Split–Merge Incremental LEarning (SMILE) of Mixture Models

Konstantinos Blekas and Isaac E. Lagaris

Department of Computer Science, University of Ioannina, 45110 Ioannina, Greece
{kblekas,lagaris}@cs.uoi.gr

Abstract. In this article we present an incremental method for building a mixture model. Given the desired number of clusters $K \geq 2$, we start with a two-component mixture and we optimize the likelihood by repeatedly applying a *Split-Merge* operation. When an optimum is obtained, we add a new component to the model by splitting in two, a properly chosen cluster. This goes on until the number of components reaches a preset limiting value. We have performed numerical experiments on several data–sets and report a performance comparison with other rival methods.

1 Introduction

Clustering, apart from being on its own a challenging field of research, is useful to a wide spectrum of application areas, such as pattern recognition, machine learning, computer vision, bioinformatics, etc. The large interest of the scientific community for the problem of clustering is reflected by the growing appearance of related monographs [1],[2],[3],[4], journal articles and conferences. With the advent of the Internet and the World Wide Web, scientific data from a wide range of fields have become easily accessible. This convenience has further raised the interest and expanded the audience of clustering techniques. Clustering can be viewed as the identification of existing intrinsic groups in a set of unlabeled data. Associated methods are often based on intuitive approaches that rely on specific assumptions and on the particular characteristics of the data sets. This in turn implies that the corresponding algorithms depend crucially on some parameters that must be properly tuned anew for each problem.

A plethora of clustering approaches has been presented over the last years. Hierarchical methods are based on a tree structure over the data according to some similarity criteria. Methods based on partitioning, relocate iteratively the data points into clusters until the optimum position of some cluster representatives (e.g. centers) is found; the popular "K-means" algorithm for instance belongs to this category. On the other hand, model-based methods are closer to the natural data generation mechanism and assume a mixture of probability distributions, where each component corresponds to a different cluster[1],[3],[4]. In these methods the *Expectation-Maximization* (EM) algorithm [5] is the preferred framework for estimating the mixture parameters due both to its simplicity and flexibility. Moreover, mixture modeling provides a powerful and useful platform for capturing data with complex structure. A fundamental concern in applying

J. Marques de Sá et al. (Eds.): ICANN, Part II, LNCS 4669, pp. 291–300, 2007.

the EM algorithm, is its strong dependence on the initialization of the model parameters. Improper initialization may lead to points corresponding to local (instead of global) maxima of the log-likelihood, a fact that in turn may weigh on the quality of the method's estimation capability. Attempts to circumvent this, using for example the K-means algorithm to initialize the mixture parameters, amounts to shifting the original problem to initializing the K-means. Recently, several methods have been presented, aiming to overcome the problem of poor initialization. They are all based on an incremental strategy for building a mixture model. In most cases these methods start from a single component and iteratively add new components to the mixture either by performing a split procedure [6], or by performing a combined scheme of global and local search over a pool of model candidates [7]. A similar in nature technique is to follow an entirely opposite route and start with several components that iteratively will be discarded [8]. An alternative strategy has been presented in [9] where a split-and-merge EM (SMEM) algorithm was proposed. Initially the SMEM method performs the usual EM algorithm to a K-order mixture model and an initial estimation of the parameters. At a second level, repeated split-merge operations are performed exhaustively among the K components of the mixture model that re-estimate the model parameters until a termination criterion is met.

The idea of the SMILE method is to start with a mixture model with $k = 2$ and then to apply a Split & Optimize, Merge & Optimize (SOMO) sequence of operations. If this leads to a model with higher likelihood we accept it and repeat the SOMO procedure. In the opposite case we choose the model created just after the Split & Optimize (SO) step, which corresponds to a mixture model with an additional component. This is continued up to a preset number of components. At that stage if the SOMO sequence does not produce a higher likelihood value, the algorithm concludes. We have tested SMILE on a suite of benchmarks, with both simulated and real data sets, taking in account a variety of cases, with promising results. Comparisons have been made with existing methods of similar nature. The quality of the solutions offered by each method is rated in terms of the associated log-likelihood value. An important test for SMILE is its application to image segmentation problems. Here we have considered data arising from MRI images and the results are quite encouraging.

