Clustering in Misspecified and High-Dimensional Models using Automatic Differentiation

Siva Rajesh Kasa
Dept. of Information Systems and Analytics
National University of Singapore

Vaibhav Rajan
Dept. of Information Systems and Analytics
National University of Singapore

Abstract

Gaussian Mixture Models (GMMs) are used for model-based clustering in a variety of applications. In this paper, we study two practically important but relatively less explored cases of fitting GMMs (1) when there is misspecification and (2) on high-dimensional data. Our empirical analyses reveal limitations of likelihood based model selection in both Expectation Maximization (EM) and Gradient Descent (GD) based inference methods. We design new KL divergence based model selection criteria and GD-based inference methods that use the criteria in fitting GMM on low- and high-dimensional data as well as for selecting the number of clusters. Our experiments on simulated and real datasets show the superiority of our approach over state-of-the-art methods for clustering[1].

1 Introduction

Mixture models are a principled statistical approach to clustering heterogeneous collections of data. Gaussian Mixture Models (GMM) have been studied and used extensively for clustering in statistics and machine learning. Fitting a GMM to the observed data entails solving two problems: (1) Model Selection: choosing the number of components (clusters), and (2) Parameter Inference: estimating the mean, covariance and weight of each Gaussian component. Standard frequentist approaches to solving both the problems are based on the likelihood.

Model selection criteria such as Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) are based on likelihood. Parameter estimation by maximizing the likelihood is non-trivial due to the non-convexity of GMM likelihood. Expectation Maximization (EM) (Dempster et al, 1977) and its variants are widely used to obtain approximate Maximum Likelihood Estimates (MLE) (McLachlan and Krishnan, 2007). Theoretical properties of MLE and EM remain active areas of research (Azizyan et al, 2013; Jin et al, 2016; Balakrishnan et al, 2017; Heinrich et al, 2018).

In this paper, we study two practically important but relatively less studied cases of misspecified models and high-dimensional data. Previous works assume that the true component distribution is Gaussian; this is not true in practice and in such conditions MLE iterates do not converge to the true model (White, 1982). The other case where EM fails is for high-dimensional data, where in the E-step inversion of a matrix that is typically not full-rank is required and there are singularities in the parameter manifolds (Amari et al, 2006). Computational difficulties also arise due to quadratic increase in covariance matrix parameters. To address these problems variable selection methods (Fop et al, 2018) and subspace methods that reduce the number of parameters by imposing constraints on the covariance structure (Bouveyron and Brunet-Saumard, 2014) have been developed; but they fail in high dimensional ($p \gg n$) cases (Kasa et al, 2019). Further, in high dimensions likelihood-based criteria cannot be used for model selection (Giraud, 2014).

In this paper we empirically analyze misspecified models and find that likelihood based criteria for model selection lead to selection of sub-optimal models with poor clustering performance (measured by the Adjusted Rand Index). To address this problem, we design a new criterion based on KL-divergence between the components of a fitted GMM, that penalizes clusterings with high overlap in the components. This criterion is used along with likelihood as an objective function for GMM parameter estimation and, with modifications, for clustering high-dimensional data. A suitable variant, called Maximum absolute Pairwise difference between KL divergence (MPKL) is designed for model selection in high dimensions.

Our algorithms use Gradient Descent (GD) based inference and leverage Automatic Differentiation (AD) tools,

[1] Code available at https://bitbucket.org/cdal/sia/

that are surprisingly under-utilized for parameter inference of GMM (and other complex statistical models). AD-based tools obviate the need to derive closed-form expressions of gradients and can be used to implement both first and second order optimization methods – that also aids traditional likelihood-based GMM inference. Moreover, through the use of AD, the entire suite of optimization algorithms (e.g., SGD, Adam) and programming frameworks (e.g., Pytorch, Tensorflow) that have powered the deep learning revolution becomes available for efficient parameter inference of mixture models.

In fact, in likelihood-based inference itself our experiments show that AD-based GD outperforms EM for both cases of misspecification and high-dimensional data. This, to our knowledge, has not been reported earlier. Using our criterion for inference – we call this algorithm Sequential Initialization Algorithm (SIA) – improves clustering performance further. For high-dimensional data, AD-based GD leads to only one of the components dominating the final clustering. We illustrate this problem and show that it can be addressed with our modified KL-divergence based criterion to further improve clustering performance. We call this algorithm SIA-HD. To summarize, our contributions in this paper are as follows:

- We empirically investigate the robustness to random initialization for mixture models (both misspecified and non-misspecified) in terms of clustering performance. We find that:
  - When there is no misspecification, EM outperforms AD-based GD inference.
  - For misspeicified models, AD-based GD inference outperforms EM.
- Our empirical analysis shows that:
  - Likelihood-based model selection criteria may yield sub-optimal clusterings with EM and GD based inference in misspecified GMMs.
  - For high-dimensional data, where EM fails, AD-based GD leads to dominating components in the fitted GMM that deteriorates clustering performance.
- We design a new KL divergence based criterion that is used as a penalization term in the objective function in our AD-based GD inference (called SIA) to fit a GMM.
- We also derive suitable modifications of this criterion that can be used for parameter inference (SIA-HD) and model selection (MPKL) for GMM with high-dimensional data.
- Our experiments on simulated datasets show that: (1) SIA outperforms both AD-based GD and EM in the case of misspecification, (2) SIA-HD outperforms state-of-the-art GD and EM-based methods on high-dimensional data and (3) MPKL outperforms BIC in model selection.
- A case study on lung cancer patient subtyping also illustrates the advantages of SIA-HD over state-of-the-art clustering methods for high-dimensional data.

2 Background and Related Work

Consider n independent and identically distributed (iid) instances of p-dimensional data, $\mathbf{x} = [x_{ij}]_{n\times p}$ where i denotes the observation, and j denotes the dimension. A K-component mixture model is given by

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k f_k\left(\boldsymbol{x};\boldsymbol{\theta}_k\right), \text{ such that } \sum_{k=1}^{K} \pi_k = 1$$

where $\boldsymbol{\theta}_k$ and π_k are the parameters and weight of the k-th mixture component, f_k, respectively. In a GMM, each individual component f_k is modeled using a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are its mean and covariance respectively. We denote by $\boldsymbol{\theta}$ the complete set of parameters comprising the mean vectors, covariance matrices and weights of all components. Appendix C has a list of symbols used in the paper for reference.

Maximum Likelihood Estimation (MLE) aims to find parameter estimates $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} f(\boldsymbol{\theta})$ where $\boldsymbol{\Theta}$ denotes the overall parameter space. The problems of degeneracy and singularities plague MLE in mixture models, particularly in high dimensions. Degeneracy occurs because one can always assign a single datapoint to a cluster, the variance of this component can be made as small possible (Pan and Shen, 2007). Penalization based approaches to address this problem have been proposed in (Pan and Shen, 2007; Bhattacharya and McNicholas, 2014). Singularities in the parameter space lead to non-identifiability of parameters due to degeneracy of the Fisher information matrix (Amari et al, 2006). See Appendix A for more details.

Model misspecification. Following (White, 1982), if the observed data are n iid samples from $f(\mathbf{X}, \boldsymbol{\theta}^*)$ and the fitted model has the same functional form $f(.)$, then the model is said to be correctly specified. Otherwise, the model is said to be misspecified. It can be shown that, with no misspecification, MLE estimate has the best empirical likelihood among all possible estimates in large sample-sizes. However, when the model is misspecified standard MLE techniques are inconsistent with respect to the true parameter (White, 1982). Recent theoretical results on misspecification in EM-based inference (Tarpey et al, 2008; Dwivedi et al, 2018) do not develop algorithmic solutions to address the problem. See Appendix D for more details.

Misspecification cannot be easily identified for mixture models. White (1982) suggests some tests for detecting misspecification. However, the assumptions in these tests are violated due to non-identifiability of mixture parameters (Bishop, 2006) and in data with near-perfect symmetry (Robert and Casella, 2010). Note that the notion of misspecification is moot in high dimensions because it is not statistically possible to establish that the datapoints are sampled from any given distribution. E.g., consider two points sampled from a 3-dimensional 2-component mixture

model. No statistical test can uniquely establish that these sampled datapoints are from any distribution.

Model Selection. Broadly there are two types of model selection methods: (a) Criterion–based methods, e.g., AIC or BIC, typically favor parsimony in modeling by penalizing overfitting and are easy to implement. Likelihood based criteria are based on asymptotic normality and hence cannot be used in high-dimensions where $n \ll p$ (Amari et al, 2006; Giraud, 2014). In high dimensions, it has been observed that BIC tends to underestimate the number of components whereas AIC overestimates the number of components (Melnykov et al, 2010). AIC assumes that there is no misspecification and for misspecified models, Takeuchi Information Criterion (Takeuchi, 1976) has been proposed. that assumes large sample sizes which makes it unsuitable when $n \ll p$. (b) Test–based methods such as Likelihood Ratio Test and bootstrapping, favor complex models but are computationally expensive (Feng and McCulloch, 1996).

Expectation Maximization. EM avoids singularities in low-dimensions when the clusters are well separated (Park and Ozeki, 2009). In high dimensions (HD), inference is difficult as the number of parameters grows quadratically with the dimension. In the E-step, the weighted sample covariance matrix has to be inverted (to compute the normal density) to calculate the conditional probability of the cluster indicator variable (z_{nk}):

$$\gamma\left(z_{nk}\right)=\frac{\pi_k \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^K \pi_j \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}$$

In HD, this matrix does not usually have a full rank, because the rank of $p \times p$ covariance matrix $\mathbf{X}\mathbf{X}^T$ is $\min(n, p) = n$. This is another reason why EM fails in HD. See (Bouveyron and Brunet-Saumard, 2014) for a review on clustering of high-dimensional data.

