A Non-Parametric Bootstrap for Spectral Clustering

Liam Welsh∗,†
Department of Statistical Sciences,
University of Toronto

Phillip Shreeves†
Department of Statistics and Actuarial Science,
Simon Fraser University

March 22, 2024

Abstract

Finite mixture modelling is a popular method in the field of clustering and is beneficial largely due to its soft cluster membership probabilities. A common method for fitting finite mixture models is to employ spectral clustering, which can utilize the expectation-maximization (EM) algorithm. However, the EM algorithm falls victim to a number of issues, including convergence to sub-optimal solutions. We address this issue by developing two novel algorithms that incorporate the spectral decomposition of the data matrix and a non-parametric bootstrap sampling scheme. Simulations display the validity of our algorithms and demonstrate not only their flexibility, but also their computational efficiency and ability to avoid poor solutions when compared to other clustering algorithms for estimating finite mixture models. Our techniques are more consistent in their convergence when compared to other bootstrapped algorithms that fit finite mixture models.

Keywords: Spectral Clustering, EM Algorithm, Bootstrapping, Mixture Models

∗Corresponding author: liam.welsh@mail.utoronto.ca
†Authors contributed equally
1 Introduction

As the world becomes more saturated with big data, the ability to produce interpretable statistical models from high dimensional data is a task of great significance. In the field of clustering, unsupervised learning algorithms can be used to find patterns or groups within data and construct models in these high dimensional settings (see Bouveyron and Brunet-Saumard (2014); Celebi and Aydin (2016)). These models can take the form of a finite mixture of distributions, a probabilistic model that can be used to represent sub-populations within data McLachlan et al. (2019). A main issue with constructing models, such as finite mixture models (FMMs) for high dimensional data is the fitting of these models and parameter estimations can be incredibly slow, converge to local solutions, or lack interpretation. Even in low dimensional settings, clustering algorithms may be fraught with issues, including converging to poor local maxima solutions or overfitting to the data available which results in the optimal log-likelihood but having overconfident group membership estimates. We demonstrate this in Section 2. Andrews (2018) has demonstrated and addressed the issue of overfitting and sub-optimal convergence in FMMs through the use of a non-parametric bootstrap. FMMs are of particular interest as they provide a probabilistic result for group memberships, allowing for valuable insight into analysis and model interpretation; see McLachlan et al. (2019). However, fitting a FMM in a high dimensional setting is computationally expensive, as it requires repeated inversions of $p \times p$ covariance matrices, where $p$ is equal to the number of variables. Further, the bootstrapping procedure Andrews (2018) employs is itself computationally non-trivial. To address the issues of overfitting and sub-optimal convergence in high dimensional FMMs, we propose the combination of a common dimensionality reduction clustering utilized in spectral clustering and the non-parametric bootstrap. Spectral clustering is a versatile clustering technique that is useful for high dimensional settings; see Vempala and Wang (2004) and Von Luxburg (2007). Despite its versatility, it is still prone to the aforementioned issues.

We develop two novel algorithms and show that our proposed algorithms not only address the issue of overfitting and convergence to sub-optimal solutions, but are also significantly faster than the current bootstrapped approaches. The remainder of this paper is organized as follows. A motivating example is introduced in Section 2. Section 3 follows with the relevant background information required to understand the methodological contribution of this research, which is presented in Section 4. Our work includes two bootstrap-augmented algorithms for spectral clustering and a convergence criterion. Algorithm performances are then analyzed using simulated and real-world data sets in Section 5 and compared against prior algorithms. Concluding remarks are provided in Section 6.
(a) First two variables of mirror data. This is two of the 150 total variables available. However, pairwise plots of any two variables will look similar.

(b) Remaining two components of mirror data after spectral transformation \( \hat{Y} = X V_2' \) where \( V_2 \) is the first two columns of the unitary matrix of \( X \) from the SVD \( X = U \Sigma V' \).

Figure 1: Original (left) and transformed (right) mirror data.

## 2 Motivating Example

We extend the motivating example of Andrews (2018) from a two-dimensional setting to a high dimensional setting. Specifically, we construct a sample of 500 data points with 150 variables centered at \((7, \ldots, 7)\) with standard normal noise. We then take a copy of this data reflecting it across the hyperplane \( X = -Y \), where \( X, Y \in \mathbb{R}^{150} \), providing us with two distinct groups. Finally, one observation is created at \((0, \ldots, 0)\) so that it is equidistant from each group. The first two variables of this data set are plotted in Figure 1a. While spectral clustering is well suited for noisy data, this extremely stylized example with well separated groups demonstrates the issue at hand.

Intuition leads one to believe that the probability of the centre point belonging in either cluster would be 50% in a two group solution. However, this is not found to be the case with many clustering algorithms. The \( k \)-means algorithm simply labels the point as belonging to one group or another as it is a hard clustering algorithm that provides no statistical reference. Conversely, the benchmark expectation-maximization (EM) algorithm with unconstrained covariances heavily overfits and finds the probability of the centre point belonging to one group to be over 99.999%, and practically 0% to belong the other. A constrained or equal covariance assumption in a high dimensional setting becomes increasingly strong, and difficult to verify given real data.

Andrews (2018) has demonstrated that finding a two group solution in the mirror data is
the result of the EM algorithm converging to a global maxima. While this may be ideal with respect to the MLE, it is flawed from an inference perspective. This issue remains in the solution when a spectral transformation is used on the data, see Figure 1b. In particular, we employ the decomposition technique used in Algorithm 1 by Löffler et al. (2021). Using this methodology, we find that it estimates the centre point’s group membership probabilities to be \((1, 0)\), counter-intuitive to the point’s relative location.

Andrews (2018) has shown that applying a non-parametric bootstrap can address this issue of overfitting in the EM algorithm. In high dimensions, the method of Andrews (2018) is extremely slow. Shreeves and Andrews (2019) further expand upon this work by using the non-parametric bootstrap to augment the alternating expectation-conditional maximization (AECM) algorithm. This uses the assumption that the observations arise from a factor analysis model and was developed in a mixture model framework by Ghahramani et al. (1996); McLachlan and Peel (2000). This work is further outlined in Section 3.4. While this removes the need for the direct computation of multiple \(p \times p\) inversions at each bootstrap iteration, it is still required once per iteration to find the likelihood of the averaged parameter space. Additionally a large number of matrix multiplications are needed to solve for the latent space, slowing the algorithm further. Our proposed methodology greatly reduces the space on which to perform matrix calculations, achieving significant gains in computational efficiency while converging to an appropriate solution.

