EM Algorithm with Genetic Evolution

David S. Hippocampus*
Department of Computer Science
Cranberry-Lemon University
Pittsburgh, PA 15213
hippo@cs.cranberry-lemon.edu

Coauthor
Affiliation
Address
email

Coauthor
Affiliation
Address
email

Coauthor
Affiliation
Address
email

Coauthor
Affiliation
Address
email
(if needed)

Abstract

The abstract paragraph should be indented 1/2 inch (3 picas) on both left and right-hand margins. Use 10 point type, with a vertical spacing of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Introduction

1.1 Motivation

1.2 Genetic algorithm

2 Method: genetic-based EM algorithm (GA-EM)

2.1 Model selection criterion: MDL

2.2 Encoding

Each individual in the population is composed of three parts, as shown in Figure 1. The first part (Part A) uses binary encoding, where the total number of bits is determined by the maximal number of allowed components $M_{\max}$. Each bit represents one particular Gaussian component in the model. If the bit is set to 0, then its associated component is omitted for modeling the mixture. In contrast, if the bit is set to 1, then it means this component is responsible for some data points in the mixture. The second part (Part B) uses floating point value encoding to record the components weight $\mathbf{w}$ of which the length is $M_{\max}$. Note that due to the switching mechanism of the components among individuals during evolution, the weight might need to be reset to uniform distribution except for the best individual (elitist). And the principle is to keep the weight as long as possible. The third part (Part C) also uses floating point value encoding to record the mean $\boldsymbol{\mu_m}$ and covariance $\boldsymbol{\Sigma_m}$ (and maybe other parameters) of the $M_{\max}$ components.

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

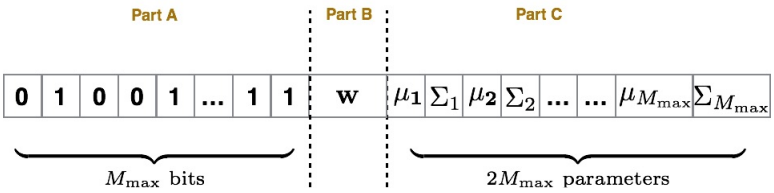


Figure 1: Encoding of individuals.

2.3 Recombination

The crossover operator selects two parent individuals randomly from the current population $\mathbf{P}$ and recombines them to form two offsprings. Total number of H ($H < K$ and is a multiple of 2) children will be generated in this step. We use *single-point crossover* (**TOCITE**), which chooses randomly a cross-over position $\chi = \{1, \ldots, M_{\max}\}$ within Part A of the individual and exchanges the value of the genes to the right of this position between two selected parent individuals, as shown in Figure 2. Part B of the offsprings are set to uniform distribution and Part C are exchanged correspondingly.

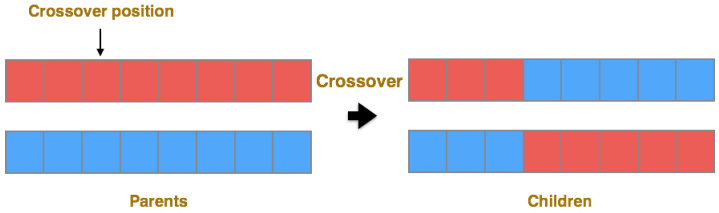


Figure 2: Recombination: single-point crossover.

2.4 Selection

2.5 Enforced mutation

2.6 Mutation

2.7 Implementation

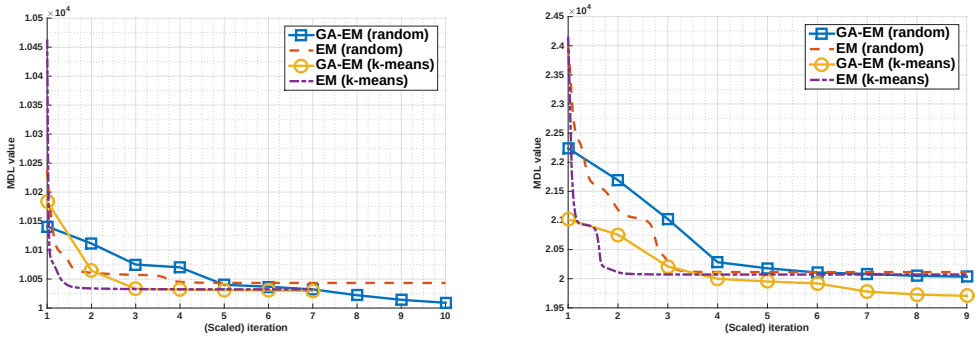
3 Results and Discussion

3.1 Initialization

The start population $\mathbf{P^0}$ is composed of a set of K individuals, where each individual has randomly selected M components. We explored two initialization methods:

1. *random*: The mean values of each component $\boldsymbol{\mu}_{\boldsymbol{m}}^{\mathbf{0}}$ are set to randomly selected data points. The covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{m}}^{\mathbf{0}}$ are initialized as the sample covariance matrix of all data. The weights $\mathbf{w}_{\mathbf{m}}^{\mathbf{0}}$ of the components are assumed to be uniformly distributed.
2. *k-means*: The parameters of the selected components are initialized by the k-means algorithm with $k = M$. The mean values of each component $\boldsymbol{\mu}_{\boldsymbol{m}}^{\mathbf{0}}$ are set to the cluster centroids. The covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{m}}^{\mathbf{0}}$ are initialized to the sample covariance matrices of the data in the corresponding cluster. The weights $\mathbf{w}_{\mathbf{m}}^{\mathbf{0}}$ of the components are the responsibilities of each cluster.