Broadly, two categories of EM-based approaches have been developed for high-dimensional data: *Subspace Clustering* methods cluster data and simultaneously attempt to reduce locally each cluster's dimensionality. Mixture of factor analyzers (MFA) (Ghahramani et al, 1996; McLachlan et al, 2003) addresses the problem of overparameterization by constrained covariance structures through a family of parsimonious Gaussian mixture models (PGMMs; McNicholas and Murphy, 2008; McNicholas et al, 2010): $\boldsymbol{\Sigma}_k = \boldsymbol{\Lambda}_k \boldsymbol{\Lambda}_k' + \boldsymbol{\Psi}_k$, where $\boldsymbol{\Lambda}_k$ is a $p \times q$ loading matrix with $q << p$ and $\boldsymbol{\Psi}_k$ is a diagonal noise matrix. Bouveyron et al (2007) developed HDDC, that uses a combination of subspace clustering and parsimonious modeling for GMMs. *Variable Selection* methods select relevant variables for clustering, that are assumed to primarily determine the cluster structure in the data. Fop et al (2018) has a recent review. Various kinds of penalizations are used to select the variables. Recent examples for HD include VarSelLCM (Marbac and Sedki, 2017) and MixGlasso (Städler et al, 2017). A new algorithm for high-dimensional GMMs, CHIME (Cai et al, 2019), uses a modified E-step that obviates the need to compute the inverse of the covariance matrix. However, it assumes equicovariance in all clusters, and this modified E-step requires solving an expensive quadratic optimization problem.

Gradient Descent and Autograd. It has been shown that EM is a special case of Gradient Descent (GD) (Xu and Jordan, 1996). Use of AD tools as black-box gradient-based solvers overcomes the difficulty of evaluating closed form expressions for the gradients which are intractable in complex models. We refer the reader to (Baydin et al, 2018) for a survey on AD and to Appendix B for a brief description.

Since we will use AD-based GD for our algorithms later, we outline how AD can solve the same problems that EM elegantly solves to obtain the MLE of GMMs (Salakhutdinov et al, 2003): (1) Intractability of evaluating the closed-forms of the derivatives, (2) Ensuring positive definiteness (PD) of the covariance estimates $\hat{\boldsymbol{\Sigma}}_k$, and (3) Ensuring the constraint on the component weights ($\sum_k \hat{\pi}_k = 1$).

Problem 1 is solved inherently because we do not need to express the gradients in closed form to use AD. Further, we can also evaluate the second-order Hessian matrix using AD, that enables us to use methods that converge faster.

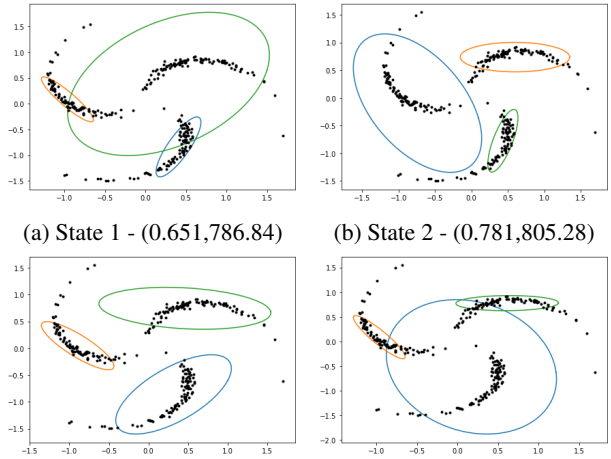
To tackle *Problem 2*, instead of gradients with respect to $\boldsymbol{\Sigma}_k$, we compute the gradients with respect to $\mathbf{U}_k$, where $\boldsymbol{\Sigma}_k = \mathbf{U}_k \mathbf{U}_k^T$. We first initialize $\mathbf{U}_k$ as identity matrices. Thereafter, we keep adding the gradients to the previous estimates of $\hat{\mathbf{U}}_k^{t+1}$, i.e.

$$\hat{\mathbf{U}}_k^{t+1} := \hat{\mathbf{U}}_k^t + \alpha \frac{\partial \mathcal{L}}{\partial \mathbf{U}_k};\ \hat{\boldsymbol{\Sigma}}_k^{t+1} := \hat{\mathbf{U}}_k^{t+1} \hat{\mathbf{U}}_k^{t+1^T} \tag{1}$$

where α is the learning rate. If the gradients are evaluated with respect to $\boldsymbol{\Sigma}_g$ directly, there is no guarantee that updated $\hat{\boldsymbol{\Sigma}}_k^{t+1} = \hat{\boldsymbol{\Sigma}}_k^t + \alpha \frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k}$ will still remain PD. However, if the gradients are evaluated with respect to $\mathbf{U}_k$, by construction $\hat{\boldsymbol{\Sigma}}_k^{t+1}$ will always remains PD. Cholesky decomposition for reparameterizing $\boldsymbol{\Sigma}_k$ can also be used. In the case of MFA models, there is no need to factorize $\boldsymbol{\Sigma}_k$ as the model ensures PD covariance matrix estimates.

Problem 3 is solved by using the log-sum-exp trick (Robert, 2014). We start with unbounded α_k as the log-proportions: $\log \pi_k = \alpha_k - \log(\sum_i e^{\alpha_i})$. We need not impose any constraints on α_g as the final computation of π_g automatically leads to normalization, because $\pi_k = \frac{\alpha_k}{\sum_i e^{\alpha_i}}$. Therefore, we reparametrize a constrained optimization problem into an unconstrained one (without using Lagrange multipliers) and we can update $\hat{\pi}_k$ as follows:

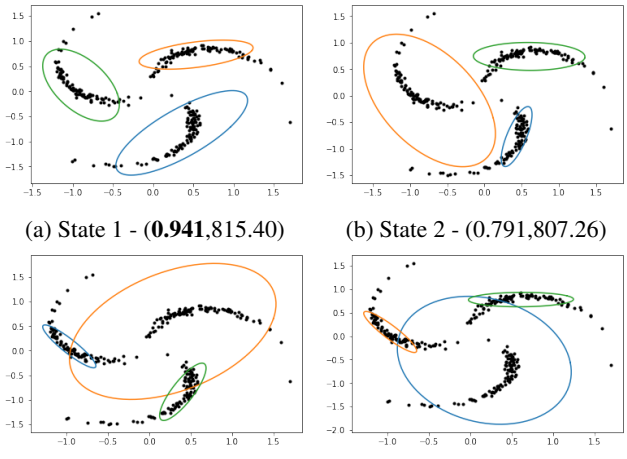
$$\hat{\alpha}_k^{t+1} := \hat{\alpha}_k^t + \alpha \frac{\partial \mathcal{L}}{\partial \alpha_k};\ \hat{\pi}_k^{t+1} := \frac{e^{\hat{\alpha}_k^{t+1}}}{\sum_i e^{\hat{\alpha}_i^{t+1}}} \tag{2}$$



(a) State 1 - (0.651,786.84) (b) State 2 - (0.781,805.28)

(c) State 3 - (**0.912**,781.08) (d) State 4 - (0.625,**771.90**)

Figure 1: Suboptimal EM clusterings in misspecified mixtures; in parentheses: (ARI, AIC), best values in bold.



(a) State 1 - (**0.941**,815.40) (b) State 2 - (0.791,807.26)

(c) State 3 - (0.651,788.32) (d) State 4 - (0.625,**773.94**)

Figure 2: Suboptimal SOGD clusterings in misspecified mixtures; in parentheses: (ARI, AIC), best values in bold.

3 Misspecification and High Dimensions

In this section we illustrate the problems that arise from (1) fitting misspecified models and (2) high-dimensional data in both EM and AD-based methods.

3.1 Low-dimensional misspecified models

Consider the 3-component, 2-dimensional pinwheel dataset (i.e. warped GMM) (Johnson et al, 2016; Lin et al, 2018) (having 300 samples). We fit a GMM using two different inference techniques: EM and Second-Order Gradient Descent (SOGD), through AD. We run both the algorithms using 100 random initialzations and the same stopping criterion (convergence threshold $1e-5$). For EM, for these 100 different initializations, there are predominately four different output clustering states as shown in fig. 1. We observe that the best (highest) ARI is for the 3rd clustering while the best (lowest) AIC value is for the 4th clustering. Fig. 2 shows the clusterings obtained through SOGD. Here, too, the best ARI (1st clustering) and best AIC (4th clustering) do not match. The key problem that in fitting misspecified models is that likelihood-based criterion (AIC/BIC) tend to select sub-optimal models, i.e., models which have poor clustering performance measured in terms of Adjusted Rand Index (ARI).

3.2 High Dimensional Data

For high-dimensional data, EM-based methods fail due to the reasons described earlier. We observe that using AD-based methods to maximize the likelihood leads to only one component 'dominating' the final clustering. The final dominating component depends on the initialization. In our experiments, we have noticed that in low dimensions, this problem does not arise when initialized with k-means. This is a well-known phenomenon that happens because the likelihood is unbounded, and, in this case, increases by fitting the covariance matrix of a component on just one datapoint (Bishop, 2006; Kim and Seo, 2014). Intuitively, the likelihood of a GMM can increase by (a) decreasing the variance over a single datapoint or (b) extending a better fit over a large number of points. In low dimensions, if the initialization is reasonable (e.g. outliers are not given a single cluster), the optimization algorithm chooses (b) over (a) because to decrease the variance over a single point, a component has to 'let go' of many datapoints, i.e., the algorithm has to cross multiple local optima in between. In high dimensions, the datapoints are sparsely scattered and there are few datapoints to let go of; hence the optimization algorithm chooses (a) over (b), i.e., fits a cluster over a single datapoint by decreasing the variance of the component and simultaneously increasing its component weight π_k.

