Genetic-Based EM Algorithm for Learning Gaussian Mixture Models

Franz Pernkopf, *Member*, *IEEE*, and
Djamel Bouchaffra, *Senior Member*, *IEEE*

Abstract—We propose a genetic-based expectation-maximization (GA-EM) algorithm for learning Gaussian mixture models from multivariate data. This algorithm is capable of selecting the number of components of the model using the *minimum description length* (MDL) criterion. Our approach benefits from the properties of *Genetic algorithms* (GA) and the EM algorithm by combination of both into a single procedure. The population-based stochastic search of the GA explores the search space more thoroughly than the EM method. Therefore, our algorithm enables escaping from local optimal solutions since the algorithm becomes less sensitive to its initialization. The GA-EM algorithm is *elitist* which maintains the monotonic convergence property of the EM algorithm. The experiments on simulated and real data show that the GA-EM outperforms the EM method since: 1) We have obtained a better MDL score while using exactly the same termination condition for both algorithms. 2) Our approach identifies the number of components which were used to generate the underlying data more often than the EM algorithm.

Index Terms—Unsupervised learning, clustering, Gaussian mixture models, EM algorithm, Genetic algorithm, minimum description length.

1 Introduction

FINITE mixture models [10], [14] are flexible methods for modeling complex probability distribution functions. These models enable statistical modeling of environments with multimodal behavior where simple parametric models fail to represent the characteristics of the data adequately. The standard approach for learning the parameters of the mixture model is the EM algorithm [5]. The EM algorithm converges to a local optimum and the result is sensitive to initialization. Additionally, the EM algorithm assumes that the number of components for modeling the distributions is known. This is not the case for many applications. In such cases, a set of candidate models is established by applying the algorithm for a different number of components. The best model is selected according to a model selection criterion.

Recently, many adaptations and extensions of the EM approach have been proposed in order to address the problem of convergence to a local optimum and the initialization issue. Figueiredo and Jain [7] suggested a component-wise EM algorithm [3] which is robust regarding initialization and capable of selecting the number of components used for generating the data. Verbeek et al. [16] presented a deterministic greedy method to learn the Gaussian mixture model. They also establish the mixture component-wise by starting with one component. Subsequently, new components are added iteratively and the EM is applied until convergence is reached. Ueda et al. [15] proposed a split-and-merge EM algorithm to alleviate the problem of local convergence of the EM method. Xu [18], [19] proposes a general statistical learning framework called the Bayesian Ying-Yang system which can be used among many other things for model selection and unsupervised learning of finite mixture models. In [8], Martinez and Vitrià combined GA and the EM algorithm for finding the optimal parameter estimates. This approach is applied to a robot navigation task in [9]. This algorithm is restricted to a predefined number of components. However, in many practical applications, the optimal number of components is unknown and has to be determined.

In this paper, we propose an algorithm for finding the optimal number of components as well as the parameters determining the components of the mixture model. The MDL criterion is used for selecting the number of components of the model. Our approach embeds the EM algorithm in the framework of the GA so that the properties of both algorithms are utilized. The population-based stochastic search of the GA explores the search space more thoroughly than the EM method. Therefore, our algorithm enables escaping from local optimal solutions since the algorithm becomes less sensitive to its initialization. Our algorithm is the generalization of the method proposed in [8]. It also enables the selection of the number of components using the MDL principle. The algorithm is presented for Gaussian mixture models.

The paper is organized as follows: In Section 2, the Gaussian mixture model and the EM algorithm are reviewed. In Section 3, the minimum description length criterion is introduced. The genetic-based EM algorithm, the genetic operators, and the encoding is presented in Section 4. We report experimental results on synthetic and real data in Section 5. Finally, conclusions and future work are presented in Section 6.