3 Background

3.1 Finite Mixture Models

FMMs (McLachlan et al. (2019)) are commonly used for clustering to provide a statistical representation of an underlying group structure and have been used in a variety of applications; see (McNicholas, 2016; Torres-Carrasquillo et al., 2002; Yu, 2012; Georgiades et al., 2022). A clear benefit of FMMs is that it is a true statistical model. Once fitted, an FMM can provide a probabilistic result when grouping new data; a feature not available with hard clustering techniques such as the \(k\)-means methods and hierarchical trees. As such, clustering using an FMM is known as a soft clustering technique. Mathematically, FMMs are a convex combination of \(G\)-many statistical densities, and can be expressed as

\[
f(x_i) = \sum_{g=1}^{G} \pi_g f_g(x_i | \Theta_g). \tag{1}
\]

Here, \(x_i\) is a \(p\) dimensional observation from a data matrix \(X\) that is \(n \times p\), \(G\) is the number of underlying groups in the model, \(\pi_g\) is the mixing proportion of group \(g\) (with the constraint
\[ \sum_{g=1}^{G} \pi_g = 1 \], and \( f_g \) is the underlying distribution of the \( g \)th group component with parameters \( \Theta_g \). In this work, we direct our focus to mixtures of multivariate normal distributions, which are represented as

\[ f(x_i) = \sum_{g=1}^{G} \pi_g \phi(x_i \mid \mu_g, \Sigma_g), \quad (2) \]

where \( \mu_g \) and \( \Sigma_g \) are the mean vector and covariance matrix of the \( g \)-th group, respectively. FMMs are well studied for the multivariate normal distribution and other families of distributions (Andrews et al., 2011; Franczak et al., 2013; Punzo and McNicholas, 2016). Let \( z_i = (z_{i1}, \ldots, z_{iG}) \) be the \( G \)-dimensional vector of group membership probabilities for \( x_i \), such that \( \sum_{g=1}^{G} z_{ig} = 1 \). Then \( \pi_g \) can be viewed as the a priori probability of observation \( i \) belonging to group \( g \), with an a posteriori probability of

\[ P(z_{ig} = 1 \mid x_i) = \frac{\pi_g \phi(x_i \mid \mu_g, \Sigma_g)}{\sum_{j=1}^{G} \pi_j \phi(x_i \mid \mu_j, \Sigma_j)}. \quad (3) \]

This quantity can be estimated as

\[ \hat{z}_{ig} = \frac{\hat{\pi}_g \phi(x_i \mid \hat{\mu}_g, \hat{\Sigma}_g)}{\sum_{j=1}^{G} \hat{\pi}_j \phi(x_i \mid \hat{\mu}_j, \hat{\Sigma}_j)}, \quad (4) \]

where \( \hat{\mu}_g, \hat{\Sigma}_g, \) and \( \hat{\pi}_g \) for \( g = 1, \ldots, G \) are the maximum likelihood estimates determined from the model likelihood

\[ L(\Theta) = \prod_{i=1}^{n} \sum_{g=1}^{G} \pi_g \phi(x_i \mid \mu_g, \Sigma_g). \quad (5) \]

The conventional multivariate normal finite mixture model requires

\[ \begin{align*}
\text{mixing proportions} \quad &+ \quad \text{group means} \quad + \quad \text{group covariances} \\
(G-1) \quad &+ \quad Gp \quad + \quad \frac{Gp(p+1)}{2}
\end{align*} \quad (6) \]

free parameters, with the parameter contributions displayed below Equation (6). The largest contributor arises from the estimation of the group covariance matrices, which is a natural target for parsimony McNicholas (2016). A method to do so is discussed in Section 3.4. However, multiple other techniques have been used to address this; see Banfield and Raftery (1993); Celeux and Govaert (1995); Tipping and Bishop (1999). The most common way to estimate these parameters is in an iterative style, alternating between updating the maximum likelihood estimates and updating the posterior probabilities Dempster et al. (1977). This is known as the expectation-maximization (EM) algorithm and is further detailed in the following section.
3.2 EM Algorithm

The EM algorithm (Dempster et al., 1977; McLachlan and Krishnan, 2007) is an extremely common method to estimate Gaussian FMMs. It iterates between two steps; the expectation (E) step and the maximization (M) step. The E-step calculates the estimated expected value of $Z_{ig}$ for all $i = 1, ..., n$ and $g = 1, ..., G$, found in (4), while the M-step calculates the maximum likelihood estimates of $\hat{\pi}_g$, $\hat{\mu}_g$, and $\hat{\Sigma}_g$. Updates for the maximum likelihood estimates are given as

$$
\hat{\pi}_g = \frac{n_g}{n}, \quad \hat{\mu}_g = \frac{1}{n_g} \sum_{i=1}^{n} \hat{z}_{ig} x_i, \quad \text{and} \quad \hat{\Sigma}_g = \frac{1}{n_g} \sum_{i=1}^{n} \hat{z}_{ig} (x_i - \hat{\mu}_g)(x_i - \hat{\mu}_g)',
$$

(7)

An algorithm summary (McNicholas, 2016) is provided in Algorithm 1. A common convergence criterion used for the EM algorithm is 'lack of progress', which compares the updated version of the log-likelihood, denoted $\ell = \log(L(\Theta))$, against its previous iteration. The lack of progress criterion is given as

$$
| \ell^{(k)} - \ell^{(k-1)} | < \epsilon
$$

(8)

where $\epsilon > 0$. However, we note that there are other criteria available. Research regarding the ‘optimal convergence criterion’ is outside the scope of this manuscript and we refer readers to McNicholas (2016) for further details. Further evaluation of model performance is often calculated using the Bayesian information criterion (BIC), such that

$$
\text{BIC} = 2 \ell(\Theta) - \rho \cdot \log(n),
$$

(9)

where $\rho$ is the number of free parameters in the model. BIC includes a penalization term for number of parameters relative to the number of observations.

