Learning a Mixture of Gaussians via Mixed Integer Optimization

Hari Bandi* Dimitris Bertsimas† Rahul Mazumder†

October 15, 2018

Abstract

We consider the problem of estimating the parameters of a multivariate Gaussian mixture model (GMM) given access to n samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ that are believed to have come from a mixture of multiple subpopulations. State-of-the-art algorithms used to recover these parameters use heuristics to either maximize the log-likelihood of the sample or try to fit first few moments of the GMM to the sample moments. In contrast, we present here a novel Mixed Integer Optimization (MIO) formulation that optimally recovers the parameters of the GMM by minimizing a discrepancy measure (either the Kolmogorov-Smirnov or the Total variation distance) between the empirical distribution function and the distribution function of the GMM whenever the mixture component weights are known. We also present an algorithm for multidimensional data that optimally recovers corresponding means and covariance matrices. We show that the MIO approaches are practically solvable for datasets with n in the tens of thousands in minutes and achieve an average improvement of 60-70% and 50-60% on mean absolute percentage error (MAPE) in estimating the means and the covariance matrices, respectively over the EM algorithm independent of the sample size n. As the separation of the Gaussians decrease and correspondingly the problem becomes more difficult the edge in performance in favor of the MIO methods widens. Finally, we also show that the MIO methods outperform the EM algorithm with an average improvement of 4-5% on the out-of-sample accuracy for real-world datasets.

1 Introduction

Finite mixture modeling is a widely used approach to modeling data that is believed to arise from multiple heterogeneous subpopulations, such as data from pattern recognition, computer vision and machine learning. A Gaussian mixture model (GMM) is an important mixture model family which is useful for modeling data that comes from one of several Gaussian distributions. Consider a set of K different univariate Gaussian distributions, with each distribution being defined by a mean $\mu_i \in \mathbb{R}$, and a variance $\sigma_i^2 \in \mathbb{R}$. Letting f_i denote the Gaussian density function of the i^{th} component $\mathcal{N}(\mu_i, \sigma_i^2)$, the density function of the mixture is given by $f = \sum_{i=1}^{K} \pi_i f_i$ where π is

*Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, Email: hbandi@mit.edu

†Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, Email: {dbertsim,rahulmaz}@mit.edu

the vector of mixture component weights that sum to one $\left(\pi^T e = 1\right)$ so that the total probability distribution normalizes to 1.

The most widely used algorithm for recovering estimates of the parameters of a GMM in practice is the EM algorithm published in Dempster et al. (1977). This algorithm is a local search heuristic that alternates between optimizing over the Gaussians' parameters $\{(\mu_1, \sigma_1), (\mu_2, \sigma_2), \ldots, (\mu_K, \sigma_K)\}$ and the component mixing weights $\{\pi_1, \pi_2, \ldots, \pi_K\}$ and converges to a set of parameters that locally maximize the likelihood of observing the data sample. Wu (1983) established guarantees that the solution of the EM algorithm converges to the maximum likelihood estimates when the maximum likelihood function is unimodal but in practice, the maximum likelihood function is usually multimodal and these guarantees are not valid anymore. Balakrishnan et al. (2017) proved statistical guarantees on the convergence of the EM algorithm solution to a local optimum that is within a *statistical precision* to the global optimum using suitable initializations.

Apart from maximizing the sample likelihood, various other algorithms have been proposed in the literature to efficiently estimate the parameters of a GMM. Given n samples Dasgupta (1999) proposed a method to provably recover good estimates for the parameters in polynomial time in n. Their technique is based on projecting data down to a randomly chosen low-dimensional subspace and then finding an accurate clustering so that the empirical means and co-variances of these clustered points would be a good estimate for the actual parameters. Sanjeev and Kannan (2001) extended these ideas to work in a more general setting in which the co-variances of each Gaussian component could be arbitrary, and not necessarily spherical as in Dasgupta (1999). Yet both of these techniques are based on the concentration of distances under random projections, and consequently required that the centers of the components be separated by at least a constant factor of $(\max_i \sigma_i)\sqrt{d}$ (d is the dimension of the data). Vempala and Wang (2002a) introduced the use of spectral techniques, to choose a subspace on which to project based on large principle components and propose an algorithm that needs the Gaussian componenets in the mixture to be separated by at least $(\max_i \sigma_i)\sqrt{K}$.

Yet all of these approaches for learning good estimates require that each pair of Gaussian components be separated by some factor of the maximum standard deviation $(\max_i \sigma_i)$. A series of works in the literature have also looked at the moment matching problem for a GMM. Belkin and Sinha (2009) showed that one can efficiently learn GMMs in the special case that all components are identical spherical Gaussians using the Method of Moments. Similarly, Kalai et al. (2010), Moitra and Valiant (2010) proposed an algorithm that searches over the space of parameters of GMM to fit the first six moments of the observed data. In constrast, our objective in this paper is to estimate the GMM distribution function f characterized by the set of parameters, $\theta = \{(\pi_1, \mu_1, \sigma_1), (\pi_2, \mu_2, \sigma_2), \ldots, (\pi_K, \mu_K, \sigma_K)\}$, so that the cumulative distribution functions of the GMM and the empirical distribution are close, i.e., $D(F, \hat{F_n}) \leq \epsilon$ where F is the cumulative distribution function (CDF) of the GMM, and $\hat{F_n}$ is the empirical cumulative distribution function and $D(\cdot, \cdot)$ is some discrepancy measure. Specifically, we use two discrepancy measures namely

the Kolmogorv Smirnov and the Total variation distance to quantify the distance between the two distributions and recover parameters of the GMM that optimally minimize these discrepancy measures. The Mixed Integer Optimization(MIO) problems that we present in this paper are not only less sensitive to pairwise distances between Gaussian components, but also are tractable and solve problems of large sizes (n in tens of thousands) in minutes due to significant improvement in speedups of MIO solvers in the last two decades.
We summarize our contributions in this paper below:

1. We present two novel MIO formulations for optimally recovering the parameters of a one-dimensional Gaussian Mixture Model (GMM) that minimize a discrepancy between the empirical distribution function and the distribution function of the GMM. We achieve this by formulating the problem of minimizing the Kolmogorov-Smirnov (KS) distance and the Total Variation (TV) distance as MIO problems. We use a piecewise linear function to approximate the standard normal CDF in our MIO formulations. We also present a novel MIO formulation to find an optimal set of breakpoints for approximating the standard normal CDF using a piecewise linear function that minimizes the maximum approximation error between the piecewise linear function and the standard normal CDF.

2. We present an algorithm for d-dimensional data that uses ideas from random projections, and makes use of the univariate algorithm to optimally recover the model parameters in higher dimensions. We also propose a Mixed Integer Quadratic Optimization (MIQO) problem and a Semidefinite Optimization (SDO) problem to correctly identify a consistent ordering among the estimates recovered across the d-dimensions of the model parameters.

3. We perform computational experiments on synthetic datasets generated using various assumptions and demonstrate that the proposed MIO problems are tractable for datasets of sizes in the tens of thousands and solves for the parameters to provable optimality. We show that the MIO approaches achieve an average improvement of 60-70% and 50-60% on mean absolute percentage error (MAPE) in estimating the means and the covariance matrices, respectively over the EM algorithm independent of the sample size n. As the separation of the Gaussians decrease and correspondingly the problem becomes more difficult the edge in performance in favor of the MIO methods widens. We also show that the MIO methods outperform the EM algorithm with an average improvement of 4-5% on the out-of-sample accuracy for real-world datasets.

The rest of the paper is structured as follows. In Section 2, we review Gaussian mixture modeling and formulate the problem of minimizing the discrepancy between the empirical distribution function and the distribution function of the GMM as a MIO problem for the univariate case by using the Kolmogorov-Smirnov (KS) distance and the Total Variation (TV) distance as discrepancy measures. We also present a novel mixed integer optimization problem to find an optimal set

of breakpoints for approximating the standard normal CDF with a piecewise linear function. In Section 3, we present an algorithm for multidimensional Gaussian mixture models by using ideas from random projections and the univariate algorithm proposed in Section 2. In Section 4, we perform computational experiments using various synthetic and real-world datasets to evaluate the performance of our method against state-of-the-art methods like the EM algorithm. In Section 5, we discuss some implications of this work and make some concluding remarks.

2 One-Dimensional Gaussian Mixture Modeling

In this section, we first give an overview of one-dimensional Gaussian mixture modeling. We then present two novel MIO formulations allowing us to solve the problem of minimizing a discrepancy (either the Kolmogorv-Smirnov distance or the Total variation distance) between empirical distribution function and the distribution function of the GMM to optimality when the mixture component weights π are known. Unlike Belkin and Sinha (2009), Dasgupta (1999) and Kannan et al. (2005), our univariate algorithm is less sensitive to separation between the Gaussian components and we do not make any assumptions on the degree of separation between Gaussian components.

Formally, a Gaussian mixture model (GMM) is a convex combination of K different one-dimensional Gaussians with weights $\pi_i \in [0,1]$ such that $\left(\sum_{i=1}^{K} \pi_i = 1\right)$, means $\mu_i \in \mathbb{R}$ and variances $\sigma_i^2 \in \mathbb{R}$. Letting $f_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ denote the distribution of the i^{th} Gaussian component of the mixture, the density of the GMM is given by $f = \sum_{i=1}^{K} \pi_i f_i$. We are interested in estimating the GMM distribution function, f characterized by the set of parameters, $\theta = \{(\pi_1, \mu_1, \sigma_1), (\pi_2, \mu_2, \sigma_2), \ldots, (\pi_k, \mu_k, \sigma_k)\}$, so that the cumulative distribution functions are close ($F \approx \hat{F_n}$ or equivalently $D(F, F_n) \leq \epsilon$) where F is the CDF of the GMM and F_n is the empirical distribution function and $D(\cdot, \cdot)$ is a discrepancy measure (either the Komogorov Smirnov distance or the Total variation distance) between two distribution functions.

2.1 Minimizing discrepancy based on the Kolmogorov-Smirnov distance

In this section, we introduce the Kolmogorov-Smirnov distance between any two distributions and incorporate this discrepancy measure into the problem of estimating the parameters of a GMM when the mixture component weights are known. In order to recover these parameters, we seek to minimize the Kolmogorov-Smirnov distance between the empirical cumulative distribution function $F_n(x)$ and the cumulative distribution function of the GMM $F(x)$.