The rest of the paper is organized as follows. In section 2 we present the mixture models and the EM algorithm for parameter estimation, in section 3 we present in detail our incremental scheme where we lay out an algorithmic description, while in section 4 we report results obtained by applying SMILE to several data sets. Our conclusions and a summary are included in section 5 along with some remarks and speculations.

2 Mixture Models

Given a set of N data points $A = \{x_i | x_i \in R^d,\ i = 1, \cdots, N\}$, the task of clustering is to find a number of K subsets $A_j \subset A$ with $j = 1, \cdots, K$, containing points with common properties. These subsets are called *clusters*. We consider here that the properties of a single cluster j, may be described implicitly via a probability distribution with parameters θ_j.

A mixture model is a linear combination of these cluster-distributions, e.g.:

$$f(x|\Theta_K) = \sum_{j=1}^{K} \pi_j p(x|\theta_j) \tag{1}$$

The parameters $0 < \pi_j \leq 1$ represent the mixing weights satisfying $\sum_{j=1}^{K} \pi_j = 1$, while $\Theta_K = \{\pi_j, \theta_j\}_{j=1}^{K}$ represents the vector of all unknown model parameters. Mixture models provide an efficient method for describing complex data sets. The parameters can be estimated by maximizing the log-likelihood, by using for example the EM algorithm [5]. EM performs a two-step iterative procedure: The E-step calculates the posterior probabilities:

$$z_{ij}^{(t)} = p(j|x_i, \theta_j^{(t)}) = \frac{\pi_j^{(t)} p(x_i|\theta_j^{(t)})}{\sum\limits_{l=1}^{K} \pi_l^{(t)} p(x_i|\theta_l^{(t)})} \ , \tag{2}$$

while the M-step updates the model parameters by maximizing the complete log-likelihood function. If we assume multivariate Normal densities $\theta_j = \{\mu_j, \Sigma_j\}$ maximization yields the following updates:

$$\pi_j^{(t+1)} = \frac{\sum_{i=1}^{N} z_{ij}^{(t)}}{N} \ , \ \mu_j^{(t+1)} = \frac{\sum\limits_{i=1}^{N} z_{ij}^{(t)} x_i}{\sum\limits_{i=1}^{N} z_{ij}^{(t)}}, \ \Sigma_j^{(t+1)} = \frac{\sum\limits_{i=1}^{N} z_{ij}^{(t)} (x_i - \mu_j^{(t+1)})(x_i - \mu_j^{(t+1)})^T}{\sum\limits_{i=1}^{N} z_{ij}^{(t)}} \ . \tag{3}$$

3 *Split–Merge* Mixture Learning

In this section we describe in detail the proposed method. To begin with, we describe the operations used in the SOMO sequence.

3.1 The Split Operation

Suppose that the model currently contains $k \geq 2$ components (clusters). The selection of the cluster to be split is facilitated with one of the criteria below.

1. Maximum Entropy:

$$H(j) = -\int p(x|\theta_j) \log p(x|\theta_j) dx \tag{4}$$

2. Minimum Mean Local Log-likelihood:

$$L(j) = \frac{\sum_{i=1}^{N} p(j|x_i \theta_j) \log(p(x_i|\theta_j))}{\sum_{i=1}^{N} p(j|x_i, \theta_j)} \tag{5}$$

3. Maximum local *Kullback divergence*: (used also by SMEM [9])

$$J(j) = \int f(x|\Theta) \log \frac{f(x|\Theta)}{p(x|\theta_j)} dx \tag{6}$$

   where the density $f(x|\Theta)$ represents an empirical distribution [9].

