SPLITTING-WHILE-MERGING FRAMEWORK FOR CLUSTERING HIGH-DIMENSION DATA WITH COMPONENT-WISE EXPECTATION CONDITIONAL MAXIMISATION

Rui Fa[1], *Basel Abu-Jamous*[1], *David J. Roberts*[2,3] *and Asoke K. Nandi*[1,4]

[1] Department of Electronic and Computer Engineering, Brunel University, Uxbridge, UB8 3PH, United Kingdom.

[2] National Health Service Blood and Transplant, Oxford, United Kingdom

[3] The University of Oxford, John Radcliffe Hospital, Oxford, United Kingdom

[4] Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland.

Email: {Rui.Fa, Basel.AbuJamous, Asoke.Nandi}@brunel.ac.uk, david.roberts@ndcls.ox.ac.uk

ABSTRACT

To meet the demand of clustering high dimensional data efficiently, in this paper, we propose a component-wise expectation conditional maximisation (CW-ECM) algorithm and integrate it within the recent proposed splitting-while-merging framework, which is called splitting-merging awareness tactics (SMART), for the mixture of factor analysers (MFA) model. The new algorithm has two advantages: it has ability to converge to actual or close actual number of clusters by a splitting-while-merging strategy, and it avoids the local maxima effectively and efficiently. Furthermore, we improve the splitting strategy in the original SMART framework and save more computational effort. We test out algorithm in two benchmark datasets and compare it with the state-of-the-art algorithms using many validation metrics. The results show that the proposed algorithm outperforms the compared algorithms in clustering performance with significantly less computational complexity.

Index Terms— mixture of factor analysers (MFA), SMART, expectation maximisation (EM), expectation conditional maximisation (ECM)

1. INTRODUCTION

Clustering data based on a measure of dissimilarity or similarity between data objects has been one of critical parts in scientific data analysis and engineering applications [1–5]. In the rich literature of clustering algorithms, model-based clustering is one of the most popular clustering families [1, 6–8]. Finite mixture models (FMM) have provided a statistical base for modelling multivariate data in a wide variety of random phenomena [6]. However, the vital problem for the widely used Gaussian mixture model is that it is a highly parametrized model with $[\frac{1}{2}p(p-1)]$ parameters for each component covariance matrix, where p is the dimensionality of the data. This fact limits the practical use of Gaussian mixture model in many applications, where people have to deal with high-dimension data, for example, gene expression data analysis or image processing.

Mixture of factor analysers (MFA) has been used to fit a mixture of Gaussians to correlate high dimensional data without requiring $\mathcal{O}(p^2)$ parameters [9–16]. Expectation maximisation (EM), which maximises the likelihood by iterating E(xpectation) and M(aximisation) steps, has been employed to fit MFA by Ghahramani and Hinton [9]. It converges stably since its M step takes the closed form of the likelihood function. However, the cost of such stability is its slow convergence speed [16]. Meng and van Dyk developed an alternative expectation conditional maximisation (AECM) to attain a trade-off between stability and convergence [10]. McLachlan and colleagues developed a software called EMMIX-GENE employing the MFA model and AECM algorithm for clustering large dimensional microarray gene expression data [12, 14]. McNicholas and Murphy devised a class of eight parsimonious Gaussian mixture models based on the MFA model and AECM algorithm, by imposing different constraints on the loading matrices and covariance structure [15]. Zhao and Yu proposed a fast ECM algorithm for the MFA model, by only treating the component-indicators as missing data [13, 16]. These algorithms have been demonstrated to provide notably good performance in dealing with large dimensional data, however, there are two critical issues remaining unsolved: (1) these algorithms highly depend on the setting of the a prior knowledge of the number of models, which is unrealistic; and (2) it is well known that EM and its derivatives may get stuck in local maxima easily.

To this end, a splitting-while-merging (SWM) clustering framework, named splitting-merging awareness tactics (SMART), was proposed in [17, 18]. Especially in [18], SMART-FMM was proposed to integrate the component-wise EM (CW-EM) algorithm, which was originally proposed by Celeux [19] and modified by Figueiredo and Jain [20], into the SWM framework. The benefits of using CW-EM come, firstly, from its feature of avoiding the local minima [20] and secondly from its ability to eliminate those weak models automatically. The SWM framework performs a top-down process without the requirement of pre-defined number of clusters (components) or an upper bound of this number. Integrating CWEM into the SWM framework meets the requirement of the clustering analysis and outperforms the counterpart algorithms [18, 21].

