ROBUST MIXTURE LINEAR EIV REGRESSION MODELS BY T-DISTRIBUTION

by

YANTONG LIU

B.A., QUFU Normal University of Information and Computation Science, China, 2006

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2013

Approved by:

Major Professor
Weixing Song

Copyright

Abstract

A robust estimation procedure for mixture errors-in-variables linear regression models is proposed in the report by assuming the error terms follow a t-distribution. The estimation procedure is implemented by an EM algorithm based on the fact that the t-distribution is a scale mixture of normal distribution and a Gamma distribution. Finite sample performance of the proposed algorithm is evaluated by some extensive simulation studies. Comparison is also made with the MLE procedure under normality assumption.

Key words and phrases: Robust Estimation; Linear Errors-in-Variables Model; EM Algorithm; Mixture; t-distribution

Table of Contents

List of Tables

Acknowledgments

I am thankful to all faculty, staff and students in statistics department. Especially, my sincere gratitude and deep appreciation are given to my major professor, Weixing Song, for his help, guidance and suggestions during the research and writing this report. I also would like to thank Dr. Kun Chen and Dr. Paul I. Nelson for their contributions of time and ideas as my committee members. In the end, thanks all the people who supported me during doing this report.

Chapter 1

Introduction

One of the first major analysis was undertaken by the famous biometrician Karl Pearson involving the use of mixture models over 100 years ago. Pearson (1984) analyzed a data set consisting of measurements on the ratio of forehead to body length of 1000 crabs. Successfully he developed a moment-based mixture model with two-component normal mixture distribution. The results suggested that there were two subspecies present in the mixing crab data. His computational efforts in fitting the data with the model became a daunting prospect to the potential users of this mixture methodology at that time. Since then, the mixture models were widely used to investigate the relationship between variables coming from several unknown latent homogenous groups, providing enormous applications in the areas of astronomy, biology, engineering, genetics, medicine and econometrics. With the further development of finite mixture models to suit the advent of high-speed computer technologies, various attempts have been made to simplify Pearson's model to the fitting of a normal mixture model in the last 60's, in particular with the development of maximum likelihood method by including the presence of multiple maxima in the mixture likelihood function, see Day (1969), Wolfe (1965, 1967) for details. However, it was the publication of the seminal paper of Dempster, Laird, and Rusbin (1977) on the EM algorithm that stimulated unprecedented interest in using finite mixture distribution for modeling heterogeneous data. To date, the fitting mixture models with maximum likelihood estimation using the EM algorithm receives considerable attention and has become one of the most important methodologies in statistics.

The issue of lacking robust regression procedures in errors-in-variables (EIV) models was probably first noticed by Huber (1972) in his Wald lecture. However, the systematic investigation on this topic could be traced back to Carroll and Gallo (1982), and Brown (1982). They pointed out that non-robustness is a very severe disadvantage of the orthogonal estimation method in such models. For functional EIV linear regression model, Brown (1982) demonstrated the instability of the maximum likelihood estimator (MLE) through simulation studies, and identified clearly the source of the instability. Then he generalized the w-estimator of Beaton and Tukey (1974) to the EIV regression model. Zamar (1989) developed an orthogonal regression M-estimates, and showed that these estimations are consistent in elliptical EIV models and robust when the loss function is bounded. Some robust generalized M-estimators were proposed by Cheng and John (1992) for the univariate normal EIV model, which have bounded influence functions in the simple case. In 2004, Fekri and Ruiz-Gazen developed a new class of robust orthogonal weighted regression estimators. The influence functions of the proposed estimators were calculated and shown to be bounded. Moreover, the asymptotic distributions of the estimators were also derived. Other related works in this area were performed by Basu and Sarkar (1997), Li (2002), Paris (2004).

Most of the research conducted in the past including aforementioned work assumed that the data were drawn from a single EIV linear regression model $Y = \alpha + \beta' X + \epsilon, W = X + u$, where α, β are 1-dimensional and $p \geq 1$ dimensional unknown regression coefficients, respectively. ε satisfies $E\varepsilon = 0$ and $E\varepsilon^2 = \sigma^2 < \infty$. The design vector X is not observable, instead, a surrogate W is available. u is the measurement error which is often assumed to have $Eu = 0$ and $\mathrm{Cov}(u) > 0$. If X is random, as described in the structural case, the classical assumptions on X, ε, u are the independence, sometimes, normality. For the sake of model identifiability, some parameters are assumed to be known, such as $\mathrm{Cov}(u)$, or σ^2, or the ratio of $\mathrm{Cov}(u)/\sigma^2$. Most literature mentioned above adopted the later assumption. While in practice, the data may come from different population, that is, a mixture linear

EIV model. To be more specific, let Z be a latent class variable such that given $z = i$, $i = 1, 2, \ldots, g$, the scalar response variable y and a p-dimensional predictor X can be modeled by the following linear EIV structure:

$$Y = \alpha_i + X'\beta_i + \varepsilon_i, \quad W = X + u. \tag{1.1}$$

where ε_i, X and u are mutually independent. $P(z = i) = \pi_i$, $i = 1, 2, \ldots, g$, and z is independent of X, u. The approach of estimating the regression parameters is the central focus of this report.

If there are no measurement errors on X, i.e., $u = 0$, then we can apply the classic mixture linear regression model, in this case, if the random error ε follows normal distribution the traditional MLE works well. However, if there are some outliers or the actual distribution of ε having a heavier tail than the normal, then the MLE based on the normal distribution becomes very unstable. Rich work has been done to robustly estimate the regression parameters under this circumstance. For instance, Markton (2000) and Shen et al. (2004) proposed using a weighting factor for each observation to robust the estimation procedure. Neykov et al. (2007) suggested to use the trimmed likelihood estimator to fit the model. A modified EM algorithm was proposed by Bai et al. (2012) to achieve the robustness. By extending the mixture of t-distributions by Peel and McLachlan (2000) to the regression setting, Yao and Wei (2012) developed a corresponding EM algorithm to estimate the regression parameters robustly. In this report, we will extend Yao and Wei (2012)'s work to the mixture linear EIV regression setting. We will propose the robust estimation procedure for the mixture linear EIV regression model, together with some discussions on the issues related to the implementation of the methodology.

1.1 Mixture Model Definition

In statistics, the mixture model is a probabilistic model to represent the presence of subpopulations within an overall population. Let $y_1, ..., y_n$ be randomly sampled from a g-component population with the density function of the i^{th} component $f_i(y; \lambda_i)$, where $i =$

$1, ..., g$. Let z be a latent class variable such that given $z = i, i = 1, 2, ..., g$. Note that the proportion of j^{th} observation belonging to i^{th} component is π_i, $P(z = i) = \pi_i$, $i = 1, 2, \ldots, g$, where $0 \leq \pi_i \leq 1$ and $\sum_{i=1}^{g} \pi_i = 1$. Now we can calculate the probability density function of Y as follows,

$$f(y, \theta) = \sum_{i=1}^{g} \pi_i f_i(y; \lambda_i), \tag{1.2}$$

where $\theta = (\pi_1, ..., \pi_g, \lambda_1, ..., \lambda_i)$ and λ_i is the parameters vector of the density function $f_i(y; \lambda_i)$.

1.2 Mixture Model: Maximum Likelihood Estimation and EM Algorithm

As we mentioned in the introduction, the maximum likelihood is widely used to estimate the unknown parameter θ. The likelihood function for θ in model (1.2) formed from the observed data y is given by

$$L(\theta, y) = \prod_{j=1}^{n} (\sum_{i=1}^{g} \pi_i f_i(y_j; \lambda_i)). \tag{1.3}$$

It is well known that to maximize the likelihood function is equivalent to maximize log likelihood function. The log likelihood function is given by

$$\log L(\theta, y) = \log(\prod_{j=1}^{n} (\sum_{i=1}^{g} \pi_i f_i(y_j; \lambda_i))) = \sum_{j=1}^{n} \log(\sum_{i=1}^{g} \pi_i f_i(y_j; \lambda_i)). \tag{1.4}$$

The maximum likelihood estimate of θ is described by

$$\hat{\theta} = \arg\max \sum_{j=1}^{n} \log(\sum_{i=1}^{g} \pi_i f_i(y_j; \lambda_i)). \tag{1.5}$$

Then, the estimation of θ can be obtained as a solution of the following equation,

$$\frac{\partial \log L(\theta, y)}{\partial \theta} = 0. \tag{1.6}$$

Obviously, the above equation has no explicit solutions. EM algorithm will play a key role for the estimation of $\hat{\theta}$.

The EM algorithm method, which includes the expectation step (E-step) and the maximization step (M-step), provides an iterative procedure for computing MLE. Denote $z = (z_1, ..., z_n)'$ as the unobservable or missing data vector, where z_j is a g-dimensional vector of zero-one indicator variables and where $z_{ij} = (z_j)_i$ is one or zero according to whether y_j belongs to or does not belong to the i^{th} component.

$$z_{ij} = \begin{cases} 1, & \text{if } j^{\text{th}} \text{ observation is from } i^{\text{th}} \text{ component;} \\ 0, & \text{otherwise.} \end{cases}$$

If we can observe z_{ij} , the complete log likelihood function of θ is

$$\begin{aligned} \log L_c(\theta, y, z) &= \log(\prod_{j=1}^{n}(\prod_{i=1}^{g}[\pi_i f_i(y_j; \lambda_i)]^{z_{ij}})) \\ &= \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij} \log(\pi_i f_i(y_j; \lambda_i)). \end{aligned} \tag{1.7}$$

The EM algorithm for estimating θ can be described as following:

1. **Take initial values $\theta^0 = (\pi_1^0, ..., \pi_g^0, \lambda_1^0, ..., \lambda_i^0)$.**

2. **E-step: At the $(k+1)^{th}$ step:**

$$\begin{aligned} Q(\theta, \theta^{(k)}) &= E[\log L_c(\theta, y, z)|y, \theta^{(k)}] \\ &= E[\sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij} \log(\pi_i f_i(y_j; \lambda_i))|y, \theta^{(k)}] \\ &= \sum_{j=1}^{n}\sum_{i=1}^{g} E[z_{ij}|y, \theta^{(k)}] \log(\pi_i f_i(y_j; \lambda_i)), \end{aligned}$$

Compute this conditional expected value of the log likelihood function, we only need to compute the following equation since $Q(\theta, \theta^{(k)})$ is a linear function of z_{ij} as shown below.

$$E[z_{ij}|y, \theta^{(k)}] = \tau_{ij}^{(k+1)} = \frac{\pi_i^{(k)} f_i(y_j; \lambda_i^{(k)})}{\sum_{i=1}^{g} \pi_i^{(k)} f_i(y_j; \lambda_i^{(k)})}. \tag{1.8}$$

3. **M-step: get the estimate of θ which maximum $Q(\theta, \theta^{(0)})$, which is**

$$\theta^{(k+1)} = \arg\max_{\theta} Q(\theta, \theta^{(k)}). \tag{1.9}$$

4. **Iterate step 2 and step 3 until certain convergence criterion is attained.**

1.3 Mixture Model with Normal Distribution

In the normal mixture model, the density function $f_i(y; \lambda_i)$ is the normal density function with mean μ_i and variance σ_i^2.

$$f_i(y; \lambda_i) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(y-\mu_i)^2}{2\sigma_i^2}}, \tag{1.10}$$

where $\lambda_i = (\mu_i, \sigma_i^2)$.

In this case, if we denote $\tau_{ij}^{(k+1)} = E[z_{ij}|y, \theta^{(k)}]$, then

$$Q(\theta, \theta^{(k)}) = \sum_{j=1}^{n}\sum_{i=1}^{g} \tau_{ij} \log \pi_i - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{g} \tau_{ij} \log 2\pi\sigma_i^2 - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{g} \tau_{ij} \frac{(y_j - \mu_i)^2}{\sigma_i^2}.$$

Let $\partial(Q(\theta, \theta^{(k)})/\partial\theta = 0$, solving this equation, we will have:

$$\begin{aligned}
\pi_i^{(k+1)} &= \sum_{j=1}^{n} \tau_{ij}^{(k+1)}/n, \\
\mu_i^{(k+1)} &= \frac{\sum_{j=1}^{n} \tau_{ij}^{(k+1)} y_j}{\sum_{j=1}^{n} \tau_{ij}^{(k+1)}}, \\
\sigma_i^{2(k+1)} &= \frac{\sum_{j=1}^{n} \tau_{ij}^{(k+1)} (y_j - \mu_i^{(k+1)})^2}{\sum_{j=1}^{n} \tau_{ij}^{(k+1)}}.
\end{aligned}$$

1.4 Mixture of Linear Regression Models with Normal Assumption

Within the family of mixture models, the linear regression model has been studied extensively, especially when the information about membership of the points assigned to each line is not available. A typical data set in this generation came from a tone perception experiment performed by Cohen (1984), where a pure fundamental tone along with electronically generated overtones was played to a trained musician. This experiment was aimed to examine the effect of tuning ratio on the perception of the tone and to determine if either of the two musical perception theories was reasonable. It was found that two lines were evidently corresponding to the behavior indicated by the two musical perception theories, and importantly they depicted correct tuning and tuning to the first overtone. Since then, the model was widely generalized and was found to be useful in various fields of statistical applications such as agriculture, biology, economics, and medicine, generic and marketing, see Susana Faria and Gilda Soromenho (2010) for more details. In practice, the data may come from different population which is a mixture linear regression model. To be specific, let z be a latent class variable such that given $z = i$, $i = 1, 2, \ldots, g$. The scalar response variable y and a p-dimensional predictor X can be modeled by the following linear regression structure:

$$Y = X'\beta_i + \varepsilon_i, \tag{1.11}$$

where $\beta_i = (\beta_{i1}, ..., \beta_{ip})$, ε_i and X are mutually independent. The density of ε_i follows normal distribution which is $f_i(\cdot)$ with mean 0 and variance σ_i^2.

