Efficient initialization of the EM algorithm for Gaussian mixture models

STATS 606 - Final report

Tim White

Due April 14th, 2023

Abstract

The expectation-maximization (EM) algorithm is a convenient method of maximum likelihood estimation for Gaussian mixture models, but it is not guaranteed to converge to the global maximum of the (log-)likelihood function for Gaussian mixtures of an arbitrary number of components. As such, the success of the algorithm often hinges on the initialization of the parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$. Many different initialization strategies have been proposed in the literature — some are based on random initialization, others involve data-driven methods like singular value decomposition or K-means clustering, and a handful are highly sophisticated procedures that require considerable computational effort. Perhaps unsurprisingly, no single initialization scheme has been shown to consistently outperform the others across all Gaussian mixture model settings. In this report, we study a relatively general model setting with a moderate number of mixture components where the parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ are treated as unknown. We consider four computationally feasible initialization methods, and we use simulation to assess the impact of these methods on the performance of the EM algorithm. In our simulation studies, we directly quantify the accuracy of the initial parameter values and characterize the post-initialization convergence behavior of the EM algorithm, and in doing so we build on earlier studies that focused primarily on the accuracy of the final EM estimates. Our results demonstrate that data-driven initialization strategies like svdEM (which uses singular value decomposition) and kmEM (which uses K-means clustering) tend to be more accurate and efficient than random initialization schemes, and in many settings they provide a reasonable alternative to more sophisticated but computationally intensive techniques. However, all of these initialization strategies tend to struggle when there is substantial overlap between the mixture components, especially when the dimension of the data is high. The optimal strategy for initializing $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ in this setting remains an open question.

Contents

1 Introduction

1.1 Overview

The expectation-maximization (EM) algorithm [DLR77] is a special type of minorization-maximization algorithm that is commonly used to compute maximum likelihood estimates for the parameters of latent variable models. It is an iterative procedure that first uses the parameter estimates from the previous iteration to construct a lower bound for the (log-)likelihood of the observed data (E step), and then updates the estimates by maximizing this lower bound with respect to the parameters (M step).

For many model classes, the EM algorithm is analytically intractable because the conditional expectation in the E step has no closed form. In these instances, one can use Monte Carlo methods or other numerical techniques to approximate this quantity before proceeding with the M step [LC01]. For some model classes, however, the lower bound in the E step *does* permit a closed-form update, and hence the EM algorithm is a particularly suitable approach to parameter estimation for these classes.

Multivariate Gaussian mixture models are one such "nice" family of models for which the EM algorithm can be applied in a straightforward manner. Borrowing notation from [MM12] and [Mur22], we consider n independent observed variables $X_1, ..., X_n \in \mathbb{R}^p$ and corresponding latent variables $Z_1, ..., Z_n$ from the generative model

$$\begin{aligned} Z_i &\sim \text{Categorical}_K(\pi_1, ..., \pi_K), \quad \pi_1, ..., \pi_K \geq 0, \ \sum_{k=1}^{K} \pi_k = 1 \\ X_i | Z_i = k &\sim N_p(\mu_k, \Sigma_k) \end{aligned} \tag{1}$$

for all $i \in \{1, ..., n\}$, where K is the number of mixture components, π_k is the kth mixing proportion, and μ_k and $\Sigma_k \in \mathbf{S}^p_{++}$ are the mean and covariance of the kth component density.

Perhaps a more natural understanding of the above model emerges when we recognize that the observations $X_1, ..., X_n$ are iid draws from the density

$$g(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \pi_k \, \phi_p(\mathbf{x}; \mu_k, \Sigma_k), \tag{2}$$

where ϕ_p is the p-variate Gaussian density. Thus, we observe that the density of each X_i is a weighted average of the K component densities — i.e., a mixture of Gaussians.

Maximum likelihood estimates for $\boldsymbol{\pi} = (\pi_1, ..., \pi_K)$, $\boldsymbol{\mu} = (\mu_1, ..., \mu_K)$, and $\boldsymbol{\Sigma} = (\Sigma_1, ..., \Sigma_K)$ can be computed via the EM algorithm as described in Section 2.1. Once these estimates are obtained, it is common to use the fitted Gaussian mixture model for clustering by assigning each observation to the component with the highest estimated posterior probability [Mur22]. We will examine Gaussian mixture models through this lens in Sections 2 and 3.

While the EM algorithm provides a convenient method of maximum likelihood estimation for Gaussian mixture models, it is not foolproof — as we will discuss in Section 1.2, there exists no general guarantee that it will converge to the global maximum of the log-likelihood for Gaussian mixtures of an arbitrary number of components. As such, the success of the EM algorithm often hinges on whether the parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ are initialized in a sufficiently close neighborhood around their respective optima. The initialization of these parameters is the central topic of this report. In the next section, we will briefly discuss several prevalent strategies for initializing $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ before narrowing our focus to four such methods for the remainder of the paper.

1.2 Related work

We first examine the convergence behavior of the EM algorithm for multivariate Gaussian mixture models. This is a relatively well-studied topic, as promising results have been obtained regarding the global convergence of the algorithm for mixtures of two components, as well as its local convergence for mixtures of two or more components.