In high dimensions, visualization is not possible, so we examine the determinants of covariance matrices. We illustrate this dominating component phenomenon using the SCADI dataset (Zarchi et al, 2018) containing 206 features and 70 observations. We fit the MFA model using AD on this dataset and find that a single cluster is obtained comprising all the 70 datapoints. Fig. 4 (a) illustrates this over multiple iterations: soon after initialization, the variance of one of the components increases to capture all the near-by points and then, the variance of this particular component is reduced while at the same time the corresponding component weight is increased, to maximize the likelihood.

3.3 Robustness to initialization

In this subsection, we take a brief detour and compare the robustness to initialization of GD-based inference and EM on pure GMM data as well as non-GMM (misspecified)

data. We do so by simulating test datasets as described below and running both the algorithms with matched settings with respect to random initialization, number of iterations (maximum of 100 iterations) and convergence threshold blue(1e-5). Both the methods are run on each dataset with 30 random initializations.

Pure GMM data : We simulated 36 (6 X 6) different GMM datasets by varying the means and covariance matrices for 3 component mixture models using scikit-learn package in Python. This is to test most possible combinations - from well-separated clusters to highly overlapping clusters. The GMM parameters are varied as follows:

- First, a 2×1 vector $\mathbf{v}_i$ and a 2×2 matrix $\mathbf{Z}_i$ for each of these 3 components is chosen randomly.
- The mean vector of the ith component is obtained by multiplying $\mathbf{v}_i$ by a factor $k_{\boldsymbol{\mu}}$. This factor $k_{\boldsymbol{\mu}}$ is varied from 0.25 to 1.5 in steps of 0.25, i.e., the factor $k_{\boldsymbol{\mu}}$ takes one of the six values in $\{0.25, 0.5, 0.75, 1, 1.25, 1.5\}$.
- The $\mathbf{Z}_i$ vector is added to the 2×2 Identity matrix $\mathbf{I}$ (This step ensures the PD of the covariance matrix that is computed in the later steps).
- We multiply the matrix $\mathbf{Z}_i + \mathbf{I}$ by $k_{\boldsymbol{\Sigma}}$ to obtain $\mathbf{U}_i$. The scaling factor $k_{\boldsymbol{\Sigma}}$ is varied from 0.05 to 0.65 in steps of 0.1, i.e., the scaling factor takes one of the six values in $\{0.05, 0.15, 0.25, 0.35, 0.45, 0.55\}$. The covariance matrix of the i$^t h$ component is $\boldsymbol{\Sigma}_i = \mathbf{U}_i \mathbf{U}_i^T$.

By choosing different values of $k_{\boldsymbol{\mu}}$ and $k_{\boldsymbol{\Sigma}}$, we simulate 36 (6x6) GMM parameters with varying cluster separation. For each these 36 GMM parameters, we simulate 30 different datasets.

Misspecified datasets: :

The pinwheel dataset is characterized by two parameters - a tangential component and a radial component. We simulated 36 (6 X 6) different pinwheel datasets by varying the radial and tangential components for 3 component mixture models using Autograd package in Python. This is to test most possible combinations – clusters with tails to warped mixtures as shown in figure 3. The parameters of pinwheel datasets are chosen as follows:

- The radial parameter r is chosen to be one of the six values { 0.05, 0.15, 0.25, 0.35, 0.45, 0.55 }
- The tangent parameter t is chosen to be one of the six values { 0.05, 0.15, 0.25, 0.35, 0.45, 0.55 }

By choosing different values of r and t, we simulate 36 (6x6) pinwheel parameters with varying shapes. For each these 36 pinwheel parameters, we simulate 30 different datasets.

Table 1: Results of GMM datasets

$k_{\boldsymbol{\mu}} - k_{\boldsymbol{\Sigma}}$	0.05	0.15	0.25	0.35	0.45	0.55
0.25	-0.71	-0.595	-0.49	-0.425	-0.395	-0.38
0.5	-0.58	-0.63	-0.57	-0.515	-0.485	-0.465
0.75	-0.545	-0.64	-0.605	-0.575	-0.545	-0.495
1	-0.5	-0.62	-0.655	-0.615	-0.57	-0.55
1.25	-0.425	-0.57	-0.63	-0.63	-0.60	-0.57
1.5	-0.38	-0.54	-0.61	-0.635	-0.64	-0.605

Table 2: Results of Pinwheel datasets

$r - t$	0.05	0.15	0.25	0.35	0.45	0.55
0.05	0.025	0.015	0.005	-0.01	0.025	0.44
0.15	0.03	0.015	0.0	-0.015	0.02	0.295
0.25	0.075	0.025	0.0	-0.03	-0.025	0.105
0.35	0.29	0.19	0.11	0.03	0.04	0.08
0.45	0.37	0.32	0.25	0.195	0.145	0.105
0.55	0.335	0.3	0.235	0.175	0.135	0.105

3.4 Results

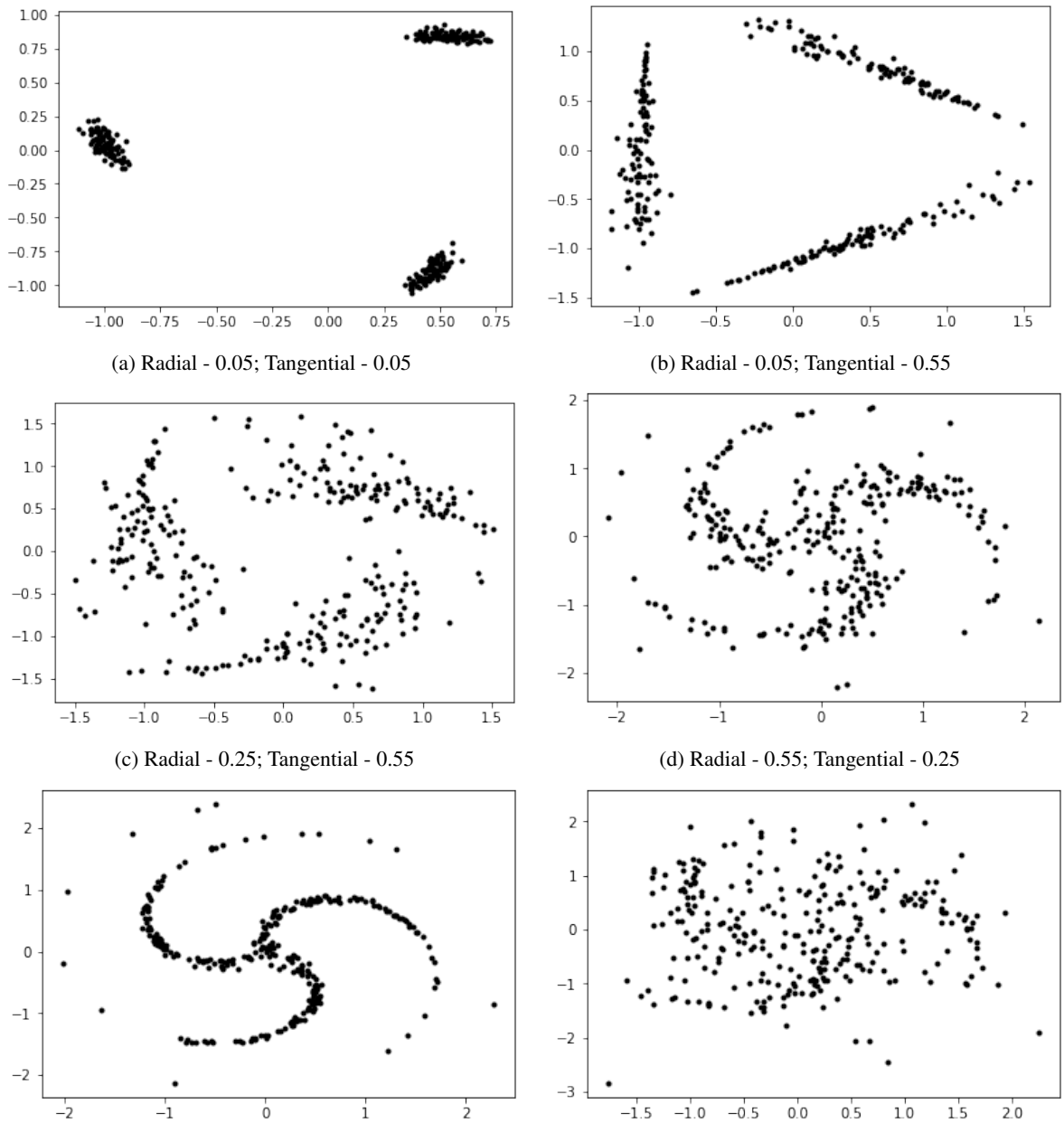
From tables 1 and 2, we observe:

- When there is no misspecification, EM outperforms AD-based GD inference.
- For misspecified models, AD-based GD inference outperforms EM.