As we mentioned in section 1.3, the EM algorithm at the $(k+1)$ iteration is:

1. **Start with the initial value** $\theta^{(0)} = (\pi_1^{(0)}, ..., \pi_g^{(0)}, \beta_1^{(0)}, ..., \beta_g^{(0)}, \sigma_1^{2(0)}, ..., \sigma_g^{2(0)})$**.**

2. **The E- step: At the** $(k+1)th$ **iteration, computation of the conditional expectation of the complete log likelihood can be simplified as:**

$$E[z_{ij}|y, \theta^{(k)}] = \tau_{ij}^{(k+1)} = \frac{\pi_i^{(k)} f_i(y_j; x_j'\beta_i^{(k)}, \sigma_i^{2(k)})}{\sum_{i=1}^{g} \pi_i^{(k)} f_i(y_j; x_j'\beta_i^{(k)}, \sigma_i^{2(k)})}, \tag{1.12}$$

Where

$$f_i(y_j; x_j'\beta_i^{(k)}, \sigma_i^{2(k)}) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(y_j - x_j'\beta_i^{(k)})^2}{2\sigma_i^{2(k)}}}.$$

3. **M-step: at the** $(k+1)$ **iteration, calculate the estimation of** $\theta^{(k)}$ **which maximizes the expectation of the complete log likelihood function.**

$$
\begin{aligned}
\pi_i^{(k+1)} &= \sum_{j=1}^{n} \tau_{ij}^{(k+1)}/n, \\
\beta_i^{(k+1)} &= \arg\min \sum_{j=1}^{n} \tau_{ij}^{(k+1)} (y_j - x_j'\beta_i)(y_j - x_j'\beta_i)' \\
&= \left(\sum_{j=1}^{n} \tau_{ij}^{(k+1)} x_j x_j' \right)^{-1} \sum_{j=1}^{n} \tau_{ij}^{(k+1)} x_j y_j \\
\sigma_i^{2(k+1)} &= \frac{\sum_{j=1}^{n} \tau_{ij}^{(k+1)} (y_j - x_j'\beta_i^{(k+1)})^2}{\sum_{j=1}^{n} \tau_{ij}^{(k+1)}},
\end{aligned}
$$

If $\sigma_1 = \sigma_2 = ... = \sigma_g = \sigma$, the estimate of σ^2 is:

$$
\sigma_i^{2(k+1)} = \frac{\sum_{j=1}^{n} \tau_{ij}^{(k+1)} (y_j - x_j'\beta_i^{(k+1)})^2}{n}.
$$

4. **Iterate step 2 and step 3 until certain convergence criterion is attained.**

1.5 Mixture Model with t-Distribution

The EM algorithm based on normality assumption is sensitive to outliers or heavy tailed distributions. Peel and McLachlan (2000) proposed using t-distribution in the mixture model to obtain robust estimate. Assume Y follows the t-distribution with the location parameter μ, the scale parameter Σ and the degree of freedom ν. The density function of Y can be written as:

$$
f(y, \theta) = \sum_{i=1}^{g} \pi_i f(y; \mu_i, \Sigma_i, \nu_i), \tag{1.13}
$$

where

$$
f(y; \mu_i, \Sigma_i, \nu_i) = \frac{\Gamma(\frac{\nu_i + p}{2}) \, |\Sigma_i|^{-1/2}}{(\pi_i \nu_i)^{\frac{1}{2}p} \Gamma(\frac{\nu_i}{2}) \left\{1 + \delta(y, \mu_i; \Sigma_i)/\nu_i\right\}^{\frac{1}{2}(\nu_i + p)}},
$$

$$
\delta(y_j, \mu_i; \Sigma_i) = (y_j - \mu_i)^T \Sigma_i^{-1} (y_j - \mu_i),
$$

$$
\theta = (\pi_1, ..., \pi_g, \mu_1, ..., \mu_g, \Sigma_1, ..., \Sigma_g, \nu_1, ..., \nu_g).
$$

The EM algorithm developed for this set up is based on the fact that t-distribution can be written as a scale mixture of normal distribution. Let u be a latent variable such that

$$y|u \sim N(\mu, \Sigma/u), \quad u \sim \text{Gamma}(\upsilon/2, \upsilon/2), \quad \upsilon > 0,$$

where the normal density function $N(\mu, \Sigma/u)$ with mean μ and covariance Σ/u is shown by

$$\phi(y; \mu, \Sigma/u) = \frac{1}{(2\pi)^{(p/2)} |\Sigma/u|^{1/2}} \exp(-\frac{1}{2}(y-\mu)^T(\Sigma/u)^{-1}(y-\mu)), \tag{1.14}$$

and the gamma density function $gamma(\frac{\upsilon}{2}, \frac{\upsilon}{2})$ with shape parameter $\frac{\upsilon}{2}$ and scale parameter $\frac{\upsilon}{2}$ is represented by the following equation:

$$f(u; \frac{\upsilon}{2}, \frac{\upsilon}{2}) = \frac{1}{\Gamma(\frac{\upsilon}{2})(\frac{\upsilon}{2})^{-\frac{\upsilon}{2}}} y^{(\frac{\upsilon}{2}-1)} e^{-\frac{2u}{\upsilon}}. \tag{1.15}$$

Then, the marginal distribution of y has the t-distribution with degree of freedom υ, location parameter μ and scale parameter Σ. If υ is available, the complete log likelihood function can be written as

$$\begin{aligned} \log L_c(\theta; y; z; u) &= \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij} \log[\pi_i \phi(y_j; \mu_i, \Sigma_i/u_j) f(u_j; \frac{\upsilon_i}{2}, \frac{\upsilon_i}{2})] \\ &= \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij} \log \pi_i + \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij} \log[\phi(y_j; \mu_i, \Sigma_i/u_j)] \\ &+ \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij} \log[f(u_j; \frac{\upsilon_i}{2}, \frac{\upsilon_i}{2})], \end{aligned} \tag{1.16}$$

where

$$\begin{aligned} &\sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij} \log[\phi(y_j; \mu_i, \sigma_i/u_j)] \\ &= \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}[-\frac{p}{2}\log(2\pi) - \frac{1}{2}|\Sigma_i/u_j| - \frac{1}{2}u_j(y_j-\mu_i)^T(\Sigma_i/u_j)^{-1}(y_j-\mu_i)], \end{aligned}$$

and

$$\sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij} \log[f(u_j; \frac{\upsilon_i}{2}, \frac{\upsilon_i}{2})] = \sum_{j=1}^{n}\sum_{i=1}^{g} z_{ij}[-\log\Gamma(\frac{\nu_i}{2}) - \frac{\nu_i}{2}\log(\frac{\nu_i}{2}) + \frac{\nu_i}{2}(\log u_j - u_j) - \log u_j].$$

Thus, the EM algorithm can be obtained by the following procedures:

1. **Start with the initial value**

$$\theta^{(0)} = (\pi_1^{(0)}, ..., \pi_g^{(0)}, \mu_1^{(0)}, ..., \mu_g^{(0)}, \Sigma_1^{(0)}, ..., \Sigma_g^{(0)}, \nu_1^{(0)}, ..., \nu_g^{(0)}).$$

2. **The E- step: At the $(k+1)th$ iteration, we can compute the conditional expected value of the log likelihood function, $E(logL_c(\theta; y, z, u)|y, \theta_{(k)})$. We only need to compute the conditional expected value of $E(z_{ij}|y, \theta^{(k)})$, $E(u_j|y, z_{ij} = 1, \theta^{(k)})$, and $E(log(u_j)|y, z_{ij} = 1, \theta^{(k)})$. Some calculations show that**

$$E[z_{ij}|y, \theta^{(k)}] = \tau_{ij}^{(k+1)} = \frac{\pi_i^{(k)} f_i(y_j; \mu_i^{(k)}, \Sigma_i^{(k)}, \nu_i^{(k)})}{\sum_{i=1}^{g} \pi_i^{(k)} f_i(y_j; \mu_i^{(k)}, \Sigma_i^{(k)}, \nu_i^{(k)})}, \tag{1.17}$$

where

$$f_i(y_j; \mu_i^{(k)}, \Sigma_i^{(k)}, \nu_i^{(k)}) = \frac{\Gamma(\frac{\nu_i^{(k)}+p}{2}) \left|\Sigma_i^{(k)}\right|^{-1/2}}{(\pi_i^{(k)} \nu_i^{(k)})^{\frac{1}{2}p} \Gamma(\frac{\nu_i^{(k)}}{2}) \left\{1 + \delta(y, \mu_i^{(k)}; \Sigma_i^{(k)})/\nu_i^{(k)}\right\}^{\frac{1}{2}(\nu_i^{(k)}+p)}},$$

and

$$\delta(y_j, \mu_i^{(k)}; \Sigma_i^{(k)}) = (y_j - \mu_i^{(k)})^T (\Sigma_i^{(k)})^{-1} (y_j - \mu_i^{(k)}).$$

$$E(u_j|y_j, z_{ij} = 1, \theta^{(k)}) = u_{ij}^{(\kappa+1)} = \frac{\nu_i^{(k)} + p}{\nu_i^{(k)} + \delta(y_j, \mu_i^{(k)}; \Sigma_i^{(k)})}. \tag{1.18}$$

$$E(\log u_j|y, z_{ij} = 1, \theta^{(k)}) = \log u_{ij}^{(k+1)} + \left\{\psi(\frac{\nu_i^{(k)} + p}{2}) - \log(\frac{\nu_i^{(k)} + p}{2})\right\}, \tag{1.19}$$

where

$$\psi(\frac{\nu_i^{(k)} + p}{2}) = \frac{\partial \Gamma(\frac{\nu_i^{(k)}+p}{2})}{\partial(\frac{\nu_i^{(k)}+p}{2})} \Big/ \Gamma(\frac{\nu_i^{(k)} + p}{2}).$$

3. **M-step: at the $(k+1)$ iteration, get the estimate of $\theta^{(k)}$ which maximize the expectation of the complete log likelihood function.**

$$\begin{aligned}
\pi_i^{(k+1)} &= \sum_{j=1}^{n} \tau_{ij}^{(k+1)}/n, \\
\mu_i^{(k+1)} &= \frac{\sum_{j=1}^{n} \tau_{ij}^{(k+1)} u_{ij}^{(k+1)} y_j}{\sum_{j=1}^{n} \tau_{ij}^{(k+1)} u_{ij}^{(k+1)}}, \\
\Sigma_i^{(k+1)} &= \frac{\sum_{j=1}^{n} \tau_{ij}^{(k+1)} u_{ij}^{(k+1)} (y_j - \mu_i^{(k+1)})(y_j - \mu_i^{(k+1)})^T}{\sum_{j=1}^{n} \tau_{ij}^{(k+1)}},
\end{aligned}$$

When $\Sigma_1 = \Sigma_2 = ... = \Sigma_g = \Sigma$, the Σ can be written as:

$$\Sigma^{(k+1)} = \frac{\sum_{i=1}^{g} \sum_{j=1}^{n} \tau_{ij}^{(k+1)} u_{ij}^{(k+1)} (y_j - \mu_i^{(k+1)})(y_j - \mu_i^{(k+1)})^T}{n}. \tag{1.20}$$

4. **Iterate step 2 and step 3 until certain convergence criterion is attained.**

1.6 Mixture of Linear Regression Model with t-Distribution

By extending the mixture of t-distributions by Peel and McLachlan (2000) to the regression setting, Yao and Wei (2012) developed a corresponding EM algorithm to estimate the regression parameters robustly. In model (1.11), let ε follow a t-distribution with location parameter 0, scale parameter σ_i and the degree of freedom ν_i. The density function of y_j can be written as:

$$f(y_j; x_j, \theta) = \sum_{i=1}^{g} \pi_i f(y_j; x_j'\beta_i, \sigma_i^2, \nu_i),$$

where

$$f(y_j; x_j'\beta_i, , \sigma_i^2, \nu_i) = \frac{\Gamma(\frac{\nu_i+1}{2}) \, |\sigma_i^2|^{-1/2}}{(\pi_i \nu_i)^{\frac{1}{2}} \Gamma(\frac{\nu_i}{2}) \left\{1 + \delta(y_j; x_j'\beta_i, \sigma_i^2)/\nu_i\right\}^{\frac{1}{2}(\nu_i+1)}},$$

and

$$\delta(y_j; x_j'\beta_i, \sigma_i^2) = (y_j - x_j'\beta_i)^2/\sigma_i^2.$$

To estimate β_i, σ_i^2, the EM algorithm can be developed in the following:

1. **Start with the initial value**

$$\theta^{(0)} = (\pi_1^{(0)}, ..., \pi_g^{(0)}, \beta_1^{(0)}, ..., \beta_g^{(0)}, \sigma_1^{2(0)}, ..., \sigma_g^{2(0)}, \nu_1^{(0)}, ..., \nu_g^{(0)})$$

2. **The E- step: At the $(k+1)th$ iteration, we can compute the conditional expected value of the log likelihood function,$E(\log L_c(\theta; x, y; z; u)|x, y, \theta_{(k)})$. we only need to compute the conditional expected value of $E(z_{ij}|x, y, \theta^{(k)})$, $E(u_j|x, y, z_{ij} = 1, \theta^{(k)})$. Some calculations show that**

$$E[z_{ij}|x, y, \theta^{(k)}] = \tau_{ij}^{(k+1)} = \frac{\pi_i^{(k)} f_i(y_j; x_j'\beta_i^{(k)}, \sigma_i^{2(k)}, \nu_i^{(k)})}{\sum_{i=1}^{g} \pi_i^{(k)} f_i(y_j; x_j'\beta_i^{(k)}, \sigma_i^{2(k)}, \nu_i^{(k)})},$$

where

$$f(y_j; x_j'\beta_i^{(k)}, , \sigma_i^{2(k)}, \nu_i^{(k)}) = \frac{\Gamma(\frac{\nu_i^{(k)}+1}{2}) \left|\sigma_i^{2(k)}\right|^{-1/2}}{(\pi_i \nu_i^{(k)})^{\frac{1}{2}} \Gamma(\frac{\nu_i^{(k)}}{2}) \left\{1 + \delta(y_j; x_j'\beta_i^{(k)}, \sigma_i^{2(k)})/\nu_i^{(k)}\right\}^{\frac{1}{2}(\nu_i^{(k)}+1)}}.$$

$$E(u_j|x, y_j, z_{ij} = 1, \theta^{(k)}) = u_{ij}^{(\kappa+1)} = \frac{\nu_i^{(k)} + 1}{\nu_i^{(k)} + \delta(y_j, \mu_i^{(k)}; \Sigma_i^{(k)})},$$

where

$$\delta(y_j; x_j'\beta_i^{(k)}, \sigma_i^{2(k)}) = (y_j - x_j'\beta_i^{(k)})^2/\sigma_i^{2(k)}. \tag{1.21}$$

3. **M-step: at the $(k+1)$ iteration, get the estimation of $\theta^{(k)}$ which maximize the expectation of complete log likelihood function.**

$$\begin{aligned}
\pi_i^{(k+1)} &= \sum_{j=1}^{n} \tau_{ij}^{(k+1)}/n, \\
\beta_i^{(k+1)} &= \left(\sum_{j=1}^{n} x_j x_j' \omega_{ij}^{(k+1)}\right)^{-1} \sum_{j=1}^{n} x_j y_j \omega_{ij}^{(k+1)}, \\
\sigma_i^{2(k+1)} &= \frac{\sum_{j=1}^{n} \tau_{ij}^{(k+1)} u_{ij}^{(k+1)} (y_j - x_j'\beta_i^{(k+1)})^2}{\sum_{j=1}^{n} \tau_{ij}^{(k+1)}},
\end{aligned}$$

where

$$\omega_{ij}^{(k+1)} = \tau_{ij}^{(k+1)} u_{ij}^{(k+1)}.$$

When $\sigma_1^2 = \sigma_2^2 = ... = \sigma_g^2 = \sigma^2$, the σ^2 can be written as:

$$\sigma^{2(k+1)} = \frac{\sum_{i=1}^{g} \sum_{j=1}^{n} \tau_{ij}^{(k+1)} u_{ij}^{(k+1)} (y_j - x_j' \beta_i^{(k+1)})^2}{n}.$$

4. Iterate step 2 and step 3 until certain convergence criterion is attained.

In the next chapter, we will try to extend Yao and Wei's work to the mixture linear EIV regression setting. We will propose the robust estimation procedure for the mixture linear EIV regression model.

