AN ENHANCED SPLITTING-WHILE-MERGING ALGORITHM WITH FINITE MIXTURE MODELS

Rui Fa[1] *and Asoke K Nandi*[1,2]

[1] Department of Electronic and Computer Engineering, Brunel University,
Uxbridge, Middlesex, UB8 3PH, United Kingdom. {Rui.Fa, Asoke.Nandi}@brunel.ac.uk
[2] Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland.

ABSTRACT

In this paper, we propose a splitting-while-merging algorithm with finite mixture models (FMM) built on an improved splitting merging awareness tactics (SMART). The main property of SMART is that it does not require any dataset-dependent parameters or *a priori* knowledge about the datasets. The improved SMART framework integrates clustering selection criterion, which plays a vital role in the new algorithm. In the SMART-FMM implementation, the modified component-wise EM of mixtures is employed as a learning and merging technique and a model order selection algorithm is used as a clustering selection criterion. One demonstration example and one real microarray gene expression dataset are studied using our approach. The numerical results show that SMART-FMM is superior and more effective than others.

Index Terms— Splitting-merging clustering, Gene expression analysis, Microarray.

1. INTRODUCTION

As one of powerful exploratory tools, clustering has been widely used in not only the fields related to computational science but also other fields which have requirements of analysing a large amount of data, say biology and medical research [1–7]. However, most of successful clustering algorithms highly depend on parameter settings and initializations, e.g., the number of clusters and the centroids. If these parameters are initialized randomly, the clustering results would be unreliable and inconsistent. It urges people to produce a framework or a strategy, which may organically integrate many clustering techniques to fulfil some automated clustering.

A self-splitting competitive learning (SSCL) algorithm was proposed to achieve the automated clustering [8] where a new competitive learning paradigm, so called *one-prototype-take-one-cluster* (OPTOC), was developed for self-splitting. However, there are two vital issues to prevent its practical uses: 1) the prototypes are easily trapped into global centroid, especially the first few ones [8]; and 2) the parameters for stopping both OPTOC learning and splitting are crucial to the algorithm but they are difficult to estimate reliably [9]. In spite of above issues, the SSCL has an attractive advantage in that it does not require *a priori* knowledge about the number of clusters in the input dataset.

There was another strategy fulfilling the automated clustering proposed in [9, 10], which employed a similar framework where the input data was over-clustered to a large number of partitions, say k_{max}, then these partitions were merged to fewer clusters, which were closer to the natural clusters. In terms of clustering techniques, [10] was based on unsupervised learning of finite mixture models (ULFMM) while the self-splitting-merging competitive learning (SSMCL) in [9] was based on OPTOC competitive learning paradigm. In ULFMM, along with the merging process from k_{max} to k_{min}, minimum message length (MML), which is a model order selection criterion, was used; while in SSMCL, as a merging criterion was defined according to the measurement of distortion between two clusters, merging process would not stop until no cluster met the merging criterion. A vital issue in their framework is that the maximum number of clusters k_{max} is hard to determine *a priori*.

Recently, a splitting-merging clustering framework, named splitting-merging awareness tactics (SMART), was proposed in [11]. Different from aforementioned over-cluster-then-merge strategy, the SMART framework employed a novel splitting-while-merging (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 mimic human perception doing the sorting and grouping. To implement the SMART framework, OPTOC competitive learning was employed as the splitting algorithm and the calculation of cohesion between two clusters [12] was used as the merging criterion. We call the algorithm SMART-CL. The main advantage of the SMART framework is that it is not necessary for users to set k_{max} *a priori*. However, the critical issue of SMART is that only a naive distance measure was used as the clustering selection criterion. Additionally, OPTOC competitive learning is a spherical or hyper-spherical algorithm, which also limited the clustering performance.