In this paper, we develop the component-wise expectation conditional maximisation (CW-ECM) algorithm for the SMART framework with the MFA model. The CW-ECM algorithm updates each component with two-stage process – E step and CM steps – in parallel. We design the CW-ECM algorithm following the suggestion by Zhao and Yu that only treating component indicators as missing data may accelerate the convergence speed. [13, 16]. Moreover, we improve the splitting strategy in the original SMART framework so that the algorithm selects candidates for splitting more efficiently. We test our algorithm in two benchmark datasets, one of which is an artificial dataset and another one is a real gene expression dataset. We compare it with the state-of-the art algorithms, namely EM, AECM and ECM algorithms for the MFA models. The numerical results

This article summarises independent research funded by the National Institute for Health Research (NIHR) under its Programme Grants for Applied Research Programme (Grant Reference Number RP-PG-0310-1004). The views expressed are those of the authors and not necessarily those of the NHS, the NIHR or the Department of Health. A. K. Nandi would like to thank TEKES for their award of the Finland Distinguished Professorship.

show that the proposed algorithm outperforms the compared algorithms in clustering performance with significantly less computational complexity.

The rest of the paper is organized as follows: Sec. 2 reviews the MFA model and the SMART framework respectively. In Sec. 3, we detail the principle of the proposed SMART-MFA and the CW-ECM algorithm. In Sec. 4, we conduct experiments and present the result comparison between the proposed algorithm and the state-of-the-art algorithms. Finally, conclusions and discussion are given in Sec. 5.

2. MFA MODEL AND SMART FRAMEWORK

In this section, we briefly review previous works in the MFA model and the SMART framework in separate subsections, respectively.

2.1 MFA Model

2.1.1 MFA Model

The MFA is a mixture of G FA submodels with mixture proportions $\{\pi_g\}_{g=1}^G$ and the constraint $\sum_{g=1}^G \pi_g = 1$. Thus, given label g for g-th submodel, the observed data vector $\boldsymbol{x}_n$ is modelled as

$$\boldsymbol{x}_n = \boldsymbol{B}_g \boldsymbol{u}_{gn} + \boldsymbol{\mu}_g + \boldsymbol{\epsilon}_{gn}, \tag{1}$$

where $\boldsymbol{B}_g$ is a $p \times q$ factor loading matrix, $\boldsymbol{u}_{gn}$ is a q-dimensional latent factor vector following $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_q)$, where $\boldsymbol{I}_q$ is an identity matrix with q dimensions. $\boldsymbol{\mu}_g$ is a p-dimensional mean vector, and $\boldsymbol{\epsilon}_{gn} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Psi}_g)$, where $\boldsymbol{\Psi}_g$ is a positive diagonal matrix diag$\{\psi_{g1}, ..., \psi_{gp}\}$. Thus, $\boldsymbol{x}_n$ is distributed as $\mathcal{N}(\boldsymbol{B}_g\boldsymbol{u}_{gn} + \boldsymbol{\mu}_g, \boldsymbol{\Psi}_g)$ conditioned on $\boldsymbol{u}_{gn}$, or is distributed as $\mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ unconditionally, where the covariance matrix $\boldsymbol{\Sigma}_g = \boldsymbol{B}_g\boldsymbol{B}_g^T + \boldsymbol{\Psi}_g$. Comparing with the general Gaussian model with $[p+p(p-1)/2]$ free parameters for each submodel, the MFA model has $[pq+p-q(q-1)/2]$ free parameters. If q is chosen sufficiently smaller than p, some constraints are imposed on the covariance matrix $\boldsymbol{\Sigma}_g$ and the number of free parameters to be estimated are significantly reduced.

The probability density function (pdf) of the observed data by G mixtures of FAs is given by

$$p(\boldsymbol{x}_n|\boldsymbol{\Theta}) = \sum_{g=1}^{G} \pi_g \phi(\boldsymbol{x}_n|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g). \tag{2}$$

where $\phi(\cdot)$ is the pdf of multivariate normal distribution, $\boldsymbol{\Theta}$ denotes the parameter set $\{\{\boldsymbol{\mu}_g\}_{g=1}^G, \{\boldsymbol{B}_g\}_{g=1}^G, \{\boldsymbol{\Psi}_g\}_{g=1}^G, \{\pi_g\}_{g=1}^G\}$.

