Published in final edited form as:

J Algebra Appl. 2020 November ; 19(11): . doi:10.1142/s0219498820502047.

IDENTIFYING THE NUMBER OF COMPONENTS IN GAUSSIAN MIXTURE MODELS USING NUMERICAL ALGEBRAIC GEOMETRY

SARA SHIRINKAM,

Department of Mathematics and Statistics, University of the Incarnate Word, 4301 Broadway, CPO 311, San Antonio, TX 78209, USA

ADEL ALAEDDINI,

Department of Mechanical Engineering, University of Texas at San Antonio, One UTSA Circle San Antonio, TX 78249, USA

ELIZABETH GROSS

Department of Mathematics, University of Hawai'i at Mānoa, 2565 McCarthy Mall, Honolulu, Hawaii 96822, USA

Abstract

Using Gaussian mixture models for clustering is a statistically mature method for clustering in data science with numerous successful applications in science and engineering. The parameters for a Gaussian mixture model are typically estimated from training data using the iterative expectation-maximization algorithm, which requires the number of Gaussian components a priori. In this study, we propose two algorithms rooted in numerical algebraic geometry, namely an area-based algorithm and a local maxima algorithm, to identify the optimal number of components. The area-based algorithm transforms several Gaussian mixture models with varying number of components into sets of equivalent polynomial regression splines. Next, it uses homotopy continuation methods for evaluating the resulting splines to identify the number of components that results in the best fit. The local maxima algorithm forms a set of polynomials by fitting a smoothing spline to a kernel density estimate of the data. Next, it uses numerical algebraic geometry to solve the system of the first derivatives for finding the local maxima of the resulting smoothing spline, which estimates the number of mixture components. The local maxima algorithm also identifies the location of the centers of Gaussian components. Using a real-world case study in automotive manufacturing and multiple simulations, we compare the performance of the proposed algorithms with that of Akaike information criterion (AIC) and Bayesian information criterion (BIC), which are popular methods in the literature. We show the proposed algorithms are more robust than AIC and BIC when the Gaussian assumption is violated.

Keywords

Mixture models; model-based clustering; smoothing spline; numerical algebraic geometry

1. Introduction

Mixture models based on Gaussian distributions have received considerable attention in recent years, leading to a large number of methodological developments and applications ranging from manufacturing [13] to economics [49] to health-care [45]. Despite their widespread use, deciding how many components in a Gaussian mixture model (GMM) are needed to adequately represent the data is still an open problem [25]. Most of the existing approaches for estimating the number of components in a Gaussian mixture model are based on the log-likelihood function with a form of penalty, also known as an information criterion. These methods typically use an iterative procedure with various numbers of components to minimize or maximize the criterion in order to estimate the true number of components. The two most commonly used information criteria are the Akaike information criterion (AIC) [32] and the Bayesian information criterion (BIC) [17]. AIC and BIC both provide results consistent with the minimum description length of the data, a framework in which the best model is the one that leads to the best compression [44]. Other common criteria include corrected AIC [34], vector corrected Kullback information criterion (KIC) [35], and weighted-average information criterion (WIC) [50]. Comparative studies of using these criteria for model selection can be found in [39]. Most of these methods need some prior that can bias the estimate, and therefore they often have the problem of over-fitting [14]. Meanwhile, some of the more recent studies have suggested variants of these criteria such as AIC_{MIX} [33] and CCL [9] that provide better accuracy under certain conditions such as large sample size.

In this paper, we develop two algorithms, both based on nonparametric regression and numerical algebraic geometry (NAG), to address the problem of possible bias in estimating the optimal number of components in a Gaussian mixture model and to illustrate how algebraic geometry can be applied to the model selection problem. Indeed, the geometry of Gaussian mixture models have been studied recently in algebraic statistics (see [5,3,6]) with fruitful results. In particular, this geometric framework has been used to answer questions about the critical points of the likelihood function [1], to establish that all univariate Gaussian mixtures are algebraically identifiable [4], and to give lower and upper bounds on the number of modes of a Gaussian mixture [43,2]. In terms of model selection for Gaussian mixtures, other than in [21] where the authors test their geometrically motivated sBIC criterion on data from univariate Gaussian mixtures, there has been little research on using computational algebraic geometry for the Gaussian mixture model selection problem. Thus, there is much room to explore how creatively combining tools from nonlinear algebra and statistics can result in new methodologies. By presenting two such approaches, we can compare and contrast the benefits and challenges of strategies rooted in nonlinear algebra.

Our first approach, the area-based algorithm, transforms several Gaussian mixture models with varying number of components, $k = 1, 2, \dots, K$, into equivalent smoothing splines and uses homotopy continuation methods from numerical algebraic geometry [7,8] to determine the regression model that is most compatible with the data. Our second approach, the local maxima algorithm, begins with fitting a kernel density to data that is assumed to be best fitted by a Gaussian mixture. Next, it transforms the estimated kernel density into a smoothing spline and uses homotopy continuation methods to find the number of local

maxima of the smoothing spline; this number estimates the number of components in the Gaussian mixture model. Since neither of the proposed approaches directly depends on the likelihood function nor needs a penalty term, they are more robust to the Gaussian assumption, and hence applicable to non-Gaussian mixture models as well. The local maxima algorithm also identifies the location of all local maxima of the equivalent smoothing spline, resulting in accurate estimates of the location of the centers of the Gaussian components.

The rest of the paper is organized as follows. In Section 2, we provide preliminary information about clustering and GMM, smoothing splines, and NAG. In Section 3, we describe the two proposed algorithms for identifying the optimal number of components in a GMM, the area-based algorithm, and the local maxima algorithm. In Section 4, first, we provide a case study in the automotive industry to demonstrate the performance of the proposed approaches in clustering manufacturing data. Next, we use seven simulated examples of GMMs with one to five dimensions to compare the performance of the proposed approaches with those of AIC and BIC methods. Then, we employ three simulated examples of non-Gaussian mixture models of two and three dimensions to evaluate the robustness of the proposed approaches for cases where the Gaussian assumption is violated. Finally, in Section 5, we present the conclusion and direction for future research.