In this paper, we improve the SMART framework by introducing the model order selection to select the best clustering. Moreover, to enhance the performance, we propose a SMART with finite mixture models (SMART-FMM) algorithm. It employs modified component-wise expectation maximization of mixtures (CEM^2) [10, 13] to fulfil splitting and merging. Once the SWM process terminates, the clustering selection algorithm plays a critical role in selecting the best clustering among the generated clusterings during the splitting procedure. A benchmark demonstration dataset is used to illustrate each step in the SMART flow. The main purpose of this paper is to develop the SMART framework and its implementation for microarray gene expression analysis. Thus, one real microarray gene expression dataset is studied using the proposed SMART-FMM. The numerical results show that our proposed method is superior and closer to human perception.

The rest of the paper is organized as follows: Sec. 2 describes the details of the SMART framework and demonstrates all the steps in the flow. Sec. 3 presents the SMART-FMM implementation. Sec. 4

The project (Ref. NIHR-RP-PG-0310-1004-AN) is supported by National Institute for Health Research (NIHR), UK. A. K. Nandi would like to thank TEKES for their award of the Finland Distinguished Professorship.

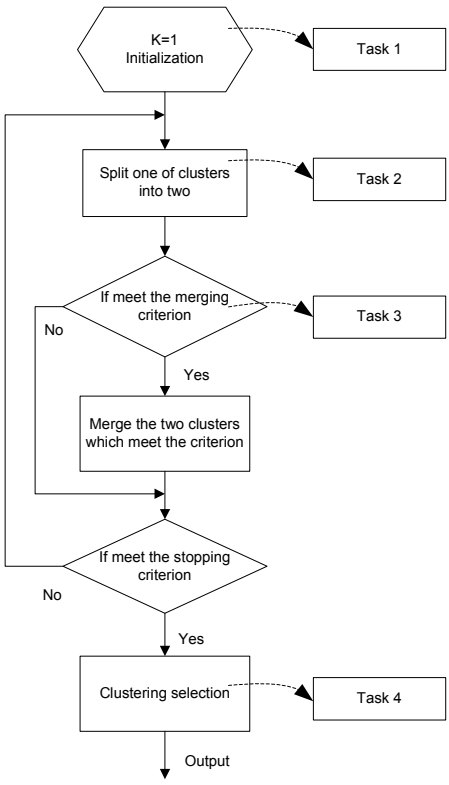

Fig. 1. The flow chart of the SMART framework.

briefly introduces the datasets explored in the paper and presents the numerical results. Finally, discussions and conclusions are given in Sec. 5.

2. SMART FRAMEWORK

In this section, we focus on the overview of the SMART framework, which is improved based on [11]. Suppose that we are going to partition the dataset $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_i | 1 \leq i \leq N\}$, where $\boldsymbol{x}_i \in \mathbb{R}^{M \times 1}$ denotes the i-th object, M is the dimension, and N is the number of objects. The flowchart of the framework is illustrated in Fig. 1.

The whole clustering procedure is divided into four tasks. SMART starts with one cluster ($K = 1$, where K is the number of clusters), and the cluster needs to be initialized, which is Task 1. Subsequently, the data goes through a SWM process, where splitting and merging are automatically conducted in iterations. In the splitting step of each iteration, which is labelled Task 2, SMART splits one cluster into two. After Task 2, the new clusters are tested by a merging criterion, which is associated with Task 3. If the condition for merging is satisfied, then merge the two clusters which meet the criterion, 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 according to Task 4.

Note that these tasks in the SMART flow can be completed using many clustering techniques in the literature, e.g., Task 1 can be done by any initialization technique either deterministic or random; Task 2 and 3 may be achieved by any splitting algorithm and merging criterion respectively or they may be combined into one algorithm; and Task 4 can be accomplished by any of either model order selection algorithms or validity indices. Different techniques will make the implementation slightly different but the flow does not change. Moreover, equipping different clustering algorithm brings different features into the framework and customizes SMART to different applications. In the next section, we will develop the SMART-FMM implementation, which uses MML [14, 15] as clustering selection algorithm and uses a termination criterion, namely the maximum number of merges N_{max}, in the SWM process. The logic behind the termination criterion is that normally merging will not start until the best clustering is reached. Once N_{max} is reached, the splitting and merging will terminate automatically.