The mathematical intuition behind these results is as follows: In EM, the E-step and M-step can be thought of a set of simultaneous equations. At convergence, given the mean and covariance parameters, mixture weights can be determined automatically (EM assumes that the data is sampled from a GMM), i.e., mixture weights are not free estimates at convergence. However, in GD based methods, all the parameters (including the mixture weights) are free parameters at every iteration, i.e., no constraint is imposed on satisfying simultaaneous equations; hence, in misspecified models, GD based inference gives a better clustering performance. **EM estimates converge quickly because it performs optimization over a constrained parameter space by utilizing the structure of the model; this computational speed comes at a cost - its clustering performance deteriorates when there is misspecification.**

4 Our Algorithms

As illustrated earlier, likelihood-based criteria do not always select the optimal clustering for misspecified models. To address this problem we design a new model selection criterion based on the Kullback Leibler (KL) divergence



(a) Radial - 0.05; Tangential - 0.05

(b) Radial - 0.05; Tangential - 0.55

(c) Radial - 0.25; Tangential - 0.55

(d) Radial - 0.55; Tangential - 0.25

(e) Radial - 0.55; Tangential - 0.05

(f) Radial - 0.55; Tangential - 0.55

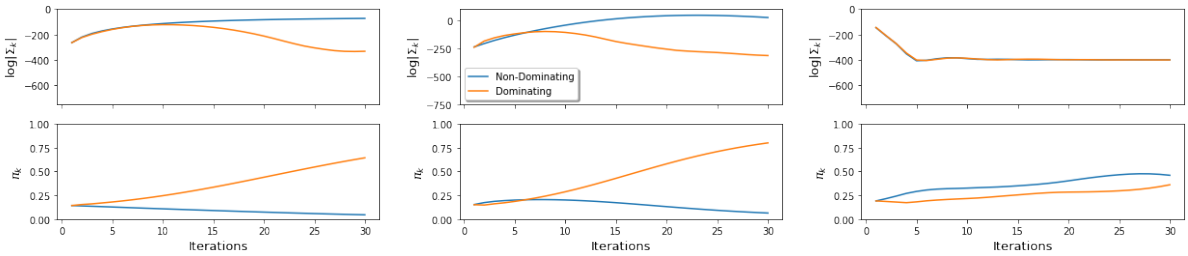
Figure 3: Data with completely different shapes can be simulated by varying the tangential and radial parameters of pinwheel dataset.

between Gaussian distributions. Our algorithm, called Sequential Initialization Algorithm (SIA), uses this criterion within its objective function and through AD-based Gradient Descent fits a GMM to obtain the final clustering.

The KL-divergence between two multivariate Gaussian distributions, $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, is given by

$$KL\left(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)\right) = \frac{1}{2}\left[\log\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} - d + \mathrm{tr}\{\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\} + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\right]$$

As seen above KL-divergence is not symmetric about $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$. If there is an order of magnitude difference in the covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, it can detected through the values of $KL\left(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)\right)$ and $KL\left(\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\right)$ – the difference in their values is primarily contributed by the difference between the terms $(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)$ and $(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)$, and $\mathrm{tr}\{\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\}$ and $\mathrm{tr}\{\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2\}$. Thus the difference in these two KL divergence values provides signal about the overlap or asymmetry of the two Gaussians. To gener-



(a) MFA (AD): ARI = 0.0
Loglikelihood = -2352

(b) SIA: ARI = 0.0
Loglikelihood = -4246

(c) SIA-HD: ARI = 0.297
Loglikelihood = -1074581

Figure 4: In HD ($p >> n$): (a) maximizing likelihood leads to reduction in variance (above) and corresponding increase in weight (below) resulting in a *dominating component* (in orange), (b) SIA: KLF and KLB are reduced by increasing the variance of non-dominating component in SIA, (c) SIA-HD: there is no dominating component in the final clustering. See text for details.

alize this notion to a K-component GMM, we define two combinatorial KL divergences – KLF (forward) and KLB (backward) as follows:

$$KLF := \sum_{i<j} KL\left(\mathcal{N}_i, \mathcal{N}_j\right); \; KLB := \sum_{i>j} KL\left(\mathcal{N}_i, \mathcal{N}_j\right)$$

where $\mathcal{N}_i$ denotes the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. Well separated clusters typically have equal and lower values of KLF and KLB. We denote both the values by $KLDivs = \{KLF, KLB\}$. We note that these two sums (KLF + KLB) together give the sum of Jeffrey's divergence between all components.

For example, for a 3 component GMM,

$$KLF = KL(\mathcal{N}_1, \mathcal{N}_2) + KL(\mathcal{N}_2, \mathcal{N}_3) + KL(\mathcal{N}_1, \mathcal{N}_3)$$

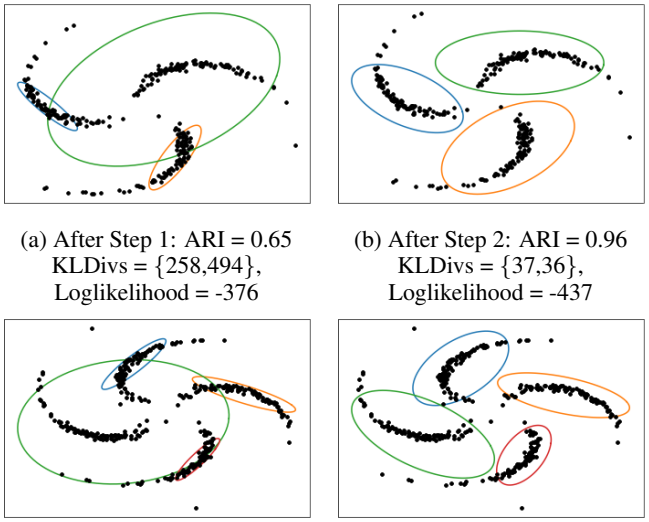
$$KLB = KL(\mathcal{N}_2, \mathcal{N}_1) + KL(\mathcal{N}_3, \mathcal{N}_2) + KL(\mathcal{N}_3, \mathcal{N}_1)$$

In the clustering shown in state 3 in fig. 2, we can visually see that cluster separation is poor. The KLF and KLB values are found to be unequal: KLDivs = {258,494}. In the following sections we use the combinatorial KL divergence to design our clustering algorithms.

SIA. SIA consists of two steps. In the first step of SIA, we use likelihood $\mathcal{L}$ as the objective and run AD-based (first or second order) Gradient Descent to fit a GMM. Typically, the output at the end of first step will have unequal KL-divergences for misspecified models. We take these parameters at the end of the first step to initialize the algorithm in second step. In the second step we modify the objective function to:

$$\mathcal{M} = \mathcal{L} - w_1 \times KLF - w_2 \times KLB$$

Notice that the weights $(-w_1, -w_2)$ are negative. If we add positive weights GD will further shrink the smaller clusters. Negative weights lead to separation as well as clusters of similar volume. We can also view the negative weights as regularization terms to prevent overfitting. After the second optimization step the likelihood decreases but the KL-divergence values, KLF and KLB, come closer. Thus by trading off likelihood for better cluster separation SIA achieves better clustering performance.



(a) After Step 1: ARI = 0.65
KLDivs = {258,494},
Loglikelihood = -376

(b) After Step 2: ARI = 0.96
KLDivs = {37,36},
Loglikelihood = -437

(c) After Step 1: ARI = 0.61
KLDivs = {1138,1038},
Loglikelihood = -531

(d) After Step 2: ARI = 0.93
KLDivs = {158,128},
Loglikelihood = -599

Figure 5: Clustering using SIA in pinwheel data: 3 components (above), 4 components (below).

Figures 5 shows the clustering obtained on 3-component and 4-component pinwheel data. In each case, we observe the difference in the clustering after the first and second steps of SIA. Note that after the second step, compared to the clustering after step 1, the likelihood decreases, both the KLF and KLB values decrease, the ARI increases, and the clusters appear to be better visually.

SIA-HD. The objective function $\mathcal{M}$ does not address the problem of dominating components in high dimensions. This is because it tends to increase the variance of the non-dominating components to reduce KLF and KLB, as shown in fig. 4 (b). To tackle this problem, we modify the objec-

tive function to penalize very small covariance matrices. The penalty function h is:

$$h(\mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_K, \lambda_1, \ldots, \lambda_K) = \sum_{k=1}^{K} (\det(\mathbf{\Sigma}_k) - \lambda_k)^2$$

The new objective function that is maximized is:

$$\mathcal{L} - w_1 \times KLF - w_2 \times KLB - w_3 \times h(\lambda)$$

To distinguish between the low-dimensional and high-dimensional cases, we call this algorithm SIA-HD. The complete algorithm is presented in Algorithm 1. Running SIA-HD on the SCADI dataset gives an ARI of 0.15 (compared to 0 for SIA and MFA); furthermore, there is no dominating component in the final clustering (fig. 4 (c)).

Model Section Criterion. We define the maximum absolute pairwise difference between KL divergence values (MPKL) for a K-component GMM as follows:

$$MPKL = \max_{i,j \leq K} |KL(\mathcal{N}_i, \mathcal{N}_j) - KL(\mathcal{N}_j, \mathcal{N}_i)|$$

As discussed earlier, it is an indicator of how well the separation between clusters is. This can be used as a criterion for selecting the number of clusters. For a chosen range of number of clusters $(2, \ldots, L)$, we compute MPKL for each value and choose K that minimizes MPKL:

$$\text{argmin}_{K \in [2, \ldots, L]} MPKL$$

Computational Complexity. The computational complexity is dominated by evaluating KLF and KLB which involves $O(K^2)$ matrix inversion steps ($O(p^3)$). Therefore, the overall computational complexity of all our algorithms (SIA,SIA-HD and model selection) is $O(K^2 p^3)$.