Chapter 2

The robust EM Algorithm for Mixture EIV Linear Regression

2.1 Mixture EIV Linear Regression Model

In practice, the data may come from different population, that is, a mixture linear EIV model. To be more specific, let z be a latent class variable such that given $z = i$, $i = 1, 2, \dots, g$, the scalar response variable y and a p-dimensional predictor X can be modeled by the following linear EIV structure:

$$Y = \alpha_i + X'\beta_i + \varepsilon_i, \quad W = X + u. \tag{2.1}$$

where ε_i, X and u are mutually independent. $P(z = i) = \pi_i$, $i = 1, 2, \dots, g$, and z is independent of X, u. The random components in EIV regression literature are often assumed to be normally distributed. For each i we can assume that

$$\varepsilon_i \sim N(0, \sigma_i^2), \quad X \sim N(\mu, \Sigma), \quad u \sim N(0, \Omega). \tag{2.2}$$

To begin with, we assume that μ, Ω, Σ are all known. The mixture EIV linear regression model can also be written as

$$Y = \alpha_i + (W - u)'\beta_i + \varepsilon_i = \alpha_i + W\beta_i' + \varepsilon_i - u'\beta_i.$$

Therefore we can obtain the MLE of β_i, σ_i^2 from either the marginal distribution of $\xi_i = \varepsilon_i - u'\beta_i$ or the conditional distribution of $\xi_i = \varepsilon_i - u'\beta_i$ given W. With the assumption from

([2.2](#)), the marginal distribution of ξ_i is $N(0, \sigma_i^2 + \beta_i'\Omega\beta_i)$, and the conditional distribution of ξ_i given W is $N(-\mu_{u|W}'\beta_i, \sigma_i^2 + \beta_i'\Lambda\beta_i)$, where

$$\mu_{u|W} = \Omega(\Omega+\Sigma)^{-1}(W-\mu), \quad \Lambda = \Omega - \Omega(\Omega+\Sigma)^{-1}\Omega,$$

which is obtained by the joint distribution of u and W

$$\begin{pmatrix} u \\ W \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ \mu \end{pmatrix}, \begin{pmatrix} \Omega & \Omega \\ \Omega & \Omega+\Sigma \end{pmatrix}\right],$$

Suppose $P(z = i) = \pi_i$, $i = 1, 2, \ldots, g$, z is independent of X. Then the conditional distribution of Y given X is:

$$f(y|x,\theta) = \sum_{i=1}^{g} \pi_i f_i(y; \alpha_i, \beta_i, \sigma_i^2) \tag{2.3}$$

where $\theta = (\pi_1, ..., \pi_g, \beta_1, ..., \beta_g, \sigma_1^2, ..., \sigma_g^2, \alpha_1, ..., \alpha_g)'$ and $f(y; \alpha_i, \beta_i, \sigma_i^2)$ is the density function of y. The log likelihood function is:

$$\sum_{j=1}^{n} \log(\sum_{i=1}^{g} \pi_i f_i(y_j; \alpha_i, \beta_i, \sigma_i^2)) \tag{2.4}$$

2.2 An EM Algorithm for Mixture EIV Linear Regression with Normal Distribution

In the first case, ξ_i marginally follows a normal-distribution with mean 0 and variance $\delta_i = \sqrt{\sigma_i^2 + \beta_i'\Omega\beta_i}$. The maximum likelihood estimate of $\theta = (\pi_1, ..., \pi_g, \beta_1, ..., \beta_g, \sigma_1^2, ..., \sigma_g^2,$ $\alpha_1, ..., \alpha_g)'$ based on the marginal distribution is given by

$$\hat{\theta} = \arg\max_{\theta} \sum_{j=1}^{n} \log(\sum_{i=1}^{g} \pi_i f_i(y_j; \alpha_i, \beta_i, \sigma_i^2)), \tag{2.5}$$

where

$$f_i(y; \alpha_i, \beta_i, \sigma_i^2) = \frac{1}{\sqrt{2\pi\delta_i^2}} e^{-\frac{[y_j - (\alpha_i + W_j'\beta_i)]^2}{2\delta_i^2}}.$$

In the second case, the given W_j, the conditional distribution of ξ_i is a normal-distribution with mean $-\mu_{u|W_j}'\beta_i$ and variance $\delta_i = \sqrt{\sigma_i^2 + \beta_i'\Lambda\beta_i}$. Let $\eta_{ji} = \alpha_i + (W_j - \mu_{u|W_j}')\beta_i$.

The maximum likelihood estimate of $\theta = (\pi_1, \alpha_1, \beta_1', \sigma_1^2, \ldots, \pi_g, \alpha_g, \beta_g', \sigma_g^2)'$, based on the conditional distribution of ξ_i given W_j, is calculated by (2.5), but with

$$f_i(y; \alpha_i, \beta_i, \sigma_i^2) = \frac{1}{\sqrt{2\pi\delta_i^2}} e^{-\frac{(y_j - \eta_{ji})^2}{2\delta_i^2}}.$$

Neither of above two cases provide explicit solutions. Therefore, the EM algorithm method is needed. Because of the similarity of above two cases, we only take the conditional case as an example. The complete log likelihood function with latent variable z_{ij} is:

$$\begin{aligned} \log L_c(\theta, y, z) &= \sum_{j=1}^{n} \sum_{i=1}^{g} z_{ij} \log(\pi_i f_i(y; \alpha_i, \beta_i, \sigma_i^2)) \\ &= \sum_{j=1}^{n} \sum_{i=1}^{g} z_{ij} \log(\pi_i \frac{1}{\sqrt{2\pi\delta_i^2}} e^{-\frac{(y_j - \eta_{ji})^2}{2\delta_i^2}}). \end{aligned}$$

The EM algorithm is as following:

1. **Start with the initial value**

$$\theta^{(0)} = (\pi_1^{(0)}, ..., \pi_g^{(0)}, \alpha_1^{(0)}, ..., \alpha_g^{(0)}, \beta_1^{(0)}, ..., \beta_g^{(0)}, \sigma_1^{2(0)}, ..., \sigma_g^{2(0)})$$

2. **The E-step: At the $(k+1)th$ iteration, compute the conditional expected value of the log likelihood function, $E(\log L_c(\theta; y, z)|y, \theta^{(k)})$, we just need to compute the conditional expected value of $E(z_{ij}|y, \theta^{(k)})$.**

$$E[z_{ij}|y, \theta^{(k)}] = \tau_{ij}^{(k+1)} = \frac{\pi_i^{(k)} f_i(y; \alpha_i, \beta_i, \sigma_i^2)}{\sum_{i=1}^{g} \pi_i^{(k)} f_i(y; \alpha_i, \beta_i, \sigma_i^2)}.$$

3. **M-step: at the $(k+1)$ iteration, calculate the estimate of $\theta^{(k)}$ which maximize the expectation of complete log likelihood function. Let $Q(\theta, \theta^{(k)}) = E(\log L_c(\theta; y, z)|y, \theta^{(k)})$. Setting the derivatives $\partial Q(\theta, \theta^{(k)})/\partial\theta = 0$. By the constraint $\sum_{i=1}^{g} \pi_i = 1$, π_i will be**

$$\pi_i^{(k+1)} = \sum_{j=1}^{n} \tau_{ji}^{(k+1)}/n,$$

and β_i, σ_i^2 will satisfies the following equations

$$\sum_{j=1}^n \tau_{ji}^{(k+1)} Y_j - \alpha_i \sum_{j=1}^n \tau_{ji}^{(k+1)} - \sum_{j=1}^n \tau_{ji}^{(k+1)} (W_j - \mu_{u|W_j})'\beta_i = 0,$$

$$\sum_{j=1}^n \tau_{ji}^{(k+1)} (Y_j - \alpha_i - (W_j - \mu_{u|W_j})'\beta_i)(W_j - \mu_{u|W_j}) = 0,$$

$$(\sigma_i^2 + \beta_i'\Lambda\beta_i) \sum_{j=1}^n \tau_{ji}^{(k+1)} - \sum_{j=1}^n \tau_{ji}^{(k+1)} (Y_j - \alpha_i - (W_j - \mu_{u|W_j})'\beta_i)^2 = 0.$$

Write $\tilde{W}_j = W_j - \mu_{u|W_j}$, the solutions have the following explicit form:

$$\begin{pmatrix} \hat{\alpha}_i^{(k+1)} \\ \hat{\beta}_i^{(k+1)} \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^n \tau_{ji}^{(k+1)} & \sum_{j=1}^n \tau_{ji}^{(k+1)} \tilde{W}_j' \\ \sum_{j=1}^n \tau_{ji}^{(k+1)} \tilde{W}_j & \sum_{j=1}^n \tau_{ji}^{(k+1)} \tilde{W}_j \tilde{W}_j' \end{pmatrix}^{-1} \begin{pmatrix} \sum_{j=1}^n \tau_{ji}^{(k+1)} Y_j \\ \sum_{j=1}^n \tau_{ji}^{(k+1)} Y_j \tilde{W}_j \end{pmatrix}$$

and

$$\hat{\sigma}_i^{2(k+1)} = \frac{\sum_{j=1}^n \tau_{ji}^{(k+1)} (Y_j - \alpha_i^{(k+1)} - \tilde{W}_j'\hat{\beta}_i^{(k+1)})^2}{\sum_{j=1}^n \tau_{ji}^{(k+1)}} - \hat{\beta}_i^{(k+1)'} \Lambda \beta_i^{(k+1)}.$$

4. Iterate step 2 and step 3 until certain convergence criterion is attained.

If we further assume that all $\sigma_i^2 = \sigma^2$, $i = 1, 2, \ldots, g$, the updated α_i, β_i and σ^2 should satisfy equations $\partial Q(\theta, \theta^{(k)})/\partial\alpha_i = 0$, $\partial Q(\theta, \theta^{(k)})/\partial\beta_i = 0$ and $\partial Q(\theta, \theta^{(k)})/\partial\sigma^2 = 0$. Thus,

$$\sum_{j=1}^n \tau_{ji}^{(k+1)} (Y_j - \alpha_i - \tilde{W}_j'\beta_i) = 0,$$

$$\delta_i^2 \Lambda\beta_i \sum_{j=1}^n \tau_{ji}^{(k+1)} - \delta_i^2 \sum_{j=1}^n \tau_{ji}^{(k+1)} (Y_j - \alpha_i - \tilde{W}_j'\beta_i)\tilde{W}_j - \Lambda\beta_i \sum_{j=1}^n \tau_{ji}^{(k+1)} (Y_j - \alpha_i - \tilde{W}_j'\beta_i)^2 = 0,$$

$$\sum_{j=1}^n \sum_{i=1}^g \frac{\tau_{ji}^{(k+1)}}{\delta_i^2} - \sum_{j=1}^n \sum_{i=1}^g \tau_{ji}^{(k+1)} \frac{(Y_j - \alpha_i - \tilde{W}_j'\beta_i)^2}{\delta_i^4} = 0.$$

These equations have no explicit solutions, and a numerical algorithm is needed. To avoid complicated computations, we might consider using the simple average of the ones obtained from unequal cases, that is,

$$\hat{\sigma}^{2(1)} = \frac{1}{g} \sum_{i=1}^g \hat{\sigma}_i^{2(1)}$$

to estimate σ^2.

2.3 The Robust EM Algorithm for Mixture EIV Linear Regression with t Distribution

As we mentioned in introduction, the maximum likelihood procedure based on normality assumption is very sensitive to outliers, and the resulting MLE becomes very unstable when the normal distribution is contaminated by heavier tail distributions. To robustly estimate the regression parameters in model (2.1), similar to Wei and Yao (2012), here we assume that marginally ξ_i follows a t-distribution, with location parameter 0, scale parameter $\delta_i = \sqrt{\sigma_i^2 + \beta_i'\Omega\beta_i}$, and degrees of freedom v_i. That is, the maximum likelihood estimate of $\theta = (\pi_1, \alpha_1, \beta_1', \sigma_1, \ldots, \pi_g, \alpha_g, \beta_g', \sigma_g)'$, based on the marginal distribution of ξ_i, is given by

$$\tilde{\theta}_n = \text{argmax}_\theta \sum_{j=1}^{n} \log \sum_{i=1}^{g} \pi_i f(Y_j, \eta_{ji}, \delta_i, v_i), \tag{2.6}$$

where $\eta_{ji} = \alpha_i + W_j'\beta_i$,

$$f(y, \eta, \delta, v) = \frac{\Gamma((v+1)/2)}{\delta\sqrt{\pi v}\Gamma(v/2)[1 + d(y, \eta, \delta)/v]^{(v+1)/2}} \tag{2.7}$$

and $d(y, \eta, \delta) = (y - \eta)^2/\delta^2$. We can also assume that given W_j, the conditional distribution of ξ_i is a t-distribution with location parameter $-\mu_{u|W_j}'\beta_i$, scale parameter $\delta_i = \sqrt{\sigma_i^2 + \beta_i'\Lambda\beta_i}$, and degrees of freedom v_i. That is the maximum likelihood estimate of $\theta = (\pi_1, \alpha_1, \beta_1', \sigma_1^2, \ldots, \pi_g, \alpha_g, \beta_g', \sigma_g^2)'$, based on the conditional distribution of ξ_i given W_j, is given by

$$\hat{\theta}_n = \text{argmax}_\theta \sum_{j=1}^{n} \log \sum_{i=1}^{g} \pi_i f(Y_j, \eta_{ji}, \delta_i, v_i), \tag{2.8}$$

where $\eta_{ji} = \alpha_i + (W_j - \mu_{u|W_j}')\beta_i$, and f is defined in (2.7).

