caption

University of Ottawa

Master of Science

Representative samples selection in an implicit mixture model with the Approximation Maximization algorithm

Zhibin Liu

300056998
Department of Mathematics and Statistics

Supervised by
Professor Chen Xu
Department of Mathematics and Statistics

April 23, 2020

Acknowledgements

I would like to give my deepest thanks to Professor Chen Xu (Department of Mathematics and Statistics, University of Ottawa) and PhD student Xingxiang Li (Department of Mathematics and Statistics, University of Ottawa) for providing me the opportunity to take part in this project. They have truly enriched my experience at University of Ottawa. It has been a great honour to work under their supervision. Professor Xu regularly organizes group meetings, giving me the opportunity to present our research results to other graduate students, as well as learn from others in similar research domain. In the last few months, he provided insightful suggestions and immediate feedback for this project, which I am much appreciated for. PhD student Li, the leader of this project, provides countless assistance to me throughout the last 2 semesters. He leads me to the path of science research, enhancing my critical thinking and coding skills. Whenever I encounter a roadblock, he would set up an in-person meeting to explain the problem patiently. Even in the current extreme pandemic situation, he still organizes online meetings frequently for us to discuss our latest findings and challenges.

As for my contributions to this project, I am mainly in charge of searching relevant literatures and implementing the numerical experiments based on Li's methodology. To be more specific, I am responsible for section 1, 2, 3, 5 and 6 in this project report.

Contents

1 Abstract

In the era of big data, a massive dataset often includes considerable noise samples, in addition to the representative samples which are of interest. Excessive noise samples will pose great challenges for some existing methods to analyse the representative samples, such as classic maximum likelihood estimation (MLE). One way to dispose this new problem is to select the representative samples before direct analysis. In order to distinguish the representative samples from noise samples in theory, we assume the massive dataset is generated by an implicit mixture model, where the representative samples follow an explicit model of interest but the noise samples are from some implicit models. To analyse the representative samples of the whole dataset, it is significant to accurately estimate the parameters of the explicit model. There are a variety of methods to estimate parameters of mixture models, such as the expectation maximization (EM) algorithm. Nevertheless, EM requires the assumptions of all component models in advance while the noise samples can be too random to be assumed. Therefore, we propose a new method named Approximation Maximization (AM) algorithm to separately estimate the parameters from explicit component model in the implicit mixture model. With the well-estimated parameters, the samples near the estimated explicit model will be selected by AM algorithm. Since AM does not rely on the assumption of implicit models, it can be more widely applied to the implicit mixture model than EM algorithm. The promising performances of our method are supported by extensive numerical examples in this report.

Keywords: representative samples; noise samples; implicit mixture model; parameter estimation.

2 Introduction

In statistical analysis, parameter estimation plays an extremely important role in many methods. Classic parameter estimation approaches are constructed on a premise that all samples are of great quality. For example, many parameter estimation methods often assume that all samples are independently and identically distributed from a same model. However, with rapid development of data generation and acquisition, a dataset with high quantity but low quality are frequently encountered in many scientific fields. These massive datasets often include two kinds of samples, representative samples and noise samples. The former ones come from the model of our interest, but the latter ones are from some models we do not focus on. The considerable amount of noise samples will pose great challenges for classic parameter estimation methods, such as MLE, to conduct on the interested model. If we directly utilize the classic methods on the dataset with many noise samples, it is highly likely that an inaccurate estimate of parameters would be obtained. Unsatisfactory parameter estimation will do harm to the following model interpretation and statistical decision. Therefore, developing an accurate parameter estimation for the representative samples, not affected by noise samples, is crucial in the era of big data.

In order to pursue more accurate parameter estimate, a straightforward way is to select representative samples from the massive dataset. To better distinguish representative samples from the noise samples, we assume the complex dataset is generated from an implicit mixture model in theory, where representative samples follow an explicit model of interest but the noise samples are from some other implicit models. When the model of noise samples can be well assumed, EM algorithm can be utilized to conduct parameter estimation and sample clustering by using posterior probability. However, if the related assumption is mismatching, the performance of EM algorithm may not be satisfactory.

In reality, since noise samples are usually scattered with strong randomness or come from many unknown sources, it is very difficult to give an appropriate assumption on noise samples' model. In this report, we just assume that noise samples are generated by some implicit models without specific forms. Thus, we define our model as implicit mixture model. Although the milder model assump-

tion makes it more widely used, it needs a new parameter estimation method. In this report, we propose a innovative algorithm for this problem by constructing a special approximate function of empirical likelihood. The parameter in the explicit component model can be updated by maximizing a part of approximate function. Thus, this algorithm is named as Approximation Maximization algorithm in this report. The AM algorithm can be applied for the implicit mixture model, since the approximate function only depends on the known explicit model. With the well-estimated parameters for the explicit model, the samples near the estimated model can be selected accurately. In numerical analysis, we demonstrate the promising performance of parameter estimation and representative samples selection in a series of experiments. Before introducing the details of our methodology, I briefly review some related literatures.

3 Literature Review

In this report, I focus on a new topic about selecting representative samples from an implicit mixture model. Before getting too far, it is always a good idea to conduct a sufficient literature review to related topics. The ingenious ideas, broad applications and solid theoretical guarantees in these good literatures will provide a lot of inspirations and experience.

At first, regarding to the general mixture model, EM algorithm proposed by Dempster, Laird, and Rubin 1977 is widely used to estimate its parameters. In addition, based on the estimated parameters, samples can be clustered into some subgroups by using the posterior probability. This means if we can well assume the implicit component models in our topic, EM algorithm also can be used to estimate the parameters and select a subgroup as representative samples set. Based on different mixture models, many variants of EM algorithm have been proposed. Assuming the dataset follows mixture models with same type, Ueda et al. 2000 presented a split-and-merge expectation-maximization (SMEM) algorithm to overcome the local maxima problem in parameter estimation, resulting in classifying samples correctly. Similarly, based on the assumption that the given synthetic dataset followed a finite mixture models, B. Zhang, C. Zhang, and Yi 2004 pre-

sented a competitive EM (CEM), capable of automatically choosing the clustering number and estimating the parameters of the mixture models accurately, so that the representative samples and noise samples can be classified. In general, the related methods need to assume that the dataset follows a finite mixture models with specific forms in advance.

EM algorithm is able to cluster the samples and select a representative subgroup, but it relies on the assumptions on all component models. This inspires me to consider some more straightforward clustering methods. Among all generated clusters, one cluster can be selected as representative samples set, based on which further estimation of the related parameters can be conducted. In this report, I review some literatures about K-means method, one of the most significant approaches for sample clustering. K-means was originally presented by MacQueen 1967. Since that, many related methods have been developed for specific applications. Some new developed methods will be introduced as follows. In unsupervised learning, Ru et al. 2015 proposed a class discovery approach, which utilized the characteristics of the mean-square-error by K-means to estimate the number of classes, then calculated the difference between clustering results and original dataset to determine the real number of classes. Koslicki et al. 2015 proposed Aggregation of Reads by K-means (ARK). This method first used a standard K-means clustering algorithm to partition a large dataset to several subsets, and then processed them to obtain the estimate for each cluster. In general, clustering methods related to K-means can be considered to select representative samples, but it may ignore using the information from the explicit model of interest. If most known information can be used, the selection and estimation performance may further be improved.

In addition to EM algorithm and the clustering methods above, another idea is to select representative samples according to their prediction performance. In machine learning, this idea is close to self-paced learning (SPL) proposed by Kumar, Packer, and Koller 2010, and the representative samples can be regarded as the "easy" samples in SPL. Kumar, Packer, and Koller 2010 mentioned that in the context of a latent variable model, for a given parameter, the easiness of a sample can be defined in two ways: (i) a sample is easy if we are confident about the value

of a hidden variable; or (ii) a sample is easy if it is easy to predict its true output. Kumar, Packer, and Koller 2010 focused on the second definition to select the easy samples and formulated this definition as a concise model through introducing a regularizer into the learning objective. By optimization, easy samples with small prediction error can be selected. Since then, there are many developments of this topic, involving some algorithmic theoretical studies of the SPL algorithm. Meng, Zhao, and Jiang 2017 discovered that the solving strategy on SPL accorded with a majorization minimization algorithm implemented on an implicit objective function. In addition, Ma et al. 2018 further gave the concrete proof the convergence of SPL under some mild conditions. From the perspective of prediction, SPL conducts a different framework for representative samples selection. However, SPL lacks the specific statistical model assumption of the dataset, which may lead to some difficulty for further statistical analysis in theory, including estimation effectiveness and selection accuracy.

In summary, although all the methods above are partly related to our research topic, they are not exactly matched to our study topic from model assumption and interpretation. Therefore, a customized method is very necessary for the representative samples selection.

4 Methodology

4.1 Approximation Maximization Algorithm

To describe the representative samples selection in an implicit mixture model clearly, we introduce some notations. Let Y and Z be two random variables in the sample space $\mathcal{Y}$ and $\mathcal{Z}$, respectively. Suppose we can observe variable Y, while Z is the latent variable. In the classical statistical setting, we have n i.i.d copies $\{y_i, z_i\}_{i=1}^n$ of $\{Y, Z\}$. In this report, it is assumed that each y_i is generated by a two-components implicit mixture model. Let $p(z_i = k) = \pi_k, k = 0, 1$, then the density of y_i can be expressed by

$$f_{\boldsymbol{\theta}^*}(y_i) = \pi_1 f_{\boldsymbol{\theta}_1^*}(y_i|z_i = 1) + \pi_0 f_{\boldsymbol{\theta}_0^*}(y_i|z_i = 0), \tag{1}$$

where $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_0^*)$. Assume that $f_{\boldsymbol{\theta}^*}$ belongs to some parameterized family $\{f_{\boldsymbol{\theta}} | \boldsymbol{\theta} \in \Omega\}$, where Ω is some nonempty convex set of parameters. In (1), the form of pdf $f_{\boldsymbol{\theta}_1^*}(y_i|z_i = 1)$ is known but the true parameter $\boldsymbol{\theta}_1^*$ needs to be estimated, and $f_{\boldsymbol{\theta}_0^*}(y_i|z_i = 0)$ is unknown with a virtual parameter $\boldsymbol{\theta}_0^*$. Under this new problem setup, estimating $\boldsymbol{\theta}_1^*$ and selecting the representative samples from $f_{\boldsymbol{\theta}_1^*}(y_i|z_i = 1)$ are two primary tasks of our research.

