of the first 9 components generated by MIGMM are closer to the input data sets than those are generated by GMM. However, MIGMM generates some redundant components (c_{10} and c_{11}). Therefore, determining these spurious components and removing them is necessary. Figure 2C, D shows that spurious component 11 (c_{11}) with the smallest p_{11} basically covers other components and spurious component 10 (c_{10}) with a small p_{10} also covers 3 components. However, the spurious components c_{10} and c_{11} would be incorrectly saved when using the distance criterion to remove components. In addition, it is unfortunate that a parameter as important as p_k, which indicates the mixture proportion, has not previously been considered when removing spurious components. Based on the detailed discussion above combined with the large amounts of experimental results by comparing output components with input actual components, the fact indicates that the spurious components generally appear smaller prior probability p_k such as components $c_8 \sim c_{11}$ in Fig. 5 or greater overlapping rates with other components such as c_9 in Fig. 6 or not only smaller prior probability p_k but also greater overlap rates with other components such as c_{12} in Fig. 5. Therefore, the spurious components can be determined and removed by smaller prior probability p_k related with output components number M and greater overlapping rates related with the distance between components. Considering these, two steps proposed to remove the spurious components are summarized as: 1) a novel adaptive methodology, a threshold curve model P_{Thv} depended on the output components number M, is proposed to remove the component k when its corresponding prior probability $p_k \le P_{\text{Thv}}$, and 2) a Mahalanobis distance between components-based novel methodology, logical matrix (LM) used to indicate the overlapping rate, is proposed to remove the components which have greater overlapping rates with other components. The details are summarized in the following subsections 3.1 and 3.2, respectively.

3.1 p_{Thv}-based spurious component deletion

We conducted extensive experiments on N_k random samples generated from a mixture of K bivariate normal distributions with the same covariance [1, 0; 0, 1] in which overlaps occur among mixture components. The parameters of the mixture of K components with $N_k(k = 1, ..., K)$ random samples per component are specified as follows: K, the number of mixture components, ranges from 3 to 10; N_k, the number of samples per component, is successively assigned the values 500, 1000, 1500, and 2000; and p_k, the prior probability, is set to $1/K$. The detailed parameter-related random samples, as input items, are listed on the left side of Table 2. The output number of components M corresponding to the input K components with N_k random samples, obtained by the update process described in Sect. 2.1 and the creation process described in Sect. 2.2, is listed on the right side of Table 2. The generated spurious components by referring to the input components should be deleted when each of them satisfies three conditions: (1) it is far away from any input components; (2) it's p_m is quite small. According to these principles, the output components number M and the biggest p_m among the spurious components are respectively listed in the M and p columns in Table 2. Moreover, according to the error-bar shown in Fig. 3, which plots the average of the biggest p_m named P_{M} versus the output components number M with symmetric error bars which is the computed standard deviations of P_{M}, we found that the behavior of the average of P_{M} with M could be fitted by the exponential equation

$$\hat{P}_{\text{M}} = ae^{b\text{M}}, \tag{17}$$

where the coefficient of b should be a negative, giving rise to the decreasing behavior shown in Fig. 3. To solve the coefficients a and b in Eq. (17), based on the curve fitting toolbox(*cftool*) which is providing a flexible interface where you can interactively fit curves to data and view plots, the coefficients (with 95% confidence bounds)

$$\begin{cases} a = 0.3098 \quad (0.2939, 0.3257) \\ b = -0.3116 \ (-0.3195, -0.3036) \end{cases} \tag{18}$$

are achieved using the Levenberg-Marquardt algorithm, and a very high adjusted R-square of 98.69% is achieved to evaluate the goodness of fit. Furthermore, based on the coefficients in Eq. (18), the new curve model

$$p_{\text{Thv}} = \frac{\pi}{10} \times e^{-\frac{\pi}{10} \times M}, \tag{19}$$

is determined by using $\pi/10$ to replace a and $-b$ for conveniences in computation, and the goodness of fit is evaluated by the adjusted R-square: 98.686%. Moreover, the fit model $0.3098 \times e^{-0.3116\text{M}}$ and the curve model specified by Eq. (19) , shown in Fig. 4, indicate it is reasonable when using p_{Thv} as the curve model for fitting P_{M}. Therefore, the curve model Eq. (19) used as threshold value to remove any component m when $p_m \leq p_{\text{Thv}}$ (as shown in Fig. 5B).

3.2 *LM*-based removal of spurious components

3.2.1 Logical matrix (*LM*) definition.

The logical matrix, denoted as LM, is defined by $LM = [a_{i,j}]_{\text{M}\times\text{M}}$ such that

$$a_{i,j} = \begin{cases} 1, \ (\mu_j - \mu_i)' \Sigma_i^{-1} (\mu_j - \mu_i) \leq \chi^2_{D,1-\beta} \\ 0, \ otherwise \end{cases}, \tag{20}$$

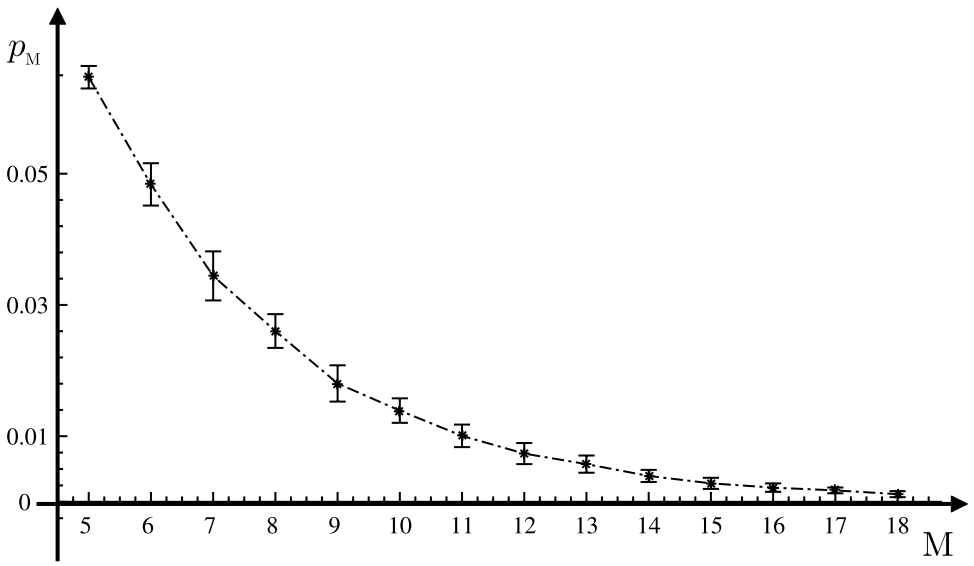


Fig. 3 The errorbar characters of the relationship between P_{M} and M

where $\chi^2_{D,0.95}$ is chosen according to the 95% confidence level, which is a widely used classification criterion in many studies [12, 23–26]. Therefore, $a_{i,j} = 1$ represents the geometric center μ_j of component j located in the confidence region of the ith component, such as $\mu_j(j \neq 3)$ vs component c_{11}, $\mu_j(j = 1, 5, 9)$ vs component c_{10} in Fig. 2D, $\mu_j(j = 3, 4, 5, 6, 7, 9, 10)$ vs component c_{12}, $\mu_i(j = 2, 3, 5, 11)$ vs component c_8 in Fig. 5a. That is, a larger proportion of the samples that construct component j are distributed in component i, which indicates a greater overlapping rates between components i and j. Therefore, LM can be used to remove the spurious components as described below.