Algorithm 1 SIA

Input: Observed n datapoints (as $n \times p$ dimensional matrix $X = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_p)$) and number of clusters K and tolerance γ for convergence

Initialize: $\hat{\boldsymbol{\mu}}_i^0$'s using K-Means or random initialization; $\hat{\mathbf{U}}_i^0$'s as identity matrices; $\hat{\alpha}_i$'s using K-Means or random initialization;

Step I

Objective:

$$\text{argmax}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}|\mathbf{X})$$

REPEAT: At every iteration $t+1$,

$$\hat{\alpha}_k^{t+1} := \hat{\alpha}_k^t + \alpha \frac{\partial \mathcal{L}}{\partial \alpha_k};\ \hat{\pi}_k^{t+1} := \frac{\hat{\alpha}_k^{t+1}}{\sum_i e^{\hat{\alpha}_i^{t+1}}};\ \hat{\boldsymbol{\mu}}_k^{t+1} := \hat{\boldsymbol{\mu}}_k^{t+1} + \alpha \frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k}$$

$$\hat{\mathbf{U}}_k^{t+1} := \hat{\mathbf{U}}_k^t + \alpha \frac{\partial \mathcal{L}}{\partial \mathbf{U}_k};\ \hat{\mathbf{\Sigma}}_k^{t+1} := \hat{\mathbf{U}}_k^{t+1} \hat{\mathbf{U}}_k^{t+1^T}$$

STOP: when convergence criterion $|\mathcal{L}^{t+1} - \mathcal{L}^t| < \gamma$ is met

Step II

For Low Dimensional Data:

$$\mathcal{M} = \mathcal{L} - w_1 \times KLF - w_2 \times KLB$$

For High-Dimensional Data: Identify the dominating components based on the determinant of $\mathbf{\Sigma}_i$'s and formulate the penalty terms λ_k's accordingly. Set

$$\mathcal{M} = \mathcal{L} - w_1 \times KLF - w_2 \times KLB - w_3 \times h(\lambda)$$

$$h(\mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_K, \lambda_1, \ldots, \lambda_K) = \sum_{k=1}^{K} (\det(\mathbf{\Sigma}_k) - \lambda_k)^2$$

Initialize: Use Output of Step I

REPEAT: At every iteration $t+1$,

$$\hat{\alpha}_k^{t+1} := \hat{\alpha}_k^t + \alpha \frac{\partial \mathcal{M}}{\partial \alpha_k};\ \hat{\pi}_k^{t+1} := \frac{\hat{\alpha}_k^{t+1}}{\sum_i e^{\hat{\alpha}_i^{t+1}}};\ \hat{\boldsymbol{\mu}}_k^{t+1} := \hat{\boldsymbol{\mu}}_k^{t+1} + \alpha \frac{\partial \mathcal{M}}{\partial \boldsymbol{\mu}_k}$$

$$\hat{\mathbf{U}}_k^{t+1} := \hat{\mathbf{U}}_k^t + \alpha \frac{\partial \mathcal{M}}{\partial \mathbf{U}_k};\ \hat{\mathbf{\Sigma}}_k^{t+1} := \hat{\mathbf{U}}_k^{t+1} \hat{\mathbf{U}}_k^{t+1^T}$$

STOP: when convergence criterion $|\mathcal{M}^{t+1} - \mathcal{M}^t| < \gamma$ is met

5 Experimental Results

We empirically evaluate the clustering performance of our algorithm and model selection criterion on simulated and real datasets. We use Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) to evaluate clustering performance where higher values indicate better clustering.

5.1 Simulations

Low-Dimensions: Here our aim is to evaluate clustering performance when there is model misspecification. To this end, we simulate 10 datasets from a 2-dimensional mixture model with 3 Gaussian components, $\mathcal{N}((\lambda, \lambda), \mathbb{I}_2), \mathcal{N}((-\lambda, \lambda), \mathbb{I}_2), \mathcal{N}((\lambda, -\lambda), \mathbb{I}_2)$, where λ is a scaling factor which controls the cluster separation and $\mathbb{I}_2$ is a unit covariance matrix. We evaluate the performance for λ values $\{3, 4, 5\}$. We sample 100 datapoints from each of these 3 components. These sampled datapoints are cubed so none of the components is normally distributed. Now we fit a *misspecified* 3-component GMM using three different inference techniques – EM, Gradient Descent using AD tools (AD-GD) and our SIA algorithm.

High Dimensions: Following (Hoff, 2004, 2005; Pan and Shen, 2007), we simulate 10 datasets with 100 datapoints from a 2-component GMM, with 50 datapoints from each component. The dimensionality (p) values evaluated are $\{50, 100, 200\}$. Both the components have unit spherical covariance matrices but differ in their mean vectors. 90% of the dimensions of the mean vector are kept same (with value 0) across both the components and 10% of the dimen-

sions of the mean vector differ (0 in one and 1 in the other cluster). Hence, for the purpose of clustering, only 10% of the dimensions are discriminating features and remaining 90% of the dimensions are noise features. We compare the performance of EM and AD-based GD methods (on both GMM and MFA) with that of SIA.

Model Selection in High Dimensions: We follow the approach used in (Hoff, 2004, 2005; Pan and Shen, 2007; Wang and Zhu, 2008; Guo et al, 2010) and simulate a 4 component 50-dimensional dataset as shown in table 3. We assume unit spherical covariance matrices for each component. Further, the parameters of each component across the 50 dimensions are varied as shown in table 3, i.e., all the components have the same parameters for 35 dimensions, for component 2 only dimensions 1-5 are discriminating, for component 3 only dimensions 5-10 are discriminating and for component 4 only dimensions 10-15 are discriminating. The value of λ controls the cluster separation. For each component we sampled ten datapoints. Three different sets of simulated data with varying cluster separation are obtained by choosing the value of λ to be 1, 5 and 10 respectively. This experiment is repeated with 10 different seeds for each value of λ. We use SIA-HD for clustering with different values of K $\{3,4,5\}$. We compare the number of clusters selected using BIC and MPKL.

Table 3: Simulations for Model Selection

Features	Cluster-1	Cluster-2	Cluster-3	Cluster-4
1-5	$\mathcal{N}(0,1)$	$\mathcal{N}(\lambda,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$
5-10	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(\lambda,1)$	$\mathcal{N}(0,1)$
10-15	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(\lambda,1)$
15-50	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$

Results

Table 4 shows the ARI obtained by SIA and baseline algorithms by fitting (misspecified) GMMs on low-dimensional datasets with varying cluster separation (λ). In all three cases, AD-GD outperforms EM and SIA outperforms AD-GD, with the improvement increasing with increasing cluster separation.

Table 4: ARI (mean and SD) for varying λ

λ	3	4	5
SIA	0.458	0.812	0.969
	(0.082)	(0.093)	(0.039)
AD-GD	0.429	0.686	0.856
	(0.092)	(0.122)	(0.076)
EM	0.162	0.416	0.548
	(0.112)	(0.061)	(0.053)

Table 5 shows the ARI obtained by SIA-HD and baseline algorithms by fitting GMMs and MFA on high-dimensional datasets. EM fails in high dimensions while all the AD-GD based algorithms run to completion. SIA-HD outperforms AD-GD for both GMM and MFA.

Table 5: ARI (mean and SD) for varying n/p

n/p	100/200	100/100	100/50
SIA-HD (GMM)	0.801	0.517	0.333
	(0.058)	(0.131)	(0.168)
AD-GD (GMM)	0.738	0.474	0.239
	(0.051)	(0.135)	(0.146)
SIA-HD (MFA)	0.753	0.330	0.250
	(0.086)	(0.114)	(0.180)
AD-GD (MFA)	0.523	0.275	0.124
	(0.119)	(0.069)	(0.113)
EM (GMM)	-	-	-
EM (MFA)	-	-	0.098
			(0.128)

Table 6: Results on Model Selection

λ	# clusters:	3	4	5
1	MPKL	10	0	0
1	BIC	10	0	0
5	MPKL	1	7	2
5	BIC	10	0	0
10	MPKL	1	8	1
10	BIC	10	0	0

Table 6 shows that at very low cluster separation ($\lambda = 1$), both the criteria do not select 4 clusters. At moderate and high cluster separation ($\lambda = 5, 10$), BIC always underestimates the number of clusters to 3, which is consistent with previous findings on BIC (Melnykov et al, 2010). In contrast, MPKL identifies 4 clusters in 7 out of 10 times in moderate separation and 8 out of 10 times in high cluster separation.

5.2 Case Study: Lung Cancer Patient Subtyping

The LUSCRNA-seq data (Kandoth et al, 2013) is a preprocessed sample of gene expression and methylation data as well as selected clinical covariates such as gender, cigarette packs smoked per year, survival time, and survival status (206 attributes in total) for 130 patients with lung squamous cell carcinoma (LUSC) as available from The Cancer Genome Atlas (TCGA) database (Weinstein et al, 2013).