We have explored the influence of these two initialization methods, as shown in Figure 3. We added EM runs with the right number of component using two initialization for comparison. Note that the iteration numbers of the EM runs were scaled to accommodate the GA-EM runs. We observe that there are no significant performance difference on the two initialization.



(a) (b)

Figure 3: Influence of initialization on GA-EM and EM algorithms. (a) Simulated data. (b) PCA transformed Pendigit data.

3.2 Model selection

There are several parameters in the GA-EM algorithm to be determined to guarantee the performance. We have run the algorithm on both simulated data (Figure 4) and real PCA transformed Pendigit data (Figure 5).

- R: number of EM steps executed in the population. It stops impacting the convergence of the algorithm after R goes beyond 10. We set $R = 10$.
- K: population size. It in general shows negligible effect on the behavior. We set $K = 6$.
- H: number of offsprings generated. Here we used $K = 20$. It shows no significant effect on the behavior. We set $H = 4$.
- $M_{\max}$: maximal number of allowed components in each individual. It shows analogy behavior for $M_{\max} \geq 15$. We set $M_{\max} = 20$.
- p_m: mutation probability. It shows negligible effect on the behavior. We set $p_m = 0.1$.
- t_{corr}: correlation threshold. It shows negligible effect on the behavior except for very low correlation values. We set $t_{corr} = 0.95$.

3.3 Application on real data

3.4 Performance

3.5 Extension of GA-EM

4 Summary and future work

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

References follow the acknowledgments. Use unnumbered third level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to 'small' (9-point) when listing the references. **Remember that this year you can use a ninth page as long as it contains *only* cited references.**