Unfortunately, neither (2.6) nor (2.8) provides explicit solutions. In the following, an EM algorithm will be developed to obtain some approximations to the MLEs.

2.3.1 EM Algorithm

The similarity between (2.6) and (2.8) suggests the same EM algorithms could be developed for both cases. For the sake of brevity, we will demonstrate the EM algorithm for (2.8).

Let's first assume that v_i's are known. Suppose we can observe the latent variable z, then the complete log-likelihood function will be

$$\log L_c(\theta; Y, W, z) = \sum_{j=1}^{n} \sum_{i=1}^{g} z_{ij} \log \pi_i f(Y_j, \eta_{ji}, \delta_i, v_i), \tag{2.9}$$

here $\eta_{ji} = \alpha_i + (W_j - \mu_{u|W_j})'\beta_i$, $\delta_i^2 = \sigma_i^2 + \beta_i'\Lambda\beta_i$, and f is defined in (2.7). Note that there is still no explicit solution to maximize (2.9).

It is well known that t-distribution can be considered as a scale mixture of normal distributions. Let R be a latent positive random variable such that

$$T|R = r \sim N(0, \delta^2/r), \quad R \sim \text{Gamma}(v/2, v/2), \quad v > 0,$$

where $N(0, \delta^2/r)$ denotes the normal distribution with mean 0 and variance δ^2/r, and Gamma$(v/2, v/2)$ denotes the Gamma distribution with density function

$$h(r, v/2, v/2) = \frac{1}{\Gamma(v/2)(v/2)^{-v/2}} r^{v/2-1} \exp(-2r/v).$$

Then marginally T has a t-distribution with degrees of freedom v and scale parameter δ. Therefore, if we can observe R, then the complete marginal likelihood has the following form

$$\begin{aligned}
\log L_c(\theta; Y, W, R, z) &= \sum_{j=1}^{n} \sum_{i=1}^{g} z_{ji} \log \pi_i \phi(Y_j, \eta_{ji}, \delta_i^2/R_j) h(R_j; v_i/2, v_i/2) \\
&= \sum_{j=1}^{n} \sum_{i=1}^{g} z_{ji} \log \pi_i + \sum_{j=1}^{n} \sum_{i=1}^{g} z_{ji} \log \phi(Y_j, \eta_{ji}, \delta_i^2/R_j) \\
&\quad + \sum_{j=1}^{n} \sum_{i=1}^{g} z_{ji} \log h(r_j; v_i/2, v_i/2),
\end{aligned} \tag{2.10}$$

where $\phi(y, a, b)$ denotes the normal density function with mean a and variance b. The second term on the right hand side of (2.10) can be written as

$$-\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{g} z_{ji} \log 2\pi\delta_i^2 + \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{g} z_{ji} \log R_j - \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{g} z_{ji} R_j (Y_j - \eta_{ji})^2/\delta_i^2.$$

and the third term on the right hand side of (2.10) can be written as

$$-\sum_{j=1}^{n} \sum_{i=1}^{g} z_{ji} \log \Gamma(v_i/2) - \sum_{j=1}^{n} \sum_{i=1}^{g} z_{ji} (v_j/2) \log(v_i/2) - 2 \sum_{j=1}^{n} \sum_{i=1}^{g} z_{ji} R_j/v_i$$

$$+\sum_{j=1}^{n}\sum_{i=1}^{g} z_{ji}(v_i/2-1)\log R_j.$$

Let's first assume that $v_i, i=1,2,\ldots,g$ are known. From EM procedure, for some initial value of $\theta^{(0)}$, we should calculate $E[\log L_c(\theta;Y,W,R,Z)|Y,W,\theta^{(0)}]$ first. Based on previous analysis, we can see that only the following term is needed in the maximization step

$$L(\theta) := \sum_{j=1}^{n}\sum_{i=1}^{g}\tau_{ji}^{(1)}\log\pi_i - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{g}\tau_{ji}^{(1)}\log\delta_i^2 - \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{g}\tau_{ji}^{(1)}r_{ji}^{(1)}(Y_j-\eta_{ji})^2/\delta_i^2,$$

where

$$\tau_{ji}^{(1)} = E[z_{ji}|Y,W,\theta^{(0)}] = \frac{\pi_i^{(0)} f(Y_j,\eta_{ji}^{(0)},\delta_i^{(0)},v_i)}{\sum_{i=1}^{g}\pi_i^{(0)} f(Y_j,\eta_{ji}^{(0)},\delta_i^{(0)},v_i)}, \tag{2.11}$$

$$r_{ji}^{(1)} = E[R_j|Y,W,\theta^{(0)},Z_{ji}=1] = \frac{v_i+p}{v_i+d(Y_j;\eta_{ji}^{(0)},\delta_i^{(0)})}, \tag{2.12}$$

and $\delta_i^{(0)} = \sqrt{\sigma_i^{2(0)}+\beta_i^{(0)\prime}\Lambda\beta_i^{(0)}}$, $\eta_{ji}^{(0)} = \alpha_i^{(0)}+(W_j-\mu_{u|W_j})'\beta_i^{(0)}$. Setting the derivatives $\partial L(\theta)/\partial\theta = 0$, keep in mind that the constraint $\sum_{i=1}^{g}\pi_i = 1$, we can show that the updated π_i will be

$$\pi_i^{(1)} = \sum_{j=1}^{n}\tau_{ji}^{(1)}/n, \tag{2.13}$$

and denoting $w_{ji}^{(1)} = \tau_{ji}^{(1)}r_{ji}^{(1)}$ the updated β_i, σ_i^2 will satisfies the following equations

$$\sum_{j=1}^{n} w_{ji}^{(1)}Y_j - \alpha_i\sum_{j=1}^{n} w_{ji}^{(1)} - \sum_{j=1}^{n} w_{ji}^{(1)}(W_j-\mu_{u|W_j})'\beta_i = 0,$$

$$\sum_{j=1}^{n} w_{ji}^{(1)}(Y_j-\alpha_i-(W_j-\mu_{u|W_j})'\beta_i)(W_j-\mu_{u|W_j}) = 0,$$

$$(\sigma_i^2+\beta_i'\Lambda\beta_i)\sum_{j=1}^{n}\tau_{ji}^{(1)} - \sum_{j=1}^{n} w_{ji}^{(1)}(Y_j-\alpha_i-(W_j-\mu_{u|W_j})'\beta_i)^2 = 0.$$

Write $\tilde{W}_j = W_j-\mu_{u|W_j}$, the solutions have the following explicit form:

$$\begin{pmatrix}\hat{\alpha}_i^{(1)}\\ \hat{\beta}_i^{(1)}\end{pmatrix} = \begin{pmatrix}\sum_{j=1}^{n} w_{ji}^{(1)} & \sum_{j=1}^{n} w_{ji}^{(1)}\tilde{W}_j'\\ \sum_{j=1}^{n} w_{ji}^{(1)}\tilde{W}_j & \sum_{j=1}^{n} w_{ji}^{(1)}\tilde{W}_j\tilde{W}_j'\end{pmatrix}^{-1}\begin{pmatrix}\sum_{j=1}^{n} w_{ji}^{(1)}Y_j\\ \sum_{j=1}^{n} w_{ji}^{(1)}Y_j\tilde{W}_j\end{pmatrix} \tag{2.14}$$

and

$$\hat{\sigma}_i^{2(1)} = \frac{\sum_{j=1}^n w_{ji}^{(1)}(Y_j - \alpha_i^{(1)} - \tilde{W}_j'\hat{\beta}_i^{(1)})^2}{\sum_{j=1}^n \tau_{ji}^{(1)}} - \hat{\beta}_i^{(1)\prime}\Lambda\beta_i^{(1)}. \tag{2.15}$$

If we further assume that all $\sigma_i^2 = \sigma^2$, $i = 1, 2, \ldots, g$, the updated α_i, β_i and σ^2 should satisfy equations $\partial L(\theta)/\partial\alpha_i = 0$, $\partial L(\theta)/\partial\beta_i = 0$ and $\partial L(\theta)/\partial\sigma^2 = 0$. That is

$$\begin{aligned}\sum_{j=1}^n w_{ji}^{(1)}(Y_j - \alpha_i - \tilde{W}_j'\beta_i) &= 0\\ \delta_i^2\Lambda\beta_i\sum_{j=1}^n \tau_{ji}^{(1)} - \delta_i^2\sum_{j=1}^n w_{ji}^{(1)}(Y_j - \alpha_i - \tilde{W}_j'\beta_i)\tilde{W}_j - \Lambda\beta_i\sum_{j=1}^n w_{ji}^{(1)}(Y_j - \alpha_i - \tilde{W}_j'\beta_i)^2 &= 0\\ \sum_{j=1}^n\sum_{i=1}^g \frac{\tau_{ji}^{(1)}}{\delta_i^2} - \sum_{j=1}^n\sum_{i=1}^g w_{ji}^{(1)}\frac{(Y_j - \alpha_i - \tilde{W}_j'\beta_i)^2}{\delta_i^4} &= 0\end{aligned} \tag{2.16}$$

These equations have no explicit solutions, and a numerical algorithm is needed. To avoid complicated computations, we might use the simple average of the ones obtained from unequal cases, that is,

$$\hat{\sigma}^{2(1)} = \frac{1}{g}\sum_{i=1}^g \hat{\sigma}_i^{2(1)} \tag{2.17}$$

to estimate σ^2. In summary, we propose the following EM algorithm to obtain (2.8).

EM Algorithm:

1. **Select initial values for $\theta^{(0)} = (\pi_1^{(0)}, \alpha_1^{(0)}, \beta_1'^{(0)}, \sigma_1^{(0)}, \ldots, \pi_g^{(0)}, \alpha_g^{(0)}, \beta_g'^{(0)}, \sigma_g^{(0)})'$.**

   At the $(k+1)$-th iteration,

2. **E-Step: Calculate $\tau_{ji}^{(k+1)}$, $r_{ji}^{(k+1)}$ using formula (2.11) and (2.12) with $\theta^{(0)}$ replaced by $\theta^{(k)}$.**

3. **M-Step: Calculate the updated $\theta^{(k+1)}$ using formula (2.13), (2.14), and (2.15). If σ_i^2's are assumed to be equal, we can either directly solve equation (2.16), or just update σ^2 using (2.17).**

4. **Repeat steps 2 and 3 until the result becomes stable.**

One can also develop a similar EM algorithm for solving (2.6) which is based on the marginal distribution of ξ_i's.

2.4 Modified EM Algorithm Against X-Outliers

It is well known that the EM algorithm for linear EIV models with t-distribution can handle the y-outliers very well, but it does not work satisfactorily in terms of identifying the outlying x-observations. Thus, extra caution should be taken when there are some x-outliers present in the data.

If X is observable, then we can detect the x-outliers through the hat matrix $H = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$, where $\mathbf{X} = (X_1, ..., X_n)'$, and h_{ii} is the i^{th} diagonal elements of the hat matrix. Note that $\sum_{i=1}^{n} h_{ii} = p$ and $0 \leq h_{ii} \leq 1$. It showed that h_{ii} is a distance between the i^{th} case of X which is X_i and the mean of X which is $\bar{X}$. A large value of h_{ii} indicates that X_i is distant with the center of all X observations. Based on Kutner's theory, The h_{ii}, which is called the leverage of the i^{th} case, can be a good rule of thumb to identify X_i as a high leverage point when $h_{ii} > 2p/n$. Denoting S as the sample covariance, one can show that

$$h_{ii} = n^{-1} + (n-1)^{-1} MD_i$$

where,

$$MD_i = (X_i - \bar{X})' S^{-1} (X_i - \bar{X}) \tag{2.18}$$

Since $\bar{X}$ and S might create masking effect which means that, in some case, using above equation of MD_i might not be able to identify the X-Outliers, Rousseeuw and Van Zomeren constructed a new modified Mahalanobis distance (MD_i) to avoid this problem, which is presented in the following:

$$MD_i = (X_i - m(\mathbf{X}))' C(\mathbf{X})^{-1} (X_i - m(\mathbf{X})), \tag{2.19}$$

where $m(\mathbf{X})$ and $C(\mathbf{X})$ are some robust estimates of the mean and covariance matrix of X. Various methods have been explored to estimate the multivariate location and scatter which is $m(\mathbf{X})$ and $C(\mathbf{X})$. Examples include Stahel Donoho estimators (SD) (Donoho 1982), minimum volume ellipsoid estimators (Rousseeuw, 1984), S-estimator (Davies, 1987), minimum covariance determinant estimators (MCD) (Rousseeuw and Van Driessen,1999) and depth based estimators (Zuo et.,2004). In this report, we adopted both minimum covariance determinant and Stahel Donoho estimators (SD) estimators to calculate $m(\mathbf{X})$ and $C(\mathbf{X})$. When $MD_i > \chi^2_{p,0.975}$, X_i is identified as a x-Outlier. Where MD_i is the same as the robust distance(Rousseeuw and Leroy, 1987), and $\chi^2_{p,0.975}$ is the cut point which improves the finite sample efficiency for raw minimum covariance determinant estimators by one step weighted estimate proposed by Pison et al. (2002).

Unfortunately, X is not observable in errors-in-variables model. It is then natural to replace the unknown quantities in (2.19) by their corresponding estimates or predictions. Based on the measurement error structure $W = X + u$, we can predict X by $\tilde{W} = W - \Omega(\Omega + \Sigma)^{-1}(W - \mu)$, the regression function of X against W. Then the modified MD_i will be defined as

$$MD_i = (\tilde{W}_i - m(\tilde{\mathbf{W}}))' C(\tilde{\mathbf{W}})^{-1} (\tilde{W}_i - m(\tilde{\mathbf{W}})), \tag{2.20}$$

where $m(\tilde{\mathbf{W}})$, $C(\tilde{\mathbf{W}})$ can be chosen as the MCD or SD estimate.