In particular, for mixtures of two equally weighted Gaussians with known and equal covariance matrices, Daskalakis et al. [DTZ17] describe conditions under which the EM algorithm converges globally to the maximum likelihood estimates. Wu and Zhou [WZ22] improve on this result by demonstrating a similar

guarantee with fewer assumptions. For mixtures of two or more Gaussians, Zhao et al. [ZLS20] show that the EM algorithm converges locally as long as the components are well-separated. Zhao et al. focus on the setting in which $\boldsymbol{\pi}$ and $\boldsymbol{\Sigma}$ are known, but their simulation results suggest that their findings extend to the case where these quantities are unknown.

Despite this recent progress, there still does not exist a more general characterization of the convergence of the EM algorithm for Gaussian mixtures of an arbitrary number of components. Such a guarantee is not expected to exist — Melnykov and Melnykov [MM12] point out that the EM algorithm strictly climbs the likelihood of a Gaussian mixture model, and this likelihood function is generally not unimodal. More concretely, Wu [Wu83] proves that the EM algorithm will converge to a stationary point or a local maximum in certain settings. Hence, all current results regarding the algorithm's convergence behavior — including those described in the previous paragraph — rely on certain assumptions, typically about mixing proportions, covariance matrices, or the amount of separation between the components.

Due to this lack of a general global convergence guarantee, one must exhibit great care when initializing $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ in most settings. Some studies of the EM algorithm for Gaussian mixture models treat the mixing proportions or the covariance structure as known, which simplifies the task of initialization. However, in this report, we consider the more general case in which $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ are all unknown. Perhaps the most naive initialization scheme in this setting is to use equal mixing proportions $\pi_1 = \cdots = \pi_K = \frac{1}{K}$ for $\boldsymbol{\pi}$, a random sample of K points $\tilde{\mathbf{x}}_1, ..., \tilde{\mathbf{x}}_K$ for $\boldsymbol{\mu}$, and K identical diagonal matrices (e.g., identity matrices) for $\boldsymbol{\Sigma}$. We refer to this naive strategy as randomEM — see Section 2.2.1.

Melnykov and Melnykov [MM12] provide a comprehensive overview of five more sophisticated initialization strategies for the general case outlined in the previous paragraph, and they also propose their own approach. The following is a brief description of each of these procedures:

- emEM [BCG03] and RndEM [Mai09] stochastically initialize the mixture model parameters by conducting one or more short runs of the EM algorithm from different starting points and comparing the likelihoods attained in each short run. emEM considers many candidate initializations but uses fewer iterations in the short EM phase, while RndEM considers one candidate but allows more short EM iterations.
- kmEM [Mai09] uses K-means clustering to initialize the model parameters. Starting from a random selection of K means from the observed data, this procedure minimizes the within-cluster sum of squares by iteratively assigning points to their closest cluster and recomputing the means.
- hierEM [SM58] initializes the parameters using hierarchical clustering with average linkages.
- rmEM [Mai09] initializes the parameters through a procedure that identifies well-separated local modes in the observed data and selects representative observations from these modes.
- Σ-EM [MM12] uses an iterative algorithm involving truncated multivariate Gaussian distributions and singular value decomposition to compute initial estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. This algorithm is embedded in another iterative procedure that estimates the number of components K through a process of cluster detection and elimination.

Melnykov and Melnykov [MM12] evaluate these initialization strategies via simulation using several different metrics, including (i) Bayesian Information Criterion of the selected model, (ii) detected number of mixture components, and (iii) adjusted Rand index [HA85]. They find that the relative performance of the six strategies varies across different model settings, although Σ-EM seems to be best equipped to handle departures from the ideal settings discussed above — i.e., it performs well when the true Gaussian mixture model has many heterogeneous mixture components with various amounts of separation between them. Σ-EM also automatically selects the number of components K; this is more efficient than trying and evaluating multiple values of K, as is required by the other five methods.

However, while Σ-EM provides a sophisticated and accurate method of covariance matrix estimation, it is a complex procedure with potentially substantial computational costs. As such, it is of interest to identify the model settings for which a less sophisticated but less computationally intensive initialization strategy can yield sufficient accuracy. Also, while Melnykov and Melnykov evaluate the accuracy of the *final* parameter estimates attained by the EM algorithm for each initialization strategy in each simulation setting, they do not directly quantify the closeness of each strategy's initial guess to the true parameter values. Further, while Melnykov and Melnykov report the runtime of each strategy in each setting, they do not elaborate on the post-initialization efficiency of the EM algorithm. Specifically, it is of interest to know how many iterations are required to reach a particular convergence threshold after the parameters

are initialized with each strategy. We will fill the gaps highlighted above by conducting simulation studies of a similar flavor as those of [MM12].

1.3 Roadmap

The remainder of this report will proceed as follows. In Section 2.1, we briefly revisit the model in (1) and (2) and write out the M step updates for $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$. In Sections 2.2 through 2.4, we outline our simulation studies by providing more details about four initialization strategies (randomEM, emEM, svdEM, and kmEM) and describing the metrics we will use to evaluate these methods. We present the results of our simulations in Section 3, with a focus on both accuracy and efficiency. Finally, we conclude in Section 4 by contextualizing our findings and discussing several potential avenues for future work.

2 Methods

2.1 EM algorithm for Gaussian mixture models

Recall the model setting introduced in (1) and (2), and assume that $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ are all unknown. Following the details in [Bis06] and the notation of [MM12], it can be shown that the log-likelihood of the parameters given $\{X_i, Z_i\}_{i=1}^n$ — also known as the complete data log-likelihood — is

$$\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathbf{x}, \mathbf{z}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \big(\log \pi_k + \log \phi_p(\mathbf{x}_i; \mu_k, \Sigma_k) \big), \tag{3}$$

where z_{ik} is an indicator variable for the kth element of z_i.