Let $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_0)$ denote the whole parameter in $f_{\boldsymbol{\theta}}$. Formally, under the i.i.d assumption, we are interested in computing some $\widehat{\boldsymbol{\theta}}$ by maximizing the log-likelihood function $l_n(\boldsymbol{\theta})$ where

$$l_n(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \log f(y_i) = \frac{1}{n}\sum_{i=1}^{n} \log \left[\int_{\mathcal{Z}} f_{\boldsymbol{\theta}}(y_i, z_i) dz_i \right].$$

The log-likelihood function can be further expressed by

$$\begin{aligned} l_n(\boldsymbol{\theta}) &= \frac{1}{n}\sum_{i=1}^{n} \int_{\mathcal{Z}} q(z_i) \log \frac{f_{\boldsymbol{\theta}}(y_i, z_i) q(z_i)}{f_{\boldsymbol{\theta}}(z_i|y_i) q(z_i)} dz \\ &= \frac{1}{n}\sum_{i=1}^{n} \left(\int_{\mathcal{Z}} q(z_i) \log f_{\boldsymbol{\theta}}(y_i, z_i) dz_i - \int_{\mathcal{Z}} q(z_i) \log q(z_i) dz_i \right. \\ &\qquad \left. + \int_{\mathcal{Z}} q(z_i) \log \frac{q(z_i)}{f_{\boldsymbol{\theta}}(z_i|y_i)} dz_i \right), \end{aligned} \tag{2}$$

where $\int q(z_i) dz_i = 1$. In EM algorithm, $q(z_i = k)$ is set by

$$f_{\boldsymbol{\theta}^{(t)}}(z_i = k|y_i) = \frac{\pi_k f_{\boldsymbol{\theta}_k^{(t)}}(y_i|z_i = k)}{\pi_1 f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1) + \pi_0 f_{\boldsymbol{\theta}_0^{(t)}}(y_i|z_i = 0)}, k = 1, 0. \tag{3}$$

Define

$$Q_n(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \frac{1}{n}\sum_{i=1}^{n} \int_{\mathcal{Z}} f_{\boldsymbol{\theta}^{(t)}}(z_i|y_i) \log f_{\boldsymbol{\theta}}(y_i, z_i) dz_i,$$

and

$$C_n(\boldsymbol{\theta}^{(t)}) = \frac{1}{n}\sum_{i=1}^{n} \int_{\mathcal{Z}} f_{\boldsymbol{\theta}^{(t)}}(z_i|y_i) \log f_{\boldsymbol{\theta}^{(t)}}(z_i|y_i) dz_i.$$

Then, we have

$$l_n(\boldsymbol{\theta}) \geq Q_n(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) - C_n(\boldsymbol{\theta}^{(t)}) \text{ and } l_n(\boldsymbol{\theta}^{(t)}) = Q_n(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) - C_n(\boldsymbol{\theta}^{(t)}).$$

By the idea of Majorization-Maximization (MM), $\boldsymbol{\theta}$ can be updated by

$$\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q_n(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}). \tag{4}$$

The update in (4) can guarantee that $l_n(\boldsymbol{\theta}^{(t+1)}) \geq l_n(\boldsymbol{\theta}^{(t)})$, this important property supports the effectiveness of EM algorithm. Given a good estimate $\widehat{\boldsymbol{\theta}}$ by EM algorithm, we can further cluster samples based on the value of $f_{\widehat{\boldsymbol{\theta}}}(z_i|y_i)$.

In an implicit mixture model, since that the density function $f_{\boldsymbol{\theta}_0}(y_i|z_i = 0)$ in (3) is unknown, we can not compute $f_{\boldsymbol{\theta}^{(t)}}(z_i|y_i)$ directly. EM algorithm may fail to obtain a good estimate when the assumption on $f_{\boldsymbol{\theta}_0}(y|z = 0)$ is incorrect. Therefore in this report, we attempt to construct a new $q(z)$ only relying on $\boldsymbol{\theta}_1^{(t)}$ in $f_{\boldsymbol{\theta}_1^{(t)}}(y|z = 1)$ and satisfying that $\int q(z)dz = 1$. We write the new $q(z) = g_{\boldsymbol{\theta}_1^{(t)}}(z|y)$. When $g_{\boldsymbol{\theta}_1^{(t)}}(z|y)$ is close to $f_{\boldsymbol{\theta}^{(t)}}(z|y)$, we can use the function $G_n(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ to approximate $Q_n(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)})$, where

$$\begin{aligned} G_n(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \frac{1}{n}\sum_{i=1}^{n}\int_{\mathcal{Z}} g_{\boldsymbol{\theta}_1^{(t)}}(z_i|y_i)\log f_{\boldsymbol{\theta}}(y_i, z_i)dz_i \\ &= \frac{1}{n}\sum_{i=1}^{n}\Big\{ g_{\boldsymbol{\theta}_1^{(t)}}(z_i = 1|y_i)\log \pi_1 f_{\boldsymbol{\theta}_1}(y_i|z_i = 1) + \\ &\qquad [1 - g_{\boldsymbol{\theta}_1^{(t)}}(z_i = 1|y_i)]\log \pi_0 f_{\boldsymbol{\theta}_0}(y_i|z_i = 0)\Big\}. \end{aligned}$$

Denote $G_{1,n}(\boldsymbol{\theta}_1|\boldsymbol{\theta}_1^{(t)}) = \frac{1}{n}\sum_{i=1}^{n} g_{\boldsymbol{\theta}_1^{(t)}}(z = 1|y)\log f_{\boldsymbol{\theta}_1}(y|z = 1)$, it is easily seen that $\boldsymbol{\theta}_1$ can be separately updated by

$$\boldsymbol{\theta}_1^{(t+1)} = \underset{\boldsymbol{\theta}_1}{\operatorname{argmax}}\ G_{1,n}(\boldsymbol{\theta}_1|\boldsymbol{\theta}_1^{(t)}). \tag{5}$$

Next, we discuss under what conditions we can construct the $g_{\boldsymbol{\theta}_1^{(t)}}(z_i|y_i)$ approximating $f_{\boldsymbol{\theta}^{(t)}}(z_i|y_i)$. Recall that

$$f_{\boldsymbol{\theta}^{(t)}}(z_i = k|y_i) = \frac{\pi_k f_{\boldsymbol{\theta}_k^{(t)}}(y_i|z_i = k)}{\pi_1 f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1) + \pi_0 f_{\boldsymbol{\theta}_0^{(t)}}(y_i|z_i = 0)}, k = 1, 0.$$

Obviously, for those $\{y_i, z_i\}$ with large values of $|\log \frac{\pi_1 f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i=1)}{\pi_0 f_{\boldsymbol{\theta}_0^{(t)}}(y_i|z_i=0)}|$, their corresponding $f_{\boldsymbol{\theta}^{(t)}}(z_i|y_i)$ can be approximated by

$$g_{\boldsymbol{\theta}^{(t)}}(z_i = 1|y_i) = \begin{cases} 1, \ f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1) \gg \dfrac{\pi_0}{\pi_1} f_{\boldsymbol{\theta}_0^{(t)}}(y_i|z_i = 0), \\ 0, \ f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1) \ll \dfrac{\pi_0}{\pi_1} f_{\boldsymbol{\theta}_0^{(t)}}(y_i|z_i = 0), \end{cases}$$

where $a \gg b$ or $a \ll b$ represent a is much greater or smaller than b. However, even if we know $\frac{\pi_0}{\pi_1}$ in advance, the function $g_{\boldsymbol{\theta}^{(t)}}(z_i|y_i)$ above still relies on $f_{\boldsymbol{\theta}_0^{(t)}}(y_i|z_i = 0)$. More specific conditions are needed to construct $g_{\boldsymbol{\theta}_1^{(t)}}(z_i|y_i)$. If we can assume that there exists a threshold $\gamma_i^{(t)}$ satisfying that $f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1) \gg \gamma_i^{(t)} \gg \frac{\pi_0}{\pi_1} f_{\boldsymbol{\theta}_0^{(t)}}(y_i|z_i = 0)$ or $f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1) \ll \gamma_i^{(t)} \ll \frac{\pi_0}{\pi_1} f_{\boldsymbol{\theta}_0^{(t)}}(y_i|z_i = 0)$ for most $\{y_i, z_i\}$, $g_{\boldsymbol{\theta}^{(t)}}(z_i = 1|y_i)$ can be further simplified by

$$g_{\boldsymbol{\theta}_1^{(t)}}(z_i = 1|y_i) = \begin{cases} 1, \ f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1) \geq \gamma_i^{(t)}, \\ 0, \ f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1) < \gamma_i^{(t)}. \end{cases} \tag{6}$$

When the function $g_{\boldsymbol{\theta}_1^{(t)}}(z_i|y_i)$ in (6) can well approximate $f_{\boldsymbol{\theta}^{(t)}}(z_i|y_i)$ for most $\{y_i, z_i\}$, the difference between $Q_n(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ and $G_n(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ will be small and the update in (5) is believed to be efficient. Given some initialization $\boldsymbol{\theta}_1^{(0)}$, we can alternate the iteration (6) and (5) to obtain the final estimate $\widehat{\boldsymbol{\theta}}_1$. If $\widehat{\boldsymbol{\theta}}_1$ is believed to be close to true parameter $\boldsymbol{\theta}_1^*$ and $g_{\widehat{\boldsymbol{\theta}}_1}(z_i|y_i)$ is a good approximate of $f_{\boldsymbol{\theta}^{(t)}}(z_i|y_i)$, we can select representative samples from $f_{\boldsymbol{\theta}_1}(y|z = 1)$ with $g_{\widehat{\boldsymbol{\theta}}_1}(z_i|y_i) = 1$.