2.5 Adaptive Choice of Degrees of Freedom

In the EM algorithm proposed in the previous section for mixture EIV linear regression with t-distribution, we assume that the degrees of freedom of t-distribution is known, which is not a practical assumption in the real applications. How can we adaptively choose the degree of freedom if it is unknown? Based on Peel and Mclachlan(2000), in this case, we have to compute on the M-step the updated estimate $\upsilon_i^{(k+1)}$ of υ_i.

$$\partial Q(\theta, \theta^{(k)}) / \partial \upsilon_i = 0$$

Unfortunately, McLachlan (2000) noticed that the convergence of the EM algorithm was slow for unknown υ and the one-dimensional search for the computation of $\upsilon^{(k+1)}$ was time consuming. To avoid this problem, we define the profile likelihood for υ, which is

$$L(\upsilon) = \max_{\theta} \sum_{j=1}^{n} \log[\sum_{i=1}^{g} \pi_i f(y_j; x_j' \beta_i, \sigma_i^2, \upsilon)]$$

Then the estimate of the υ is given by

$$\hat{\upsilon} = \arg\max_{\upsilon} L(\upsilon)$$

Since for fixed υ, $L(\upsilon)$ can be easily calculated based on the robust EM algorithm for the linear EIV model(section 2.3). Here for the unknown υ, we use the same method to calculate $L(\upsilon)$ in a set of grid points of υ, which is $\upsilon = 1, ..., \upsilon_{max}$. When υ_{max} is large enough, the t-distribution will converge to the normal distribution. Actually,υ_{max} need not be too large, usually between 15 and 20 is large enough. For simplicity, in the simulation study, we shall assume that all the degrees of freedom are equal, that is $\upsilon_1 = \upsilon_2 = ... = \upsilon_g = \upsilon$, although the same methodology also applies when the degrees of freedom are not equal.

Chapter 3

Numerical Studies

Simulation studies will be conducted in this section to evaluate the effectiveness of the proposed algorithm. In the simulation study, if one observation exactly lies on one component line and the corresponding variance goes to 0, the log likelihood function (2.4) is unbounded and goes to infinity. This makes simulation very unstable. Also because of considering the simplicity of computation and shortening the computation time, we choose equal variance for all components. We generate the independent and identically distributed (i.i.d.) data $\{(W_{1i}, W_{2i}, y_i), i = 1, \ldots, n\}$ from the model

$$Y = \begin{cases} 1 + X_1 + X_2 + \epsilon_1, & \text{if } z = 1; \\ -1 - X_1 - X_2 + \epsilon_2, & \text{if } z = 2. \end{cases}$$

$$W_1 = X_1 + u_1, W_2 = X_2 + u_2,$$

where z is component indicator of Y with $P(z = 1) = 0.25$, $X_1 \sim N(0, 1)$, $X_2 \sim N(0, 1)$, where X_1 and X_2 are independent, $u \sim N(0, 0.25)$, ϵ_1 and ϵ_2 have the same distribution and are independent. To see the effects of different distributions of ϵ and the high leverage outliers in x-direction on various estimation methods, we consider the following six cases:

Case I : $\epsilon \sim N(0, 1)$, normal distribution with mean zero and variance 1.

Case II : $\epsilon \sim$ Laplace distribution with location parameter 0 and scale parameter 1.

Case III : $\epsilon \sim t_1$, t-distribution with degree freedom 1 or the Cauchy distribution.

Case IV : $\epsilon \sim t_3$, t-distribution with degree freedom 3.

Case V : $\epsilon \sim 0.95N(0,1) + 0.05N(0,25)$, a mixture of two normal distribution with mean zero and variance $\sqrt{0.95 * 1 + 0.05 * 25}$.

Case VI : $\epsilon \sim N(0,1)$ with 5% high leverage outliers being $\tilde{W}_1 = \tilde{W}_2 = 20$ and $Y = 100$.

Four different methods will be compared in the simulation study:

1. the EM algorithm for mixture EIV linear regression with normal distribution, which is the traditional EM algorithm method for the mixture EIV linear regression model. We termed it as MLE.

2. the robust EM algorithm for mixture EIV linear regression with t-distribution, which is the robust EM algorithm, we proposed, for mixture EIV linear regression model. We termed it as Mixregt.

3. the proposed modified EM algorithm against X-Outliers for the robust EM algorithm for mixture EIV linear regression with t distribution, with MCD trimming method. We termed it as Mixregt-MCD.

4. the trimmed mixture regression based on t-distribution, with SD trimming method. We termed it as Mixregt-SD.

In each case, we report the mean squared errors (MSE) and bias (Bias) of the parameter estimates for the different estimation methods. Sample sizes in each case are 100, 200 and 400 with 200 repetitions. It is well known that the label switching issue is always an issue when evaluating different estimation methods in mixture models, and there are no widely accepted labeling standard. In our simulation, similar to Yao and Wei (2012), we simply choose the labels by minimizing the distance to the true parameter values.

In case I, the error ϵ follows normal distribution with mean 0 and variance 1, which is often used to test the effectiveness of robust EM algorithm methods compared with traditional method. In this case, the traditional method is most efficient. The simulation shows that, when n=100, the MSE and Bias of the MLE is slightly bigger than the methods we proposed, but the superiority of MLE over all other methods becomes clear when the sample size gets bigger.

In case II, $\epsilon \sim$ Laplace distribution with location parameter 0 and scale parameter 1 is also called double exponential distribution. All of these four methods works well for this case.

In case III, the error ϵ follows t-distribution with degree of freedom 1, which is a Cauchy distribution with an extremely heavy tail. In case IV, the error ϵ follows t-distribution with degree of freedom 3, which is also heavy tail distribution, since the t-distribution with degree of freedom from 3 to 5 are often consider to present heavy tail distribution. In the simulation, we see that the traditional EM algorithm method can not provide reasonable estimates. Our proposed robust EM algorithm for mixture EIV linear regression model works better than the traditional EM algorithm method in both of these two cases.

In case V, $\epsilon \sim 0.95N(0,1) + 0.05N(0,25)$ is the contaminated normal mixture model, which is often used to mimic the outlier situation. The five percent data from $N(0,25)$ are likely to be low leverage outliers. From the table 3.1, table 3.2 and table 3.3, it is clear that the bias of our robust method is smaller than the bias of traditional method, which means that our proposed robust EM algorithm for mixture EIV linear regression model works much better than the traditional EM algorithm method.

In case VI, $\epsilon \sim N(0,1)$ with 5% high leverage outliers being $\tilde{W}_1 = \tilde{W}_2 = 20$ and $Y = 100$ means that 5% observations are replicated serving as the high leverage outliers, which will be used to check the robustness of estimation procedures against the outlier in the x-direction. Compared following three tables, we notice that except the MLE can not provided reasonable estimates, the modified procedure Mixregt-MCD and Mixregt-SD works better than the EM algorithm method (Mixregt), which means that the high leverage outliers problem is remedied by the modified procedure Mixregt-MCD and Mixregt-SD.

In addition, we also notice that, for all six cases, all of these four methods provide better results when the sample size becomes bigger. The bigger MSEs in the Mixregt, Mixregt-MCD and Mixregt-SD procedures might be resulted from that the extra step involved in the algorithm, the selection of v, which is the degrees of freedom of the t-distribution.

	MLE	Mixregt	Mixregt-MCD	Mixregt-SD
	Case I: $\varepsilon \sim N(0,1)$			
β_{10}	0.117(-0.050)	0.049(-0.113)	0.034(-0.099)	0.044(-0.150)
β_{11}	0.122(-0.130)	0.073(-0.220)	0.060(-0.130)	0.047(-0.123)
β_{12}	0.136(-0.119)	0.078(-0.223)	0.049(-0.114)	0.063(-0.139)
β_{20}	0.019(-0.029)	0.011(0.012)	0.010(0.028)	0.011(0.022)
β_{21}	0.019(0.023)	0.044(0.186)	0.015(0.039)	0.018(0.062)
β_{22}	0.024(0.009)	0.044(0.187)	0.017(0.059)	0.018(0.051)
π_1	0.004(0.041)	0.006(0.057)	0.009(0.078)	0.008(0.079)
	Case II: $\varepsilon \sim$ Laplace(1)			
β_{10}	0.249(0.068)	0.172(0.023)	0.107(0.064)	0.134(0.017)
β_{11}	0.250(-0.073)	0.155(-0.240)	0.136(-0.158)	0.141(-0.131)
β_{12}	0.218(-0.122)	0.143(-0.238)	0.156(-0.150)	0.172(-0.172)
β_{20}	0.042(-0.089)	0.034(-0.072)	0.036(-0.099)	0.030(-0.073)
β_{21}	0.042(-0.014)	0.051(0.174)	0.030(0.024)	0.036(0.032)
β_{22}	0.044(-0.029)	0.061(0.195)	0.034(0.037)	0.035(0.007)
π_1	0.005(0.039)	0.008(0.055)	0.010(0.084)	0.009(0.073)
	Case III: $\varepsilon \sim t_1$			
β_{10}	500.444(-2.926)	0.975(-0.713)	0.847(-0.671)	0.909(-0.710)
β_{11}	178.742(-1.207)	0.922(-0.713)	1.118(-0.642)	0.914(-0.567)
β_{12}	694.489(0.280)	0.849(-0.676)	1.023(-0.594)	1.025(-0.647)
β_{20}	485.054(-1.691)	0.099(-0.026)	0.073(-0.017)	0.064(-0.069)
β_{21}	164.852(-0.074)	0.081(0.147)	0.109(-0.036)	0.095(-0.039)
β_{22}	668.405(1.279)	0.091(0.158)	0.105(-0.020)	0.091(-0.078)
π_1	0.078(0.210)	0.025(0.138)	0.027(0.147)	0.024(0.140)
	Case IV: $\varepsilon \sim t_3$			
β_{10}	1.640(0.124)	0.395(0.090)	0.310(0.137)	0.413(0.178)
β_{11}	1.657(-0.023)	0.284(-0.289)	0.417(-0.231)	0.381(-0.221)
β_{12}	1.703(-0.069)	0.328(-0.303)	0.280(-0.171)	0.425(-0.250)
β_{20}	0.090(-0.021)	0.066(-0.074)	0.103(-0.185)	0.076(-0.140)
β_{21}	0.084(0.024)	0.066(0.160)	0.094(0.029)	0.078(0.005)
β_{22}	0.105(0.026)	0.081(0.205)	0.082(-0.001)	0.079(-0.018)
π_1	0.009(0.010)	0.009(0.050)	0.011(0.074)	0.011(0.071)
	Case V: $\varepsilon \sim 0.95N(0,1) + 0.05N(0,25)$			
β_{10}	1.462(0.119)	0.235(0.026)	0.144(0.074)	0.159(0.010)
β_{11}	2.873(0.083)	0.259(-0.298)	0.245(-0.144)	0.211(-0.164)
β_{12}	1.225(-0.054)	0.247(-0.249)	0.186(-0.139)	0.190(-0.133)
β_{20}	0.212(0.003)	0.047(-0.052)	0.040(-0.081)	0.048(-0.082)
β_{21}	0.200(0.031)	0.059(0.164)	0.039(-0.004)	0.046(-0.012)
β_{22}	0.320(-0.058)	0.068(0.181)	0.043(0.026)	0.055(0.017)
π_1	0.015(0.016)	0.008(0.049)	0.011(0.079)	0.011(0.077)
	Case VI: $\varepsilon \sim N(0,1)$ with 5% high leverage outliers			
β_{10}	1.600(-0.330)	0.038(-0.142)	0.041(-0.100)	0.044(-0.084)
β_{11}	2.124(1.407)	0.051(-0.189)	0.048(-0.079)	0.046(-0.094)
β_{12}	2.326(1.478)	0.049(-0.185)	0.046(-0.077)	0.058(-0.093)
β_{20}	8.749(-1.586)	0.008(0.033)	0.011(-0.007)	0.010(0.002)
β_{21}	12.014(3.400)	0.035(0.169)	0.013(0.020)	0.013(0.021)
β_{22}	13.084(3.551)	0.037(0.176)	0.012(0.024)	0.016(0.024)
π_1	0.112(0.317)	0.005(0.052)	0.007(0.057)	0.007(0.066)