The MPKL criterion identifies 2 clusters:

# of clusters	2	3	4
MPKL	6.739	27.26	9.74

We compare the performance of SIA-HD using MFA with state-of-the-art model-based clustering algorithms HDDC (Bouveyron and Brunet-Saumard, 2014), VarSelLCM

(Marbac and Sedki, 2017) and MixGlasso (Städler et al, 2017). We use Gaussian mixture models (GMM) and K-means as additional baselines. For all the EM-based algorithms, 5 different K-Means initializations were used, where K-Means itself uses a random initialization. The result with the best BIC is reported. Table 7 shows the ARI obtained by all the methods. EM and HDDC fail to run. Among the other methods, SIA-HD achieves the best ARI. MixGlasso has the next best ARI.

Table 7: Clustering performance

Algorithm	ARI
SIA-HD (MFA)	0.0228
EM (MFA)	-
HDDC	-
Mixglasso	-0.01
VarSelLCM	-0.08

We evaluate the clusters obtained on by SIA-HD and MixGlasso, with respect to their clinical subtypes, in terms of survival probability. The survival probabilities are estimated using Kaplan-Meier method and the p-values are obtained by the log-rank test on the survival analysis based on the cluster memberships. Fig. 6 shows the survival curves obtained on clusters from SIA-HD and MixGlasso. We observe that only SIA-HD yields significant clusters (p-value < 0.05) that are the least overlapping indicating a good separation of patient subtypes.

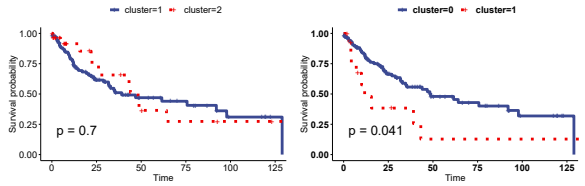
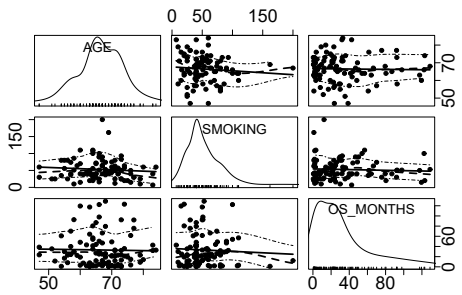


Figure 6: Kaplan-Meier survival curves with p-values;Clusters obtained from MixGlasso (left) and SIA-HD (right).

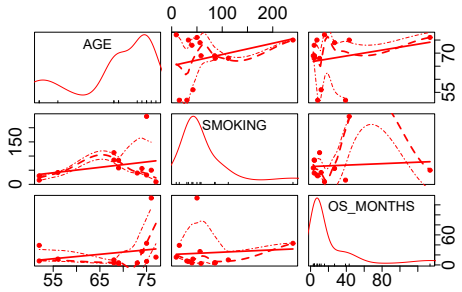
Fig. 7 shows bivariate dependencies in the two clusters from SIA-HD, thereby illustrating its ability to find meaningful patterns. Each scatterplot shows pairwise dependencies and the univariate distribution (diagonal). The bivariate pattern between smoking and survival is distinctly different between the two clusters: in the cluster with higher survival probability patients tend to smoke less when they are older.

6 Conclusion

In this paper we designed a model selection criterion based on combinatorial KL divergences between pairwise distributions in a mixture model. Using this criterion we developed algorithms SIA, SIA-HD for clustering and MPKL for selecting the number of clusters. Experiments on simulations and real datsets demonstrate the efficacy of our



(a) High Survival



(b) Low Survival

Figure 7: Bivariate scatterplots of three features – age (in years), smoking (number of packs per year) and survival (in months) – for two clusters obtained from SIA-HD.

methods in the practically important cases of misspecification and high dimensions. Our algorithms and experiments also illustrate the benefits of automatic differentiation (AD), that is well known but rarely used, for parameter inference in mixture models. SIA can be used for many families of mixture models (e.g., MFA and PGMM), as different constraints imposed by these models can be easily handled by AD. A limitation of SIA is that when cluster covariances are asymmetric in terms of their volume, its performance deteriorates. Future work can address this limitation and explore several extensions such as the use of our model selection criterion for other models. Theoretical properties such as convergence and sample complexity can be investigated.

References

Akaho S, Kappen HJ (2000) Nonmonotonic generalization bias of Gaussian mixture models. Neural Computation 12(6):1411–1427

Amari SI, Park H, Ozeki T (2006) Singularities affect dynamics of learning in neuromanifolds. Neural Computation 18(5):1007–1065

Azizyan M, Singh A, Wasserman L (2013) Minimax theory for high-dimensional Gaussian mixtures with sparse mean separation. In: Advances in Neural Information Processing Systems, pp 2139–2147

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

Baydin AG, Pearlmutter BA, Radul AA, Siskind JM (2018) Automatic differentiation in machine learning: a survey. Journal of Machine Learning Research 18:1–43

Bhattacharya S, McNicholas PD (2014) A LASSO-penalized BIC for mixture model selection. Advances in Data Analysis and Classification 8(1):45–61

Bishop CM (2006) Pattern recognition and machine learning. Springer-Verlag New York

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

Bouveyron C, Girard S, Schmid C (2007) High-dimensional data clustering. Computational Statistics & Data Analysis 52(1):502–519

Cai D, Campbell T, Broderick T (2017) Finite mixture models are typically inconsistent for the number of components. In: NeurIPS Workshop on Advances in Approximate Bayesian Inference (AABI)

Cai TT, Ma J, Zhang L, et al (2019) Chime: Clustering of high-dimensional Gaussian mixtures with EM algorithm and its optimality. The Annals of Statistics 47(3):1234–1267

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological) 39(1):1–22

Dwivedi R, Khamaru K, Wainwright MJ, Jordan MI, et al (2018) Theoretical guarantees for EM under misspecified Gaussian mixture models. In: Advances in Neural Information Processing Systems, pp 9681–9689

Feng ZD, McCulloch CE (1996) Using bootstrap likelihood ratios in finite mixture models. Journal of the Royal Statistical Society: Series B (Methodological) 58(3):609–617

Fop M, Murphy TB, et al (2018) Variable selection methods for model-based clustering. Statistics Surveys 12:18–65

Ghahramani Z, Hinton GE, et al (1996) The EM algorithm for mixtures of factor analyzers. Tech. rep., Technical Report CRG-TR-96-1, University of Toronto

Giraud C (2014) Introduction to high-dimensional statistics. Chapman and Hall/CRC

Guo J, Levina E, Michailidis G, Zhu J (2010) Pairwise variable selection for high-dimensional model-based clustering. Biometrics 66(3):793–804

Heinrich P, Kahn J, et al (2018) Strong identifiability and optimal minimax rates for finite mixture estimation. The Annals of Statistics 46(6A):2844–2870

Hill BM (1963) Information for estimating the proportions in mixtures of exponential and normal distributions. Journal of the American Statistical Association 58(304):918–932

Hoff P (2004) Discussion of "Clustering objects on subsets of attributes", by Friedman and Meulman. Journal of the Royal Statistical Society, Series B 66(4):845

Hoff P (2005) Subset clustering of binary sequences, with an application to genomic abnormality data. Biometrics 61(4):1027–1036

Hubert L, Arabie P (1985) Comparing partitions. Journal of Classification 2(1):193–218

Jin C, Zhang Y, Balakrishnan S, Wainwright MJ, Jordan MI (2016) Local maxima in the likelihood of Gaussian mixture models: Structural results and algorithmic consequences. In: Advances in Neural Information Processing Systems, pp 4116–4124

Johnson M, Duvenaud DK, Wiltschko A, Adams RP, Datta SR (2016) Composing graphical models with neural networks for structured representations and fast inference. In: Advances in Neural Information Processing Systems, pp 2946–2954

Kandoth C, McLellan MD, Vandin F, Ye K, Niu B, Lu C, Xie M, Zhang Q, McMichael JF, Wyczalkowski MA, et al (2013) Mutational landscape and significance across 12 major cancer types. Nature 502(7471):333

Kasa SR, Bhattacharya S, Rajan V (2019) Gaussian mixture copulas for high-dimensional clustering and dependency-based subtyping. Bioinformatics

Kass RE, Raftery AE (1995) Bayes factors. Journal of the american statistical association 90(430):773–795

Kim D, Seo B (2014) Assessment of the number of components in Gaussian mixture models in the presence of multiple local maximizers. Journal of Multivariate Analysis 125:100–120

LeCam L (1953) On some asymptotic properties of maximum likelihood estimates and related bayes estimates. Univ California Pub Statist 1:277–330

Lin W, Hubacher N, Khan ME (2018) Variational message passing with structured inference networks. arXiv preprint arXiv:180305589

Marbac M, Sedki M (2017) Variable selection for model-based clustering using the integrated complete-data likelihood. Statistics and Computing 27(4):1049–1063

McLachlan G, Krishnan T (2007) The EM algorithm and extensions, vol 382. John Wiley & Sons

McLachlan GJ, Peel D, Bean R (2003) Modelling high-dimensional data by mixtures of factor analyzers. Computational Statistics & Data Analysis 41(3-4):379–388

McNicholas PD, Murphy TB (2008) Parsimonious Gaussian mixture models. Statistics and Computing 18:285–296

McNicholas PD, Murphy TB, McDaid AF, Frost D (2010) Serial and parallel implementations of model-based clustering via parsimonious Gaussian mixture models. Computational Statistics and Data Analysis 54(3):711–723

Melnykov V, Maitra R, et al (2010) Finite mixture models and model-based clustering. Statistics Surveys 4:80–116

Milnor J (1985) On the concept of attractor. In: The theory of chaotic attractors, Springer, pp 243–264