In the E step of the EM algorithm, we take the expectation of (3) with respect to the conditional distribution of Z given X and obtain

$$E_{Z|X}\big[\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathbf{x}, \mathbf{z})\big] = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ik} \big(\log \pi_k + \log \phi_p(\mathbf{x}_i; \mu_k, \Sigma_k) \big), \tag{4}$$

where for all $i \in \{1, ..., n\}$ and $k \in \{1, ..., K\}$, the posterior probability that the ith observation belongs to the kth mixture component is

$$\gamma_{ik} = E_{Z|X}\big[z_{ik}\big] = \frac{\pi_k \phi_p(\mathbf{x}_i; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \phi_p(\mathbf{x}_i; \mu_j, \Sigma_j)}. \tag{5}$$

The expectation in (4) is amenable to straightforward maximization. Given values $\boldsymbol{\pi}^{(t)}$, $\boldsymbol{\mu}^{(t)}$, and $\boldsymbol{\Sigma}^{(t)}$ for iteration t, we obtain updated estimates $\boldsymbol{\pi}^{(t+1)}$, $\boldsymbol{\mu}^{(t+1)}$, and $\boldsymbol{\Sigma}^{(t+1)}$ in the M step of the EM algorithm by maximizing (4). For all $k \in \{1, ..., K\}$, we obtain

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ik}^{(t+1)}}{n}, \quad \mu_k^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ik}^{(t+1)} \mathbf{x}_i}{\sum_{i=1}^{n} \gamma_{ik}^{(t+1)}}, \quad \Sigma_k^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ik}^{(t+1)} (\mathbf{x}_i - \mu_k^{(t+1)})(\mathbf{x}_i - \mu_k^{(t+1)})^\top}{\sum_{i=1}^{n} \gamma_{ik}^{(t+1)}}, \tag{6}$$

where for all $i \in \{1, ..., n\}$ and $k \in \{1, ..., K\}$,

$$\gamma_{ik}^{(t+1)} = \frac{\pi_k^{(t)} \phi_p(\mathbf{x}_i; \mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_{j=1}^{K} \pi_j^{(t)} \phi_p(\mathbf{x}_i; \mu_j^{(t)}, \Sigma_j^{(t)})}. \tag{7}$$

We reiterate that our focus in this report is on the choice of the initial values $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\mu}^{(0)}$, and $\boldsymbol{\Sigma}^{(0)}$. Once the parameters are initialized using any of the methods described in the following section, the EM algorithm continues updating the estimates via (6) and (7) until it converges.

2.2 Initialization strategies

Consider the setting in which $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ are unknown. We will evaluate four different strategies for initializing these parameters — two that involve some degree of randomness (randomEM and emEM), and

two that are data-driven (svdEM and kmEM).

2.2.1 randomEM

randomEM is the naive approach proposed in Section 1.2. Recall that we initialize $\boldsymbol{\pi}$ by assigning equal mixing proportions $\pi_1 = \cdots = \pi_K = \frac{1}{K}$. To initialize $\boldsymbol{\mu}$, we draw a random sample of K observations $\tilde{\boldsymbol{x}}_1, ..., \tilde{\boldsymbol{x}}_K$ without replacement from $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ and set $\mu_k = \tilde{\boldsymbol{x}}_k$ for all $k \in \{1, ..., K\}$. To initialize $\boldsymbol{\Sigma}$, we set $\Sigma_k = I_{p \times p}$ for all $k \in \{1, ..., K\}$, where p is the dimension of the observed data $\mathbf{x} \in \mathbb{R}^{n \times p}$.

Note that randomEM is our baseline initialization strategy — we do not expect it to perform as accurately in most settings as the three methods described below. We also do not necessarily expect randomEM to be more efficient than the other strategies in obtaining the final estimates $\hat{\boldsymbol{\pi}}$, $\hat{\boldsymbol{\mu}}$, and $\hat{\boldsymbol{\Sigma}}$. The initialization procedure itself is essentially as fast as possible, but the algorithm might subsequently take longer to converge since a random initialization of the parameters is not guaranteed to be near the true optima.

2.2.2 emEM

emEM [BCG03] is an extension of randomEM. Recall from Section 1.2 that emEM involves an initial phase in which several short runs of the EM algorithm are conducted with random parameter initializations. In our simulations, we consider three random candidates in this short EM phase, and for each short run of the algorithm we set a loose convergence tolerance of (long EM tolerance)$^{1/4}$ and limit the maximum number of iterations to n.

Each short run of the algorithm yields a set of preliminary estimates $\{\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\}$. For each of these sets of estimates, we compute the log-likelihood of the observed data (the formula for which can be obtained by taking the logarithm of the product of (2) over $i \in \{1, ..., n\}$). We then identify the set of estimates that achieves the largest log-likelihood, and we use these estimates to initialize $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ in the final, longer run of the EM algorithm.

2.2.3 svdEM

The third initialization strategy that we will evaluate in our simulation studies is a singular value decomposition (SVD) technique proposed by Maitra [Mai01] that is implemented in the `R` package `EMCluster` [Che+22]. This strategy, which we denote as svdEM, uses SVD to decompose the observed data and obtain its singular values, chooses an initial set of component means based on these singular values, and then groups the observations around these means via a procedure similar to the K-means algorithm.

