Scientific Reports

ARTICLE | Article in Press

Discriminant analysis-guided multi-split segmentation for expectation maximization initialization in time-series data

Ipeleng Labius Machele · Adeiza James Onumanyi · Adnan M. Abu-Mahfouz · Anish Kurien

Received: 4 May 2026 / Accepted: 2 September 2026
© The Author(s) 2026

Abstract

In this paper, a discriminant analysis–guided multi-split segmentation approach referred to as K-between class segmentation (K-BCS) is presented, which is developed as an effective initialization strategy for the expectation–maximization (EM) algorithm in time-series data. The proposed approach is aimed at improving parameter estimation and convergence stability in the EM algorithm. Although EM is widely used for mixture modeling, its performance is highly sensitive to its initialization estimates, which often results in slow convergence, convergence to suboptimal local maxima, and unstable likelihood behavior. To address these limitations, K-BCS-EM leverages a proposed between-class measure to identify intrinsic data structures and generate informed initial parameter estimates. Unlike random or centroid-based methods, the proposed approach is fully data-driven and captures separation in both location and dispersion, thereby providing more reliable starting points for the EM algorithm. The proposed approach was evaluated via extensive Monte Carlo experiments across both synthetic and real-world datasets. The results obtained show that the K-BCS-EM approach consistently outperformed conventional initialization strategies, such as random sampling, random partitioning, jittered sampling, and k-means-based methods particularly in terms of their likelihood convergence, mean and variance RMSE performance. The K-BCS-EM demonstrates strong robustness in challenging scenarios such as in heavy-tailed distributions and highly overlapping mixtures while maintaining scalability as the number of components increases. Although the proposed approach introduces a modest computational overhead due to its preprocessing stage, the significant improvements in estimation accuracy, convergence stability, and robustness highlight K-BCS-EM as an effective initialization method for EM-based mixture modeling in time series data.

Keywords Parameter initialization · Discriminant analysis · Multi-split segmentation · Expectation maximization · Gaussian mixture models

This Article in Press is shared early to give you faster access to new research. It is citable and carries a permanent DOI. The final edited version will replace it automatically.

Introduction

The expectation–maximization (EM) algorithm is a foundational methodological tool for maximum likelihood estimation in mixture models and latent-variable approach[1]. Its iterative refinement of latent assignments and model parameters has enabled widespread adoption across diverse domains, including pattern recognition[2], signal processing[3], anomaly detection[4], and IoT-based intelligent monitoring systems[5]. Owing to its conceptual elegance and computational efficiency, EM remains a preferred approach for modeling complex and heterogeneous data distributions. Nevertheless, the practical performance of EM is highly sensitive to its initialization parameters[6] as shown in Fig. 1. Essentially, inaccurate initial parameter values may lead to slow convergence, entrapment in suboptimal local maxima, unstable likelihood trajectories, and inaccurate parameter estimation, which undermines the reliability of many EM-based modeling pipelines.

The red curve in Fig. 1 highlights two key limitations of the EM algorithm under poor initialization: slow convergence and convergence to suboptimal local maxima. Poor initialization leads to a lower starting log-likelihood and slower progression toward stability, and also results in convergence to an inferior final solution. In contrast, good initialization enables faster convergence and attainment of higher likelihood values. Although EM guarantees monotonic improvement, it does not ensure global optimality which makes initialization critical for accurate and reliable parameter estimation.

To address this, conventional initialization strategies have been proposed such as the random sampling (RS)[7], random partitioning (RP)[8], jittered sampling (JS)[9], and one-dimensional K-means (KM)[10] based approaches. These methods offer simplicity and ease of implementation but lack robustness in complex data environments. They typically rely on stochastic or heuristic assumptions and do not explicitly account for the underlying structural or modal characteristics of the data. As a result, their performance degrades in the presence of multimodality, overlapping components, heterogeneous variances, or significant noise conditions commonly encountered in real-life applications such as in adaptive thresholding[11], behavioral modeling[12], biomedical analytics[13] and time series telemetry within IoT-based systems[14]. As modern datasets continue to grow in volume, dimensionality, and temporal complexity, the limitations of such initialization techniques increasingly constitute a critical bottleneck for EM-based inference.

In this paper, an alternative initialization strategy approach is explored that exploits the structural separability of the data prior to parameter estimation. In this context, a discriminant analysis (DA)–guided segmentation approach is proposed as a mechanism for identifying statistically meaningful partitions by maximizing separability between data regions. Unlike heuristic initializations, this approach operates directly on the observable data structure, which enables explicit detection of unimodal and multimodal structures without imposing restrictive generative assumptions.

Building upon this idea, a K-between class segmentation approach for the EM algorithm is proposed termed K-BCS-EM. This approach extends classical between-class variance (BCV) analysis to a multi-split segmentation strategy capable of identifying (K) structurally separable regions in time-series data. By formulating segmentation as a global optimization problem, K-BCS-EM yields deterministic and interpretable partitions that directly inform robust initialization of EM mixture parameters.

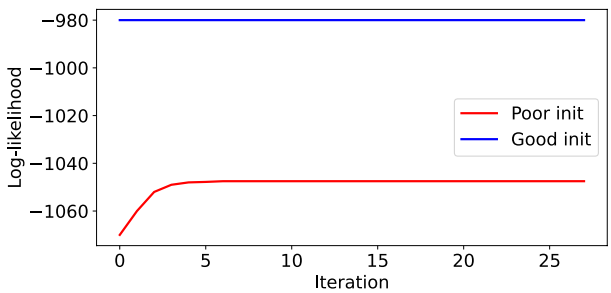


Fig. 1 Typical EM algorithm's log-likelihood response to poor vs good initialization estimates.

Despite the theoretical appeal of BCV-guided segmentation, its integration into EM initialization particularly in generalized (K)-modal and time series settings remains largely unexplored. Existing studies typically restrict BCV-based approaches to bimodal thresholding or image-processing applications and do not systematically develop them as EM initialization mechanisms in diverse data regimes. Consequently, practitioners lack clear guidance on whether BCV-guided segmentation can reliably improve EM convergence behavior, parameter estimation accuracy, and stability relative to established methods in unimodal, bimodal, and multimodal scenarios.

Consequently, to address these gaps, the following contributions are presented:

- A K-between class segmentation approach is proposed which leverages the newly developed between-class guided multi-split segmentation strategy to identify *K*-modal structures in time-series datasets.
- An EM initialization strategy termed K-BCS-EM is developed based on the K-BCS framework through which structurally informed estimates of mixture weights, means, and variances are obtained leading to improved convergence reliability and estimation accuracy.
- A comparative evaluation is conducted against classical EM initialization methods where convergence behavior, log-likelihood evolution, parameter estimation accuracy, and robustness are assessed across both synthetic and real datasets.

The remainder of this paper is organized as follows. Section "Related work" reviews EM initialization techniques and BCV-based segmentation principles. Section III introduces the proposed K-BCS-EM approach and its theoretical motivation. Section IV presents the mathematical formulation of the K-BCS-EM-based parameter initialization approach. Section V describes the experimental setup, datasets, and evaluation metrics. Section VI reports and discusses the comparative results. Finally, Section VII concludes the paper and outlines directions for future research.

Related work

Research on EM initialization has pursued different strategies to balance simplicity, computational efficiency, and robustness. Random sampling[15] and random partitioning[16] remain popular because of their ease of implementation. However, their stochastic nature can produce inconsistent results, particularly for noisy or complex data. Jittered sampling[17] improves initialization diversity through controlled perturbations, but remains sensitive to outliers and overlapping components.

Clustering-based methods provide more structured alternatives. K-means initialization[18,19] estimates the initial component centres, but is most effective for balanced, well-separated, and approximately spherical clusters. Its performance may deteriorate for imbalanced, non-Gaussian, or highly multimodal data. Mode-seeking approaches, such as mean-shift clustering[20], can better represent the underlying distributional structure, but are sensitive to bandwidth selection and may scale poorly in high-dimensional settings.

Despite these developments, most initialization methods do not explicitly quantify and exploit the intrinsic modality and inter-component separability of the data before EM optimization. Consequently, they remain susceptible tobunstable convergence and local optima.

To address this limitation, this paper proposes K-BCS-EM, a discriminant-guided segmentation method that maximizes the between-class variance (BCV) criterion to identify statistically separable regions and generate structurally informed initial parameters for EM. Unlike stochastic or centroid-based methods, K-BCS-EM directly identifies unimodal and multimodal structures, making it suitable for complex and time-series data. By incorporating discriminability into the initialization process, the proposed method promotes more stable and accurate EM convergence.

EM algorithm: theoretical background

Consider a parametric model with the observed-data likelihood $L(\boldsymbol{\theta}; \mathrm{x})$, where x denotes the observed data and $\boldsymbol{\theta}$ represents the unknown parameters. Directly maximizing this likelihood may be analytically or computationally challenging when the model involves latent or missing variables. The EM algorithm addresses this problem by introducing a latent variable z such that

$$p(\mathrm{x}; \boldsymbol{\theta}) = \int p(\mathrm{x}, \mathrm{z}; \boldsymbol{\theta})\, d\mathrm{z}, \tag{1}$$

where (x, z) constitutes the complete data. At iteration t, the expectation (E) step computes the expected complete-data log-likelihood as

$$Q\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}\right) = \mathbb{E}_{\mathrm{z}|\mathrm{x},\boldsymbol{\theta}^{(t)}}\left[\log p(\mathrm{x}, \mathrm{z}; \boldsymbol{\theta})\right]. \tag{2}$$

The maximization (M) step subsequently updates the parameters according to

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}\right). \tag{3}$$

By alternating between these two steps, the EM algorithm produces a non-decreasing sequence of observed-data likelihood values and converges to a stationary point, which may be a local rather than a global optimum. These properties make EM particularly suitable for mixture models and other latent-variable or incomplete-data estimation problems.

The EM algorithm configuration

This section discusses how the EM algorithm is configured to estimate the parameters of a univariate K-component Gaussian mixture model (GMM)[21]. The same EM procedure is applied across all initialization strategies to ensure consistency and fair comparison. Algorithm 1 summarizes the overall workflow, while the governing equations are presented alongside the description for clarity.

The implementation follows a standard maximum likelihood EM framework, with additional safeguards to maintain numerical stability, particularly when handling very small probabilities or near-degenerate variance estimates.

Require: Sorted $\mathbf{X} = \{x_1, \ldots, x_N\}$, maximum iterations $T_{\max}$, tolerance ε, number of components K, $\delta_{\min}$, $\sigma^2_{\min}$, and δ
Initialize $\Theta^{(0)}$ using a selected initialization method (e.g., RS, RP, JS, KM, or K-BCS)
Compute the initial log-likelihood $\mathcal{L}^{(0)}$ using Eq. (10)
for $t = 0$ to $T_{\max} - 1$ **do**
    E-step: Compute responsibilities $\gamma_{ik}^{(t)}$ using Eq. (6)
    M-step: Update parameters using Eqs. (7)–(9)
    Evaluate $\mathcal{L}^{(t+1)}$ using Eq. (10)
    if $\left|\mathcal{L}^{(t+1)} - \mathcal{L}^{(t)}\right| < \varepsilon$ **then**
        break
    end if
end for
return Final parameter estimates Θ

Algorithm 1 EM Algorithm for $K > 1$.

Model specification

Given sorted observations $\mathrm{X} = \{x_1, \ldots, x_N\} \subset \mathbb{R}$, the data is modeled as a mixture of K Gaussian components, where $K > 1$ is typically given as an input into the algorithm which can be represented as:

$$p(x \mid \Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \sigma_k^2), \tag{4}$$

where $\Theta = \{\pi_k, \mu_k, \sigma_k^2\}_{k=1}^{K}$, with $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$. Each Gaussian component is defined as

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{5}$$

Initialization interface

The EM algorithm starts from an initial set of parameters $(\boldsymbol{\pi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\sigma}^{2(0)})$, as represented in line 1 of Algorithm 1. The quality of these initial values plays a critical role in determining both the convergence speed and the final solution.

E-step: responsibility computation

At each iteration t (line 3 of Algorithm 1), the responsibility of component k for data point x_i is computed (in line 4 of Algorithm 1) as

$$\gamma_{ik}^{(t)} = \frac{\pi_k^{(t)} \mathcal{N}(x_i \mid \mu_k^{(t)}, \sigma_k^{2(t)})}{\sum_{j=1}^{K} \pi_j^{(t)} \mathcal{N}(x_i \mid \mu_j^{(t)}, \sigma_j^{2(t)}) + \delta_{min}}, \tag{6}$$

where $\delta_{min} > 0$ is a small constant introduced to ensure numerical stability. This step provides a soft assignment of each data point to the mixture components.

M-step: parameter updates

Using the responsibilities from the E-step, the parameters are updated (in line 5 of Algorithm 1) as given below:

$$\pi_k^{(t+1)} = \frac{N_k^{(t)}}{N}, \tag{7}$$

$$\mu_k^{(t+1)} = \frac{1}{N_k^{(t)}} \sum_{i=1}^{N} \gamma_{ik}^{(t)} x_i, \tag{8}$$

$$\sigma_k^{2(t+1)} = \max\left(\frac{1}{N_k^{(t)}} \sum_{i=1}^{N} \gamma_{ik}^{(t)} (x_i - \mu_k^{(t+1)})^2,\ \sigma_{\min}^2\right), \tag{9}$$

where $N_k^{(t)} = \sum_{i=1}^{N} \gamma_{ik}^{(t)}$.