3.2.2 Removal procedure

After extensive experimental studies on how the output components relate to the input components of simulated samples, we can draw the conclusions that a component i such that $a_{i,j} = 1$ and length(j) ≥ 2 or a component i such that $a_{i,j} = 1$(length(j) $= 1$) and $p_i \leq p_j$, can be considered spurious and removed. On this basis, the LM-based spurious component deletion algorithm is summarized in Algorithm 3.

Algorithm 3: LM-based spurious components deletion algorithm.

Input:	μ, Σ, $\lvert\Sigma\rvert$, p, sp
	Determine components number M = length(p)
	Resort component $i(i = 1, 2, \ldots, M-1)$ subject to $p_i \geq p_{i+1}$
	Generate logical matrix $LM = [a_{i,j}]_{\mathrm{M}\times\mathrm{M}}$ by Eq. (20)
	Calculate the sum of each column in LM, SLM=sum(LM.').
	While SLM(k) ≥ 2 **Do**
	$LM(k,:) = [], LM(:,k) = []$; $\mu_k = []$;$p_k = []$; $sp_k = []$; $\Sigma_k = []$; $\lvert\Sigma_k\rvert = []$ subject to $p_k \leq p_m(SLM(m) \geq 2)$
Progress:	SLM=sum(LM).
	End
	While SLM(k) $= 1$ **Do**
	LM(k,:)=[],LM(:,k)=[]; $\mu_k = []$;$p_k = []$; $sp_k = []$; $C_k = []$; $dc_k = []$ subject to $p_k \leq p_m(SLM(m) = 1)$
	SLM=sum(LM).
	End
	Recompute the prior probability p_m, $\forall m \in [1, M]$ where M = length(sp), by Eq. (10)
Output:	μ, Σ, $\lvert\Sigma\rvert$, p, sp

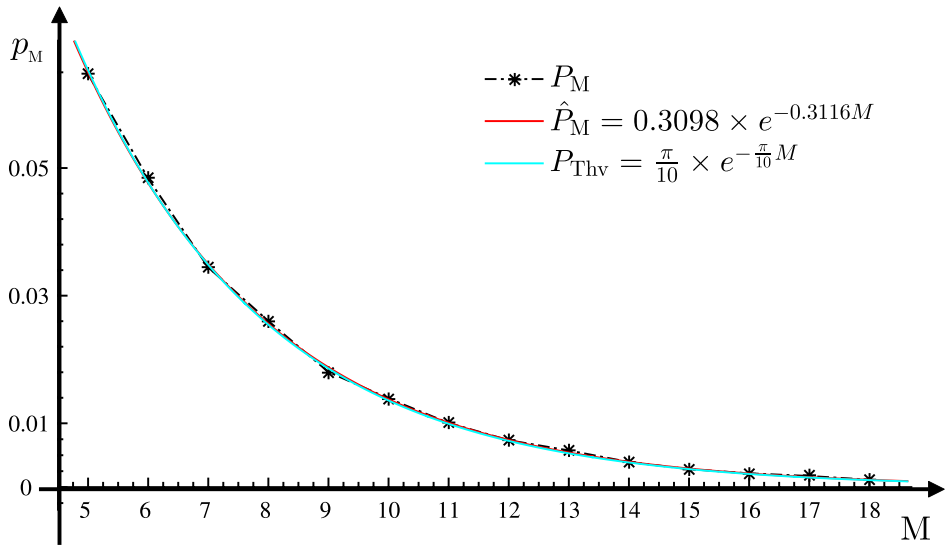


Fig. 4 Comparative analysis curve results for the estimated exponential model by the parameters in Eq. (17) and the specified exponential model by Eq. (19)

Algorithm 4: The spurious component removal procedure

1: Input μ, Σ, $|\Sigma|$, p, sp
2: Determine the components number M = length of p
3: Find component k subject to $p_k \leq \frac{\pi}{10} e^{-\frac{\pi}{10}M}$
4: Remove component k, that is, $p_k = [], C_k = [], sp_k = [], dc_k = [], \mu_k = [], M = M - length(k)$.
5: Remove components based on LM-based spurious components deletion algorithm described in **Algorithm 3**
6: Output μ, Σ, $|\Sigma|$, p, sp

Therefore, based on the threshold value line calculated by Eq. (19), which is related to the number of output components M, and based on LM, the removal procedure algorithm is summarized in Algorithm 4. Furthermore, to better describe our proposed MIGMM, we illustrate typical experiment on simulated samples in Fig. 5. Figure 5A shows a scatterplot of the simulated random samples generated from a mixture of 5 bivariate normal distributions with the same covariance [1, 0; 0, 1], and Table 3 lists the parameters of the mixture of 5 components, with $N_k(k = 1, ..., 5)$ random samples per component. The bars of prior probability p_k plotted in Fig. 5B are the results from applying the update and creation procedures of the

Table 2 Experimental data sets for fitting threshold value line of prior probability