| Algorithm 1: EM algorithm for finite GMM estimation |
|---------------------------------------------------|
| **Input:** X                                     |
| 1. Initialize $\hat{z}$ using either k-means or discrete uniform randomization; |
| 2. while not converged do                         |
| 3. for $g = 1, \ldots, G$ do                      |
| 4. Update the MLEs using Equations in (7);       |
| 5. Compute $\hat{z}_g$ using the new MLE estimates; |
| 6. Check for convergence;                        |
| 7. end                                           |
| **Output:** $\hat{\pi}_g$, $\hat{\mu}_g$, and $\hat{\Sigma}_g$ for $g = 1, \ldots, G$ |

3.3 Bootstrapping and the BootEM Algorithm

The bootstrap procedure, introduced in Efron (1981) and Efron (1982), is a resampling technique that enables estimates of parameter standard errors that otherwise may not be possible to calculate. Specifically, for a parameter $\theta$ with estimate $\hat{\theta}$, we can calculate new estimates of $\theta$ by
sampling with replacement from the data \( X \), creating a (non-parametric) bootstrap sample. Repeating this process \( M \) times, we can compute the standard deviation of the parameter estimate as 
\[
\hat{\sigma}_{\theta} = \sqrt{\frac{1}{M-1} \sum_{j=1}^{M} (\hat{\theta}_j - \hat{\theta}^*)^2}
\]
where \( \hat{\theta}_j \) corresponds to the estimate from the \( j \)th bootstrap sample and \( \hat{\theta}^* = \frac{1}{M} \sum_{j=1}^{M} \hat{\theta}_j \) is the averaged parameter estimate. Further, the non-parametric bootstrap sampling technique allows for built-in cross validation, as the non-sampled data from the \( j \)th bootstrap sample provides us with out-of-bag (OOB) estimates for the \( j \)th model.

Andrews (2018) notes that the conventional EM algorithm falls victim to a number of issues. The first of which is that it has the ability to converge to a local maximum, as opposed to the global maximum, in the log-likelihood resulting in an underfit. The EM may also overfit to the data, as was demonstrated in Section 2.

These issues can be addressed by evaluating EM algorithms over repeated bootstrap samples of the original data matrix and estimating the OOB group memberships for observations left out of the sample. Additionally, the parameter estimates are averaged over the multiple bootstrap iterations, preventing the convergence to a local maximum. It should be noted that these averaged parameter estimates are not maximum likelihood estimates; although the final likelihoods are very similar. As a result, conventional lack of progress criterion cannot be used to determine the stopping point in the algorithm. Instead, it is assumed that the algorithm has converged once the last \( M \) log-likelihood values return a null from result from the Durbin-Watson test; a test for auto-correlation over a set of indices. When this is achieved, it is assumed that the algorithm has converged and deviations in the likelihood are due to the randomness from the bootstrap sampling. We denote this algorithm as BootEM, and it is presented below in Algorithm 2.

**Algorithm 2:** Bootstrapped EM algorithm for GMM estimation

| Input: \( X \) |
|---|
| 1 Initialize \( \hat{z}^{(0)} \) using either k-means or randomization; |
| 2 Set \( k = 1 \); |
| 3 while not converged do |
| 4 Sample with replacement from \( X \), denoting the sample as \( X^{(k)} \) and take the corresponding cluster memberships from \( \hat{z}^{(k-1)} \); |
| 5 Perform the EM Algorithm (see Algorithm 1) on \( X^{(k)} \) using \( \hat{z}^{(k-1)} \) as cluster initializations to compute parameter estimates; |
| 6 Compute group memberships for \( X \) using parameter estimates; |
| 7 Check for convergence; |
| 8 end |
| Output: \( \hat{\theta}^* \) |

Andrews (2018) demonstrates that this methodology addresses the issue of converging to sub-optimal solutions corresponding. However due to the bootstrapping procedure and the use of the Durbin-Watson test as a convergence criterion, it is also possible that this algorithm may take an unreasonably large amount of iterations to converge, or not converge at all (Andrews, 2018).
3.4 BootAECM Algorithm

Shreeves and Andrews (2019) demonstrated that the bootstrapped EM algorithm can be improved by employing factor analysis. The factor analysis model assumes that the $p$-dimensional observation $x_i$ arises from

$$x_i = \mu + \Lambda u_i + \epsilon_i, \quad (10)$$

where $\mu$ is a $p$-dimensional mean vector, $\Lambda$ is a $p \times q$ matrix of factor loadings, $u_i \sim N(0, I_q)$ is a $q$-dimensional vector of latent variables, and $\epsilon_i \sim N(0, \Psi)$ where $\Psi$ is a $p \times p$ diagonal matrix.

As a result, it can be shown that the marginal distribution for $x_i$ is $\phi(x_i \mid \mu, \Lambda\Lambda' + \Psi)$. This methodology was applied to the field of mixture models (Ghahramani et al., 1996; McLachlan and Peel, 2000) and named “mixtures of factor analyzers.” The reason for using this method over the conventional finite mixture model is two-fold; (1) it allows one to find underlying factors of groups and (2) the computational benefits in terms of both storage and run time are needed for moderate-to-high dimensional data sets. This is due to the fact that the number of free parameters to be estimated is reduced and the evaluation of $\hat{\Sigma}^{-1}$ and $|\hat{\Sigma}|$ are calculated in a computationally advantageous manner. Under a fully unconstrained mixture of factor analyzers model, the number of free parameters is reduced to

$$G \left( pq - \frac{q(q - 1)}{2} \right) + Gp. \quad (11)$$

The key difference between (6) and (11) is that the $p^2$ term, the main source of free parameters, no longer exists in (11). Additionally, this implementation does not require multiple inversions of the $p \times p$ covariance matrix for the calculation of the likelihood. These calculations are instead approximated using the latent subspace, reducing computation time (McNicholas, 2016).

This was implemented in a bootstrap framework by Shreeves and Andrews (2019) using the techniques from Andrews (2018), allowing for the use of the algorithm in moderate-to-high dimensional settings, which was not initially feasible with BootEM.