2 Learning Gaussian Mixture Models

A finite mixture model $p(\mathbf{x}|\boldsymbol{\Theta})$ is the weighted sum of $M > 1$ components $p(\mathbf{x}|\theta_m)$ in $\mathbb{R}^d$,

$$p(\mathbf{x}|\boldsymbol{\Theta}) = \sum_{m=1}^{M} \alpha_m p(\mathbf{x}|\boldsymbol{\theta}_m),$$

where $\mathbf{x} = [x_1, \ldots, x_d]^T$ is the d-dimensional data vector, α_m corresponds to the weight of each component $m = 1, \ldots, M$. These weights are constrained to be positive $\alpha_m \geq 0$ and $\sum_{m=1}^{M} \alpha_m = 1$. For Gaussian mixture models, each component $p(\mathbf{x}|\boldsymbol{\theta}_m)$ is represented as normal distribution which is denoted by the parameters $\boldsymbol{\theta}_m = \{\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}$, the mean vector and the covariance matrix. The Gaussian mixture is specified by the set of parameters $\boldsymbol{\Theta} = \{\alpha_1, \alpha_2, \ldots, \alpha_M, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_M\}$.

The EM algorithm [5] consists of an *expectation* step (E-step) and an *maximization* step (M-step) which are alternately used until the $\log p(\mathcal{X}|\boldsymbol{\Theta}) = \log \prod_{i=1}^{N} p(\mathbf{x}^i|\boldsymbol{\Theta})$ converges to a local optimum, where $\mathcal{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$ are N i.i.d. samples. The performance of the EM algorithm depends strongly on the choice of the initial parameters $\boldsymbol{\Theta}^{t=0}$. Different initialization strategies are given in [10].

E-step: The data $\mathcal{X}$ are assumed to be incomplete and the complete data set $\mathcal{Y} = (\mathcal{X}, \mathcal{Z})$ is determined by estimating the set of variables $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M\}$, where each $\mathbf{z}_m$ is an N-dimensional vector $[z_m^1, z_m^2, \ldots, z_m^N]^T$. The log likelihood of the complete data $\mathcal{Y}$ is

$$\log p(\mathcal{Y}|\boldsymbol{\Theta}) = \sum_{i=1}^{N} \sum_{m=1}^{M} z_m^i \log[\alpha_m p(\mathbf{x}^i|\boldsymbol{\theta}_m)],$$

where

$$z_m^i = P(m|\mathbf{x}^i, \boldsymbol{\Theta}^t) = \frac{\alpha_m^t p(\mathbf{x}^i|\boldsymbol{\theta}_m^t)}{\sum_{l=1}^{M} \alpha_l^t p(\mathbf{x}^i|\boldsymbol{\theta}_l^t)}$$

is the posterior probability and $\boldsymbol{\Theta}^t$ is the parameter estimate obtained after t iterations.

- *F. Pernkopf is with the Department of Electrical Engineering, University of Washington, M254 EE/CSE Building, Box 352500, Seattle, WA 98195-2500 and the Laboratory of Signal Processing and Speech Communication Graz University of Technology, Inffeldgasse 12, 8010 Graz, Austria. E-mail: fpernkop@ee.washington.edu, pernkopf@tugraz.at.*
- *D. Bouchaffra is with the Department of Computer Science and Engineering, Oakland University, 131 Dodge Hall, Rochester, MI 48309. E-mail: bouchaff@oakland.edu.*

Manuscript received 3 Nov. 2003; revised 27 Dec. 2004; accepted 4 Jan. 2005; published online 13 June 2005.
Recommended for acceptance by J. Goutsias.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0352-1103.

M-step: In this step, the parameters $\boldsymbol{\Theta}^{t+1}$ are determined according to the estimate of the variables z_m^i. For Gaussian mixture models this corresponds to reestimating the α_m^{t+1}, the $\boldsymbol{\mu}_m^{t+1}$, and $\boldsymbol{\Sigma}_m^{t+1}$ for each m according to

$$\alpha_m^{t+1} = \frac{1}{N}\sum_{i=1}^{N} z_m^i, \qquad \boldsymbol{\mu}_m^{t+1} = \frac{\sum_{i=1}^{N} z_m^i \mathbf{x}_i}{\sum_{i=1}^{N} z_m^i},$$
$$\text{and} \quad \boldsymbol{\Sigma}_m^{t+1} = \frac{\sum_{i=1}^{N} z_m^i \left(\mathbf{x}_i - \boldsymbol{\mu}_m^{t+1}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_m^{t+1}\right)^T}{\sum_{i=1}^{N} z_m^i}.$$