Suppose that cluster j^* is being selected for the split operation. Two clusters are then created labeled as j_1^* and j_2^*. Their parameters are initialized as follows:

$$\pi_{j_1^*} = \pi_{j_2^*} = \frac{\pi_{j^*}}{2} \quad , \Sigma_{j_1^*} = \Sigma_{j_2^*} = \frac{\Sigma_{j^*}}{2} \tag{7}$$

$$\mu_{j_1^*} = \mu_{j^*} + \frac{\sqrt{\lambda_{max}}}{2} v_{max} \quad , \mu_{j_2^*} = \mu_{j^*} - \frac{\sqrt{\lambda_{max}}}{2} v_{max} \ , \tag{8}$$

where λ_{max}, v_{max} are the maximum eigenvalue and its corresponding eigenvector of the covariance matrix Σ_{j^*} .

3.2 The Optimization Operation

Let $f(x|\Theta_k^*)$ be the mixture without the j^*-th component, i.e.:

$$f(x|\Theta_k^*) = f(x|\Theta_k) - \pi_{j^*} p(x|\theta_{j^*}) = \sum_{j=1, j \neq j^*}^{k} \pi_j p(x|\theta_j) \tag{9}$$

The resulting mixture after the split operation takes the following form:

$$f(x|\Theta_{k+1}) = f(x_i|\Theta_k^*) + (\pi_{j^*} - \alpha) p(x|\theta_{j_1^*}) + \alpha p(x|\theta_{j_2^*}) \tag{10}$$

with $0 \leq \alpha \leq \pi_{j^*}$. In this mixture, the first term is inherited from the original model, while the rest two, are the newly introduced components by the split operation. The first term remains intact, while the other two are to be adjusted so as to maximize the likelihood. This is facilitated by a partial application of the EM algorithm, that modifies only the new component parameters $\alpha, \theta_{j_1^*}, \theta_{j_2^*}$. The following updates are obtained:

At the E-step:

$$z_{ij_1^*}^{(t)} = \frac{(\pi_{j^*} - \alpha^{(t)}) p(x_i|\theta_{j_1^*}^{(t)})}{f(x|\Theta_{k+1}^{(t)})}, \quad z_{ij_2^*}^{(t)} = \frac{\alpha^{(t)} p(x_i|\theta_{j_2^*}^{(t)})}{f(x|\Theta_{k+1}^{(t)})} \ , \tag{11}$$

and at the M-step:

$$\alpha^{(t+1)} = \pi_{j^*} \frac{\sum_{i=1}^{N} z_{ij_2^*}^{(t)}}{\sum_{i=1}^{N} z_{ij_2^*}^{(t)} + z_{ij_1^*}^{(t)}}, \tag{12}$$

$$\mu_m^{(t+1)} = \frac{\sum_{i=1}^{N} z_{im}^{(t)} x_i}{\sum_{i=1}^{N} z_{im}^{(t)}}, \ \Sigma_m^{(t)} = \frac{\sum_{i=1}^{N} z_{im}^{(t)} (x_i - \mu_m^{(t+1)})(x_i - \mu_m^{(t)})^T}{\sum_{i=1}^{N} z_{im}^{(t)}}, \ m = \{j_1^*, j_2^*\} \tag{13}$$

After obtaining an optimum in the subspace of the newly introduced parameters, a full space optimization is performed, again by the EM algorithm (Eqs. 2, 3).

3.3 The Merge Operation

During this operation two clusters are fused into one. Two clusters $\{k_1, k_2\}$ are selected according to any of the criteria that follow.