Table 3.1: MSE(Bias) of Point Estimates for $n = 100$

	MLE	Mixregt	Mixregt-MCD	Mixregt-SD
	Case I: $\varepsilon \sim N(0,1)$			
β_{10}	0.038(-0.032)	0.022(-0.086)	0.037(-0.143)	0.030(-0.114)
β_{11}	0.047(-0.046)	0.060(-0.208)	0.046(-0.155)	0.043(-0.147)
β_{12}	0.044(-0.057)	0.056(-0.197)	0.049(-0.155)	0.049(-0.158)
β_{20}	0.009(-0.021)	0.006(0.002)	0.007(0.021)	0.006(0.020)
β_{21}	0.008(-0.001)	0.036(0.178)	0.009(0.041)	0.009(0.038)
β_{22}	0.008(0.002)	0.037(0.183)	0.008(0.045)	0.010(0.042)
π_1	0.003(0.040)	0.004(0.051)	0.007(0.072)	0.007(0.072)
	Case II: $\varepsilon \sim$ Laplace(1)			
β_{10}	0.082(0.039)	0.071(0.044)	0.070(0.036)	0.084(0.071)
β_{11}	0.109(-0.075)	0.119(-0.241)	0.087(-0.117)	0.106(-0.157)
β_{12}	0.083(0.003)	0.106(-0.231)	0.069(-0.119)	0.111(-0.155)
β_{20}	0.017(-0.061)	0.020(-0.062)	0.016(-0.072)	0.022(-0.080)
β_{21}	0.018(-0.021)	0.041(0.177)	0.015(0.000)	0.017(0.009)
β_{22}	0.019(-0.038)	0.043(0.182)	0.017(-0.014)	0.016(0.019)
π_1	0.003(0.035)	0.004(0.039)	0.007(0.071)	0.008(0.076)
	Case III: $\varepsilon \sim t_1$			
β_{10}	111.496(-1.015)	0.939(-0.701)	1.210(-0.881)	1.074(-0.840)
β_{11}	78.213(-1.009)	0.858(-0.726)	1.220(-0.847)	1.061(-0.794)
β_{12}	60.951(-1.920)	0.857(-0.707)	1.258(-0.880)	1.106(-0.782)
β_{20}	118.134(0.613)	0.033(-0.018)	0.033(-0.028)	0.036(0.003)
β_{21}	80.524(0.913)	0.057(0.173)	0.049(-0.032)	0.060(-0.030)
β_{22}	56.469(-0.059)	0.062(0.189)	0.043(-0.016)	0.060(-0.042)
π_1	0.062(0.227)	0.023(0.133)	0.028(0.154)	0.030(0.160)
	Case IV: $\varepsilon \sim t_3$			
β_{10}	0.538(-0.122)	0.201(0.099)	0.232(0.149)	0.298(0.238)
β_{11}	0.550(-0.185)	0.168(-0.238)	0.197(-0.104)	0.218(-0.167)
β_{12}	0.482(-0.096)	0.150(-0.240)	0.167(-0.125)	0.222(-0.198)
β_{20}	0.391(-0.047)	0.036(-0.090)	0.047(-0.133)	0.046(-0.130)
β_{21}	0.189(-0.029)	0.047(0.159)	0.037(-0.013)	0.037(-0.003)
β_{22}	0.056(-0.006)	0.057(0.190)	0.041(-0.029)	0.039(-0.009)
π_1	0.024(0.038)	0.004(0.030)	0.007(0.055)	0.007(0.049)
	Case V: $\varepsilon \sim 0.95N(0,1) + 0.05N(0,25)$			
β_{10}	0.432(-0.055)	0.090(0.013)	0.124(0.081)	0.166(0.154)
β_{11}	0.408(0.005)	0.117(-0.223)	0.108(-0.140)	0.152(-0.177)
β_{12}	0.365(0.055)	0.113(-0.213)	0.127(-0.184)	0.125(-0.149)
β_{20}	0.038(0.026)	0.021(-0.050)	0.025(-0.087)	0.028(-0.080)
β_{21}	0.249(-0.010)	0.042(0.170)	0.026(0.000)	0.022(0.004)
β_{22}	0.107(-0.007)	0.046(0.182)	0.024(-0.008)	0.024(0.006)
π_1	0.007(-0.002)	0.004(0.036)	0.008(0.067)	0.007(0.060)
	Case VI: $\varepsilon \sim N(0,1)$ with 5% high leverage outliers			
β_{10}	1.221(-0.331)	0.026(-0.130)	0.025(-0.099)	0.026(-0.099)
β_{11}	2.186(1.460)	0.044(-0.191)	0.028(-0.100)	0.032(-0.109)
β_{12}	2.086(1.425)	0.042(-0.186)	0.031(-0.103)	0.045(-0.104)
β_{20}	10.514(-1.951)	0.004(0.023)	0.005(0.005)	0.005(-0.006)
β_{21}	12.611(3.527)	0.031(0.165)	0.006(0.007)	0.007(0.022)
β_{22}	12.746(3.545)	0.031(0.165)	0.007(0.020)	0.007(0.017)
π_1	0.132(0.346)	0.004(0.058)	0.008(0.073)	0.007(0.070)

Table 3.2: MSE(Bias) of Point Estimates for $n = 200$

	MLE	Mixregt	Mixregt-MCD	Mixregt-SD
	Case I: $\varepsilon \sim N(0,1)$			
β_{10}	0.015(-0.003)	0.015(-0.090)	0.029(-0.136)	0.033(-0.145)
β_{11}	0.012(-0.020)	0.047(-0.203)	0.034(-0.142)	0.038(-0.154)
β_{12}	0.014(-0.025)	0.049(-0.208)	0.034(-0.152)	0.038(-0.154)
β_{20}	0.004(-0.027)	0.002(-0.004)	0.004(0.016)	0.006(0.025)
β_{21}	0.004(-0.010)	0.033(0.178)	0.007(0.041)	0.007(0.051)
β_{22}	0.004(-0.005)	0.035(0.181)	0.006(0.040)	0.008(0.053)
π_1	0.002(0.035)	0.003(0.053)	0.005(0.065)	0.005(0.066)
	Case II: $\varepsilon \sim$ Laplace(1)			
β_{10}	0.050(0.048)	0.038(0.056)	0.030(-0.002)	0.047(0.032)
β_{11}	0.046(0.016)	0.073(-0.226)	0.030(-0.071)	0.039(-0.100)
β_{12}	0.035(-0.026)	0.076(-0.237)	0.031(-0.080)	0.043(-0.106)
β_{20}	0.009(-0.052)	0.009(-0.058)	0.011(-0.067)	0.013(-0.073)
β_{21}	0.009(-0.031)	0.036(0.176)	0.007(-0.013)	0.008(0.002)
β_{22}	0.009(-0.023)	0.037(0.182)	0.008(-0.020)	0.008(-0.006)
π_1	0.002(0.023)	0.002(0.031)	0.007(0.074)	0.007(0.072)
	Case III: $\varepsilon \sim t_1$			
β_{10}	10.223(-1.141)	1.073(-0.848)	1.636(-1.138)	1.670(-1.153)
β_{11}	17.222(-1.544)	1.018(-0.845)	1.572(-1.070)	1.742(-1.153)
β_{12}	12.170(-1.383)	0.998(-0.828)	1.692(-1.111)	1.689(-1.127)
β_{20}	9.660(0.859)	0.027(0.017)	0.031(0.024)	0.034(0.048)
β_{21}	15.046(0.456)	0.049(0.179)	0.038(0.015)	0.036(0.022)
β_{22}	10.637(0.616)	0.047(0.178)	0.030(0.003)	0.038(0.033)
π_1	0.057(0.238)	0.026(0.142)	0.037(0.183)	0.039(0.189)
	Case IV: $\varepsilon \sim t_3$			
β_{10}	0.268(-0.098)	0.084(0.085)	0.107(0.133)	0.148(0.197)
β_{11}	0.241(-0.054)	0.087(-0.211)	0.081(-0.052)	0.115(-0.101)
β_{12}	0.254(-0.083)	0.094(-0.222)	0.097(-0.100)	0.104(-0.078)
β_{20}	0.079(0.033)	0.015(-0.055)	0.029(-0.121)	0.027(-0.116)
β_{21}	0.235(-0.019)	0.041(0.184)	0.017(-0.017)	0.019(-0.015)
β_{22}	0.119(-0.020)	0.043(0.184)	0.018(-0.028)	0.021(-0.017)
π_1	0.015(0.011)	0.002(0.023)	0.004(0.044)	0.004(0.038)
	Case V: $\varepsilon \sim 0.95N(0,1) + 0.05N(0,25)$			
β_{10}	0.353(0.009)	0.047(0.062)	0.052(0.054)	0.056(0.055)
β_{11}	0.240(0.077)	0.079(-0.226)	0.060(-0.047)	0.054(-0.076)
β_{12}	0.155(-0.032)	0.076(-0.213)	0.049(-0.078)	0.061(-0.074)
β_{20}	0.014(0.029)	0.012(-0.046)	0.016(-0.065)	0.020(-0.079)
β_{21}	0.013(0.013)	0.041(0.183)	0.014(-0.003)	0.010(0.003)
β_{22}	0.018(0.026)	0.039(0.182)	0.013(-0.010)	0.011(0.000)
π_1	0.003(-0.016)	0.002(0.024)	0.006(0.055)	0.006(0.058)
	Case VI: $\varepsilon \sim N(0,1)$ with 5% high leverage outliers			
β_{10}	1.209(-0.443)	0.026(-0.145)	0.025(-0.127)	0.021(-0.098)
β_{11}	2.103(1.441)	0.041(-0.193)	0.028(-0.113)	0.019(-0.096)
β_{12}	2.118(1.446)	0.044(-0.200)	0.022(-0.113)	0.026(-0.115)
β_{20}	8.789(-1.659)	0.003(0.029)	0.003(0.000)	0.003(-0.004)
β_{21}	12.414(3.513)	0.029(0.164)	0.005(0.023)	0.004(0.015)
β_{22}	12.608(3.540)	0.029(0.166)	0.004(0.016)	0.005(0.027)
π_1	0.126(0.340)	0.003(0.054)	0.006(0.068)	0.006(0.069)

Table 3.3: MSE(Bias) of Point Estimates for $n = 400$

Chapter 4

Conclusion

The traditional estimation of the mixture linear error-in-variable regression model is based on normal assumption. The parameter are estimated by traditional MLE of EM algorithm method. But since the traditional MLE method becomes very unstable if there are outliers or the actual distribution of ε having a heavier tail than normal, in this report, we propose a robust estimation method for the mixture linear error-in-variable regression model which is based t-assumption instead of traditional normal assumption. Because this robust model is not sensitive to the high leverage outliers, we further proposed a modified EM algorithm method for the X-outliers which delete the high leverage points on the cut points $\chi^2_{p-1,0.975}$. In addition, we also provided adaptive choice of degree freedom method which based on the grid points $[1 : \upsilon_{max}]$. In our report, because the reasonable range of υ_{max} is between 15 and 20, we choose $\upsilon_{max} = 15$ to be our maximum grid point. Based on the simulation results, the robust mixture linear error-in-variable regression models by t-distribution is comparable and even more work well than the traditional method.

Bibliography

[1] Adesina, A. A., Zinnah, M. M. (1993). Technology characteristics, farmers' perceptions and adoption decisions: A Tobit model application in Sierra Leone. *Agric. Econ.*, **9**(4), 297-311.

[2] Pearson, K. (1894). Contributions to the theory of mathematical evolution. *Philosophical Transactions of the Royal Society of LondonA*, 185,71-110.

[3] Day, N.E. (1969). Estimating the components of a mixture of two normal distributions. *Biometrika*, 56, 463-474.

[4] Wolfe, J.H. (1965). A computer program for the computation of maximum likelihood analysis of types. *Research Memo.*, SRM65-12. SanDiego: U.S. Naval Personnel Research Activity.

[5] Wolfe, J.H. (1967). NORMIX: Computation methods fro estimating the parameters of multivariate normal mixtures of distribution. *Research Memo.*, SRM68-2. SanDiego: U.S. Naval Personnel Research Activity.

[6] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of Royal Statistical Society, Ser B.*, 39, 1-38.

[7] Markatou, M. (2000). Mixture models, robustness, and the weighted likelihood methodology. *Biometrics*, 56, 483-486.

[8] Cohen, E. (1984). Some effects of inharmonic partials on interval perception. *Finite Mixture Models*. New York: Wiley.

[9] Yan W. (2002). Robust mixture regerrsion models using t-distribution, Master report, Kansas State University, unpublished.

[10] Donoho, D. L. (1982). Breakdown properties of multivariate location estimators. Qualifying paper, Harvard University, Boston.

[11] Rosseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79, 871-880.

[12] Davies, L. (1987). Asymptotic behavior of S-estimators of multivariate location parameters and dispersion matrices. *Annals of Statistics*, 15, 1269-1292.

[13] Rousseeuw, P. J. and Van Driessen, K. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41, 212-223.

[14] Zuo, Y., Cui, H. and He, X. (2004). On the Stahel-Donoho estimator and depthweighted means of multivariate data. *Annals of Statistics*, 32, 167-188.

[15] Peel, D. and McLachlan, G. J. (2000). *Robust mixture modelling using the t distribution Statistics and Computing*, 10, 339-348.

[16] Susana Faria and Gilda Soromenho (2010). Fitting mixtures of linear regressions. *Journal of American Statistical Computation and Simulation*, 80, 201-205.

[17] Carroll, R. J. and Gallo, P. P. (1982). Some aspects of robustness in functional errors-in-variables regression models. *Communications in Statistics, Series A*, 11, 2573-2585.

[18] Beaton, A. E. and Tukey, J. W. (1974). The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data. *Technometrics*, 16, 147C185.

[19] Ayanendranath Basu and Sahadeb Sarkar (1997). Robust estimation in the errors-in-variables model via weighted likelihood estimating equations. *TEST*, 6, 187C203.

[20] Chi-Lun Cheng and John W. Van Ness (1992). Generalized M-Estimators for Errors-in-Variables Regression. *Ann. Statist*, 20, 385-397.

[21] Paris, Quirino (2004). Robust estimators of errors-in-variables models part I. *Agriculture and Resource Economics Working Papers.*

[22] Peter J. Huber (1972). The 1972 Wald Lecture Robust Statistics: A Review. *Ann. Math. Statist*, 43, 1041-1067.

[23] H. Zamar (1989). Robust estimation in the errors-in-variables model. *Biometrika*, 76, 149-160.

[24] M. Brown (1982). Robust line estimation with errors in both variables. *J. Amer. Statist. Assoc.*, 77, 71-79.

[25] Li, Tong (2002). Robust and consistent estimation of nonlinear errors-in-variables models. *Journal of Econometrics*, **110**(1), 1-26.