3.5 Spectral Clustering

Spectral clustering (Ng et al., 2001) is an extremely popular spectral methodology used in cluster analysis. Spectral clustering utilizes a singular value decomposition (SVD) of the data matrix to transform the data into a new space. Statistical analysis is then performed on the transformed data. Vempala and Wang (2004) show that simple spectral clustering algorithms are able to effectively learn Gaussian mixture models under minor assumptions. In our work, we perform a spectral transformation on the data matrix prior to mixture model estimation in two cases: before and after
taking a bootstrap sample. We utilize the data’s similarity matrix in our clustering technique, and assume that singular values along the similarity matrix’s diagonal are in decreasing order. This type of spectral clustering, where one makes use of the spectrum of the data’s similarity matrix to reduce its dimension, is well studied (Kannan et al., 2009; Kumar and Kannan, 2010).

We further borrow techniques from Löffler et al. (2021). There, the authors’ perform SVD on the data matrix $X$ to perform $G$-means clustering on the matrix product $\hat{Y} = XV_G'$, where $V_G'$ is composed of the first $G$ columns of the unitary matrix $V'$ from the SVD of $X$ such that $X = UV \Sigma V'$. This technique reduces the original $n \times p$ data matrix to one that is $n \times G$, significantly reducing the dimensionality of the data. By clustering on $\hat{Y}$, the spectral gap condition (Von Luxburg (2007), Löffler et al. (2021)) can be removed. Using this spectral transformation from Löffler et al. (2021), we create an EM algorithm that fits Gaussian mixture models. We call this the Spectral-EM algorithm. While this methodology is dynamic in the sense the number of singular values is determined based on the number of expected groups present ($G$), it assumes that this information is known.

4 Methodology

4.1 Bootstrapped Clustering with Spectral Decomposition

We present two algorithms to address convergence to local maxima and high dimensionality using a bootstrapped EM with SVD. This is accomplished by utilizing the spectral transformation from Löffler et al. (2021) and the bootstrapping procedure employed by Andrews (2018), allowing for increased computational speed and reduction in overfitting to the data. These algorithms are presented in Sections 4.1.1 and 4.1.2.

4.1.1 Spectral-BootEM Algorithm

The first algorithm is an adjustment of the algorithm original developed by Andrews (2018). In our adjustment, we perform an SVD on the transpose of the $n \times p$ data matrix $X$ and then perform the non-parametric bootstrap on $\hat{Y} = XV_G'$ in addition to altering the convergence criteria. We refer to this algorithm as Spectral-BootEM, and it is given in 3.

This algorithm will be referred to as the Spectral Bootstrapped EM, or Spectral-BootEM for short, for the remainder of this paper.

We modify the convergence criteria from the method proposed by Andrews (2018), where the Durbin-Watson test is used to check for auto-correlation. In the modification, convergence is determined by computing the component-wise relative difference in the averaged parameter between iterations space divided by the number of free parameters. Specifically, if we denote the
Algorithm 3: Spectral bootstrapped EM algorithm for GMM estimation (Spectral-BootEM)

| Input: X |
|---|
| 1 Perform SVD on X and compute $\hat{Y} = X V^G_G$; |
| 2 Create initial cluster memberships $z^{(0)}$ using $\hat{Y}$; |
| 3 Set the bootstrap index $k = 1$; |
| while not converged do |
| 5 Sample with replacement from $\hat{Y}$, denoting the sample as $Y^{(k)}$ and take the corresponding cluster memberships from $z^{(k-1)}$; |
| 6 Perform the EM (Algorithm 1) on $Y^{(k)}$ using $z^{(k-1)}$ as cluster initializations to compute parameter estimates; |
| 7 Compute group memberships for $\hat{Y}$ using parameter estimates; |
| 8 Check for convergence; |
| end |
| Output: $\hat{\theta}^*$ |

$k^{th}$ relative difference in averaged parameter space by $R_{\theta}^{(k)}$, convergence is achieved when

$$ R_{\theta}^{(k)} = \frac{1}{G^2 + G p^2 + G p (p + 1)/2} < \epsilon_B, \quad (12) $$

where $\epsilon_B$ is a small, user-defined value and not necessarily equal to the $\epsilon$ used for convergence discussed in Section 3.2. The reason for doing so is simple. While the Durbin-Watson criteria has the potential of never converging to a solution (see Andrews (2018)), the relative difference in the averaged parameter space will reach convergence as the number of iterations increases. Written algebraically, this takes the form

$$ R_{\theta}^{(k)} = \sum_{g=1}^{G} \left| \frac{\hat{\pi}^{*(k)} - \hat{\pi}^{*(k-1)}}{\hat{\pi}^{*(k-1)}} \right| + \sum_{g=1}^{G} \sum_{j=1}^{p} \left| \frac{\hat{\mu}^{*(k)} - \hat{\mu}^{*(k-1)}}{\hat{\mu}^{*(k-1)}} \right| + \sum_{g=1}^{G} \sum_{j=1}^{p} \sum_{k=j}^{p} \left| \frac{\hat{\sigma}^{*(k)} - \hat{\sigma}^{*(k-1)}}{\hat{\sigma}^{*(k-1)}} \right|. \quad (13) $$

Here, $\hat{\pi}^{*(k)}$, $\hat{\mu}^{*(k)}$, and $\hat{\sigma}^{*(k-1)}$ represent the averaged mixing proportion, mean vector, and covariance matrix, respectively, for the $k^{th}$ bootstrap sample. When computing the component wise relative differences in the covariance matrices, we do not include the lower triangles of the group matrices, as these are symmetric and hence the off-diagonal components would be contributing twice to the relative difference. In both (12) and (13), we have written $p$ to represent the dimensionality of the data. However, in the case our spectral transformation $p$ is reduced down to $G$, significantly reducing the number of terms required to be calculated. We include a minimum number of bootstrap samples to be performed before we start to check for convergence.