These updates refine the mixture weights, means, and variances based on the current soft assignments, while the variance floor $\sigma_{\min}^2$ prevents numerical instability.

Log-likelihood evaluation and convergence

After each iteration, the observed-data log-likelihood is evaluated (in line 6 of Algorithm 1) as

$$\mathcal{L}^{(t)} = \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k^{(t)} \mathcal{N}(x_i \mid \mu_k^{(t)}, \sigma_k^{2(t)}) + \delta\right), \tag{10}$$

where δ ensures numerical stability.

The algorithm terminates when the change in log-likelihood falls below a predefined threshold (line 7), indicating convergence. The full log-likelihood trajectory is also stored to analyze convergence behavior and compare initialization strategies.

K-BCS-EM approach

This section outlines the foundation of the proposed K-BCS-EM method and its role as an initialization strategy for the EM algorithm. It explains how the K-BCS-EM objective is formulated, how the discriminant-driven segmentation method identifies meaningful data splits, and how cutoff values are used to guide multi-split decisions. The section also describes how these segments are translated into initial estimates for EM parameters. Finally, the computational complexity of the approach is examined to evaluate its scalability and suitability for time-series applications.

Motivation

The K-BCS-EM approach is inspired by the underlying idea behind Otsu's method, which uses between-class variance (BCV) as a criterion for selecting an optimal threshold[22]. Although originally developed for image segmentation, the core principle of maximizing class separability translates naturally into a statistical setting. In this subsection, the key ideas behind Otsu's method are briefly revisited to build intuition for the proposed K-BCS-EM framework.

In image analysis, Otsu's method determines an optimal threshold T by analyzing the histogram of pixel intensities and selecting the value that maximizes the separation between two classes: the foreground C_0 and the background C_1. This separation is quantified using the BCV, denoted by σ_B^2, which is evaluated across all possible threshold values $T = 1, 2, \ldots, \mathrm{T}$, where T denotes the total number of quantized pixel intensity levels. The BCV is defined as[22].

$$\sigma_B^2(T) = w_0(T)w_1(T)\left[\mu_0(T) - \mu_1(T)\right]^2, \tag{11}$$

where $\mu_0(T)$ and $\mu_1(T)$ represent the mean intensities of the two classes, and $w_0(T)$ and $w_1(T)$ denote their respective probabilities. The optimal threshold T^* is then obtained by selecting the value that maximizes this separation given by:

$$T^* = \underset{1 \le T \le N}{\arg\max}\ \sigma_B^2(T). \tag{12}$$

Building on this idea, K-BCS-EM extends the notion of between-class separability beyond pixel intensities to general time series data. Instead of identifying a single threshold, the method frames segmentation as a discriminant-driven process that can be applied recursively to uncover multiple, well-separated regions within the data. By maximizing BCV across successive splits, K-BCS-EM is able to explicitly reveal both unimodal and multimodal structures.

This extension provides a more flexible and principled way to segment data, particularly in time-series settings. Importantly, it also lays a strong foundation for initializing EM parameters, as the identified segments naturally translate into meaningful and structurally informed starting points for mixture modeling.

K-BCS-EM algorithm: computing the BCV measure

This section outlines the operational steps of the K-BCS-EM algorithm with a focus on the computation of the proposed BCV measure used for segmentation purposes. The algorithmic process for computing the discriminant statistic namely the total between-class difference measure $\mathcal{B}$ is documented in Algorithm 2.The between-class variance (BCV) measure is defined as a discriminant criterion used to evaluate the statistical separation produced by a candidate split point. It combines the weighted difference between the means and variances of the two resulting classes, thereby measuring separation in both location and dispersion. A larger BCV value indicates that the split produces two more clearly distinguishable and reasonably balanced classes. In the K-BCS-EM approach, the candidate split with the highest BCV value is selected to guide the segmentation of the data and the subsequent initialization of the EM parameters. While the immediate goal of the algorithm is to compute a robust discriminant measure, its broader purpose is to identify structurally meaningful partition points in the data. These partitions provide a strong foundation for initializing the EM algorithm, ultimately improving both convergence behavior and estimation accuracy. The core functions of Algorithm 2 are described as follows:

Require: Dataset $\mathbf{X} \in \mathbb{R}^N$
Ensure: $\mathcal{B}(T)$ and $(p_L, p_U, \mu_L, \mu_U, \sigma_L^2, \sigma_U^2)$ for all T
1: Initialize $\mathcal{B}, p_L, p_U, \mu_L, \mu_U, \sigma_L^2, \sigma_U^2 \in \mathbb{R}^N \leftarrow 0$
2: **for** $T = 1$ to N **do**
3: $\mathbf{C}_L \leftarrow \{x_{(1)}, \ldots, x_{(T)}\}$
4: $\mathbf{C}_U \leftarrow \{x_{(T+1)}, \ldots, x_{(N)}\}$
5: Compute $p_L(T)$ using (13), $p_U(T)$ using (14)
6: Compute $\mu_L(T)$ using (15), $\mu_U(T)$ using (16)
7: Compute $\sigma_L^2(T)$ using (17), $\sigma_U^2(T)$ using (18)
8: Compute $\mathcal{B}(T)$ using (21)
9: **end for**
10: **return** $\mathcal{B}(T)$ and $(p_L, p_U, \mu_L, \mu_U, \sigma_L^2, \sigma_U^2)$

Algorithm 2 $\mathcal{B}$-measure computation.

Dataset definition, partitioning, and K-BCS-EM initialization

Let N denote the number of observations in the time series dataset represented as an input vector $\mathrm{X} \in \mathbb{R}^N$. The K-BCS-EM algorithm examines X by considering a sequence of candidate is split indices $T \in \{1, 2, \ldots, N\}$. Each candidate split corresponding to each data point defines a temporary partition of the ordered observations into two subsets. For a given split point T, the lower class of X is defined as

$$\mathrm{C}_L = \{x_1, x_2, \ldots, x_T\},$$

and the upper class as

$$\mathrm{C}_U = \{x_{T+1}, x_{T+2}, \ldots, x_N\}.$$

Within the K-BCS-EM framework, each candidate split T represents a potential two-segment view of the data. Instead of relying on random assignments or distance-based clustering, the method evaluates whether each candidate partition reflects meaningful statistical separation in the underlying distribution. Split points that exhibit

strong between-class discrimination are treated as structurally informative split points and are used to guide initialization in likelihood-based mixture models.

For every candidate split T, the K-BCS-EM algorithm computes a set of class-specific statistics, including the class proportions $p_L(T)$ and $p_U(T)$, the class means $\mu_L(T)$ and $\mu_U(T)$, the class variances $\sigma_L^2(T)$ and $\sigma_U^2(T)$, and the corresponding between-class separation score $\mathcal{B}(T)$. Together, these quantities provide a comprehensive measure of how suitable each split is for generating reliable and well-informed mixture-component initialization.

Class-wise statistical estimation

For a given split index T, the class proportions are defined as

$$p_L(T) = \frac{T}{N}, \tag{13}$$

$$p_U(T) = 1 - p_L(T), \tag{14}$$

which naturally translate into mixture weights when used for EM initialization. These proportions reflect how the data is distributed across the two segments induced by the split. The corresponding class means are computed as

$$\mu_L(T) = \frac{1}{T} \sum_{x \in C_L} x_i, \tag{15}$$

$$\mu_U(T) = \frac{1}{N-T} \sum_{x \in C_U} x_i, \tag{16}$$

providing data-driven estimates of component locations that are directly informed by the segmentation. Similarly, the class variances are estimated as

$$\sigma_L^2(T) = \frac{1}{T} \sum_{x \in C_L} (x_i - \mu_L(T))^2, \tag{17}$$

$$\sigma_U^2(T) = \frac{1}{N-T} \sum_{x \in C_U} (x_i - \mu_U(T))^2, \tag{18}$$

capturing the spread of each segment around its corresponding mean. Accurate estimation of these dispersion parameters is critical, as poor variance initialization is a well-known source of slow convergence, numerical instability, and likelihood plateaus in EM-based mixture modeling[23].

Computing the $\mathcal{B}$-measure for structural separation

The K-BCS-EM algorithm evaluates the quality of each candidate split using the $\mathcal{B}$-measure, a balanced between-class criterion that captures separation in both location and dispersion. The contribution from mean separation is defined as

$$\beta_\mu(T) = p_L(T)\, p_U(T)\, \|\mu_L(T) - \mu_U(T)\|^2, \tag{19}$$

while the contribution from variance differences is given by

$$\beta_{\sigma^2}(T) = p_L(T)\, p_U(T)\, \|\sigma_L^2(T) - \sigma_U^2(T)\|, \tag{20}$$

reflecting differences in dispersion between the two segments.

The overall between-class statistic is obtained by combining these two effects given by:

$$\mathcal{B}(T) = \frac{\beta_{\mu}(T) + \beta_{\sigma^2}(T)}{2}, \tag{21}$$

providing a unified measure of how well a given split separates the data.

Unlike the classical criteria used in Otsu's method[22] that focus only on the variance differences, the K-BCS-EM $\mathcal{B}$-measure explicitly incorporates both mean and variance heterogeneity in Eq. (21). This approach makes it possible to detect meaningful partitions even when components overlap in locations but differ in spread. At the same time, it remains responsive to shifts in mean when variances are similar. This dual sensitivity allows K-BCS-EM to capture a wider range of separation patterns commonly observed in real-world data.

From a mixture modeling perspective, large values of $\mathcal{B}(T)$ indicate split points where the data shows clear evidence of component-level separation. Initializing EM parameters at or near these points can reduce ambiguity in early responsibility assignments, stabilize the likelihood updates, and promote faster convergence. In this way, the EM algorithm can be guided toward regions of the parameter space that better reflect the underlying structure of the data.

By evaluating $\mathcal{B}(T)$ across all candidate split points, K-BCS-EM offers a principled, data-driven way to identify segmentation points that correspond to genuine distributional structure rather than arbitrary or heuristic partitions. This directly addresses a key limitation of conventional EM initialization methods, which often fail to account for latent modality and variance differences.

Importantly, the design of the $\mathcal{B}$-measure aligns closely with the objectives of the EM algorithm. Depending only on mean separation can fail when components differ primarily in variance, while variance-only criteria may overlook shifts in location. However, by integrating both aspects, the $\mathcal{B}(T)$-measure can identify partitions that are jointly informative for initializing mixture weights, means, and variances, the key parameters governing EM performance.

Require: Time series $\mathbf{X} = \{x_1, \ldots, x_N\}$, target segments K, cutoff τ, minimum size $n_{\min}$, constant ε
Ensure: Segments $\mathcal{S} = \{[a_k, b_k)\}_{k=1}^{K_{\text{eff}}}$ and effective number of segments K_{eff}
Initialize $\mathcal{S} \leftarrow \{[0, N)\}$
while $|\mathcal{S}| < K$ **do**
  $bestScore \leftarrow -\infty$, $bestSeg \leftarrow \varnothing$, $bestSplit \leftarrow \varnothing$
  for each segment $[a, b) \in \mathcal{S}$ **do**
    $n \leftarrow b - a$
    if $n < 2n_{\min}$ **then**
      continue
    end if
    Compute $segVar$ over $\{x_{a+1}, \ldots, x_b\}$
    $localMaxBCV \leftarrow -\infty$, $localBestT \leftarrow \varnothing$
    for $T = 0$ to $n - 2$ **do**
      Form $L_T = \{x_{a+1}, \ldots, x_{a+T+1}\}$ and $U_T = \{x_{a+T+2}, \ldots, x_b\}$
      if $|L_T| < n_{\min}$ **or** $|U_T| < n_{\min}$ **then**
        continue
      end if
      Use Algorithm 2 to compute $p_L(T)$, $p_U(T)$, $\mu_L(T)$, $\mu_U(T)$, $\sigma_L^2(T)$, $\sigma_U^2(T)$, $\beta_\mu(T)$, $\beta_{\sigma^2}(T)$, and $\mathcal{B}(T)$
    end for
    Obtain $T^\star$ using Eq. (22)
    Obtain $\mathcal{B}(T^\star)$ using Eq. (23)
    if $\mathcal{B}(T^\star) > localMaxBCV$ **then**
      $localMaxBCV \leftarrow \mathcal{B}(T^\star)$
      $localBestT \leftarrow T^\star$
    end if
    if $localBestT \neq \varnothing$ **then**
      $\mathcal{B}_{\text{norm}} \leftarrow \dfrac{localMaxBCV}{segVar + \varepsilon}$
      if $\mathcal{B}_{\text{norm}} \geq \tau$ **and** $\mathcal{B}_{\text{norm}} > bestScore$ **then**
        $bestScore \leftarrow \mathcal{B}_{\text{norm}}$
        $bestSeg \leftarrow [a, b)$
        $bestSplit \leftarrow a + localBestT + 1$
      end if
    end if
  end for
  if $bestSeg = \varnothing$ **then**
    break
  end if
  Split $bestSeg$ at $bestSplit$ and update $\mathcal{S}$
end while
$K_{\text{eff}} \leftarrow |\mathcal{S}|$
return $\mathcal{S}, K_{\text{eff}}$

Algorithm 3 K-BCS-EM multi-split segmentation.