3. SMART-FMM IMPLEMENTATION

Here, we present the principle of SMART-FMM, where the finite mixture model is employed and the key technique is the modified CEM^2 [10]. The greatest advantage of the modified CEM^2 is that the weaker component may naturally be excluded in the iterative process, which gives the stronger ones better chance of survival. From the merging point of view, it is a merging process combined with learning.

3.1. Finite Mixture Model

Let us assume that dataset $\boldsymbol{\mathcal{X}}$ follows a K-component finite mixture distribution. Its probability density function can be given by

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \alpha_k p(\boldsymbol{x}|\boldsymbol{\theta}_k), \tag{1}$$

where $\boldsymbol{\theta}$ contains the means $\{\boldsymbol{\mu}_k | k = 1, ..., K\}$ and the covariance matrices $\{\boldsymbol{\Psi}_k | k = 1, ..., K\}$, $\boldsymbol{\theta}_k$ is the parameter vector defining the k-th component and α_k, which has been mentioned, is the *a priori* mixing probability for the k-th component. Thus, for the whole dataset $\boldsymbol{\mathcal{X}}$, a set of N independent and identically distributed (i.i.d.) samples, the log-likelihood of the k-component mixture is

$$\log p(\boldsymbol{\mathcal{X}}|\boldsymbol{\theta}) = \log \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} \alpha_k p(\boldsymbol{x}_i|\boldsymbol{\theta}_k). \tag{2}$$

Thus, clustering a dataset $\boldsymbol{\mathcal{X}}$ becomes the discovery of the missing labels $\boldsymbol{\mathcal{Z}} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_N\}$ associated with the N data objects. The EM algorithm [16,17], which applies two steps, E (Expectation) step and M (Maximization) step, is a popular choice for obtaining maximum likelihood (ML) or maximum a posteriori (MAP) estimates of the mixture parameters.

3.2. The Modified Component-Wise EM of Mixtures

The original CEM^2 was proposed in [13] and modified in [10]. Unlike conventional EM algorithm, CEM^2 updates the model parameters $\{\boldsymbol{\theta}_k | 1 \leq k \leq K\}$ and the probabilities of components $\{\alpha_k | 1 \leq k \leq K\}$ sequentially, rather than simultaneously. In CEM^2, the estimation is also two-step process, but in each iteration, only one component has the opportunity to update its parameters. For the j-th component, it alternates the steps:

- **CEM^2 *E-step***: Compute for $i = 1, ..., N$ and $k = 1, ..., K$

$$\gamma_{k,i} \equiv E[\hat{z}_{k,i}|\boldsymbol{\mathcal{X}}, \hat{\boldsymbol{\theta}}] = \frac{\hat{\alpha}_k p(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}_k)}{\sum_{l=1}^{K} \hat{\alpha}_l p(\boldsymbol{x}_i|\hat{\boldsymbol{\theta}}_l)}. \quad (3)$$

- **CEM^2 *M-step***: Set

$$\hat{\alpha}_j^* = \frac{\sum_{i=1}^{N} \gamma_{j,i}}{\sum_{l=1}^{K}\sum_{i=1}^{N} \gamma_{l,i}}, \quad (4)$$

$$\hat{\boldsymbol{\theta}}_j^* = \arg\max_{\hat{\boldsymbol{\theta}}_j} \{\log p(\boldsymbol{\mathcal{X}}|\hat{\boldsymbol{\theta}})\} \quad (5)$$

For $l \neq j$, $\hat{\alpha}_l^* = \hat{\alpha}_l$ and $\hat{\boldsymbol{\theta}}_l^* = \hat{\boldsymbol{\theta}}_l$.