To sum up, the representative samples selection procedure can be summarized as the following AM algorithm.

Algorithm 1: The AM algorithm of representative samples selection

1. Input $\gamma_i^{(0)}$, assume $y_i \sim f_{\boldsymbol{\theta}_1}(y|z=1), i=1,2,...,n$, compute

$$\boldsymbol{\theta}_1^{(0)} = \underset{\boldsymbol{\theta_1}}{\text{argmax}}\ l_n(\boldsymbol{\theta}_1).$$

2. **Approximation** Step: Calculate

$$g_{\boldsymbol{\theta}_1^{(t)}}(z_i|y_i) = \begin{cases} 1,\ f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i=1) \geq \gamma_i^{(t)}, \\ 0,\ f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i=1) < \gamma_i^{(t)}, \end{cases} \quad i=1,2,...,n.$$

3. **Maximization** Step: Update $\boldsymbol{\theta}_1$ by

$$\boldsymbol{\theta}_1^{(t+1)} = \underset{\boldsymbol{\theta}_1}{\text{argmax}}\ G_{1,n}(\boldsymbol{\theta}_1|\boldsymbol{\theta}_1^{(t)}).$$

4. Repeat step 2 and 3 until satisfying $g_{\boldsymbol{\theta}_1^{(t+1)}}(z_i|y_i) = g_{\boldsymbol{\theta}_1^{(t)}}(z_i|y_i), i=1,2,...,n.$

5. Output the final $\boldsymbol{\theta}_1^{(t+1)}$ and $S = \left\{ y_i \ \mid \ g_{\boldsymbol{\theta}_1^{(t+1)}}(z_i|y_i) = 1, i=1,2,...,n. \right\}$.

In some special cases, the maximization step in (5) may be numerically challenging. For example, in linear regression, computing MLE requires the calculation of the inverse of a $p \times p$ matrix, which may require much more computational cost when we are dealing with a high dimensional dataset. So we can consider an alternative method, the first-order AM (FAM) algorithm by taking a gradient step. Given some initialization $\boldsymbol{\theta}_1^{(0)}$ and an appropriate step-size $\alpha > 0$, the first-order AM algorithm performs the update by

$$\boldsymbol{\theta}_1^{(t+1)} \ = \ \boldsymbol{\theta}_1^{(t)} + \alpha \nabla G_{n,1}(\boldsymbol{\theta}_1|\boldsymbol{\theta}_1^{(t)})|_{\boldsymbol{\theta}_1=\boldsymbol{\theta}_1^{(t)}}, \quad t=0,1,2... \tag{7}$$

In order to find an appropriate step-size α, we will apply backtracking method. The details of first-order AM algorithm (FAM) will be given in the following Al-

gorithm 2.

Algorithm 2: The FAM algorithm of representative samples selection

1. Initialize $\boldsymbol{\theta}_1^{(0)}$ as $\mathbf{0}$, assume $y_i \sim f_{\boldsymbol{\theta}_1}(y|z=1), i=1,2,...,n$, set $\gamma_i^{(0)}$, $\alpha_0 > 0$, and give $a, b \in (0,1)$.

2. **Approximation** Step: Calculate

$$g_{\boldsymbol{\theta}_1^{(t)}}(z_i|y_i) = \begin{cases} 1, \ f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i=1) \geq \gamma_i^{(t)}, \\ 0, \ f_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i=1) < \gamma_i^{(t)}, \end{cases} \quad i=1,2,...,n.$$

3. **First-order** Step:

   Set $j=0$, until the following condition is satisfied that

$$G_{j+1} > G_j + a\alpha_j \|\nabla G_{n,1}(\boldsymbol{\theta}_1|\boldsymbol{\theta}_1^{(t)})|_{\boldsymbol{\theta}_1=\boldsymbol{\theta}_1^{(t)}}\|_2^2,$$

   repeatly increase j and set $\alpha_{j+1} = b\alpha_j$.

   Return α_{j+1} as the appropriate step-size and update

$$\boldsymbol{\theta}_1^{(t+1)} = \boldsymbol{\theta}_1^{(t)} + \alpha_{j+1}\nabla G_{n,1}(\boldsymbol{\theta}_1|\boldsymbol{\theta}_1^{(t)})|_{\boldsymbol{\theta}_1=\boldsymbol{\theta}_1^{(t)}}.$$

4. Repeat Step 2, 3 until satisfying $g_{\boldsymbol{\theta}_1^{(t+1)}}(z_i|y_i) = g_{\boldsymbol{\theta}_1^{(t)}}(z_i|y_i), i=1,2,...,n.$

5. Output the final $\boldsymbol{\theta}_1^{(t+1)}$ and $S = \left\{ y_i \ \middle| \ g_{\boldsymbol{\theta}_1^{(t+1)}}(z_i|y_i) = 1, i=1,2,...,n. \right\}$.

4.2 Examples

The proposed representative samples selection procedure based on AM algorithm is suitable for many implicit mixture models. It is noteworthy that when the model of interest is a regression model, the distribution of y_i relies on its corresponding covariate vector $\boldsymbol{x}_i$ and latent variable z_i. By assuming that the distribution of z_i does not depend on $\boldsymbol{x}_i$, we have

$$f_{\boldsymbol{\theta}^*}(y_i|\boldsymbol{x}_i) = \pi_1 f_{\boldsymbol{\theta}_1^*}(y_i|\boldsymbol{x}_i, z_i=1) + \pi_0 f_{\boldsymbol{\theta}_0^*}(y_i|\boldsymbol{x}_i, z_i=0). \tag{8}$$

Obviously, the AM algorithm is still suitable for the representative samples selection based on conditional likelihood $l_{n,x}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\log f(y_i|\boldsymbol{x}_i)$. For ease of writing, we assume $\boldsymbol{x}_i$ is observed and still use $f_{\boldsymbol{\theta}}(y_i|z_i=k)$ to denote $f_{\boldsymbol{\theta}_1}(y_i|\boldsymbol{x}_i, z_i=k)$. The same goes for $f_{\boldsymbol{\theta}^*}(y_i|\boldsymbol{x}_i)$ and $l_{n,x}(\boldsymbol{\theta})$. Then, we denote $w_i^{(t)} = g_{\boldsymbol{\theta}_1^{(t)}}(z_i=1|y_i)$ and give some concrete examples as follows.

1. Implicit Gaussian mixture model

   In this model, $y_i|z_i=1$ follows the normal distribution $\mathcal{N}(\boldsymbol{\theta}_1, \boldsymbol{\Sigma}_i)$, where the mean vector $\boldsymbol{\theta}_1$ is the parameter of interest and $\boldsymbol{\Sigma}_i$ is the known covariance matrix. The update of $\boldsymbol{\theta}_1^{(t)}$ in (5) has the closed form,

   $$\boldsymbol{\theta}_1^{(t+1)} = \left(\sum_{i=1}^{n} w_i^{(t)}\boldsymbol{\Sigma}_i^{-1}\right)^{-1}\left(\sum_{i=1}^{n} w_i^{(t)}\boldsymbol{\Sigma}_i^{-1}y_i\right).$$

2. Implicit linear mixture model

   In this model, $y_i|\boldsymbol{x}_i, z_i=1$ follows the normal distribution $\mathcal{N}(\boldsymbol{x}_i^T\boldsymbol{\theta}_1, \sigma_i^2)$, where the regression coefficient vector $\boldsymbol{\theta}_1$ is the parameter of interest and σ_i^2 is the known variance. The update of $\boldsymbol{\theta}_1^{(t)}$ in (5) has the closed form,

   $$\boldsymbol{\theta}_1^{(t+1)} = \left(\sum_{i=1}^{n} w_i^{(t)}\boldsymbol{x}_i\boldsymbol{x}_i^T/\sigma_i^2\right)^{-1}\left(\sum_{i=1}^{n} w_i^{(t)}\boldsymbol{x}_i y_i/\sigma_i^2\right).$$

3. Implicit logistic mixture model

   In this model, $y_i|\boldsymbol{x}_i, z_i=1$ follows Binomial distribution Binomial$(m_i, \frac{1}{1+\exp(-\boldsymbol{x}_i^T\boldsymbol{\theta}_1)})$. The update of $\boldsymbol{\theta}_1^{(t)}$ in (5) can be updated by

   $$\boldsymbol{\theta}_1^{(t+1)} = \arg\max_{\boldsymbol{\theta}_1}\frac{1}{n}\sum_{i=1}^{n} w_i^{(t)}\left(y_i\boldsymbol{x}_i^T\boldsymbol{\theta}_1 - m_i\log(1+\exp(\boldsymbol{x}_i^T\boldsymbol{\theta}_1))\right). \tag{9}$$

4. Implicit Poisson mixture model

   In this model, $y_i|\boldsymbol{x}_i, z_i=1$ follows Poisson distribution Pois$(\exp(\boldsymbol{x}_i^T\boldsymbol{\theta}_1))$.