4.1.2 BootSpectral Algorithm

The second algorithm we propose is more computationally expensive than Algorithm 3. After creating the bootstrap sample, $X^{(k)}$, SVD is then performed on the bootstrap sample and $\hat{Y}^{(k)} = X^{(k)} V^G_G$ is computed. The rest of the bootstrap EM is then continued algorithm as in Andrews (2018). With this method, a new SVD is computed for each sample. We note that SVD is not a
unique operation. Despite this, we assume that SVD across samples is similar enough to allow for a reputable model, while also creating enough diversity during model fitting to address the issue of overfitting. The algorithm is as follows:

**Algorithm 4**: Bootstrapped spectral EM algorithm for GMM estimation (BootSpectral)

| Input: X |
|----------|
| 1. Perform SVD on $X$ and compute $\hat{Y} = X V_G'$; |
| 2. Create initial cluster memberships $\hat{z}^{(0)}$ using $\hat{Y}$; |
| 3. Set the bootstrap index $k = 1$; |
| 4. while not converged do |
| 5. Sample with replacement from $X$, denoting the sample as $X^{(k)}$; |
| 6. Perform SVD on $X$ and compute $\hat{Y}^{(k)} = X^{(k)} V_G'$ and take the corresponding cluster memberships from $\hat{z}^{(k-1)}$; |
| 7. Perform the EM algorithm on $Y^{(k)}$ using $\hat{z}^{(k-1)}$ as cluster initializations to compute parameter estimates; |
| 8. Compute group memberships for $\hat{Y}$ using parameter estimates; |
| 9. Check for convergence; |
| 10. end |
| Output: $\hat{\theta}^*$ |

As in the Spectral-BootEM algorithm, lack of progress in the parameter space is again used as the convergence criteria. We call this algorithm the Bootstrapped Spectral Clustering algorithm, or BootSpectral for short. We compare this algorithm to Spectral BootEM, as well as the standard spectral clustering, BootEM, EM, and BootAECM algorithms in Section 4.

Both the BootSpectral and Spectral-BootEM algorithms allow for the computation out-of-bag group membership estimates, which provide a more statistically favourable interpretation of the results. It assumes that an observation omitted from the bootstrap sample has not been observed, resulting in a built-in cross validation technique.

### 4.2 Potential Issues

A bootstrap sample typically omits one third of the original data. While this allows for a natural cross-validation of the fitted model, it also poses the issue in the event that there exists a small sized group, comparative to $n$. As such, when taking a bootstrap sample there is a risk that not enough observations from potential groups are sampled, resulting either in empty groups or unrealistic parameter estimates. These are well known problems in the clustering community which many algorithms are prone to, including the ones we present in Sections 4.1.1 and 4.1.2. Further, with any data transformation technique, there is a risk in losing a non-trivial amount of the variance within the original data. In data with geometrically close groups, a loss of variance due to a transformation may be unable to fit one or more groups. However, spectral transformations perform well on noisy data, so this is not a significant concern and only mentioned for completeness.
5 Results

5.1 Simulations

5.1.1 Mirror Data

We begin by returning to our mirror data set, first presented in Section 2. While it is difficult to compare log-likelihood values between our proposed algorithms and the non-spectral algorithms due to the spectral transformation impacting model complexity, the key metrics of interest are run times, number of bootstrap samples required, and the estimated probability vectors for the centre point. The two novel algorithms are applied and compared to the EM, SpectralEM, BootEM, and BootAECM in Table 1. The probability vector for group membership of the centre point is denoted as \( z_c \), and the OOB group membership for the centre point is provided in the column \( OOB z_c \). A seed is set for reproducibility. Additional settings include \( \epsilon = 0.1 \) for all EM convergence parameters, \( \epsilon_B = 0.001 \) for the Spectral-BootEM and BootSpectral convergence parameters, and the Durbin-Watson p-value is equal to 0.05 for the BootEM algorithm.

| Algorithm              | \( z_c \)       | OOB \( z_c \) | Log-Likelihood | Bootstraps | Elapsed Time (s) |
|------------------------|-----------------|---------------|----------------|------------|------------------|
| EM                     | [0, 1]          | -             | -202011.6      | -          | 0.301            |
| AECM                   | [0, 1]          | -             | -214085.7      | -          | 1.81             |
| SpectralEM             | [0.493, 0.507]  | [0.496, 0.504]| -2465.0        | -          | 0.074            |
| BootEM                 | [0.490, 0.510]  | [0.506, 0.494]| -202441.8      | 4202       | 3769.097         |
| BootAECM               | [0.422, 0.578]  | [0.475, 0.525]| -214244.2      | 500        | 415.56           |
| Spectral-BootEM        | [0.498, 0.502]  | [0.493, 0.507]| -5022.308      | 1186       | 128.447          |
| BootSpectral           | [0.503, 0.497]  | [0.496, 0.504]| -5021.03       | 2659       | 450.017          |

Table 1: Comparative results for high-dimensional mirror data.

Analyzing Table 1, we see that the EM, AECM, and SpectralEM algorithms heavily overfit the data, as they are classifying the centre point as belonging exclusively to a single group. The BootEM algorithm performs much better at classifying the centre point, however it takes the most bootstrap samples and longest time to complete, performing at a speed of (approximately) 1.1 bootstrap samples per second. BootAECM again addresses the overfitting issue and requires significantly fewer bootstrap samples to complete than the BootEM case, however the returned probability vector is the furthest from the intuitive [0.5, 0.5] for the bootstrapped algorithms. It performs at a speed of 1.2 bootstrap samples per second, a slight improvement over the BootEM algorithm. Analyzing our first algorithm, Spectral-BootEM, we find that we address the issue of convergence to local maxima and overfitting in the original SpectralEM, demonstrated by the Spectral-BootEM’s group membership probabilities. It required 1186 bootstrap samples and completed in 128.447 seconds, attaining a speed 9.2 bootstrap samples per second. This is a significant increase in computational speed over both the BootEM and BootAECM and is due to the spec-
tral transformation that is performed. Our second algorithm, the BootSpectral algorithm, also addresses overfitting, achieving a similar log-likelihood as the Spectral-BootEM and nearly equivalent group membership probabilities for both the actual memberships and OOB memberships of the centre point. It performed in 2659 bootstrap samples and completed approximately 6.0 bootstrap samples per second, a decreases in speed from the Spectral-BootEM. This is due to the fact that BootSpectral is computing a unique SVD for each bootstrap sample, resulting in a more computationally expensive process. In contrast, Spectral-BootEM computes an SVD only once. This motivating example demonstrates that the novel algorithms address the issues of overfitting and convergence to local maxima solutions in a high-dimensional setting.