1. Minimum Distribution Distance (*Symmetric Kullback Leibler*)

$$\int p(x|\theta_{k_1}) \log \frac{p(x|\theta_{k_1})}{p(x|\theta_{k_2})} dx + \int p(x|\theta_{k_2}) \log \frac{p(x|\theta_{k_2})}{p(x|\theta_{k_1})} dx \tag{14}$$

2. Maximum Distribution Overlap (used also in [9])

$$\sum_{i=1}^{N} p(k_1|x_i, \theta_{k_1}) p(k_2|x_i, \theta_{k_2}) \tag{15}$$

Let the resulting cluster be labeled by k. Its parameters are then initialized as:

$$\pi_k = \pi_{k_1} + \pi_{k_2} \ , \mu_k = \frac{\pi_{k_1} * \mu_{k_1} + \pi_{k_2} * \mu_{k_2}}{\pi_{k_1} + \pi_{k_2}} \ , \Sigma_k = \frac{\pi_{k_1} * \Sigma_{k_1} + \pi_{k_2} * \Sigma_{k_2}}{\pi_{k_1} + \pi_{k_2}} \tag{16}$$

The optimization step following the merge operation is in the same spirit as that of section 3.2, e.g. we perform partial EM steps, allowing only the new (merged) cluster parameters to vary. After obtaining an optimum in the subspace of the newly introduced parameters, a full space EM optimization is performed.

3.4 Description of the Method

Initially we construct a mixture with two components, i.e. $k = 2$. Denote by Θ_k^1 the mixture parameters, and by $L(\Theta_k^1)$ the corresponding value of the log-likelihood function. We perform in succession a split and an optimization operation, obtaining so a model Θ_{k+1}^m with $k+1$ components. Similarly in what follows, we perform a merge and an optimization operation, that creates a model again with k components. Let Θ_k^2 be the new mixture parameters after this split-merge operation and $L(\Theta_k^2)$ the corresponding log-likelihood. If $L(\Theta_k^2) > L(\Theta_k^1)$ then we update the k-order model to Θ_k^2 and we repeat the SOMO procedure. In the case where $L(\Theta_k^2) \leq L(\Theta_k^1)$, i.e. when the SOMO procedure fails to obtain a better value for the likelihood, we discard the last merge operation and update our model to Θ_{k+1}^m, which was obtained after the last SO operation, with $k+1$ components. The algorithm proceeds so, until we obtain a model with the prescribed number of components (K) and the SOMO iterations fail to provide further improvement to the likelihood.

We now can proceed and describe our method algorithmically.

```
– Start with $k = 2$
– while $k < K$
  1. Estimate the current log-likelihood $L_1$
  2. Perform SOMO operation:
     • Split: select a cluster $j^*$ and divide it into two clusters $j_1^*$ and $j_2^*$.
     • Optimization operation: Perform partial-EM and then full EM.
     • Merge: select two clusters $k_1$ and $k_2$ and merge them.
     • Optimization operation: Perform partial-EM and then full EM.
     • Estimate the log–likelihood $L_2$
  3. if $L_2 > L_1$ then:
       Accept the fused cluster. Set $L_1 \leftarrow L_2$ and go to step 2.
     else:
       Reject the last merge operation. Set $k \leftarrow k + 1$.
– endwhile
```

4 Experimental Results

We have performed several experiments to examine the effectiveness of the SMILE method. We have considered both real and simulated data sets of varying dimensionality. We compare against three incremental approaches, namely the Greedy EM method[1] [7], the Split and Merge EM (SMEM) [9], the MML-EM [2] [8], as well as with the simple K-means initialized EM. The initialization scheme in SMILE is predetermined in distinction to the contestant schemes that depend heavily on random numbers. Hence, in order to obtain a meaningful comparison, we performed 30 different runs for each data set with different seeds and kept records of the mean value and the standard deviation of the log-likelihood.