Hence, we expect svdEM to perform similarly to kmEM, which is described below. Both of these initialization procedures make a preliminary guess about the locations of the component means and then perform a grouping operation to cluster the observed data based on this guess. The computational trade-off faced by svdEM (as well as emEM and kmEM) is the opposite of that faced by randomEM — the initialization process itself is somewhat costly, but the EM algorithm may require fewer iterations to converge since the initial parameter values are likely to be closer to the true optima.

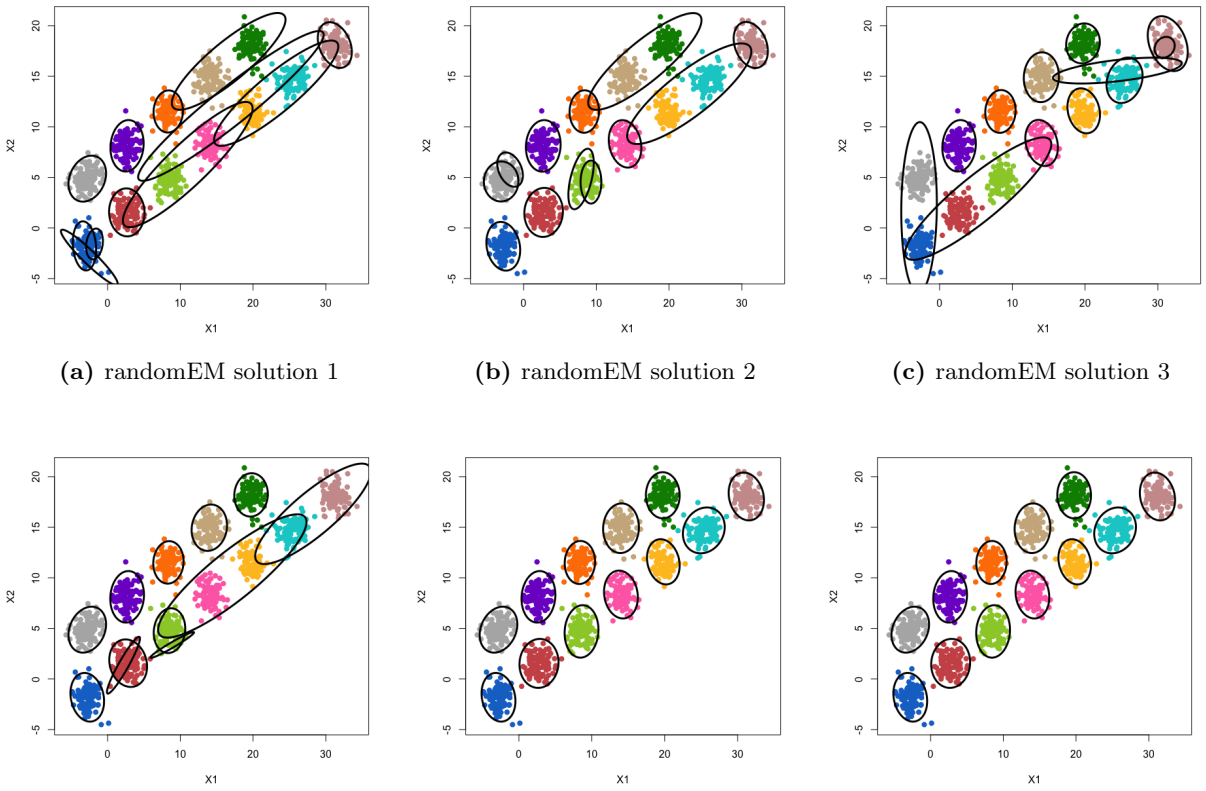
2.2.4 kmEM

Finally, we consider kmEM [Mai09], which initializes $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ via the K-means clustering algorithm described in Section 1.2. We first select a random sample of K observations $\tilde{\boldsymbol{x}}_1, ..., \tilde{\boldsymbol{x}}_K$ without replacement from $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ and use these observations as an initial guess for $\boldsymbol{\mu}$. We then assign each of the n observations to its closest component and update our estimates by computing the means of the reformulated components. This process continues iteratively until it converges, at which point we obtain estimates $\hat{\boldsymbol{\pi}}$, $\hat{\boldsymbol{\mu}}$, and $\hat{\boldsymbol{\Sigma}}$ that we subsequently use to initialize the EM algorithm. In practice, it is common to consider several initial guesses when running the K-means algorithm. Following the example of Melnykov and Melnykov [MM12], we allow n such restarts in our simulations.

2.3 Demonstration of initialization strategies

Before moving on to a more extensive simulation study, we first consider an example which illustrates how the estimates yielded by the EM algorithm can vary substantially for certain initialization procedures in more complex Gaussian mixture model settings. We generate $n = 1200$ observations of dimension $p = 2$ from a model with $K = 12$ components. We set the index of separation between the components at $s = 0.25$ — see Section 2.4 for details about this separation index. We conduct three runs of the EM

Figure 1: Solutions obtained by the four initialization methods ($n = 1200$, $K = 12$, $p = 2$, $s = 0.25$)



(a) randomEM solution 1 **(b)** randomEM solution 2 **(c)** randomEM solution 3

(d) emEM solution **(e)** svdEM solution **(f)** kmEM solution

algorithm on this data set using randomEM to initialize $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$. We then conduct three more runs in which we initialize the algorithm with emEM, svdEM, and kmEM, respectively. The results of these six runs are presented in Figure 1. We plot the true clusters and overlay the 95% confidence ellipsoids implied by the estimates $\{\hat{\mu}_k, \hat{\Sigma}_k\}_{k=1}^K$. Panels (a)-(c) display the solutions obtained by the randomEM runs, while panels (d)-(f) display the solutions obtained by emEM, svdEM, and kmEM.