	Input			Output	
Ds	μ		$\frac{SN}{c}(\times 100)$	M	p
Ds_3	0 0 0 5 0 -5	3.5 -3.5 0 0 0 3.5	5	5 5 5 5	0.0652 0.0636 0.0623 0.0643
			10	5 5 5 6	0.0657 0.0669 0.0671 0.0508
	0 1 -1 0 5 -5	3 0 -3 3 0 -3	15	5 5 9 6	0.0659 0.0620 0.0189 0.0482
			20	6 6 11 7	0.0464 0.0450 0.0097 0.0324
Ds_4	2.5 -2.5 -2.5 2.5 2.5 2.5 -2.5 -2.5	4 0 -4 0 0 4 0 -4	5	6 5 8 8	0.0485 0.0634 0.0246 0.0262
			10	5 6 7 6	0.0634 0.0496 0.0273 0.0515
	5 0 0 5 0 0 -5 0	0 0 0 5 0 5 -5 0	15	6 8 9 7	0.0535 0.0234 0.0170 0.0313
			20	7 7 6 7	0.0349 0.0338 0.0515 0.0319
Ds_5	0 -5 0 0 5 5 0 0 -5 0	3.5 -3.5 -3.5 3.5 0 3.5 3.5 -3.5 -3.5 0	5	7 7 7 6	0.0358 0.0375 0.0395 0.0496
			10	8 9 8 8	0.0248 0.0182 0.0290 0.0273
	0 0 0 5 5 5 0 -5 3 -3	5 0 -5 3 -3 3 3 3 -3 -3	15	10 6 6 7	0.0138 0.0450 0.0473 0.0411
			20	10 6 8 7	0.0120 0.0418 0.0315 0.0375
Ds_6	3 -3 3 -3 -3 3 4 4 0 0 -4 -4	5 0 -5 -5 0 5 5 5 5 0 0 0	5	7 7 8 8	0.0331 0.0345 0.0251 0.0264
			10	8 8 8 7	0.0283 0.0290 0.0271 0.0386
	5 0 -5 5 0 -5 2.5 2.5 2.5 -2.5 -2.5 -2.5	5 0 -3 3 0 0 0 5 3 -3 0 5	15	8 10 7 9	0.0237 0.0127 0.0317 0.0195
			20	7 9 10 9	0.0299 0.0225 0.0149 0.0205
Ds_7	5 0 -5 0 -5 0 5 4 5 4 0 -4 -5 -4	0 -5.5 5 0 5 5.5 0 5.5 5.5 0 0 0 -5.5 -5.5	5	8 10 9 9	0.0248 0.0159 0.0172 0.0184
			10	9 11 8 11	0.0193 0.0092 0.0239 0.0129
	0 -3.5 -3.5 0 3.5 3.5 7 -7 3.5 -3.5 0 3.5 -3.5 0	4 0 -3.5 0 -5 0 4 3.5 -1.5 4 0 -3.5 -3.5 -4	15	9 9 10 10	0.0124 0.0145 0.0156 0.0168
			20	12 11 11 13	0.0064 0.0083 0.0094 0.0074
Ds_8	6 4 0 -4 -6 -4 0 4 0 4 6 4 0 -4 -6 -4	5 0 -5 2.5 -2.5 -5 0 5 5 5 5 0 0 -5 -5 -5	5	10 10 15 11	0.0136 0.0147 0.0037 0.0113
			10	12 12 13 13	0.0099 0.0076 0.0067 0.0081
	8 -8 -3 -3 0 0 3 3 0 0 3 -3 8 -8 3 -3	5 -5 -5 -5 0 0 5 5 5 5 0 -5 3 -3 0 -5	15	13 11 11 13	0.0053 0.0081 0.0113 0.0037
			20	12 13 13 12	0.0064 0.0062 0.0046 0.0090
Ds_9	6 0 -6 6 0 -6 6 0 -6 6 6 6 0 0 0 -6 -6 -6	6 3.5 0 -3.5 -6 -3.5 0 3.5 0 0 3.5 6 3.5 0 -3.5 -6 -3.5 0	5	10 13 10 11	0.0122 0.0071 0.0108 0.0106
			10	13 13 12 13	0.0039 0.0058 0.0074 0.0058
	9 -9 -4.5 -4.5 0 0 0 4.5 4.5 0 0 4.5 -4.5 9 0 -9 4.5 -4.5	5 5 5 0 0 0 -5 -5 -5 5 0 -5 5 0 -5 5 0 -5	15	12 11 13 12	0.0053 0.0124 0.0064 0.0055
			20	12 11 16 13	0.0076 0.0106 0.0016 0.0062
Ds_{10}	4 1.5 -1.5 -4 2.5 0 -2.5 -2.5 0 2.5 0 0 0 0 3 4 3 -3 -4 -3	2.5 2.5 0 0 0 0 0 2.5 -3 -2.5 0 2.5 5 2.5 0 -2.5 -4 -2.5 -2.5 0	5	11 11 12 11	0.0108 0.0120 0.0076 0.0092
			10	11 11 12 12	0.0067 0.0085 0.0083 0.0097
	2.5 2.5 0 0 0 0 -3 -4 -4 -3.5 -2 2.5 2 4 -1.5 -4 0 0 2.5 -3	3.5 0 -3 -3 -3 0 2.5 2 0 6 0 0 0 3 -3.5 -3.5 -3.5 3 3 -2.5	15	14 14 14 14	0.0051 0.0035 0.0025 0.0037
			20	15 15 14 14	0.0035 0.0023 0.0051 0.0039
Ds_{11}	2 5 5 5 2 2 -1 -1 -1 -3 -3 0.5 0.5 3.5 -2.5 -2.5 3.5 3.5 0 -2.5 0 -2	-3 -3 -3 0 0 0 2.5 2.5 2.5 5 5 3.5 0 -3.5 3.5 0 -3 3.5 0 -3 2 -1	5	12 13 12 14	0.0064 0.0048 0.0048 0.0044
			10	13 15 14 14	0.0042 0.0025 0.0046 0.0051
	-2 -2 0 0.5 2 1 5 5 4.5 -3 -4.5 2.5 -3 0 2.5 0 -3 -3 0 3 0 3	-4.5 -4.5 -4.5 -1.5 -1.5 -1.5 1.5 1.5 1.5 4 4 -3 0 3 0 3 -3 -3 0 3.5 -1.5 2	15	14 16 15 16	0.0037 0.0021 0.0035 0.0032
			20	15 16 15 16	0.0025 0.0023 0.0016 0.0023
Ds_{12}	5 5 5 2.5 2.5 2.5 0 0 0 -3 -3 -3 -6 4 0 0 4 -4 -4 0 4 4 0 4	-5.5 -5.5 -5.5 -2 -2 -2 1 1 1 4 4 4 4 0 -4 4 0 -4 4 0 -4 4 0 -4	5	14 14 15 15	0.0028 0.0044 0.0028 0.0042
			10	14 15 15 17	0.0025 0.0023 0.0035 0.0014
	-6 -4 0 1.5 4 4 4 1.5 1.5 -1 -1 -3.5 -4 -4 -4 -4 -4 0 4 4 0 0 3.5 4	0 0 0 0 3.5 3.5 4 4 -3.5 -3.5 -3.5 -3.5 6 2.5 -1.5 -5 6 2.5 -1.5 -6 6 2.5 -1.5 -6	15	16 15 17 18	0.0014 0.0014 0.0023 0.0016
			20	17 17 18 18	0.0014 0.0019 0.0012 0.0007