3 Model Selection Criterion: MDL

A comprehensive overview of model selection approaches is given in [10]. The MDL criterion,

$$MDL = -\log p(\mathcal{X}|\boldsymbol{\Theta}) + \frac{M(L+1)}{2}, \log N$$

is the most commonly used selection criterion, where L is the number of parameters defining each component (for Gaussian mixture models $L = d + d(d+1)/2$). This equation has the intuitive interpretation that the log likelihood $-\log p(\mathcal{X}|\boldsymbol{\Theta})$ is the code length of the *encoded* data. The term $\frac{M(L+1)}{2}\log N$ models the optimal code length for all parameters $\boldsymbol{\Theta}$.

4 Genetic-Based EM Algorithm (GA-EM)

The main goal of interweaving GA [1], [12] with the EM algorithm is to utilize the properties of both algorithms. In our GA-EM algorithm, each individual in the population represents a possible solution of the Gaussian mixture model. The MDL criterion (see Section 3) is used as fitness function for model selection. The best individual is the one that has the *lowest* MDL value. The evaluation of the individuals in the population is two-fold. First, R cycles of the EM algorithm are performed on each individual which results in an update of the set of parameters $\boldsymbol{\Theta}^t$ (at iteration t) and consequently of the individual which encodes these parameters. In cases where the relative log likelihood (see (1)) drops below a threshold ϵ, we terminate the EM and, consequently, do not perform all R cycles. This might be the case for a large value of R. Second, the MDL value is determined for each updated individual to judge the model. Hence, the evaluation process of the individual provides both, a fitness value and an update of the parameters encoded by the individual.

The convergence properties of the EM algorithm for Gaussian mixtures towards a local optimum are well-studied [10], [19]. To maintain the monotonic convergence property, we extended our GA-EM so that it is elitist which means that the best individual of the current generation is copied unaltered to the next generation. Thus, the mixing weights α_m of the best individual have to be saved for the subsequent generation. This mechanism guarantees that the best member of the population at generation $t+1$ does not perform worse than the best individual at generation t. The evolution process of the GA-EM is terminated when the number of components used by the best model does not change within five consecutive generations. Once the evolution is stopped, the EM algorithm is used to improve the best individual $\mathbf{a}_{min}$ found so far until the relative log likelihood of the mixture model

$$\left|\frac{\log p\left(\mathcal{X}|\boldsymbol{\Theta}^t\right) - \log p\left(\mathcal{X}|\boldsymbol{\Theta}^{t+1}\right)}{\log p\left(\mathcal{X}|\boldsymbol{\Theta}^t\right)}\right| \tag{1}$$

drops below a certain threshold ϵ (e.g., $\epsilon = 0.00001$).

In the following, the framework of the GA-EM algorithm is presented:

procedure GA-EM
begin
  $t \longleftarrow 0$
  $OldSize \longleftarrow 0$
  $c_{end} \longleftarrow 0$
  Initialize $P(t)$
  while $(c_{end} \neq 5)$
    $P(t)' \longleftarrow$ perform R EM steps on $(P(t))$
    $MDL' \longleftarrow$ evaluate $(P(t)')$
    $P(t)'' \longleftarrow$ recombine $(P(t)')$
    $P(t)''' \longleftarrow$ perform R EM steps on $(P(t)'')$
    $MDL'' \longleftarrow$ evaluate $(P(t)''')$
    $[P(t)'''', MDL] \longleftarrow$ select $[(P(t)''', MDL'') \cup (P(t)', MDL')]$
    $MDL_{min} \longleftarrow \min(MDL)$
    $\mathbf{a}_{min} \longleftarrow \arg\min_{MDL}\ (P(t)'''')$
    if $(|\mathbf{a}_{min}| \neq OldSize)$ **then**
      $c_{end} \longleftarrow 0$
      $OldSize = |\mathbf{a}_{min}|$
    else
      $c_{end} \longleftarrow c_{end} + 1$
    end
    $P(t)''''' \longleftarrow$ enforce mutation $(P(t)'''')$
    $P(t+1) \longleftarrow$ mutate $(P(t)''''')$
    $t \longleftarrow t+1$
  end
  EM $(\mathbf{a}_{min})$ until convergence of the log likelihood is reached
end