Experiments with simulated data sets

In Fig. 1 we give an example of the performance of our algorithm in a typical 2-dimensional data set that has been generated from a $K = 5$ Gaussian mixture. Note that ellipses were created by covariances. Step 0 shows the solution with one cluster, which in step 1 is split into two, with a log-likelihood estimation $L_1 = -1727$. Then, SMILE tests the optimality of this solution by performing a SOMO procedure (steps 2a, 2b), leading to a solution with $L_2 = -1973$. Since the SOMO fails ($L_2 < L_1$), we discard the last MO operation leading to a $K = 3$ mixture model and continue with the next SOMO process (steps 3a, 3b). In this case, this SOMO operation found a better solution $L_2 = -1272$ (step 3b) in comparison with the one $L_1 = -1537$ of step 2a. Therefore, we accept this updated $K = 3$ model and perform another SOMO operation (steps 4a,

[1] The software was downloaded from http://staff.science.uva.nl/∼vlassis/software/

[2] The software was downloaded from http://www.lx.it.pt/∼mtf/

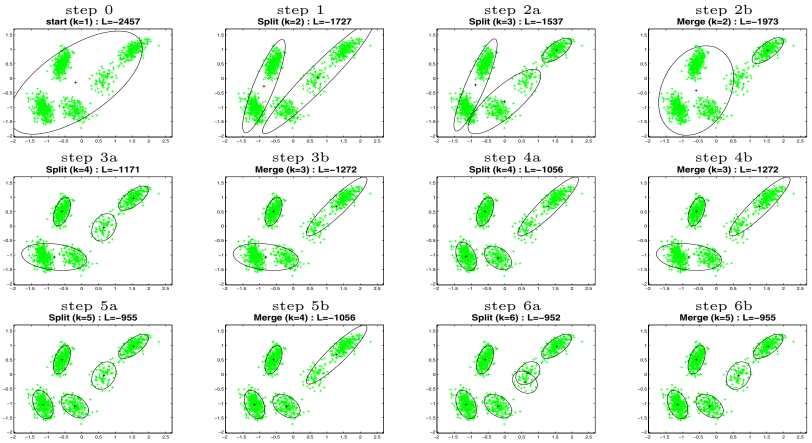
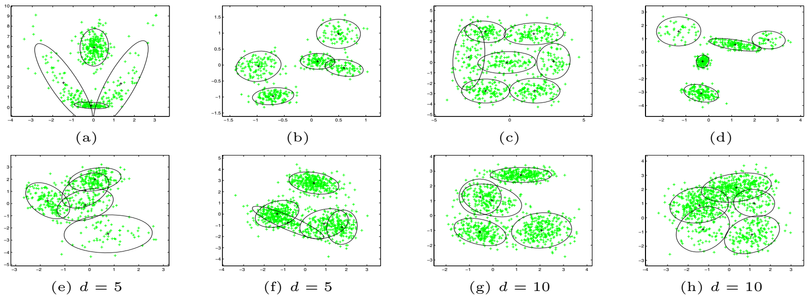


Fig. 1. Visualization of the SMILE steps on a typical data set. Each figure shows the current clusters and the corresponding value of the log-likelihood.


(a) (b) (c) (d)

(e) $d = 5$ (f) $d = 5$ (g) $d = 10$ (h) $d = 10$

Fig. 2. Simulated data sets used during experiments. We give also the clustering solution obtained by our method, i.e. their centers and the elliptic shapes.

4b), which however fails to further improve the likelihood. Finally, two other SOMO calls are made that both fail before the final $K = 5$ solution is reached (step 5a).

Several experiments were conducted using simulated data sets created by sampling from Gaussian mixture models. Figure 2 illustrates eight (8) such data sets containing $N = 500$ points. The first four sets (a,b,c,d) are in 2-dimensions, while (e,f) (g,h) are in 5 and 10-dimensions respectively. The visualization for the sets with dimensionality 5 and 10, is performed by projecting on the plane spanned by the first two principal components. The clustering obtained by SMILE is displayed in Fig. 2. Table 1 summarizes the results obtained by the application of the five contestants to the above mentioned data sets. Note that SMILE has recovered the global maximum in all cases; from the rest, only the Greedy EM and SMEM methods yielded comparable results. For the data set of Fig.2c, SMILE was the only method that obtained the global solution.