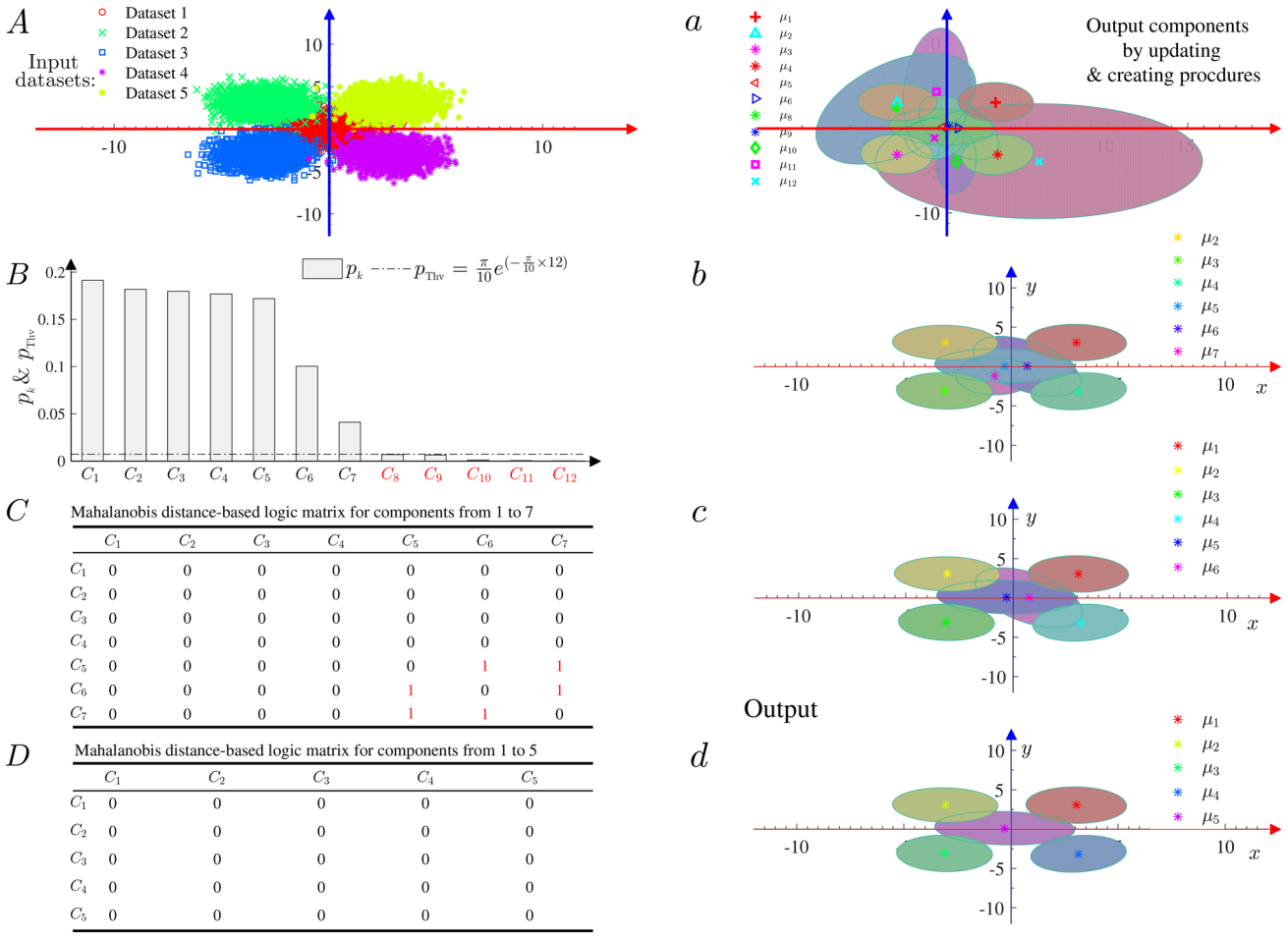


Fig. 5 The spurious component removal procedure

proposed MIGMM. The confidence region corresponding to each component generated by the update and creation procedures of MIGMM is shown in Fig. 5a. Based on the threshold line calculated by Eq. (19) and plotted in Fig. 5B, the components from 1 to 7 are preserved; their confidence regions are shown in Fig. 5b. Then, (1) the logical matrix $LM = [a_{i,j}]_{7\times7}$ is generated by Eq. (20) and shown in Fig. 5C; and (2) component 7 is removed based on the geometric centers of components 5 and 6, which are included in the confidence region of component 7, and the contour plots of the remaining components from 1 to 6 are shown in Fig. 5c. Finally, (1) the logical matrix $LM = [a_{i,j}]_{6\times6}$, generated by removing row 7 and column 7 in $LM = [a_{i,j}]_{7\times7}$, is shown in Fig. 5C; and (2) components 1 to 5 are saved by removing component 6 based on the rule described in Algorithm 3. The contour plots of the remaining components (1–5) are shown in Fig. 5d, and the specified output parameters of components 1–5 are shown in Table 3.

4 Experiments

To evaluate the performance of the proposed MIGMM, we conducted a comparison analysis with other state-of-the-art unsupervised algorithms on simulated bivariate simulation data sets and real multi-dimensional data sets. The results are verified in the following subsections from two aspects.

4.1 Efficiency evaluation on the number of components

A simulated bivariate dataset with 8 components whose parameters are shown in Table 4 was successfully used to evaluate the performance of the SWRLCF method proposed in study [27] by comparing it with other typical criteria, including the Bayes information criterion (BIC) [28], Akaike information criterion (AIC) [29], stepwise split-and-merge EM (SSMEM) [30] and the Akaike information criterion with a penalty of 3 (AIC3) [31]. We adopted this dataset to evaluate the performance of the MIGMM method proposed

Table 3 The specified parameters of five mixture Gaussian components depicted in Fig. 5

Input a mixture of 5						Output components			
– Data set	p	N_k	μ	Σ		p	μ	Σ	
1	$\frac{1}{5}$	2000	3	1	0	0.2122	3.0469	1.0846	0.0399
			3	0	1		3.0508	0.0399	1.1145
2	$\frac{1}{5}$	2000	− 3	1	0	0.2016	− 3.0697	0.9906	0.0498
			3	0	1		3.0548	0.0498	1.2499
3	$\frac{1}{5}$	2000	− 3	1	0	0.1993	− 3.1036	1.1987	0.0611
			− 3	0	1		− 3.0975	0.0611	1.1182
4	$\frac{1}{5}$	2000	3	1	0	0.1961	3.1560	1.2263	− 0.1611
			− 3	0	1		− 3.1423	− 0.1611	1.1053
5	$\frac{1}{5}$	2000	0	1	0	0.1908	− 3.1036	1.2702	0.0173
			0	0	1		0.0368	0.0173	1.0507

Table 4 A simulated dataset of eight mixture Gaussian components

Data set	p	μ		Σ
1	0.1	17	2	[1 0; 0 2]
2	0.1	1	4	[1 0; 0 4]
3	0.1	− 2	12	[2 0; 0 1]
4	0.1	21	14	[4 0; 0 3]
5	0.1	8	9	[3 0; 0 1]
6	0.1	3	17	[2 0; 0 1]
7	0.2	12	23	[3 0; 0 3]
8	0.2	− 1	25	[1 0; 0 1]

in this study. In this dataset, the number of samples varies from 1000, to 5000, 10,000, 15,000, 20,000, 25,000, 30,000 and 50,000. The final number of components M determined by each method is reported in Table 5. For the first four methods, BIC, AIC, AIC3 and SSMEM, Table 5 shows the same results as those reported in study [27], that is, (1) for N = (15,000, 20,000 and 25,000), the first four methods return the overfitting number of components $M = 9$; and (2) when $N = 50{,}000$, they even return overfitted values of k(k=35, 25, 14, 13), respectively. Although the SWRLCF method returns the overfitting number of components $M = 9$ when N=50,000, the initial component number (ICN) must be subject to ICN > 9, as introduced in study [27]. However, our MIGMM method generates all correct numbers $M = 8$ even when number of samples is $N = 50{,}000$. Furthermore, to highlight the efficiency of MIGMM, the process of handling the dataset in Table 4 with 50,000 samples is illustrated in Fig. 6. Figure 6A shows a scatterplot of the simulated random samples generated from a mixture of 8 bivariate normal distributions; their parameters are described in Table 3. The confidence region corresponding to each component generated by the updating and creation procedures of MIGMM is shown in Fig. 6a and the bars of prior probability p_k plotted in Fig. 6B are the results from the update and creation procedures of the proposed MIGMM. Based on the threshold line calculated by Eq. (19) and plotted in Fig. 6B, the components from 12 to 17 are removed first; their confidence regions are shown in Fig. 6b. Then, (1) the logical matrix $LM = [a_{i,j}]_{11\times11}$ is generated by Eq. (20) and shown in Fig. 6C; and (2) component 11 (c_{11}) is removed based on the geometric centers of components 2 and 5, which are included in the confidence region of component 11. Similarly, component 9 is removed, and the contour plots of the remaining components 1–10 (excluding 9) are shown in Fig. 6c. Finally, (1) the logical matrix $LM = [a_{i,j}]_{8\times8}$, generated by removing row 10 and column 10 in $LM = [a_{i,j}]_{9\times9}$, is shown in Fig. 6D; and (2) Components 1–8 are preserved by removing component 10 based on the rule described in Algorithm 3. The contour plots of the remaining components 1–8 are shown in Fig. 6d, and the specified output parameters of components 1–8 are listed in Table 5. Clearly, our