The best evaluation value achieved during the evolution process is stored in MDL_{min} and the corresponding individual in $\mathbf{a}_{min}$, where $|\mathbf{a}_{min}|$ denotes the number of components used for this model. $P(t)$ denotes a population of K individuals at generation t and $P(t)'$ the resulting population after performing the EM steps. $P(t)''$ is an offspring population of $P(t)'$ with size H. Performing the EM steps and evaluation of the offspring population delivers $P(t)'''$ and MDL''. In the following, the parameters and operators of the GA-EM are discussed in more detail.

Encoding: Each individual is composed of two parts. The first part (Part A) uses binary encoding, where the length of this part is determined by the maximal number of allowed components M_{max}. Each of these bits is related to a particular component. If a bit is set to zero, then its associated component is omitted for modeling the mixture, while setting the bit to one includes the component. The second part (Part B) uses floating point value encoding to encode the mean $\boldsymbol{\mu}_m$ and covariance $\boldsymbol{\Sigma}_m$ parameters of M_{max} components. Each component uses $L = d + d(d+1)/2$ parameters. Due to the switching mechanism of the components among the individuals during evolution of the GA, the components weight α_m cannot be encoded. Except for the best individual, these weights are assumed to be uniformly distributed.

Recombination: The crossover operator selects two parent individuals randomly from the population $P(t)'$ and recombines them to form two offsprings. The crossover probability p_c determines the number of offspring individuals H ($H = p_c K$). We use the *single-point crossover* [1], [12] which chooses randomly a crossover position $\chi \in \{1, \ldots, M_{max}\}$ within Part A of the individual and exchanges the value of the genes to the right of this position between both individuals for Part A with its associated parameters in Part B.

Selection: For selection, the (K, H)-strategy [2] is used. This approach refers to both the parent population $P(t)'$ and the offspring population $P(t)'''$ containing K and H individuals, respectively. After both populations have been evaluated, the K best individuals are selected to form the population $P(t)''''$ for the next generation.

Enforced Mutation: If more components model the data points in a similar manner some of their parameters are forced to mutate.

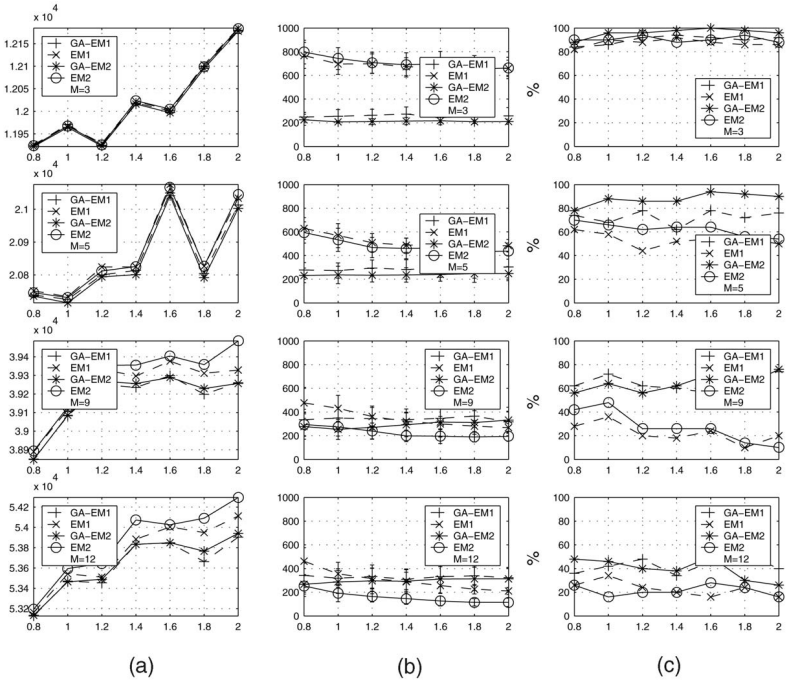

Fig. 1. Comparison of EM and GA-EM over 50 independent simulations for different settings of M, c, and $d = 5$. EM1 and GA-EM1 use random starting values and EM2 and GA-EM2 use the k-means algorithm for initialization. (a) Average achieved MDL. (b) Average number of required EM steps. (c) Percentage of correctly identified number of components.