2. Preliminaries

2.1. Gaussian mixture models

Gaussian mixture models are commonly used in distribution-based clustering. Clustering is the process of partitioning a set of observations into meaningful groups and is one of the most common objectives in statistical data analysis [36]. Distribution-based clustering uses statistical distributions to model the similarity between objects for grouping [24]. The advantage of distribution-based clustering is that it resembles the real random sampling process from a heterogeneous set of objects.

A distribution from a Gaussian mixture model can be represented as a weighted sum of k component Gaussian densities as given by the equation

$$g(x) = \sum_{i=1}^k \omega_i h(x; \mu_i, \Sigma_i),$$

where $x \in \mathbb{R}^d$, the ω_i 's are the mixture weights such that $\omega_1 + \dots + \omega_k = 1$ and $\omega_i \geq 0$ for all $1 \leq i \leq k$, and each $h(x; \mu_i, \Sigma_i)$ is a multivariate Gaussian density with mean vector μ_i and covariance matrix Σ_i [25].

Example 2.1 (Example of a Gaussian mixture).—Recall that a univariate Gaussian density with mean μ and variance σ has the following form:

$$h(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Let $h_1(x; \mu_1 = 1, \sigma_1 = 1)$, $h_2(x; \mu_2 = 5, \sigma_2 = 1.25)$, and $h_3(x; \mu_3 = 8, \sigma_3 = 0.5)$ be three univariate Gaussian densities with means 1, 5, and 8 and variances 1, 1.25, 0.5, respectively. We construct a Gaussian mixture from these three densities by choosing three weights $\omega = (\omega_1, \omega_2, \omega_3)$ that sum to 1. Let $\omega = (0.5, 0.25, 0.25)$, then the resulting Gaussian mixture is

$$g(x) = 0.5h_1(x; 1, 1) + 0.25h_2(x; 5, 1.25) + 0.25h_3(x; 8, 0.5).$$

A plot of $g(x)$ is pictured in Figure 1. Notice, that in this case, the graph has three modes, each centered at the mean of one of the component densities.

A distribution from a Gaussian mixture model is determined by the values of its parameters $\lambda = \cup_{i=1}^k \{\omega_i, \mu_i, \Sigma_i\}$. Given a set of observations $x_1, \dots, x_n$, a standard statistical approach to estimating the parameters λ is through maximum likelihood estimation, which involves maximizing the likelihood function. In the case of GMMs, the likelihood is a highly nonlinear function [12]. Thus it is difficult to maximize directly, especially when the membership of each data point is unknown (e.g. it is unknown whether x_j belongs to the first component, second component, etc.). So the standard practice is to use an effective approximation method known as the expectation-maximization (EM) algorithm [40]. Given a set of observations and a fixed number of components k , the EM algorithm estimates the best fitting parameters by iteratively assigning a component membership weight to each observation and then computing the parameters λ that maximize the likelihood with the assigned membership weights. The algorithm converges to a local optimum, so multiple runs may produce different results. Despite the fact that the number of critical points of the likelihood function can be arbitrarily large [1], the EM algorithm is a very effective tool in parameter estimation for Gaussian mixture models, but it requires knowing the number of components in the Gaussian mixture model in advance. Several statistical criteria including the AIC and BIC have been proposed in the literature to estimate the number of components. Most of these algorithms introduce a penalty term to likelihood function for the number of parameters in the GMM model. Here, we propose two new hybrid approaches to identify the optimal number of components based on smoothing splines and numerical algebraic geometry.

2.2. Smoothing splines

Smoothing splines is a curve fitting technique for modeling noisy data using a spline function [18]. Let (x_i, y_i) with $x_1 < \dots < x_n$ be a sequence of univariate observations, modeled by the relation $y_i = f(x_i) + \epsilon_i$ where the ϵ_i are independent, zero mean random variables with constant variance. The univariate smoothing spline estimate $\hat{f}$ of the function f is defined to be the minimizer over the class of twice differentiable functions of:

$$\sum_{i=1}^n (y_i - \hat{f}(x_i))^2 + \lambda \int_{x_1}^{x_n} \hat{f}''(x)^2 dx$$

where $\hat{f}(x_i)$ is a polynomial function. In our study, we consider $\hat{f}(x_i)$ as a polynomial function that is cubic in each variable in our study. Also, λ is a smoothing parameter,

controlling the trade-off between the faithfulness to the data and roughness of the function estimate. The roughness penalty based on the second derivative is most common in modern statistics literature. Multivariate splines can be obtained from univariate splines by the tensor product construct [20]. The tensor-product idea is straightforward. For a bivariate smoothing spline, if f is a function of x , and g is a function of y , then their tensor-product $p(x, y) := f(x)g(y)$ is a function of x and y . Having d -variate data organized over a d -dimensional rectangular grid, the tensor product smoothing spline has the form:

$$p(x_1, x_2, \dots, x_d) = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \dots \sum_{i_d=1}^{n_d} c_{i_1, i_2, \dots, i_d} f_1(x_{i_1}) f_2(x_{i_2}) \dots f_d(x_{i_d}).$$

When data is not organized over a rectangular grid, one can use kernel density estimation to approximate the values at the grid points to fit the spline [46]. The alternative approach is to use generalized additive models (GAM), which use an additive model where the impact of each variable x_i , $i = 1, \dots, d$ is captured through a smooth function depending on the underlying pattern in the data [29]. Generally the first approach (tensor product) works effectively when the dimensionality of the problem is low, $d \leq 3$, while the second approach (GAM) is more suitable for higher dimensions.

2.3. Numerical algebraic geometry

Numerical algebraic geometry refers to the use of numerical methods to compute approximations to solutions of polynomial systems. The core technique for most numerical algebraic geometry algorithms is homotopy continuation (for a complete background see [7,47]).

To be specific, let $F: \mathbb{C}^N \rightarrow \mathbb{C}^n$ be a polynomial system with n equations and N unknowns with only isolated solutions. NAG seeks to find numerical approximations of all $x \in \mathbb{C}^N$ such that $F(x) = 0$. This is done by selecting a polynomial system $G: \mathbb{C}^N \rightarrow \mathbb{C}^n$ in the same parameterized family as F that is easy to solve. For example, if F is a collection of $N = n$ polynomials with respective degrees $d_1, \dots, d_n$, then G may be $(x_i^{d_i} - 1, 0 \leq i \leq n)$. Once G is chosen, a NAG package then constructs a homotopy of the following form $H(x, t) = (1 - t)G(x) + \gamma t F(x)$ with $\gamma \in \mathbb{C}$ a random complex number. From there, predictor-corrector methods are used to track the paths of the solutions of $H(x, t) = 0$ as t moves from 0 to 1. For zero-dimensional systems, the final set of solutions is a set of approximations for all complex solutions to the system $F(x) = 0$. When working in applications over $\mathbb{R}$, the user can then filter these solutions to obtain all real solutions.