K-BCS-EM algorithm: computing K multi-split segments

This section presents the extension of the binary discriminant split mechanism into a multi-split segmentation framework capable of producing up to K statistically meaningful partitions. The resulting segmentation serves as a structured and data-driven foundation for initializing the EM algorithm. The complete procedure is summarized in Algorithm 3, while the key steps are described below.

Data representation and initialization

Let the input time series remain as $\mathrm{X} = \{x_1, \ldots, x_N\} \subset \mathbb{R}$. The segmentation is represented as a set of contiguous index intervals

$$\mathcal{S} = \{[a_1, b_1), [a_2, b_2), \ldots, [a_{K_{\text{eff}}}, b_{K_{\text{eff}}})\},$$

such that

$$0 = a_1 < b_1 = a_2 < \cdots < a_{K_{\text{eff}}} < b_{K_{\text{eff}}} = N,$$

which ensures that the segments are contiguous, non-overlapping, and collectively span the dataset.

The algorithm is initialized with a single segment covering the entire data domain, i.e., $\mathcal{S} = \{[0, N)\}$.

BCV-based binary split evaluation within a segment

Consider a candidate segment $\{x_{a+1}, \ldots, x_b\}$ of length $n = b - a$. Before evaluating candidate split points, a feasibility condition is imposed to ensure that any resulting partition satisfies the minimum segment size requirement. Specifically, a valid binary split requires both resulting subsets to contain at least $n_{\min}$ observations. This implies that a necessary condition for splitting is $n \geq 2n_{\min}$. Consequently, segments with $n < 2n_{\min}$ are not considered for further splitting, as no admissible partition can be formed. Thus, for any admissible split index $T \in \{0, \ldots, n-2\}$, the segment is partitioned into

$$L_T = \{x_{a+1}, \ldots, x_{a+T+1}\},$$

$$U_T = \{x_{a+T+2}, \ldots, x_b\}.$$

with consideration given to only valid splits satisfying the minimum size constraint

$$|L_T| \geq n_{\min}, \quad |U_T| \geq n_{\min}$$

The upper bound $T \leq n-2$ ensures that both resulting classes are non-empty. In particular, if $T = n-1$, the upper class (i.e. right subset) would be empty, which makes it impossible to compute class statistics such as the mean and variance. This constraint guarantees that each candidate split produces two valid partitions.

For each admissible split, the between-class statistic $\mathcal{B}(T)$ (defined in Eq. (21)) is computed using class proportions, means, and variances derived from L_T and U_T using Algorithm 2. The optimal split within a segment is selected as

$$T^\star = \arg\max_T \mathcal{B}(T). \tag{22}$$

Once the optimal split index $T^\star$ is selected, then the corresponding between-class score is obtained as

$$\mathcal{B}(T^\star) = \max_T \mathcal{B}(T). \tag{23}$$

Normalized split statistic and acceptance criterion

To ensure comparability across segments of different scales, the maximum between-class value $\mathcal{B}(T^\star)$ is normalized by the variance of the segment:

$$\mathcal{B}_{norm} = \frac{\mathcal{B}(T^\star)}{\mathrm{Var}(\{x_{a+1}, \ldots, x_b\}) + \varepsilon}, \tag{24}$$

where $\varepsilon > 0$ is a small numerical constant introduced to prevent division by zero and ensure stability when the segment variance is very small. This is particularly important for segments with near-constant values, where $\mathrm{Var}(\cdot)$ may approach zero and lead to unstable or inflated values of the normalized statistic. Following the computation of $\mathcal{B}_{norm}$, a segment is considered splittable if

$$\mathcal{B}_{norm} \geq \tau, \tag{25}$$

where τ is a user-defined threshold controlling the strength of acceptable separation.

Greedy multi-split segmentation procedure

At each iteration, the algorithm evaluates all current segments in $\mathcal{S}$ and computes their corresponding $\mathcal{B}_{norm}$. Among all admissible candidates, the split with the highest $\mathcal{B}_{norm}$ is selected globally and applied to its corresponding segment.

Let the best segment be $[a, b)$ and its optimal split index be $T^\star$. The segment is divided at the boundary

$$c = a + (T^\star + 1),$$

yielding two new segments given by:

$$[a, c) \quad \text{and} \quad [c, b).$$

The original segment is then replaced by these two new segments in $\mathcal{S}$. This process is repeated iteratively in a greedy manner, which ensures that at each step the most statistically significant split is applied.

Stopping criteria and final segmentation

The segmentation procedure terminates when either of the following conditions are met:

- the desired number of segments K has been reached, or
- no remaining segment satisfies the acceptance criterion.

As a result, the algorithm produces an effective number of segments $K_{\text{eff}} \leq K$, which reflects the intrinsic structure supported by the data. The final segmentation

$$\mathcal{S} = \{[a_k, b_k)\}_{k=1}^{K_{\text{eff}}}$$

provides a set of structurally meaningful partitions that can be directly used for initializing mixture model parameters.

The proposed K-BCS-EM multi-split method differs from conventional clustering or random initialization methods by explicitly identifying statistically significant partition points prior to model fitting. By combining both mean and variance separation through the $\mathcal{B}$-measure and applying a normalized acceptance criterion $\mathcal{B}_{norm}$, the method ensures that only meaningful structural splits are retained. This segmentation-driven approach improves interpretability, reduces sensitivity to initialization, and provides a principled pathway for initializing EM in time-series datasets.

K-BCS-EM algorithm: initializing the EM algorithm

Given the final segmentation $\mathcal{S} = \{[a_k, b_k)\}_{k=1}^{K_{\text{eff}}}$ obtained from Algorithm 3, the initial mixture parameters are computed directly from segment-level statistics as

$$\pi_k^{(0)} = \frac{b_k - a_k}{N}, \tag{26}$$

$$\mu_k^{(0)} = \frac{1}{b_k - a_k} \sum_{i=a_k+1}^{b_k} x_i, \tag{27}$$

$$\sigma_k^{2(0)} = \frac{1}{b_k - a_k} \sum_{i=a_k+1}^{b_k} \left(x_i - \mu_k^{(0)}\right)^2. \tag{28}$$

The resulting set $\{(\pi_k^{(0)}, \mu_k^{(0)}, \sigma_k^{2(0)})\}_{k=1}^{K_{\text{eff}}}$ defines the K-BCS-EM initialization parameter estimates, which are used to seed subsequent EM iterations in a manner that reflects the underlying structure of the data. Thereafter, the EM algorithm proceeds until convergence or a stopping criterion is reached via Algorithm 1. The final initialization based on the initial estimated parameters and handoff to the EM algorithm is documented in Algorithm 4.

Unlike conventional EM initialization strategies, which often rely on random assignment or clustering-based heuristics, the K-BCS-EM formulation identifies candidate partitions directly from the data by detecting structural changes in both the mean and variance. This is particularly important for EM, as its convergence behavior and final solution are highly sensitive to the choice of initial parameters. Consequently, the K-BCS-EM algorithm is designed to assess whether meaningful statistical separability exists and to guide initialization decisions that encourage faster convergence and reduce the risk of the EM algorithm becoming trapped in poor local optima.

In summary, the K-BCS-EM initialization pipeline begins by accepting the data and initializing the required statistical parameters. It then systematically evaluates candidate partition points, computes class-specific statistics, and quantifies the between-class separability. Based on this process, structurally informative splits are selected and mapped directly to mixture components, resulting in data-driven initial estimates of the EM parameters that naturally reflect the underlying modality of the data. By separating mode discovery from likelihood optimization, this discriminant-guided approach introduces the potential to improve convergence stability and enhances robustness in time-series datasets.

Require: Final segmentation $\mathcal{S} = \{[a_k, b_k)\}_{k=1}^{K_{\text{eff}}}$ from Algorithm 3, maximum EM iterations $I_{\max}$
Ensure: Estimated mixture parameters Θ
Initialize $\Theta^{(0)} \leftarrow \varnothing$
for $k = 1$ to K_{eff} **do**
  $n_k \leftarrow b_k - a_k$
  $\pi_k^{(0)} \leftarrow \frac{n_k}{N}$
  $\mu_k^{(0)} \leftarrow \frac{1}{n_k} \sum_{i=a_k+1}^{b_k} x_i$
  $\sigma_k^{2(0)} \leftarrow \frac{1}{n_k} \sum_{i=a_k+1}^{b_k} \left(x_i - \mu_k^{(0)}\right)^2$
  Append $(\pi_k^{(0)}, \mu_k^{(0)}, \sigma_k^{2(0)})$ to $\Theta^{(0)}$
end for
Use $\Theta^{(0)}$ to initialize the EM algorithm in Algorithm 1
Run EM for at most $I_{\max}$ iterations
return Θ

Algorithm 4 K-BCS-EM: initialization and EM handoff.

Time complexity analysis of K-BCS-EM

This subsection analyzes the computational complexity of the K-BCS-EM algorithm by examining the cost of Algorithm 2 ($\mathcal{B}$-measure computation) and Algorithm 3 (multi-split segmentation).

Complexity of $\mathcal{B}$-measure computation (Algorithm 2)

Algorithm 2 evaluates the between-class statistic $\mathcal{B}(T)$ for all candidate split indices $T \in \{1, \ldots, N\}$. For each T, the algorithm computes class proportions, means, and variances over the classes C_L and C_U.

In the naive implementation, computing the mean and variance for each split requires iterating over the corresponding classes, resulting in a cost of $\mathcal{O}(N)$ per split. Since this is repeated for all N candidate split points, the overall time complexity of Algorithm 2 will be

$$\mathcal{O}(N^2).$$

However, this cost can be reduced to $\mathcal{O}(N)$ by using cumulative sums and cumulative squared sums to compute means and variances incrementally. Nevertheless, in its general form, Algorithm 2 exhibits quadratic complexity.

Complexity of multi-split segmentation (Algorithm 3)

Algorithm 3 applies a greedy multi-split procedure to recursively partition the data. At each iteration, all current segments are evaluated to determine the best admissible split.

Let K denote the maximum number of segments. In the worst case, the algorithm performs up to $K - 1$ splitting iterations. During each iteration, every current segment is examined, and for a segment of length n, all candidate split indices $T \in \{0, \ldots, n-2\}$ are evaluated.

Since the sum of segment lengths across all segments remains N, the total cost of evaluating all splits across all segments in a single iteration is bounded by $\mathcal{O}(N^2)$ when Algorithm 2 is used directly.

Thus, over K iterations, the overall time complexity of Algorithm 3 is

$$\mathcal{O}(KN^2).$$

Overall complexity of K-BCS-EM

Combining both components, the total computational complexity of the K-BCS-EM framework is dominated by the repeated evaluation of the $\mathcal{B}$-measure within the multi-split procedure, yielding an overall worst-case complexity of

$$\mathcal{O}(KN^2).$$

In practice, the effective number of segments K_{eff} is often significantly smaller than K, and the early termination condition based on the cutoff τ further reduces the number of iterations. Additionally, the use of cumulative statistics can reduce the cost of $\mathcal{B}$-measure computation, thus improving scalability for large datasets.

Although the theoretical complexity is quadratic in N, the algorithm remains computationally feasible for moderate dataset sizes. Furthermore, the segmentation process is inherently parallelizable across segments, which offers opportunities for further optimization in large-scale implementations.

Impact and limitations of EM initialization methods

The EM algorithm's performance depends strongly on the initial mixture weights, means, and variances. Appropriate initial values improve responsibility assignments, accelerate convergence, and increase estimation accuracy, whereas poor initialization may cause instability, degenerate variances, or convergence to a suboptimal solution.

Existing approaches have several limitations. RS, RP, and JS initialization are stochastic and may produce inconsistent results. Although KM is data-driven, it focuses mainly on differences in component means and

may struggle with overlapping components, unequal variances, imbalanced weights, and masked modes. These methods may also impose a predefined number of components without verifying whether the data supports the resulting partitions.

Consequently, the K-BCS-EM initialization method presented in this section has been introduced to address these limitations. It evaluates the observable structure of the data before the EM iterations begin while considering separation in both the component location and dispersion of the dataset. The experimental results following the stress-test of the K-BCS-EM and other competing methods using both synthetic and real datasets will be presented in the next section.

Results and discussion

This section examines how the performance of the EM algorithm varies across different initialization strategies. The experimental design is structured to isolate the effect of initialization quality on convergence stability, estimation accuracy, and computational cost under a range of controlled scenarios, using both synthetic and real datasets. To ensure a fair comparison, all methods are evaluated under the same mixture configurations, convergence criteria, and stopping conditions. In addition, results are averaged over multiple Monte Carlo trials to account for inherent stochastic variability.

Competing approaches

The performance of the proposed K-BCVS-EM approach was evaluated against other widely used EM initialization strategies. The RS-EM method[15] initializes component means by randomly sampling from the data, assigns equal mixture weights, and assumes a common global variance. In contrast, RP-EM[8] begins with randomly generated data partitions, from which all model parameters are directly estimated. The JS-EM approach[17] introduces controlled randomness by perturbing the global mean with Gaussian noise, although it remains largely insensitive to underlying multimodal structure. Finally, KM-EM[10] utilizes one-dimensional K-means clustering to derive initial parameters from cluster-level statistics, which provides more structured initialization than purely random methods, but still relying primarily on distance-based separation.