Table 1. Comparative results obtained form the experiments in data sets of Fig. 2

Data set	*SMILE*	*Greedy EM*	*SMEM*	*MML-EM*	*K-means EM*
(a)	−2.82	−2.87(0.02)	−2.82(0.00)	−2.86(0.05)	−2.89(0.01)
(b)	−0.83	−0.83(0.01)	−0.85(0.02)	−0.98(0.13)	−0.86(0.05)
(c)	−3.92	−3.94(0.01)	−3.94(0.00)	−3.95(0.02)	−3.93(0.01)
(d)	−1.87	−1.87(0.00)	−1.89(0.04)	−2.00(0.17)	−1.99(0.19)
(e)	−4.54	−4.63(0.04)	−4.55(0.02)	−4.59(0.04)	−4.61(0.02)
(f)	−4.74	−4.74(0.00)	−4.75(0.03)	−4.80(0.07)	−4.85(0.14)
(g)	−7.19	−7.19(0.00)	−7.19(0.00)	−7.40(0.09)	−7.44(0.33)
(h)	−7.71	−7.71(0.00)	−7.75(0.16)	−7.82(0.03)	−7.83(0.23)

Experiments with real data sets

Additional experiments were made using real data sets. In particular, we have selected two widely used benchmarks. The first one is the CRAB data set of Ripley [2], that contains $N = 200$ data belonging to four clusters ($K = 4$). Original CRAB data are in five dimensions. Here we have also created a 2-dimensional data set by projecting the data on the plane defined by the second and third principal components. We have also considered the renowned Fisher-IRIS data set [10] with $N = 150$ points in $d = 4$ dimensions belonging to three clusters ($K = 3$). In Table 2 we summarize the results obtained by the 5 contestants. Note, that in the case of the original CRAB data set, SMILE was the only one that recovered the optimal solution.

Another experimental benchmark used is the Phoneme data set [10]. This is a collection of two-class five dimensional data points. In our study we have randomly selected a training set with $N = 2800$ and a test set with 2604 data points. Figure 3 illustrates the performance of each method by plotting the log-likelihood value versus the number of components $K = [2, 10]$, in both the training and

Table 2. Comparative results obtained from the CRAB and the IRIS data sets

Data set	*SMILE*	*Greedy EM*	*SMEM*	*MML-EM*	*K-means EM*
CRAB $d = 5$	−6.14	−6.35(0.14)	−6.35(0.12)	−6.86(0.01)	−6.60(0.19)
CRAB $d = 2$	−2.49	−2.50(0.01)	−2.50(0.00)	−2.55(0.06)	−2.52(0.06)
IRIS $d = 4$	−1.21	−1.23(0.02)	−1.23(0.04)	−1.25(0.04)	−1.28(0.09)

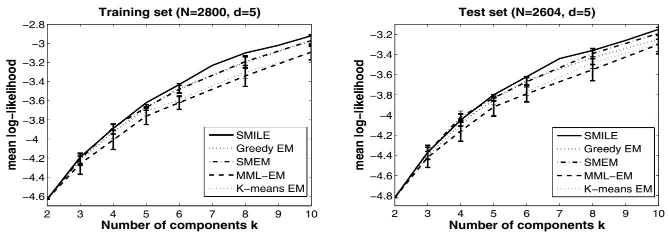


Fig. 3. Plots of mean log-likelihood objective function estimated by each method against number of components K to the Phoneme data set

the test sets. Observe that SMILE's curve is consistently above all others, both for the training and for the test set, implying superiority in performance and in generalization as well. Note that again, there were cases where SMILE was the only method that arrived at the global solution.

Application in image segmentation

In computer vision clustering finds application in image segmentation, i.e. the grouping of image pixels based on attributes such as their intensity and spatial location. We have tested SMILE to simulated brain MRI images available on the site BrainWeb [11], where we have reduced them into half of their original size (181 × 217). The segmentation of MRI mainly requires the classification of the brain into three types of tissue: (GM, WM, CSF). Since we are aware of the true class labels of the pixels we evaluate each method according to the computed total classification error. Figure 4 illustrates four such MRI images together with the segmentation result using a $K = 5$ Gaussian mixture, where in the reconstructed images every pixel assumes the intensity value of the cluster center that belongs. The overall classification error obtained from all the clustering methods to these images are presented at Table 3. It is obvious that our method achieves superior results for the tissue segmentation.

