TLEmix: A General Framework for Robust Fitting of Finite Mixture Models in R

Raja Patnaik, Alexander Eisl, Roland Boubela, and Peter Filzmoser
Vienna University of Technology

Abstract

TLEmix implements a general framework for robustly fitting discrete mixtures of regression models in the R statistical computing environment. It implements the FAST-TLE algorithm and uses the R package FlexMix as a computational engine for fitting mixtures of general linear models (GLMs) and model-based clustering in R.

Keywords: R, finite mixture models, model based clustering, robustness.

1. Introduction

The initial approach to mixture analysis was first undertaken by the biometrician Karl Pearson when he was given a data set by the famous zoologist Walter Frank Raphael Weldon in 1894. In his extensive data analysis Pearson fitted a model consisting of two normal probability density functions with different means (μ_1 and μ_2) and different variances (σ_1^2 and σ_2^2) in proportions π_1 and π_2 to the data. Although he did not use maximum likelihood to fit the model Pearson's estimation involving the method of moments was nearly as accurate (see McLachlan and Peel (2000)).

Since then finite mixture models have found a variety of applications over almost 100 years, but have experienced a significant boost in popularity during the last decade as available computing power has been growing exponentially. In particular, the 1977 published approach of the EM algorithm lead to increasing interest in finite mixture models as it tremendously simplified the maximum likelihood estimation. This paper is therefore organised as follows: First the basic concept of trimmed likelihood estimation as well as the FAST-TLE algorithm implemented by TLEmix are discussed. Section 3 then demonstrates how to use TLEmix to robustly fit finite mixture models.

2. Trimmed Likelihood Estimator

2.1. The Weighted Trimmed Likelihood Estimator

Let the observations $y_1, \ldots, y_n$ denote values generated by an arbitrary probability density function $\varphi(y;\theta)$ with unknown parameter vector $\theta \in \Theta^p \subset \mathbb{R}^p$. Then, according to Vandev and Neykov (1998), the Weighted Trimmed Likelihood Estimator (WTLE) is given by

$$WTLE(k)(y_1, \ldots, y_n) = \hat{\theta}_{WTLE} = \arg\min_{\theta \in \Theta^p} \sum_{i=1}^{k} w_{\nu(i)} f(y_{\nu(i)}; \theta), \tag{1}$$

where $f(y_i;\theta)$ corresponds to $-\log\varphi(y_i;\theta)$ and $f(y_{\nu(1)};\theta) \leq f(y_{\nu(2)};\theta) \leq \ldots \leq f(y_{\nu(n)};\theta)$ are the ordered density values for a fixed θ. The permutation $\nu = (\nu(1),\ldots,\nu(n))$ of the indices $1,\ldots,n$ may depend on θ as well.

Depending on the nondecreasing weight function $w_{\nu(i)}$ with $w_i \geq 0$ for $i = 1,\ldots,n$ (at least $w_{\nu(k)} > 0$), the WTLE can accommodate different estimators. The median likelihood estimator MedLE(k) defined by Neykov and Neytchev (1990)

$$\text{MedLE}(y_1,\ldots,y_n) = \arg\min_{\theta\in\Theta^p} \operatorname*{med}_i(-\ln\varphi(y_i;\theta)), \tag{2}$$

is obtained by $w_{\nu(i)} = 0$ for $i = 1,\ldots,k-1,k+1,\ldots,n$ and $w_{\nu(k)} = 1$, and the TLE

$$\text{TLE}(k)(y_1,\ldots,y_n) = \arg\min_{\theta\in\Theta^p} \sum_{i=1}^{k} \left\{-\ln(\varphi(y;\theta))_{(i)}\right\}, \tag{3}$$

is obtained if $w_{\nu(i)} = 1$ for $i = 1,\ldots,k$ and $w_{\nu(i)} = 0$ otherwise.

From a combinatorial point of view the WTLE yields the minima by subsampling the data:

$$\min_{\theta\in\Theta^p} \sum_{i=1}^{k} \left\{w_{\nu(i)} f(y_{\nu(i)};\theta)\right\} = \min_{\tau} \min_{\theta\in\Theta^p} \sum_{i=1}^{k} \left\{w_i f(y_{\tau(i)};\theta)\right\}, \tag{4}$$

where $\tau = (\tau(1),\ldots,\tau(n))$ denotes any permutation of the indices, so that all k subsets of the set $\{1,\ldots,n\}$ are considered.

2.2. The FAST-TLE Algorithm

Recalling the definition of the WTLE in Eq. (1) it is assumed that minimisation is achieved by subsampling the data. In view of the combinatorial nature of the algorithm, computing the WTLE for large data sets can proceed very slowly. Thus, an approximative solution was developed in Neykov and Müller (2003).

The general idea is to divide the algorithm into two separate, iterative processes. The trial step consists of drawing finitely many random subsamples of size k^*. In the following, for any one of those the ML estimate is computed and the subsamples of size k with the lowest TL values are kept for further processing. This way in every step those sets are selected, which give an improved fit of the model. However, given, a small data set is being processed, all possible subsets with size k can be considered.

Continuing this iterative procedure yields guaranteed convergence of the algorithm as there are only $\binom{n}{k}$ k-subsets in all. To assure that there always exists a solution to the optimisation problem in Eq. (1), it is assumed that the finite set F is d-full and $k \leq d$. Note that in the normal linear regression case the FAST-TLE reduces to the FAST-LTS algorithm.

As for the choice of k, a reasonable value would be $\lfloor(n+d+1)/2\rfloor$ as it would maximise the BP of the TLE algorithm. However, any value between d and n can be chosen for k. The size k^* of the subsamples largely depends on the fullness parameter d and can be defined as $k^* = d+1$ in order to increase the chance of drawing at least one outlier free subsample (see Neykov and Müller (2003)).