Parameter settings

This subsection outlines the initialization rules and parameter configurations used for all baseline methods and the proposed K-BCS-EM approach, as summarized in Table 1. The goal is to ensure a fair and consistent comparison across all methods by standardizing key experimental conditions while allowing each initialization strategy to operate under its natural design.

Table 1 Initialization rules and parameter settings for all baselines and K-BCS-EM.

Method	Initialization rule	Key parameters	Values used
RS-EM	Randomly sample K data points as means; equal weights; global variance	K, restarts R	K per scenario; $R = 20$
RP-EM	Random partition into K groups; cluster moments	K, restarts R	K per scenario; $R = 20$
JS-EM	Jittered global mean initialization	K, jitter scale λ, R	$\lambda = 0.5\,\mathrm{sd}(X)$; $R = 20$
KM-EM	1D K-means cluster moments	K, K-means restarts	K per scenario; 10 restarts
K-BCS-EM	Multi-split segmentation via $\mathcal{B}$; segment moments $\lvert\mathcal{S}\rvert = K$	$\pi_k^{(0)}$, $\mu_k^{(0)}$, $\sigma_k^{2(0)}$; K, min segment size n_{min}	$n_{min} = \max\{5, \lfloor 0.05N \rfloor\}$

All methods use identical EM settings: $I_{\max} = 50$, $\varepsilon = 10^{-6}$, and variance floor $\delta_{\min} = 10^{-6}$. Randomized initializers report the best solution over R restarts

For the random-based approaches, RS-EM and RP-EM, the initialization variability is mitigated through multiple independent restarts. In both cases, $R = 20$ restarts are performed, and the best solution is selected based on the final log-likelihood. RS-EM samples initial means directly from the data, while RP-EM generates random partitions from which initial parameters are estimated. Similarly, JS-EM introduces stochasticity through a jitter mechanism, where the global mean is perturbed using Gaussian noise scaled by $\lambda = 0.5\,\mathrm{sd}(X)$, where sd denotes the standard deviation of the dataset X. This scaling ensures that the perturbation adapts to the spread of the data while remaining sufficiently controlled.

For KM-EM, initialization is based on one-dimensional K-means clustering with 10 independent restarts used to reduce sensitivity to local minima. This provides a more structured alternative to purely random initialization, although it remains dependent on distance-based partitioning.

In contrast, the proposed K-BCS-EM approach eliminates the need for stochastic restarts by deriving initial parameters directly from the data through multi-split segmentation. The number of segments is aligned with the target number of mixture components, i.e., $|\mathcal{S}| = K$, and the initial mixture weights, means, and variances are computed from segment-level statistics. To ensure statistical reliability, a minimum segment size is enforced using $n_{min} = \max\{5, \lceil 0.05N \rceil\}$, which adapts to the dataset size while preventing degenerate or unstable partitions.

To maintain consistency across all methods, the EM algorithm is executed using identical settings, with a maximum of $I_{\max} = 50$ iterations. In addition, a variance floor of $\varepsilon = 10^{-6}$ is applied to prevent numerical instability arising from vanishing variance estimates. These standardized settings ensure that observed performance differences can be attributed primarily to the quality of the initialization strategy rather than discrepancies in EM configuration.

Performance metrics

To evaluate the performance of the different initialization strategies, a ground-truth mixture model is considered which is defined as:

$$f(x) = \sum_{k=1}^{K} \pi_k^\star \mathcal{N}(x \mid \mu_k^\star, \sigma_k^{2\star}), \tag{29}$$

where $\pi_k^\star$, $\mu_k^\star$, and $\sigma_k^{2\star}$ represent the true mixture weights, means, and variances, respectively. After convergence of the EM algorithm, each method produces a set of estimated parameters

$$\widehat{\Theta} = \{\hat{\pi}_k, \hat{\mu}_k, \hat{\sigma}_k^2\}_{k=1}^{\hat{K}}.$$

Based on these estimates, several performance metrics are computed to assess estimation accuracy, convergence behavior, and computational efficiency.

Parameter estimation accuracy (RMSE)

Estimation accuracy is evaluated using the root mean squared error (RMSE), which measures the deviation between estimated parameters and their true values.

Mean RMSE

$$\mathrm{RMSE}_\mu = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\hat{\mu}_{(k)} - \mu_{(k)}^\star\right)^2}. \tag{30}$$

Variance RMSE

$$\mathrm{RMSE}_{\sigma^2} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\hat{\sigma}_{(k)}^2 - \sigma_{(k)}^{2\star}\right)^2}. \tag{31}$$

Together, these metrics provide a direct measure of how accurately each method recovered the underlying mixture parameters.

Log-Likelihood and Convergence Behavior

The evolution of the log-likelihood provides insight into the convergence dynamics of the EM algorithm. At each iteration t, the log-likelihood[24] is computed as

$$\ell^{(t)} = \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \hat{\pi}_k^{(t)} \mathcal{N}(x_i \mid \hat{\mu}_k^{(t)}, \hat{\sigma}_k^{2(t)})\right). \tag{32}$$

To assess convergence behavior, the following diagnostics are considered:

- Final log-likelihood $\ell^{(\text{final})}$
- Log-likelihood trajectory $\{\ell^{(t)}\}$
- Number of iterations required for convergence

A higher final log-likelihood typically indicates convergence to a better local optimum, while a smooth and monotonic trajectory reflects stable and well-initialized parameter updates.

Monte carlo aggregation

To ensure robustness and reduce the influence of randomness, each experimental scenario was evaluated over M independent Monte Carlo trials using different random seeds[25]. A value of $M = 100$ was applied across all experiments conducted in this study

For each performance metric $\mathcal{M}$ and method j, the mean and standard deviation were computed as

$$\bar{\mathcal{M}}_j = \frac{1}{M} \sum_{m=1}^{M} \mathcal{M}_j^{(m)}, \tag{33}$$

$$\text{Std}(\mathcal{M}_j) = \sqrt{\frac{1}{M-1} \sum_{m=1}^{M} \left(\mathcal{M}_j^{(m)} - \bar{\mathcal{M}}_j\right)^2}. \tag{34}$$

Results are reported in the form

$$\text{mean} \pm \text{standard deviation}.$$

In addition, Monte Carlo averaging enables the construction of smoother log-likelihood convergence curves, aggregated histogram densities, and representative mixture probability density functions based on mean parameter estimates.

Synthetic datasets

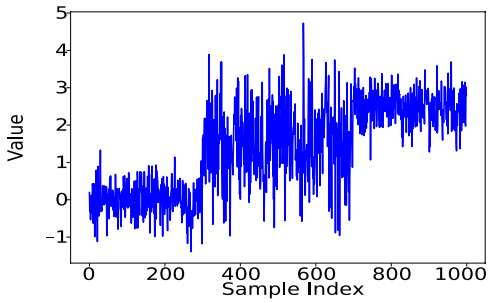
To evaluate the robustness and initialization sensitivity of the EM algorithm, a suite of one-dimensional synthetic datasets were constructed. Each dataset was carefully designed as a controlled stress-test scenario, with the aim of exposing well-known challenges in likelihood-based mixture modeling, including component overlap, class imbalance, heteroscedasticity, heavy-tailed contamination, and identifiability issues. The datasets were generated from finite mixtures of Gaussian distributions of the form

$$x_i \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \sigma_k^2), \tag{35}$$

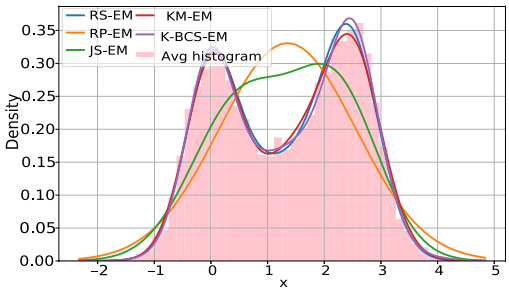
where K denotes the true number of mixture components, π_k represents the component weights, and (μ_k, σ_k^2) correspond to the component-specific means and variances. The number of samples drawn from each component is proportional to π_k, and all datasets were independently replicated across Monte Carlo trials to ensure statistical reliability of the results. The results obtained are discussed as follows:

Case 1: overlap with unequal variances

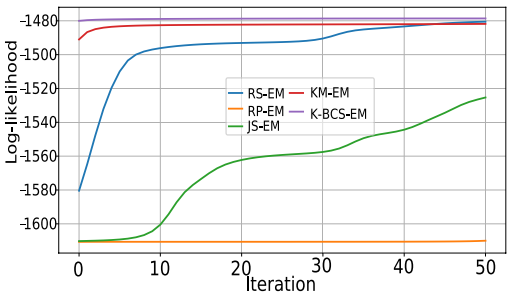
The dataset representing case 1 is shown in Fig. 2a. This scenario consists of $K = 3$ components with closely spaced means and heterogeneous variances. Two relatively narrow components were positioned around a more diffused central component, resulting in substantial density overlap. The resulting likelihood surface was nearly flat in certain regions, which makes it difficult for the EM algorithm to clearly distinguish between components. As a result, this setting often leads to component merging, slow convergence, and unstable parameter updates, particularly when initialization is poorly chosen.



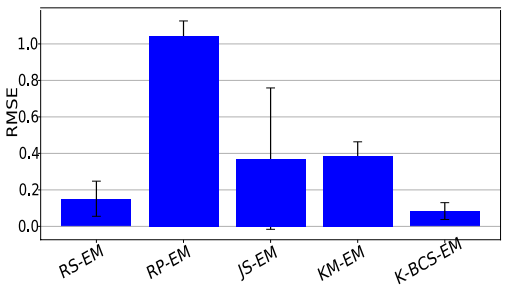
(a) Severe overlap with unequal variances



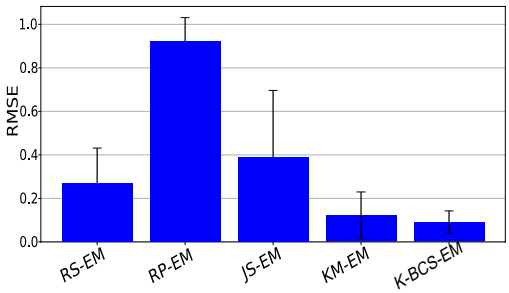
(b) Fitted histogram



(c) EM convergence



(d) Variane RMSE



(e) Mean RMSE

Fig. 2 Case 1 – average performance under severe overlap with unequal variances ($K = 3$).

Figs. 2eand 2dpresent the Monte Carlo mean and variance RMSE results for the overlap scenario. From these results, it is clear that K-BCS-EM consistently achieved the lowest errors in both the mean and variance estimation evaluations. In addition, the error bars were noticeably smaller, which indicates that the method produced more stable estimates across repeated trials. In contrast, RP-EM recorded the largest errors, while RS-EM and JS-EM demonstrated moderate performance but with higher variability. Although KM-EM performed reasonably well in estimating the means, it still fell short of K-BCS-EM in accurately estimating the variance. These observations were consistent across both the mean and variance comparison plots.

The convergence behavior is illustrated in Fig. 2c, which shows the Monte Carlo averaged log-likelihood trajectories. As expected, all methods exhibited a monotonic increase in log-likelihood, which confirms correct implementation of the EM algorithm. However, K-BCS-EM stood out by converging more smoothly and reaching a slightly higher final log-likelihood compared to the other initialization strategies. While most methods stabilized within a similar number of iterations, the trajectory of K-BCS-EM was more stable, which reflects the benefit of improved initialization.

Further insight is provided in Fig. 2b, where the averaged data histogram is shown alongside the reconstructed probability density functions. Visually, the K-BCS-EM curve aligns most closely with the empirical density, particularly in the overlapping regions where separating components are inherently difficult to estimate. By comparison, RP-EM and JS-EM exhibited noticeable distortions in both the peak location and spread, which indicates less accurate parameter estimation. KM-EM captured the overall shape of the distribution but still showed slight inconsistencies in dispersion. This visual agreement reinforces the quantitative findings.

Essentially, both the RMSE and convergence results consistently demonstrate that the K-BCS-EM approach provided superior estimation accuracy, improved stability, and more reliable density reconstruction in the presence of overlapping mixture components.

Case 2: heavy-tailed distribution

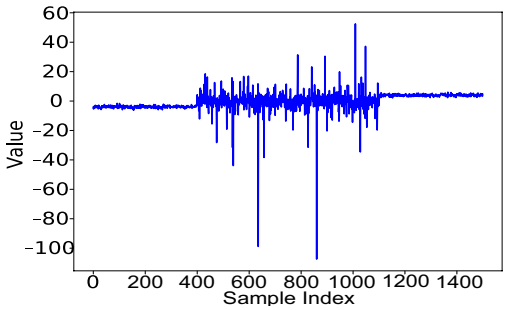
Fig. 3apresents the data that produced the heavy-tailed distribution. The central component in this scenario was generated from a Student-*t* distribution with low degrees of freedom, while the remaining components were Gaussian. Although the fitted model assumes Gaussianity throughout, the heavy-tailed component introduced outliers and inflated variance. This can mislead the EM algorithm into assigning excessive spread to Gaussian components or producing spurious splits, thereby degrading estimation accuracy.

The mean RMSE comparison (Fig. 3d) clearly shows that K-BCS-EM achieved substantially lower error relative to the other methods. In contrast, KM-EM exhibited extreme mean estimation instability, with dramatically larger RMSE values. RS-EM, RP-EM, and JS-EM also displayed elevated errors and considerable dispersion, which reflects their sensitivity to extreme observations in heavy-tailed data.