This similarity is measured using the correlation coefficient r_{jk} which is computed pairwise between the components j and k ($1 \leq j, k \leq M, j > k$) from the posterior probability $\mathbf{z}_j$ and $\mathbf{z}_k$. If the correlation coefficient is above the threshold $t_{Correlation} < |r_{jk}|$, one of both components is randomly selected and added to the candidate set for mutation. Once the candidate set for enforced mutation is complete, a binary value is sampled from an uniform distribution for each candidate. According to this value, either the candidate component is removed by resetting the corresponding bit in Part A of the individual or a randomly chosen data point is assigned as mean value for the candidate component.

Mutation: The mutation operator inverts the binary value of each gene in Part A of the individuals with the mutation probability p_m. For Part B of the individual, an uniform distributed random number sampled within an upper and lower bound is assigned to genes that are mutated. These bounds were determined from the data set. The mutation rate for value encoding is scaled down by a factor of L, i.e., $\frac{p_m}{L}$. The mutation for the value encoded part of the individual is restricted to the mean values. Since our GA-EM is elitist, there are no mutations performed on the best individual.

5 Experiments: Comparison between EM and GA-EM

We have used two initialization methods throughout the experiments:

1. A variant with random starting values: The covariance matrix is initialized in a similar manner as in [7]. The mean values of the components $\boldsymbol{\mu}_m^{t=0}$ are set to randomly selected data points. The weights $\alpha_m^{t=0}$ of the components are assumed to be uniformly distributed, i.e., $\alpha_m^{t=0} = \frac{1}{M}$ $\forall m = 1, \ldots, M$.
2. k-means clustering [6]: The parameters of the selected components are initialized by the k-means algorithm. All unselected components are initialized to random starting values.

For the GA-EM, the start population $P(0)$ is composed of a set of individuals, where each individual has a different number of selected components. Hence, $P(0)$ consists of $\max\{M_{max}, K\}$ individuals. The number of individuals in subsequent populations is restricted to K.

If a component m is not supported by the data, the component is annihilated. This is the case when the sum of the posterior probability z_m^i over all data points is below a threshold $\sum_{i=1}^{N} z_m^i < t_{Annihilate}$. A reasonable threshold depends on the dimension d of the data.

5.1 Synthetic Data

Data sets with a dimension of $d \in \{2, 5, 10\}$ have been generated, whereby the sample size N varies with the number of components M according to $N = 300M$. The weight of each component α_m is selected randomly, whereby it is guaranteed that $\alpha_m > \frac{1}{2M}, \forall m = 1, \ldots, M$. The data were drawn from a mixture of Gaussian distribution with a different number of components $M \in \{3, 5, 9, 12\}$. Additionally, the minimum separation between the components were determined to be $c \in \{0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$. Dasgupta [4] defines that two Gaussians are c-separated if $\| \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \|_2 \geq c\sqrt{d \max(\lambda_{max}(\boldsymbol{\Sigma}_1), \lambda_{max}(\boldsymbol{\Sigma}_2))}$, where $\lambda_{max}(\boldsymbol{\Sigma})$ denotes the largest eigenvalue of $\boldsymbol{\Sigma}$. A mixture of components is considered to be c-separated if the components are pairwise c-separated. We generated 50 data sets for each configuration of M, c, and d. The maximum number of Gaussian components in the data is assumed to be $M_{max} = 15$ for the EM and the GA-EM algorithm. The parameter setting for the GA-EM is $p_m = 0.02$ for the mutation probability, $p_c = 0.8$ for the recombination probability, $K = 6$ for the population size, $R = 3$ for the number of EM steps within one GA iteration, and $t_{Correlate} = 0.95$ for the component correlation threshold. According to Section 5.3 (see Fig. 2), this parameter setting gives a good tradeoff between performance and running time of the algorithm on simulated data. The EM algorithm is executed for 2 to M_{max} components. The selected model is the one that achieves the lowest MDL value within the set of obtained candidate models. It is assumed that the proper number of components lies in the given range of $[2..M_{max}]$. The