Appendix A

R code

```
# solving label switching
slab=function(theta)
{
theta0=c(0.25,1,1,1,0.75,-1,-1,-1)
ind1=c(1,2,3,4,5,6,7,8)
ind2=c(5,6,7,8,1,2,3,4)
res1=sum((theta[ind1]-theta0)^2)
res2=sum((theta[ind2]-theta0)^2)
theta=theta[ind1]*(res1<=res2)+theta[ind2]*(res1>res2)
return(theta)
}
mix.MLE.normal=function(tw,y,beta,sigma,prob,group,tol=1e-6)
{
n=length(y);
tW=cbind(rep(1,n),tw);
run=1;
gsumi=matrix(0,n,group);
for(i in seq(group))
```

```
{
gsumi[,i]=prob[i]*dnorm(y,tW%*%beta[i,],sigma);
}
prest=sum(log(apply(gsumi,1,sum)));
repeat
{
tji=matrix(0,n,group)
for(i in seq(group))
{
tji[,i]=prob[i]*pmax(dnorm(y,tW%*%beta[i,],sigma),10^(-6));
}
tji=solve(diag(apply(tji,1,sum)))%*%tji;
prob=apply(tji,2,mean);
sse=rep(0,group);
sig2=rep(0,2);
for(i in seq(group))
{
beta[i,]=lm(y~tw,weight=tji[,i])$coefficient;
sig2[i]=sum(tji[,i]*(y-tW%*%beta[i,])^2)/sum(tji[,i])-
         t(beta[i,-1])%*%Lmd%*%beta[i,-1];
sig2[i]=sig2[i]*(sig2[i]>0.0001)+0.0001*(sig2[i]<=0.0001)
}
sigma=sqrt(mean(sig2));
for(i in seq(group))
{
gsumi[,i]=prob[i]*dnorm(y,tW%*%beta[i,],sigma);
}
```

```
latest=sum(log(apply(gsumi,1,sum)));
dif=abs(latest-prest);
prest=latest;
run=run+1;
if(dif<=tol|run>500){break}
}
btemp=c(prob[1],beta[1,],prob[2],beta[2,]);
btemp=slab(btemp);
prob=c(btemp[1],btemp[5]);
beta=rbind(btemp[2:4],btemp[6:8]);
result=cbind(beta,prob)
result
}
lev.p=function(X,method="classic")
{
if(method=="sd")
{
# Stahel-Donoho Estimator, Package needed: rrcov
Temp=CovSde(X)
mx=Temp@center
cx=Temp@cov
}
else if(method=="mcd")
{
# Fast MCD Estimate, Package Needed: robustbase
Temp=covMcd(X)
mx=Temp$center
```

```
cx=Temp$cov
}
else
{
mx=apply(X,2,mean)
cx=var(X)
}
list(mean=mx,covar=cx)
}

# MLE
library(MASS)
BV=matrix(0,nrow=42, ncol=6);
total=200
kk=1;
for(n in c(100,200,400))
{
jj=1;
for(dist in seq(6))
{
set.seed(88888)
b101=b111=b121=b201=b211=b221=pi11=rep(0,total)
for(k in seq(total))
{
u=runif(n,0,1);   # A random number for assigning groups
p1=(u<=0.25);     # probability of group 1
p2=1-p1;          # probability of group 2
```

```
mx=0;
sx=1;
Sx=diag(c(sx^2,sx^2));
x1=rnorm(n,0,sx);
x2=rnorm(n,0,sx);
su=0.5;
u1=rnorm(n,0,su);
u2=rnorm(n,0,su);
Omg=diag(c(su^2,su^2));
Lmd=Omg-Omg%*%solve(Omg+Sx)%*%Omg;
W1=x1+u1;
W2=x2+u2;
e1n=rnorm(n,0,su);  # 1: normal,
e2n=rnorm(n,0,su);
e1L=rexp(n,sqrt(2))-rexp(n,sqrt(2)); # 2: laplace,
e2L=rexp(n,sqrt(2))-rexp(n,sqrt(2));
e1t1=rt(n,1);   # 3: t(1),
e2t1=rt(n,1);
e1t3=rt(n,3);   # 4: t(3),
e2t3=rt(n,3);
u=runif(n,0,1)
e1mn=(u<0.95)*rnorm(n,0,1)+(u>=0.95)*rnorm(n,0,5)  # 5: normal_mixture,
e2mn=(u<0.95)*rnorm(n,0,1)+(u>=0.95)*rnorm(n,0,5)
e1=e1n*((dist==1)|(dist==6))+e1L*(dist==2)+e1t1*(dist==3)+
   e1t3*(dist==4)+e1mn*(dist==5)
e2=e2n*((dist==1)|(dist==6))+e2L*(dist==2)+e2t1*(dist==3)+
   e2t3*(dist==4)+e2mn*(dist==5)
```

```
y1=1+x1+x2+e1;
y2=-1-x1-x2+e2;
W=cbind(W1,W2);
tW=W-(W-mx)%*%solve(Omg+Sx)%*%Omg;
y=y1*p1+y2*p2;
if(dist==6)   # 6: normal+0.05outlier
{
nout=round(n*0.95)+1;
tW[nout:n,]=20;
y[nout:n]=100;
}
# Initial Values
prob=c(0.5,0.5);
b1=c(0.9,0.9,0.9);
b2=c(-0.9,-0.9,-0.9);
beta=rbind(b1,b2);
clust=kmeans(y,2)
sig01=sd(y[clust$cluster==1])
sig02=sd(y[clust$cluster==2])
if(is.na(sig01)){sig01=0};
if(is.na(sig02)){sig02=0};
if((sig01==0)|(sig02=0)){sig0=max(sig01,sig02)} else
{sig0=(sig01+sig02)/2};
group=2;
result=mix.MLE.normal(tW,y,beta,sig0,prob,group,1e-6);
b101[k]=result[1,1]
b111[k]=result[1,2]
```

```
b121[k]=result[1,3]
b201[k]=result[2,1]
b211[k]=result[2,2]
b221[k]=result[2,3]
pi11[k]=result[1,4]
}
tb10=1; tb11=1;  tb12=1;
tb20=-1; tb21=-1; tb22=-1;
tpi1=0.25;
cat("jj=",jj,"\n")
Rmle=cbind(b101-tb10,b111-tb11,b121-tb12,b201-tb20,b211-tb21,b221-tb22,pi11-tpi1);
BV[seq((jj-1)*7+1,jj*7),c((kk-1)*2+1,kk*2)]=cbind(
   apply(Rmle,2,function(x){mean(x^2)}),apply(Rmle,2,mean))
jj=jj+1;
}
cat("kk=",kk,"\n")
kk=kk+1
}
round(BV,3)

# Mixtregt
dent=function(y,mu,sig,v)
{
est=gamma((v+1)/2)*sig^(-1)/((pi*v)^(1/2)*gamma(v/2)*
(1+(y-mu)^2/(sig^2*v))^(0.5*(v+1)));
est
}
```

```
library(MASS)
BV=matrix(0,nrow=42,ncol=2)
n=400    # Sample Size
total=200
for(dist in seq(6))
{
set.seed(98765)
b107=b117=b127=b207=b217=b227=pi17=rep(0,total)
for(k in seq(total))
{
repeat{
u=runif(n,0,1);   # A random number for assigning groups
p1=(u<=0.25);     # probability of group 1
p2=1-p1;          # probability of group 2
mx=0;
sx=1;
Sx=diag(c(sx^2,sx^2));
x1=rnorm(n,0,sx);
x2=rnorm(n,0,sx);
su=0.5;
u1=rnorm(n,0,su);
u2=rnorm(n,0,su);
Omg=diag(c(su^2,su^2));
Lmd=Omg-Omg%*%solve(Omg+Sx)%*%Omg;
W1=x1+u1;
W2=x2+u2;
e1n=rnorm(n,0,su);  # 1: normal,
```

```
e2n=rnorm(n,0,su);
e1L=rexp(n,sqrt(2))-rexp(n,sqrt(2)); # 2: laplace,
e2L=rexp(n,sqrt(2))-rexp(n,sqrt(2));
e1t1=rt(n,1);   # 3: t(1),
e2t1=rt(n,1);
e1t3=rt(n,3);   # 4: t(3),
e2t3=rt(n,3);
u=runif(n,0,1)
e1mn=(u<0.95)*rnorm(n,0,1)+(u>=0.95)*rnorm(n,0,5)  # 5: normal_mixture,
e2mn=(u<0.95)*rnorm(n,0,1)+(u>=0.95)*rnorm(n,0,5)
e1=e1n*((dist==1)|(dist==6))+e1L*(dist==2)+e1t1*(dist==3)+
   e1t3*(dist==4)+e1mn*(dist==5)
e2=e2n*((dist==1)|(dist==6))+e2L*(dist==2)+e2t1*(dist==3)+
   e2t3*(dist==4)+e2mn*(dist==5)
y1=1+x1+x2+e1;
y2=-1-x1-x2+e2;
W=cbind(W1,W2);
tW=W-(W-mx)%*%solve(Omg+Sx)%*%Omg;
y=y1*p1+y2*p2;
if(dist==6)   # 6: normal+0.05outlier
{
nout=round(n*0.95)+1;
tW[nout:n,]=20;
y[nout:n]=100;
}
# Initial Values
prob=c(0.5,0.5);
```

```
b1=c(0.9,0.9,0.9);
b2=c(-0.9,-0.9,-0.9);
beta=rbind(b1,b2);
clust=kmeans(y,2)
sig01=sd(y[clust$cluster==1])
sig02=sd(y[clust$cluster==2])
if(is.na(sig01)){sig01=0};
if(is.na(sig02)){sig02=0};
if((sig01==0)|(sig02=0)){sig0=max(sig01,sig02)} else
{sig0=(sig01+sig02)/2};
group=2;
tW=cbind(rep(1,n),tW);
m=2;
maxv=15;
likhod=flag=rep(0,15);
theta=array(0,dim=c(2,5,15));
for(v in seq(maxv))
{
bet=beta
sig=sig0;
pr=prob;
run=0;
n=length(y);
tW=cbind(rep(1,n),W);
lh=-10^10;
a=dim(W);
p=a[2]+1;
```

```
pk=u=r=matrix(rep(0,m*n),nrow=n);
for(j in seq(m))
{
r[,j]=(y-tW%*%bet[j,])/sig;
}
repeat
{
run=run+1;
prelh=lh;
for(j in seq(m))
{
pk[,j]=pr[j]*dent(y, tW%*%bet[j,],sig,v);
u[,j]=(v+1)/(v+r[,j]^2)
}
lh=sum(log(apply(pk,1,sum)));
pk=pk/matrix(rep(apply(pk,1,sum),m),nrow=n);
dif=lh-prelh;
if(!is.finite(lh)){flag[v]=1; break}
if(dif<1e-6|run>500){flag[v]=0; break}
np=apply(pk,2,sum);pr=np/n;
sig2=rep(0,m);
for(j in seq(m))
{
w=diag(pk[,j]*u[,j]);
bet[j,]=ginv(t(tW)%*%w%*%tW)%*%t(tW)%*%w%*%y;
r[,j]=y-tW%*%bet[j,];
sig2[j]=sum(pk[,j]*r[,j]^2*u[,j])/sum(pk[,j])-t(bet[j,-1])%*%Lmd%*%bet[j,-1];
```

```
sig2[j]=sig2[j]*(sig2[j]>0.0001)+0.0001*(sig2[j]<=0.0001)
}
sig=sqrt(mean(sig2));
r=r/sig;
}
sig=sig*rep(1,m);
theta[,,v]= matrix(c(bet,pr,sig),nrow=m);
likhod[v]=lh;
}
if(all(flag==0)){break}
}
fv=which(likhod==max(likhod))
if(fv<15)
{result=theta[,,fv]} else
{
run=0;
n=length(y);
tW=cbind(rep(1,n),W);
r=matrix(rep(0,m*n),nrow=n);
pk=r;
lh=0
for(j in seq(m))
{
r[,j]=y-tW%*%bet[j,];
lh=lh+pr[j]*dnorm(r[,j],0,sig);
}
lh=sum(log(lh));
```

```
repeat
{
prest=c(bet,sig,pr);
run=run+1;
plh=lh;
for(j in seq(m))
{
pk[,j]=pr[j]* pmax(10^(-6),dnorm(r[,j],0,sig))
}
pk=pk/matrix(rep(apply(pk,1,sum),m),nrow=n);
np=apply(pk,2,sum);
pr=np/n;
sig2=rep(0,2);
for(j in seq(m))
{
w=diag(pk[,j]);
bet[j,]=ginv(t(tW)%*%w%*%tW)%*%t(tW)%*%w%*%y;
r[,j]= y-tW%*%bet[j,];
sig2[j]=sum(pk[,j]*r[,j]^2*u[,j])/sum(pk[,j])-t(bet[j,-1])%*%Lmd%*%bet[j,-1];
sig2[j]=sig2[j]*(sig2[j]>0.0001)+0.0001*(sig2[j]<=0.0001)
}
sig=sqrt(mean(sig2));
lh=0;
for(j in seq(m))
{
lh=lh+pr[j]*dnorm(r[,j],0,sig);
}
```

```
lh=sum(log(lh));
dif=lh-plh;
if((dif<1e-6)|(run>500)){break}
}
sig=sig*rep(1,m)
result=matrix(c(bet,pr,sig),nrow=m)
}
btemp=c(result[1,4], result[1,1:3], result[2,4], result[2,1:3]);
btemp=slab(btemp);
prob=c(btemp[1],btemp[5]);
beta=rbind(btemp[2:4],btemp[6:8]);
rest7=cbind(beta,prob)
b107[k]=rest7[1,1]
b117[k]=rest7[1,2]
b127[k]=rest7[1,3]
b207[k]=rest7[2,1]
b217[k]=rest7[2,2]
b227[k]=rest7[2,3]
pi17[k]=rest7[1,4]
cat("k",k,"\n")
}
tb10=1;  tb11=1;   tb12=1;
tb20=-1;  tb21=-1;   tb22=-1;
tpi1=0.25;
RL=cbind(b107-tb10,b117-tb11,b127-tb12,b207-tb20,b217-tb21,b227-tb22,pi17-tpi1);
BV[seq((dist-1)*7+1,dist*7),1:2]=cbind(
   apply(RL,2,function(x){mean(x^2)}),apply(RL,2,mean))
```

```
print(BV[seq((dist-1)*7+1,dist*7),1:2])
}
round(BV,3)


# Mixregt-MCD
library(robustbase)
library(rrcov)
library(MASS)
dent=function(y,mu,sig,v)
{
est=gamma((v+1)/2)*sig^(-1)/((pi*v)^(1/2)*gamma(v/2)*
(1+(y-mu)^2/(sig^2*v))^(0.5*(v+1)));
est
}
BV=matrix(0,nrow=42,ncol=2)
n=400    # Sample Size
total=200
for(dist in seq(6))
{
set.seed(98765)
b107=b117=b127=b207=b217=b227=pi17=rep(0,total)
for(k in seq(total))
{
repeat{
n=400;
u=runif(n,0,1);   # A random number for assigning groups
p1=(u<=0.25);     # probability of group 1
```

```