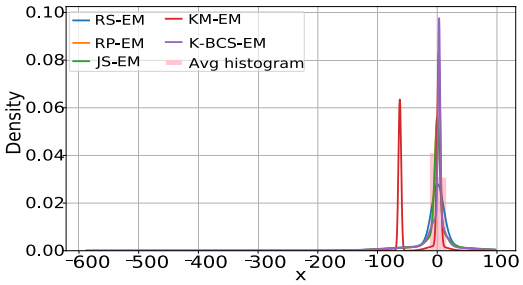
The variance RMSE plot (Fig. 3e) reveals very large dispersion errors across all methods, which was expected under heavy-tailed conditions. However, K-BCS-EM consistently maintained lower variance RMSE compared to the alternatives. The scale of the variance error bars illustrated the instability of random-based initializations when extreme values inflate dispersion estimates.

The averaged EM convergence curves (Fig. 3c) demonstrated monotonic log-likelihood ascent for all methods. Although convergence speeds appear broadly similar, the trajectories indicate that unstable initializations (particularly KM-EM) exhibit more irregular ascent behavior before stabilizing. K-BCS-EM showed a comparatively smoother convergence path.

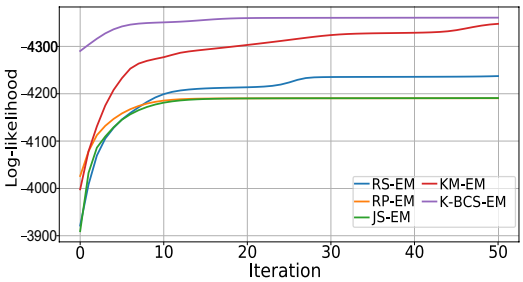
Finally, the averaged histogram with reconstructed PDFs (Fig. 3b) visually confirms the parameter estimation differences. The K-BCS-EM density aligns more closely with the empirical histogram, particularly around the central mass, while alternative methods either over concentrate or misplace mass due to distorted parameter estimates. KM-EM, in particular, shows pronounced misalignment consistent with its extreme mean RMSE.



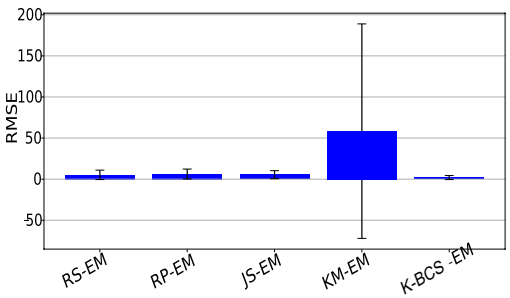
(a) Heavy-Tailed Distribution



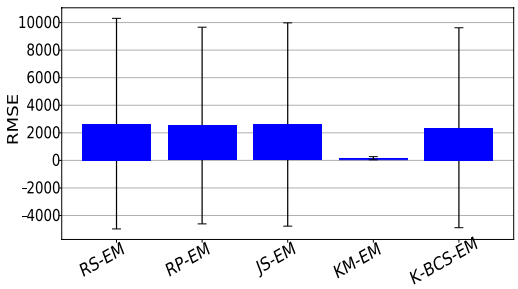
(b) Fitted histogram



(c) EM convergence



(d) Variance RMSE



(e) Mean RMSE

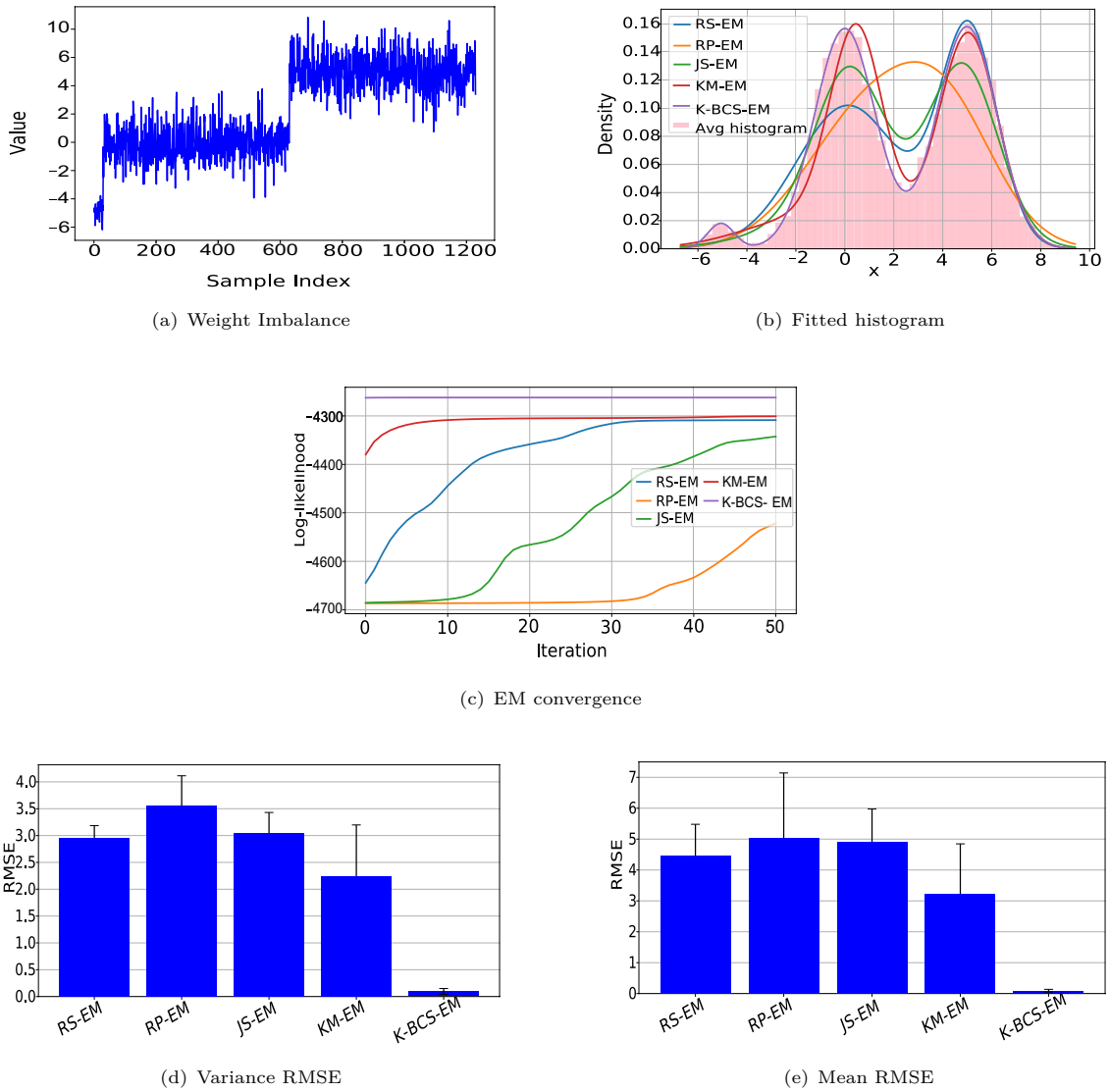
Fig. 3 Case 2 - average performance under severe heavy tail with unequal variances($K = 3$).

Essentially, the outputs corroborate the numerical results: heavy-tailed distributions severely challenge traditional initialization schemes, whereas K-BCS-EM demonstrated improved robustness, stability, and density reconstruction despite increased computational cost.

Case 3: weight imbalance

In this data as shown in Fig. 4a, the first component was represented only by a small fraction of the overall data, while the remaining two components dominated the mixture. Although the component means were well separated, the minor component was frequently ignored or absorbed during the EM updates. This scenario is therefore useful for evaluating whether an initialization strategy can reliably detect and preserve small but meaningful modes in the presence of severe class imbalance.

The mean RMSE comparison (Fig. 4d) shows a clear performance gap between K-BCS-EM and the competing methods. K-BCS-EM achieved the lowest mean estimation error, followed by KM-EM, while RS-EM,



(a) Weight Imbalance

(b) Fitted histogram

(c) EM convergence

(d) Variance RMSE

(e) Mean RMSE

Fig. 4 Case 3 - imbalanced weights with unequal mean ($K = 3$).

RP-EM, and JS-EM exhibited substantially larger RMSE values. The visual separation between K-BCS-EM and the random-based initializations highlights the sensitivity of traditional schemes to unequal mixing proportions.

A similar trend was observed in the variance RMSE plot (Fig. 4e). K-BCS-EM again recorded the lowest dispersion error, with the KM-EM's performance being second best. The remaining methods display noticeably larger variance errors, which indicated difficulty in accurately recovering component spreads when cluster sizes differ significantly.

The EM convergence curves (Fig. 4c) demonstrated monotonic log-likelihood ascent for all methods. Convergence speeds appeared broadly comparable; however, the smoother and more stable trajectory of K-BCS-EM reflects improved initial parameter placement under imbalance.

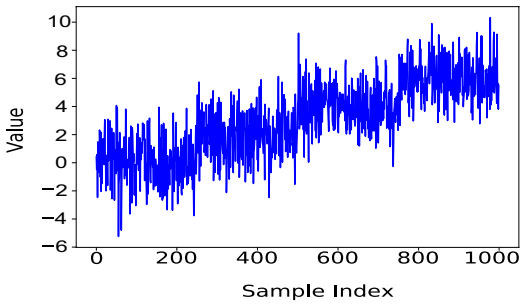
Finally, the averaged histogram with reconstructed PDFs (Fig. 4b) provided clear visual confirmation of parameter recovery quality. The K-BCS-EM density closely aligned with the empirical histogram, particularly in representing both dominant and minority components. In contrast, RP-EM and JS-EM showed visible distortion in peak height and spread, while RS-EM exhibited mild misalignment. KM-EM approximates the structure reasonably well but remained less precise than K-BCS-EM.

Essentially, the results confirm that under class imbalance, the K-BCS-EM approach delivered superior accuracy, improved stability, and more faithful density reconstruction compared to conventional initialization strategies.

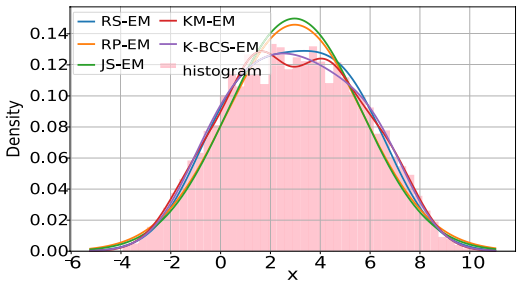
Case 4: Mode Masking under High Variance

This four-component dataset shown in Fig. 5aconsisted of equally weighted and regularly spaced components, each with relatively large variance. Despite the underlying multimodal structure, the combined density appears nearly unimodal or only weakly multimodal. This makes it difficult for standard initialization methods to identify distinct modes, which often causes the EM algorithm to collapse multiple components into fewer effective clusters. This scenario is particularly useful for evaluating the ability of discriminant-guided methods to uncover hidden structure.

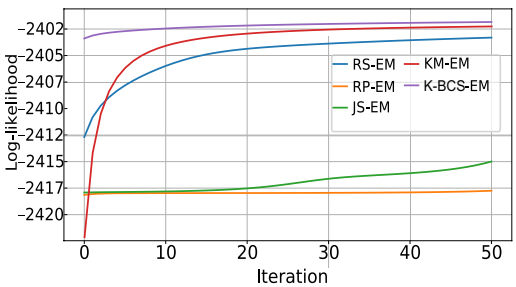
The mean RMSE comparison (Fig. 5d) clearly shows that K-BCS-EM attained the lowest mean estimation error among all competing methods. KM-EM performed second best, while RS-EM, JS-EM, and particularly



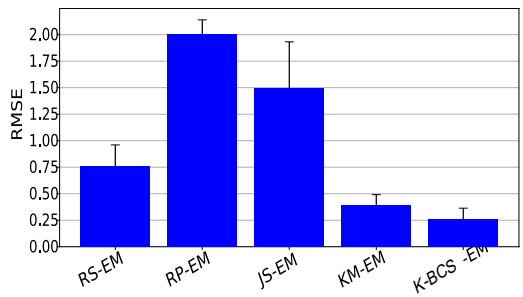
(a) Mode Masking under High Variance



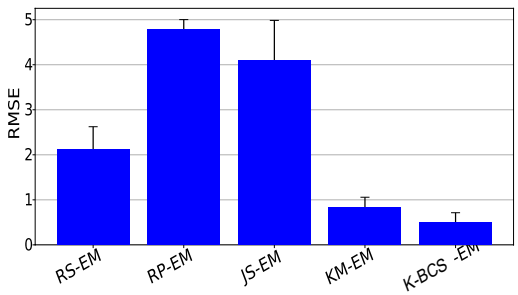
(b) Fitted histogram



(c) EM convergence



(d) Variance RMSE



(e) Mean RMSE

Fig. 5 Case 4 - average performance under mode_masking with unequal variances ($K = 3$).

RP-EM exhibited noticeably larger errors. The visible separation between K-BCS-EM and the remaining methods highlights its improved ability to resolve masked components.