The Kolmogorov-Smirnov distance (Massey Jr, 1951) between any two distributions $F(x)$ and $G(x)$ is given by

$$D_{KS}(F, G) = \sup_x |F(x) - G(x)|.$$

Simiarly, the Kolmogorov-Smirnov distance between an empirical distribution function $F_n(x)$ on $\{x_1, x_2, \ldots, x_n\}$ (where we assume without loss of generality that the sample is ordered and non-

decreasing) and any other distibution function $F(x)$ is defined as

$$D_{KS}(F_n, F) \quad = \quad \max_{x \in \{x_1, x_2, \ldots, x_n\}} |F_n(x) - F(x)| = \max_{i \in \{1,2,\ldots,n\}} \left| \frac{i}{n} - F\left(x_i\right) \right|.$$

Recall that the cumulative distribution function of the GMM $F(x)$ is given by,

$$F(x) \quad = \quad \sum_{i=1}^{K} \pi_i F_i\left(x\right) = \sum_{i=1}^{K} \pi_i \Phi\left(\frac{x - \mu_i}{\sigma_i}\right),$$

where F_i is the CDF of the i^{th} Gaussian component $\mathcal{N}(\mu_i, \sigma_i^2),\ i \in \{1, 2, \ldots, K\}$. We thus propose to solve the following MIO problem in order to estimate the parameters $(\mu_1, \sigma_1), (\mu_2, \sigma_2), \ldots, (\mu_K, \sigma_K)$.

$$\min_{\{\mu_i, \sigma_i\}_{i=1}^{K}} \max_{j \in \{1,2,\ldots,n\}} \left| \frac{j}{n} - \sum_{i=1}^{K} \pi_i \Phi\left(\frac{x_j - \mu_i}{\sigma_i}\right) \right|. \tag{1}$$

Since the standard normal CDF $\Phi(\cdot)$ does not admit a closed form representation and it is neither a convex nor a concave function, we incorporate a piecewise linear approximation so that Problem (1) can be reformulated as a MIO problem.

2.1.1 A piecewise linear approximation to the standard normal CDF.

In order to reformulate Problem (1) as an MIO problem, we first define auxillary variables $s_j = 1/\sigma_j, t_j = \mu_j/\sigma_j, j = 1, 2, \ldots, K$ so that we eliminate nonlinear terms in the expression, $\dfrac{x_j - \mu_i}{\sigma_i}$. Therefore, we seek to solve the following MIO problem:

$$\min_{\{t_i, s_i\}_{i=1}^{K}} \max_{j \in \{1,2,\ldots,n\}} \left| \frac{j}{n} - \sum_{i=1}^{K} \pi_i \Phi\left(s_i x_j - t_i\right) \right|. \tag{2}$$

Observe that since the standard normal CDF does not admit a closed-form representation, we need to use some approximation to the CDF in Problem (2). Specifically, we use the closed-form approximations proposed in (Tocher, 1967, Zelen and Severo, 1964), and solve the corresponding non-linear and non-convex problems using Baron commercial solver. However, these methods do not scale well for large problems (See Table 3), therefore, we propose to use a piecewise linear approximation to the standard normal CDF so that the complete problem can be reformulated as a linear MIO problem.

A piecewise linear function is composed of a series of line segments joining a set of predefined break-points. In lemma 1, we provide a bound on the objective function of Problem (2) when we approximate the standard normal CDF by a piecewise linear function $L(\cdot)$.

Lemma 1. *The objective of Problem (18) is related to the objective of the problem (4) as follows:*

$$\max_{i\in\{1,2,\ldots,n\}}\left|\frac{i}{n}-\sum_{j=1}^{K}\pi_j\Phi\left(\frac{x_i-\mu_j^t}{\sigma_j^t}\right)\right| \quad\leq\quad \max_{i\in\{1,2,\ldots,n\}}\left|\frac{i}{n}-\sum_{j=1}^{K}\pi_j L\left(\frac{x_i-\mu_j^t}{\sigma_j^t}\right)\right|+\epsilon^{\star}_{PWL}$$

where $\epsilon^{\star}_{PWL}$ is the maximum absolute approximation error between the standard normal CDF $\Phi(\cdot)$ and a piecewise linear approximation function $L(\cdot)$.

We construct an approximation function to the standard normal CDF using binary variables as follows. First, denote $v_1 < v_2 < \ldots < v_p$ as the break-points for approximating $\Phi(\cdot)$. Note that, the approximation error when using a piecewise linear function depends both on the number of break-points used and also how these break-points are chosen. Trivially, as we increase the number of break-points, the approximation error decreases while the computational burden increases.

Since the approximation error also depends on the location of these predefined p break-points, we formulate the problem of finding an optimal set of break-points that minimizes the total maximum approximation error across all of the linear pieces as a shortest path problem in Section 2.1.2. In Figures (1a) and (1b) we plot the piecewise linear approximations to the standard normal CDF with 5 and 10 linear pieces obtained as solutions of solving the shortest path problem (6).

Once we have an optimal set of break-points $v_1 < v_2 < \ldots < v_p$, the function $\Phi(\cdot)$ can then be approximated using a piecewise linear function $L(x)$ over the interval $[v_1, v_p]$ as follows,

$$\begin{aligned}
L(x) &= \sum_{k=1}^{p}\Phi(v_k)y_k \qquad (3)\\
x &= \sum_{k=1}^{p}v_k y_k\\
y_1 &\leq z_1\\
y_k &\leq z_{k-1}+z_k,\ k\in\{2,\ldots,p-1\}\\
y_p &\leq z_{p-1}\\
\sum_{k=1}^{p-1}z_k &= 1\\
\sum_{k=1}^{p}y_k &= 1\\
z_k &\in \{0,1\},\ k\in\{1,\ldots,p-1\}\\
y_k &\geq 0,\ k\in\{1,\ldots,p-1\}.
\end{aligned}$$

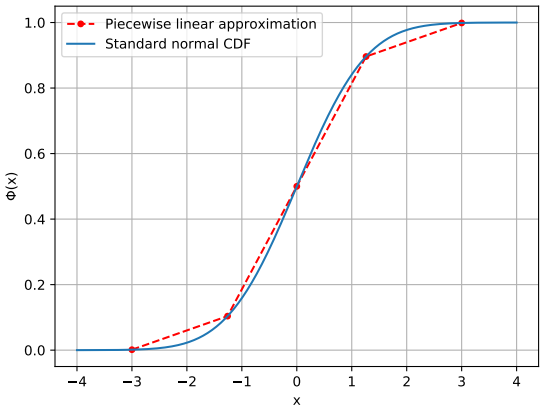
where the binary variables $\{z_i\}_{i=1}^{p-1}$ are defined as

$$z_i = \begin{cases} 1, & \text{if } x \in [v_i, v_{i+1}], \\ 0, & \text{otherwise.} \end{cases}$$

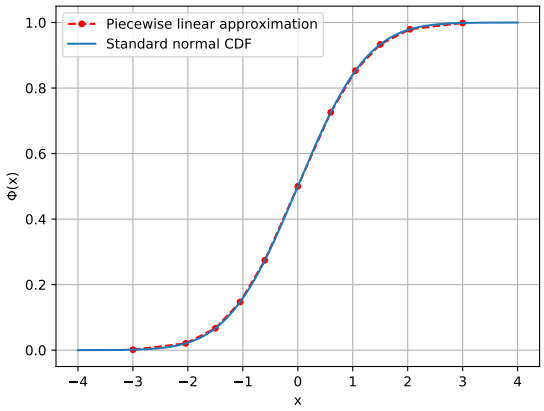
Observe that when $x \in [v_i, v_{i+1}]$, the value of $\Phi(x)$ is approximated by a weighted average of $\Phi(v_i)$ and $\Phi(v_{i+1})$ which is captured by the variables $\{y_i, z_i\}_{i=1}^{p}$. Combining Problem (2) and Equations (3) we obtain the following MIO problem:

$$\begin{aligned}
& \min_{\{t_i, s_i\}_{i=1}^{K}} \ \epsilon \qquad (4) \\
\text{s.t.} \quad \epsilon &\geq \frac{j}{n} - \sum_{i=1}^{K} \pi_i \sum_{k=1}^{p} \Phi(v_k) y_{i,j}^k, \ j \in \{1, 2, \ldots, n\} \\
-\epsilon &\leq \frac{j}{n} - \sum_{i=1}^{K} \pi_i \sum_{k=1}^{p} \Phi(v_k) y_{i,j}^k, \ j \in \{1, 2, \ldots, n\} \\
s_i x_j - t_i &= \sum_{k=1}^{p} v_k y_{i,j}^k, \ j \in \{1, 2, \ldots, n\}, \ i \in \{1, 2, \ldots, K\} \\
y_{i,j}^1 &\leq z_{i,j}^1, \ i \in \{1, 2, \ldots, K\}, \ j \in \{1, 2, \ldots, n\} \\
y_{i,j}^k &\leq z_{i,j}^{k-1} + z_{i,j}^k, \ i \in \{1, 2, \ldots, K\}, \ j \in \{1, 2, \ldots, n\}, \ k \in \{2, \ldots, p-1\} \\
y_{i,j}^p &\leq z_{i,j}^{p-1}, \ i \in \{1, 2, \ldots, K\}, \ j \in \{1, 2, \ldots, n\} \\
\sum_{k=1}^{p-1} z_{i,j}^k &= 1, \ i \in \{1, 2, \ldots, K\}, \ j \in \{1, 2, \ldots, n\} \\
\sum_{k=1}^{p} y_{i,j}^k &= 1, \ i \in \{1, 2, \ldots, K\}, \ j \in \{1, 2, \ldots, n\} \\
s_i &\geq 0, \ i \in \{1, 2, \ldots, K\} \\
z_{i,j}^k &\in \{0, 1\}, \ i \in \{1, 2, \ldots, K\}, \ j \in \{1, 2, \ldots, n\}, \ k \in \{1, 2, \ldots, p-1\} \\
y_{i,j}^k &\geq 0, \ i \in \{1, 2, \ldots, K\}, \ j \in \{1, 2, \ldots, n\}, \ k \in \{1, 2, \ldots, p-1\}.
\end{aligned}$$

Finally, the means and the variances can be retrieved from the optimal solution $(s_j^\star, t_j^\star)$ as $\mu_j = \frac{t_j^\star}{s_j^\star}$ and $\sigma_j = \frac{1}{s_j^\star}$. Observe that since the coordinates x_i are given to be non-decreasing, the first term inside the absolute value of each of the constraints $\epsilon \geq \left| \frac{i}{n} - \sum_{j=1}^{K} \pi_j \Phi\left(s_j x_i - t_j\right) \right|$, $i \in \{1, 2, \ldots, n\}$ is increasing with i, therefore since the CDF is a monotonic function, the optimal solution $s_j^\star$ has to be positive (and not zero) and sufficiently large so that for each x_i, the value of the second term $\sum_{j=1}^{K} \pi_j \Phi\left(s_j x_i - t_j\right)$ increases with i as well.



(a) PWL approximation with 5 break-points.



(b) PWL approximation with 10 break-points.

Figure 1: Optimal piecewise linear approximations for the standard normal CDF.

Observe that the computational burden on the solver arises from the constraints of the type,

$$\epsilon \geq \left| \frac{j}{n} - \sum_{i=1}^{K} \pi_i \sum_{k=1}^{p} \Phi(v_k) y_{i,j}^k \right|, \; j \in \{1, 2, \ldots, n\}, \tag{5}$$

as each of these constraints links $K(p-1)$ binary variables $z_{i,j}^k,\ i \in \{1, 2, \ldots, K\}, k \in \{1, 2, \ldots, p-1\}$ with a shared variable ϵ in the MIO formulation. Note that since we are minimizing the maximimum absolute difference in the CDFs at n points; at the optimal solution the majority of these constraints may not be binding unless the solution is highly degenerate. Therefore, in order to accelerate the solution of the MIO problem (4), we generate a subset of these constraints dynamically using a greedy strategy as illustrated in Section 2.1.3, rather than defining all the constraints from Equation (5) upfront.