In [10], the adoption of Dirichlet-type prior for α_k's results in a new M-step

$$\hat{\alpha}_k^* = \frac{\max\left\{0, \sum_{i=1}^{N} \gamma_{k,i} - \frac{N_p}{2}\right\}}{\sum_{l=1}^{K}\left\{0, \sum_{i=1}^{N} \gamma_{l,i} - \frac{N_p}{2}\right\}}, \text{for } k = 1, 2, ..., K, \quad (6)$$

where N_p is the number of parameters which is required for each component. The corresponding components $\hat{\boldsymbol{\theta}}_k$'s with $\hat{\alpha}_k^* = 0$ is eliminated and become irrelevant. Modified CEM^2 can fulfil learning and merging, which are associated with Tasks 2 (only learning part) and 3, respectively, in SMART-FMM.

3.3. Minimum Message Length

Although there are a lot of model order selection algorithms and validity indices, to avoid losing our focus by comparing different selection algorithms, in this work, we choose MML [10, 14, 15] for Task 4. The rational behind the MML criteria is that if one can build a short code for the given data, then the code is a good data generation model. The shortest code length for set $\mathcal{X}$ is $\lceil -\log p(\boldsymbol{\mathcal{X}}|\boldsymbol{\theta}))\rceil$. If $p(\boldsymbol{\mathcal{X}}|\boldsymbol{\theta})$ is fully known to both the transmitter and receiver, they can build the same code and communication can proceed. However, if $\boldsymbol{\theta}$ is *a priori* unknown, the transmitter has to start by estimating and transmitting θ. This leads to a two-part message, whose total length is given by

$$\text{Length}(\boldsymbol{\theta}, \boldsymbol{\mathcal{X}}) = \text{Length}(\boldsymbol{\theta}) + \text{Length}(\boldsymbol{\mathcal{X}}|\boldsymbol{\theta}). \quad (7)$$

All minimum encoding length criteria state that the parameter estimate is the one minimizing Length($\boldsymbol{\theta}$, $\boldsymbol{\mathcal{X}}$). The criterion may be derived in the following form [10]

$$\text{Length}(\boldsymbol{\theta}, \boldsymbol{\mathcal{X}}) = \frac{N_p}{2}\sum_{k=1}^{K} \log \alpha_k + \frac{N_p+1}{2} K \log N - \log p(\boldsymbol{\mathcal{X}}|\boldsymbol{\theta}) + C, \quad (8)$$

where $\{\alpha_k, 1 \leq k \leq K\}$ is the mixing probability of the k-th component with the constraint $\sum_{k=1}^{K} \alpha_k = 1$, and $C = (N_p + 1)K(1 - \log 12)/2$ is a constant. Note that the components with zero-probability in α_k have been eliminated and K is the number of non-zero-probability components.

3.4. SMART-FMM Implementation

In SMART-FMM, we initially start with $K = 2$ because $K = 1$ does not need learning, but $K = 1$ is still included in the candidate list for selection in the output. Splitting process cannot be done by modified CEM^2 and has to be specified. Once all components converge and all zero-probability components are discounted, a new component will be injected into the framework. This new component is initialized deterministically by using the object farthest away from the closest component among all the components as the mean and averaged covariance matrix of all components' covariance matrices, as given by

$$\boldsymbol{\mu}_{K+1} = \arg\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}}\{\min_{1\leq k\leq K} \mathcal{D}(\boldsymbol{x}, \boldsymbol{\mu}_k)\}, \quad (9)$$

$$\boldsymbol{\Psi}_{K+1} = \frac{1}{K}\sum_{k=1}^{K}\{\boldsymbol{\Psi}_k\}, \quad (10)$$

where $\mathcal{D}(\cdot)$ is a distance metric, and then the clustering splits $K = (K + 1)$. The pseudo-code for SMART-FMM is in Table 1. The stage for recoding the candidate clustering is after all current components converge and all merges finish, and before the splitting for new component starts.