The update of $\boldsymbol{\theta}_1^{(t)}$ in (5) can be updated by

$$\boldsymbol{\theta}_1^{(t+1)} = \arg\max_{\boldsymbol{\theta}_1} \frac{1}{n} \sum_{i=1}^{n} w_i^{(t)} \left(y_i \boldsymbol{x}_i^T \boldsymbol{\theta}_1 - \exp(\boldsymbol{x}_i^T \boldsymbol{\theta}_1) \right). \tag{10}$$

4.3 Deviance threshold rule

In this subsection, we discuss the setting of the threshold $\gamma_i^{(t)}$ in AM algorithm. The threshold $\gamma_i^{(t)}$ is very important for representative samples selection because it decides whether y_i can be selected in tth iteration. The most straightforward way is to set $\gamma_i^{(t)}$ as a fixed positive constant for all $i = 1, ..., n$ and $t \geq 0$. However, this setting may be unfair for different samples. We take poisson regression model for example, where $y_i|\boldsymbol{x}_i, z_i = 1 \sim \text{Pois}(\exp(\boldsymbol{x}_i^T \boldsymbol{\theta}_1))$. We find that large mean $\exp(\boldsymbol{x}_i^T \boldsymbol{\theta}_1)$ will lead to small density even if y_i is very close to $\exp(\boldsymbol{x}_i^T \boldsymbol{\theta}_1)$. And small mean $\exp(\boldsymbol{x}_i^T \boldsymbol{\theta}_1)$ will result in a relatively large density for a close y_i. Similar analysis can be used to Gaussian distribution with different $|\boldsymbol{\Sigma}_i|$ and Binomial distribution with different m_i. Thus, for each sample y_i, we need to construct its corresponding threshold to guarantee the fairness. In this report, we propose a deviance threshold rule based on the deviance between y_i and its conditional expectation $E_{\boldsymbol{\theta}_1}(y_i|z_i = 1)$.

At first, we assume that the conditional expectation $E(y_i|z_i = 1)$ depends on $\boldsymbol{\theta}_1$, and it can be further written as $E_{\boldsymbol{\theta}_1}(y_i|z_i = 1)$ for accuracy. Then, for four examples we discussed above, we have that maximizing weighted log-likelihood function in (5) is equivalent to minimizing the weighted deviance between y_i and its conditional expectation $E_{\boldsymbol{\theta}_1}(y_i|z_i = 1)$. To be specific,

$$\begin{aligned} \boldsymbol{\theta}_1^{(t+1)} &= \arg\max_{\boldsymbol{\theta}_1} \frac{1}{n} \sum_{i=1}^{n} g_{\boldsymbol{\theta}_1^{(t)}}(z_i = 1|y_i) \log f_{\boldsymbol{\theta}_1}(y_i|z_i = 1) \\ &= \arg\min_{\boldsymbol{\theta}_1} \frac{1}{n} \sum_{i=1}^{n} g_{\boldsymbol{\theta}_1^{(t)}}(z = 1|y) D(y_i, a_i), \end{aligned} \tag{11}$$

where $a_i = E_{\boldsymbol{\theta}_1}(y_i|z_i = 1)$ and the deviance is defined as

$$D(y_i, a_i) = \begin{cases} (y_i - a_i)^T \boldsymbol{\Sigma}_i^{-1}(y_i - a_i), \; y_i|z_i = 1 \sim \mathcal{N}(a_i, \boldsymbol{\Sigma}_i), \\ 2\left[y_i \log\left(\dfrac{y_i}{a_i}\right) + (m_i - y_i)\log\left(\dfrac{m_i - y_i}{m_i - a_i}\right)\right], \; y_i|z_i = 1 \sim \text{Bin}(m_i, a_i), \\ 2(a_i - y_i) + 2y_i \log(y_i/a_i), \; y_i|z_i = 1 \sim \text{Pois}(a_i). \end{cases}$$

Especially, when $y_i|z_i = 1$ follows one-dimensional normal distribution $\mathcal{N}(a_i, \sigma_i^2)$, the corresponding deviance can be simplified as $D(y_i, a_i) = (y_i - a_i)^2/\sigma_i^2$. Based on (11), the corresponding approximation step like (6) in AM algorithm can be given as follows,

$$g_{\boldsymbol{\theta}_1^{(t)}}(z_i = 1|y_i) = \begin{cases} 1, \; D(y_i, E_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1)) \leq \gamma_{\mathcal{D}}^{(t)}, \\ 0, \; D(y_i, E_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1)) > \gamma_{\mathcal{D}}^{(t)}, \end{cases} \tag{12}$$

where the new threshold $\gamma_{\mathcal{D}}^{(t)}$ is fixed for all $D(y_i, a_i), i = 1, ..., n$ in tth iteration because the deviance is believed to be a fair measure of the difference between y_i and a_i. In practice, $\gamma_{\mathcal{D}}^{(t)}$ can be chosen by our priori information. For ease of implementation, we set $\gamma_{\mathcal{D}}^{(t)}$ as a fixed threshold for all $t \geq 0$.

Given the setting of $\gamma_{\mathcal{D}}^{(t)}$, we bridge the relationship between $\gamma_{\mathcal{D}}^{(t)}$ and $\gamma_i^{(t)}$ in AM. Since that $f_{\boldsymbol{\theta}_1}(y_i|z_i = 1)$ can be expressed as a function of $D(y_i, a_i)$, the $\gamma_i^{(t)}$ in (6) of AM can be set as

$$\gamma_i^{(t)} = \begin{cases} \dfrac{1}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}_i|}} \exp\left(-\gamma_{\mathcal{D}}^{(t)}/2\right), & y_i|z_i = 1 \sim \mathcal{N}(a_i, \boldsymbol{\Sigma}_i), \\ \exp(-\gamma_{\mathcal{D}}^{(t)}/2 + h(m_i, y_i)), & y_i|z_i = 1 \sim \text{Bin}(m_i, a_i), \\ \exp(-\gamma_{\mathcal{D}}^{(t)}/2 - \log(y_i!) - y_i + y_i \log(y_i)), & y_i|z_i = 1 \sim \text{Pois}(a_i), \end{cases}$$

where $h(m_i, y_i) = y_i \log y_i + (m_i - y_i)\log(m_i - y_i) - m_i \log m_i + \log \binom{m_i}{y_i}$. It is obvious that AM algorithm based on $\gamma_i^{(t)}$ above is equivalent to (11) with $\gamma_{\mathcal{D}}^{(t)}$. This means $\gamma_i^{(t)}$ inherits the fairness from the fixed $\gamma_{\mathcal{D}}^{(t)}$. From the equation of $\gamma_i^{(t)}$, we can further find that, in addition to the $\gamma_{\mathcal{D}}^{(t)}$, the threshold $\gamma_i^{(t)}$ may depend on the other known parameters in $f(y_i|z_i = 1)$ or the specific value of y_i. This corresponds to the analysis at the begin of this subsection.

5 Numerical Analysis

5.1 Experiment Settings

In this section, I experiment our AM algorithm under four classes of implicit mixture models, where the explicit models are Gaussian distribution, linear regression, logistic regression and Poisson regression, respectively. The last three models can be summarized as implicit general linear mixture models. Regarding to implicit component models, I design some totally different models. Based on the above four models, I compare the estimating and selecting performance of our AM and FAM with three methods. In the first method, all samples are directly assumed to come from the explicit model, and the parameters of interest are estimated by maximum likelihood estimation. The first method just focuses on parameter estimation and is called "MLE". The second method considers using EM algorithm based on mixture models to conduct estimation and representative samples selection. For ease of implementation, the implicit model is assumed to have same distribution function as the explicit model, but their parameters are different. Among all subgroups after clustering, I choose the subgroup with largest mean likelihood as the representative samples set. This method is denoted as "EM". In the third method, the whole dataset will be clustered into two subsets by K-means, and the parameters will be estimated by MLE under the explicit model assumption. The representative samples set is determined by the same way as "EM". This estimating and selection procedure is referred to as "K-means".

In different models, I need to generate some multiple dimensional variables, such as $\boldsymbol{y}_i = (y_{i1}, y_{i2}, ..., y_{iq})^T$ in implicit multiple Gaussian mixture model or the covariate vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, ..., x_{ip})^T$ in implicit general linear mixture model. In this report, I consider three different covariance structures for $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$. The first structure is that the variables are independent with each other. The second structure allows that variables are auto-regressive correlated, which means the correlation of adjacent variables are higher than that of the distant ones. The final structure is compound symmetry, where the correlation between two variables remains the same. Take $\boldsymbol{x}_i$ as an example, I assume that each $\text{var}(x_{ij}) = 1$ and

covariances are demonstrated as below,

S1: $\text{cov}(x_{ij}, x_{il}) = 0$, for all pairs of j and l;

S2: $\text{cov}(x_{ij}, x_{il}) = 0.5^{|j-l|}$, for all pairs of j and l;

S3: $\text{cov}(x_{ij}, x_{il}) = 0.5$, for all pairs of j and l.

5.2 Implementation Details

The numerical experiments are conducted by software R on a Windows server with 2.6 GHZ CPUs. For the implicit Gaussian mixture model, R funtion "msnFit" of package "fMultivar" is implemented for the "MLE". Besides, the "Mclust" function from the package "mclust" is used for the "EM" method. Regarding to "K-means" method, the "kmeans" function is applied to cluster the data, and then I utilize its accessory function "center" to solve the parameters for both classes. As for the implicit general linear mixture model, "glm" function and "flexmix" function of "flexmix" package is used for "MLE" and "EM" method respectively. The function to implement "K-means" method remains the same as that in implicit Gaussian mixture model. In addition, for the implicit logistic mixture model, I implement "glmnet" function and its corresponding cross validation function "cv.glmnet"of "glmnet" package to add a penalty to the norm of the true parameter $\boldsymbol{\theta}_1$.