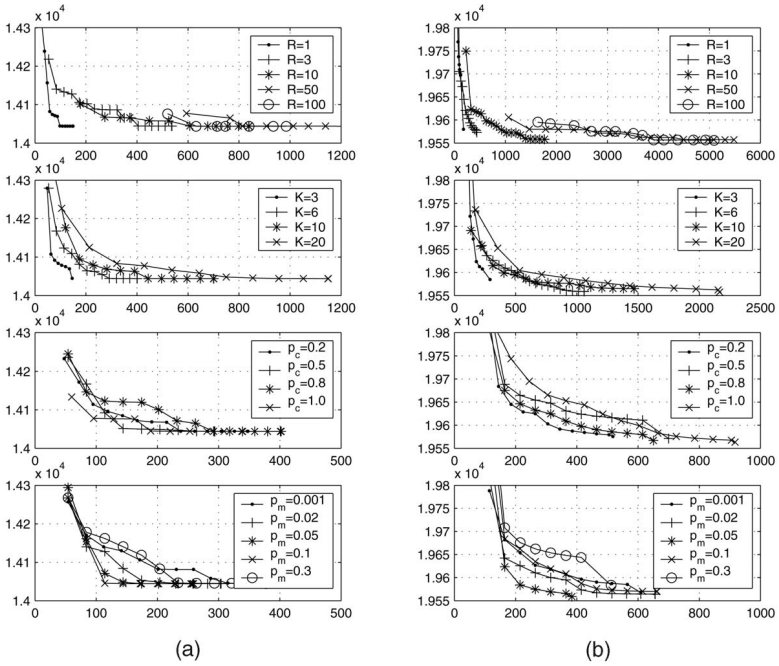

Fig. 2. Influence of the parameters of the GA-EM on the convergence behavior. (a) Simulated data ($M = 7$, $c = 1.2$, $d = 2$, and 2,100 samples). (b) PCA transformed Pendigit data (Digit 0,1,2,3, and 4).

termination condition of both algorithms is reached when the relative log likelihood drops below $\epsilon = 0.00001$.

In Fig. 1, both algorithms are compared with respect to the achieved average MDL criterion (see Fig. 1a), the average number of EM steps used to establish the model (see Fig. 1b), and the percentage of the correctly identified number of components (see Fig. 1c) which were used to generate the data set. The x-axis represents the value of c-separation. The rows of the figure correspond to the different number of components used to generate the data. GA-EM1 and EM1 use random starting values and GA-EM2 and EM2 are initialized using the k-means algorithm. Fig. 1 shows the results only for the dimension $d = 5$. The performance for $d = 2$ and $d = 10$ is similar. Further experiments on simulated and real data are reported in [13]. In the following, the observations are discussed.

Average MDL score (see Fig. 1a): Since both algorithms use the same termination condition, the obtained MDL score for selecting the finite mixture model is similar. However, especially, for larger numbers of components M, the GA-EM algorithm yields a better score. This fact is accredited to the dependency of the EM to its initialization. The population-based stochastic search behavior of the GA-EM explores the search space more thoroughly. This enables to escape from local optimal solutions since the algorithm becomes less sensitive to its initialization.

Average number of required EM steps (see Fig. 1b): The GA-EM converges faster than the EM algorithm for a small number of components M. The EM converges faster by increasing the separation of the components, whereby, the GA-EM is almost independent of this change. For $M = 9$, both algorithms require approximately the same number of EM steps when initialized with random starting values. The use of the k-means initialization speeds up the convergence of the EM algorithm, especially for a large number of components in the underlying data. Note that, for the GA-EM algorithm, the computational costs required for the genetic operators such as recombination, mutation, and selection are neglected.