To further investigate the convergence and speed of our algorithm, we run the BootEM, Spectral-BootEM, and BootSpectral algorithms on the mirror data 5 times with varying seeds. All other input parameters are the same as before. The speed at which these algorithms reach convergence and the required number of bootstrap samples to do this are displayed in Table 2a.

| Algorithm      | Run 1  | Run 2  | Run 3  | Run 4  | Run 5  |
|----------------|--------|--------|--------|--------|--------|
| BootEM         | 1575.210 | 482.769 | 448.512 | 722.639 | 2540.199 |
| Spectral-BootEM| 99.306  | 127.603 | 78.259  | 60.740  | 97.106  |
| BootSpectral   | 401.097 | 520.417 | 181.603 | 221.718 | 318.404 |

(a) Time (s) to reach convergence.

| Algorithm      | Run 1  | Run 2  | Run 3  | Run 4  | Run 5  |
|----------------|--------|--------|--------|--------|--------|
| BootEM         | 1733   | 562    | 500    | 842    | 2836   |
| Spectral-BootEM| 856    | 1174   | 714    | 554    | 923    |
| BootSpectral   | 2350   | 2956   | 1182   | 1428   | 1970   |

(b) Bootstrap samples required for convergence.

Table 2: Run times and samples required to reach convergence for mirror data.

It can be seen that the BootEM consistently takes the longest time to reach convergence, and even its quickest run is longer than the slowest run of Spectral-BootEM and four of the five runs of BootSpectral. We find that Spectral-BootEM is consistently the fastest algorithm to reach convergence its defined convergence criteria, with BootSpectral being slower than Spectral-BootEM but still much quicker than the standard BootEM. Table 2b shows that the required samples for the BootEM to reach convergence is fairly sporadic in nature, as in run 3 it reaches convergence at its minimum amount of 500 samples while run 5 takes over 2800 samples. Again, Spectral-BootEM is the most consistent of the three algorithms. Interestingly, BootSpectral in run 2 requires the most samples out of all runs for each algorithm to reach convergence. This is again due to the fact that BootSpectral recomputes the SVD for each bootstrap sample generating a larger latent space on which to estimate each model, and hence certain sample estimates may act as outliers impacting...
the averaged parameter space more significantly than others. Despite the longer runtime, the increased exploration can allow the algorithm to better explore the parameter space. We conclude that that our novel bootstrapped spectral clustering algorithms and convergence criteria generally produce more consistent results when compared to the BootEM.

5.1.2 Cross-Over Data

The next example involves a simulated longitudinal dataset. We create a two group longitudinal data set where at the halfway point, the groups cross over one another, as demonstrated in Figure 2a. One group trends downwards while the trends up. In this dataset, there are 150 observations in each group where at each time $t = \{1, \ldots, 41\}$ a value is recorded. The first group has a mean that travels from -20 to 20 by an increment of one per time step, while the second group mean travels from 20 to -20 again by an increment of one per time step. The covariance matrix for each group is constructed to be 1 on the diagonal and 0.9 on the off-diagonal. Observations are simulated using a multivariate normal distribution. Three observations are then created which belong to the bottom group for the first 20 time points, and join the other group at the cross-over event, continuing on the new path (see Figure 2b). We refer to this as the cross-over data set. The EM and SpectralEM algorithm both overfit to this data, and as was seen in the mirror simulation from Section 5.1.1, assign a group probability vector to each group changing observation of $[1, 0]$. One would intuitively expect the correct solution would be closer to a group probability vector of $[0.5, 0.5]$, as was the case with the mirror data.

We run three experiments to demonstrate the versatility of our convergence criteria, and compare it to the results of the BootEM algorithm. For both the Spectral-BootEM and BootSpectral, the model is fitted using the cross-over data set with $\epsilon_B$ set to 0.01, 0.005, and 0.0001. The resulting group probability vectors for the group changing observations are analyzed in Tables 3a, 3b,
and 3c. For all cases, we set the minimum bootstrap samples to be 300. The group changing observations are denoted as Obs 1, Obs 2, and Obs 3. We further analyze the convergence of the averaged parameter space of Spectral-BootEM and BootSpectral in Figures 3a and 3b, respectively.

|                  | Spectral-BootEM $z_i$ | BootSpectral $z_i$ | BootEM $z_i$ |
|------------------|-----------------------|-------------------|--------------|
| Obs 1            | [0.703, 0.297]        | [0.620, 0.380]    | [0.493, 0.507]|
| Obs 2            | [0.710, 0.290]        | [0.617, 0.383]    | [0.446, 0.554]|
| Obs 3            | [0.707, 0.293]        | [0.620, 0.380]    | [0.490, 0.510]|

(a) $\epsilon_B = 0.01$.

|                  | Spectral-BootEM $z_i$ | BootSpectral $z_i$ | BootEM $z_i$ |
|------------------|-----------------------|-------------------|--------------|
| Obs 1            | [0.490, 0.510]        | [0.577, 0.423]    | [0.493, 0.507]|
| Obs 2            | [0.480, 0.520]        | [0.573, 0.427]    | [0.490, 0.510]|
| Obs 3            | [0.487, 0.513]        | [0.573, 0.427]    | [0.490, 0.510]|

(b) $\epsilon_B = 0.005$.

|                  | Spectral-BootEM $z_i$ | BootSpectral $z_i$ | BootEM $z_i$ |
|------------------|-----------------------|-------------------|--------------|
| Obs 1            | [0.487, 0.513]        | [0.509, 0.491]    | [0.493, 0.507]|
| Obs 2            | [0.485, 0.515]        | [0.506, 0.494]    | [0.490, 0.510]|
| Obs 3            | [0.486, 0.514]        | [0.508, 0.492]    | [0.490, 0.510]|

(c) $\epsilon_B = 0.0001$.

Table 3: Averaged $z$ vectors for group changing observations.