2.1.2 Maximum Likelihood Estimation

Let us denote $\boldsymbol{Z} = \{\boldsymbol{z}_n\}_{n=1}^N$ as latent label indicators, where $\boldsymbol{z}_n = \{z_{n1}, z_{n2}, ..., z_{nG}\}$, where $z_{ng} = 1$ if $\boldsymbol{x}_n$ belongs to group g and $z_{ng} = 0$ otherwise. $\boldsymbol{U} = \{\boldsymbol{u}_n\}_{n=1}^N$ is a set of latent factor vectors. They are missing data to be estimated in the clustering problem. The complete log-likelihood is given by

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{Z}|\boldsymbol{\Theta}) = \sum_{n=1}^{N}\sum_{g=1}^{G} z_{ng} \ln\left[\pi_g \phi(\boldsymbol{x}_n|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)\right]. \tag{3}$$

The EM algorithm and its derivatives, namely ECM and AECM, has been widely used to find maximum likelihood (ML) estimates of $\boldsymbol{Z}$, $\boldsymbol{U}$ and $\boldsymbol{\Theta}$ for the MFA model. The general EM algorithm performs an E step and an M step iteratively. In E step, it calculates the expected $\mathcal{L}$ with given $\boldsymbol{\Theta}$ and in M step it maximises $\mathcal{L}$ with respect to (w.r.t.) $\boldsymbol{\Theta}$. The ECM and AECM algorithms share the similar idea that the M step of the general EM algorithm is replaced by a number of computationally simpler conditional maximization (CM) steps.

2.2 SMART Framework

In this part, we emphasize two critical points of the SMART framework, which are the splitting-while-merging (SWM) strategy and the component-wise EM (CW-EM) algorithm [17, 18]. Due to the page limit, we cannot fully describe the whole framework. The interested readers are referred to [17, 18] for the details.

The SMART framework employed an SWM strategy. While splitting, a merging process is also taking place to merge the clusters which meet the merging criterion. In such a process, SMART has self-awareness to split and merge clusters automatically in iterations. To do so, many clustering tasks have to be performed. In the splitting task of each iteration (Task 2 in [17, 18]), SMART splits one cluster into two. Then, the new clusters are tested by a merging criterion which is called merging task (Task 3 in [17, 18]). If any pair of clusters meet the merging criterion, we merge the two clusters, otherwise skip the merging step. Then SMART goes through a termination-check, where a stopping criterion is applied. If the condition for termination is not satisfied, SMART goes to the next iteration and continues to split, otherwise, SMART finishes SWM process. The last step is clustering selection task (Task 4 in [17, 18]). Minimum message length (MML) [25] was employed in clustering selection task. In this paper, we also employ MML for clustering selection task.

Unlike conventional EM algorithm, CW-EM updates the model parameters $\{\boldsymbol{\theta}_g|1 \le g \le G\}$ and the probabilities of components $\{\pi_g|1 \le g \le G\}$ sequentially, rather than simultaneously. In CW-EM, the estimation is also two-step process, but in each iteration, only one component has the opportunity to update its parameters. Thus, strong clusters can thrive and weak clusters lose their members and vanish eventually. This mechanism does the merging task by eliminating empty clusters implicitly. By this mechanism, CW-EM can prevent the algorithm getting stuck into local maxima effectively.

3. SMART-MFA

In this section, we develop the SMART framework with the MFA model. There are two additional principal contributions over the original SMART framework [18]. The first, obviously, is the component-wise expectation conditional maximisation (CW-ECM) which is designed for the MFA model, and the second is the splitting strategy which avoids calculating the whole pairwise distances and makes the splitting more efficient.