The observed data log-likelihoods attained in panels (a)-(d) are -6599, -6499, -6512, and -6514, respectively. In panels (e) and (f), we observe that svdEM and kmEM initialization yield the same solution, which has a log-likelihood of -6308. Hence, the EM algorithm achieved a better solution when it was initialized with svdEM and kmEM than it did in any of the runs with randomEM or emEM. This difference in performance is also apparent visually — while the ellipsoids in panels (e) and (f) seem to perfectly capture the twelve true mixture components, panels (a)-(d) all contain several errors. This example illustrates the potential perils of using random initialization strategies like randomEM and emEM when the underlying mixture model has many components.

2.4 Outline of simulation studies

We now carry out a more comprehensive simulation study that aims to assess how accurately and efficiently the EM algorithm is able to estimate $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ after it is initialized with each of the four strategies introduced in Section 2.2. We consider a data-generating process in which n observations of dimension p are drawn from a Gaussian mixture model with K^* components, where the true $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ are determined by the `genRandomClust()` function in `R` and the true $\boldsymbol{\Sigma}$ is assumed to comprise K^* identity matrices. The components are separated by an index s corresponding to the `sepVal` argument of `genRandomClust()`, which ranges from -1 to 1 with higher values indicating a greater amount of separation [QJ20]. We consider three different values of s in our simulations — $s = 0.01$ corresponds to a model with substantial overlap between the components, $s = 0.15$ corresponds to a model with a small amount of overlap, and $s = 0.3$ corresponds to a model with separated components.

We also vary the true number of mixture components K^*, as well as the dimension p of the observed data. Specifically, we consider $K^* \in \{5, 10\}$ and $p \in \{2, 4\}$. For each combination of K^*, p, and s, we generate twenty independent data sets of n observations as described in the previous paragraph, where $n = 500$ for $K^* = 5$ and $n = 1000$ for $K^* = 10$. For each data set, we run the EM algorithm three times for each initialization strategy — once each for $K \in \{K^* - 1, K^*, K^* + 1\}$ — and select the K that attains the highest Bayesian Information Criterion (BIC), where $\text{BIC} = 2\,\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathbf{x}) - (Kp + K\frac{p(p+1)}{2} + K - 1)\log n$. Each run of the EM algorithm continues until the relative difference in observed data log-likelihood between the current and previous iterations dips below a convergence threshold of 0.0001 — i.e., until $[\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathbf{x})^{(t+1)} - \ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathbf{x})^{(t)}]/[\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathbf{x})^{(t+1)} - \ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathbf{x})^{(1)}] \leq 0.0001$.

For each of the twenty data sets, we rank the BIC values attained by the four initialization strategies from one to four, where one indicates the highest (i.e., best) BIC and four indicates the lowest. We also report the estimated number of mixture components $\hat{K}$ for each initialization strategy. Finally, for each strategy, we use the posterior probabilities estimated by the EM algorithm to predict the component assignments of the n observations, and we compute the adjusted Rand index between these predicted components and the true components. The Rand index is a value between zero and one that measures the similarity between two vectors of cluster assignments, and the adjusted Rand index is a modification of this quantity that accounts for a baseline level of randomness in the cluster assignments [HA85]. Larger values of the adjusted Rand index indicate greater similarity between the predicted and actual component assignments, and hence greater accuracy in the EM estimates. For example, the solutions in panels (a)-(f) have adjusted Rand indexes of 0.731, 0.780, 0.788, 0.800, 0.998, and 0.998, respectively.

We also consider three metrics that describe the efficiency of each initialization strategy. The first of these metrics is the absolute difference between the Frobenius norms of the initialized component means and the true component means — i.e., $\left| \|\boldsymbol{\mu}_{\text{init}}\|_F - \|\boldsymbol{\mu}^*\|_F \right| = \left| \sqrt{\text{Tr}(\boldsymbol{\mu}_{\text{init}}^T \boldsymbol{\mu}_{\text{init}})} - \sqrt{\text{Tr}(\boldsymbol{\mu}^{*T} \boldsymbol{\mu}^*)} \right|$. We claim that this is a reasonable measure of the distance between $\boldsymbol{\mu}_{\text{init}}$ and $\boldsymbol{\mu}^*$ — it reflects the accuracy of each strategy's initial guess for the component means. In cases where the subsequent run of the EM algorithm converges to the true optimum, this metric also reflects the distance the algorithm must "travel" to converge (and hence how many iterations and seconds are required). Contrary to intuition, $\|\boldsymbol{\mu}_{\text{init}} - \boldsymbol{\mu}^*\|_F$ is not an appropriate distance metric for these purposes since the rows of $\boldsymbol{\mu}_{\text{init}}$ may be a permutation of the rows of $\boldsymbol{\mu}^*$ even if the rows themselves are identical.

The second efficiency metric we consider is the number of iterations required for convergence after the parameters are initialized, and the third is the cumulative runtime (in seconds) of the initialization procedure and the EM algorithm. The utility of these quantities is straightforward — the former reflects the accuracy of the initial parameter estimates, while the latter carries information about the computational efficiency of each initialization strategy.