5.3 Summary Statistics

The performance of parameter estimation and representative samples selection is evaluated by repeating the simulation for $T = 100$. Let $\boldsymbol{\theta}_1^*$ be the true parameter of the explicit model, and $\hat{\boldsymbol{\theta}}_{1,t}$ denotes the corresponding estimate in the tth simulation. I use the deviation (DEV) of true parameter $\boldsymbol{\theta}_1^*$ to evaluate the estimation performance. To be specific,

$$\text{DEV} = \frac{1}{T}\sum_{t=1}^{T} \frac{\|\hat{\boldsymbol{\theta}}_{1,t} - \boldsymbol{\theta}_1^*\|_2}{\|\boldsymbol{\theta}_1^*\|_2} .$$

DEV is a good criteria to measure the relative difference between $\hat{\boldsymbol{\theta}}_{1,t}$ and $\boldsymbol{\theta}_1^*$ by dividing $\|\boldsymbol{\theta}_1^*\|_2$. In this way, I can fairly assess the estimation performance of $\hat{\boldsymbol{\theta}}_{1,t}$ corresponding to different $\boldsymbol{\theta}_1^*$. In addition, I evaluate the performance of representative samples selection in terms of averaging positive selection rate (PSR) and false discovery rate (FDR). Let $\hat{s}_t$ be the index set of selected representative samples in the tth experiment, and s^* denotes the set of true representative samples in the dataset. The aforementioned two evaluation indexes are calculated as follows,

$$\text{PSR} = \frac{1}{T}\sum_{t=1}^{T} \frac{\|\hat{s}_t \cap s^*\|_0}{\|s^*\|_0} ,$$

$$\text{FDR} = \frac{1}{T}\sum_{t=1}^{T} \frac{\|\hat{s}_t - s^*\|_0}{\|\hat{s}_t\|_0} .$$

A high PSR indicates a great deal of representative samples are selected, and a low FDR indicates few noise samples are chosen. Further to that, the final number (FN) of selected representative samples $\|\hat{s}_t\|_0$ by each algorithm is also given, which is a more intuitive index to show the selection performance, compared to the true value $\|s^*\|_0$ in every experiment. Finally, I report the average computational time (in seconds) for all methods in each experiment.

5.4 Implicit normal mixture model

5.4.1 Two variables case

In order to vividly demonstrate the representative samples selection process by AM algorithm, I design a toy experiment for implicit Gaussian mixture model with two variables. The iteration procedure is visualized in Figure 1. The involved dataset includes 100 representative samples and 100 noise samples, which are demonstrated in red and blue points, respectively. The representative samples are generated from a multivariate normal distribution

$$\boldsymbol{y_i}|z_i = 1 \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) ,$$

where $\boldsymbol{\mu} = (2, 6)^T, \boldsymbol{\Sigma}$ is the identity matrix. As for the setting of noise samples, half of them are expressed as

$$(\boldsymbol{y_i}|z_i = 0) = (y_{0,1}, y_{0,2})^T,$$

where

$$\begin{cases} y_{0,1} = 9cos\alpha + \epsilon, \\ y_{0,2} = 3 + 12sin\alpha + \epsilon, \end{cases}$$

where $\alpha = \frac{2\pi \cdot k}{100}$, $k = 1, 2, ..., 100$, $\epsilon \sim \mathcal{N}(0, 0.2)$. The rest 50 noise samples are created by 2 independent variables from different Uniform distribution

$$(\boldsymbol{y_i}|z_i = 0) = (y'_{0,1}, y'_{0,2})^T,$$

$$y'_{0,1} \sim U(-15, -2), \quad y'_{0,2} \sim U(-5, 5).$$

In this case, the parameter of interest is $\boldsymbol{\theta}_1^* = \boldsymbol{\mu}$. Besides, $\boldsymbol{\Sigma}$ is known, and $\gamma_{\mathcal{D}}^{(t)}$ is set as 10, a fixed threshold for all $t \geq 0$.

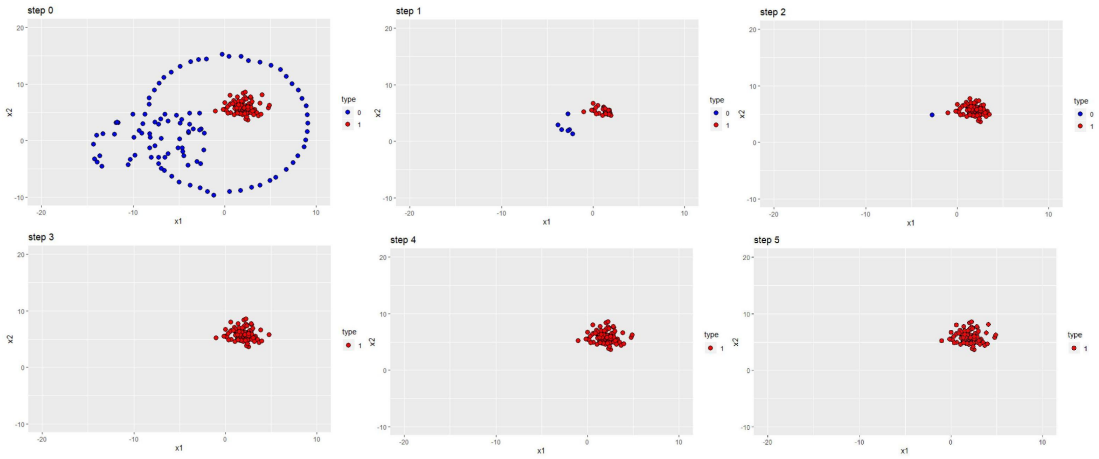

Figure 1: Representative samples selection process under two variables case in implicit Gaussian mixture model.

From Figure 1, it is obvious that more and more red points are selected into representative samples set as iteration grows. On the other side, the blue noise samples are ruled out gradually by our AM algorithm, which means AM algorithm can effectively conduct the representative samples selection. In addition, this result

also demonstrates that by iteration, the estimated parameter $\hat{\boldsymbol{\theta}}_1^{(t)}$ can approach to the true parameter $\boldsymbol{\theta}_1^*$ gradually. Eventually, only the samples near the true model will be selected into representative samples set.

5.4.2 Multiple variables case

In this case, I consider the implicit Gaussian mixture model with multiple variables. In this experiment, 5,000 representative samples and 5,000 noise samples are generated from the mixture model. The explicit model is that

$$\boldsymbol{y}_i|z_i = 1 \sim \mathcal{N}(\boldsymbol{\mu}_1,\, \boldsymbol{\Sigma}_1)\,,$$

where $\boldsymbol{\mu}_1 = (2, 4, 6, 8, 10)^T$, and $\boldsymbol{\Sigma}_1$ has three different structures as mentioned in section 5.1. Half of noise samples $(\boldsymbol{y}_i|z_i = 0) = (y_{0,1}, y_{0,2}, y_{0,3}, y_{0,4}, y_{0,5})^T$ follow a linear structure as

$$y_{0,5} = 1 + y_{0,1} + y_{0,2} + y_{0,3} + y_{0,4},$$

where $(y_{0,1}, y_{0,2}, y_{0,3}, y_{0,4})^T \sim \mathcal{N}(\boldsymbol{\mu}_2,\, \boldsymbol{\Sigma}_2)$. Specifically, $\boldsymbol{\mu}_2 = (2, 3, 4, 5)^T$ and $\boldsymbol{\Sigma}_2$ is set as the same structure with $\boldsymbol{\Sigma}_1$. The rest 2,500 noise samples are generated from 5 independent variables

$$(\boldsymbol{y}_i|z_i = 0) = (y'_{0,1}, y'_{0,2}, y'_{0,3}, y'_{0,4}, y'_{0,5})^T,$$

where

$$y'_{0,j} \sim U(-10, 20), \quad j = 1, 2, 3, 4, 5.$$

Then I apply 5 methods to select the representative samples of the dataset. For AM algorithm, I set $\gamma_{\mathcal{D}}^{(t)}$ as 18, a fixed threshold for all $t \geq 0$. As for FAM, I initialize the α, a, b as 0.002, 0.3, 0.7 respectively. It is also assumed that $\boldsymbol{\Sigma}_1$ is known. Especially, I set the $\gamma_{\mathcal{D}}^{(t)}$ in S1 as 18, and set the ones in S2 and S3 as 28.

The five methods introduced in subsection 5.1 are used to select the representative samples of this dataset, and the results are summarized in Table 1. For all five methods, both AM and FAM perform better than other three methods by estimation accuracy and selection consistency. In comparison, since "MLE" ignores the existence of noise samples, its estimation performance has the largest DEV.