p2=1-p1;          # probability of group 2

mx=0;
sx=1;
Sx=diag(c(sx^2,sx^2));
x1=rnorm(n,0,sx);
x2=rnorm(n,0,sx);
su=0.5;
u1=rnorm(n,0,su);
u2=rnorm(n,0,su);
Omg=diag(c(su^2,su^2));
Lmd=Omg-Omg%*%solve(Omg+Sx)%*%Omg;
W1=x1+u1;
W2=x2+u2;
e1n=rnorm(n,0,su);  # 1: normal,
e2n=rnorm(n,0,su);
e1L=rexp(n,sqrt(2))-rexp(n,sqrt(2)); # 2: laplace,
e2L=rexp(n,sqrt(2))-rexp(n,sqrt(2));
e1t1=rt(n,1);   # 3: t(1),
e2t1=rt(n,1);
e1t3=rt(n,3);   # 4: t(3),
e2t3=rt(n,3);
u=runif(n,0,1)
e1mn=(u<0.95)*rnorm(n,0,1)+(u>=0.95)*rnorm(n,0,5)  # 5: normal_mixture,
e2mn=(u<0.95)*rnorm(n,0,1)+(u>=0.95)*rnorm(n,0,5)
e1=e1n*((dist==1)|(dist==6))+e1L*(dist==2)+e1t1*(dist==3)+
    e1t3*(dist==4)+e1mn*(dist==5)
```

```
e2=e2n*((dist==1)|(dist==6))+e2L*(dist==2)+e2t1*(dist==3)+
    e2t3*(dist==4)+e2mn*(dist==5)
y1=1+x1+x2+e1;
y2=-1-x1-x2+e2;
W=cbind(W1,W2);
tW=W-(W-mx)%*%solve(Omg+Sx)%*%Omg;
y=y1*p1+y2*p2;
if(dist==6)   # 6: normal+0.05outlier
{
nout=round(n*0.95)+1;
tW[nout:n,]=20;
y[nout:n]=100;
}
# Initial Values
prob=c(0.5,0.5);
b1=c(0.9,0.9,0.9);
b2=c(-0.9,-0.9,-0.9);
beta=rbind(b1,b2);
clust=kmeans(y,2)
sig01=sd(y[clust$cluster==1])
sig02=sd(y[clust$cluster==2])
if(is.na(sig01)){sig01=0};
if(is.na(sig02)){sig02=0};
if((sig01==0)|(sig02=0)){sig0=max(sig01,sig02)} else
{sig0=(sig01+sig02)/2};
group=2;
mv=lev.p(tW,method="mcd");
```

```
TX=cbind(tW[,1]-mv$mean[1],tW[,2]-mv$mean[2]);
ind=which(diag(TX%*%solve(mv$covar)%*%t(TX))>=qchisq(0.975,2));
if(length(ind)>0)
{
tW=tW[-ind,];
y=y[-ind];
}
tW1=tW[,1];
tW2=tW[,2];
n=length(y);
W=cbind(tW1,tW2);
m=2;
maxv=15;
likhod=flag=rep(0,15);
theta=array(0,dim=c(2,5,15));
for(v in seq(maxv))
{
bet=beta
sig=sig0;
pr=prob;
run=0;
n=length(y);
tW=cbind(rep(1,n),W);
lh=-10^10;
a=dim(W);
p=a[2]+1;
pk=u=r=matrix(rep(0,m*n),nrow=n);
```

```
for(j in seq(m))
{
r[,j]=(y-tW%*%bet[j,])/sig;
}
repeat
{
run=run+1;
prelh=lh;
for(j in seq(m))
{
pk[,j]=pr[j]*dent(y, tW%*%bet[j,],sig,v);
u[,j]=(v+1)/(v+r[,j]^2)
}
lh=sum(log(apply(pk,1,sum)));
pk=pk/matrix(rep(apply(pk,1,sum),m),nrow=n);
dif=lh-prelh;
if(!is.finite(lh)){flag[v]=1; break}
if(dif<1e-6|run>500){flag[v]=0; break}
np=apply(pk,2,sum);pr=np/n;
sig2=rep(0,m);
for(j in seq(m))
{
w=diag(pk[,j]*u[,j]);
bet[j,]=ginv(t(tW)%*%w%*%tW)%*%t(tW)%*%w%*%y;
r[,j]=y-tW%*%bet[j,];
sig2[j]=sum(pk[,j]*r[,j]^2*u[,j])/sum(pk[,j])-t(bet[j,-1])%*%Lmd%*%bet[j,-1];
sig2[j]=sig2[j]*(sig2[j]>0.0001)+0.0001*(sig2[j]<=0.0001)
```

```
}
sig=sqrt(mean(sig2));
r=r/sig;
}
sig=sig*rep(1,m);
theta[,,v]= matrix(c(bet,pr,sig),nrow=m);
likhod[v]=lh;
}
if(all(flag==0)){break}
}
fv=which(likhod==max(likhod))
if(fv<15)
{result=theta[,,fv]} else
{
run=0;
n=length(y);
tW=cbind(rep(1,n),W);
r=matrix(rep(0,m*n),nrow=n);
pk=r;
lh=0
for(j in seq(m))
{
r[,j]=y-tW%*%bet[j,];
lh=lh+pr[j]*dnorm(r[,j],0,sig);
}
lh=sum(log(lh));
repeat
```

```
{
prest=c(bet,sig,pr);
run=run+1;
plh=lh;
for(j in seq(m))
{
pk[,j]=pr[j]* pmax(10^(-6),dnorm(r[,j],0,sig))
}
pk=pk/matrix(rep(apply(pk,1,sum),m),nrow=n);
np=apply(pk,2,sum);
pr=np/n;
sig2=rep(0,2);
for(j in seq(m))
{
w=diag(pk[,j]);
bet[j,]=ginv(t(tW)%*%w%*%tW)%*%t(tW)%*%w%*%y;
r[,j]= y-tW%*%bet[j,];
sig2[j]=sum(pk[,j]*r[,j]^2*u[,j])/sum(pk[,j])-t(bet[j,-1])%*%Lmd%*%bet[j,-1];
sig2[j]=sig2[j]*(sig2[j]>0.0001)+0.0001*(sig2[j]<=0.0001)
}
sig=sqrt(mean(sig2));
lh=0;
for(j in seq(m))
{
lh=lh+pr[j]*dnorm(r[,j],0,sig);
}
lh=sum(log(lh));
```

```
dif=lh-plh;
if((dif<1e-6)|(run>500)){break}
}
sig=sig*rep(1,m)
result=matrix(c(bet,pr,sig),nrow=m)
}
btemp=c(result[1,4], result[1,1:3], result[2,4], result[2,1:3]);
btemp=slab(btemp);
prob=c(btemp[1],btemp[5]);
beta=rbind(btemp[2:4],btemp[6:8]);
rest7=cbind(beta,prob)
b107[k]=rest7[1,1]
b117[k]=rest7[1,2]
b127[k]=rest7[1,3]
b207[k]=rest7[2,1]
b217[k]=rest7[2,2]
b227[k]=rest7[2,3]
pi17[k]=rest7[1,4]
cat("k",k,"\n")
}
tb10=1;  tb11=1;   tb12=1;
tb20=-1;  tb21=-1;   tb22=-1;
tpi1=0.25;
RL=cbind(b107-tb10,b117-tb11,b127-tb12,b207-tb20,b217-tb21,b227-tb22,pi17-tpi1);
BV[seq((dist-1)*7+1,dist*7),1:2]=cbind(
   apply(RL,2,function(x){mean(x^2)}),apply(RL,2,mean))
print(BV[seq((dist-1)*7+1,dist*7),1:2])
```

```
}
round(BV,3)

# Mixtreg-SD
library(robustbase)
library(rrcov)
library(MASS)
dent=function(y,mu,sig,v)
{
est=gamma((v+1)/2)*sig^(-1)/((pi*v)^(1/2)*gamma(v/2)*
(1+(y-mu)^2/(sig^2*v))^(0.5*(v+1)));
est
}
BV=matrix(0,nrow=42,ncol=2)
n=400    # Sample Size
total=200
for(dist in seq(6))
{
set.seed(98765)
b107=b117=b127=b207=b217=b227=pi17=rep(0,total)
for(k in seq(total))
{
repeat{
n=400;
u=runif(n,0,1);   # A random number for assigning groups
p1=(u<=0.25);     # probability of group 1
p2=1-p1;          # probability of group 2
```

```
mx=0;
sx=1;
Sx=diag(c(sx^2,sx^2));
x1=rnorm(n,0,sx);
x2=rnorm(n,0,sx);
su=0.5;
u1=rnorm(n,0,su);
u2=rnorm(n,0,su);
Omg=diag(c(su^2,su^2));
Lmd=Omg-Omg%*%solve(Omg+Sx)%*%Omg;
W1=x1+u1;
W2=x2+u2;
e1n=rnorm(n,0,su);  # 1: normal,
e2n=rnorm(n,0,su);
e1L=rexp(n,sqrt(2))-rexp(n,sqrt(2)); # 2: laplace,
e2L=rexp(n,sqrt(2))-rexp(n,sqrt(2));
e1t1=rt(n,1);   # 3: t(1),
e2t1=rt(n,1);
e1t3=rt(n,3);   # 4: t(3),
e2t3=rt(n,3);
u=runif(n,0,1)
e1mn=(u<0.95)*rnorm(n,0,1)+(u>=0.95)*rnorm(n,0,5)  # 5: normal_mixture,
e2mn=(u<0.95)*rnorm(n,0,1)+(u>=0.95)*rnorm(n,0,5)
e1=e1n*((dist==1)|(dist==6))+e1L*(dist==2)+e1t1*(dist==3)+
   e1t3*(dist==4)+e1mn*(dist==5)
e2=e2n*((dist==1)|(dist==6))+e2L*(dist==2)+e2t1*(dist==3)+
   e2t3*(dist==4)+e2mn*(dist==5)
```

```
y1=1+x1+x2+e1;
y2=-1-x1-x2+e2;
W=cbind(W1,W2);
tW=W-(W-mx)%*%solve(Omg+Sx)%*%Omg;
y=y1*p1+y2*p2;
if(dist==6)   # 6: normal+0.05outlier
{
nout=round(n*0.95)+1;
tW[nout:n,]=20;
y[nout:n]=100;
}
# Initial Values
prob=c(0.5,0.5);
b1=c(0.9,0.9,0.9);
b2=c(-0.9,-0.9,-0.9);
beta=rbind(b1,b2);
clust=kmeans(y,2)
sig01=sd(y[clust$cluster==1])
sig02=sd(y[clust$cluster==2])
if(is.na(sig01)){sig01=0};
if(is.na(sig02)){sig02=0};
if((sig01==0)|(sig02=0)){sig0=max(sig01,sig02)} else
{sig0=(sig01+sig02)/2};
group=2;
mv=lev.p(tW,method="sd");
TX=cbind(tW[,1]-mv$mean[1],tW[,2]-mv$mean[2]);
ind=which(diag(TX%*%solve(mv$covar)%*%t(TX))>=qchisq(0.975,2));
```

```
if(length(ind)>0)
{
tW=tW[-ind,];
y=y[-ind];
}
tW1=tW[,1];
tW2=tW[,2];
n=length(y);
W=cbind(tW1,tW2);
m=2;
maxv=15;
likhod=flag=rep(0,15);
theta=array(0,dim=c(2,5,15));
for(v in seq(maxv))
{
bet=beta
sig=sig0;
pr=prob;
run=0;
n=length(y);
tW=cbind(rep(1,n),W);
lh=-10^10;
a=dim(W);
p=a[2]+1;
pk=u=r=matrix(rep(0,m*n),nrow=n);
for(j in seq(m))
{
```

```
r[,j]=(y-tW%*%bet[j,])/sig;
}
repeat
{
run=run+1;
prelh=lh;
for(j in seq(m))
{
pk[,j]=pr[j]*dent(y, tW%*%bet[j,],sig,v);
u[,j]=(v+1)/(v+r[,j]^2)
}
lh=sum(log(apply(pk,1,sum)));
pk=pk/matrix(rep(apply(pk,1,sum),m),nrow=n);
dif=lh-prelh;
if(!is.finite(lh)){flag[v]=1; break}
if(dif<1e-6|run>500){flag[v]=0; break}
np=apply(pk,2,sum);pr=np/n;
sig2=rep(0,m);
for(j in seq(m))
{
w=diag(pk[,j]*u[,j]);
bet[j,]=ginv(t(tW)%*%w%*%tW)%*%t(tW)%*%w%*%y;
r[,j]=y-tW%*%bet[j,];
sig2[j]=sum(pk[,j]*r[,j]^2*u[,j])/sum(pk[,j])-t(bet[j,-1])%*%Lmd%*%bet[j,-1];
sig2[j]=sig2[j]*(sig2[j]>0.0001)+0.0001*(sig2[j]<=0.0001)
}
sig=sqrt(mean(sig2));
```

```
r=r/sig;
}
sig=sig*rep(1,m);
theta[,,v]= matrix(c(bet,pr,sig),nrow=m);
likhod[v]=lh;
}
if(all(flag==0)){break}
}
fv=which(likhod==max(likhod))
if(fv<15)
{result=theta[,,fv]} else
{
run=0;
n=length(y);
tW=cbind(rep(1,n),W);
r=matrix(rep(0,m*n),nrow=n);
pk=r;
lh=0
for(j in seq(m))
{
r[,j]=y-tW%*%bet[j,];
lh=lh+pr[j]*dnorm(r[,j],0,sig);
}
lh=sum(log(lh));
repeat
{
prest=c(bet,sig,pr);
```

```
run=run+1;
plh=lh;
for(j in seq(m))
{
pk[,j]=pr[j]* pmax(10^(-6),dnorm(r[,j],0,sig))
}
pk=pk/matrix(rep(apply(pk,1,sum),m),nrow=n);
np=apply(pk,2,sum);
pr=np/n;
sig2=rep(0,2);
for(j in seq(m))
{
w=diag(pk[,j]);
bet[j,]=ginv(t(tW)%*%w%*%tW)%*%t(tW)%*%w%*%y;
r[,j]= y-tW%*%bet[j,];
sig2[j]=sum(pk[,j]*r[,j]^2*u[,j])/sum(pk[,j])-t(bet[j,-1])%*%Lmd%*%bet[j,-1];
sig2[j]=sig2[j]*(sig2[j]>0.0001)+0.0001*(sig2[j]<=0.0001)
}
sig=sqrt(mean(sig2));
lh=0;
for(j in seq(m))
{
lh=lh+pr[j]*dnorm(r[,j],0,sig);
}
lh=sum(log(lh));
dif=lh-plh;
if((dif<1e-6)|(run>500)){break}
```

```
}
sig=sig*rep(1,m)
result=matrix(c(bet,pr,sig),nrow=m)
}
btemp=c(result[1,4], result[1,1:3], result[2,4], result[2,1:3]);
btemp=slab(btemp);
prob=c(btemp[1],btemp[5]);
beta=rbind(btemp[2:4],btemp[6:8]);
rest7=cbind(beta,prob)
b107[k]=rest7[1,1]
b117[k]=rest7[1,2]
b127[k]=rest7[1,3]
b207[k]=rest7[2,1]
b217[k]=rest7[2,2]
b227[k]=rest7[2,3]
pi17[k]=rest7[1,4]
cat("k",k,"\n")
}
tb10=1;  tb11=1;   tb12=1;
tb20=-1;  tb21=-1;   tb22=-1;
tpi1=0.25;
RL=cbind(b107-tb10,b117-tb11,b127-tb12,b207-tb20,b217-tb21,b227-tb22,pi17-tpi1);
BV[seq((dist-1)*7+1,dist*7),1:2]=cbind(
   apply(RL,2,function(x){mean(x^2)}),apply(RL,2,mean))
print(BV[seq((dist-1)*7+1,dist*7),1:2])
}
round(BV,3)
```