Correctly identified number of components (see Fig. 1c): The GA-EM is more often identifying the correct number of components which were used for sampling the data. For a small number of components, $M = 3$, both algorithms work well. However, for an increasing number of components, the GA-EM is able to identify the correct number of producing components more often. This discrepancy is again attributed to the strong dependency of the EM on the initialization. The GA framework looses this tight relation due to its population-based search and the use of genetic operators.

5.2 Real Data: Pendigit Data

This data set [11] contains handwritten digit images of the 10 digits. From this data set, we have used 5,629 samples of the first five digits (i.e., 0, 1, 2, 3, and 4). The class distribution is roughly uniform. Each sample is described by 16 features. We have reduced the dimensionality of the data to 2 using principal component analysis (PCA). We have performed with each algorithm 10 runs using both initialization strategies. The maximum number of clusters is set to $M_{max} = 20$, $p_m = 0.05$, and $R = 7$. All the remaining parameters are the same as in the first experiment. Table 1 compares the EM and the GA-EM. We present the average number of required EM steps (EM-Steps), the average and the best value of the achieved MDL criterion abbreviated as AvMDL and MinMDL, respectively. We show the average selected number of components (AvComp) and the number of components selected for the best model (BestComp). Additionally, we determine the percentage of samples (%) that are correctly clustered using the best model. The best achieved result is emphasized by boldface letters. Similar to the discussion of the first experiment, the GA-EM slightly outperforms the EM algorithm.

5.3 Parameter Setting of the GA-EM Algorithm

The GA-EM algorithm requires additional parameters for the crossover and mutation probability, the population size K, the number of EM cycles R performed on each individual, and the threshold for enforced mutation. These parameters influence the running time of the algorithm. We have used the same setting of the parameters throughout the experiments. In the following, we have studied (see Fig. 2) the convergence behavior of the GA-EM algorithm using synthetic ($M = 7$, $c = 1.2$, $d = 2$, and 2,100 samples) and the PCA-transformed Pendigit data (Digit 0, 1, 2, 3, and 4). We have used random starting values to initialize the

TABLE 1
Clustering the Pendigit Data Set Using the EM and the GA-EM Algorithm

	EM	GA-EM	EM	GA-EM
	Random Initialization		k-means Initialization	
MinMDL	19583	**19556**	19560	19559
AvMDL	19611 ± 21.9	19580 ± 19.4	19583 ± 8.8	$\mathbf{19577 \pm 14.8}$
BestComp	13	14	13	13
AvComp	14.2	13.5	12.2	12.3
EM-Steps	1459.1 ± 121.1	851.2 ± 264.5	966.4 ± 71.0	$\mathbf{604.6 \pm 108.8}$
%	88.73	**89.39**	89.28	89.32

GA-EM algorithm. All the parameters of the GA-EM algorithm were selected as in Experiments 5.1 and 5.2. We have modified the number of EM steps R performed on each individual of the GA-EM, the population size K, the mutation probability p_m, and the recombination probability p_c which directly determines the number of offspring individuals. We have performed 10 independent runs and present the best result with the lowest MDL score. The x-axis represents the number of EM steps required before the termination condition of the GA-EM is met. The markers of the graphs give the best MDL score of the current generation t of the GA-EM. For simulated data (see Fig. 2a) the GA-EM converges for each parameter setting to almost the same MDL value. It is not surprising that, in the case of real data, the best MDL score is obtained for large populations (K). In general, a large value of R or K leads to a long execution time but also to a good solution. When we choose R = 100, the EM algorithm converges for each individual in each generation of the GA-EM. A high mutation rate is equivalent to a random walk and leads to inferior performance of the GA-EM. We have selected p_m and p_c according to commonly used values [12].