Table 1: Simulation results for implicit Gaussian mixture model with multiple variables

Setup	**Method**	**DEV(%)**	**PSR(%)**	**FDR(%)**	**FN**	**Time**
S1	MLE	31.63	-	-	-	1.02
	EM	13.92	99.96	33.29	7493	5.56
	K-means	14.89	100.00	42.41	8772	0.02
	AM	0.30	99.96	1.15	5056	3.14
	FAM	0.30	99.96	1.15	5056	21.88
S2	MLE	30.76	-	-	-	1.32
	EM	14.36	99.98	33.28	7493	5.61
	K-means	14.83	100.00	42.71	8727	0.01
	AM	0.59	99.96	2.01	5100	3.99
	FAM	1.63	100.00	4.82	5253	59.52
S3	MLE	30.83	-	-	-	1.02
	EM	13.97	99.95	33.29	7494	5.18
	K-means	14.92	100.00	42.71	8727	0.01
	AM	0.60	99.96	2.08	5104	8.31
	FAM	1.11	100.00	2.51	5129	42.86

In addition, as the assumption of the implicit component model in "EM" is far from the true model, the DEV and selection indexes of "EM" are not impressive. This means when model assumption is mismatching, "EM" can not obtain an accurate estimate for $\boldsymbol{\theta}_1^*$, let alone get an accurate posteriori probability for further clustering samples. As for "K-means", it also fails to achieve accurate estimate and representative samples set, which demonstrates that it is difficult to conduct representative samples selection without using any priori information of interested model.

From Table 1, it is noticeable that more complex covariance structure in this experiment does not lead to obvious reduction of accuracy for AM and FAM. By comparing AM with FAM, AM is computationally more efficient than FAM, which is due to a close form of $\boldsymbol{\theta}_1^{(t)}$ in this model. As for FAM, it may need much more iterations to conduct backtracking to determine the step-size before convergence. To demonstrate this procedure clearly, I plot the convergence process of estimated parameters in Figure 2, which shows the tendency of DEV for AM and FAM, respectively. From Figure 2, the estimated parameter in AM converges to a stable

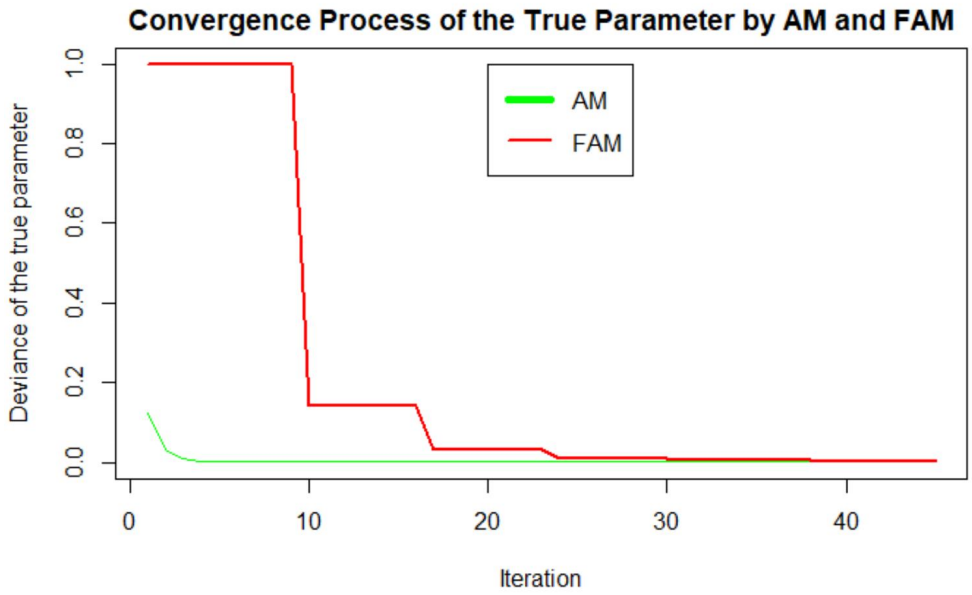

Figure 2: Convergence process of the true parameter by AM and FAM in implicit Gaussian mixture model.

point after around five iterations. However, it takes more than forty iterations to obtain a stable parameter in FAM. In many iterations of FAM, the DEV remains invariant, which means step-size backtracking needs some extra iterations. This also results in more computational cost for FAM. Thus, when it is easy to get the solution in maximization step, I recommend to use AM algorithm.

5.5 Implicit linear mixture model

5.5.1 One variable case

Similar to 5.4.1, I design a toy experiment to visualize the representative samples selection process by AM algorithm for the implicit linear mixture model. The overall dataset includes 100 representative samples and 100 noise samples. The red points in Figure 3 are the representative samples generated from a linear regression

$$y_i|\boldsymbol{x}_i, z_i = 1 \sim \mathcal{N}(\boldsymbol{x}_{1,i}^T\boldsymbol{\theta}_1, \sigma_i^2),$$

where $\boldsymbol{x}_i = (1, x_i)^T$, $x_i \sim U(-30, 30)$, $\boldsymbol{\theta}_1 = (5, 1)^T$, and $\sigma_i^2 = 1$. Half of the noise samples are generated from

$$(y_i|x_i, z_i = 0) = 0.5x_i^2 + 20.$$

The rest 50 noise samples are created from

$$y_i'|x_i, z_i = 0 \sim U(-80, -50).$$

In this case, $\gamma_{\mathcal{D}}^{(t)}$ is set as a fixed value 250. According to the following Figure 3, AM algorithm successfully selects almost all the red representative samples and few blue noise samples after 7 iterations.

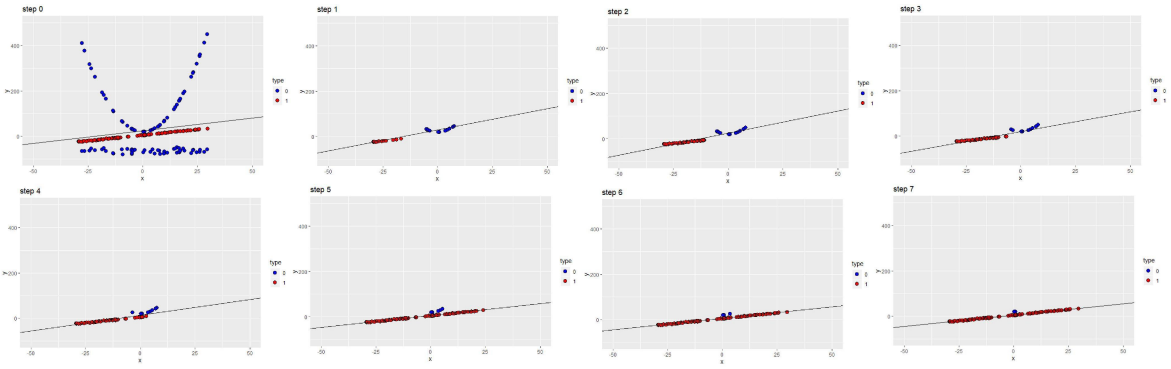

Figure 3: Representative samples selection process with one covariate in implicit linear mixture model.

5.5.2 Multiple variables case

Similar to the setup under normal distribution, the dataset for the multiple variables case consists of 10,000 samples and 5,000 of them are representative samples generated from a multiple linear regression

$$(y_i|\boldsymbol{x}_i, z_i = 1) \sim \mathcal{N}(\boldsymbol{x}_i^T\boldsymbol{\theta}_1, \sigma_i^2),$$

where $\boldsymbol{x}_i = (1, \boldsymbol{x}_i')$, $\boldsymbol{x}_i' = (x_1, ..., x_4)^T \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, $\boldsymbol{\mu}_1 = (2, 4, 6, 8)^T$, $\boldsymbol{\theta}_1 = (5, 4, 3, 2, 1)^T$, $\sigma_i^2 = 1$, $\boldsymbol{\Sigma}_1$ is set as 3 types of covariance matrix mentioned in

subsection 5.1. Half of the rest noise samples are constructed from

$$(y_i|\boldsymbol{x}_i, z_i = 0) = -x_1 + x_2 + x_3^2 + x_4^2,$$

where $\boldsymbol{x}_i = (1, \boldsymbol{x}_i')$ and $\boldsymbol{x}_i' = (x_1, x_2, x_3, x_4)^T$ follows the same distribution as that in explicit model. The rest 2500 noise samples are generated from

$$y_i|\boldsymbol{x}_i, z_i = 0 \sim U(-10, 20).$$

In this implicit linear mixture model, I still compare the performance of 5 approaches to estimate the related parameters and select the representative samples. For AM algorithm, $\gamma_{\mathcal{D}}^{(t)}$ is set as 80, a fixed threshold for all $t \geq 0$. As for FAM, I initialize the $\gamma_{\mathcal{D}}^{(t)}$, α, a and b as 80, 0.0001, 0.4, 0.7 respectively.

Table 2: Simulation results for implicit linear mixture model with multiple variables

Setup	**Method**	**DEV(%)**	**PSR(%)**	**FDR(%)**	**FN**	**Time**
S1	MLE	120.11	-	-	-	0.01
	EM	284.61	100.00	31.44	7289	0.22
	K-means	106.89	100.00	33.97	7572	0.01
	AM	1.88	100.00	0.04	5002	1.74
	FAM	71.12	99.94	0.00	4997	9.94
S2	MLE	101.87	-	-	-	0.01
	EM	125.12	100.00	34.98	7702	0.16
	K-means	197.81	99.92	34.88	7673	0.01
	AM	4.46	100.00	0.28	5014	2.16
	FAM	68.34	100.00	0.14	5007	9.52
S3	MLE	96.26	-	-	-	0.02
	EM	127.15	100.00	35.12	7707	0.24
	K-means	207.07	99.86	34.81	7660	0.01
	AM	3.32	100.00	0.10	5005	2.13
	FAM	74.17	99.96	0.04	5000	9.67

The results are summarized in Table 2. In this experiment, the performance of all 5 methods is similar in 3 setups. AM obviously outperforms other methods in terms of DEV and selection indexes. Due to the similar reasons discussed in

subsection 5.4.2, "MLE", "EM" and "K-means" fail to obtain accurate estimate and representative samples set. As for FAM, although it remains good selection performance in this experiment, it is noteworthy that its estimation accuracy is lower than that in previous experiments. The reason is that by using gradient accent method, FAM tends to converge before obtaining the optimal solution. In other words, the estimate from FAM is not the MLE of selected samples. Compared with other methods except AM, the estimate of FAM is more close to the true parameter $\boldsymbol{\theta}_1^*$. Together with a relatively large deviance threshold in this experiment, FAM also obtain a good selection performance as AM.