Due to the multitude of polynomial systems in engineering, physics, biology, statistics, and economics, numerical algebraic geometry has been used for applications in a variety of fields. For example, statistical applications of numerical algebraic geometry include model selection and parameter estimation in systems biology [27], tensor decomposition [10,30], and maximum likelihood estimation for discrete mixture models [19,28]. There are several software packages available for solving polynomial systems using polynomial homotopy

continuation. These packages include Bertini [8], Hom4PS-3 [37], and PHCpack [48]. For the computations in this paper we mainly use both PHCpack and Bertini through their Macaulay2 [26] interfaces. The algorithms behind these numerical algebraic geometry packages are probability-one algorithms designed to return all approximate solutions to a zero-dimensional polynomial system. For applications where theoretical guarantees are critical, solutions can be certified using α -theory [31]. For the algorithms discussed in this paper, typical software for numerical algebraic geometry like PHCpack and Bertini can efficiently handle cases when the data is of dimension $d \leq 3$ (the resulting systems have polynomials up to degree 9). To solve the dimensionality problem in the latter case, we suggest using Kernel Principle Component Analysis (KPCA) as a preprocessing step to first reduce the number of dimensions while considering the nonlinearity in the correlations structures, and then apply the smoothing spline and numerical algebraic geometry methods in lower dimensions. **We mention KPCA here as a preprocessing step, but never show how we use this. We should mention in the numerical examples where we use KPCA.**

3. Proposed approaches

We propose two approaches based on smoothing splines and numerical algebraic geometry for identifying the optimal number of components in a Gaussian mixture model. The area-based approach evaluates Gaussian mixture models of various number of components according to an area-based measure based on the validation error to estimate the optimal number of components. The local maxima approach approximates the probability density function using kernel density estimation and finds the number of local maxima of the estimated density to estimate the optimal number of components.

3.1. Area-based approach

In the area-based approach, first, Gaussian mixture models of various number of components are fitted to the training data. Next, each mixture density is approximated by a low order, i.e. cubic in each variable, smoothing spline. Finally, to find the model most compatible with the data, for each test point, we compute the area of a triangle formed by the test point (TE), the orthogonal projection of the test point on the spline (MD), and the estimated value of the test point on smoothing spline (EV) (see Figure 6). Methods from numerical algebraic geometry are used to find the orthogonal projection of the test point on the spline. We describe the algorithm in detail below, first, stating the algorithm, and then, walking through each step in finer detail.

Algorithm 1:

Area-based estimation of number of GMM components

Input A set of n observations $x_1, x_2, \dots, x_n$ of dimension d ; Maximum number of components K .

Output Estimated number of components.

(1) Split the dataset into training and test sets.

FOR $i = 1, \dots, K$ (Number of components)

(2) Fit a GMM with i components.

- (3) Approximate each GMM with a smoothing spline.
- FOR** each point TE in test set
- (4) Use numerical algebraic geometry to identify the point on the spline with minimum distance to the test point (MD).
- (5) Find the estimated values (EV) of the test point on smoothing spline.
- (6) Form a triangle T with vertices TE, MD, EV and calculate the area of T .
- END**
- END**
- (7) Return the GMM with minimum average area of triangles as the GMM with the optimal number of components.

Step 1: The area-based algorithm evaluates Gaussian mixture models of different number of components to estimate the number of components. Therefore, given a dataset, the first step of the proposed algorithm splits the dataset into: (1) a training set, which is used to estimate the GMM parameters after fixing the number of components, and (2) a test set for evaluating the fit of each GMM. We suggest randomly assigning about 2/3 of the dataset for the training and 1/3 of the dataset for testing.

Step 2: After splitting the data into training and test sets, GMMs with different number of components, i.e. $i = 1, \dots, K$, are fitted to the training set using the EM algorithm. For each i , we will refer to resulting mixture distribution as g_i .

Step 3: In this step, the probability density of each g_i is approximated using a smoothing spline based on the training set data. As mentioned in Section 2.2, a smoothing spline splits the domain of a GMM into a set of subintervals and defines a polynomial which is cubic in each variable on each subinterval such that the resulting piecewise function is smooth. We consider the observations in the training set as the knots for fitting the smoothing spline. ←
I want to double-check that we really mean all the training points, because for the $d=3$ and $d=5$ this would mean a lot of polynomials. Or do we first fit the GMM with the full set of training points and then fit the smoothing spline with a subset of those points? If we use the exact same set for both, I see us either having too many polynomial systems to solve or too little points for fitting the GMM. For example, for one-dimensional GMMs, with a training set of m points, using a smoothing spline results in $m-1$ subintervals and $m-1$ cubic univariate polynomial equations. For higher dimensional GMMs, e.g. $d=2, 3, \dots$, a tensor product of univariate smoothing splines can be used, which results in $(m-1)^d$ subregions and equations for the training dataset of m points, which is then placed on a hyperrectangular grid of d dimensions. Since the number of equations grows quickly with the number of dimensions, when dealing with very high-dimensional problem, i.e. $d \geq 3$, one can use generalized additive models instead of a multivariate smoothing spline to reduce the number of resulting equations. We use Matlab curve fitting toolbox for fitting the smoothing spline.

Step 4: From Step 3, we have each mixture distribution g_i represented as a set of polynomials which are cubic in each variable $f_1, \dots, f_{(m-1)^d} \in \mathbb{R}[y_1, \dots, y_d]$ defined over a set of hyperrectangles $\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_{(m-1)^d}\}$. Let x_j be a test point with $x_j \in \mathcal{R}_{j'} \subset \mathbb{R}^d$, and let $f_{j'}$ be the polynomial defining the smoothing spline over $\mathcal{R}_{j'}$. Let V be the surface in $\mathbb{R}^{d+1}$

defined by $y_{d+1} - f_{j'}$. For each test point x_j , we can find the point

$MD = (y'_1, \dots, y'_d, y'_{d+1})$ on $\mathcal{V} \cap \mathcal{R}_{j'}$ with the minimum Euclidean distance from $TE = (x_j, g_j(x_j))$ by solving a system of polynomial equations. Indeed, we can use the Karush-Kuhn-Tucker (KKT) equations [51].