3.1 CW-ECM

To estimate the latent label indicators $\boldsymbol{Z}$ conditional on the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_g\}_{g=1}^G = \{\{\boldsymbol{\mu}_g\}_{g=1}^G, \{\boldsymbol{B}_g\}_{g=1}^G, \{\boldsymbol{\Psi}_g\}_{g=1}^G, \{\pi\}_{g=1}^G\}$, the CW-ECM updates each component in parallel with a two-stage process, that is, E step and CM steps. Suppose that in t-th iteration, for the g-th component, it has parameters $\tilde{\boldsymbol{\Theta}}^{(t,g)} = \{\boldsymbol{\Theta}_1^{(t,g)}, ..., \boldsymbol{\Theta}_{g-1}^{(t,g)}, \boldsymbol{\Theta}_g^{(t-1,g)}, ..., \boldsymbol{\Theta}_G^{(t-1,g)}\}$ and alternates the steps as

- **CW-ECM *E-step***: Compute for $n = 1, ..., N$

$$\gamma_{gn} \equiv E[\hat{z}_{gn}|\boldsymbol{X}, \tilde{\boldsymbol{\Theta}}^{(t,g)}] = \frac{f(\boldsymbol{x}_n|\tilde{\boldsymbol{\Theta}}_g^{(t,g)})}{\sum_{l=1}^G f(\boldsymbol{x}_n|\tilde{\boldsymbol{\Theta}}_l^{(t,g)})}, \tag{4}$$

where

$$\tilde{\boldsymbol{\Theta}}_l^{(t,g)} = \begin{cases} \boldsymbol{\Theta}_l^{(t,g)} & \text{If } l < g \\ \boldsymbol{\Theta}_l^{(t-1,g)} & \text{If } l \ge g \end{cases} \tag{5}$$

Thus, we may obtain

$$f(\boldsymbol{x}_n|\tilde{\boldsymbol{\Theta}}_l^{(t,g)}) = \begin{cases} \hat{\pi}_l^t p(\boldsymbol{x}_n|\boldsymbol{\mu}_l^{(t,g)}, \boldsymbol{B}_l^{(t,g)}, \boldsymbol{\Psi}_l^{(t,g)}) & \text{If } l < g \\ \hat{\pi}_l^{(t-1,g)} p(\boldsymbol{x}_n|\boldsymbol{\mu}_l^{(t-1,g)}, \boldsymbol{B}_l^{(t-1,g)}, \boldsymbol{\Psi}_l^{(t-1,g)}) & \text{If } l \ge g \end{cases} \tag{6}$$

- **CW-ECM *CM-step***: Set

$$\boldsymbol{\Theta}_g^{(t,g)} = \arg\max_{\boldsymbol{\Theta}_g^{(t-1,g)}} \{\log p(\boldsymbol{X}|\tilde{\boldsymbol{\Theta}}^{(t,g)})\}. \quad (7)$$

More precisely, (7) can be broken down into many individual updates as following:

$$\hat{\pi}_g^{(t,g)} = \frac{\sum_{n=1}^N \gamma_{gn}}{\sum_{l=1}^G \sum_{n=1}^N \gamma_{ln}}, \quad (8)$$

$$\boldsymbol{\mu}_g^{(t,g)} = \frac{\sum_{n=1}^N \gamma_{gn} \boldsymbol{x}_n}{\sum_{n=1}^N \gamma_{gn}}, \quad (9)$$

and

$$\boldsymbol{B}_g^{(t,g)} = \boldsymbol{S}_g^{(t,g)} \boldsymbol{\beta}_g^{(t,g)} (\boldsymbol{\beta}_g^{(t,g)^T} \boldsymbol{S}_g^{(t,g)} \boldsymbol{\beta}_g^{(t,g)} + \omega_g^{(t,g)})^{-1} \quad (10)$$

where

$$\begin{aligned} \boldsymbol{S}_g^{(t,g)} &= \frac{\sum_{n=1}^N \gamma_{gn} (\boldsymbol{x}_n - \boldsymbol{\mu}_g^{(t,g)})(\boldsymbol{x}_n - \boldsymbol{\mu}_g^{(t,g)})^T}{\sum_{n=1}^N \gamma_{gn}}, \\ \boldsymbol{\beta}_g^{(t,g)} &= (\boldsymbol{B}_g^{(t-1,g)} \boldsymbol{B}_g^{(t-1,g)^T} + \boldsymbol{\Psi}_g^{(t-1,g)})^{-1} \boldsymbol{B}_g^{(t-1,g)}, \\ \omega_g^{(t,g)} &= I_q - \boldsymbol{\beta}_g^{(t,g)^T} \boldsymbol{B}_g^{(t-1,g)}. \end{aligned}$$

Then the updated estimate $\boldsymbol{\Psi}_g^{(t,g)}$ is given by

$$\boldsymbol{\Psi}_g^{(t,g)} = \text{diag}\{\boldsymbol{S}_g^{(t,g)} - \boldsymbol{S}_g^{(t,g)} \boldsymbol{\beta}_g^{(t,g)} \boldsymbol{B}_g^{(t,g)^T}\}. \quad (11)$$

Note that in all EM algorithms for the MFA models, selecting the number of loading factors q is arbitrary. In our algorithm, we search a range of q in $[1, q_{max}]$.