In addition to the above experiment settings, I further investigate the impact of representative samples' percentage in the selection process based on the implicit linear mixture model under setup S1. Other settings except the percentage of the representative samples remain the same with the previous experiment.

According to the experiment results in Table 3, AM and FAM still achieve satisfactory selection performance in other 4 setups in terms of near-perfect PSR and relatively low FDR. The performance of "EM" and "K-means" improves as the percentage of representative samples rises. However, even when the percentage is up to 70%, at least 10% of the selected samples turn out to be the original noise samples. Furthermore, another obvious result is that the computation time of AM and FAM decreases as the percentage of representative samples increases. The potential reason is that higher percentage of representative samples contributes to finding a better initial parameter input $\boldsymbol{\theta}_1^{(0)}$. When $\boldsymbol{\theta}_1^{(0)}$ is closer to $\boldsymbol{\theta}_1^*$, fewer iterations will be needed for AM and FAM. In summary, AM is consistently the best approach in this experiment according to a near-perfect PSR, DEV, FDR and reasonable computation time, and its performance further improves when the percentage of the representative samples rises.

5.6 Implicit logistic mixture model

5.6.1 One variable case

As mentioned in subsection 4.2, in the implicit logistic mixture model, $(y_i|\boldsymbol{x}_i, z_i = 1)$ follows a Binomial distribution. In this section, I will consider a special case of Binomial distribution, Bernoulli distribution, which leads to the widespread

Table 3: Simulation results for implicit linear mixture model with various percentage of representative samples.

Per	Method	DEV(%)	PSR(%)	FDR(%)	FN	Time
30%	MLE	176.56	-	-	-	0.01
	EM	605.55	81.98	64.27	11473	0.16
	K-means	77.68	99.98	54.21	10916	0.01
	AM	2.72	100.00	0.16	5008	3.64
	FAM	70.85	99.94	0.00	4997	133.17
40%	MLE	176.57	-	-	-	0.03
	EM	106.37	100.00	41.79	8591	0.07
	K-means	88.87	100.00	43.41	8837	0.01
	AM	1.39	100.00	0.08	5004	1.84
	FAM	70.02	99.94	0.00	4997	12.97
50%	MLE	120.11	-	-	-	0.01
	EM	284.61	100.00	31.44	7289	0.22
	K-means	106.89	100.00	33.97	7572	0.01
	AM	1.88	100.00	0.04	5002	1.74
	FAM	71.12	99.94	0.00	4997	9.94
60%	MLE	196.19	-	-	-	0.01
	EM	276.97	88.54	29.93	6318	0.11
	K-means	330.22	98.14	25.34	6573	0.01
	AM	2.01	100.00	0.04	5002	0.54
	FAM	69.12	99.98	0.05	4999	2.77
70%	MLE	268.72	-	-	-	0.01
	EM	228.42	100.00	10.51	5587	0.10
	K-means	99.35	100.00	18.25	6116	0.01
	AM	2.64	100.00	0.04	5002	0.51
	FAM	69.65	99.98	0.04	5000	2.42

0-1 logistic regression. In the following toy experiment, the complete dataset includes 200 representative samples and 50 noise samples. The distribution of the representative samples is

$$y_i|\boldsymbol{x}_i, z_i = 1 \sim \text{Bernoulli}(\frac{1}{1+\exp(-\boldsymbol{x}_i^T\boldsymbol{\theta}_1)}),$$

where $\boldsymbol{x}_i = (1, x_i')^T$, $x_i' \sim \mathcal{N}(0, 4)$ and $\boldsymbol{\theta}_1 = (1, 3)^T$. As for the noise samples, they are generated from

$$y_i|\boldsymbol{x}_i, z_i = 0 \sim \text{Bernoulli}(\frac{1}{1 + \exp(\boldsymbol{x}_i^T \boldsymbol{\theta}_1)}).$$

The whole model is a logistic mixture model. The sample selection process is presented in Figure 4.

In summary, with a fixed $\gamma_{\mathcal{D}}^{(t)}$ as 2.5, the AM algorithm successfully selects a majority of the representative samples with only few noise samples after 4 iterations. According to Figure 4, it is noticeable that the yellow estimated regression curve is getting steeper, which means the norm of the $\hat{\boldsymbol{\theta}}_1$ becomes larger and larger. It will result in the extreme estimate without any restriction. The solution to deal with this defect is to apply ridge regression to add a penalty to the $\boldsymbol{\theta}_1$. More detailed discussion on this issue will be shown in the following subsection.

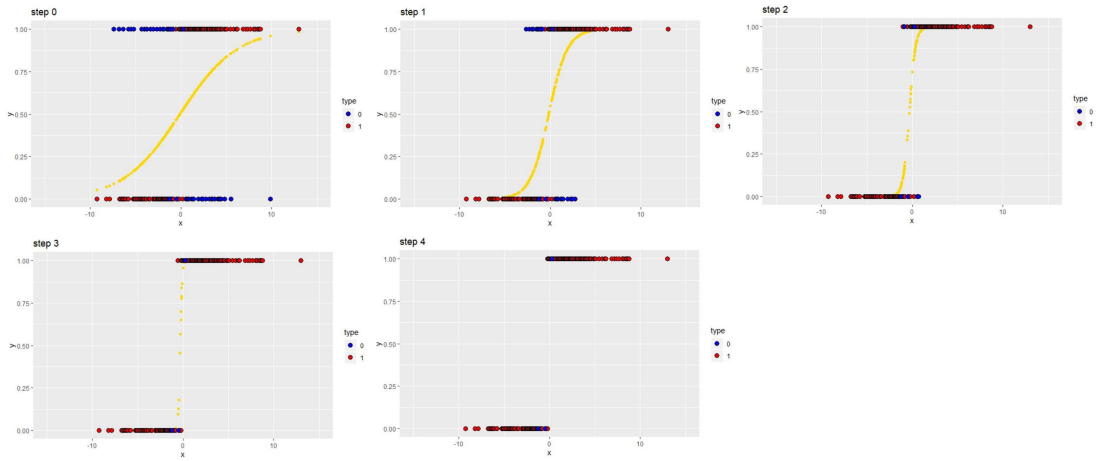

Figure 4: Representative samples selection process with one covariate in implicit logistic mixture model.

5.6.2 Multiple variables case

In this case, I consider the implicit logistic mixture model with multiple variables. In this experiment, the dataset contains 5000 representative samples and

1250 noise samples. The explicit component model is

$$y_i|\boldsymbol{x}_i, z_i = 1 \sim \text{Bernoulli}\left(\frac{1}{1+\exp(-\boldsymbol{x}_i^T\boldsymbol{\theta}_1^*)}\right),$$

where $\boldsymbol{\theta}_1^* = (1, 1, -1, 1, -1)^T$, $\boldsymbol{x}_i = (1, \boldsymbol{x}_i')$ and $\boldsymbol{x}_i' \sim \mathcal{N}(\mathbf{0}, 9 \cdot \boldsymbol{\Sigma}_1)$. $\boldsymbol{\Sigma}_1$ is one of three covariance matrix mentioned in subsection 5.1. As for the noise samples, they are generated from the implicit logistic model

$$y_i|\boldsymbol{x}_i, z_i = 0 \sim \text{Bernoulli}\left(\frac{1}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\theta}_1^*)}\right).$$

The whole model is actually a logistic mixture model. In order to accurately select the representative samples from the explicit model instead of the implicit model, 20% of samples will be set as noise samples in this example. By doing this, it is more likely to obtain a good initial value $\boldsymbol{\theta}_1^{(0)}$ for AM.

Then I compare 5 approaches to select the representative samples of the dataset. Under setup S1, $\gamma_{\mathcal{D}}^{(t)}$ is set as a fixed value 1.5 for AM and FAM. In addition, α, a and b in FAM are initialized as 0.0003, 0.2, 0.9 respectively. As for the setup S2 and S3, the only difference is the $\gamma_{\mathcal{D}}^{(t)}$ in AM and FAM is changed to 2.

The estimation and selection results of five methods are presented in Table 4. To start with, "EM", AM and FAM perform well in selecting representative samples in terms of relatively high PSR and low FDR. From the perspective of DEV, "EM" performs more accurate than AM and FAM. The result of "EM" is different from that in previous models, which is due to that the implicit model assumption of "EM" matches the true model setup. Even so, our AM and FAM still obtain satisfactory results without the assumption on implicit component model. It is noteworthy that, unlike the experiments in Gaussian or linear mixture models, the PSR or FDR in this experiment does not reach the perfect level 1 or 0. This phenomenon is relevant to the logistic model setup only with 0-1 response. In AM, only the samples near $E_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1)$ will be selected into the representative samples set. Because of randomness of the response, there exist a handful of representative samples far from $E_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1)$. In 0-1 logistic model, the definition space of response only have two different values, which means that if a sample is

Table 4: Simulation results for implicit logistic mixture model with multiple variables

Setup	**Method**	**DEV(%)**	**PSR(%)**	**FDR(%)**	**FN**	**Time**
S1	MLE	81.02	-	-	-	0.03
	EM	19.07	93.46	2.91	4813	2.23
	K-means	83.16	50.96	19.57	3168	0.02
	AM	52.42	92.70	2.73	4765	12.39
	FAM	35.42	91.54	3.01	4717	0.44
S2	MLE	73.08	-	-	-	0.03
	EM	29.84	90.03	4.02	4688	2.48
	K-means	73.15	50.52	20.03	3159	0.02
	AM	56.65	88.30	7.52	4765	29.42
	FAM	39.43	90.46	4.28	4726	0.55
S3	MLE	75.06	-	-	-	0.03
	EM	19.72	91.42	4.29	4776	3.61
	K-means	74.15	50.54	19.98	3159	0.01
	AM	44.46	94.27	6.09	5045	18.31
	FAM	43.84	90.96	15.76	4745	0.54

far from $E_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1)$, it will be in the opposite class. When $E_{\boldsymbol{\theta}_1^{(t)}}(y_i|z_i = 1)$ is close to 1 or 0, these true representative samples with "odd" values tend to be ruled out in AM. By the same reason, a handful of noise samples will be selected in representative samples set in 0-1 logistic model. The above discussion explains the reason why the PSR and FDR in this experiment are not close to perfect level. Furthermore, another noticeable outcome is that, unlike experiments in the former 2 implicit mixture models, the computation cost of AM is higer than that of FAM. In fact, in the maximization step of AM, since the ridge regression is applied to add a penalty to the $\boldsymbol{\theta}_1$, it involves the cross validation process to determine the ideal estimated parameter, which can be time-consuming. Therefore, the overall computational time for AM is longer than that in FAM under the implicit logistic mixture model.