Let $\mathcal{R}_{j'} = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times \dots \times [\alpha_d, \beta_d]$. Let $\lambda, \mu_1, \dots, \mu_d, \nu_1, \dots, \nu_d$ be indeterminates (these are the KKT multipliers) and let $h_\ell = \alpha_\ell - x_\ell$, $h'_\ell = x_\ell - \beta_\ell$ for $\ell = 1, \dots, d$. Then, the KKT conditions gives us the following system:

$$\begin{cases} y_{d+1} - f_{j'} = 0 \\ \lambda \nabla f_{j'} + \mu_1 \nabla h_1 + \dots \mu_d \nabla h_d + \nu_1 \nabla h'_1 + \dots \nu_d \nabla h'_d + (y - \begin{bmatrix} x_j \\ g_i(x_j) \end{bmatrix}) = 0 \\ \mu_\ell h_\ell = 0, \text{ for } \ell = 1, \dots, d \\ \nu_\ell h'_\ell = 0, \text{ for } \ell = 1, \dots, d. \end{cases}$$

Generally, solving systems of polynomial equations is computationally expensive, however, we will exploit the use of numerical algebraic geometry solvers, making such polynomial optimization problems tractable when d is small (e.g. $d \leq 3$). In particular, in our examples, we solve the systems above using PHCpack and Bertini (we found PHCpack worked better in lower dimensions and Bertini in higher dimensions).

Step 5: In addition to finding the point of minimum distance MD , we also find the estimated value EV of each test point x_j on the smoothing spline. The estimated value is obtained by evaluating the smoothing spline function at the test point, specifically $EV = (x_j, \hat{f}_j(x_j))$.

Step 6: For each test point in the dataset we form a triangle whose three vertices are TE , EV and MD points, and calculate the area of the formed triangles (see Figure 2).

Step 7: We use the minimum of average area of triangles to select the GMM with optimal number of components. That is, we return the g_j with the smallest average area of triangle as the optimal model. We tested different measures including means absolute deviation ($MAD = \text{Average} | TE - EV |$), mean squared error ($MSE = \text{Average} (TE - EV)^2$), etc. based on several simulated cases, and found the areas of triangles provides the best performance. In Section 4, we demonstrate the performance of the proposed methodology using a number of numerical examples with different dimensionality based on simulated and real-world data.

Example 3.1 (Area-based algorithm).—Let us illustrate the steps of the area-based algorithm with the one dimensional GMM with three components and parameters $\omega = (0.5, 0.25, 0.25)$, $\mu = (1, 5, 8)$, $\Sigma = ([1], [1.25], [0.5])$ from Example 2.1 (see Figure 1). (Step 1) We consider a dataset of 3000 points drawn from the underlying GMM and randomly split them into a training set of 2000 points and a test set of 1000 points, the evaluation of these points under the GMM are highlighted in blue (training points) and in red (test points) in Figure 3, Part (a). (Step 2) Next, we fit GMMs with 1 to 5 components to the training set; the fitted distributions (g_1, g_2, g_3, g_4, g_5) are shown in Figure 3, Part (b). (Step 3) Then, for each fitted mixture distribution g_i , we evaluate each training point on g_i giving us a set of

2000 points of the form $(x_j, g_j(x_j))$. We use these points to approximate each GMM with a smoothing spline as shown in Figure 3, Part (c). (Steps 4 – 6) Next, we compare the evaluation of the 1000 test points on g_j with each of the approximated smoothing splines. This comparison is done by forming 1000 triangles whose vertices are the test point and its evaluation on g_j , the point on the spline with minimum distance to the test point, and the estimated value of the test point on the spline. We then calculate the average area of the 1000 triangles. (Step 7) Finally, we use the minimum average area to identify the optimal number of components as shown in Figure 3, Part (d).

3.2. Local maxima approach

The area-based algorithm compares GMMs of different number of components to identify the optimal number of components. A drawback of the area-based algorithm is that, like AIC and BIC methods, it requires a reasonable upper-bound on the number of possible components. Here, we propose another algorithm, which uses the number of local maxima of the polynomial equations from the smoothing splines to directly determine the optimal number of components. Our main assumption is that the number of mixing components in the GMM is equal to the number of local maxima, an assumption that does not hold in all settings. However, while, theoretically, the number of local maxima is not necessarily equal to the number of components [43,2]. Simulations suggest that this behavior is rare in practice [15].

Our local maxima algorithm fits within the framework of mode-finding approaches to estimating the number of components [16,38]. The main difference between other mode-finding approaches and ours is that hill-climbing algorithms are used in the former, while we use numerical algebraic geometry. As with the area-based algorithm, we describe the algorithm in detail below, first, stating the algorithm, and then, walking through each step in finer detail.

Algorithm II:

Local maxima estimation of number of GMM components

Input Kernel density estimate of a set of n observations $x_1, x_2, \dots, x_n$ of dimension d .

Output Estimated number of components;

Estimated mean for each Gaussian component.

- (1) Fit a smoothing spline to the kernel density estimate of the data.
 - (2) For each region of the smoothing spline, set the gradient of the corresponding polynomial defining the smoothing spline to 0 to form a system of equations.
 - (3) For each region, solve the systems of equations from Step (2) using NAG to find the local extrema. Filter the solutions, keeping only those that lie in the region.
 - (4) For each point returned in Step (3), use the eigenvalues of the associated Hessian matrix to determine the local maxima.
 - (5) Return the the local maxima found in Step (4) and the count of local maxima. The local maxima are estimates of the means of each component, while the count corresponds to the number of GMM components.
-

Step 1: Instead of separate training and test datasets to compare the errors of models with various number of components as in the case of the area-based algorithm, the local maxima

algorithm starts with a kernel density estimate of the data points [46]. Having the kernel density of the data available, the local maxima algorithm fits a smoothing spline to the kernel density of the data, where the number/location of the spline notches and bandwidth of the kernel can be determined effectively using cross validation (most statistical software packages determine them automatically). In particular, at the end of this step we have a set of polynomials $f_1, \dots, f_m \in \mathbb{R}[y_1, \dots, y_d]$ defined over a set of hyperrectangles

$$\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_m\}.$$

Step 1 of the local maxima algorithm is similar to Step 3 of area-based algorithm. However, the local maxima algorithm does not need the iterative fitting of smoothing splines for GMM of various number of components. Therefore, the computational effort is reduced.