For each initialization strategy, we average each of the accuracy and efficiency metrics described above across the twenty data sets. Our results for the settings with $K^* = 5$ and $K^* = 10$ components are summarized in Table 1 and Table 2, respectively.

3 Results

We first investigate the performance of the four initialization strategies in the setting with $K^* = 5$ mixture components, as presented in Table 1. One immediate takeaway is that the EM algorithm generally performs better as the index of separation between the components increases, regardless of which method is used for initialization. Specifically, for all four strategies, the algorithm achieves a higher adjusted Rand index as s increases, and it tends to require fewer iterations and less time to converge. Similarly, the distance between $\boldsymbol{\mu}_{\text{init}}$ and $\boldsymbol{\mu}^*$ tends to decrease as s increases, although this trend is more apparent for svdEM and kmEM than it is for randomEM and emEM. For all four initialization strategies, we observe no substantial difference in accuracy or efficiency for the two different values of p.

Across the four initialization strategies, the EM algorithm does a relatively good job of detecting the true number of mixture components. For all four methods and nearly all combinations of p and s, we find that $\hat{K}$ is quite close to K^*. The exception to this result is the setting where $p = 4$ and $s = 0.01$ — here, the algorithm incorrectly detects four components for each strategy. Overall, the settings where $s = 0.01$ yield a much worse overall performance than settings with more separation between the components. This suggests that it is inherently difficult to estimate $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ in low-separation settings, even when relatively sophisticated initialization schemes like svdEM and kmEM are used.

Table 1: Performance of randomEM, emEM, svdEM, and kmEM for $K^* = 5$

		$p = 2$			$p = 4$		
		$s = 0.01$	$s = 0.15$	$s = 0.30$	$s = 0.01$	$s = 0.15$	$s = 0.30$
randomEM	*Adjusted Rand index*	0.766	0.930	0.946	0.562	0.895	0.941
	BIC rank	3.2	3.0	2.9	3.0	3.0	3.0
	$\hat{K}$	4.8	5.3	5.4	4.0	5.0	5.0
	$\lvert \lVert \boldsymbol{\mu}_{\text{init}} \rVert_F - \lVert \boldsymbol{\mu}^* \rVert_F \rvert$	1.206	1.325	2.282	1.249	0.873	1.594
	Number of iterations	57.4	38.0	25.4	42.6	35.6	16.8
	Runtime (seconds)	2.5	1.3	0.9	2.1	1.7	0.9
emEM	*Adjusted Rand index*	0.785	0.967	0.986	0.598	0.925	0.984
	BIC rank	3.7	3.9	3.2	2.7	3.7	3.4
	$\hat{K}$	4.8	5.2	5.4	4.0	5.0	5.3
	$\lvert \lVert \boldsymbol{\mu}_{\text{init}} \rVert_F - \lVert \boldsymbol{\mu}^* \rVert_F \rvert$	0.709	0.641	1.196	1.124	0.612	0.598
	Number of iterations	64.5	31.0	37.4	39.5	24.0	25.0
	Runtime (seconds)	2.9	1.5	1.5	2.5	1.9	1.7
svdEM	*Adjusted Rand index*	0.839	0.972	0.999	0.607	0.963	0.998
	BIC rank	1.8	2.1	2.4	2.8	2.1	2.2
	$\hat{K}$	5.0	5.0	5.0	4.0	5.0	5.0
	$\lvert \lVert \boldsymbol{\mu}_{\text{init}} \rVert_F - \lVert \boldsymbol{\mu}^* \rVert_F \rvert$	0.168	0.024	0.002	0.959	0.013	0.001
	Number of iterations	30.2	9.8	4.8	24.9	12.3	6.2
	Runtime (seconds)	1.1	0.4	0.2	0.9	0.6	0.4
kmEM	*Adjusted Rand index*	0.840	0.972	0.999	0.613	0.963	0.998
	BIC rank	1.2	1.1	1.5	1.6	1.1	1.4
	$\hat{K}$	5.0	5.0	5.0	4.0	5.0	5.0
	$\lvert \lVert \boldsymbol{\mu}_{\text{init}} \rVert_F - \lVert \boldsymbol{\mu}^* \rVert_F \rvert$	0.116	0.020	0.001	0.962	0.011	0.001
	Number of iterations	27.3	9.5	5.0	22.5	12.3	6.2
	Runtime (seconds)	1.3	0.5	0.3	1.2	0.8	0.5

By all metrics, we find that svdEM and kmEM yield the best performance out of the four initialization methods, with kmEM likely holding a slight edge due to its advantage in BIC. svdEM and kmEM achieve high accuracy in recovering the true Gaussian mixture model parameters, and they do so efficiently. The strong performance of these two methods is most apparent for $s \in \{0.15, 0.30\}$, as both strategies produce a $\boldsymbol{\mu}_{\text{init}}$ close to $\boldsymbol{\mu}^*$ and an adjusted Rand index close to one in just a few iterations. For all combinations of p and s, svdEM and kmEM yield a higher adjusted Rand index and require fewer iterations and less runtime than randomEM and emEM. This is not surprising given our discussion in Sections 2.2 and 2.3 about the potential shortcomings of random initialization.