Note that since the objective function in the KS model contains the empirical CDF, the model is not as robust as the EM-algorithm to addition/removal of a small set of data.

2.1.2 Optimal set of breakpoints over a discretized grid for piecewise linear approximation.

The accuracy and computational complexity of our univariate algorithm depends on how well we approximate the standard normal CDF using as low a number of binary variables as possible. Therefore, it is important to have an optimal (by minimizing the total approximation error) piecewise linear approximation to the standard normal CDF for a given number of break-points. In this section, we formulate the problem to find an optimal set of p break-points that minimizes the total sum of the maximum approximation error in each piece between a piecewise linear function using $(p-1)$ linear pieces and the standard normal CDF $\Phi(\cdot)$ as a shortest path problem on a network.

First, we discretize the interval $[-3, 3]$ which spans almost 99.7% of the probability density of the standard normal distribution. We define a uniform grid of m points $\mathcal{X} = \{u_i\}_{i=1}^m$ uniform over $[-3, 3]$ and formulate a shortest path problem to choose p breakpoints from $\mathcal{X}$ that minimize the total sum of the maximum approximation error across all of the $p-1$ linear pieces.

We define a directed acyclic graph $G(V, E)$ such that the discretized set of points $\mathcal{X}$ are the set of nodes V and each pair of edges $(u_i, u_j) \in E$ has a cost of c_{ij} which is equal to the maximum approximation error between the standard normal CDF and the line segment joining $\Phi(u_i)$ and $\Phi(u_j)$, i.e,

$$c_{ij} = \max_{x \in [u_i, u_j]} \left| \Phi(x) - \left(\Phi(u_i) + \frac{\Phi(u_j) - \Phi(u_i)}{u_j - u_i}(x - u_i) \right) \right|, \ i < j.$$

Note that the maximum approximation error between the curve $\Phi(\cdot)$ and the line segment joining $\Phi(u_i)$ and $\Phi(u_j)$ occurs when the slope of the line segment and the curve are the same, i.e,

$$s = \frac{\Phi(u_j) - \Phi(u_i)}{u_j - u_i} = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Therefore, the maximum approximation error occurs at $x_{ij}^\star = \pm\sqrt{-\log(2\pi) - 2\log(s)} \in [u_i, u_j]$. The cost c_{ij} for each pair of edges $(u_i, u_j) \in E$ is given by,

$$c_{ij} = \begin{cases} \left| \Phi(x_{ij}^\star) - \left(\Phi(u_i) + \frac{\Phi(u_j) - \Phi(u_i)}{u_j - u_i}(x_{ij}^\star - u_i) \right) \right|, \ i < j \\ \infty, \ \text{otherwise.} \end{cases}$$

The problem is now to find a directed path of length $p-1$ from $u_1 = -3$ to $u_m = 3$ with the smallest cost. We can solve this problem using dynamic programming as follows. We define $D(k, u)$ to be the cost of the shortest path of length k from node u to node u_m. Therefore, we have the following recursion:

$$D(k+1, u_i) = \min_{j > i} \{c_{ij} + D(k, u_j)\}.$$

Finally, the optimal cost of a path of length $p-1$ from u_1 to u_m is given by $D(p-1, u_1)$. Now we define the "maximum absolute approximation error" between the standard normal CDF and the optimal PWL approximation $L(\cdot)$ as

$$\epsilon_{\text{PWL}}^\star = \max_x |\Phi(x) - L(x)|. \tag{6}$$

During the solution process, we solve the shortest path problem only once to find an optimal subset of break-points of $\mathcal{X}$ and use these break-points in formulation (4). The shortest path problem to find an optimal set of 10 break-points from a discretized set $\mathcal{X}$ of size $m = 1000$ can be solved under a second. In figures 1a, 1b, we plot the optimal PWL approximations with five and ten

break-points respectively. With as low as 10 break-points we have an optimal PWL approximation with a maximum approximation error of $\epsilon^{\star}_{PWL} = 0.00059$.

2.1.3 Dynamic constraint generation.

As previously mentioned, the computational burden to solve the MIO problem (4) arises from the constraints of the type,

$$\epsilon \geq \left| \frac{j}{n} - \sum_{i=1}^{K} \pi_i \sum_{k=1}^{p} \Phi(v_k) y_{i,j}^{k} \right|, \; j \in \{1, 2, \ldots, n\}, \tag{7}$$

as each of these constraints links $K(p-1)$ binary variables $z_{i,j}^{k}, \; i \in \{1, 2, \ldots, K\}, k \in \{1, 2, \ldots, p-1\}$ with a shared variable ϵ in the MIO formulation. Note that since we are minimizing the maximimum absolute difference in the CDFs at n points; at the optimal solution majority of these constraints may not be binding. Therefore, rather than defining all the constraints from Equation (7) upfront, we generate a subset of these constraints dynamically using a greedy strategy to accelerate the solution of the MIO problem as follows.

We maintain a dynamic set of indices $\mathcal{I}$ for which we have constraints

$$\epsilon \geq \left| \frac{j}{n} - \sum_{i=1}^{K} \pi_i \sum_{k=1}^{p} \Phi(v_k) y_{i,j}^{k} \right|, \; j \in \mathcal{I},$$

in the MIO formulation. Whenever the solver finds an integer feasible solution for the MIO with the current set of indices $\mathcal{I}$, we use the current solution $(\mu_1, \sigma_1), \ldots, (\mu_k, \sigma_k)$ to check if all the constraints in (7) are satisfied by calculating the maximum discrepancy over the remaining set of indices as:

$$j^{\star} = arg \max_{j \in \mathcal{N} \setminus \mathcal{I}} \left| \frac{j}{n} - \sum_{i=1}^{K} \pi_i \Phi\left(\frac{x_j - \mu_i}{\sigma_i} \right) \right|,$$

where $\mathcal{N} = \{1, 2, \ldots, n\}$. We update the set of indices $\mathcal{I} = \{j^{\star}\} \cup \mathcal{I}$ and add a constraint for $j^{\star}$ if

$$\max_{j \in \mathcal{N} \setminus \mathcal{I}} \left| \frac{j}{n} - \sum_{i=1}^{K} \pi_i \Phi\left(\frac{x_j - \mu_i}{\sigma_i} \right) \right| > \epsilon$$

i.e, the constraint at $j = j^{\star}$ violates an inequality from (7). We do this using lazy constraint callbacks available in Gurobi 6.5 and CPLEX 12.3 that allow a user to add user cuts whenever the solver finds an integer feasible solution.

2.2 Minimizing discrepancy based on the Total Variation distance

In this section, we introduce the Total Variation distance between any two distributions and extend the approach in Section 2.1 to the total Variation distance to estimate the parameters of a GMM for the case when the mixture component weights π are known.

The Total Variation distance between any two distributions P and Q on a σ-algebra $\mathcal{F}$ is defined as the supremum of the difference between the probability of P and Q over all the Borel sets $\mathcal{A} \in \mathcal{F}$ given by

$$D_{TV}(F, G) = \sup_{A \in \mathcal{F}} |P(A) - Q(A)|.$$

To make the problem tractable, we choose a subset $\mathcal{J}$ of the σ-algebra $\mathcal{F}$ such that $\mathcal{J} = \{(l_1, u_1), (l_2, u_2), \ldots, (l_m, u_m)\} \subset \mathcal{F}$ and minimize the Total variation distance on this set $\mathcal{J}$. Therefore, the distance metric that we consider is as follows:

$$D_{TV}(F, G) = \max_{A \in \mathcal{J}} |\mathbb{P}(X \in A) - \mathbb{P}(Y \in A)|,$$

where $X \sim P, Y \sim Q$.

We propose two different approaches of choosing the subset $\mathcal{J} \subset \mathcal{F}$. In the first approach, we choose dynamic intervals that are centered around the means of each of the Gaussian components. Recall that the density of a Gaussian distribution is centered around its mean, therefore considering intervals centered around the mean would capture the high probability density intervals of a Gaussian. We thereofore choose multiple intervals centered around the means of each Gaussian component so that $\mathcal{J} = \{(\mu_j - \delta\sigma_j, \mu_j + \delta\sigma_j) |\ i = 1, \ldots, K,\ \delta = 1, 2, 3\}$. We propose to solve the following MIO problem:

$$\min_{\mu_i, \sigma_i} \max_{\substack{\delta \in \{1,2,3\} \\ j \in \{1,2,\ldots,K\}}} |P_n\left((\mu_j - \delta\sigma_j, \mu_j + \delta\sigma_j)\right) - P\left((\mu_j - \delta\sigma_j, \mu_j + \delta\sigma_j)\right)|, \tag{8}$$

where $P_n(\cdot)$ is the probability measure of the empirical distribution and $P(\cdot)$ is the probability measure of the GMM. To reformulate this problem as a MIO problem, we make use of the binary variables to keep track of the count of number of samples that lie in the interval of size $\delta\sigma_i$ around the mean μ_i. As in the previous section, the univariate algorithm proposed here assumes that mixture component weights π are known and we later propose an alternating optimization approach to estimate both the component weight π and the parameters of the GMM.

As defined earlier, let F_i denote the cumulative distribution function for i^{th} Gaussian component with weight π_i. Therefore, the CDF of the mixture of Gaussians F is given by: $F(x) = \sum_{i=1}^{n} \pi_i F_i(x)$. Therefore, the probability of the GMM inside the interval $(\mu_j - \delta\sigma_j, \mu_j + \delta\sigma_j)$ is given by:

$$P\left((\mu_j - \delta\sigma_j, \mu_j + \delta\sigma_j)\right) \quad = \quad F(\mu_j + \delta\sigma_j) - F(\mu_j - \delta\sigma_j) \tag{9}$$

$$
\begin{aligned}
&= \sum_{k=1}^{K} \pi_k \left(F_k(\mu_j + \delta\sigma_j) - F_k(\mu_j - \delta\sigma_j) \right) \\
&= \sum_{k=1}^{K} \pi_k \left(\Phi\left(\frac{\mu_j + \delta\sigma_j - \mu_k}{\sigma_k} \right) - \Phi\left(\frac{\mu_j - \delta\sigma_j - \mu_k}{\sigma_k} \right) \right).
\end{aligned}
$$

To keep track of the count of the number of samples that fall in the interval $(\mu_j - \delta\sigma_j, \mu_j + \delta\sigma_j)$, we define binary variables $a_{i,j}^{\delta}$ such that,

$$
a_{i,j}^{\delta} = \begin{cases} 1, & \text{if } x_i \in (\mu_j - \delta\sigma_j, \mu_j + \delta\sigma_j), \\ 0, & \text{otherwise.} \end{cases}
$$

We model the above constraints on the binary variables $a_{i,j}^{\delta}$ as follows:

$$
\begin{aligned}
a_{i,j}^{\delta} x_i &\leq a_{i,j}^{\delta} \mu_j + \delta \cdot \sigma_j, \\
a_{i,j}^{\delta} x_i &\geq a_{i,j}^{\delta} \mu_j - \delta \cdot \sigma_j.
\end{aligned} \tag{10}
$$

Observe that the above constraints are satisfied for any data point which has $a_{i,j}^{\delta} = 0$. In order to make the count of the number of data points inside the intervals accurate, we add the following constraints to the formulation:

$$
(1 - a_{i,j}^{\delta}) \left| x_i - \mu_j \right| \geq (1 - a_{i,j}^{\delta}) \delta \cdot \sigma_j. \tag{11}
$$

Since the probability of the empirical distribtuion inside the interval $(\mu_j - \delta\sigma_j, \mu_j + \delta\sigma_j)$ is given by the proportion of samples that fall in this interval, we have

$$
P_n \left((\mu_j - \delta\sigma_j, \mu_j + \delta\sigma_j) \right) = \sum_{i=1}^{n} \frac{a_{i,j}^{\delta}}{n}. \tag{12}
$$

Once the constraints on the binary variables $a_{i,j}^{\delta}$ are defined, the objective is to minimize the discrepancy between the CDF of the GMM $F(x)$ and the empirical CDF $F_n(x)$ by calculating the corresponding probabilities in all of the intervals, $(\mu_j - \delta\sigma_j, \mu_j + \delta\sigma_j)$ $\delta \in \{1, 2, 3\}$, $j \in \{1, \ldots, K\}$.