Table 5 The number of components M for different sizes of N obtained by AIC, AIC3, BIC, SSMEM, and MIGMM

Method	1000	5000	10,000	15,000	20,000	25,000	30,000	50,000
AIC	8	8	8	9	9	30	30	35
AIC3	8	8	8	9	9	9	25	25
BIC	8	8	8	9	9	9	10	14
SSEMM	8	8	8	9	9	9	10	13
SWRLCF	8	8	8	8	8	8	8	10
MIGMM	8	8	8	8	8	8	8	8

method is more efficient at determining the correct number of components than are the other methods(AIC, AIC3, BIC, SSMEM, and SWRLCF) for large-sample applications, and it avoids overfitting. Furthermore, our method also avoids the underfitting produced by the other methods listed in Table 5 when the ICN does not exceed the real number of components.

4.2 Classification and computational complexity evaluation

4.2.1 Classification performance evaluation

Because the MIGMM is an improvement of the FIGMM proposed in study [17], its classification performance was evaluated on 10 standard data sets [32, 33], as shown in Table 6, by comparing it with FIGMM and other well-known unsupervised learning methods, including the random forest (RF), K-Means, GMM, IGMM and FIGMM. The classification accuracies achieved in study [17] are recorded in Table 6. The parameters for MIGMM are set to $\delta = 1$ and $\beta = 0.005$ to ensure that Gaussian components are created only when strictly necessary, and the performance evaluation results were obtained using the 10-fold cross-validation (where the training sets included 90% of the data and the test sets included the remaining 10%) method [34]. The experimental results in Table 6 show that (1) On average, the MIGMM algorithm outperforms the FIGMM, which might be because the number of components obtained by MIGMM is smaller than the number of components obtained by the FIGMM algorithm; (2) The output components number obtained by MIGMM algorithm is similar to those generated by GMM, and is much less than those generated by IGMM or FIGMM. Therefore, the efficiency of our proposed removing spurious components is evaluated; (3) MIGMM can also be applied well in high-dimensional data by the comparative analysis results from standard 3072-dimensional data set in Table 6. Besides, MIGMN learns incrementally from each data point.

4.2.2 Computational complexity comparison

Actually, the essential differences between MIGMM and FIGMM [17] are the iterations of the reverse covariance matrix Σ_k^{-1} and the determination of the reverse covariance matrix $|\Sigma|_k^{-1}$. The equations regarding the computations of Σ_k^{-1} and $|\Sigma|_k^{-1}$ were rewritten as follows: (1) In FIGMM, Σ_k^{-1} and $|\Sigma|_k^{-1}$, transformed from Eqs. (20–21) on page 5 and Eqs. (25–26) on page 6 in [17], which appear as Eq. (21) in study [17], are rewritten as shown in

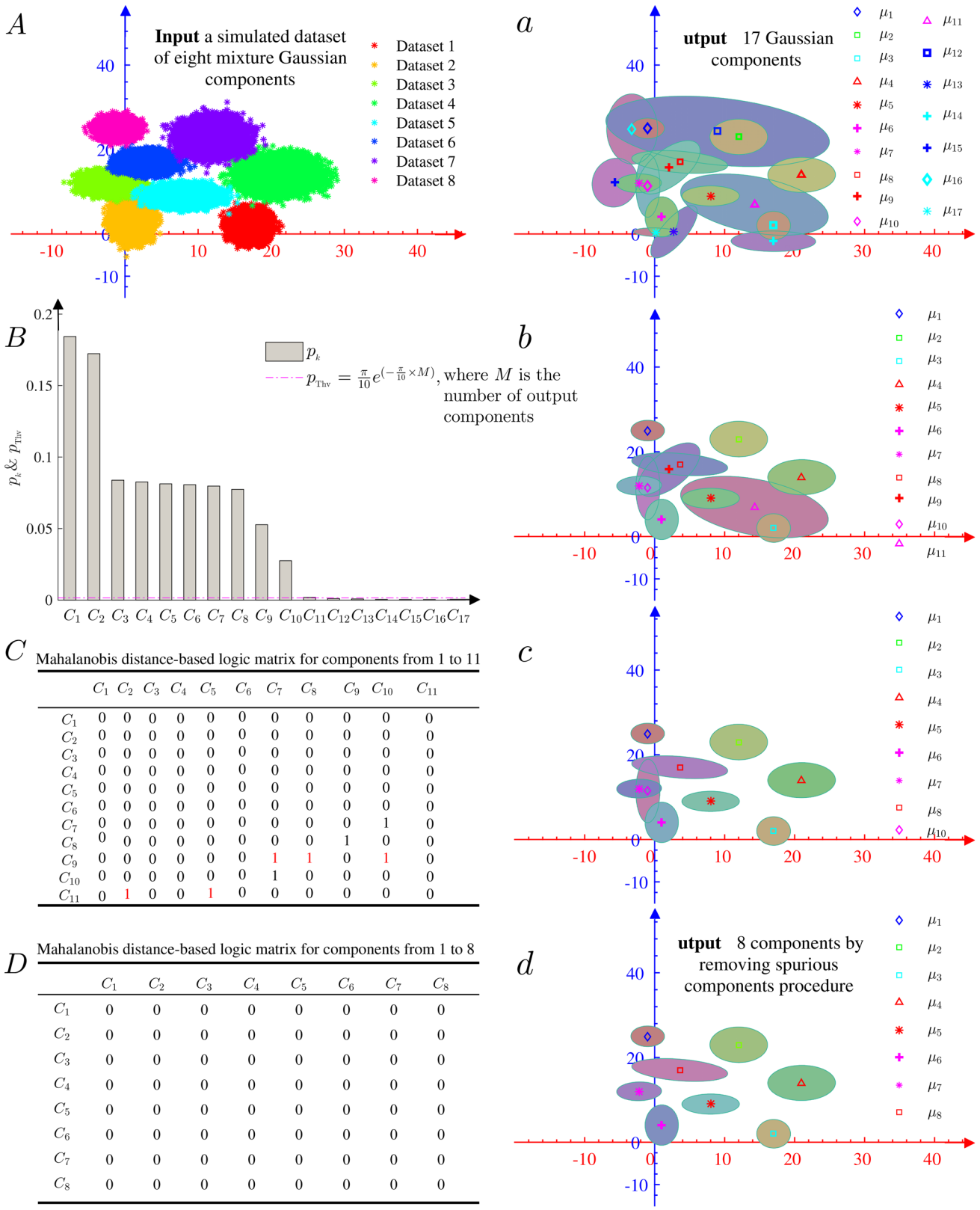
$$\begin{cases} \bar{\Sigma}_k^{-1} = \frac{\Sigma_k^{-1}}{1-\omega_k} - \frac{\frac{\omega_k}{(1-\omega_k)^2}\Sigma_k^{-1}e^*e^{*T}\Sigma_k^{-1}}{1+\frac{\omega_k}{1-\omega_k}e^{*T}\Sigma_k^{-1}e^*}, \text{where} \quad e^* = \mathbf{x} - \mu_k \\ \Sigma_k^{-1} = \bar{\Sigma}_k^{-1} + \frac{\bar{\Sigma}_k^{-1}\Delta\mu_k\Delta\mu_k^T\bar{\Sigma}^{-1}{}_k}{1-\Delta\mu_k^T\bar{\Sigma}_k^{-1}\Delta\mu_k}, \text{where} \quad \Delta\mu_k = \omega_k(\mathbf{x} - \mu_k) \\ |\bar{\Sigma}_k^{-1}| = (1-\omega_k)^{-D}\frac{|\Sigma|_k^{-1}}{\left(1+\frac{\omega_k}{1-\omega_k}e^{*T}\Sigma_k^{-1}e^*\right)} \\ |\Sigma|_k^{-1} = \frac{|\bar{\Sigma}_k^{-1}|}{1-\Delta\mu_k^T\bar{\Sigma}^{-1}{}_k\Delta\mu_k} \end{cases}, \tag{21}$$