Table 1. The pseudo-code for SMART-MFA.

Task 1: Initializing SMART with $G = 2$
Randomly initialize $\hat{\boldsymbol{\mu}}_g$, $\hat{\boldsymbol{B}}_g$, $\hat{\boldsymbol{\Psi}}_g$ and $\hat{\pi}_k$ for $g = 1, 2$;
terminate = 0;
while !terminate **do**
  Learning and Merging:
  while !converge **do**
    for $g = 1 : G$ **do**
      Use CW-ECM for the learning and merging based on (4) and (7).
      if $\hat{\pi}_g \to 0$ **then**
        Get rid of g-th cluster; $G = G - 1$;
      end if
    end for
  end while(!converge)
  Splitting:
  Employ the splitting strategy in Sec. 3.2; $G = G + 1$;
  if The number of merges is greater than or equal to N_m **then**
    terminate = 1;
  end if
end while(!terminate)
Selecting:
Select the best clustering based on MML criterion.

3.2 Splitting Strategy

We improve the splitting strategy in the original SMART framework, which calculates all pairwise distances in the dataset and performs a Kauffman approach (KA) style [22] to search for next candidate to split. Such calculation is huge when the size of the dataset is merely moderate, say around thousand data objects. We notice that the likelihood of the given object allocated to the g-th cluster, which is $f(\boldsymbol{x}_n|\boldsymbol{\Theta}_g)$, has been calculated during the EM algorithm, and it can be used to judge how likely the given object should be allocated in the cluster. If an object has very small likelihood in every cluster, there may be two possibilities: one is that the object is an outlier, and another is that the object belongs to a cluster, which has not been discovered. Even for an outlier, it may be an outlier near the existing clusters or near the cluster not been discovered. In the CW-type EM (or ECM) algorithm, an actual cluster may survive from iteration to iteration even the initial point is relatively far from its center, on the other hand, the cluster may vanish if it is not an actual cluster.

Thus, we design our splitting strategy as following few steps:

1. To create a pool to record the splitting candidates has been selected;
2. To find a data object, which has minimum value of $\sum_{g=1}^G f(\boldsymbol{x}_n|\boldsymbol{\Theta}_g)$ among all objects not in the pool;
3. To assign the data object as the μ_{G+1} of the new $(G+1)$-th cluster, and generate $\boldsymbol{B}_{G+1}$, $\boldsymbol{\Psi}_{G+1}$, π_{G+1} randomly, record the data object in the pool, and $G = G + 1$.

To summarise, the pseudo-code of the proposed algorithm is shown in Table 1.

4. NUMERICAL RESULTS

Here, we describe two benchmark datasets employed to test our algorithm, and present the experiment setups and numerical results. Our proposed SMART-MFA algorithm is compared with the state-of-the-art algorithms for the MFA models, namely EM [9], AECM [14], and ECM [16] using many validation metrics.

4.1 Datasets

We employ a model similar to the one used in [16] to generate the artificial dataset. It has $N = 600$ data objects in $\mathbb{R}^d (d = 50)$. The parameters for generating the datasets are given as $G = 3, q = 8, \boldsymbol{\mu}_g = 5g\mathbf{1}_{d\times 1}, \boldsymbol{B}_g = \text{Unif}(d, q), \boldsymbol{\Psi}_g = \text{diag}\{\text{Unif}(d, 1))\}, \pi_g = \frac{1}{3}, g = 1, ..., G$. Here, $\text{Unif}(a, b)$ is an $a \times b$ uniformly distributed random number matrix on the unit interval.

The real microarray gene expression dataset consists of 38 bone marrow samples obtained from acute leukaemia patients at time of diagnosis. There are 999 genes in the dataset [24]. The biological truth is that the samples include 3 groups: 11 acute myeloid leukaemia (AML) samples, 8 T-lineage acute lymphoblastic leukaemia (ALL) samples, and 19 B-lineage ALL samples [23, 24].