Using Equations (9–12), we obtain a MIO formulation for the univariate case when number of Gaussians in the mixture K and mixture component weights π are known as follows:

$$
\min_{\{\mu_i, \sigma_i\}_{i=1}^{K}} \max_{\substack{\delta \in \{1,2,3\} \\ j \in \{1,2,\ldots,K\}}} \left| \frac{\sum_{i=1}^{n} a_{i,j}^{\delta}}{n} - \sum_{\ell=1}^{K} \pi_\ell \left(\Phi\left(\frac{\mu_j - \mu_\ell + \delta\sigma_j}{\sigma_\ell} \right) - \Phi\left(\frac{\mu_j - \mu_\ell - \delta\sigma_j}{\sigma_\ell} \right) \right) \right| \tag{13}
$$

$$
\begin{aligned}
\text{s.t.} \quad a_{i,j}^{\delta} x_i &\leq a_{i,j}^{\delta} \mu_j + \delta\sigma_j, \ i \in \{1, \ldots, n\}, \ j \in \{1, \ldots, K\}, \ \delta \in \{1, 2, 3\} \\
a_{i,j}^{\delta} x_i &\geq a_{i,j}^{\delta} \mu_j - \delta\sigma_j, \ i \in \{1, \ldots, n\}, \ j \in \{1, \ldots, K\}, \ \delta \in \{1, 2, 3\}
\end{aligned}
$$

$$
\begin{aligned}
(1-a^{\delta}_{i,j})\,|x_i-\mu_j| &\geq (1-a^{\delta}_{i,j})\delta\sigma_j,\ i\in\{1,\dots,n\},\ j\in\{1,\dots,K\},\ \delta\in\{1,2,3\}\\
a^{\delta}_{i,j} &\in \{0,1\},\ i\in\{1,\dots,n\},\ j\in\{1,\dots,K\},\ \delta\in\{1,2,3\}.
\end{aligned}
$$

Although, the "Big-M" formulations are weak, for tractability of the problem, we linearize constraints (10) using McCormick type linearization and introduce new variables $p^{\delta}_{i,j}=a^{\delta}_{i,j}\mu_j$ by incorporating the following constraints,

$$
\begin{aligned}
p^{\delta}_{i,j}-\delta\cdot\sigma_j &\leq a^{\delta}_{i,j}x_i &&\leq p^{\delta}_{i,j}+\delta\cdot\sigma_j,\\
\underline{M_\mu}a^{\delta}_{i,j} &\leq p^{\delta}_{i,j} &&\leq \bar{M}_\mu a^{\delta}_{i,j}\\
\mu_j-(1-a^{\delta}_{i,j})\bar{M}_\mu &\leq p^{\delta}_{i,j} &&\leq \mu_j-(1-a^{\delta}_{i,j})\underline{M_\mu}
\end{aligned}
$$

where the Big-M constants $[\underline{M_\mu},\bar{M}_\mu]$ are taken as $[x_1,x_n]$.

Similarly, constraint (11) can be linearized by reformulating the product $t^{\delta}_{i,j}=a^{\delta}_{i,j}\sigma_j$ as presented below

$$
\begin{aligned}
\delta\sigma_j &\leq (1-a^{\delta}_{i,j})x_i-\mu_j+p^{\delta}_{i,j}+\delta t^{\delta}_{i,j}+M_\mu(1-b^{\delta}_{i,j})\\
-\delta\sigma_j &\geq (1-a^{\delta}_{i,j})x_i-\mu_j+p^{\delta}_{i,j}-\delta t^{\delta}_{i,j}-M_\mu b^{\delta}_{i,j}\\
t^{\delta}_{i,j} &\leq M_\sigma a^{\delta}_{i,j}\\
t^{\delta}_{i,j} &\leq \sigma_j\\
t^{\delta}_{i,j} &\geq \sigma_j-M_\sigma(1-a^{\delta}_{i,j})\\
b^{\delta}_{i,j} &\in \{0,1\}
\end{aligned}
$$

where the Big-M constant M_σ for the standard deviation is taken as $M_\sigma=\sqrt{\frac{\hat{\sigma}_{\text{mix}}}{\pi_{\min}}}$, $\hat{\sigma}_{\text{mix}}$ is the empirical estimate of the standard deviation of the mixture.

To make the problem tractable, we reformulate Problem (13) by using a piecewise linear approximation to the standard normal CDF. Observe that the expression $\dfrac{\mu_j-\mu_\ell+\delta\sigma_j}{\sigma_\ell}$ cannot be linearized by defining auxillary variables as in Problem (2) as now we cannot retrieve the means and the variances from the auxillary variables. Therefore, to eliminate the nonlinearity imposed by σ_ℓ in the denominator of the expression $\dfrac{\mu_j-\mu_\ell+\delta\sigma_j}{\sigma_\ell}$, we approximate the product between binary variables $z^{k}_{\ell,j}$ and σ_ℓ with $r^{k}_{\ell,j}$ variables as follows:

$$
\begin{aligned}
r^{k}_{\ell,j} &\leq M_\sigma z^{k}_{\ell,j},\ k\in\{1,\dots,p-1\},\ \ell,\ j\in\{1,\dots,K\} \qquad (14)\\
r^{k}_{\ell,j} &\leq \sigma_\ell,\ k\in\{1,\dots,p-1\},\ \ell,\ j\in\{1,\dots,K\}\\
r^{k}_{\ell,j} &\geq \sigma_\ell-M_\sigma(1-z^{k}_{\ell,j}),\ k\in\{1,\dots,p-1\},\ \ell,\ j\in\{1,\dots,K\}\\
r^{k}_{\ell,j} &\geq 0,\ k\in\{1,\dots,p-1\},\ \ell,\ j\in\{1,\dots,K\}.
\end{aligned}
$$

Therefore, using Equations (3) and (14), a piecewise linear approximation to $\Phi\left(\frac{\mu_j - \mu_\ell + \delta\sigma_j}{\sigma_\ell}\right)$ is given by,

$$
\begin{aligned}
L\left(\frac{\mu_j - \mu_\ell + \delta\sigma_j}{\sigma_\ell}\right) &= \sum_{k=1}^{p} \Phi(v_k) y_{\ell,j}^k && (15)\\
\mu_j - \mu_\ell + \delta\sigma_j &= \sum_{k=1}^{p} v_k y_{\ell,j}^k,\ \ell,\ j \in \{1,\ldots,K\}\\
y_{\ell,j}^1 &\le r_{\ell,j}^1, \ell,\ j \in \{1,\ldots,K\}\\
y_{\ell,j}^k &\le r_{\ell,j}^{k-1} + r_{\ell,j}^k,\ k \in \{1,\ldots,p-1\},\ \ell,\ j \in \{1,\ldots,K\}\\
y_{\ell,j}^p &\le r_{\ell,j}^{p-1},\ \ell,\ j \in \{1,\ldots,K\}\\
\sum_{k=1}^{p-1} z_{\ell,j}^k &= 1,\ \ell,\ j \in \{1,\ldots,K\}\\
\sum_{k=1}^{p} y_{\ell,j}^k &= \sigma_\ell,\ \ell,\ j \in \{1,\ldots,K\}\\
r_{\ell,j}^k &\le M z_{\ell,j}^k,\ k \in \{1,\ldots,p-1\},\ \ell,\ j \in \{1,\ldots,K\}\\
r_{\ell,j}^k &\le \sigma_\ell,\ k \in \{1,\ldots,p-1\},\ \ell,\ j \in \{1,\ldots,K\}\\
r_{\ell,j}^k &\ge \sigma_\ell - M(1 - z_{\ell,j}^k),\ k \in \{1,\ldots,p-1\},\ \ell,\ j \in \{1,\ldots,K\}\\
z_{\ell,j}^k &\in \{0,1\},\ k \in \{1,\ldots,p-1\},\ \ell,\ j \in \{1,\ldots,K\}\\
r_{\ell,j}^k, y_{\ell,j}^k &\ge 0,\ k \in \{1,\ldots,p-1\},\ \ell,\ j \in \{1,\ldots,K\}
\end{aligned}
$$

From Eqs. (13) and (15) we obtain the following MIO formulation for Problem (8):

$$
\min \max_{\substack{\delta\in\{1,2,3\}\\ j\in\{1,2,\ldots,K\}}} \left| \frac{\sum_{i=1}^n a_{i,j}^\delta}{n} - \sum_{\ell=1}^{K} \pi_\ell \left(\sum_{k=1}^{p} \Phi(v_k) y_{\ell,j}^k \right) \right| \qquad (16)
$$

$$
\begin{aligned}
\text{s.t.} \quad p_{i,j}^\delta - \delta\cdot\sigma_j &\le a_{i,j}^\delta x_i \le p_{i,j}^\delta + \delta\cdot\sigma_j,\ i \in \{1,\ldots,n\},\ j \in \{1,\ldots,K\},\ \delta \in \{1,2,3\}\\
\underline{M_\mu} a_{i,j}^\delta &\le p_{i,j}^\delta \le \bar{M_\mu} a_{i,j}^\delta\ i \in \{1,\ldots,n\},\ j \in \{1,\ldots,K\},\ \delta \in \{1,2,3\}\\
\mu_j - (1 - a_{i,j}^\delta)\bar{M_\mu} &\le p_{i,j}^\delta \le \mu_j - (1 - a_{i,j}^\delta)\underline{M_\mu}\ i \in \{1,\ldots,n\},\ j \in \{1,\ldots,K\},\ \delta \in \{1,2,3\}\\
\delta\sigma_j &\le (1 - a_{i,j}^\delta)x_i - \mu_j + p_{i,j}^\delta + \delta t_{i,j}^\delta + M_\mu(1 - b_{i,j}^\delta)\\
-\delta\sigma_j &\ge (1 - a_{i,j}^\delta)x_i - \mu_j + p_{i,j}^\delta - \delta t_{i,j}^\delta - M_\mu b_{i,j}^\delta\\
t_{i,j}^\delta &\le M_\sigma a_{i,j}^\delta\ i \in \{1,\ldots,n\},\ j \in \{1,\ldots,K\},\ \delta \in \{1,2,3\}\\
t_{i,j}^\delta &\le \sigma_j\ i \in \{1,\ldots,n\},\ j \in \{1,\ldots,K\},\ \delta \in \{1,2,3\}\\
t_{i,j}^\delta &\ge \sigma_j - M_\sigma(1 - a_{i,j}^\delta)\ i \in \{1,\ldots,n\},\ j \in \{1,\ldots,K\},\ \delta \in \{1,2,3\}\\
\mu_j - \mu_\ell + \delta\sigma_j &= \sum_{k=1}^{p} v_k y_{\ell,j}^k,\ \ell,\ j \in \{1,\ldots,K\}\\
y_{\ell,j}^1 &\le r_{\ell,j}^1,\ \ell,\ j \in \{1,\ldots,K\}
\end{aligned}
$$