6 Summary and Future Work

This paper proposes a genetic-based EM algorithm for learning Gaussian mixture models from multivariate data. This algorithm is capable of selecting the number of components based on the MDL criterion. Our approach is less sensitive to the initialization compared to the standard EM algorithm. This is attributed to population-based search behavior of the GA-EM which explores the parameter space more thoroughly. Since the GA-EM is elitist it maintains the monotonic convergence property of the EM algorithm.

The experiments demonstrated that our algorithm outperforms the EM algorithm. In fact, we have obtained a better MDL score while using exactly the same initialization and termination condition for both algorithms. Additionally, the number of components which were used to generate the underlying data were more often correctly identified compared to the EM algorithm. However, one drawback of the GA-EM algorithm is that it requires additional parameters.

In the future, we aim to address the following: 1) Investigation of the behavior of the GA-EM algorithm by using different model selection criteria. 2) Extension of the GA-EM algorithm in order to make it more robust against outliers.

References

[1] T. Bäck, *Evolutionary Algorithms in Theory and Practice.* Oxford Univ. Press, 1996.
[2] T. Bäck and H. Schwefel, "Evolutionary Computation: An Overview," *Proc. IEEE Conf. Evolutionary Computation,* pp. 20-29, 1996.
[3] G. Celeux, S. Chrétien, F. Forbes, and A. Mkhadri, "A Component-Wise EM Algorithm for Mixtures," Technical Report 3746, INRIA, France, 1999.
[4] S. Dasgupta, "Learning Mixtures of Gaussian," *Proc. IEEE Symp. Foundations of Computer Science,* pp. 634-644, 1999.
[5] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood Estimation from Incomplete Data via the EM Algorithm," *J. Royal Statistic Soc.*, vol. 30, no. B, pp. 1-38, 1977.
[6] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification.* John Wiley & Sons, 2000.
[7] M.A.T. Figueiredo and A.K. Jain, "Unsupervised Learning of Finite Mixture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 3, pp. 1-16. Mar. 2002.
[8] A.M. Martinez and J. Vitrià, "Learning Mixture Models Using a Genetic Version of the EM Algorithm," *Pattern Recognition Letters,* vol. 21 pp. 759-769, 2000.
[9] A.M. Martinez and J. Vitrià, "Clustering in Image Space for Place Recognition and Visual Annotations for Human-Roboter Interaction," *IEEE Trans. Systems, Man, and Cybernetics B,* vol. 31, no. 5, pp. 669-682, 2001.
[10] G. McLachlan and D. Peel, *Finite Mixture Models.* John Wiley & Sons, 2000.
[11] C. Merz, P. Murphy, and D. Aha, "UCI Repository of Machine Learning Databases," Univ. California, Irvine, www.ics.uci.edu/mlearn/MLRepository.html, 1997.
[12] Z. Michalewicz and D.B. Fogel, *How to Solve It: Modern Heuristics.* Springer Verlag, 2000.
[13] F. Pernkopf, "Genetic-Based EM Algorithm for Component Selection and Parameter Estimation of Gaussian Mixture Models," technical report, Graz Univ. of Technology, 2004.
[14] D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions.* John Wiley & Sons, 1985.
[15] N. Ueda, R. Nakano, Z. Ghahramani, and G.E. Hinton, "SMEM Algoritm for Mixture Models," *Neural Computation,* vol. 12, no. 9, pp. 2109-2128, 2000.
[16] J.J. Verbeek, N. Vlassis, and B.J.A. Kröse, "Efficient Greedy Learning of Gaussian Mixture Models," *Neural Computation,* vol. 15, no. 2, pp. 469-485, 2003.
[17] L. Xu and M.I. Jordan, "On Convergence Properties of the EM Algorithm for Gaussian Mixtures," *Neural Computation,* vol. 8, pp. 129-151, 1996.
[18] L. Xu, "Bayesian Ying-Yang Machine, Clustering and Number of Clusters," *Pattern Recognition Letters,* vol. 18, pp. 1167-1178, 1997.
[19] L. Xu, "BYY Harmony Learning, Structural RPCL, and Topological Self-Organizing on Mixture Models," *Neural Networks,* vol. 15, pp. 1125-1151, 2002.