4.2 Numerical results

It is worth noting that all state-of-the-art algorithms require the number of models (clusters) as one of the inputs and cannot automatically converge to the clustering with the actual or nearly actual number of clusters. For the sake of a fair comparison, for each dataset, we design two experimental settings for the compared algorithms: setting one (S1) is an exhaustive search over a range of numbers of clusters $[2, g_{max}]$; setting two (S2) is that algorithms are given the actual number of clusters. Both experimental settings search a range of numbers of factors $[1, q_{max}]$ and MML is employed as clustering selection criterion. In this paper, q_{max} is always set to 10. Considering that EM-type algorithms may get stuck in the local maxima, we set N_{init} random starting points, half of which are purely random and the rest are generated by kmeans with random initialisation. All experiments are run 100 times. We employ many validation metric to validate the clustering results, namely adjusted Rand index (ARI) [28], Jaccard index (JI) [29] (ARI and JI only for artificial dataset since the ground truth of membership is know), mean square error (MSE), MML, Calinski-Harabasz (CH) index [27], Silhouette index (SI) [26]. The mean and standard deviation of above metrics are presented. We also measure correct selection rate (CSR) of number of clusters, and the statistics of estimated number of clusters $\hat{G}$. For comparing the complexities, we measure both total iterations used in each algorithm and total processing time in seconds. For

Table 2. Performance comparison of all algorithms in the artificial dataset.

Metric	EM		AECM		ECM		Proposed
	S1(10)	S2(50)	S1(10)	S2(50)	S1(10)	S2(50)	
JI	0.8±0.2	1±1.5E-3	0.81±0.2	1±1.5E-3	0.96±0.1	1±1.5E-3	**1±2.5E-3**
ARI	0.79±0.2	1±3E-3	0.82±0.2	1±3E-3	0.96±0.1	1±3E-3	**1±3.3E-3**
MSE	9.97E5±8.1E3	1.93E5±1.8E-10	2.85E5±9.8E4	1.93E5±1.8E-10	3.32E5±7.5E4	1.93E5±1.8E-10	**1.93E5±7.3E-3**
MML	4.84E4±2.3E2	4.34E4±5.8E2	3.72E4±3.1E4	4.34E4±82	4.49E4±3.2E2	4.34E4±5.4E2	**4.32E4±1.7E2**
Sil	0.89±7.9E-2	0.97±2.1E-15	0.33±0.4	0.97±2.1E-15	0.90±0.2	0.97±2.1E-15	**0.97±2.4E-2**
CH	169.8±1.3E2	294.2±3.8E-13	91.03±88	294.2±3.6E-13	5.62±1.3	294.2±4.1E-13	**291.4±1.1E-4**
$\hat{G}$	2.5±0.5	/	8±2.4	/	3.31±1.3	/	**3±0**
CSR	50%	/	14%	/	86%	/	**100%**
Inters	6.2E6	4.6E6	6.3E6	4.8E6	7.8E6	5.1E6	**2.4E6**
Time (s)	2.8E4	1.5E4	2.8E4	1.7E4	2.7E4	1.4E4	**1.1E4**

Table 3. Performance comparison of all algorithms in Leukemia dataset.

Metric	EM		AECM		ECM		Proposed
	S1(10)	S2(50)	S1(10)	S2(50)	S1(10)	S2(50)	
MSE	2.72E4±2.2E2	2.28E4±15	2.08E4±6.2E2	2.53E4±2.3E2	2.73E4±1.8E3	2.33E4±68	**2.28E4±17**
MML	4.62E4±1.4E2	4.60E4±3.0E2	3.87E4±1.5E2	4.41E4±1.8E2	4.4E4±1.6E2	4.46E4±4.2E2	**4.36E4±45**
Sil	0.33±1.2E-2	0.36±7.1E-5	0.26±1.9E-2	0.32±1.3E-2	0.29±5.4E-2	0.33±3.9E-2	**0.36±9E-4**
CH	6.28±0.3	6.47±1.3E-3	2.91±0.1	6.21±0.1	5.62±1.3	6.43±4.7E-2	**6.48±1.4E-3**
$\hat{G}$	2±0	/	9.75±0.5	/	2.22±0.5	/	**3±0**
CSR	0	/	0	/	10%	/	**100%**
Inters	6.1E6	3.6E6	5.9E6	2.9E6	4.4E6	2.1E6	**1.8E6**
Time (s)	5.6E4	1.6E4	5.5E4	1.6E4	4.5E4	1.3E4	**5.5E3**

both datasets and all compared algorithms, we set $g_{max} = 10$ and $N_{init} = 10$ for S1 and we set $N_{init} = 50$ for S2. For our proposed algorithm, we set the maximum times for merging N_m to 20.