The variance RMSE plot (Fig. 5e) revealed a similar pattern. K-BCS-EM achieved the smallest dispersion error, followed by KM-EM. The larger variance RMSE values for RS-EM, RP-EM, and JS-EM indicated difficulty in accurately estimating spread when one mode partially obscures another. The averaged EM convergence curves (Fig. 5c) demonstrated monotonic log-likelihood increase for all methods. The convergence behavior appears broadly similar; however, K-BCS-EM exhibited a smoother and slightly more stable ascent path, reflecting improved initial parameter placement in challenging masking conditions.

Finally, the averaged histogram with reconstructed PDFs (Fig. 5b) provided visual confirmation of density recovery performance. The K-BCS-EM density curve aligned closely with the empirical histogram across the central region and tails, while alternative methods showed mild distortions in peak height or spread. Although the differences are subtler than in the heavy-tail case, K-BCS-EM consistently offers the most faithful representation of the underlying mixture structure. Essentially, the figures confirm that under mode masking, where component separation is inherently difficult, the K-BCS-EM delivered superior estimation accuracy and stable convergence relative to conventional initialization strategies.

Case 5: near-duplicate components

This dataset shown in Fig. 6awas constructed such that two of the components had very similar means and variances, while the third component was more dispersed. The presence of near-duplicate components created a challenging identifiability problem, which can lead to label switching and unstable parameter estimates. This represents a classical ill-posed mixture scenario in which the EM algorithm becomes highly sensitive to the choice of initial parameters.

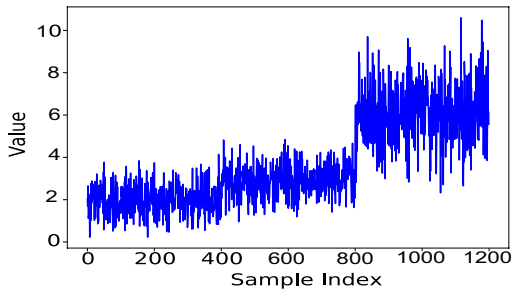
The mean RMSE comparison (Fig. 6e) showed that K-BCS-EM achieved the lowest estimation error among all competing strategies. KM-EM performed second best, while RS-EM, RP-EM, and JS-EM exhibited noticeably larger mean errors. The separation between K-BCS-EM and the remaining methods highlighted its improved ability to distinguish extremely close component means. A similar trend was observed in the variance RMSE plot (Fig. 6d). K-BCS-EM records the smallest dispersion error, followed closely by KM-EM. The higher variance RMSE values for RS-EM, RP-EM, and JS-EM indicated greater sensitivity to component proximity, particularly when dispersion parameters were difficult to separate.

The EM convergence curves (Fig. 6c) demonstrated monotonic log-likelihood ascent across all methods. Convergence speeds appeared broadly similar; however, K-BCS-EM showed a smoother and more stable ascent path, reflecting improved initial parameter placement in the presence of nearly indistinguishable components. The averaged histogram with reconstructed PDFs (Fig. 6b) provided visual confirmation of density recovery quality. The K-BCS-EM density curve aligned most closely with the empirical histogram across the central region, while alternative methods exhibited mild distortions in peak shape and spread. Although differences were subtle due to the inherent similarity of components, K-BCS-EM consistently provided the most faithful reconstruction of the underlying mixture structure.

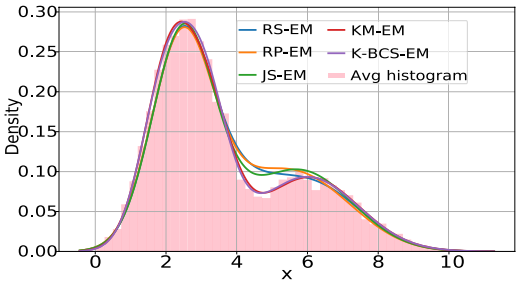
Case 6: asymmetric spacing with variance gradient

In this dataset shown in Fig. 7a, the component means were unevenly spaced, and the variances increased progressively across components. Such variance gradients introduced heteroscedasticity, which can lead the EM algorithm to incorrectly allocate observations when initialization does not adequately account for differences in dispersion. This scenario therefore tests the sensitivity of the initialization strategies to variance structure, rather than relying solely on mean separation.

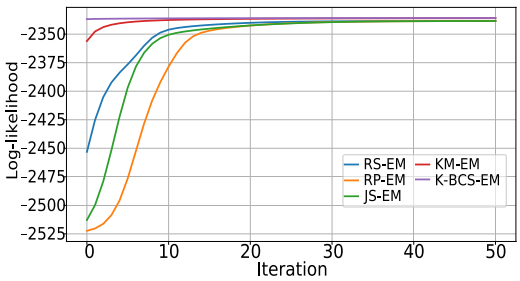
The mean RMSE comparison (Fig. 7e) showed that K-BCS-EM achieved the lowest mean estimation error among all competing methods. RS-EM performed comparatively well, while RP-EM, JS-EM, and particularly



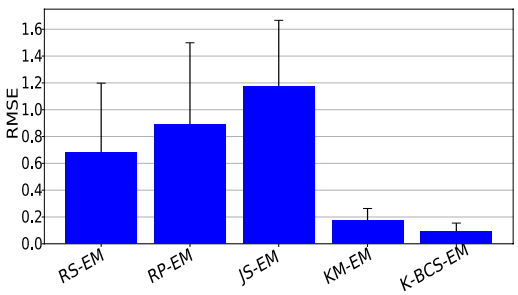
(a) Near-Duplicate Components



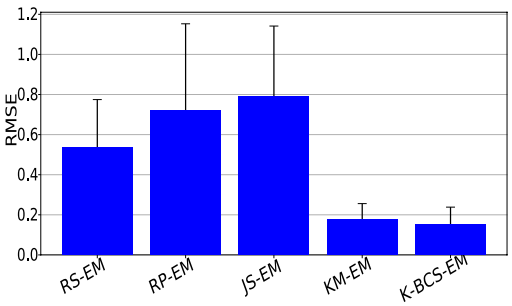
(b) Fitted histogram



(c) EM convergence



(d) Variance RMSE

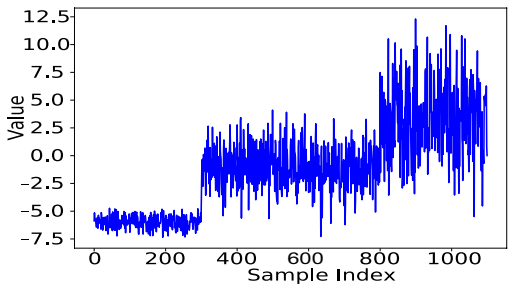


(e) Mean RMSE

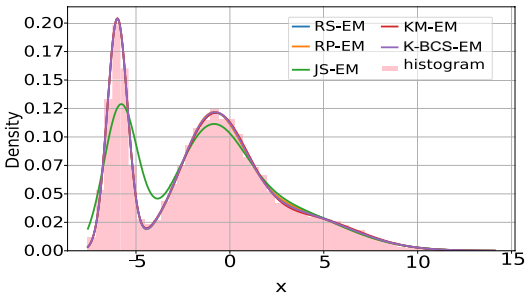
Fig. 6 Case 5 - average performance under near duplicate with unequal variances ($K = 3$).

KM-EM exhibited higher errors. The separation between K-BCS-EM and the remaining strategies highlights its robustness under progressively increasing component variances. The variance RMSE plot (Fig. 7d) revealed a similar pattern. K-BCS-EM attained the smallest dispersion error, followed by RS-EM. In contrast, JS-EM and KM-EM showed substantially larger variance errors, indicating difficulty in recovering heteroscedastic structures when initialization is not variance-sensitive.

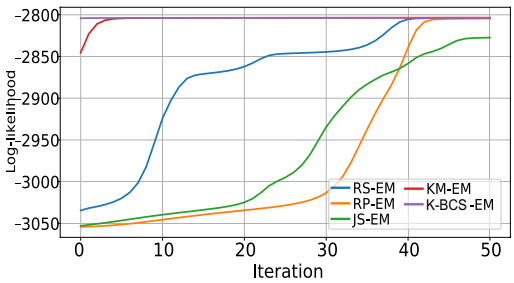
The averaged EM convergence curves (Fig. 7c) demonstrated monotonic log-likelihood ascent for all methods. Convergence speeds were broadly similar; however, K-BCS-EM displayed a smooth and stable ascent trajectory, reflecting improved initial placement across components with varying spreads. Finally, the averaged histogram with reconstructed PDFs (Fig. 7b) visually confirms density reconstruction performance. The K-BCS-EM curve aligned closely with the empirical histogram across both narrow and wide components, which captured the variance gradient effectively. Alternative methods showed mild distortions in peak width or tail behaviour, which were consistent with their higher variance RMSE values.



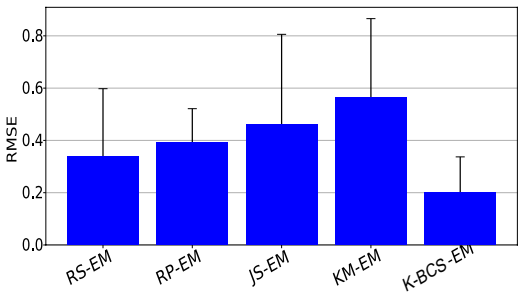
(a) Asymmetric Spacing with Variance Gradient



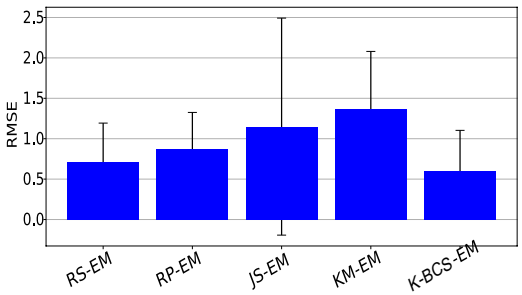
(b) Fitted histogram



(c) EM convergence



(d) Variance RMSE



(e) Mean RMSE

Fig. 7 Case 6 - average performance under variance gradient with unequal variances ($K = 3$).

Table 2 Runtime performance across datasets (mean ± std in seconds).

Method	Overlap	Heavy	Imbal.	Masking	Near-Dup.	Var-Grad.
RS-EM	0.0268 ± 0.0015	0.1236 ± 0.0147	0.0304 ± 0.0013	0.0882 ± 0.0057	0.0930 ± 0.0231	0.0836 ± 0.0074
RP-EM	0.0263 ± 0.0008	0.1245 ± 0.0129	0.0302 ± 0.0013	0.0874 ± 0.0059	0.0969 ± 0.0242	0.0802 ± 0.0068
JS-EM	0.0268 ± 0.0006	0.1193 ± 0.0165	0.0302 ± 0.0011	0.0945 ± 0.0134	0.0959 ± 0.0267	0.0805 ± 0.0090
KM-EM	0.0286 ± 0.0009	0.1214 ± 0.0226	0.0370 ± 0.0147	0.1018 ± 0.0128	0.0964 ± 0.0255	0.0804 ± 0.0123
K-BCS-EM	0.1097 ± 0.0027	0.6675 ± 0.0737	0.1361 ± 0.0037	0.4546 ± 0.0776	0.4185 ± 0.1028	0.3515 ± 0.0793

Timing performance analysis

The runtime comparison across all datasets is summarized in Table 2. A consistent pattern was observed across all scenarios: RS-EM, RP-EM, JS-EM, and KM-EM exhibited very similar computational costs, while K-BCS-EM required additional processing time.

For the overlap and imbalance datasets, all the conventional initialization methods completed within a very short time window (approximately 0.02–0.04 seconds), with only minor differences between them. KM-EM is slightly slower in some cases due to the clustering step, but the overall computational cost remained comparable across these methods.

As the datasets became more complex, particularly under the heavy-tail and mode masking conditions, the runtime of all the methods increased slightly. However, the relative differences between RS-EM, RP-EM, JS-EM, and KM-EM remained minimal, which indicated that their computational complexity scaled similarly with data difficulty.

In contrast, K-BCS-EM consistently incurred higher runtime across all scenarios. This increase was most pronounced in the heavy-tail and mode masking cases, where execution time rose significantly as compared to the other methods. This additional cost was attributed to the BCV-based multi-split segmentation step, which introduces a preprocessing phase prior to EM execution.

Despite this overhead, the increase in runtime remained predictable and stable across datasets, with relatively low variability. Importantly, the additional computational effort is incurred only once during initialization, after which the EM procedure proceeds in the same manner as the other methods.

These timing results highlighted a clear trade-off while conventional methods offer lower computational cost, K-BCS-EM introduces a modest but consistent increase in runtime in exchange for improved initialization quality. This trade-off becomes particularly relevant in more challenging scenarios, where the benefits of improved stability and accuracy outweigh the additional preprocessing time.

Performance under increasing number of components

The scalability of the proposed approach was evaluated by increasing the number of mixture components beyond $K = 3$, with results summarized in Fig. 8. Across all cases, a consistent trend emerges: K-BCS-EM maintained the lowest estimation errors and the most stable performance as model complexity increased.

For $K = 4$, Fig. 8(a) and (b) show that K-BCS-EM achieved the lowest mean and variance RMSE among all the methods. KM-EM followed as the next best performer, while RS-EM and JS-EM exhibited moderate errors with greater variability. RP-EM performed worst, indicating a clear sensitivity to increasing model complexity.