Analyzing first Table 3a, where $\epsilon_B = 0.01$, we see that the Spectral-BootEM still partially overfits to the data, as is shown by the poor probabilities in the group probability vectors. Meanwhile, the BootSpectral algorithm performs well, with group probability vectors close to $[0.5, 0.5]$. The BootEM also performs admirably, and much better than Spectral-BootEM. However, as our convergence criteria allows for flexibility in $\epsilon_B$, we lower this to 0.005, and display these results in table 3a. Immediately we see a an improvement in membership probabilities in both the Spectral-BootEM and BootSpectral algorithms. Note that the BootEM remains unchanged, as it’s convergence criteria uses a Durbin-Watson test for auto-correlation, and as such does not allow for any flexibility assuming a significance level of $\alpha = 0.05$ is kept.

Next, the convergence criteria is again reduced such that the relative difference (scaled by the number of free parameters) in the average parameter space cannot exceed 0.0001, and display the results in Table 3c. Here, we find the most intuitively logical results for Spectral-BootEM and BootSpectral, as both provide group probability vectors close to $[0.5, 0.5]$ for all changing observations. This demonstrates the flexibility of the convergence criteria.

We now analyze the behaviour of convergence criteria for the cross-over data. Beginning with Spectral-BootEM in Figure 3a, we see that setting $\epsilon_B = 0.0001$ results in the most iterations,
(a) Impact of convergence parameter $\epsilon_B$ on convergence rate of averaged parameter space in Spectral-BootEM.

(b) Impact of $\epsilon_B$ on convergence rate of averaged parameter space in BootSpectral.

Figure 3: Convergence of bootstrapped spectral algorithms with varying $\epsilon_B$. 

16
as is expected, and achieves an asymptotic behaviour demonstrating that bootstrap samples will have a smaller impact on the overall averaged parameter space; something which has been found to have a significant negative impact the convergence BootEM algorithm (Andrews, 2018). The results are analogous for the BootSpectral, which we show in Figure 3b. In both figures, a large $\epsilon_B$ results in extremely fast convergence. As we do not begin checking for convergence before iteration 300, it is possible that our algorithms could converge earlier for these less strict values of $\epsilon_B$. To avoid potential overfits we maintain that a minimum of 300 bootstraps samples is required. We note that in Figure 3a, despite the convergence measure being below 0.005 for multiple iterations, extreme bootstrap samples significantly impact the averaged parameter space, and we see a large spike occurring early in the convergence measure growing to larger than 0.01. This is despite the criterion being relatively small to begin with, as our two other runs with larger $\epsilon_B$ values had already converged. As such, we find the choice of $\epsilon_B$ to be of great importance to achieve a stable averaged parameter space. These figures also show the importance of suitable minimum iteration count, in order to avoid any significant spikes in the relative difference of the averaged parameter space between iterations in the early stages of the bootstrapped model estimation.

5.2 Real Data

5.2.1 Raman Data

Raman spectroscopy (McCreery, 2005) is an optical interrogation method which consists of using inelastic light scattering to identify constituent chemicals by their vibrational modes. The acquired spectra can therefore provide detailed information on a range of constituents within a single sample acquisition. Changes in the positioning of peaks, as well as their relative amplitudes, can be used to identify the change in molecular dynamics due to a particular perturbation of a system.

Matthews et al. (2015) collected a series of Raman spectra that are classified into three different groups; lung (H460), breast (MCF-7), and prostate (LNCaP) tumor cells. Their research identified that using dimensionality reduction techniques, such as principal component analysis, is a potential way to identify the discrepancies between different cell types while also monitoring the underlying biochemical changes within given cell lines. Specifically, they found that one can distinguish the lung and breast tumor cells from the prostate tumor cells through the levels of glycogen found in their underlying spectra. Deng et al. (2020) further verified this using a variety of dimensionality reduction techniques, such as nonnegative matrix factorization and nonnegative least squares, comparing the results to those previously found by Matthews et al. (2015). Additionally, they were able to identify both glycogen and lipid-like components to further discern the disparity between different cell lines. Herein, we elect to utilize our novel methodology to identify the underlying groups structure under the assumption that there is no group information available.
Figure 4: A demonstration of observations in the Raman spectroscopy data set. Observations are coloured in grey, whereas the group means are displayed in red, green, and blue.

The data consists of 1080 observations in the 381-dimensional space. Here each observation consists of a series of Raman intensities (arbitrary units) at specific wavenumbers (cm$^{-1}$). Figure 4 displays the collection of observations in our data set coloured in grey. Additionally, we include the group means which identify the averaged spectra for the H460 (red), MCF-7 (blue), and LNCaP (green) cell lines. Each group has an even number (360) of observations, which are differentiated by radiation dose, number of days, and individual number; all of which are left out of the clustering model.

| Algorithm       | CR  | ARI  | Log-Likelihood | Bootstraps | Elapsed Time (s) |
|-----------------|-----|------|----------------|------------|------------------|
| AECM            | 1.00| 1.00 | 3,648,714      | -          | 2000.84          |
| SpectralEM      | 0.99| 0.98 | 21,270.54      | -          | 0.83             |
| Spectral-BootEM | 0.99| 0.98 | 21,250.56      | 300        | 53.75            |
| BootSpectral    | 0.99| 0.97 | 20,874.64      | 300        | 195.55           |

Table 4: Comparative results for Raman spectroscopy data. Classification rate (CR) and adjusted Rand index (ARI) are both measures of agreement between true classes and the assigned cluster memberships.

Table 4 displays results from the Spectral-EM algorithm and its bootstrapped counterparts on the Raman spectroscopy data set. These results include the percentage of correctly classified observations (CR), the adjusted rand indices (ARI; Hubert and Arabie (1985)), log-likelihoods, number of bootstrap iterations, and elapsed run time of the algorithms. While the conventional Spectral-EM performs well, it should be noted that both the Spectral-BootEM and BootSpectral algorithms also achieve similar results. This is indicative of a clear solution in the data. However, it should be noted that this clustering problem is not feasible using the conventional EM algorithm, its bootstrapped counterpart, or the BootAECM due to the requirement of $p \times p$ matrix inversions; an operation that often results in computational errors. This is due to the fact that matrix
singularities arise from a deficient number of observations within groups and is further compounded when taking into consideration the data left out in a bootstrapped sample. While the conventional AECM algorithm does not require a $p \times p$ matrix inversions, its bootstrapped version does since it requires an inversion of the averaged sample space covariance matrix. This cannot be avoided as an inversion using the averaged latent parameters can be perturbed due to discoveries of different orthogonal rotations in the data. The conventional AECM algorithm performs admirably, with a perfect classification rate and ARI. However, its elapsed run time is over ten times the next slowest algorithm in table 4. While it is the user’s decision to decide which algorithm is more appropriate, the expense of a few misclassifications in exchange for much faster run time was deemed worthy in this case. Further, the using the bootstrapped algorithms assuage concern of overfitting to the data, something that is a more likely in AECM algorithm given its perfect classification rate.