The performance comparisons between our proposed algorithm and all state-of-the-art algorithms in the artificial dataset are shown in Table 2. We find that the performance of the proposed algorithm is very similar to those of the state-of-the-art algorithms with S2 which is given the number of clusters $G = 3$ and relatively large number of starting points ($N_{init} = 50$). The proposed algorithm has slightly higher standard deviations in JI, ARI, MSE, and SI, and has a bit lower value in CH. Generally speaking, the differences are not significant. Nevertheless, two facts are worth noting : firstly the processing time of the proposed algorithm is much shorter than that of the compared algorithm, and nearly 30%-40% computational efforts are saved; secondly, in S2, the actual number of clusters is given to the compared algorithms while the proposed algorithm does not need such information, which is not a fair comparison. Let us consider S1, which is a more realistic scenario, all compared algorithms cannot provide satisfactory performance with comparable complexity. Among the compared algorithms, ECM performs the best and uses the least processing time. Its performance, however, is much worse than the proposed algorithm and it takes twice as much processing time as the proposed algorithm takes.

Table 3 shows the performance comparisons using real leukaemia dataset. In this case, our proposed algorithm performs slightly better than the state-of-the-art algorithms with S2, but only takes less than half of their processing time. All compared algorithms with S1 perform poorly. All these experimental results indicate that the proposed algorithm outperforms the state-of-the-art algorithms in a more realistic scenario and provides better results with less computational complexity.

5. CONCLUSIONS AND DISCUSSIONS

Mixture of factor analysers (MFA) model has been widely used for modelling large dimensional data and there are many state-of-the-art algorithms in the literature, namely EM, AECM and ECM. However, all of them have two vital limitations: 1) need the knowledge of number of cluster; 2) get stuck in local maxima easily.

In this paper, we proposed a component-wise ECM (CW-ECM) algorithm and integrated it within the recent reported SMART framework to fit the MFA model. We also modified the splitting strategy in the original framework so that the proposed algorithm resolves the aforementioned two issues in an efficient way.

We tested the proposed algorithm in two benchmark datasets and compared its performance in many different validation metrics with those state-of-the-art algorithms. The results showed that our proposed algorithm provided as good performance as the compared algorithms when they were given the *a prior* information of number of clusters and a relatively large number of starting points, but had significantly less computational complexity. If in S1, which is a more realistic scenario, all state-of-the-art performed poorly in both datasets. EM suffered more with local maxima problem and chose two clusters as the best clustering more often. AECM suffered with the overfitting problem and always chose clustering with large number of clusters. Fairly speaking, this problem actually originated from the MML criterion. The reason why the proposed algorithm does not have the same problem is that the SWM strategy prevents the overfitting by merging those unnecessary clusters so that the proposed algorithm seldom reaches that point which makes MML fail. ECM performed the best among the state-of-the-art algorithms, but was much worse and more complex than the proposed algorithm.

In the future, we will investigate the proposed algorithm with more real datasets, especially the gene expression datasets. Our long term target is to discover biological knowledge buried under massive collected data by using our proposed algorithm.

6. REFERENCES

[1] Xu R., and Wunsch D., "Survey of clustering algorithms," IEEE Transactions on Neural Networks 16: 645–78, 2005.

[2] Bishop C. M., Nasrabadi N. M. *Pattern recognition and machine learning,* New York: springer, 2006.

[3] Fa R., Nandi A. K., and Gong L. Y., "Clustering analysis for gene expression data: A methodological review," 5th International IEEE Symposium on Communications Control and Signal Processing (ISCCSP), 2012.

[4] Abu-Jamous B., Fa R., Roberts D. J., and Nandi, A. K., "Paradigm of Tunable Clustering Using Binarization of Consensus Partition Matrices (Bi-CoPaM) for Gene Discovery," PloS ONE 8:2, e56432. doi:10.1371/journal.pone.0056432, 2013.