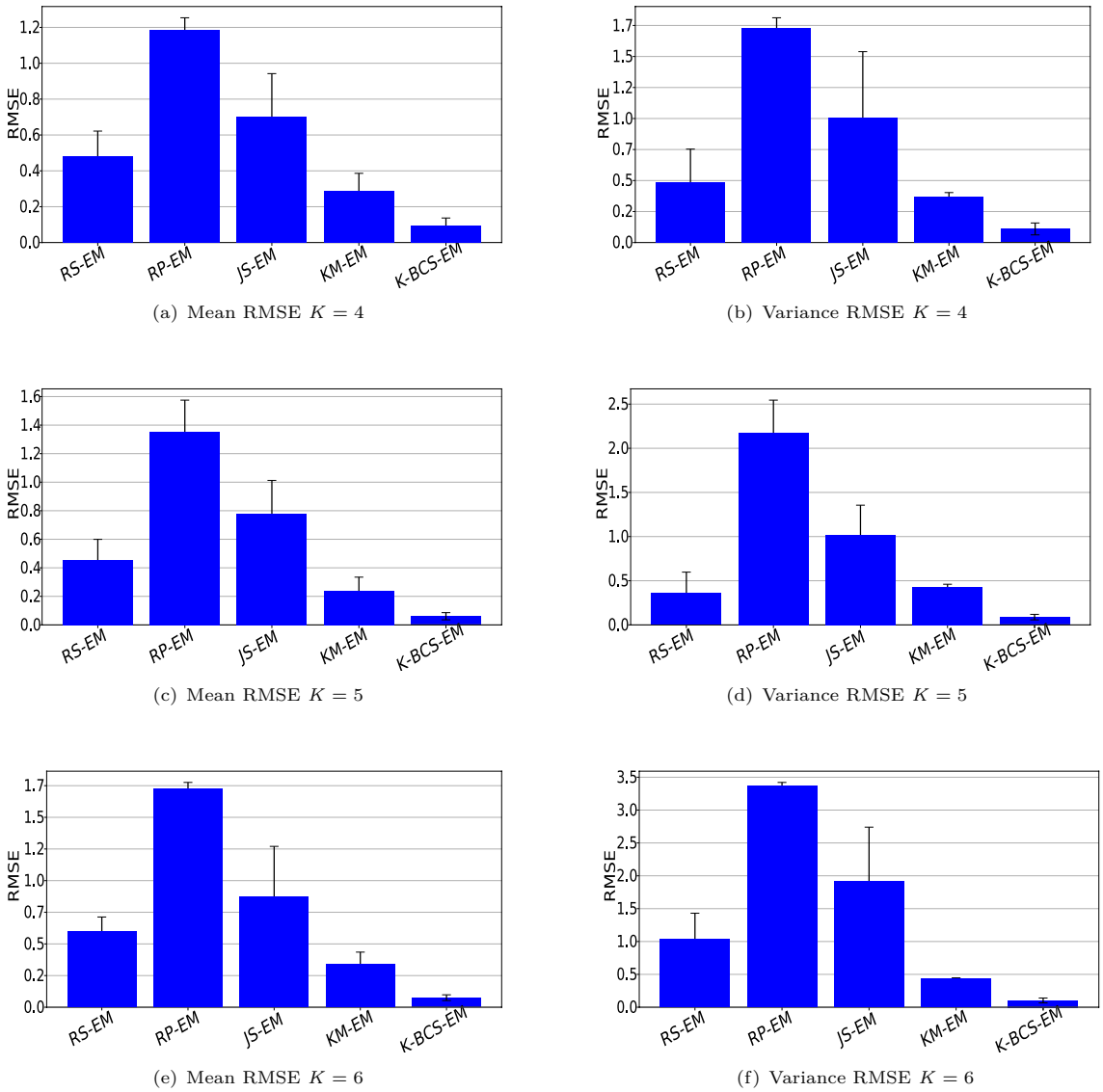
As the number of components was increased to $K = 5$, this performance gap became even more pronounced. From Fig. 8(c) and (d), K-BCS-EM continued to deliver the most accurate and stable estimates for both mean and variance. Although KM-EM remained relatively competitive, the performance of RS-EM and JS-EM deteriorated, and RP-EM showed consistently poor results. The separation between K-BCS-EM and the remaining methods became more evident, thus highlighting the benefit of structure-aware initialization.

For $K = 6$, the results in Fig. 8(e) and (f) demonstrated that K-BCS-EM not only preserved its advantage but also maintained stable error levels despite the increased complexity. While variance estimation became more challenging at this stage, K-BCS-EM remained highly competitive and continued to outperform RS-EM, JS-EM, and RP-EM. The KM-EM approach performed comparably in some cases, but showed signs of reduced stability relative to K-BCS-EM.

Essentially, the results clearly show that as the number of mixture components increases, K-BCS-EM consistently provides superior or highly competitive mean and variance recovery. Unlike conventional initialization methods whose performance degrades with increasing K, the proposed approach maintained both accuracy and stability. This highlights its strong scalability and robustness in more complex mixture modeling scenarios.

Table 3 evaluates the robustness of each initialisation strategy as the number of mixture components increases ($K = 4, 5, 6$). The results clearly demonstrate the superior scalability of KBCVS-EM.

Across all values of K, KBCVS-EM consistently achieves the lowest mean RMSE. Notably, its performance remains stable and even slightly improves as K increases (0.0804, 0.0787, and 0.0688 for $K = 4, 5, 6$, respectively), with relatively small standard deviations. This indicates that the BCV-based multi-split segmentation effectively preserves separation structure even in higher-dimensional partitioning.



(a) Mean RMSE $K = 4$ (b) Variance RMSE $K = 4$

(c) Mean RMSE $K = 5$ (d) Variance RMSE $K = 5$

(e) Mean RMSE $K = 6$ (f) Variance RMSE $K = 6$

Fig. 8 Performance under increasing number of components.

Table 3 Runtime performance results of different initialization methods and mixture components $K > 3$. Values are reported as mean ± standard deviation.

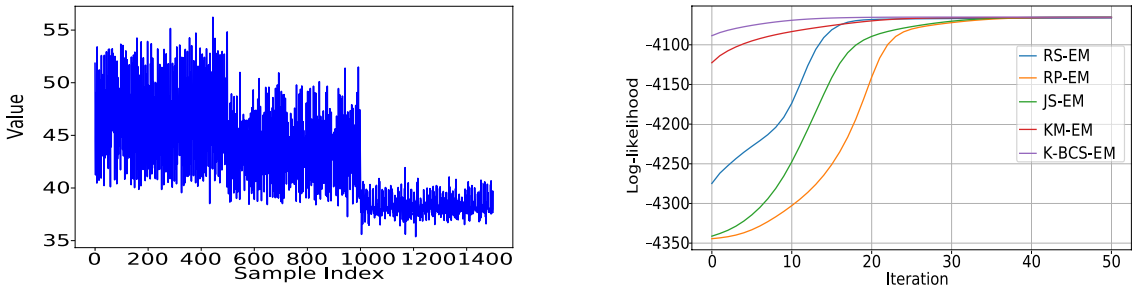
Method	**K = 4**	**K = 5**	**K = 6**
RS-EM	0.4132 ± 0.1689	0.3619 ± 0.1303	0.5600 ± 0.1273
RP-EM	1.2269 ± 0.0243	1.2686 ± 0.1499	1.7291 ± 0.0232
JS-EM	0.9172 ± 0.2093	0.8163 ± 0.2169	0.9987 ± 0.3595
KM-EM	0.2118 ± 0.0655	0.1944 ± 0.1274	0.4473 ± 0.1340
KBCVS-EM	**0.0804 ± 0.0329**	**0.0787 ± 0.0494**	**0.0688 ± 0.0224**

KM-EM performs second best overall, particularly for $K = 4$ and $K = 5$, but its error increases substantially at $K = 6$, suggesting reduced robustness as model complexity grows. RS-EM and JS-EM exhibit moderate to high errors with noticeable variability, while RP-EM performs worst across all values of K, showing limited scalability.

Overall, the results confirm that KBCVS-EM maintains strong estimation accuracy and stability as the number of mixture components increases, demonstrating effective scalability in more complex clustering regimes.

Table 4 Runtime performance results of different initialization methods and mixture components $K > 3$. Values are reported as variance ± standard deviation.

Method	K = 4	K = 5	K = 6
RS-EM	0.3119 ± 0.1179	0.2999 ± 0.1289	1.0958 ± 0.2616
RP-EM	1.6922 ± 0.0531	2.0595 ± 0.3602	3.3849 ± 0.0651
JS-EM	1.3280 ± 0.4509	1.0084 ± 0.4504	2.1183 ± 0.8383
KM-EM	0.3633 ± 0.0419	0.4230 ± 0.0297	**0.4101 ± 0.0414**
KBCVS-EM	**0.0950 ± 0.0445**	**0.0836 ± 0.0414**	0.4101 ± 0.0506



(a) CPU Utilisation Data (b) EM Convergence

Fig. 9 CPU utilisation dataset and corresponding EM convergence behaviour.

Table 4 presents the variance estimation performance as the number of mixture components increases beyond three. The results indicate that KBCVS-EM consistently achieves the lowest σ^2 RMSE for $K = 4$ and $K = 5$, demonstrating strong robustness in recovering dispersion parameters under increasing model complexity.

For $K = 6$, the variance error of KBCVS-EM increases to 0.4101 ± 0.0506, aligning closely with KM-EM (0.4101 ± 0.0414). Nevertheless, both methods substantially outperform RS-EM, JS-EM, and RP-EM, whose variance errors increase markedly as K grows. In particular, RP-EM exhibits severe degradation in performance, indicating poor scalability in heteroscedastic multi-component settings.

Overall, the findings confirm that KBCVS-EM maintains superior variance recovery and competitive stability as the number of components increases, highlighting its scalability and robustness in more complex mixture configurations.

Real-world dataset evaluation

To further assess the robustness and practical applicability of the proposed K-BCS-EM approach, experiments were conducted on real-world datasets spanning cloud computing, wireless communication, and speech processing. Each dataset presented unique structural characteristics, which allowed for a comprehensive evaluation under different application areas.

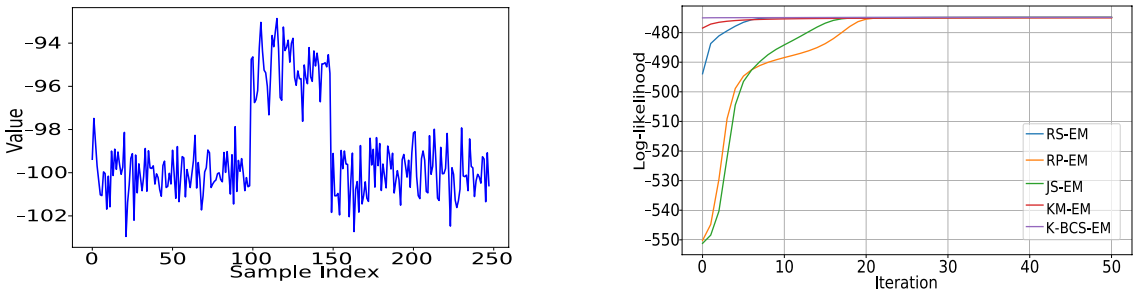
EC2 CPU utilisation

The CPU utilisation dataset obtained from[26] presents a challenging real-world scenario characterized by overlapping behaviours and fluctuating workload dynamics. As shown in Fig. 9, the underlying distribution exhibits regions of ambiguity where component separation is not immediately obvious.

From Table 5, K-BCS-EM clearly outperformed all competing methods in both mean and variance estimation. The reduction in variance RMSE was particularly significant, indicating that the method is able to accurately capture dispersion patterns that conventional approaches fail to model effectively. In contrast, KM-EM showed substantial error in variance estimation, hence suggesting that distance-based clustering is insufficient in handling overlapping and heteroscedastic structures.

Table 5 CPU utilisation dataset: mean and variance RMSE with runtime performance.

Method	μ_R	σ_R^2	Time (s)
RS-EM	0.9242	1.5234	0.0458
RP-EM	0.9420	1.4736	0.0865
JS-EM	0.9014	1.2725	0.0474
KM-EM	1.3422	2.9481	0.0427
K-BCS-EM	0.6498	0.9331	0.1755



(a) OFDM Signal Data (b) EM Convergence

Fig. 10 OFDM dataset and corresponding EM convergence behaviour.

Table 6 OFDM dataset: mean and variance RMSE with runtime performance.

Method	μ_R	σ_R^2	Time (s)
RS-EM	0.0942	0.2430	0.0217
RP-EM	1.0717	1.2368	0.0208
JS-EM	0.1199	0.2701	0.0218
KM-EM	0.3108	0.2235	0.0214
K-BCS-EM	0.0939	0.2499	0.0441

The convergence curves further reinforce this observation. While all methods exhibited monotonic likelihood ascent, K-BCS-EM followed a more stable trajectory, indicating better initial alignment with the underlying data structure. The improved initialization reduced ambiguity during early EM iterations, which led to more reliable parameter updates.

