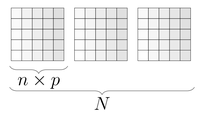
An Evolutionary Algorithm for Matrix-Variate Model-Based Clustering

Tommy Flynn, Paul D. McNicholas

Department of Mathematics & Statistics, McMaster University, Hamilton, Ontario, Canada

Introduction

- **Setting**: Clustering three-way data represented by N $n \times p$ random matrices, such as multivariate longitudinal data or greyscale images.



- **Model**: G-component finite mixtures of matrix-variate normals proposed by Viroli (2011).

$$f(\mathbf{X}|\boldsymbol{\vartheta}) = \sum_{g=1}^{G} \pi_g f_g(\mathbf{X}|\mathbf{M}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g)$$

$$f_g(\mathbf{X}|\mathbf{M}_g, \boldsymbol{\Sigma}_g, \boldsymbol{\Psi}_g) = \frac{1}{(2\pi)^{\frac{np}{2}}|\boldsymbol{\Sigma}_g|^{\frac{p}{2}}|\boldsymbol{\Psi}_g|^{\frac{n}{2}}} \exp\left\{-\frac{1}{2}\text{tr}\left(\boldsymbol{\Sigma}_g^{-1}(\mathbf{X}-\mathbf{M}_g)\boldsymbol{\Psi}_g^{-1}(\mathbf{X}-\mathbf{M}_g)'\right)\right\}$$

where f_g is the component matrix-variate normal density with location matrix $\mathbf{M}_g$ and scale matrices $\boldsymbol{\Sigma}_g$, $\boldsymbol{\Psi}_g$ and $\pi_g > 0$ is the mixing proportion with $\sum_{g=1}^{G} \pi_g = 1$.

- **Problem**: Parameters are typically estimated using the expectation-maximization (EM) algorithm which is frequently trapped at local maxima leading to suboptimal clusterings.
- **Solution**: McNicholas et al. (2021) proposed an evolutionary algorithm (EA) with crossover and mutation for an alternative search of the likelihood surface in the multivariate setting. We extend this EA to the matrix-variate setting.

Evolutionary Algorithms

- **Main Idea**: EAs maximize a fitness function by evolving a population of candidate solutions. Each generation, the population reproduces and the fittest individuals survive. This process continues until the population stagnates for a desired number of generations.

Fitness

- The population consists of matrices $\tilde{\mathbf{Z}} = (\tilde{z}_{ig})$, which estimate the latent cluster membership labels of the data. In these matrices, $\tilde{z}_{ig} = 1$ indicates that observation i belongs to cluster g, while $\tilde{z}_{ig} = 0$ means otherwise. To compute the fitness of an individual, we update the model parameters and then calculate the observed log-likelihood.

$$\text{Fitness Function } \left\{ \ell(\tilde{\boldsymbol{\vartheta}}) = \sum_{i=1}^{N} \log\left\{\sum_{g=1}^{G} \tilde{\pi}_g f_g(\mathbf{X}_i|\tilde{\mathbf{M}}_g, \tilde{\boldsymbol{\Sigma}}_g, \tilde{\boldsymbol{\Psi}}_g)\right\}\right.$$

$$\text{Parameter Updates } \begin{cases} \tilde{\pi}_g = \frac{N_g}{N}, \quad N_g = \sum_{i=1}^{N} \tilde{z}_{ig} \\ \tilde{\mathbf{M}}_g = \frac{\sum_{i=1}^{N} \tilde{z}_{ig}\mathbf{X}_i}{N_g} \\ \tilde{\boldsymbol{\Sigma}}_g = \frac{\sum_{i=1}^{N} \tilde{z}_{ig}(\mathbf{X}_i-\tilde{\mathbf{M}}_g)\tilde{\boldsymbol{\Psi}}_g^{-1}(\mathbf{X}_i-\tilde{\mathbf{M}}_g)'}{pN_g} \\ \tilde{\boldsymbol{\Psi}}_g = \frac{\sum_{i=1}^{N} \tilde{z}_{ig}(\mathbf{X}_i-\tilde{\mathbf{M}}_g)\tilde{\boldsymbol{\Sigma}}_g^{-1}(\mathbf{X}_i-\tilde{\mathbf{M}}_g)'}{nN_g} \end{cases}$$

Crossover

- Our EA reproduces by cloning the population many times and performing crossover on the clones. Crossover randomly swaps distinct rows of $\tilde{\mathbf{Z}}$ so that each clone has two different observation labels than its parent. The fittest individuals among the clones and parents become the next generation of parents.

$$\tilde{\mathbf{Z}}_{\text{Parent}} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1G} \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \end{bmatrix} \implies \tilde{\mathbf{Z}}_{\text{Clone}} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1G} \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Mutation

- Next we perform a greedy mutation on the surviving parents by randomly swapping distinct elements in random rows of $\tilde{\mathbf{Z}}$ until their fitness improves. The population has stagnated if we select the same parents after crossover and mutation.

$$\tilde{\mathbf{Z}}_{\text{Parent}} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1G} \\ z_{21} & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ z_{N1} & z_{N2} & \cdots & z_{NG} \end{bmatrix} \implies \tilde{\mathbf{Z}}_{\text{Mutate}} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1G} \\ z_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ z_{N1} & z_{N2} & \cdots & z_{NG} \end{bmatrix}$$

Algorithm 1 EA for matrix-variate model-based clustering.