[5] Abu-Jamous B., Fa R., Roberts D. J., and Nandi, A. K., "Yeast gene CMR1/YDL156W is consistently co-expressed with genes participating in DNA-metabolic processes in a variety of stringent clustering experiments," J R Soc Interface 2013 10: 20120990, 2013.

[6] McLachlan G., and Peel D., *Finite mixture models*, Wiley, 2004.

[7] Fraley C., and Raftery A. E., "How many clusters? Which clustering method? Answers via model-based cluster analysis," *The computer journal*, vol. 41, no. 8, 1998.

[8] Yeung K. Y., Fraley C., Murua A., Raftery A. E., and Ruzzo W. L., "Model-based clustering and data transformations for gene expression data," Bioinformatics 17: 977–987, 2001.

[9] Ghahramani Z., and Hinton G. E., "The EM algorithm for mixtures of factor analyzers," Tech. Rep., Technical Report CRG-TR-96-1, University of Toronto, 1996.

[10] Meng X. L., and van Dyk D., "The EM Algorithm-an Old Folk Song Sung to a Fast New Tune," *Journal of the Royal Statistical Society: . . .*, vol. 59, no. 3, pp. 511–567, 1997.

[11] McLachlan G. J., Bean R. W., and Peel D., "A mixture model-based approach to the clustering of microarray expression data," *Bioinformatics*, vol. 18, no. 3, pp. 413–422, 2002.

[12] McLachlan G. J., Peel D., and Bean R. W., "Modelling high-dimensional data by mixtures of factor analyzers," *Computational Statistics & Data Analysis*, vol. 41, no. 3-4, pp. 379–388, Jan. 2003.

[13] Zhao J.-H., Yu P. L. H., and Jiang Q. B. , "ML estimation for factor analysis: EM or non-EM?," *Statistics and Computing*, vol. 18, no. 2, pp. 109–123, Nov. 2007.

[14] McLachlan G.J., Bean R.W., and Ben-Tovim Jones L. , "Extension of the mixture of factor analyzers model to incorporate the multivariate t-distribution," *Computational Statistics & Data Analysis*, vol. 51, no. 11, pp. 5327–5338, July 2007.

[15] McNicholas P. D., and Murphy T. B., "Parsimonious Gaussian mixture models," *Statistics and Computing*, vol. 18, no. 3, pp. 285–296, Apr. 2008.

[16] Zhao J. H., and Yu P. L. H., "Fast ML estimation for the mixture of factor analyzers via an ECM algorithm," *Neural Networks, IEEE Transactions on*, vol. 19, no. 11, pp. 1956–1961, 2008.

[17] Fa R., and Nandi A. K., "Smart: Novel self splitting-merging clustering algorithm," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012, pp. 2198–2202.

[18] Fa R., and Nandi A. K., "An enhanced splitting-while-merging algorithm with finite mixture models," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2013*. IEEE, 2013.

[19] Celeux G., "A Component-wise EM Algorithm for Mixtures," *Journal of Computational and Graphical Statistics*, vol. 10, no. 4, 2001.

[20] Figueiredo M. A. T., and Jain A. K., "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381 –396, march 2002.

[21] Fa R., Abu-Jamous B., Roberts D. J., and Nandi A. K., "Enhanced smart framework for gene clustering using successive processing," in *Proceedings of the 2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP-2013),*, 2013.

[22] Rousseeuw P. J., Kaufman L., *Finding groups in data: An introduction to cluster analysis,* John Wiley & Sons, Oxford, UK, 1990.

[23] Golub T. R., Slonim D. K., Tamayo P., Huard C., Gaasenbeek M., et al. "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," Science 286: 531–537, 1999.

[24] Monti S., Tamayo P., Mesirov J., and Golub T., "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data," Machine Learning 52: 91–118, 2003.

[25] Wallace C. S., and Dowe D. L., "Minimum message length and Kolmogorov complexity," *The Computer Journal*, 1999.

[26] Rousseeuw P. J., "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," Journal of Computational and Applied Mathematics, 20, 53–65, 1987.

[27] Calinski T., and Harabasz J., "A dendrite method for cluster analysis," Communications in Statistics - Theory and Methods, 3:1, 1–27, 1974.

[28] Hubert L., Arabie P., "Comparing partitions," Journal of Classification 2: 193–218, 1985.

[29] Tan P. N., *Introduction to data mining*, Pearson Education India, 2007.