and (2) In MIGMM, Σ_k^{-1} and $|\Sigma|_k^{-1}$, which are calculated by Eqs. (7–8) in this study, are rewritten as Eq. (22):

$$\begin{cases} \Sigma_k^{-1} = \frac{1}{1-\omega_k}\left(\Sigma_k^{-1} - \frac{\omega_k\Sigma_k^{-1}(\mathbf{x}-\mu_k)(\mathbf{x}-\mu_k)'\Sigma_k^{-1}}{1+\omega_k d_{\mathbf{x},k}}\right) \\ |\Sigma|_k^{-1} = \frac{|\Sigma|_k^{-1}}{(1-\omega_k)^{\mathrm{D}}(1+\omega_k d_{\mathbf{x},k})} \end{cases}. \tag{22}$$

As the above equations show, the complexity of MIGMM is less than that of FIGMM. Moreover, to evaluate the time efficient between our method MIGMM and FIGMM with the parameters $\delta = 1$ and $\beta = 1$, such that a single component was created and we could focus on speedups due only to dimensionality. They were conducted on the 3 highest dimensional data sets in Table 6, namely, the Gesture, MNIST, and CIFAR-10 datasets. The Gesture data set was partitioned into a training set wit 9000 data points and a testing set containing 900 data points [33], the MNIST dataset was split into a training set with 60000 data points and a testing set containing 10000 data points according to study [37], and the CIFAR-10 dataset was split into 50, 000 training data points and 10, 000 testing data points, a standard procedure for this data set [38]. The comparative results summarized in Table 7 showed that a bigger difference could be generated as the dimensions or sample size increase, and the saving time affected by dimensions was greater than the sample size, such as for the 784-dimensional MNIST data set, the saving time was from 6.22×10^4 s to 100.9×10^4 s corresponding to the sample size from 10, 000 to 60, 000. However, for the 10,000 sample size 784-dimensional MNIST and 3072-dimensional data sets, the saving time from 6.22×10^4 s to 13.97×10^4 s. It makes sense that the running time is related to the square of dimensions D^2 because there exists the iteration calculation of covariance $\Sigma_{D\times D}$ in Eq. (1).

In addition, to verify the scalability of MIGMM algorithm, both the FIGMM and MIGMM algorithms with $\delta = 1$ and $\beta = 0$ were compared on 10 random data sets generated by the *rand* function in MATLAB 2018b. All data sets have 1000 data points drawn from a single uniformly random number in the interval (0, 1) and an exponentially



C Mahalanobis distance-based logic matrix for components from 1 to 11

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
C_1	0	0	0	0	0	0	0	0	0	0	0
C_2	0	0	0	0	0	0	0	0	0	0	0
C_3	0	0	0	0	0	0	0	0	0	0	0
C_4	0	0	0	0	0	0	0	0	0	0	0
C_5	0	0	0	0	0	0	0	0	0	0	0
C_6	0	0	0	0	0	0	0	0	0	0	0
C_7	0	0	0	0	0	0	0	0	0	1	0
C_8	0	0	0	0	0	0	0	0	1	0	0
C_9	0	0	0	0	0	0	1	1	0	1	0
C_{10}	0	0	0	0	0	0	1	0	0	0	0
C_{11}	0	1	0	0	1	0	0	0	0	0	0

D Mahalanobis distance-based logic matrix for components from 1 to 8

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	0	0	0	0	0	0	0	0
C_2	0	0	0	0	0	0	0	0
C_3	0	0	0	0	0	0	0	0
C_4	0	0	0	0	0	0	0	0
C_5	0	0	0	0	0	0	0	0
C_6	0	0	0	0	0	0	0	0
C_7	0	0	0	0	0	0	0	0
C_8	0	0	0	0	0	0	0	0

Fig. 6 An illustration of the dataset in Table 4 with 50,000 samples

Table 6 Accuracy of different algorithms and the number of Gaussian components created on standard data sets