Inputs:
$\mathbf{X} \leftarrow$ list of observation matrices
$\mathbf{Z} \leftarrow$ list of membership matrices
$G \leftarrow$ number of clusters
`pars` $\leftarrow$ number of parents
`clones` $\leftarrow$ number of clones
`stagnation` $\leftarrow$ max number of stagnations

$N \leftarrow$ length of $\mathbf{X}$
`stag` $\leftarrow 0$
while `stag` $<$ `stagnation` **do**
  for $a = 1$ to `pars` **do**
    for $b = 1$ to `clones` **do**
      Crossover: randomly swap two distinct labels from parent a to get clone b
      Fitness: update parameters and calculate log-likelihood of clone b
    end for
  end for
  Surivival: sort parents and clones by descending fitness and take top `pars` as new parents
  for $a = 1$ to `pars` **do**
    for r in random permutation of 1 to N **do**
      Mutate: swap two distinct elements in row r
      if fitness increases **then**
        break for
      else
        swap back
      end if
    end for
  end for
  if parents are identical to last generation **then**
    `stag` $\leftarrow$ `stag` $+ 1$
  else
    `stag` $\leftarrow 0$
  end if
end while
Return Top parent

Results

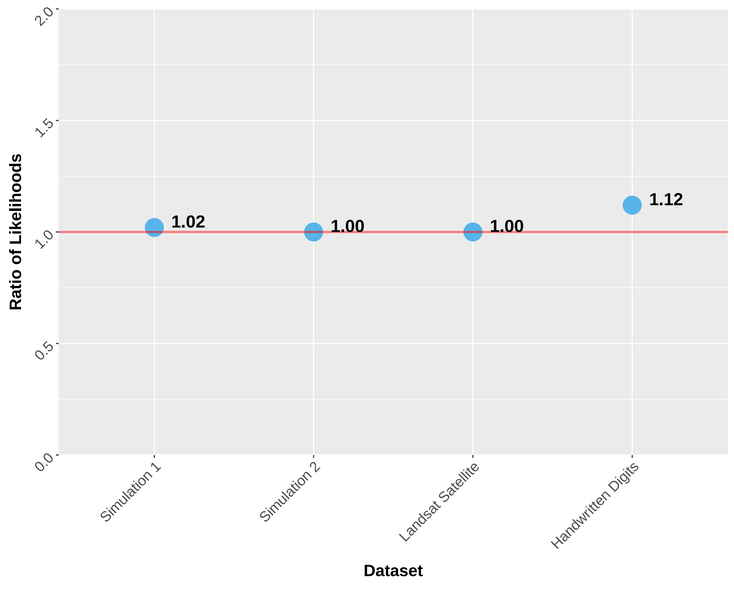


Figure 1. Ratio of the EA observed likelihood to the EM observed likelihood for each dataset.

- EA and EM were applied to two simulated datasets and two real world datasets described in Table 1.
- Clustering performance was measured by the likelihood ratio in Figure 1 and adjusted rand index (ARI) in Figure 2.
- EA was run with 2 parents, 10 clones, 4 stagnation, and a K-means start.

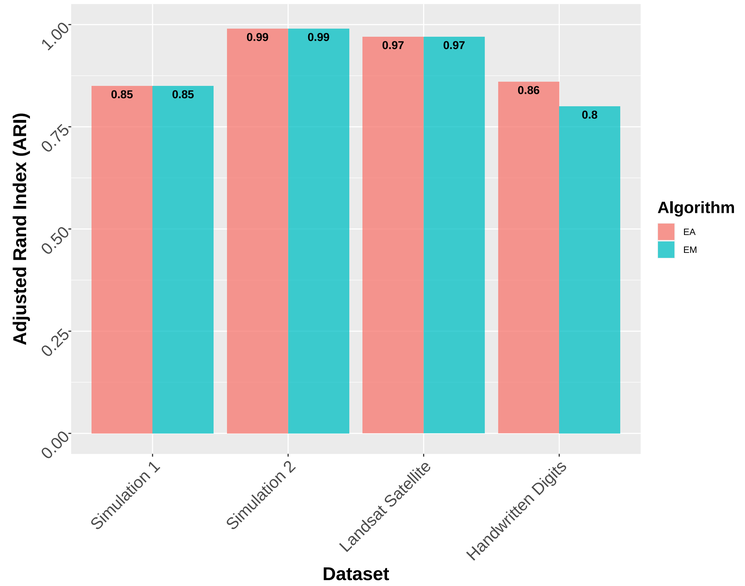


Figure 2. ARI for EA and EM on each dataset.

Dataset	(N)	$(n \times p)$	(G)	Description
Simulation 1	300	2 ×4	2	Simulated data from similar location matrices and randomly generated scale matrices.
Simulation 2	450	5 ×3	3	Simulated data from unbalanced classes with $\boldsymbol{\pi} = (0.6, 0.3, 0.1)$.
Landsat Satellite	600	4 ×9	2	3 ×3 satellite images of the same region in 4 light spectrums from the UCI machine learning repository. Classes: red soil and cotton crop.
Handwritten Digits	400	16 ×16	4	Normalized greyscale images of handwritten digits from the Elements of Statistical Learning. Classes: 1,6,7,8.

Table 1. Description of the datasets.

Results

- EA and EM displayed similar clustering performances as seen by the likelihood ratio and ARI in Figures 1 and 2.
- Both EA and EM selected the appropriate number of clusters using the Bayesian information criterion (BIC).
- EA was superior at clustering the large greyscale images of handwritten digits.

Conclusion

- The EA developed is a competitive alternative to the EM in terms of clustering performance as measured by the likelihood and the ARI.

References

1. C. Viroli, Finite mixtures of matrix normal distributions for classifying three-way data, Stat. Comput. 21 (4) (2011) 511–522.

2. S.M. McNicholas, P.D. McNicholas, D.A. Ashlock, An Evolutionary Algorithm with Crossover and Mutation for Model-Based Clustering, J. Classif. 38 (2021) 264–279.