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

Task 1: Initializing SMART with $K = 2$
Randomly initialize $\hat{\boldsymbol{\theta}}_k$ and $\hat{\alpha}_k$ for $k = 1, 2$;
terminate = 0;
while !terminate **do**
  Tasks 2 & 3: Use modified CEM^2 for the learning and merging based on (3) and (6).
  if the prototype $\hat{\boldsymbol{\theta}}_k$ does not converge **then**
    Go back to Tasks 2 & 3;
  end if
  The stage for recoding candidate clustering.
  Splitting: Calculate the parameters for new components (9) and (10);
  if The number of merges is greater than or equal to N_m **then**
    terminate = 1;
  end if
end while
Task 4: Calculate the length for every converged clustering, output the clustering with the minimum length.

4. DATASETS AND NUMERICAL RESULTS

In this paper, one demonstration dataset, which is a 3-component bivariate mixture model [10], is used to illustrate each step in the SMART flow. Since we have more interests in the microarray gene expression data analysis, we also study a real microarray gene expression datasets, which is a subset of the leukemia dataset [3]. The SMART-FMM algorithm is compared with SSMCL, ULFMM and SMART-CL in the experiments. In the following experiments, the parameter $N_m = 5$ for both SMART-FMM and SMART-CL; $k_{max} = 30$ for both SSMCL and ULFMM.

The demonstration example (Demo) is a 3-component bivariate Gaussian mixture dataset used in [10], whose mixture probabilities are α_1 = α_2 = α_3 = 1/3, with mean vectors at $[0, -2]^T$, $[0, 0]^T$, $[0, 2]^T$, and equal covariance matrices of diag$\{4, 0.4\}$. The covariance matrices are diag$\{2, 0.2\}$ in [10], but we double them in our study because we try to discern the best algorithm by enlarging the differences among their performances. We repeat the clustering experiments 1000 times for each method. In our study, three metrics are investigated: adjusted RAND index [18, 19], correct selection rate (CSR) of number of clusters, and the statistics

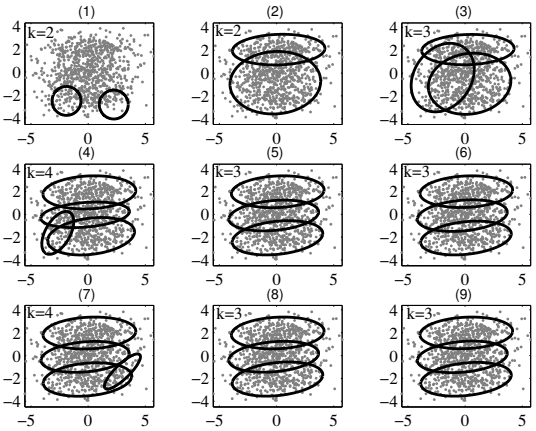

Fig. 2. The demonstration of SMART-FMM using Gaussian mixture dataset in Demo example. The sub-figures (1)-(8) demonstrate the procedure of SMART-FMM. The sub-figures (9) is the final clustering result. Parameter setting: $N_m = 5$.

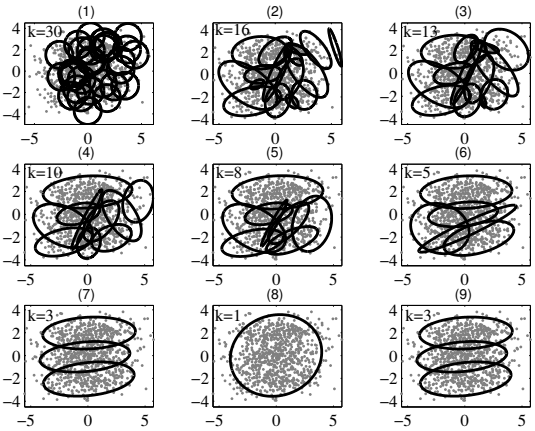

Fig. 3. The demonstration of ULFMM using Gaussian mixture dataset in Demo example. The sub-figures (1)-(8) demonstrate the procedure of ULFMM. The sub-figures (9) is the final clustering result. Parameter setting: $k_{max} = 30$.