Turning our attention to Table 2, we find that nearly all of the trends described above also hold for the model with $K^* = 10$ components. Specifically, we observe poor performance when $s = 0.01$, stronger performance as s increases, and no substantial difference in accuracy or efficiency between $p = 2$ and $p = 4$ for all four initialization methods. We again find that svdEM and kmEM outperform randomEM and emEM across the board. If anything, the gap in performance between the random initialization methods and the data-driven approaches is slightly larger in Table 2 than in Table 1 — see, for example, the relatively low adjusted Rand indexes attained by randomEM and emEM.

The overall accuracy and efficiency of the four initialization strategies are worse in the setting with ten components than the setting with five components. The random initialization methods appear to suffer more than svdEM and kmEM in terms of accuracy, but all four strategies experience sizable losses in efficiency. For instance, for nearly all combinations of p and s in Table 2, the runtime required by each strategy is more than double the runtime required for the corresponding setting in Table 1. These findings suggest that the initialization and subsequent estimation of $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ are more challenging in models with many mixture components, especially when the components are not well-separated.

4 Discussion

In this report, we have studied the task of initializing the Gaussian mixture model parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ for the purpose of maximum likelihood estimation via the EM algorithm. We considered the case where $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ are all unknown, and we used simulation to assess the impact of four initialization strategies on the accuracy and efficiency of the EM algorithm in a variety of model settings. Our results demonstrate a clear discrepancy in performance between random initialization strategies like randomEM and emEM and data-driven schemes like svdEM and kmEM. We find that svdEM and kmEM attain a higher adjusted Rand index and better BIC than randomEM and emEM for nearly all of the combinations of p and s in Table 1 and Table 2, and they accomplish this in fewer iterations and less runtime.

Table 2: Performance of randomEM, emEM, svdEM, and kmEM for $K^* = 10$

		$p = 2$			$p = 4$		
		$s = 0.01$	$s = 0.15$	$s = 0.30$	$s = 0.01$	$s = 0.15$	$s = 0.30$
randomEM	*Adjusted Rand index*	0.705	0.882	0.890	0.538	0.843	0.881
	BIC rank	3.9	3.6	3.5	3.9	3.5	3.4
	$\hat{K}$	9.7	10.2	10.2	9.0	9.7	10.2
	$\lvert \lVert \boldsymbol{\mu}_{\text{init}} \rVert_F - \lVert \boldsymbol{\mu}^* \rVert_F \rvert$	3.046	3.733	4.518	2.015	2.385	3.671
	Number of iterations	117.0	58.5	21.6	87.8	58.0	21.8
	Runtime (seconds)	9.3	4.7	1.7	9.0	5.7	1.9
emEM	*Adjusted Rand index*	0.725	0.945	0.928	0.599	0.878	0.894
	BIC rank	2.5	3.2	3.5	2.6	3.5	3.6
	$\hat{K}$	9.7	10.3	10.4	9.1	9.9	10.2
	$\lvert \lVert \boldsymbol{\mu}_{\text{init}} \rVert_F - \lVert \boldsymbol{\mu}^* \rVert_F \rvert$	1.836	2.057	3.404	1.386	1.351	2.346
	Number of iterations	118.0	48.0	23.4	104.0	37.1	34.1
	Runtime (seconds)	10.0	5.7	2.8	12.4	5.4	3.6
svdEM	*Adjusted Rand index*	0.771	0.956	0.998	0.644	0.945	0.999
	BIC rank	2.1	1.9	1.9	2.1	1.6	1.6
	$\hat{K}$	9.6	9.9	10.0	9.0	9.9	10.0
	$\lvert \lVert \boldsymbol{\mu}_{\text{init}} \rVert_F - \lVert \boldsymbol{\mu}^* \rVert_F \rvert$	0.710	0.279	0.028	0.869	0.101	0.015
	Number of iterations	106.6	23.4	6.0	81.2	13.1	6.7
	Runtime (seconds)	7.4	1.9	0.6	7.7	1.4	0.7
kmEM	*Adjusted Rand index*	0.742	0.970	0.998	0.672	0.941	0.999
	BIC rank	1.5	1.2	1.2	1.4	1.4	1.4
	$\hat{K}$	9.2	10.0	10.0	9.0	9.9	10.0
	$\lvert \lVert \boldsymbol{\mu}_{\text{init}} \rVert_F - \lVert \boldsymbol{\mu}^* \rVert_F \rvert$	1.054	0.044	0.006	0.800	0.136	0.002
	Number of iterations	64.3	10.9	6.3	64.6	16.0	6.8
	Runtime (seconds)	5.8	1.8	1.2	8.5	3.3	1.7

4.1 Contextualization of main findings

The overall success of svdEM and kmEM in our simulations suggests that basic data-driven initialization methods are capable of providing a reasonable and less computationally costly alternative to more complex procedures such as Melnykov and Melnykov's Σ-EM [MM12] in some Gaussian mixture model settings. In particular, we infer that methods like svdEM and kmEM have the best chance of achieving a level of accuracy comparable to Σ-EM when the mixture components are relatively well-separated. Similar to Melnykov and Melnykov, we find that no initialization method performs particularly well when there is considerable overlap among the components, particularly if the dimension of the data is relatively high. The optimal initialization strategy in this setting is unclear, but Σ-EM is likely a better option than svdEM or kmEM due to its careful estimation of Σ and K.