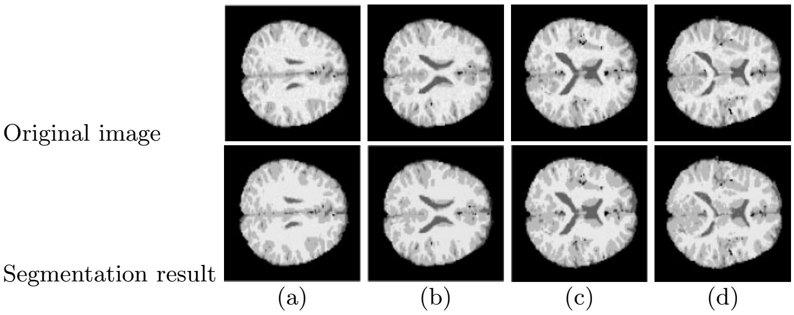


Fig. 4. Image segmentation results obtained by our method in four MRI images

Table 3. Percentage of misclassified pixels for the MRI of Fig.4 using $K = 5$ Gaussians

MRI image	*SMILE*	*Greedy EM*	*SMEM*	*MML-EM*	*K-means EM*
(a)	36.76	37.24(0.23)	37.12(0.00)	38.31(0.94)	37.84(0.01)
(b)	35.88	36.50(0.04)	36.69(0.00)	37.48(0.89)	36.57(0.08)
(c)	35.48	35.60(0.12)	36.18(0.30)	37.05(0.48)	36.21(0.29)
(d)	37.88	38.20(0.19)	37.98(0.00)	39.37(0.58)	38.90(0.32)

5 Conclusion

In this study we have presented SMILE, a new incremental mixture learning method based on successive split and merge operations. Starting from a two-component mixture, the method performs split-merge steps to improve the current solution maximizing the log-likelihood. SMILE has the advantage of not relying on good initial estimates, unlike the other rival methods studied in this article. The results of the comparative study are very promising. Several developments are possible that need further research. For example, consecutive multiple split operations followed by corresponding merge steps may lead to even better models. The persistent issue of discovering the optimal number of clusters in a data set may be examined in the framework of this method as well.

Acknowledgments

This work was partially supported by Interreg IIIA (Greece-Italy) Grant (No I2101005).

References

1. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, San Diego (1990)
2. Ripley, B.: Pattern Recognition and Neural Networks. Cambridge Univ. Press Inc, Cambridge, UK (1996)
3. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley-Interscience, New York (2001)
4. Bishop, C.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
5. Dempster, A., Laird, N., Rubin, D.: Maximum Likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc. B 39, 1–38 (1977)
6. Li, J., Barron, A.: Mixture density estimation. In: Advances in Neural Information Processing Systems, pp. 279–285. The MIT Press, Cambridge, MA (2000)
7. Vlassis, N., Likas, A.: A greedy EM algorithm for Gaussian mixture learning. Neural Processing Letters 15, 77–87 (2001)
8. Figueiredo, M., Jain, A.: Unsupervised learning of finite mixture models. IEEE Trans. Pattern Analysis and Machine Intelligence 24(3), 381–396 (2002)
9. Ueda, N., Nakano, R., Ghahramani, Z., Hinton, G.: SMEM algorithm for mixture models. Neural Computation 12(9), 2109–2128 (2000)
10. Merz, C., Murphy, P.: UCI repository of machine learning databases.Irvine, CA (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
11. Cocosco, C., Kollokian, V., Kwan, R.S., Evans, A.: BrainWeb: Online interface to a 3D MRI simulated brain database. NeuroImage 5(3), 2/4, S425 (1997)