Experimental data sets				Accuracies obtained (%)						Components number obtained			
Name	N	D	K	RF	K-means	GMM	IGMM	FIGMM	MIGMM	GMM	IGMM	FIGMM	MIGMM
Breast-cancer	286	9	2	69.6 ± 9.1	71.3 ± 5.5	69.3 ± 3.5	71.4 ± 7.4	71.4 ± 7.4	73.2 ± 6.1	10.3 ± 0.9	14.2 ± 1.9	14.2 ± 1.9	10.7 ± 1.1
Pima-diabetes	768	8	2	75.8 ± 3.5	78.5 ± 6.1	77.5 ± 3.1	73.0 ± 4.5	73.0 ± 4.5	73.4 ± 3.9	5.4 ± 2.2	19.4 ± 1.3	19.4 ± 1.3	5.3 ± 2.1
Glass	214	9	7	79.9 ± 5.0	66.7 ± 4.9	62.7 ± 3.2	65.4 ± 4.9	65.4 ± 4.9	69.5 ± 5.7	9.1 ± 1.3	15.9 ± 1.1	17.3 ± 1.5	8.9 ± 1.3
Ionosphere	351	34	2	92.9 ± 3.6	81.7 ± 5.3	91.3 ± 4.9	92.6 ± 3.8	92.6 ± 3.8	93.9 ± 3.7	67.1 ± 1.7	78.6 ± 4.7	82.7 ± 5.5	65.9 ± 1.5
Iris	150	4	3	95.3 ± 4.5	90.3 ± 6.1	96.7 ± 3.7	97.3 ± 3.4	97.3 ± 3.4	93.1 ± 4.3	2.1 ± 0.3	2.7 ± 0.7	2.7 ± 0.7	2.1 ± 0.3
Labor-neg	57	16	2	89.7 ± 14.3	86.2 ± 10.3	93.1 ± 4.8	94.7 ± 8.6	94.7 ± 8.6	96.8 ± 3.2	6.9 ± 1.3	12.0 ± 1.2	12.0 ± 1.2	7.1 ± 1.5
Soybean	683	35	19	93.0 ± 3.1	91.2 ± 6.4	93.9 ± 4.1	88.7 ± 3.0	91.5 ± 5.4	95.2 ± 4.3	40.3 ± 7.0	42.6 ± 2.2	43.3 ± 2.2	39.1 ± 6.9
Gesture [33]	9900	32	5	34.6 ± 3.9	33.12 ± 2.1	49.3 ± 5.6	41.2 ± 3.2	44.1 ± 4.1	52.1 ± 3.9	9.3 ± 2.1	15.6 ± 7.6	12.6 ± 4.9	10.5 ± 3.1
MNIST [35]	70,000	784	10	83.1 ± 6.3	73.8 ± 6.2	73.6 ± 4.7	68.5 ± 3.9	69.7 ± 4.2	80.1 ± 6.3	21.3 ± 3.6	25.6 ± 6.6	23.2 ± 3.9	22.6 ± 5.3
CIFAR-10 [36]	60,000	3072	10	63.6 ± 6.5	70.3 ± 6.9	73.2 ± 5.6	69.3 ± 6.8	70.2 ± 7.1	75.6 ± 7.1	15.3 ± 3.1	19.6 ± 6.6	17.6 ± 5.2	13.9 ± 2.9
Average				77.75	74.312	78.06	76.21	76.99	80.29	18.71	24.62	24.5	18.61

Note: The components number, corresponding to the accuracy when using GMM, is achieved by the characters described in Data sets combined with stationary character of accuracy. The MNIST data set is available at (http://yann.lecun.com/exdb/mnist) and the CIFAR-10 data set is available at (http://www.cs.toronto.edu/~kriz/cifar.html)

growing number of dimensions: 4, 8, 16, 32, 64, 128, 256, 512, 1024, and 2048. This experiment was performed for comparing the scalability of both algorithms. Results plotted in Fig. 7 show that the difference between FIGMM and MIGMM become more obvious with growing number of dimensions, such as the difference is only 15.4 s corresponding to the dimensions of 512, the difference extents 65.9 s corresponding to the dimensions of 1024, and while the difference is 189.6 s as the dimensions of 2048. The computer used for this experiment was equipped with a Windows 10 Ultimate (64-bit) operating system, an Intel(R) Core(TM) i7-8700 CPU@3.20GHz, 256 GB of RAM, and MATLAB version: 2018b.

5 Conclusion

In this study, we proposed a novel algorithm called the modified incremental Gaussian mixture model (MIGMM) which is an improvement of the fast incremental Gaussian mixture model (FIGMM), and a novel adaptive methodology for removing spurious components in the MIGMM. The major contributions in this study are twofold. First, a more simple and efficient prediction matrix update, which is the core of the update procedure in the MIGMM algorithm, is proposed compared to that described in the FIGMM. Second, an effective exponential model ($p_{\text{Thv}} = \frac{\pi}{10}e^{-\frac{\pi}{10}M}$) related to the number of output components generated in the update and creation procedures in the MIGMM, combined with the Mahalanobis distance-based logical matrix (LM), is proposed to remove spurious components and determine the correct components. To demonstrate the above contributions, (1) comparative experiments on simulated synthetic and real data were conducted, revealing that the proposed framework achieves robust performances when determining the number of components compared with other well-known methods, such as AIC AIC3, BIC, SSEMM and SWRLCF; and (2) the classification performance was evaluated via 10-fold cross-validation on real-world data sets by comparing MIGMM to other well-known unsupervised learning algorithms, including random forest, GMM, IGMM and FIGMM. Generally, the results show that the MIGMM is a good option for supervised learning from a single pass through the data, that it has short runtimes, ant that it achieves good accuracy.

Merging components to be used as one cluster should be further studied in the future; this could further increase the recognition accuracy, especially in cases with multidimensional features that do not satisfy normal distributions. Furthermore, how to apply β and δ, and how to apply MIGMM for supervised learning using a double pass through the data also warrant further research.

Table 7 Running time for train and test process for the MIGMM and FIGMM algorithm on the larger data sets

Data set			Training			Testing		
Name	Samples	Dimensions	Samples	Running time ($\times 10^4$)		Samples	Running time ($\times 10^4$)	
				FIGMM	MIGMM		FIGMM	MIGMM
Gesture	9900	32	9000	0.1127	0.1001	900	0.01051	0.0087
MNIST	70,000	784	60,000	484.87	383.97	10,000	14.31	8.09
CIFAR-10	60,000	3072	50,000	4125.42	3930.38	10,000	165.28	151.31

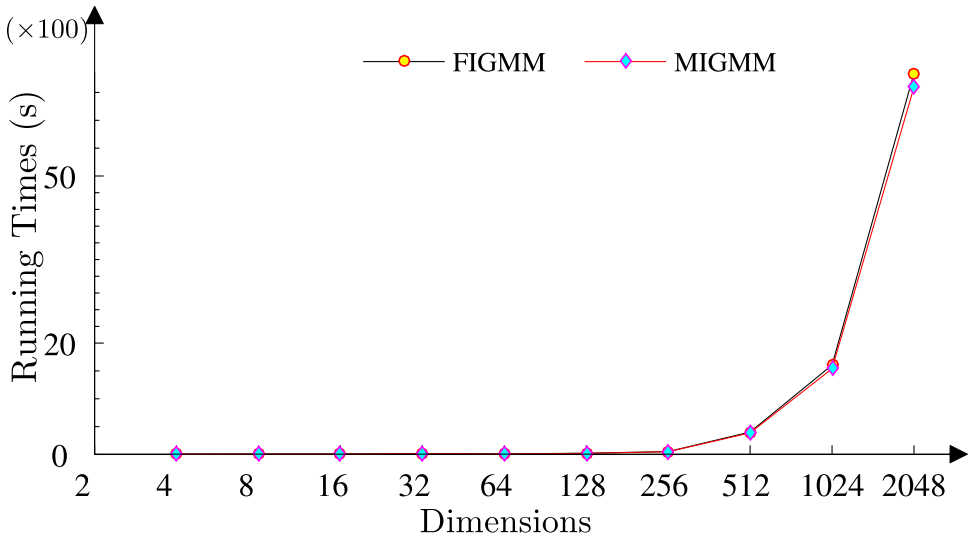


Fig. 7 Running time for the MIGMM and FIGMN algorithms with growing number of dimensions

Acknowledgements This work was supported by the Key Projects of Hunan Provincial Department of Education (Nos. 21A0403 and 21A0405), the Hunan Provincial Natural Science Foundation of China (No. 2022JJ30282), the Key Laboratory of Hunan Province (No. 2019TP1014), and the university-industry collaborative project (No. 202102211006).