$$
\begin{aligned}
y^k_{\ell,j} &\leq r^{k-1}_{\ell,j} + r^k_{\ell,j},\ k \in \{1,\ldots,p-1\},\ \ell,\ j \in \{1,\ldots,K\} \\
y^p_{\ell,j} &\leq r^{p-1}_{\ell,j},\ \ell,\ j \in \{1,\ldots,K\} \\
\sum_{k=1}^{p-1} z^k_{\ell,j} &= 1,\ \ell,\ j \in \{1,\ldots,K\} \\
\sum_{k=1}^{p} y^k_{\ell,j} &= \sigma_\ell,\ \ell,\ j \in \{1,\ldots,K\} \\
r^k_{\ell,j} &\leq M z^k_{\ell,j},\ k \in \{1,\ldots,p-1\},\ \ell,\ j \in \{1,\ldots,K\} \\
r^k_{\ell,j} &\leq \sigma_\ell,\ k \in \{1,\ldots,p-1\},\ \ell,\ j \in \{1,\ldots,K\} \\
r^k_{\ell,j} &\geq \sigma_\ell - M(1 - z^k_{\ell,j}),\ k \in \{1,\ldots,p-1\},\ \ell,\ j \in \{1,\ldots,K\} \\
a^\delta_{i,j} &\in \{0,1\},\ i \in \{1,\ldots,n\},\ j \in \{1,\ldots,K\},\ \delta \in \{1,2,3\} \\
z^k_{\ell,j} &\in \{0,1\},\ k \in \{1,\ldots,p-1\},\ \ell,\ j \in \{1,\ldots,K\} \\
b^\delta_{i,j} &\in \{0,1\},\ i \in \{1,\ldots,n\},\ j \in \{1,\ldots,K\},\ \delta \in \{1,2,3\} \\
r^k_{\ell,j}, y^k_{\ell,j} &\geq 0,\ k \in \{1,\ldots,p-1\},\ \ell,\ j \in \{1,\ldots,K\}.
\end{aligned}
$$

Observe that the above MIO problem has a total of $K(6n + K(p-1))$ binary variables.

As an alternate approach for minimizing the total variation distance between the empirical distribution function and the distribution function of the GMM, we consider the problem of minimizing the total variation distance over the set of all intervals $\mathcal{J} = \{(x_i, x_j)|\ i,j \in \{1,2,\ldots,n\}\}$. Therefore, we propose to solve the following problem,

$$
\min_{\{\mu_i,\sigma_i\}_{i=1}^K} \max_{(i,j)\in\mathcal{N}\times\mathcal{N}} \left| \frac{j-i}{n} - \sum_{\ell=1}^{K} \pi_\ell \left\{ \Phi\left(\frac{x_j - \mu_\ell}{\sigma_\ell}\right) - \Phi\left(\frac{x_i - \mu_\ell}{\sigma_\ell}\right) \right\} \right|, \tag{17}
$$

where, $\mathcal{N} = \{1,2,\ldots,n\}$. In order to speed up the solver we use a similar approach of generating dynamic constraints as in Section 2.1.3. We maintain a dynamic set of ordered indices $\mathcal{I}$ so that we solve the problem:

$$
\min_{\{\mu_i,\sigma_i\}_{i=1}^K} \max_{(i,j)\in\mathcal{I}} \left| \frac{j-i}{n} - \sum_{\ell=1}^{K} \pi_\ell \left\{ \Phi\left(\frac{x_j - \mu_\ell}{\sigma_\ell}\right) - \Phi\left(\frac{x_i - \mu_\ell}{\sigma_\ell}\right) \right\} \right|.
$$

Whenever the solver finds an integer feasible solution $\{\epsilon, (\mu_\ell, \sigma_\ell),\ \ell = 1,2,\ldots,K\}$, we find an interval (x_i, x_j) that has the maximum absolute difference in probability between the empirical distribution function and the distribution function of the GMM inside this interval. Finally, we update the set of indices $\mathcal{I} = \mathcal{I} \cup \{(i,j)\}$ and keep solving the problem by adding lazy constraints to the model as shown in Section 2.1.3 until

$$
\max_{(i,j)\in\mathcal{N}\times\mathcal{N}\setminus\mathcal{I}} \left| \frac{j-i}{n} - \sum_{\ell=1}^{K} \pi_\ell \left\{ \Phi\left(\frac{x_j - \mu_\ell}{\sigma_\ell}\right) - \Phi\left(\frac{x_i - \mu_\ell}{\sigma_\ell}\right) \right\} \right| \leq \epsilon.
$$

This makes sure that we solve the problem (17) to optimality.

2.3 Estimating mixture component weights

In this section, we consider the case when mixture component weights π are unknown but the number of Gaussians in the mixture, K is still known. In this case, we use an Alternating Optimization (AO) technique motivated by AO methods for convex optimization (Bezdek and Hathaway, 2002) that alternate between optimizing over a collection of non-overlapping subsets of variables and are shown to converge. In our AO approach we alternate between optimizing over the parameters of the GMM given a set of mixture component weights and optimizing over the mixture component weights given an estimate of the parameters of GMM. With this approach, the objective improves monotonically with each iteration of the AO algorithm. However, since the objective function in Problem (2) is non-convex, we cannot prove convergence of the AO method.

Algorithm 1 below allows us to jointly estimate the component mixture weights π along with the Gaussians' parameters of the mixture. Note that we run Algorithm 1 from multiple starting points π_j^0, $j = 1, \ldots, K$ and keep the solution with highest log-likelihood.

2.4 Choosing the number of Gaussian components

In this section, we consider the case when we do not know the number of Gaussian components or the mixture component weights, we are only given i.i.d. data from a mixture of Gaussians.

To address the problem of choosing the right number of Gaussian components in the mixture we use cross-validation, a classical model selection technique in machine learning. First, we split the dataset into training, validation and testing datasets. Then, to choose the right value of K, we do the following: starting with the number of Gaussians in the mixture $K = 2$, we learn the parameters of GMM on the training dataset using either MIO Problem (4) or (16) depending on whether we use the Kolmogorov-Smirnov or the Total Variation distance while assuming that the mixture consists of K Gaussian components and compute the log-likelihood on the validation dataset.

Finally, we plot the log-likelihood calculated on the test set against K and choose the value of K for which the likelihood is the highest.

Algorithm 1 Joint estimation of mixture weights and Gaussian parameters

Input: Data $\{x_i|\ x_i \in \mathbb{R}, i = 1, 2, \ldots, n\}$, number of Gaussians components K, initial weights π_j^0, $j = 1, \ldots, K$ and stopping criterion ϵ.
Ouput: $\theta = \{(\pi_1, \mu_1, \sigma_1), (\pi_2, \mu_2, \sigma_2), \ldots, (\pi_K, \mu_K, \sigma_K)\}$.
Algorithm:

1. Let $t := 0$. Using $(\pi_1^t, \ldots, \pi_K^t)$ as estimates for the weights, solve for the parameters $\{\mu_i^t, \sigma_i^t\}_{i=1}^K$ of the GMM using either Problem (4) or (16) depending on whether we use the Kolmogorov-Smirnov or the Total Variation distance. Let $\theta^t = \left\{(\pi_1^t, \mu_1^t, \sigma_1^t), (\pi_2^t, \mu_2^t, \sigma_2^t), \ldots, (\pi_K^t, \mu_K^t, \sigma_K^t)\right\}$.

2. Solve the following linear optimization problem over weights π using the estimates of the parameters of GMM $\left(\mu_i^t, \sigma_i^t\right)$ obtained from previous step.

$$\min_{\{\pi_i\}_{i=1}^K} \max_{i \in \{1,2,\ldots,n\}} \left| \frac{i}{n} - \sum_{j=1}^{K} \pi_j L\left(\frac{x_i - \mu_j^t}{\sigma_j^t}\right) \right| \tag{18}$$

$$\begin{aligned} \text{s.t.} \quad \sum_{i=1}^{K} \pi_i &= 1, \\ \pi_i &\geq 0. \end{aligned}$$

   where $L(\cdot)$ is the piecewise linear approximation for the standard normal CDF used in formulations (4, 16).

3. Let π_j^{t+1}, $j = 1, \ldots, K$ be an optimal solution to problem (18). Using π_j^{t+1}, $j = 1, \ldots, K$ as estimates for the weights, solve for the parameters $\{\mu_i^{t+1}, \sigma_i^{t+1}\}_{i=1}^K$ of the GMM using either Problem (4) or (16). Let $\theta^{t+1} = \left\{(\pi_1^{t+1}, \mu_1^{t+1}, \sigma_1^{t+1}), (\pi_2^{t+1}, \mu_2^{t+1}, \sigma_2^{t+1}), \ldots, (\pi_K^{t+1}, \mu_K^{t+1}, \sigma_K^{t+1})\right\}$.

4. If

$$\frac{|\mathcal{L}(\theta^{t+1}) - \mathcal{L}(\theta^t|}{|\mathcal{L}(\theta^t|} \leq \epsilon$$

   then stop and output θ^{t+1} where $\mathcal{L}(\cdot)$ is the log-likelihood,

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{K} \frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} \exp^{-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}} \right).$$

5. Else, $t := t + 1$ and go to Step 2.

3 Multivariate Gaussian Mixture Modeling using MIO

In this section, we propose a multivariate learning algorithm to estimate the parameters of a mixture of Gaussians given d-dimensional data, as an extension of the univariate learning algorithm proposed in Section 2. Given data $\{\mathbf{x}_i|\ \mathbf{x}_i \in \mathbb{R}^d, i = 1, 2, , \ldots, , n\}$, we propose a multivariate algorithm that learns the parameters of the GMM, $\theta = \{(\pi_1, \mu_1, \Sigma_1), (\pi_2, \mu_2, \Sigma_2), \ldots, (\pi_K, \mu_K, \Sigma_K)\}$ without making any additional assumptions on the model.