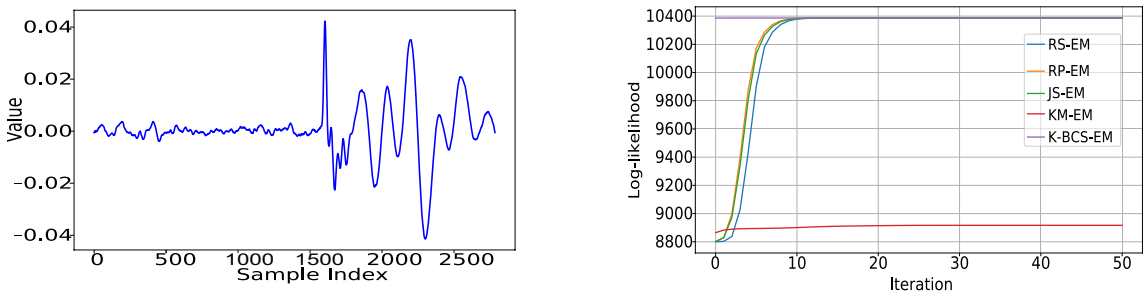
Although K-BCS-EM incurred a higher runtime due to the segmentation step, the gain in estimation accuracy and stability makes this trade-off highly favorable, particularly for complex time-series data such as CPU utilisation.

OFDM signal

The OFDM dataset obtained from[27] exhibits a more structured distribution compared to the CPU dataset, with clearer separation between components, as illustrated in Fig. 10ba. This relatively well-defined structure allows most initialization methods to perform reasonably well.

As shown in Table 6, K-BCS-EM achieved the lowest mean RMSE, further indicating superior accuracy in recovering the central tendency of the signal. While KM-EM attains the lowest variance RMSE, its performance in mean estimation was less consistent. In contrast, K-BCS-EM provided a more balanced performance across both metrics, maintaining strong accuracy without significant trade-offs.

The convergence behaviour observed in Fig. 10bafurther supported these findings. All methods exhibited similar convergence trends, reflecting the relative simplicity of the dataset. However, K-BCS-EM maintained a smooth and stable convergence trajectory, thus suggesting that its initialization remains effective even when the data structure is less challenging.



(a) Voice Signal Data (b) EM Convergence

Fig. 11 Voice dataset and corresponding EM convergence behaviour.

Table 7 Voice dataset: mean and variance RMSE with timing performance.

Method	μ_R	σ_R^2	Time (s)
RS-EM	0.0001	0	0.0520
RP-EM	0.0001	0	0.0375
JS-EM	0.0001	0	0.0401
KM-EM	0.0229	0	0.0403
K-BCS-EM	0.0001	0	0.2123

In terms of computational cost, Table 6 shows that all methods had comparable run times with K-BCS-EM incurring only a slight increase due to the segmentation step. This additional cost remains minimal and does not significantly impact practical applicability.

Essentially, the results indicate that while conventional methods can perform adequately in well-structured datasets, K-BCS-EM offers more consistent and reliable estimation across both mean and variance, which reinforces its robustness across different data conditions.

Human and AI generated voice signal

The voice dataset as shown in Fig. 11baobtained from[28] represents an audio-based application area where signal modeling may be required to improve auditory performance. The distribution exhibits clear distinction between noise and signal zones, thus making the EM algorithm a suitable approach for modeling purposes.

Table 7 presents the results where nearly all methods achieved identical performance in both mean and variance estimation. The EM algorithm consistently converged to the same solution which indicates that initialization plays a minimal role in this setting. The only exception was the KM-EM approach which showed degraded performance both the likelihood and mean-RMSE performance which suggests sensitivity to its clustering-based initialization. K-BCS-EM performed on par with the best methods in terms of accuracy which demonstrates that it does not introduce instability.

Although K-BCS-EM incurred a higher computational cost, this does not translate into additional accuracy gains in this particular dataset. However, its consistent performance confirms its robustness across both simple and complex scenarios. In summary, the real-world dataset results demonstrate that K-BCS-EM provides significant advantages in complex datasets while remaining competitive in simpler ones. Its ability to adapt to varying data structures highlights its effectiveness as a general-purpose EM initialization strategy.

Conclusion

This study has introduced K-BCS-EM, a discriminant-guided multi-split segmentation initialization approach designed to improve parameter estimation and convergence stability in the expectation maximization (EM) algorithm. The proposed method was evaluated against widely used initialization strategies (RS-EM, RP-EM, JS-EM, and KM-EM) within a comprehensive Monte Carlo framework spanning challenging synthetic and real-world datasets, as well as scenarios involving increasing numbers of mixture components ($K > 3$), which illustrates its applicability across a wide range of conditions.

Across nearly all experimental settings, K-BCS-EM consistently achieved the lowest mean and variance RMSE values, which demonstrates improved accuracy in both location and dispersion estimation. These improvements were particularly evident in more challenging scenarios, including overlapping components, heavy-tailed distributions, and imbalanced mixtures, which illustrates the limitations of random and partition-based initialization strategies that frequently exhibited instability and large estimation errors. Although likelihood-based performance measures appeared comparable across methods, a closer examination at the parameter level revealed clear differences in estimation accuracy, which highlights the importance of robust initialization beyond likelihood convergence alone.

The scalability experiments further showed that K-BCS-EM maintains stable performance as the number of mixture components increases. In contrast to competing methods whose errors tend to grow with model complexity, K-BCS-EM preserved low and consistent RMSE values across $K = 4$, $K = 5$, and $K = 6$, which illustrates its robustness in larger partitioning settings.

While K-BCS-EM introduces additional computational cost due to the BCV-guided multi-split preprocessing step, the increase in runtime remains modest in absolute terms, which is supported by the substantial gains in robustness, stability, and estimation accuracy. In practical applications where reliable clustering and precise parameter recovery are essential, this trade-off is well justified.

The findings demonstrate that BCV-driven segmentation provides a principled and effective foundation for EM initialization. K-BCS-EM delivers improved robustness under challenging mixture configurations, enhanced scalability, and consistent convergence behaviour, which illustrates its suitability for reliable parameter estimation in complex mixture modeling tasks.

Acknowledgements The authors would like to appreciate the Council for Scientific and Industrial Research (CSIR), Pretoria, South Africa.

Author contributions I. L. M: Conceptualization, Methodology, Writing-Original draft preparation, Software. A.J.O: Methodology, Software, Writing-Original draft preparation, Writing- Reviewing and Editing, Project administration, Supervision. A.A: Validation, Funding, Writing- Reviewing and Editing, Supervision. A.K: Writing- Reviewing and Editing, Supervision.

Funding This research was funded by the Council for Scientific and Industrial Research (CSIR) under Grant No. 05400 054AT KR7EDNP.

Data availability All data generated or analyzed during this study are included in this published article.

Declarations

Competing interests The authors declare no competing interests.

Ethical approval We confirm that the manuscript has been read and approved by all named authors.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material

derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

References

1. Ng, S.K., Krishnan, T. & McLachlan, G.J. In: Gentle, J.E., Härdle, W.K., Mori, Y. (eds.) The EM Algorithm, pp. 139–172. Springer, Berlin, Heidelberg https://doi.org/10.1007/978-3-642-21551-3_6 (2012).
2. Akinin, M.V., Akinina, A.V., Sokolov, A.V. & Tarasov, A.S. Application of em algorithm in problems of pattern recognition on satellite images. In: 2017 6th Mediterranean Conference on Embedded Computing (MECO), pp. 1–4 https://doi.org/10.1109/meco.2017.7977190. (2017).
3. Kozick, R., Blum, R. & Sadler, B. Signal processing in non-gaussian noise using mixture distributions and the em algorithm. In: Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers (Cat. No. 97CB36136), vol. 1, pp. 438–442 https://doi.org/10.1109/ACSSC.1997.680365 . (1997).
4. Ma, M., Fu, L., Zhai, Z. & Sun, R..-B. Transformer based kalman filter with em algorithm for time series prediction and anomaly detection of complex systems. *Measurement* **229**, 114378. https://doi.org/10.1016/j.measurement.2024.114378 (2024).
5. Wang, L.-L., Gui, J.-S., Deng, X.-H., Zeng, F. & Kuang, Z.-F. Routing algorithm based on vehicle position analysis for internet of vehicles. *IEEE Inter. Things J.* **7**(12), 11701–11712. https://doi.org/10.1109/jiot.2020.2999469 (2020).
6. Moon, T. K. The expectation-maximization algorithm. *IEEE Signal Process. Magz.* **13**(6), 47–60. https://doi.org/10.1109/79.543975 (1996).
7. Noor, S., Tajik, O. & Golzar, J. Simple random sampling. International Journal of Education & Language Studies 1(2), 78–82 https://doi.org/10.22034/ijels.2022.162982 (2022).
8. Mahmud, M. S., Huang, J. Z., Salloum, S., Emara, T. Z. & Sadatdiynov, K. A survey of data partitioning and sampling methods to support big data analysis. *Big Data Min. Anal.* **3**(2), 85–101. https://doi.org/10.26599/bdma.2019.9020015 (2020).
9. Keros, A.D., Divakaran, D. & Subr, K. Jittering samples using a kd-tree stratification. arXiv preprint arXiv:2002.07002 https://doi.org/10.48550/arXiv.2002.07002 (2020).
10. Brejna, B., Pietranik, M. & Kozierkiewicz, A. The new k-means initialization method. In: International Conference on Computational Collective Intelligence, pp. 372–381 https://doi.org/10.1007/978-3-031-70816-9_29. (2024).
11. Malafaia, D., Vieira, J. & Tomé, A. Adaptive threshold spectrum sensing based on expectation maximization algorithm. *Phys. Commun.* **21**, 60–69. https://doi.org/10.1016/j.phycom.2016.10.004 (2016).
12. Garriga, J., Palmer, J. R., Oltra, A. & Bartumeus, F. Expectation-maximization binary clustering for behavioural annotation. *PLoS One* **11**(3), 0151984. https://doi.org/10.1371/journal.pone.0151984 (2016).
13. Moreira, M. W. et al. Biomedical data analytics in mobile-health environments for high-risk pregnancy outcome prediction. *J. Ambient Intell. Humanized Comput.* **10**(10), 4121–4134. https://doi.org/10.1007/s12652-019-01230-4 (2019).
14. Nalepa, J. et al. Evaluating algorithms for anomaly detection in satellite telemetry data. *Acta Astronautica* **198**, 689–701. https://doi.org/10.1016/j.actaastro.2022.06.026 (2022).
15. Olken, F. & Rotem, D. Random sampling from databases: a survey. *Statis. Comput.* **5**(1), 25–42. https://doi.org/10.1007/bf00140664 (1995).
16. Anupama, C. & Lakshmi, C. A comprehensive review on data partitioning and sampling techniques for processing big data. In: 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), pp. 1–6 https://doi.org/10.1109/icpects56089.2022.10047766. (2022).
17. Tourabaly, J. & Osseiran, A. A jittered-sampling correction technique for adcs. In: 4th IEEE International Symposium on Electronic Design, Test and Applications (delta 2008), pp. 249–252 https://doi.org/10.1109/delta.2008.79 . (2008).
18. Harris, S. & De Amorim, R. C. An extensive empirical comparison of k-means initialization algorithms. *IEEE Access* **10**, 58752–58768. https://doi.org/10.1109/ACCESS.2022.3175487 (2022).
19. McLachlan, G.J. & Peel, D. Finite Mixture Models. John Wiley & Sons, ??? https://doi.org/10.1146/annurev-statistics-031017-100325 (2004).
20. Comaniciu, D. & Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 603–619. https://doi.org/10.1109/34.1000236 (2002).
21. Przyborowski, M. & Ślęzak, D. Approximation of the expectation-maximization algorithm for gaussian mixture models on big data. In: 2022 IEEE International Conference on Big Data (Big Data), pp. 6256–6260 https://doi.org/10.1109/bigdata55660.2022.10020450 . (2022).

22. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076 (1979).
23. Shireman, E., Steinley, D. & Brusco, M. J. Examining the effect of initialization strategies on the performance of gaussian mixture modeling. *Behav. Res. Methods* **49**(1), 282–293. https://doi.org/10.3758/s13428-015-0647-4 (2017).
24. Gutiérrez, I., Ramírez, S. & Jofré, L. Efficient em estimation for the pogit model via polya-gamma augmentation. *Entropy* **28**(2), 207. https://doi.org/10.3390/e28020207 (2026).
25. Irbäck, A. & Mohanty, S. Profasi: a monte carlo simulation package for protein folding and aggregation. *J. Comput. Chem.* **27**(13), 1548–1555. https://doi.org/10.1002/jcc.20452 (2006).
26. Numenta, N. *Numenta anomaly benchmark* https://doi.org/10.1109/ICMLA.2015.141 (2015).
27. Onumanyi, A. Dataset for testing the performance of the particle swarm optimization based recursive one-sided hypothesis testing (pso-roht) algorithm. Mendeley Data 1, (2018).
28. Azis, H., Rismayanti, N., Abdullah, M.B. & Ismail, S.B. Speech Dataset of Human and AI-Generated Voices. Kaggle https://doi.org/10.34740/KAGGLE/DSV/10957363 . https://www.kaggle.com/dsv/10957363 (2025).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Ipeleng Labius Machele[1,2] · Adeiza James Onumanyi[2] · Adnan M. Abu-Mahfouz[2] · Anish Kurien[1]

✉ Ipeleng Labius Machele
LMachele@csir.co.za

[1] Dept. of Electrical Engineering, Faculty of Engineering and the Built Environment, Tshwane University of Technology (TUT), Staatsartillerie Rd, Pretoria 0183, Gauteng, South Africa

[2] Next Generation Enterprises and Institutions, Council for Scientific and Industrial Research (CSIR), Meiring Naudé Rd, Pretoria 0001, Gauteng, South Africa