Step 2: Having a set of polynomials $f_1, \dots, f_m \in \mathbb{R}[y_1, \dots, y_d]$ approximating the true mixture distribution, the systems based on the first derivatives of the polynomials is constructed in order to find the local extrema of the smoothing spline. In particular, for each polynomial f_j , we obtain the following system:

$$\begin{cases} \frac{\partial}{\partial y_1} f_j = 0 \\ \vdots \\ \frac{\partial}{\partial y_d} f_j = 0 \end{cases} \quad (3.1)$$

Step 3: Next, NAG solvers are used to solve the resulting systems to identify the local extrema. For each pair, $\mathcal{R}_j$ and f_j , we solve the system described above and then filter the solutions. We keep only those solutions that belong to $\mathcal{R}_j$. Note that we need to solve multiple systems in the form of (3.1), so the size of d is important. However, when d is small, as in the examples in this paper, the entire process is reasonable. For example, when $d = 3$, each system takes about 0.0015 seconds to solve on a standard laptop (MacBook Pro, 2.6 GHz Intel Core i5 processor).

We use PHCpack and Bertini to solve the systems (again, we found PHCpack worked better in lower dimensions and Bertini in higher dimensions). Using a NAG solver gives us a probability-one guarantee of finding *all* local extrema, which is the main advantage of NAG over hill-climbing algorithms.

Step 4: Step 3 will give us all candidate extrema for the piecewise polynomial approximation of the kernel density. After identifying all critical points of the fitted spline, we check the size of eigenvalues of the Hessian matrix associated with each local extrema to determine the best candidates for actual local maxima. Not only will the eigenvalues of the Hessian matrix allow us to discriminate which critical points are local maxima but by restricting to maxima with significant eigenvalues, we are also able to rule out false positives, i.e. local maxima with weak signals that appear due to noise in the data and/or the fitting process. In our experiments, we used the threshold $|\lambda| > .001$ for each eigenvalue.

Step 5: At this step, we return the the local maxima found in Step 4 and the count of local maxima. The local maxima are estimates of the means of each Gaussian component.

Example 3.2 (Local maxima algorithm).—Here we provide an example of the local maxima algorithm based the same one dimensional GMM with three components from Example 2.1 and Example 3.1 with parameters $\omega = (0.5, 0.25, 0.25)$, $\mu = (1, 5, 8)$, and $\Sigma = ([1], [1.25], [0.5])$. We consider a dataset of 3000 points $(x_1, \dots, x_{3000})$ simulated from this GMM and use the kernel density estimate of the data $\hat{g}$ as the input to the algorithm. (Step 1) Next, we evaluate each data point on the kernel density estimate; these points are shown in Figure 4, Part (a). Using the points $(x_1, \hat{g}(x_1)), (x_2, \hat{g}(x_2)), \dots, (x_{3000}, \hat{g}(x_{3000}))$, we estimate a smoothing spline as shown in Figure 4, Part (b). (Steps 2–5) Then, we find the local maxima of the estimated spline using a NAG solver and use the number of local maxima to estimate the number of GMM components as shown in Figure 4, Part(c).

4. Case studies and simulated experiments

In this section, we first use seven simulated examples from one to five dimensions to show that the proposed approaches perform as expected. Next, using real data, we provide a case study to validate the performance of the proposed approaches for identifying components of in-control and out-of control parts from an auto-manufacturing company. Then, we use three simulated examples from one to three dimensions to compare the performance of the proposed approaches with AIC and BIC when the Gaussian assumption is violated.

4.1. Simulated experiments

Here we test the accuracy of the proposed area-based and local maxima algorithms using data simulated from seven GMMs of one to five dimensions with varying number of components and compare our results with AIC and BIC. Table 1 illustrates some information about each of the seven GMMs. Specifically the second column of the table shows the number of dimensions (variables) in each GMM. The third column shows the number of Gaussian components (clusters) in each GMM. The 4th, 5th, and 6th columns show the three major parameters of each GMM including the mean vectors (μ), covariance matrices (Σ) and mixture weights (ω). Finally the last column shows the number of data points simulated under each GMM (see Figure 5 for scatter plots of simulated points). For the area-based algorithm, which requires separate training and test sets, we use 2/3 of available data points for training, and 1/3 of them for validation. Meanwhile, for the methods that do not require training and test sets which include, AIC, BIC, and local maxima algorithm, we use all the available data. In addition, for iterative methods, including AIC, BIC, and area-based algorithm, which check various number of components to identify the optimal number, we let the number of components range from $i = 1$ to $i = 5$. To improve the confidence each simulation has been replicated five times and the median result has been reported.

For the GMMs described in Table 1, AIC and BIC successfully estimated the correct number of components. In addition, for all examples, both the area-based algorithm and the local maxima algorithm estimated the correct number of components, demonstrating a successful

application of the proposed approaches. In the next couple of sections, we show that our methods are more robust to violations of the Gaussian assumption.

4.2. Case study: auto manufacturing assembly line quality control

Here we apply the proposed approaches to a dataset of measurement errors from an auto manufacturing assembly line in Michigan to identify components of in-control and out-of-control sub-assemblies. The dataset provides the deviation of 5 important quality characteristics of 1,824 sub-assemblies to their nominal values, which play an important role in the quality of the final assemblies. Figure 6 shows a matrix plot where the (i, j) th entry is the projection of the dataset on variables i and j when $i \neq j$ and the histogram of the i th variable when $i = j$.

Ideally most of the measurement errors are expected to be close to each other and very small, forming a single Gaussian component with small covariance. However in practice, due to the variability in the manufacturing process, some of the sub-assemblies measurements are different from the others resulting in more than one component (as shown in Figure 6). Analysis of the components formed from the measurement errors, especially the ones with very high or very low measurement errors, provides important information about the root cause of the problems and reduction of out-of-specification products.