Author contributions SS: Conceptualization, Methodology, Data curation, Algorithm design, Software, Writing original draft, Writing review and editing. YT and BZ: Experimental analysis, Investigation, Algorithm validation, Writing review and editing. BY, LY, PH and HX: Data collection, Experimental analysis and editing.

Declarations

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

References

1. Zhang J, Yin Z, Wang R (2017) Pattern classification of instantaneous cognitive task-load through GMM clustering, Laplacian Eigenmap, and ensemble SVMs. IEEE/ACM Trans Comput Biol Bioinf 14:947–965
2. Li Z, Xia Y, Ji Z, Zhang Y (2017) Brain voxel classification in magnetic resonance images using niche differential evolution based Bayesian inference of variational mixture of Gaussians. Neurocomputing 269:47–57
3. Ortiz-Rosario A, Adeli H, Buford JA (2017) MUSIC-expected maximization gaussian mixture methodology for clustering and detection of task-related neuronal firing rates. Behav Brain Res 317:226–236
4. Davari A, Aptoula E, Yanikoglu B, Maier A, Riess C (2018) GMM-based synthetic samples for classification of hyperspectral images with limited training data. IEEE Geosci Remote Sens Lett 15:942–946
5. Simms LM, Blair B, Ruz J, Wurtz R, Kaplan AD, Glenn A (2018) A pulse discrimination with a Gaussian mixture model on an FPGA. Nucl Inst Methods Phys Res A 900:1–7
6. Xue W, Jiang T (2018) An adaptive algorithm for target recognition using Gaussian mixture models. Meas J Int Meas Confed 124:233–240
7. Heinen MR, Engel PM, Pinto RC (2012) Using a gaussian mixture neural network for incremental learning and robotics. In: The 2012 international joint conference on neural networks (IJCNN), pp 1–8
8. Heinen MR, Engel PM, Pinto RC (2011) IGMN: an incremental gaussian mixture network that learns instantaneously from data flows. In: Proc VIII Encontro Nacional de Inteligência Artificial (ENIA2011)
9. Dempster AP (1977) Maximum likelihood estimation from incomplete data via the EM algorithm. J R Stat Soc Ser B (Stat Methodol) 39:1–38
10. Engel PM, Heinen MR Incremental learning of multivariate gaussian mixture models. In: Brazilian symposium on artificial intelligence. Springer, pp 82–91

11. Grossberg S (1987) Competitive learning: from interactive activation to adaptive resonance. Cogn Sci 11:23–63
12. Pinto RC, Engel PM (2015) A fast incremental gaussian mixture model. PLOS ONE 10(10):e0141942
13. Pragr M, Cizek P (2018) Cost of transport estimation for legged robot based on terrain features inference from aerial scan. In: IEEE international conference on intelligent robots and systems, pp 1745–1750
14. Prágr M, Čížek P (2019) Incremental learning of traversability cost for aerial reconnaissance support to ground units, lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) 11472 LNCS, pp 412–421
15. Zhao R, Li Y, Sun Y (2018) Statistical convergence of the EM algorithm on Gaussian mixture models. http://arxiv.org/abs/1810.04090
16. Robbins H, Monro S (1951) A stochastic approximation method. Ann Math Stat 22:400–407
17. Pinto RC, Engel PM (2015) A fast incremental gaussian mixture model. PLoS ONE 10:1–12
18. Chamby-Diaz CJ, Recamonde-Mendoza M, Bazzan LCA, Grunitzki R (2018) Adaptive incremental gaussian mixture network for non-stationary data stream classification. In: Proceedings of the international joint conference on neural networks 2018-July, pp 1–8
19. Koert D, Trick S, Ewerton M, Lutter (2019) Online learning of an open-ended skill library for collaborative tasks. In: IEEE-RAS international conference on humanoid robots 2018-November, pp 599–606
20. Drumond DA, Rolo RM, Costa JFCL (2019) Using Mahalanobis distance to detect and remove outliers in experimental covariograms. Nat Resour Res 28:145–152
21. Singh R, Pal BC, Jabr RA (2009) Statistical representation of distribution system loads using gaussian mixture model. IEEE Transa Power Syst 25:29–37
22. Salmond DJ, Atherton DP, Bather JA (1989) Mixture reduction algorithms for uncertain tracking. IFAC Proc Ser 2:775–780
23. Proïa F, Pernet A, Thouroude T, Michel G, Clotault J (2016) On the characterization of flowering curves using Gaussian mixture models. J Theor Biol 402:75–88
24. Mungai PK (2017) Using keystroke dynamics in a multi-level architecture to protect online examinations from impersonation. In: 2017 IEEE 2nd international conference on big data analysis, pp 622–627
25. Aryafar A, Mikaeil R, Ardejani FD, Haghshenas SS, Jafarpour A (2019) Application of non-linear regression and soft computing techniques for modeling process of pollutant adsorption from industrial wastewaters. J Min Environ 10(2):327–337
26. Sun S, Wang H, Chang Z, Mao B, Liu Y (2019) On the Mahalanobis distance classification criterion for a ventricular septal defect diagnosis system. IEEE Sens J 19:2665–2674
27. Xie C, Chang J, Liu Y (2013) Estimating the number of components in Gaussian mixture models adaptively. J Inf Comput Sci 10:4453–4460
28. Keribin C (2000) Consistent estimate of the order of mixture models. Sankhy A Ser A 62:49–66
29. Akaike H (1974) A new look at the statistical model identification. IEEE Trans Autom Control 19:716–723
30. Wang H, Luo B, Zhang Q, Wei S (2004) Estimation for the number of components in a mixture model using stepwise split-and-merge EM algorithm. Pattern Recognit Lett 25:1799–1809
31. Shi L, Xu L (2006) Local factor analysis with automatic model selection: a comparative study and digits recognition application. In: Kollias S, Stafylopatis A, Duch W, Oja E (eds) Artificial neural networks—ICANN 2006. Springer, Berlin, pp 260–269
32. Pinto R (2015) Experiment data for "A Fast Incremental Gaussian Mixture Model". https://doi.org/10.6084/M9.FIGSHARE.1552030.V2
33. Deb S, Tian Z, Fong S, Wong R, Millham R, Wong KK (2018) Elephant search algorithm applied to data clustering. Soft Comput 22:6035–6046
34. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830
35. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86:2278–2324
36. Krizhevsky HGLA (2009) Learning multiple layers of features from tiny images. Technical Report, Computer Science Department, University of Toronto
37. Kusetogullari H, Yavariabdi A, Cheddad A, Grahn H, Hall J (2020) ARDIS: a Swedish historical handwritten digit dataset. Neural Comput Appl 32:16505–16518
38. Wang Y, Chaib-draa B (2016) KNN-based Kalman filter: an efficient and non-stationary method for Gaussian process regression. Knowl-Based Syst 114:148–155