Table 2. Performance comparison of three metrics, including Adjusted RAND, CSR and mean($\hat{K}$) $\pm$ std($\hat{K}$), for all algorithms in Demo example.

Algorithms	Adj RAND	CSR	mean($\hat{K}$) $\pm$ std($\hat{K}$)
SSMCL	0	0	1 ± 0
SMART-CL	18.85%	14.60%	3.26 ± 1.4
ULFMM	64.13%	88.1%	3.21 ± 0.71
SMART-FMM	69.52%	100%	3 ± 0

Table 3. Performance comparison of three metrics, including Adjusted RAND, CSR and mean($\hat{K}$)$\pm$std($\hat{K}$), for all algorithms in Leukemia dataset.

Algorithms	Adj RAND	CSR	mean($\hat{K}$) $\pm$ std($\hat{K}$)
SSMCL	0	0	1 ± 0
SMART-CL	97.98%	99.0%	2.99 ± 0.13
ULFMM	93.15%	69.4%	3.23 ± 0.54
SMART-FMM	98.5%	100%	3 ± 0

of estimated number of clusters $\hat{K}$ (mean($\hat{K}$) $\pm$ std($\hat{K}$)). Adjusted RAND is averaged over total number of experiments (which is 1000 in our case), and CSR is the ratio of the times of correct selections of number of clusters to total number of experiments. It is shown in Table 2 that SMART-CL and SSMCL do not work properly, where SMART-CL has only 14.60% in CSR and 18.85% in adjusted RAND, and SSMCL has no correct selections. The reason is that the competitive learning is a spherical or hyper-spherical algorithm so it is not suitable for the clustering of elliptical or hyper-elliptical datasets. It is worth noting that the proposed SMART-FMM has 100% CSR in the experiment and its adjusted RAND is also the highest. The clustering procedures of SMART-FMM and ULFMM are shown in Fig. 2 and 3, respectively. This demo shows how the mechanism of SMART is working. To some extent, it also shows that the SMART framework is more effective and more practical than ULFMM, because it is not necessary for SMART to set k_{max}.

The real microarray gene expression dataset consists of 38 bone marrow samples obtained from acute leukemia patients at time of diagnosis. There are 999 genes in the dataset [20]. The biological truth is that the samples include 3 groups: 11 acute myeloid leukemia (AML) samples, 8 T-lineage acute lymphoblastic leukemia (ALL) samples, and 19 B-lineage ALL samples [3, 20, 21]. Thus, the 999 genes are grouped into 3 clusters, and each group has 333 genes. We repeat the clustering experiments 1000 times for each method. The results are shown in Table 3. SMART-FMM has the superior performance as in the Demo experiment. It always provides 100% CSR and the highest adjusted RAND value. Impressively, SMART-CL has significantly better performance than ULFMM and has nearly 30% more CSR and 5% more adjusted RAND. In terms of mean and standard deviation of $\hat{K}$, SMART-CL has closer mean to the true value and significantly smaller standard deviation than ULFMM. SSMCL totally fails in this experiment, as it always converges to one cluster, which reveals that its merging criterion does not fit the data at all.

5. DISCUSSIONS AND CONCLUSIONS

We proposed a splitting-while-merging algorithm with finite mixture models (FMM) built on an improved splitting merging awareness tactics (SMART). The SMART framework was originally proposed in [11]. The main property of SMART is that it does not require any dataset-dependent parameters or *a priori* knowledge. However, the original SMART employed a naive distance measure as the clustering selection criterion. Additionally, its implementation used OPTOC competitive learning [8], which is a hyper-spherical algorithm. These configurations limited the performance critically. The improved SMART framework integrates clustering selection criterion, which plays a vital role in the new algorithm. Inspired by [10], in the SMART-FMM implementation, the modified component-wise EM of mixtures is employed as learning and merging technique and a model order selection algorithm is used as clustering selection criterion. One demonstration example and one real microarray gene expression dataset were studied using our approach. The numerical results showed that SMART-FMM is superior and more effective than others. Most importantly, it is closer to human perception and does not need any *a priori* knowledge about the datasets.