Both the area-based algorithm and the local maxima algorithm identified two components in the measurement error dataset, while AIC and BIC both only identified one. Comparing the results with the actual number of components determined by the quality control engineers at the manufacturing facility, who suggest two components of in- and out-of-control, shows the local max approach and the area-based approach provide the correct estimate. Meanwhile, AIC and BIC approaches underestimate the number of component, which we expect can be associated to a violation of Gaussian assumption in the case study data. In Section 4.3, we provide three simulated examples to further analyze the performance of the comparing approaches when the data comes from non-Gaussian mixture models.

4.3. Simulated experiments for non-Gaussian mixture models

The results from the previous section suggest that the area-based algorithm and the local maxima algorithm might perform better than AIC and BIC on data from non-Gaussian mixture models (NGMMs). Thus, in this section, we compare the performance of all four methods using data simulated from three NGMMs from one to three dimensions. Table 2 list the distributions, parameters, and number of data-points generated for each of the three NGMMs. (See also Figure 7).

Table 3 illustrates the number of NGMM components estimated by each method under comparison. As shown in the table, the local maxima algorithm provided the best performance across all the methods under comparison and examples. After that, the area-based algorithm shows better performance than AIC and BIC for identifying the correct number of components. We expect that the reason for the robust performance of the proposed methods is due to the fact that they are not directly based on the likelihood function such as AIC and BIC, so they get less affected by when the data deviate from the normality assumption.

4.4. Discussion

The results of the numerical examples in Sections 4.1 and 4.2 show the proposed algorithms provide similar, and in some cases, more accurate, estimates of the number of GMM components in comparison to AIC and BIC methods. Section 4.3 also shows that the proposed methods provide more accurate estimates of the number of components in NGMM including those that are mixtures of Weibull and Lognormal distributions, which can represent the important problem of dealing with edge cases. In addition, for problems with low dimensionality ($d = 3$) the computation time is similar to AIC and BIC. Therefore, considering the better accuracy of the proposed methods, they may be a better choice for low dimensional problems.

For high dimensional problems ($d > 3$), the computation times of the proposed methods are higher than AIC and BIC. This difference is mainly because the proposed algorithms incorporate a nonparametric approach for identifying the number of components. While this means that the algorithms can be directly applied in a wide range of settings (including the non-Gaussian setting), it does come at the expense of increasing the computational effort by building the systems of polynomial equations and then solving them. Nonetheless, the structure of the algorithms allows for parallel computing (indeed, the methods are embarrassingly parallel), which can considerably reduce the computation time. Additionally, one may use alternative methods such as generalized additive models (GAM) for approximating GMM, instead of smoothing spline, which scale up better for high dimensional problems.

5. Conclusion

Identifying the correct number of components is a key problem for Gaussian mixture models. In this paper, we propose two nonparametric approaches for estimating the number of components, namely an area-based approach and a local maxima approach. Both of the proposed approaches use smoothing splines to transform fitted densities into systems of polynomial equations and employ numerical algebraic geometry to solve the resulting systems to estimate the number of components. Unlike the popular approaches in the literature, the proposed approaches are not based on the likelihood function and do not require regularization. Therefore, they have less bias compared to most of the existing approaches and are more robust to violation of Gaussian assumption. In particular, using a real-world case study in automotive manufacturing and extensive simulations, we demonstrate that the proposed algorithms provide the same or more accurate results than those of Akaike information criterion (AIC) and the Bayesian information criterion (BIC). We also showed the proposed algorithms provide more accurate estimates of the number of components than AIC and BIC with data is simulated from non-Gaussian mixture models.

Meanwhile, the proposed approaches are computationally more expensive compared to existing likelihood based approaches, with significant differences starting to appear when $d > 3$. However, due to the structure of the algorithms, we believe investigations into effectively scaling would be fruitful. While the examples in this paper are low dimensional, they demonstrate how exploiting recent tools in polynomial solving can lead to new inference approaches rooted in nonparametric fitting.

Acknowledgments

Adel Alaeddini was partially supported by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-16-1-0171 and by the National Institutes of Health (NIH/NIGMS) under 1SC2GM118266-01. Elizabeth Gross was supported by the National Science Foundation under grant DMS-1620109.

References

- [1]. Améndola C, Drton M and Sturmfels B (2015) Maximum likelihood estimates for Gaussian mixtures are transcendental. In International Conference on Mathematical Aspects of Computer and Information Sciences, 579–590. Springer, Cham.
- [2]. Améndola C, Engström A and Haase C (2017) Maximum Number of Modes of Gaussian Mixtures. arXiv preprint arXiv:1702.05066.
- [3]. Améndola C, Faugere JC and Sturmfels B (2016) Moment Varieties of Gaussian Mixtures. Journal of Algebraic Statistics, 7(1).
- [4]. Améndola C, Ranestad K and Sturmfels B (2016) Algebraic identifiability of Gaussian mixtures. International Mathematics Research Notices.
- [5]. Aoyagi M (2010) A Bayesian learning coefficient of generalization error and Vander-monde matrix-type singularities. Communications in Statistics?Theory and Methods, 39(15), 2667–2687.
- [6]. Awange JL, Palancz B, Lewis R, Lovas T, Heck B and Fukuda Y (2016) An algebraic solution of maximum likelihood function in case of Gaussian mixture distribution. Australian Journal of Earth Sciences, 63(2), pp.193–203.
- [7]. Bates DJ, Hauenstein JD, Sommese AJ and Wampler CW (2013). Numerically solving polynomial systems with Bertini, 25, SIAM.
- [8]. Bates DJ, Hauenstein JD, Sommese AJ and Wampler CW Bertini: software for numerical algebraic geometry. Available at <https://bertini.nd.edu/>.
- [9]. Baudry JP (2015). Estimation and model selection for model-based clustering with the conditional classification likelihood. Electronic journal of statistics, 9(1), 1041–1077.
- [10]. Bernardi A, Daleo NS, Hauenstein JD and Mourrain B (2017). Tensor decomposition and homotopy continuation. Differential Geometry and its Applications.
- [11]. Biernacki C and Celeux G and Govaert G (2000). Assessing a mixture model for clustering with the integrated completed likelihood. IEEE transactions on pattern analysis and machine intelligence, 22(7), 719–725.
- [12]. Bishop CM (2006). Pattern recognition and machine learning. springer.
- [13]. Blundell R and Bond S (2000). GMM estimation with persistent panel data: an application to production functions. Econometric reviews, 19(3):321–340.
- [14]. Burnham KP and Anderson DR (2003). Model selection and multimodel inference: a practical information-theoretic approach. Springer Science & Business Media.
- [15]. Carreira-Perpinan M and Williams C (2003) An isotropic Gaussian mixture can have more modes than components. Technical Report EDI-INF-RR-0185, School of Informatics, U. Edinburgh.
- [16]. Carreira-Perpinan MA (2000). Mode-finding for mixtures of Gaussian distributions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11), 1318–1323.
- [17]. Chen S and Gopalakrishnan P (1998) February. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In Proc. darpa broadcast news transcription and understanding workshop 8:127–132.
- [18]. Cook ER and Peters K (1981). The smoothing spline: a new approach to standardizing forest interior tree-ring width series for dendroclimatic studies. Tree-ring bulletin.
- [19]. Del Campo AM and Rodriguez JI (2017). Critical points via monodromy and local methods. Journal of Symbolic Computation, 79:559–574.
- [20]. De Boor C, Mathmaticien EU, De Boor C and De Boor C (1978). A practical guide to splines, 27:325, New York: Springer-Verlag.