5.7 Implicit Poisson mixture model

5.7.1 One variable case

The below experiment is for the implicit Poisson mixture model. This dataset consists of 100 representative samples and 100 noise samples. The representative samples are generated from

$$y_i|\boldsymbol{x}_i, z_i = 1 \sim \text{Pois}(\exp(\boldsymbol{x}_i^T\boldsymbol{\theta}_1)),$$

where $\boldsymbol{x}_i = (1, x_i)^T$, $x_i \sim U(-2, 3)$, $\boldsymbol{\theta}_1 = (3, 3)^T$. The noise samples are generated from

$$y_i|\boldsymbol{x}_i, z_i = 0 \sim U(0, 500).$$

Specifically, I set the above y_i as integer because the response variable of Poisson regression should be an integer. The selection process will be presented in Figure 5. In summary, with $\gamma_{\mathcal{D}}^{(t)}$ as a fixed value 20, most of the representative samples with few noise samples are selected by AM after 5 iterations.

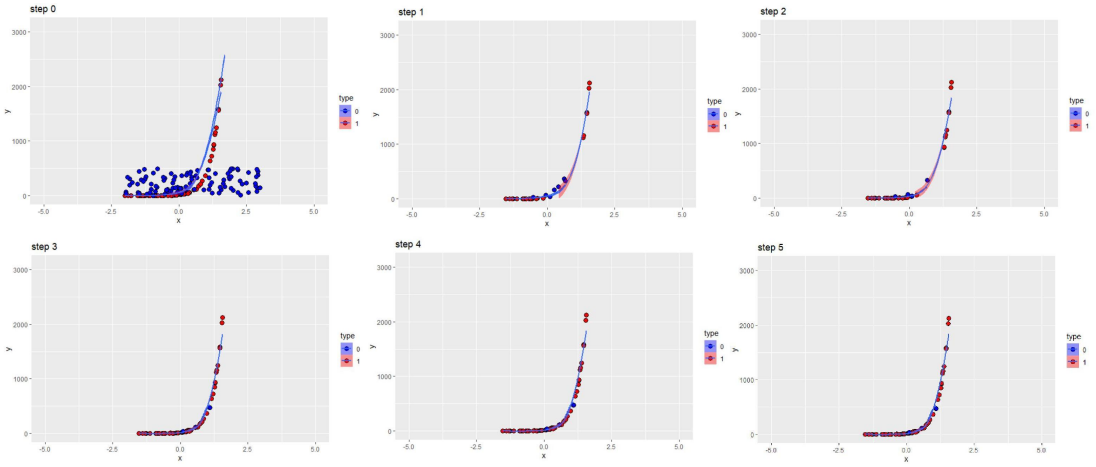

Figure 5: Representative samples selection process with one covariate in implicit Poisson mixture model.

5.7.2 Multiple variables case

In this subsection, I consider the implicit Poisson mixture model with multiple variables. In this experiment, the dataset containing 5,000 representative samples

and 5,000 noise samples is generated from

$$y_i|\boldsymbol{x}_i, z_i = 1 \sim \text{Pois}(\exp(\boldsymbol{x}_i^T\boldsymbol{\theta}_1)),$$

where $\boldsymbol{\theta}_1 = (1, -1, 2, 2, 1)^T$, $\boldsymbol{x}_i = (\mathbf{1}, \boldsymbol{x}_i')$, $\boldsymbol{x}_i' \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ with $\boldsymbol{\mu}_1 = (0, 1, 1, 0)^T$. $\boldsymbol{\Sigma}_1$ is one of three covariance matrix mentioned in subsection 5.1. The noise samples from the implicit model are constructed from

$$y_i|\boldsymbol{x}_i, z_i = 0 \sim U(1, 2000).$$

Then I apply 5 methods in the below table to select the representative samples of the dataset. For AM algorithm, $\gamma_{\mathcal{D}}^{(t)}$ is set as 40, a fixed threshold for all $t \geq 0$. As for FAM, I initialize the $\gamma_{\mathcal{D}}^{(t)}$, α, a, b as 40, 0.03, 0.2 and 0.9 respectively.

Table 5: Simulation results for implicit Poisson mixture model with multiple variables

Setup	**Method**	**DEV(%)**	**PSR(%)**	**FDR(%)**	**FN**	**Time**
S1	MLE	18.87	-	-	-	0.04
	EM	0.01	99.78	15.01	5870	0.71
	K-means	65.90	99.96	50.01	9998	0.02
	AM	0.03	94.78	5.71	5026	0.31
	FAM	1.51	94.26	5.82	5004	14.1
S2	MLE	64.03	-	-	-	0.03
	EM	0.01	99.78	15.21	5884	0.82
	K-means	91.33	99.98	50.00	9999	0.02
	AM	0.01	93.00	5.31	4911	0.39
	FAM	1.41	92.40	5.44	4886	25.54
S3	MLE	10.29	-	-	-	0.05
	EM	12.78	99.32	45.29	9080	2.34
	K-means	43.10	99.94	50.02	9997	0.01
	AM	0.01	92.78	5.28	4844	0.41
	FAM	2.19	91.68	5.31	4895	14.20

Based on implicit Poisson mixture model, Table 5 summarizes the estimation and selection results of five methods. In general, AM and FAM perform better

than other methods when estimation and selection are together considered. It is noteworthy that, by the above data generation methods, some noise samples are close to true regression model, which means a small amount of noise samples will be chosen as representative samples. In order to reduce the amount of noise samples as much as possible, I use a moderate threshold in this example, which also leads to losing some representative samples. This fact can be reflected on the PSR and FDR of AM and FAM. By comparison, "EM" have good parameter estimation and higher PSR, but over 15% of noise samples are selected into representative samples set under S1 and S2 covariance structures. Under S3, "EM" is seriously affected by the strong correlation among covarites, given the fact that it obtains larger DEV as well as FDR, and over 40% of noise samples are selected. On the contrary, AM and FAM are more robust to different covariance structures in this example. As for "MLE" and K-means, both of them have no competitive performance in this experiment.

6 Conclusions

Technological innovations have made a profound impact on knowledge discovery. Extracting useful samples from massive dataset is essential in many modern scientific areas. In this report, we developed an Approximation Maximization algorithm to select representative samples in an implicit mixture model. This approach shows potential to improve the accuracy of estimated parameters of the explicit model, which helps us extract the representative samples out of a complicate dataset. Beside, we further designed a more general first-order AM algorithm. The corresponding estimating and selecting procedure is compatible with a broad range of explicit models.

I implemented experiments in 4 cases, where the explicit models are Gaussian distribution, linear regression, logistics regression and Poisson regression, respectively. Under 3 different correlation structures among variables, the AM algorithm is robust and outperforms other method in terms of estimation accuracy and selection consistency in most experiments.

In our future investigation, we will attempt to study the specific influence of

initial parameter input in our algorithm. Besides, we will design a general and data-driven threshold rule. At last, the theoretical guarantees of AM is another important point in our future work.

References

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). "Maximum Likelihood from Incomplete Data Via the EM Algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1, pp. 1–22.

Ueda, Naonori et al. (2000). "SMEM Algorithm for Mixture Models". eng. In: *Neural Computation* 12.9, pp. 2109–28.

Zhang, Baibo, Changshui Zhang, and Xing Yi (2004). "Competitive EM algorithm for finite mixture models". In: *Pattern Recognition* 37.1, pp. 131–144.

MacQueen, J. (1967). "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, pp. 281–297.

Ru, Xiaohu et al. (Oct. 2015). "Class discovery based on K-means clustering and perturbation analysis". In: pp. 1236–1240.

Koslicki, David et al. (Oct. 2015). "ARK: Aggregation of Reads by K-Means for Estimation of Bacterial Community Composition". In: *PloS one* 10, e0140644.

Kumar, M. P., Benjamin Packer, and Daphne Koller (2010). "Self-Paced Learning for Latent Variable Models". In: *Advances in Neural Information Processing Systems 23*. Ed. by J. D. Lafferty et al., pp. 1189–1197.

Meng, Deyu, Qian Zhao, and Lu Jiang (2017). "A theoretical understanding of self-paced learning". In: *Information Sciences* 414, pp. 319–328.

Ma, Zilu et al. (2018). "On Convergence Properties of Implicit Self-paced Objective". In: *Information Sciences* 462, pp. 132–140.