It is well known that learning the parameters of GMM is computationally hard in higher dimensions. Observe that given a Gaussian random variable $X \sim \mathcal{N}(\mu, \Sigma)$, and some direction $\rho \in \mathbb{R}^d$, the random variable X projected onto ρ is also a normally distributed random variable with $\rho' X \sim \mathcal{N}(\rho'\mu, \rho'\Sigma\rho)$. Therefore using the fact that the projection of a multivariate GMM onto a line is a univariate GMM, we project the data down onto multiple directions in 1-d space and learn the parameters of GMM in those particular projected directions. We iteratively project the data onto various random directions so as to learn all the parameters of the d-dimensional GMM. The approaches proposed in Vempala and Wang (2002a), Sanjeev and Kannan (2001), Dasgupta (1999) are based on projecting data to a randomly chosen low-dimensional subspace and then finding an accurate clustering in the lower dimensions where the separation between the Gaussian components is at least a factor of $\max_{i \in \{1,2,\ldots,K\}} \sigma_i$. In constrast, our univariate algorithm is less sensitive to separation between Gaussian components, therefore in our case, we can project the data into any random direction.

In the multivariate algorithm proposed here, we first project the data onto a series of d^2 random directions $\mathrm{D} = \{\rho_i|\ \rho_i \in \mathbb{R}^d, i = 1, 2, \ldots, d^2\}$ in order to estimate all of the Kd mean and $K\frac{d(d+1)}{2}$ covariance parameters along with K component mixture weights. Note that when two Gaussian components in the mixture have the same weights, the univariate algorithm outputs some permutation of the parameter estimates. Therefore, this induces a permutation learning problem to correctly identify a consistent ordering among the estimates of the means and variances across the d-dimensions. However, if we knew exactly the permutations of the mixture component means for all of the projected directions, we can use an orthonormal basis $W = [b_1\ b_2 \ldots b_d]$ as a set of projection directions and estimate the means of the Gaussian components by inverting W, which is directly given by $W^{-1} = W^T$. Similarly, to estimate the covariance matrices, we can choose a set of orthonormal basis matrices that span all of the symmetric matrices and estimate all of the covariance matrices. However, since permutations of the estimates are usually unknown; in order to recover the true ordering tractably, we formulate a MIQO problem to identify a consistent ordering among the means and then using the recovered ordering, we formulate an SDO problem to estimate the covariance matrices.

3.1 The multivariate algorithm

Here we present an algorithm for modeling multidimensional data as a mixture of Gaussians. As explained in the previous section, we project the data via a series of projections and solve either Problem (4) or (16) depending on whether we use the Kolmogorov-Smirnov or the Total Variation distance iteratively to learn parameters of the GMM in the projected space. We finally formuate a MIQO problem to identify a consistent ordering across the parameter estimates in different coordinates and using this consistent ordering, we formulate an SDO problem to estimate the covariance matrices.

To find a consistent ordering of the means and the variances across d-dimensions, we project the data onto a series of d^2 random directions $\mathcal{D} = \{\rho_i |\ \rho_i \in \mathbb{R}^d, i = 1, 2, \ldots, d^2\}$ and run Algorithm 1 to find estimates of the means and variances of K components in the projected space of ρ_k as: $\big\{(m_1^k, s_1^k), (m_2^k, s_2^k), \ldots, (m_K^k, s_K^k)\big\}$. Note that since for each random direction $\{\rho_1, \rho_2, \ldots, \rho_{d^2}\}$, we run the algorithm independently, the estimates of the means and variances recovered in the projected space are some permutation of the true ordering. In order to recover a consistent ordering among the estimates across d-dimensions, we formulate a MIQO problem.

Let us denote $\big\{\mu_i |\ \mu_i \in \mathbb{R}^d, i = 1, 2 \ldots, K\big\}$ as the true values of the means of the Gaussian components in the mixture. We now define a projection matrix P^k for each $\rho_k \in \mathcal{D}$ as follows:

$$P_{ij}^k = \begin{cases} 1, & \text{if } m_j^k \text{ is an estimate of } \rho_k' \mu_i, \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, we need to find permutation matrices $P^k, k \in \{1, 2 \ldots, d^2\}$ such that

$$P^k m^k \approx \begin{pmatrix} \rho_k' \mu_1 \\ \rho_k' \mu_2 \\ \ldots \\ \rho_k' \mu_K \end{pmatrix}, k \in \{1, 2, \ldots, d^2\},$$

where $m^k = \big(m_1^k, m_2^k, \ldots, m_K^k\big)$.

Since the estimates of the means recovered in the projected space are noisy estimates of the true means, we minimize ℓ_2^2 error of the estimates with the true values of the means in the projected space. We thus propose to solve the following MIQO problem:

$$\min_{\{\pi^{p_l}\}_{k=1}^{d^2}, \{\mu_i\}_{i=1}^K} \sum_{k=1}^{d^2} \left\| P^k m^k - \begin{pmatrix} \rho_k' \mu_1 \\ \rho_k' \mu_2 \\ \ldots \\ \rho_k' \mu_K \end{pmatrix} \right\|_2^2 \tag{19}$$

$$
\begin{aligned}
\text{s.t.} \quad \sum_{i=1}^{K} P_{ij}^k &= 1,\ j \in \{1,2,\ldots,d\},\ k \in \{1,2\ldots,d^2\} \\
\sum_{j=1}^{K} P_{ij}^k &= 1,\ i \in \{1,2,\ldots,d\},\ k \in \{1,2,\ldots,d^2\} \\
P_{ij}^k &\in \{0,1\},\ i \in \{1,2,\ldots,K\},\ j \in \{1,2,\ldots,K\},\ k \in \{1,2,\ldots,d^2\}.
\end{aligned}
$$

Problem (19) has K^2d^2 binary variables. As (K,d) are usually not very large in practice, the MIQO problem is solved to optimality in a few minutes.

Using the solution $\{P^k, k = 1,2,\ldots,d^2\}$ of Problem (19), we formulate an SDO problem to recover the estimates of the covariance matrices as follows:

$$
\begin{aligned}
\min_{\{\Sigma_i\}_{i=1}^K} \sum_{k=1}^{d^2} \left\| P^k s^k - \begin{pmatrix} \rho_k' \Sigma_1 \rho_k \\ \rho_k' \Sigma_2 \rho_k \\ \cdots \\ \rho_k' \Sigma_K \rho_k \end{pmatrix} \right\|_1 \\
\text{s.t.} \quad \Sigma_i \succeq 0, \quad i \in \{1,2,\ldots,K\}.
\end{aligned} \tag{20}
$$

The above SDO problem has K semidefinite matrices ($\Sigma_i \in \mathcal{S}_d\ i = 1,2,\ldots,K$). When K and d are small, the problem is solved to optimality within a few minutes. Algorithm 2, below learns the prameters of a multivariate GMM.

3.2 Choosing the number of Gaussian components

Similar to the univariate case, we first split the dataset into training, validation and testing datasets. We then learn the parameters of GMM using Algorithm 2 and perform cross-validation to choose the number of Gaussian components K that gives the highest log-likelihood on the validation dataset.

Algorithm 2 Algorithm for learning parameters of a multivariate GMM

Input: Data $\{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^d,\ i = 1, 2, \ldots, n\}$, number of Gaussians components K, stopping criterion ϵ and a set of d^2 random directions $\mathcal{D} = \{\rho_i |\ \rho_i \in \mathbb{R}^d, i = 1, 2, \ldots, d^2\}$.
Ouput: $\theta = \{(\pi_1, \mu_1, \Sigma_1), (\pi_2, \mu_2, \Sigma_2), \ldots, (\pi_K, \mu_K, \Sigma_K)\}$.
Algorithm:

1. For each $k \in \{1, 2, \ldots, d^2\}$:
   - Project the data down onto the line ρ_k: $X_k = \{\rho_k' \mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^d, i = 1, 2, \ldots, n\}$.
   - Apply Algorithm 1 to (X_k, ϵ) to recover estimates of the component weights, means and variances. Denote the estimates as $\left\{(\pi_1^k, m_1^k, s_1^k), (\pi_2^k, m_2^k, s_2^k), \ldots, (\pi_K^k, m_K^k, s_K^k)\right\}$.
2. Set $\pi_i = \dfrac{\sum_{k=1}^{d^2} \pi_i^k}{d^2}, i = 1, 2, \ldots, K$.
3. Using $\left\{\left(m_1^k, m_2^k, \ldots, m_K^k\right) | k = 1, 2, \ldots, d^2\right\}$ as problem data, solve the MIQO problem (19) to identify a consistent ordering of the means and the variances across d-dimensions for the estimates of the means and variances in all projected spaces.
4. Using the consistent ordering (permutation matrices) recovered above, solve the SDO problem (20) to estimate the covariance matrices.
5. Output $\theta = \{(\pi_1, \mu_1, \Sigma_1), (\pi_2, \mu_2, \Sigma_2), \ldots, (\pi_K, \mu_K, \Sigma_K)\}$.

4 Data and Computational Results

In this section, we describe the data used and report the performance of our models on both synthetic and real-world datasets. We study the performance of Algorithms 1, 2 and compare them to the EM algorithm, and the models with (Tocher, 1967, Zelen and Severo, 1964) approximations to the standard normal CDF. Specifically, we study the dependence of the accuracy in estimating means, variances and mixture component weights on the training sample size. We also study how close the recovered distribution function of the GMM is to the empirical distribution function quantified by the Kolmogorov-Smirnov and the Total Variation distances. We use mean absolute percentage error (MAPE) and weighted-MAPE to quantify the errors in estimating means, variances and the mixture component weights. Specifically, we use the following metrics to compare the performance of Algorithms 1 and 2 with the EM algorithm:

- The Kolmogorov-Smirnov distance between the GMM distribution function F and the empirical distribution function F_n is given by

$$D_{KS}(F_n, F) = \max_{x \in \{x_1, x_2, \ldots, x_n\}} |F_n(x) - F(x)|.$$

- The Total Variation distance between the GMM distribution function F and the empirical

distribution function F_n is given by

$$D_{TV}\left(F_n, F\right) = \max_{i<j} |\{F_n\left(x_j\right) - F_n\left(x_i\right)\} - \{F\left(x_j\right) - F\left(x_i\right)\}|.$$

- The MAPE in estimating means is given by

$$T_\mu = \frac{1}{k}\sum_{i=1}^{k} \frac{\|\Delta\mu_i\|_2}{\|\mu_i\|_2},\ \Delta\mu_i = \mu_i - \mu_i^{true}.$$

- The MAPE in estimating variances is given by

$$T_\sigma = \frac{1}{k}\sum_{i=1}^{k} \frac{\|\Delta\Sigma_i\|_{F_2}}{\|\Sigma\|_{F_2}},\ \Delta\Sigma_i = \Sigma_i - \Sigma_i^{true},$$

  where the Frobenius q^{th} norm of a matrix $A \in \mathbb{R}^{m\times n}$ is defined as $\|A\|_{F_q} = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |A_{ij}|^q\right)^{\frac{1}{q}}$.

- The MAPE in estimating mixture component weights is given by

$$T_\pi = \frac{1}{k}\sum_{i=1}^{k} \frac{|\Delta\pi_i|}{\pi_i},\ \Delta\pi_i = \pi_i - \pi_i^{true}.$$

All of the experiments were performed on a computer with Xeon @2.3GHz processors, 4 cores,16GB RAM and all of the code implemented in Julia language (v 0.6) using commercial solver Gurobi 6.5.2.