- [21]. Drton M and Plummer M (2017). A Bayesian information criterion for singular models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(2), 323–380.
- [22]. Drton M, Sturmfels B and Sullivant S (2008). *Lectures on algebraic statistics* (Vol. 39). Springer Science & Business Media.
- [23]. Eisenbud D, Grayson DR, Stillman M and Sturmfels B eds. (2001). *Computations in algebraic geometry with Macaulay 2* (Vol. 8). Springer Science & Business Media.
- [24]. Fraley C and Raftery AE (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458), 611–631.
- [25]. Fraley C and Raftery AE (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The computer journal*, 41(8):578–588.
- [26]. Grayson DR and Stillman ME *Macaulay 2*, a software system for research in algebraic geometry. Available at <http://www.math.uiuc.edu/Macaulay2/>.
- [27]. Gross E, Davis B, Ho KL, Bates DJ and Harrington HA (2016). Numerical algebraic geometry for model selection and its application to the life sciences. *Journal of The Royal Society Interface*, 13(123):20160256.
- [28]. Gross E and Rodriguez JI (2014) 7. Maximum likelihood geometry in the presence of data zeros. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, 232–239. ACM.
- [29]. Hastie T, Tibshirani R (1990). *Generalized additive models*. John Wiley & Sons, Inc..
- [30]. Hauenstein JD, Oeding L, Ottaviani G and Sommese AJ (2014). Homotopy techniques for tensor decomposition and perfect identifiability. *Journal für die reine und angewandte Mathematik (Crelles Journal)*.
- [31]. Hauenstein JD and Sottile F (2012). Algorithm 921: alphaCertified: certifying solutions to polynomial systems. *ACM Transactions on Mathematical Software (TOMS)*, 38(4):28.
- [32]. Hu S (2007). Akaike information criterion. *Center for Research in Scientific Computation*, 93.
- [33]. Hui FK, Warton DI and Foster SD (2015). Order selection in finite mixture models: complete or observed likelihood information criteria?. *Biometrika*, 102(3), 724–730.
- [34]. Hurvich CM and Tsai CL (1991). Bias of the corrected AIC criterion for underfitted regression and time series models. *Biometrika*, 78(3):499–509.
- [35]. Hurvich CM and Tsai CL (1993). A corrected Akaike information criterion for vector autoregressive model selection. *Journal of time series analysis*, 14(3):271–279.
- [36]. Jain AK and Dubes RC (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.
- [37]. Lee TL, Li TY, & Tsai CH (2008). HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method. *Computing*, 83(2–3), 109. Available at <http://www.hom4ps3.org>.
- [38]. Li J, Ray S, & Lindsay BG (2007). A nonparametric statistical approach to clustering via mode identification. *Journal of Machine Learning Research*, 8(Aug), 1687–1723.
- [39]. McLachlan GJ and Rathnayake S (2014). On the number of components in a Gaussian mixture model. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(5), 341–355.
- [40]. Moon TK (1996). The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6), 47–60.
- [41]. Pachter L and Sturmfels B eds. (2005). *Algebraic statistics for computational biology* (Vol. 13). Cambridge university press.
- [42]. Putinar M and Sullivant S eds. (2008). *Emerging applications of algebraic geometry* (Vol. 149). Springer Science & Business Media.
- [43]. Ray S and Lindsay BG (2005). The topography of multivariate normal mixtures. *The Annals of Statistics*, 33(5), pp.2042–2065.
- [44]. Rissanen J (1978). Modeling by shortest data description. *Automatica*, 14(5), 465–471.
- [45]. Ritchey RJ (1990). Call option valuation for discrete normal mixtures. *Journal of Financial Research*, 13(4):285–296.
- [46]. Sheather SJ and Jones MC (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 683–690.

- [47]. Sommese AJ and Wampler CW (2005). The Numerical solution of systems of polynomials arising in engineering and science. World Scientific.
- [48]. Verschelde J (1999). Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. ACM Transactions on Mathematical Software (TOMS), 25(2):251–276.
- [49]. Wang W, Wang H, Hempel M, Peng D, Sharif H and Chen HH (2011). Secure Stochastic ECG Signals Based on Gaussian Mixture Model for e-Healthcare Systems. IEEE Systems Journal, 5(4):564–573.
- [50]. Wu TJ and Sepulveda A (1998). The weighted average information criterion for order selection in time series and regression models. Statistics & probability letters, 39(1):1–10.
- [51]. Wu HC (2007). The Karush Kuhn Tucker optimality conditions in an optimization problem with interval-valued objective function. European Journal of Operational Research, 176(1), 46–59.
- [52]. Zhang Y, Brady M and Smith S (2001). Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE transactions on medical imaging, 20(1):45–57. [PubMed: 11293691]

Fig. 1.