Pan W, Shen X (2007) Penalized model-based clustering with application to variable selection. Journal of Machine Learning Research 8(May):1145–1164

Park H, Ozeki T (2009) Singularity and slow convergence of the EM algorithm for Gaussian mixtures. Neural Processing letters 29(1):45–59

Robert C (2014) Machine learning, a probabilistic perspective. Taylor & Francis

Robert CP, Casella G (2010) Introducing Monte Carlo methods with R, vol 18. Springer

Salakhutdinov R, Roweis ST, Ghahramani Z (2003) Optimization with EM and expectation-conjugate-gradient. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp 672–679

Städler N, Dondelinger F, Hill SM, Akbani R, Lu Y, Mills GB, Mukherjee S (2017) Molecular heterogeneity at the network level: high-dimensional testing, clustering and a TCGA case study. Bioinformatics 33(18):2890–2896

Takeuchi K (1976) Distribution of informational statistics and a criterion of model fitting. Suri-Kagaku (Mathematical Sciences), Japanese 153(3):12–18

Tarpey T, Yun D, Petkova E (2008) Model misspecification: finite mixture or homogeneous? Statistical Modelling 8(2):199–218

Wald A (1949) Note on the consistency of the maximum likelihood estimate. The Annals of Mathematical Statistics 20(4):595–601

Wang S, Zhu J (2008) Variable selection for model-based high-dimensional clustering and its application to microarray data. Biometrics 64(2):440–448

Weinstein JN, Collisson EA, Mills GB, Shaw KRM, Ozenberger BA, Ellrott K, Shmulevich I, Sander C, Stuart JM, Network CGAR, et al (2013) The cancer genome atlas pan-cancer analysis project. Nature Genetics 45(10):1113

White H (1982) Maximum likelihood estimation of misspecified models. Econometrica: Journal of the Econometric Society pp 1–25

Xu L, Jordan MI (1996) On convergence properties of the EM algorithm for Gaussian mixtures. Neural Computation 8(1):129–151, URL https://doi.org/10.1162/neco.1996.8.1.129

Zarchi M, Bushehri SF, Dehghanizadeh M (2018) Scadi: A standard dataset for self-care problems classification of children with physical and motor disability. International Journal of Medical Informatics

A Singularities in Mixture Models

In this section we begin by describing singularities for a simple univariate case which is qualitatively same as that of multivariate case. A typical univariate mixture model can described as follows:

$$f(x, \boldsymbol{\theta}) = \sum_{i=1}^{K} \pi_i f_i \left(x - \mu_i\right) \tag{3}$$

where the π_i are the mixture component proportions that satisfy the constraint $\sum_i \pi_i = 1$; μ_i's are the location parameters of each component. Consider a two-component univariate Gaussian mixture model with both variance equal to 1:

$$f(x|\pi, \mu_1, \mu_2) = \pi f_1\left(x - \mu_1\right) + (1 - \pi) f_2\left(x - \mu_2\right) \tag{4}$$

Hill (1963) showed that information for π in this normal mixture can be approximated as follows:

$$I(\pi) = -E\frac{\partial^2 \ln f(X)}{\partial^2 \pi} = E\left\{\frac{f_1(X - \mu_1) - f_2(X - \mu_2)}{f(X)}\right\}^2 \tag{5}$$

$$\approx \left(\mu_1 - \mu_2\right)^2 \tag{6}$$

Thus when μ_1 and μ_2 become closer, the information about π reduces. Similarly, when π becomes 0 or 1, then then it is impossible to infer the parameter μ_1 & μ_2. The Fisher information matrix becomes degenerate at these points (Amari et al, 2006; Park and Ozeki, 2009). These points are called **Singularities.** Amari and Park also discuss the effect of these singular points on dynamics of learning and attribute these effects to the non-identifiability of parameters. In our experiments too, specifically in high-dimensional datasets, we notice that these singular points acts as Milnor attractors (Milnor, 1985; Park and Ozeki, 2009; Amari et al, 2006) – specifically, we find that naively running an AD-based inference leads to only one component with a mixing proportion close to 1 (we refer to this component as dominating component) and the remaining components with mixing proportions close to 0. More over, this dominating component changes with initialization of the parameters; which of these initialized components ends up as the dominating one depends on how close the parameters of a component are to one of the Milnor attractor points in the parameter space. The presence of Milnor attractors adversely affects learning in high dimensions.

B Brief Introduction of Automatic Differentiation

We follow the description in Baydin et al (2018). Automatic Differentiation is the workhorse of optimization routines used in today's machine learning. While it is ease to see the difference between Automatic Differentiation (AD) vs Numerical Differentiation (ND) (which is marred by approximation errors), the subtle difference between AD vs Symbolic Differentiation (SDs) is not easy to appreciate.

The key idea is this: In symbolic differentiation, we first naively evaluate the complete expression and then differentiate a complex expression using sum rule, product rule etc. However, it may so happen that we are indeed repeatedly evaluating the same expression multiple times, e.g.,

$$\frac{d}{dx}\left(f(x) + g(x)\right) \rightsquigarrow \frac{d}{dx}f(x) + \frac{d}{dx}g(x) \tag{7}$$

$$\frac{d}{dx}\left(f(x)\, g(x)\right) \rightsquigarrow \left(\frac{d}{dx}f(x)\right) g(x) + f(x)\left(\frac{d}{dx}g(x)\right) \tag{8}$$

In the equation above, consider the multiplication rule and a function $h(x) = f(x)g(x)$. Note that $h(x)$ and $\frac{d}{dx}h(x)$ have common components: $f(x)$ and $g(x)$ and on the right hand side, $f(x)$ and $\frac{d}{dx}f(x)$ appear separately. In symbolic differentiation we plug the derivative of $f(x)$ and thus have nested duplications of any computation that appears in common between $f(x)$ and $\frac{d}{dx}f(x)$. In this manner symbolic differentiation can produce exponentially large symbolic expressions which take correspondingly long to evaluate. This problem is called **expression swell**.

If the symbolic form is not required and only numerical evaluation of derivatives is required, computations can be simplified by storing the values of intermediate subexpressions. Differentiation and simplification steps can be interleaved for further efficiency in computation. This is the underlying principle of AD that applies symbolic differentiation only for the elementary operations and stores intermediate numerical results to efficiently evaluate the main function.

To illustrate the problem, consider the following example: Iterations of the logistic map $l_{n+1} = 4l_n(1 - l_n)$, $l_1 = x$ and the corresponding derivatives of l_n with respect to x, illustrating expression swell. The table 8 clearly shows that the number of repetitive evaluations increase with n.

Table 8: Iterations of the logistic map $l_{n+1} = 4l_n(1 - l_n)$, $l_1 = x$ and the corresponding derivatives of l_n with respect to x, illustrating expression swell - Taken from Baydin et al (2018)

n	l_n	$\frac{d}{dx}l_n$	$\frac{d}{dx}l_n$ (Simplified form)
1	x	1	1
2	$4x(1-x)$	$4(1-x)-4x$	$4-8x$
3	$16x(1-x)(1-2x)^2$	$16(1-x)(1-2x)^2-16x(1-2x)^2-64x(1-x)(1-2x)$	$16(1-10x+24x^2-16x^3)$
4	$64x(1-x)(1-2x)^2(1-8x+8x^2)^2$	$128x(1-x)(-8+16x)(1-2x)^2(1-8x+8x^2)+64(1-x)(1-2x)^2(1-8x+8x^2)^2-64x(1-2x)^2(1-8x+8x^2)^2-256x(1-x)(1-2x)(1-8x+8x^2)^2$	$64(1-42x+504x^2-2640x^3+7040x^4-9984x^5+7168x^6-2048x^7)$

A rudimentary solution to counter expression swell would be: for e.g.: $x(1-x)$ occurs many times in the derivative. So, while computing the $\frac{dl}{dx}$ say at $x=0.5$, we can compute this value of $x(1-x)=0.5*0.5=0.25$ once and use it whenever we encounter. What AD does actually is that rather than first getting the expression of l_{n+1} entirely in terms of x and then differentiating, we can do the following: The derivative of $l_{n+1}=4l_n(1-l_n)$ can be found using the chain rule $\frac{dl_{n+1}}{dl_n}\frac{dl_n}{dl_{n-1}}\dots\frac{dl_1}{dl_x}$ which simplifies to $4(1-l_n-l_n)4(1-l_{n-1}-l_{n-1})\dots4(1-x-x)$.

The thing to note is that evaluating the AD way is computationally linear in n (because we add only one $(1-l_n-l_n)$ for each increase by 1) whereas using the SD way is exponential (as evident in the table above). This linear time complexity is achieved due to carry-over of the derivatives at each step, rather than evaluating the derivative at the end and substituting the value of x. Also note the simplicity of the code for this evaluation:

```
from autograd import grad
def my_func(x,n):
    p = x
    y = x*(1 x)
    for i in range(n):
        y = y*(1 y)

    return y
grad_func = grad(my_func)
grad_func(0.5,4)
```

Also, doing the differentiation using the SD way is suitable only when the function is expressed in a closed mathematical form such as polynomials, trigonometric functions, exponential functions, etc. However, if the function is a computer program with control structures such as `for, if, while`, we cannot use SD, as the function is not in a closed-form. In such cases only AD and ND are suitable.