6 Conclusion

We have demonstrated that our two novel bootstrapped spectral clustering algorithms effectively and efficiently address the previously defined issues of convergence to sub-optimal solutions. These algorithms have shown to achieve a significant boost in speed without a loss in accuracy occurring from the spectral transformation. We demonstrated the versatility and efficacy of our novel algorithms compared to two prior bootstrapped clustering algorithms on simulated and real data experiments. Our choice of convergence technique by way of the relative difference in the averaged parameter space allows for a more consistent model solution than other convergence criteria methods, such as the Durbin-Watson test.

Both algorithms have their benefits and drawbacks. The Spectral-BootEM is easier to interpret and less likely to result in substantial deviations in the averaged parameter space for each bootstrapped model, while the BootSpectral is more computationally expensive but allows for the discovery of latent spaces that may not be immediately obvious in the full data set. This results in greater variation of the fitted models of each bootstrap, causing full exploration of the parameter space.

7 Acknowledgements

The authors kindly thank Dr. Jeffrey Andrews for providing infrastructure allowing for a more streamlined research process. The authors also thank Dr. Jeffrey Andrews and Dr. Ryan Browne for permission to use the cross-over data simulation example. Finally, we thank Dr. Andrew Jirasek and his medical physics research group for permission to use the Raman Spectroscopy data.
References

Andrews, J. L. (2018). Addressing overfitting and underfitting in gaussian model-based clustering. *Computational Statistics & Data Analysis*, 127:160–171.

Andrews, J. L., McNicholas, P. D., and Subedi, S. (2011). Model-based classification via mixtures of multivariate t-distributions. *Computational Statistics & Data Analysis*, 55(1):520–529.

Banfield, J. D. and Raftery, A. E. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821.

Bouveyron, C. and Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, 71:52–78.

Celebi, M. E. and Aydin, K. (2016). *Unsupervised learning algorithms*. Springer.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern recognition*, 28(5):781–793.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Deng, X., Ali-Adeeb, R., Andrews, J. L., Shreeves, P., Lum, J. J., Brolo, A., and Jirasek, A. (2020). Monitor ionizing radiation-induced cellular responses with raman spectroscopy, non-negative matrix factorization, and non-negative least squares. *Applied spectroscopy*, 74(6):701–711.

Efron, B. (1981). Nonparametric estimates of standard error: the jackknife, the bootstrap and other methods. *Biometrika*, 68(3):589–599.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*. SIAM.

Franczak, B. C., Browne, R. P., and McNicholas, P. D. (2013). Mixtures of shifted asymmetric laplace distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1149–1157.

Georgiades, S., Tait, P. A., McNicholas, P. D., Duku, E., Zwaigenbaum, L., Smith, I. M., Bennett, T., Elsabbagh, M., Kerns, C. M., Mirenda, P., et al. (2022). Trajectories of symptom severity in children with autism: Variability and turning points through the transition to school. *Journal of autism and developmental disorders*, 52(1):392–401.

Ghahramani, Z., Hinton, G. E., et al. (1996). The EM algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto.
Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2:193–218.

Kannan, R., Vempala, S., et al. (2009). Spectral algorithms. *Foundations and Trends® in Theoretical Computer Science*, 4(3–4):157–288.

Kumar, A. and Kannan, R. (2010). Clustering with spectral norm and the k-means algorithm. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 299–308. IEEE.

Löffler, M., Zhang, A. Y., and Zhou, H. H. (2021). Optimality of spectral clustering in the gaussian mixture model. *The Annals of Statistics*, 49(5):2506–2530.

Matthews, Q., Isabelle, M., Harder, S. J., Smazynski, J., Beckham, W., Brolo, A. G., Jirasek, A., and Lum, J. J. (2015). Radiation-induced glycogen accumulation detected by single cell raman spectroscopy is associated with radioresistance that can be reversed by metformin. *PloS one*, 10(8):e0135356.

McCreery, R. L. (2005). *Raman spectroscopy for chemical analysis*. John Wiley & Sons.

McLachlan, G. and Peel, D. (2000). Mixtures of factor analyzers. In *In Proceedings of the Seventeenth International Conference on Machine Learning*. Citeseer.

McLachlan, G. J. and Krishnan, T. (2007). *The EM algorithm and extensions*. John Wiley & Sons.

McLachlan, G. J., Lee, S. X., and Rathnayake, S. I. (2019). Finite mixture models. *Annual review of statistics and its application*, 6:355–378.

McNicholas, P. D. (2016). *Mixture model-based classification*. Chapman and Hall/CRC.

Ng, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14.

Punzo, A. and McNicholas, P. D. (2016). Parsimonious mixtures of multivariate contaminated normal distributions. *Biometrical Journal*, 58(6):1506–1537.

Shreeves, P. and Andrews, J. L. (2019). A bootstrap-augmented alternating expectation-conditional maximization algorithm for mixtures of factor analyzers. *Stat*, 8(1):e243.

Tipping, M. E. and Bishop, C. M. (1999). Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482.

Torres-Carrasquillo, P. A., Reynolds, D. A., and Deller, J. R. (2002). Language identification using gaussian mixture model tokenization. In *2002 IEEE international conference on acoustics, speech, and signal processing*, volume 1, pages I–757. IEEE.
Vempala, S. and Wang, G. (2004). A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

Yu, J. (2012). Online quality prediction of nonlinear and non-gaussian chemical processes with shifting dynamics using finite mixture model based gaussian process regression approach. *Chemical engineering science*, 82:22–30.