An example of one dimensional GMM with three Gaussian components ($k = 3$) with parameters $\omega = (0.5, 0.25, 0.25)$, $\mu = (1, 5, 8)$ and $\Sigma = ([1], [1.25], [0.5])$.

Fig. 2.
Graphical representation of Steps 4–6 of area-based algorithm: Triangles with vertices $T=(TE, MD, EV)$.

Fig. 3.

Illustration of some of the steps of the area-based algorithm as described in Example 3.1.

Data for this example is generated from a one-dimensional GMM with $\omega = (0.5, 0.25, 0.25)$, $\mu = (1, 5, 8)$, $\Sigma = ([1], [1.25], [0.5])$. The 2000 training points are highlighted in blue and the 1000 test points are highlighted in red in Part (a).

Fig. 4.

Illustration of some of the steps of the local maxima algorithm (Algorithm II) as described in Example 3.2. Data for this example is generated from a one-dimensional GMM with $\omega = (0.5, 0.25, 0.25)$, $\mu = (1, 5, 8)$, $\Sigma = ([1], [1.25], [0.5])$. In this example, the dataset contains 3000 points.

Fig. 5. Scatter plot of some of the GMMs used in the simulated examples: (a)Example 1: One dimension-Three components, (b)Example 3: Two dimensions-Three components, (c)Example 5: Three dimensions-Three components, (d)Example 7: Five dimension-Three components.

Fig. 6. Pairwise scatter plot of the measurement errors, where diagonal cells represent the histogram of each of the 5 quality characteristics, and off-diagonal cells show the scatter plots of each pair of quality characteristics.

Fig. 7. Scatter plot of the NGMMs used in the simulated examples: (a)One dimension, (b)Two dimensions, (c)Three dimensions

Table 1.

The parameters of the GMMs used in simulated experiments.

Example	Dimension	Component	GMM Parameters			
			MU	SIG	Pr	n
1	1	3	[1 5 8]	[1 1.25 .5]	[.5 .25 .25]	500
2	2	2	$\begin{bmatrix} 1.2 & 3.2 \\ 3.4 & 2.3 \end{bmatrix}$	$\begin{bmatrix} .3 & .1 \\ .1 & .3 \end{bmatrix}, \begin{bmatrix} .25 & -.1 \\ -.1 & .25 \end{bmatrix}$	[.5 .5]	2500
3	2	3	$\begin{bmatrix} .5 & 3 \\ 2 & 1 \\ 2.7 & 3.1 \end{bmatrix}$	$\begin{bmatrix} .5 & -.2 \\ -.2 & .3 \end{bmatrix}, \begin{bmatrix} .65 & .1 \\ .1 & .25 \end{bmatrix}, \begin{bmatrix} .3 & -.1 \\ -.1 & .25 \end{bmatrix}$	[.35 .35 .3]	2500
4	3	2	$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 3 & 3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} .75 & .1 & .1 \\ .1 & .75 & .1 \\ .1 & .1 & .75 \end{bmatrix}$	[.5 .5]	8000
5	3	3	$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 3 & 3 \\ 0 & 2 & 2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} .75 & .1 & .1 \\ .1 & .75 & .1 \\ .1 & .1 & .75 \end{bmatrix}, \begin{bmatrix} .5 & -.12 & -.12 \\ -.12 & .5 & -.12 \\ -.12 & -.12 & .5 \end{bmatrix}$	[.3 .4 .3]	8000
6	5	2	$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 3 & 2 & 3 & 2 & 3 \end{bmatrix}$	$\begin{bmatrix} .5 & 0 & 0 \\ 0 & .5 & 0 \\ 0 & 0 & .5 \end{bmatrix}, \begin{bmatrix} .75 & 0 & 0 \\ 0 & .75 & 0 \\ 0 & 0 & .75 \end{bmatrix}$	[.5 .5]	16807
7	5	3	$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 3 & 3 & 3 & 3 & 3 \\ 0 & 2 & 0 & 2 & 0 \end{bmatrix}$	$\begin{bmatrix} .5 & 0 & 0 \\ 0 & .5 & 0 \\ 0 & 0 & .5 \end{bmatrix}, \begin{bmatrix} .75 & 0 & 0 \\ 0 & .75 & 0 \\ 0 & 0 & .75 \end{bmatrix}, \begin{bmatrix} .5 & 0 & 0 \\ 0 & .5 & 0 \\ 0 & 0 & .5 \end{bmatrix}$	[.3 .3 .4]	16807

Table 2.

The parameters of NGMMs used in the simulated experiments.

Example	Dimension	Component	Distribution/Parameter	Pr	n
1	2	2	$X C_1 \sim \text{Weib}(\alpha = 50, \beta = 1)$ $X C_2 \sim \text{Weib}(\alpha = 500, \beta = 3)$	[.5 .5]	100
2	2	2	$X C_1 \sim \log N\left(\mu = [.25 \ .25], \Sigma = \begin{bmatrix} .6 & .3 \\ .3 & .6 \end{bmatrix}\right)$ $X C_2 \sim N\left(\mu = [5 \ 6], \Sigma = \begin{bmatrix} .7 & -.2 \\ .2 & .8 \end{bmatrix}\right)$	[.5 .5]	400
3	3	3	$X C_1 \sim N\left(\mu = [11 \ 11 \ 5], \Sigma = \begin{bmatrix} .4 & .01 & .01 \\ .01 & .5 & .05 \\ .01 & .05 & .4 \end{bmatrix}\right)$ $X C_2 \sim \log N\left(\mu = [.2 \ .25 \ .3], \Sigma = \begin{bmatrix} .6 & .01 & .01 \\ .01 & .5 & .01 \\ .01 & .01 & .4 \end{bmatrix}\right)$ $X C_3 \sim N\left(\mu = [6 \ 4 \ 11], \Sigma = \begin{bmatrix} .8 & -.01 & -.01 \\ -.01 & .9 & -.01 \\ -.01 & -.01 & .8 \end{bmatrix}\right)$	[.33 .33 .33]	1500

Table 3.

The estimated number of NGMM components identified by each method.

Example	Algorithm I Area-based	Algorithm II Local maxima	AIC	BIC
1	1	2	3	4
2	2	2	4	4
3	2	3	6	6