Another contribution that this report makes to Melnykov and Melnykov's previous work [MM12] (and other works reviewed therein, such as [BCG03], [Mai09], and [SM58]) is our more direct assessment of the quality of each strategy's initial estimates for $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$. As described in Section 2.4, we considered two metrics for this purpose: (i) $\lvert \lVert \boldsymbol{\mu}_{\text{init}} \rVert_F - \lVert \boldsymbol{\mu}^* \rVert_F \rvert$ and (ii) the number of iterations required for convergence. In our simulations, we find that these two quantities are generally consistent with the other metrics in Table 1 and Table 2. However, in other Gaussian mixture model settings, they could potentially provide a more comprehensive characterization of the post-initialization behavior of the EM algorithm.

4.2 Avenues for future research

While our simulations provide valuable insights about the performance of randomEM, emEM, svdEM, and kmEM in a relatively general Gaussian mixture model setting with $K^* \in \{5, 10\}$, $p \in \{2, 4\}$, $s \in \{0.01, 0.15, 0.30\}$, and unknown parameters $\{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, it is of interest to determine if these insights extend beyond this setting. Specifically, it would be informative to consider larger values of K^* and p, a wider range of values for s, and a non-diagonal true covariance structure. A focus on settings with small values of s (i.e., little separation between the mixture components) would be particularly beneficial considering that neither we nor Melnykov and Melnykov were able to identify an initialization method that performed well in low-separation settings.

Of course, the computational costs of repeatedly running the EM algorithm for larger Gaussian mixture models are likely to accumulate quickly. Even for the $\{K^* = 10, p = 4\}$ setting, I found that a nontrivial amount of time was required to run the simulations in Section 3 even when utilizing the parallel processing capabilities of my machine. As such, further investigation into the trade-off between computational feasibility and Gaussian mixture model size is another potentially fruitful direction of research.

References

[BCG03] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. "Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models". In: *Computational statistics & data analysis* 41.3–4 (2003), pp. 561–575. URL: http://dx.doi.org/10.1016/s0167-9473(02)00163-9.

[Bis06] Christopher Bishop. *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006. ISBN: 9780387310732.

[Che+22] Wei Chen Chen et al. *EM algorithm for model-based clustering of finite mixture Gaussian distribution*. 2022. URL: https://cran.r-project.org/web/packages/EMCluster/.

[DTZ17] Constantinos Daskalakis, Christos Tzamos, and Manolis Zampetakis. "Ten steps of EM suffice for mixtures of two Gaussians". In: *Proceedings of the 2017 Conference on Learning Theory* 65 (2017), pp. 704–710. URL: https://proceedings.mlr.press/v65/daskalakis17b.html.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society* 39.1 (1977), pp. 1–22. URL: http://dx.doi.org/10.1111/j.2517-6161.1977.tb01600.x.

[HA85] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of classification* 2.1 (1985), pp. 193–218. URL: http://dx.doi.org/10.1007/bf01908075.

[LC01] Richard A. Levine and George Casella. "Implementations of the Monte Carlo EM algorithm". In: *Journal of computational and graphical statistics* 10.3 (2001), pp. 422–439. URL: http://dx.doi.org/10.1198/106186001317115045.

[Mai01] Ranjan Maitra. "Clustering massive datasets with application in software metrics and tomography". In: *Technometrics* 43.3 (2001), pp. 336–346. URL: http://dx.doi.org/10.1198/004017001316975925.

[Mai09] Ranjan Maitra. "Initializing partition-optimization algorithms". In: *IEEE/ACM transactions on computational biology and bioinformatics* 6.1 (2009), pp. 144–157. URL: http://dx.doi.org/10.1109/TCBB.2007.70244.

[MM12] Volodymyr Melnykov and Igor Melnykov. "Initializing the EM algorithm in Gaussian mixture models with an unknown number of components". In: *Computational statistics & data analysis* 56.6 (2012), pp. 1381–1395. URL: http://dx.doi.org/10.1016/j.csda.2011.11.002.

[Mur22] Kevin Murphy. *Probabilistic Machine Learning: An Introduction*. MIT Press, 2022. URL: https://probml.github.io/pml-book/book1.html.

[QJ20] Weiliang Qiu and Harry Joe. *clusterGeneration: Random cluster generation (with specified degree of separation)*. 2020. URL: https://cran.r-project.org/web/packages/clusterGeneration/index.html.

[SM58] R. Sokal and C. Michener. "A statistical method for evaluating systematic relationships". In: *University of Kansas Science Bulletin* 38 (1958), pp. 1409–1438. URL: https://ia800703.us.archive.org/5/items/cbarchive_33927_astatisticalmethodforevaluatin1902/astatisticalmethodforevaluatin1902.pdf.

[Wu83] C. F. Jeff Wu. "On the convergence properties of the EM algorithm". In: *Annals of statistics* 11.1 (1983), pp. 95–103. URL: http://dx.doi.org/10.1214/aos/1176346060.

[WZ22] Yihong Wu and Harrison H. Zhou. "Randomly initialized EM algorithm for two-component Gaussian mixture achieves near optimality in $O(\sqrt{n})$ iterations". In: *Mathematical statistics and learning* 4 (2022), pp. 143–220. URL: http://dx.doi.org/10.4171/msl/29.

[ZLS20] Ruofei Zhao, Yuanzhi Li, and Yuekai Sun. "Statistical convergence of the EM algorithm on Gaussian mixture models". In: *Electronic journal of statistics* 14.1 (2020), pp. 632–660. URL: http://dx.doi.org/10.1214/19-ejs1660.