6. REFERENCES

[1] D. X. Jiang, C. Tang, and A. D. Zhang, "Cluster analysis for gene expression data: A survey," *IEEE Trans. Know. and Data Eng.*, vol. 16, no. 11, pp. 1370–1386, 2004.

[2] R. Xu and D. II Wunsch, "Clustering Algorithms in Biomedical Research: A Review," *IEEE Reviews in Biomedical Engineering*, vol. 3, pp. 120–154, 2010.

[3] T. R. Golub, *et al.*, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[4] P. Tamayo, *et al.*, "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation," *Proc. Nat. Academy Sci. USA*, vol. 96, no. 6, pp. 2907–2912, 1999.

[5] Doulaye Dembélé and Philippe Kastner, "Fuzzy c-means method for clustering microarray data," *Bioinformat.*, vol. 19, no. 8, pp. 973–980, 2003.

[6] Jie Qin, Darrin P. Lewis, and William Stafford Noble, "Kernel hierarchical gene clustering from microarray expression data," *Bioinformat.*, vol. 19, no. 16, pp. 2097–2104, 2003.

[7] Sanghamitra Bandyopadhyay, Anirban Mukhopadhyay, and Ujjwal Maulik, "An improved algorithm for clustering gene expression data," *Bioinformat.*, vol. 23, no. 21, pp. 2859–2865, 2007.

[8] Ya-Jun Zhang and Zhi-Qiang Liu, "Self-splitting competitive learning: a new on-line clustering paradigm," *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 369–380, 2002.

[9] S. H. Wu, A. W.-C. Liew, H. Yan, and M. S. Yang, "Cluster analysis of gene expression data based on self-splitting and merging competitive learning," *IEEE Trans. Inf. Tech. in Biomed.*, vol. 8, no. 1, pp. 5–15, 2004.

[10] M. Figueiredo,and A. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, 24, pp. 381–396, 2002.

[11] Rui Fa and Asoke K. Nandi, "SMART: Novel Self Splitting-Merging Clustering Algorithm," *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, 2012.

[12] Cheng-Ru Lin and Ming-Syan Chen, "Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging," *IEEE Trans. Know. and Data Eng.*, vol. 17, no. 2, pp. 145–159, 2005.

[13] G. Celeux, S. Chretien, F. Forbes and A. Mkhadri, "A Component-wise EM Algorithm for Mixtures," *Technical Report 3746*, INRIA Rhone-Alpes, France, 1999. Availability: http://hal.inria.fr/docs/00/07/29/16/PDF/RR–3746.pdf

[14] J. Oliver, R. Baxter and C. S. Wallace, "Unsupervised Learning Using MML," *Proceedings of the Thirteenth International Conference In Machine Learning (ICML 96)*, pp. 364–372, 1996.

[15] C. S. Wallace and Dowe, D. L., "Minimum Message Length and Kolmogorov Complexity," *The Computer Journal*, vol. 42, no. 4, pp. 270–283, 1999.

[16] A. P. Dempster, N. M. Laird and D. B. Rubin "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[17] C. Fraley and A. Raftery, "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis," *Technical Report 329*, Dept. Statistics, Univ. Washington, Seattle, WA, 1998.

[18] W. M. Rand "Objective criteria for the evaluation of clustering methods", *Journal of the American Statistical Association*, 66(336), 1971.

[19] L. Hubert and P. Arabie, "Comparing Partitions", *Journal of Classification*, 2(1): 193-218, 1985.

[20] S. Monti, P. Tamayo, J. Mesirov and T. Golub, "Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data", *Machine Learning*, 52(1), pp. 91–118, 2003.

[21] Rui Fa and Asoke K. Nandi, "Comparisons of validation criteria for clustering algorithms in microarray gene expression data analysis," in *The second Int. Workshop on Genomic Sig. Proc. (GSP2011)*, 2011.