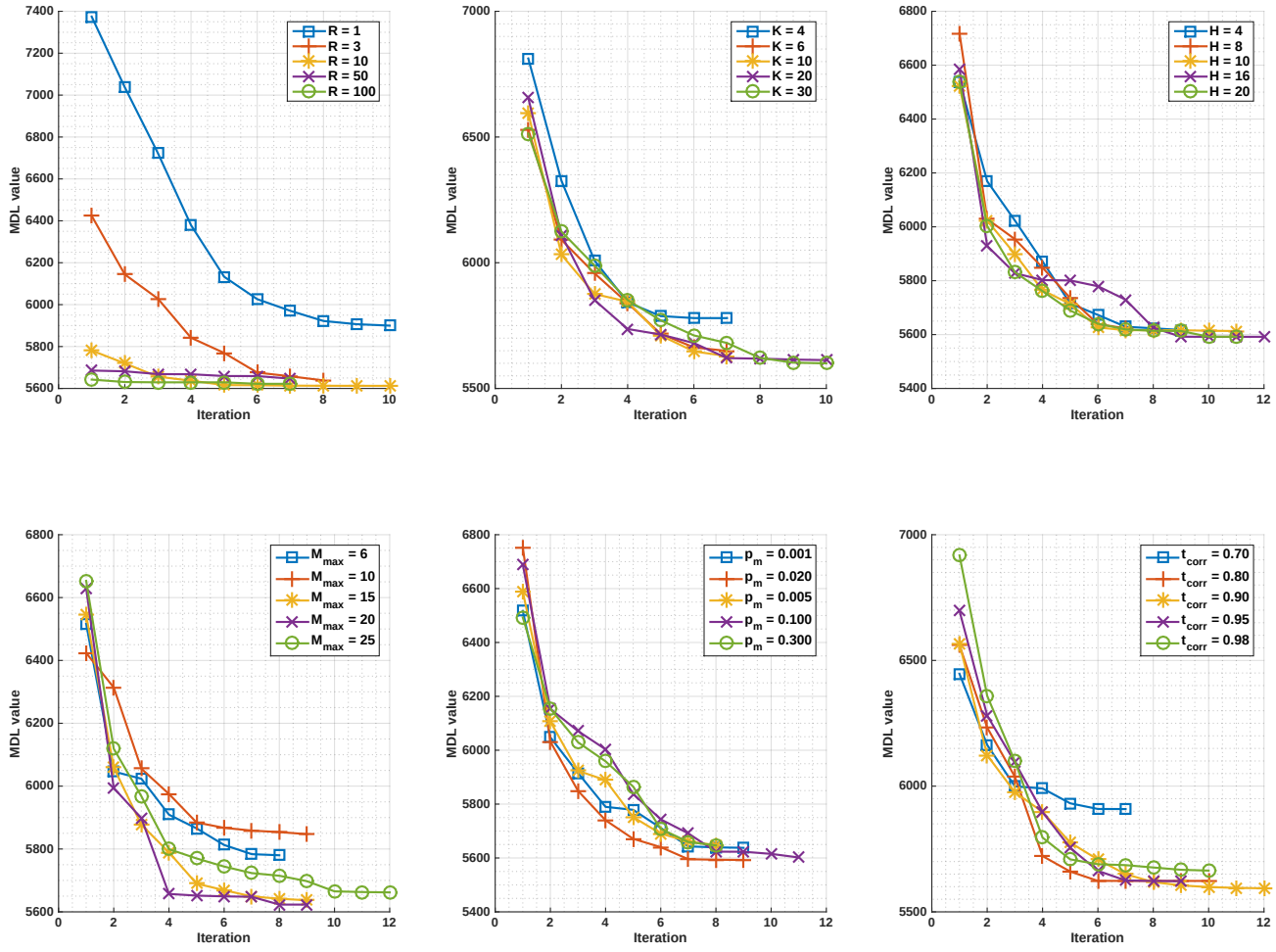


Figure 4: Influence of parameters on GA-EM algorithm with simulated data.

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D. S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609-616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer-Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

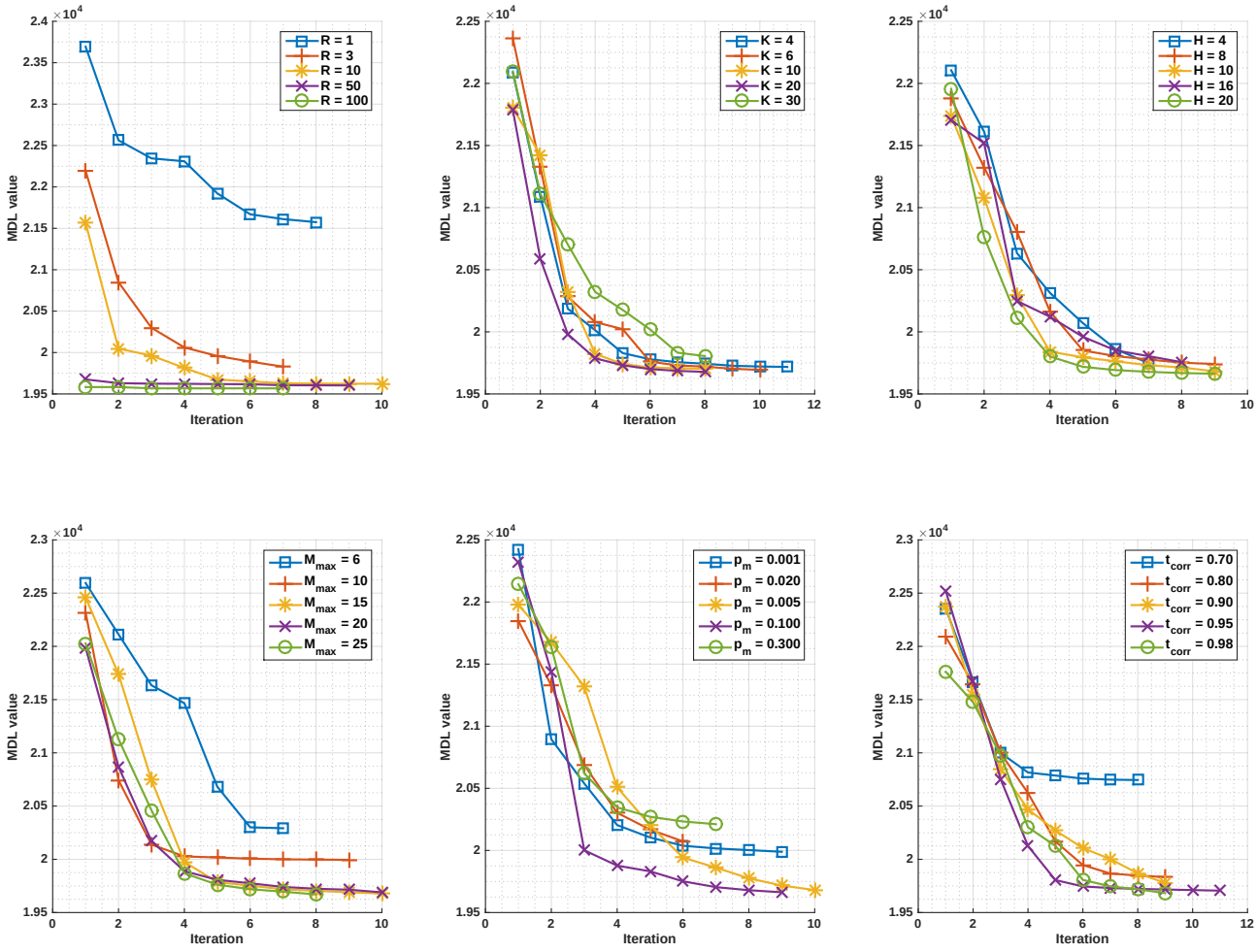


Figure 5: Influence of parameters on GA-EM algorithm with PCA transformed Pendigit data.