4.1 Computational results with synthetic datasets

We generated a number of synthetic datasets from one-dimensional Gaussian mixture consisting of two Gaussian components ($K = 2$) with

1. Larger separation between the Gaussian components: $\dfrac{|\mu_1 - \mu_2|}{\sigma_{\max}} = 2$, where $\sigma_{\max} = \max_{i\in\{1,2,\ldots,K\}} \sigma_i$.
2. Smaller separation between the Gaussian components: $\dfrac{|\mu_1 - \mu_2|}{\sigma_{\max}} = 1$.
3. Varying separation between the Gaussian components: $\dfrac{|\mu_1 - \mu_2|}{\sigma_{\max}} \in [0, 6]$.

For each of the above datasets, we generated multiple samples with n ranging from 100 to 2000 to study the dependence of the performance of our models on n. In Figures 2, 3 we compare the performance of our MIO problems (4,16) with the EM algorithm for the case $\dfrac{|\mu_1 - \mu_2|}{\sigma_{\max}} = 2$ or 1, respectively as a function of n. In Figure 4, we compare the performance of MIO problems (4,16) with the EM algorithm for training sample size, n=500 as a function of the separation between the Gaussians $\frac{|\mu_1-\mu_2|}{\sigma_{max}}$ varying from 0 to 6.

Observations

1. In all cases we observe a significant improvement in all performance measures of the MIO based methods compared to the EM algorithm independent of the sample size n. Specifically the MIO based methods achieve an average improvement of 60-70% and 50-60% over the EM algorithm for MAPE in estimating the means and the covariance matrices, respectively.

2. For large separations (around 6), the MIO based methods had comparable performance compared to the EM methods. As the separation decreased, the edge in performance in favor of the MIO methods widened.

3. The performance of the MIO based methods based on either the Kolmogorov-Smirnov or the Total Variation distance is very similar.

In Table 1, we present the runtimes of the EM algorithm and algorithm 1 for both cases when it solves either Problem (4) or (16) depending on whether we use the Kolmogorov-Smirnov or the Total Variation distance. We also report the number of iterations performed until the stopping criteria in Algorithm 1 is met. The table on the left shows mean runtime for synthetic data of various sizes with a separation of $\frac{|\mu_1 - \mu_2|}{\sigma_{\max}} = 2$ and the table on the right shows mean runtime for datasets of size $n = 500$ with separation varying from 0 to 6.

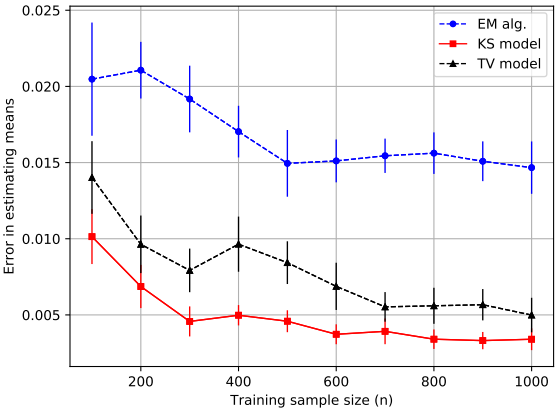
	Mean runtime (sec.)			Iterations		
n	EM	KS	TV	EM	KS	TV
100	16.8	109	222	1,000	3	3
200	17.3	186	350	1,000	3	3
300	18.2	231	528	1,000	3	4
400	17.8	240	685	1,000	4	4
500	18.3	318	915	1,000	4	4
600	17.9	364	1181	1,000	4	5
700	17.9	398	1342	1,000	5	6
800	18.1	434	1613	1,000	6	7
900	18.2	527	1856	1,000	7	7
1000	18.6	595	2039	1,000	7	8

	Mean runtime (sec.)			Iterations		
$\frac{\lvert\mu_1-\mu_2\rvert}{\sigma_{\max}}$	EM	KS	TV	EM	KS	TV
0	17.6	1752	2053	1,000	12	14
0.25	17.6	1141	2004	1,000	11	10
0.5	17.8	865	1830	1,000	12	11
1	18	823	1660	1,000	9	11
1.5	18.2	475	1284	1,000	10	12
2	18.2	383	1118	1,000	6	8
3	18.1	470	1204	1,000	8	10
4	18.3	301	924	1,000	7	7
5	17.5	227	859	1,000	7	6
6	17.4	83	761	1,000	5	5

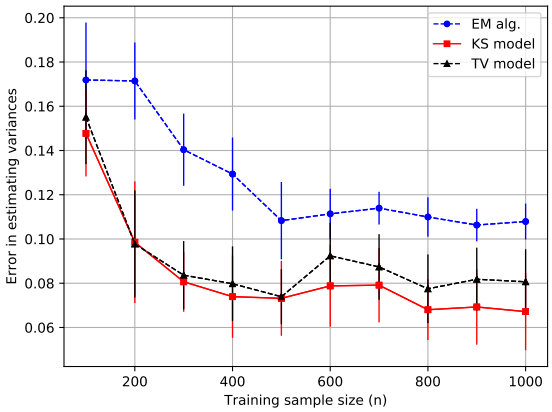
Table 1: Comparison of runtimes of algorithm 1 and the EM algorithm on synthetic datasets versus size of data and the separation between gaussian component (table on the left shows mean runtime for datasets with separation between the Gaussian components $\frac{|\mu_1 - \mu_2|}{\sigma_{\max}} = 2$ and the table on the right shows mean runtime for datasets of size $n = 500$ with varying separation) along with the number of iterations for $\epsilon = 0.01$.

4.2 Computational results with real-world datasets

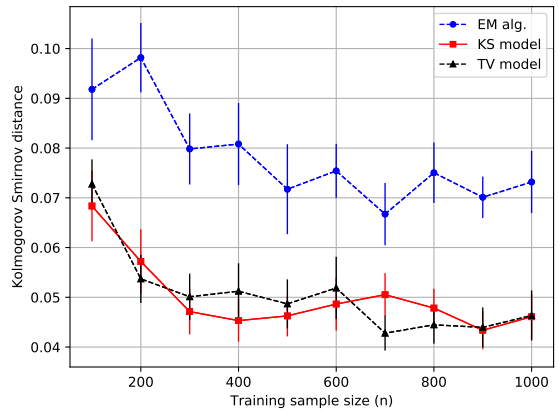
In the second part of the experiments we applied Algorithm 2, the EM algorithm and state-of-the art methods for classification, namely, Support Vector Machines (SVM), Classification and Regression



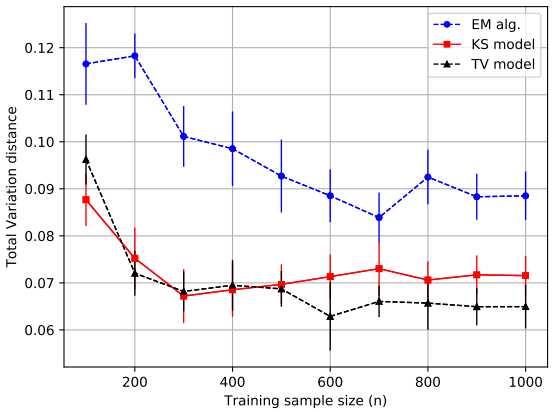
(a) Error in estimating means.



(b) Error in estimating variances.



(c) Kolmogorov-Smirnov distance.



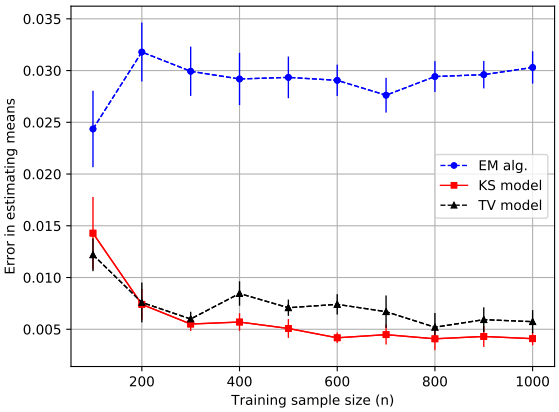
(d) Total Variation distance.

Figure 2: Performance as a function of n for a one-dimensional Gaussian mixture with K=2 components and separation $\frac{|\mu_1 - \mu_2|}{\sigma_{\max}} = 2$.

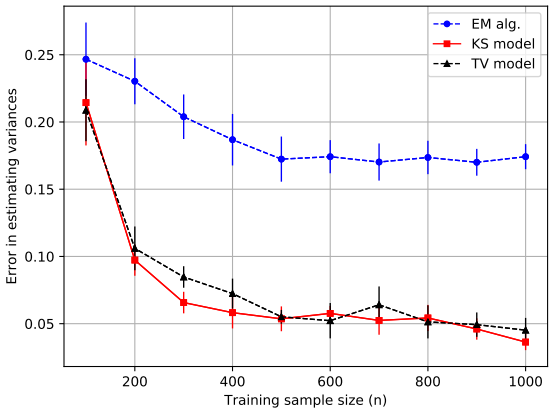
Trees (CART) and Random forests (RF) on various publicly available data sets from the UCI repository (Asuncion and Newman, 2007). Specifically, we chose Breast Cancer, Diabetes, Image segmentation, Iris and US income census data sets to compare the performance of our algorithm in terms of out-of-sample accuracy. For each of these data sets, we randomly split the data into two parts: training set (70%) and test set (30%). We then perform random splits on the data sets five times and report the mean out-of-sample accuracy.

We first estimate the parameters of the Gaussian mixture model by applying Algorithm 2 that solves both Problem (4) of minimizing the Kolmogorov-Smirnov distance and Problem (16) of minimizing the Total Variation distance.

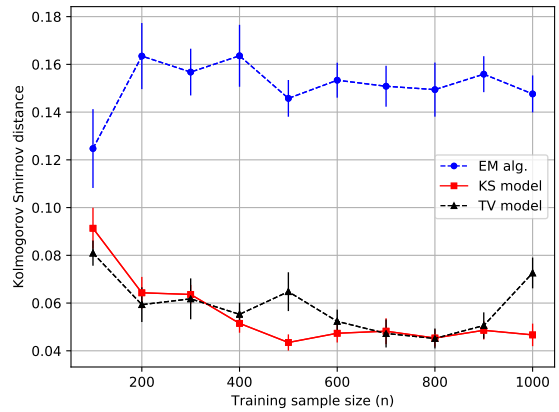
After solving for the parameters of the mixture of Gaussians, we estimate the posteriori component assignment probability using Bayes' theorem for each of the samples in the test set. Given



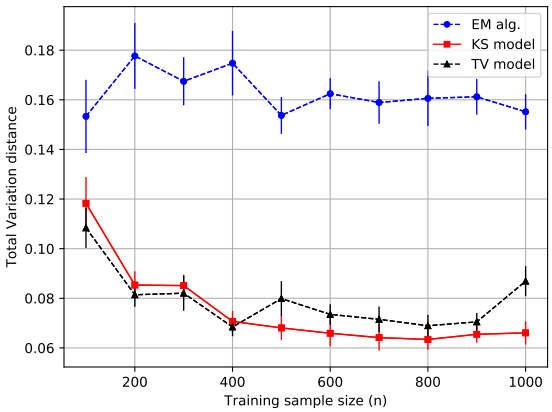
(a) Error in estimating means.



(b) Error in estimating variances.



(c) Kolmogorov-Smirnov distance.



(d) Total Variation distance.