We use toy problem to motivate the advantages of using AD tools vis$-\grave{a}-$vis SD tools when the expressions to differentiate are complicated. Consider the following recursive expressions: $l_0=\frac{1}{1+e^x}$, $l_1=\frac{1}{1+e^{l_0}}$, ..., $l_n=\frac{1}{1+e^{l_{n-1}}}$

We evaluate the derivative of l_n wrt x and compare the runtime in Mathematica (SD) vs Python (AD) for various values of n. As n increases, it is expected that runtime also increases. However, it can be seen from the results in Table 9 that runtime increases linearly for Python (AD) whereas it increases exponentially for Mathematica (SD).

Table 9: Average runtime (over 1000 runs)

n	Python	Mathematica
1	0.00013	0.00000
5	0.00030	0.00005
10	0.00051	0.00023
50	0.00293	0.00437
100	0.00433	0.15625
200	0.00917	1.45364

C Symbols and Notation

Please refer table 10 for the details of symbols and notation used in the paper.

D Maximum Likelihood Estimation and Model Selection under misspecification

Maximum Likelihood Estimation (MLE) is a widely used frequentist approach for parameter inference. The aim is to find the best estimate of parameters $\hat{\boldsymbol{\theta}}$ from the overall parameter space $\boldsymbol{\Theta}$ for the probability distribution $f(\mathbf{X},\boldsymbol{\theta})$ such that probability of observing the samples $\mathbf{x}_1,\dots,\mathbf{x}_n$ from this distribution is maximized, i.e.,

$$f(\hat{\boldsymbol{\theta}})=\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} f(\boldsymbol{\theta}) \tag{9}$$

When the observed data $\mathbf{x}_1,\dots,\mathbf{x}_n$ are indeed n iid samples from $f(\mathbf{X},\boldsymbol{\theta}^*)$ where $\boldsymbol{\theta}^*$ is the true set of parameters and the model being fit has the same functional form $f(.)$, then model is said to be correctly specified or there is no misspecification. If the model being fit is some other probability distribution g, the model is said to be misspecified.

Estimates from MLE enjoy properties such as consistency and asymptotic normality, with inverse Hessian as the covariance matrix (LeCam, 1953; Wald, 1949). It can be also be shown that, when there is no misspecification, MLE estimate has the best empirical likelihood among all possible estimates in large sample-sizes.

However, when the model is misspecified standard MLE techniques are inconsistent with respect to the true parameter (White, 1982). In misspecified models, maximizing likelihood leads to quasi-Maximum Likelihood estimator (QMLE) with similar consistency and asymptotic normality properties as described earlier but with respect to a different parameter estimate (not the true parameters) (White,

Symbol	Meaning
f	overall probability density from which data is sampled
g	misspecified model for data sampled from f
f_i	probability density of the i-th component
π_i	mixing proportions or weight of the i-th component
$\boldsymbol{\theta}_i$	parameters of i-th mixture component
$\boldsymbol{\theta}$	all parameters of the model
$\boldsymbol{\theta}^*$	true parameters of the model from which data is sampled
$\boldsymbol{\Theta}$	set of all possible parameters i.e. the entire parameter space
K	total of number of components in the model
$\mathbf{X}_i$	i-th random datapoint
$\mathbf{x}_i$	i-th observed datapoint
$\mathbf{X}$	all the n random datapoints
$\mathbf{x}$	all the n observed datapoints
n	Total number of iid datapoints considered
p	Total number of dimensions or features of the data
$\mathcal{L}(\boldsymbol{\theta})$	Likelihood of observing $\mathbf{x}$ if the parameters were $\boldsymbol{\theta}$
$\mathcal{N}$	density of multi-variate Gaussian distribution
$\boldsymbol{\mu}_i, \mu_{ij}$	mean of i-th component of a GMM and its value of along j-th dimension respectively
$\boldsymbol{\Sigma}_i, \Sigma_{ijk}$	Covariance matrix of i-th component of a GMM and its jk-th element respectively
$\hat{a}^t$	Estimate of parameter a at the end of iteration t

Table 10: Symbols used in the paper

1982, Theorem 2.1). If the misspecified model that is being is fit is g whose paramaters are $\boldsymbol{\theta}^g$, then under certain regularity conditions, QMLE $\hat{\boldsymbol{\theta}}^g$ satisfies

$$g(\hat{\boldsymbol{\theta}}^g) = \max_{\boldsymbol{\theta}^g \in \boldsymbol{\Theta}^g} g(\mathbf{x}, \boldsymbol{\theta}^g) \tag{10}$$

where $\boldsymbol{\Theta}^g$ is parameter space of the probability distribution g. Moreover, it has also been shown that $\hat{\boldsymbol{\theta}}^g \rightarrow \boldsymbol{\theta}^{g*}$ as $n \rightarrow \infty$, where $\boldsymbol{\theta}^{g*}$ is the quantity that minimizes the KL-divergence

$$KL(g, f) \equiv \mathbb{E}_f \left(\log \left[g\left(\mathbf{X}, \boldsymbol{\theta}^g\right) / f\left(\mathbf{X}, \boldsymbol{\theta}\right) \right] \right) \tag{11}$$

White suggests several tests for detecting misspecification - which rely on the key assumption that there is a unique parameter vector that maximizes the misspecified likelihood. There are two limitations of White's test:

- Most of these tests are asymptotic in nature and require large sample sizes.
- Assumption A3 of White : "$KL(g, f)$ has a unique minimizer in $\boldsymbol{\Theta}^g$ " – which is used to prove the consistency property of QMLE - is violated in mixture models due to non-identifiability of parameters because all the mixture components can be permutated K! times (Bishop, 2006). Further, in data with near-perfect symmetry, this assumption is violated because more than one component will have the same weight (Robert and Casella, 2010). E.g., consider data sampled from two equally weighted univariate Gaussians with $\mu_1 = 0$ and $\mu_2 = 1$ and both with $\sigma = 1$. Different initializations lead to one of these estimates $(\mu_1, \mu_2) = (0, 1)$ or $(\mu_1, \mu_2) = (1, 0)$. This leads to multiple different parameter estimates with the same likelihood.

Hence, it is not possible to apply these simple tests for detecting misspecification before inference.

Model Selection Predominately there are two types of model selection methods - a) Criterion based methods and b) Testing based methods. Criterion based methods, such as AIC (Akaike Information Criterion) or BIC (Bayesian Information Criterion), typically favor parsimony in modeling (i.e. they have a term that penalizes overfitting in the model) and are easy to implement. It has been pointed out that MLE based criteria such as AIC/BIC are based on asymptotic normality, hence cannot be used for model selection in high-dimensions where $n << p$ (Amari et al, 2006; Akaho and Kappen, 2000; Giraud, 2014). Moreover, in high dimensions, it has been observed that BIC tends to underestimate the number of components whereas AIC overestimates the number of components (Melnykov et al, 2010). Kass and Raftery (1995) suggests, without a theoretical basis, that difference between BIC values should be greater than at least 10 to reliably choose one model over the other. All these drawbacks of criterion based methods make them unsuitable for high-dimensional model selection. Testing based methods, which involve Likelihood Ra-

tio Test (LRT) and bootstrapping, favor sophisticated models but are time-consuming to implement (Feng and McCulloch, 1996). AIC assumes that there is no misspecification which is usually not the case in high dimensions. For misspecified models, a generalization of AIC called Takeuchi Information Criterion (TIC) (Takeuchi, 1976) has been proposed. Let f be the actual density from which $\mathbf{X}$ is sampled. Let g be the misspecified model that is being fit. The TIC is defined as:

$$TIC = 2\mathcal{L}(\hat{\boldsymbol{\theta}}) + 2\,\mathrm{tr}\left(\hat{\mathbf{J}}^{-1}\hat{\mathbf{V}}\right) \tag{12}$$

where, $\hat{\mathbf{J}} = -\frac{1}{n}\sum_{i=1}^{n}\frac{\partial^2 \log g(\mathbf{x}_i,\hat{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}$ and $\hat{\mathbf{V}} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\partial \log g(\mathbf{x}_i,\hat{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}}\frac{\partial \log g(\mathbf{x}_i,\hat{\boldsymbol{\theta}})'}{\partial\boldsymbol{\theta}}\right)$

When there is no misspecification, $\mathbf{J} = \mathbf{V}$ due to the information-matrix equality, implying that $TIC = AIC = 2\mathcal{L}(\hat{\boldsymbol{\theta}}) + 2|\boldsymbol{\theta}|_0$ where $|\boldsymbol{\theta}|_0$ is the zero-norm indicating the number of elements in $\boldsymbol{\theta}$. Nevertheless, in the derivation of TIC, there is an assumption of large sample sizes which makes it unsuitable when $n << p$.

Compared to the literature on mixture models, prior literature on the properties of misspecified mixture models is scant. Tarpey et al (2008) investigates the effects of fitting a GMM using EM on population data where there are no mixtures to begin with. They also introduce a population-based EM for misspecified models where the misspecified population density $g(\mathbf{x}, .)$ needs to be specified apriori. This work is pertinent to the clinical trials and health care community where, artificially fitting clusters/mixtures in sample data may lead to unintended consequences. Dwivedi et al (2018), one of the first works to study the theoretical guarantees of misspecified models, analyze the theoretical behavior of EM algoirithm under mis-specification of the number of components; in particular, they study the bias and convergence rates of EM in univariate mixtures. Another recent work on misspecification of number of clusters in the Bayesian approach is by Cai et al (2017) where the authors show using experimental results that while fitting mixture model with an unknown number of components, there are severe inconsistencies while estimating the number of components using posterior distribution. Our work is complementary to all these previous works in the sense we move away from EM to a general GD based approach for inference and show its superior performance.