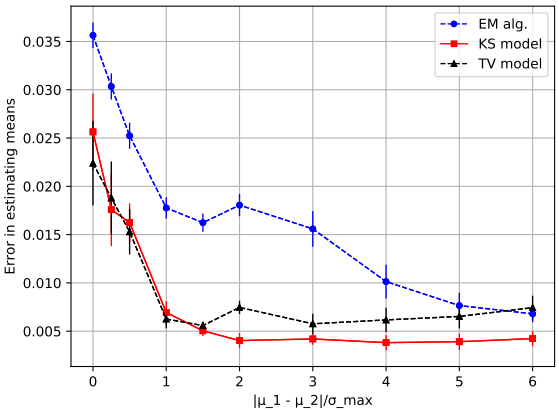
Figure 3: Performance as a function of n for a one-dimensional Gaussian mixture with K=2 components and separation $\dfrac{|\mu_1 - \mu_2|}{\sigma_{\max}} = 1$.

a data point x, the probability that it belongs to class $\mathcal{C}_i, i = 0, 1, 2, \ldots, K$ is given by

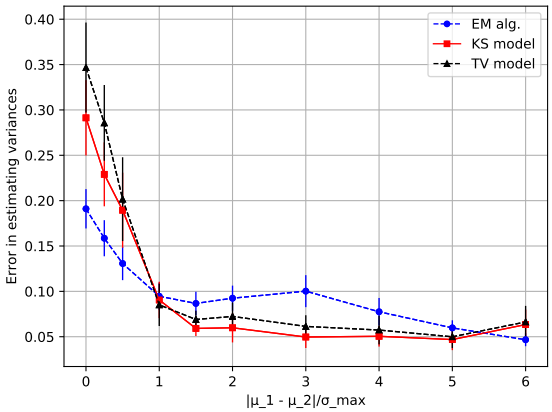
$$\mathbb{P}\left(\mathcal{C}_i|x\right) = \frac{\mathbb{P}\left(\mathcal{C}_i\right)\mathbb{P}\left(x|\mathcal{C}_i\right)}{\sum_{j=1}^{K}\mathbb{P}\left(\mathcal{C}_j\right)\mathbb{P}\left(x|\mathcal{C}_j\right)} = \frac{\pi_i\mathcal{N}\left(x|\mu_i,\sigma_i\right)}{\sum_{j=1}^{K}\pi_j\mathcal{N}\left(x|\mu_j,\sigma_j\right)}.$$

Finally, we classify the each sample x based on the most likely component assignment using posteriori component assignment probabilities. Note that for the Image segmentation dataset, we used Principal Component Analysis (PCA) to reduce the dimensionality of the data from 19 to 4 by choosing the first four principal components that explained more than 94% of the total variance in the data set.

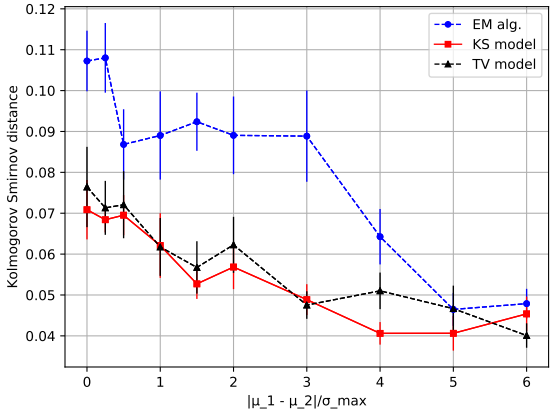
In Table 2, we compare the performance of our algorithm (KS & TV) with the EM algorithm, the models with (Tocher, 1967, Zelen and Severo, 1964) approximations to the standard normal



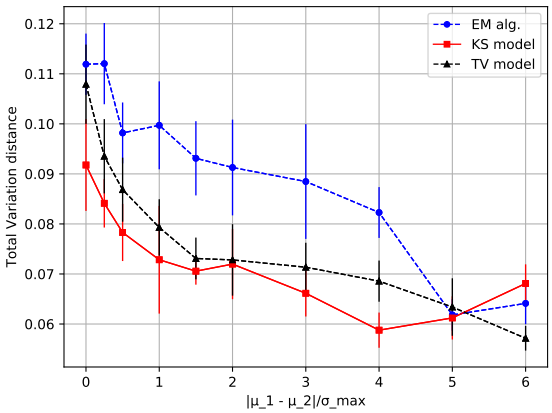
(a) Error in estimating means



(b) Error in estimating variances



(c) Kolmogorov-Smirnov distance



(d) Total Variation distance

Figure 4: Performance as a function of separation $\frac{|\mu_1-\mu_2|}{\sigma_{max}}$ between the Gaussian components for a one-dimensional Gaussian mixture with K=2 components and training sample size n=500.

CDF and the state-of-the art methods for classification in terms of out-of-sample accuracies for each of the datasets. In all of the tests on the real-world datasets, we observed that the MIO based methods outperform the EM algorithm with an average improvement of 4-5% on out-of-sample accuracy. Although we have compared the performance of our algorithms with the state-of-the-art methods in classification, we believe the comparison is not fair. Since different classification methods have different operating characteristics – for example, mixtures of discriminant analysis methods (Friedman et al., 2001) do flexible modeling of covariates (via mixture models), whereas SVM, CART and RF do not model the distribution of the covariates. A by-product of the mixture discriminant analysis framework is uncertainty quantification via probabilistic modeling (which is not natural in the context of SVMs). Hence, our primary motivation here is to empirically study the gains (in classification accuracy) by using our proposal for GMM estimation, when compared to EM-based procedures.

In Table 3, we report the number of iterations performed till the convergence criteria is met for Algorithm 2 using either KS or TV distance and the EM algorithm. We also report the training time for each of these methods to estimate a GMM with a cut-off time at 720 mins. Both the models with (Tocher, 1967, Zelen and Severo, 1964) approximations to the standard normal CDF solved using Baron commercial solver do not make the cut-off time for the Image segmentation and US census datasets due to their large sizes. Observe that even though the methods KS and TV have comparable performance to Tr and ZS in terms of out-of-sample accuracy, the training times for both the methods Tr and ZS are approximately 2-orders of magnitude higher. Also observe that we gain an average improvement of 4-5% in out-of-sample accuracy over the EM-algorithm by paying a price in training time as shown in Table 3.

Dataset				Out-of-sample accuracy							
Name	n	d	K	EM	KS	TV	Tr	ZS	SVM	CART	RF
Breast Cancer	683	9	2	76.7%	80.2%	80.7%	79.8%	80.4%	87.8%	92.3%	93.8%
Diabetes	768	8	2	58.8%	65.1%	64.6%	65.5%	65.7%	68.9%	70.6%	72.7%
Image Segmentation	2,310	4	7	32.9%	40.4%	39.9%	-	-	44.2%	52.5%	64.2%
Iris	150	4	3	88.2%	92.3%	91.9%	90.8%	91.5%	92.0%	92.4%	94.1%
US Census	45,222	6	2	85.4%	87.7%	87.1%	-	-	90.1%	92.6%	94.6%

Table 2: Comparative results of algorithm 2(KS & TV), the EM algorithm, model using Tocher(Tr) approximation, model using Zelen & Severo(ZS) approximation, support vector machine(SVM), classification and regression trees(CART) and random forest(RF) on data sets from UCI ML Repository in terms of out-of-sample accuracy.

Dataset				Iterations			Training time(min)				
Name	n	d	K	EM	KS	TV	EM	KS	TV	Tr	ZS
Breast Cancer	683	9	2	1,274	4	4	0.23	5.92	7.86	455.27	587.34
Diabetes	768	8	2	1,498	6	7	0.28	6.76	8.16	582.45	714.38
Image Segmentation	2,310	4	7	2,763	12	16	0.84	10.28	12.76	-	-
Iris	150	4	3	686	3	5	0.12	1.84	2.78	126.14	162.56
US Census	45,222	6	2	29,375	39	52	3.96	126.48	168.83	-	-

Table 3: Comparative results of algorithm 2(KS & TV), the EM algorithm, model using Tocher(Tr) approximation, model using Zelen & Severo(ZS) approximation on data sets from UCI ML Repository in terms of the number of iterations for convergence with stopping criterion $\epsilon = 0.01$ and the training time.

5 Conclusions

In this paper, we propose a new methodology to solve the problem of recovering estimates of a Gaussian mixture model (GMM) given data that is believed to come from multiple heterogeneous subpopulations. We minimize a discrepancy (either the Kolmogorov-Smirnov or the Total Variation distance) between the empirical distribution function and the distribution function of the GMM. We

presented two novel MIO models to solve the problem of minimizing a discrepancy to optimality. Using both synthetic and real datasets, we illustrated that our algorithms outperform the EM algorithm under various settings. The algorithms proposed in this paper can easily be extended to a variety of univariate distribution families thereby opening the door to MIO based algorithms for optimally learning the parameters of a mixture of various distribution families.

References

A. Asuncion and D. Newman. Uci machine learning repository, 2007.

S. Balakrishnan, M. J. Wainwright, B. Yu, et al. Statistical guarantees for the em algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120, 2017.

M. Belkin and K. Sinha. Learning gaussian mixtures with arbitrary separation. *arXiv preprint arXiv:0907.1054*, 2009.

M. Belkin and K. Sinha. Polynomial learning of distribution families. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 103–112. IEEE, 2010.

J. C. Bezdek and R. J. Hathaway. Some notes on alternating optimization. In *AFSS International Conference on Fuzzy Systems*, pages 288–300. Springer, 2002.

K. Chaudhuri. *Learning mixtures of distributions*. University of California, Berkeley, 2007.

S. Dasgupta. Learning mixtures of gaussians. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 634–644. IEEE, 1999.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

A. T. Kalai, A. Moitra, and G. Valiant. Efficiently learning mixtures of two gaussians. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 553–562. ACM, 2010.

R. Kannan, H. Salmasian, and S. Vempala. The spectral method for general mixture models. In *International Conference on Computational Learning Theory*, pages 444–457. Springer, 2005.

F. J. Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

G. McLachlan and D. Peel. Mixtures of factor analyzers. *Finite Mixture Models*, pages 238–256, 2000.

A. Moitra and G. Valiant. Settling the polynomial learnability of mixtures of gaussians. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 93–102. IEEE, 2010.

S. Ray and B. G. Lindsay. Model selection in high dimensions: a quadratic-risk-based approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):95–118, 2008.

R. Rossi, S. A. Tarim, S. Prestwich, and B. Hnich. Piecewise linear lower and upper bounds for the standard normal first order loss function. *Applied Mathematics and Computation*, 231:489–502, 2014.

A. Sanjeev and R. Kannan. Learning mixtures of arbitrary gaussians. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 247–257. ACM, 2001.

K. D. Tocher. *The art of simulation.* English Universities Press, 1967.

S. Vempala and G. Wang. A spectral algorithm for learning mixtures of distributions. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 113–122. IEEE, 2002a.

S. Vempala and G. Wang. A spectral algorithm for learning mixtures of distributions. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 113–122. IEEE, 2002b.

C. J. Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103, 1983.

M. Zelen and N. C. Severo. Probability functions. *Handbook of mathematical functions*, 5:925–995, 1964.