3. Using TLEmix

As a simple example we use the artificial data set gaussData included in the library tlemix consisting of 80 observations from a mixture of two normal distributions. In order to analyse the performance of the robust FAST-TLE algorithm 20 outliers were added.

First a model framework has to be created that will be passed to the TLE method as the data set

```
> library(tlemix)
> data(gaussData)
```

We can fit this model in R using the following commands:

```
> est.tle <- TLE(y ~ x, family = "gaussian", data = gaussData,
+     Density = flexmix.Density, Estimate = flexmix.Estimate,
+     msglvl = 1, nc = 2, kTrim = 80, nit = 10)
```

The argument kTrim=80 could be omitted, but then too many data points would be trimmed. Moreover, the solution could be improved by increasing the number of iterations for instance to nit=100. We can get a first look at the estimated parameters of mixture component 1 by

```
> parameters(est.tle@estimate, component = 1)
```

```
                   Comp.1
coef.(Intercept) 1.9712529
coef.x           0.9861301
sigma            0.1091538
```

and

```
> parameters(est.tle@estimate, component = 2)
```

```
                    Comp.2
coef.(Intercept) -1.9632190
coef.x           -0.9709648
sigma             0.1009793
```

for component 2. A cross-tabulation of true classes and cluster memberships can be obtained by

```
> table(gaussData$c, est.tle@tleclusters)
```

```
     1  2
  1  0 40
  2 40  0
  3 15  5
```

The summary method

```
> summary(est.tle)

Call:
TLE(formula = y ~ x, family = "gaussian", data = gaussData,
    kTrim = 80, nit = 10, msglvl = 1, nc = 2, Density = flexmix.Density,
    Estimate = flexmix.Estimate)



Call:
flexmix(formula = model, data = data.frame(data),
    k = nc, cluster = cluster, model = FLXglm(model,
        family = family), control = control)

Cluster sizes:
 1  2
40 40

convergence after 8 iterations

kTrim: 80    Number of Observations: 100    Number of Outliers: 20
```

gives the trimming parameter, the number of observations, the number of outliers and prints the `estimate` object.

Calling the summary method for the flexmix object `est.tle@estimate`

```
> summary(est.tle@estimate)

Call:
flexmix(formula = model, data = data.frame(data),
    k = nc, cluster = cluster, model = FLXglm(model,
        family = family), control = control)

       prior size post>0 ratio
Comp.1 0.512   40     47 0.851
Comp.2 0.488   40     45 0.889

'log Lik.' 16.78899 (df=7)
AIC: -19.57797   BIC: -2.903785
```

displays the estimated prior probabilities $\hat{\pi}_k$, the number of observations assigned to the corresponding clusters, the number of observations where $p_{nk} > \delta$ (with a default of $\delta = 10^{-4}$), and the ratio of the latter two numbers.

For this example the method `tleplot` can be used to visualise the 2-dimensional data frame. For each cluster identified by the method `TLE` a different colour is used for indication purposes. Outliers are depicted as black triangles (see Figure 1).

```
> tleplot(est.tle, gaussData)
```

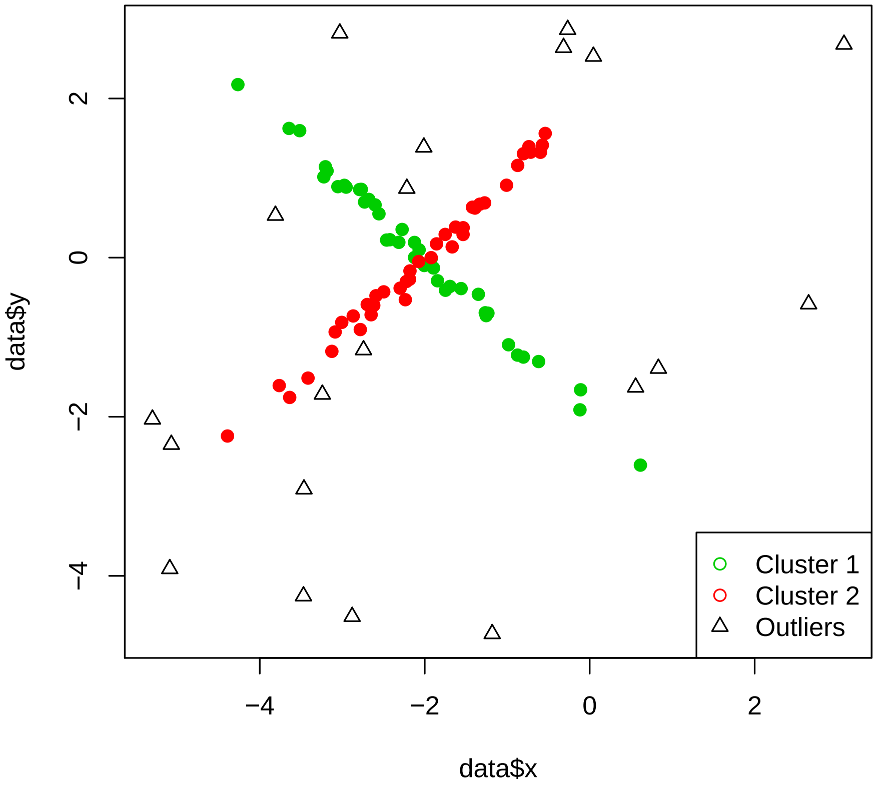

Figure 1: Mixture of two normal components with noise. Scatterplot with cluster memberships.

Additionally, the flexmix object can be plotted by

```
> plot(est.tle@estimate)
```

and will yield rootograms of the posterior class probabilities to visually assess the cluster structure (see Figure 2).

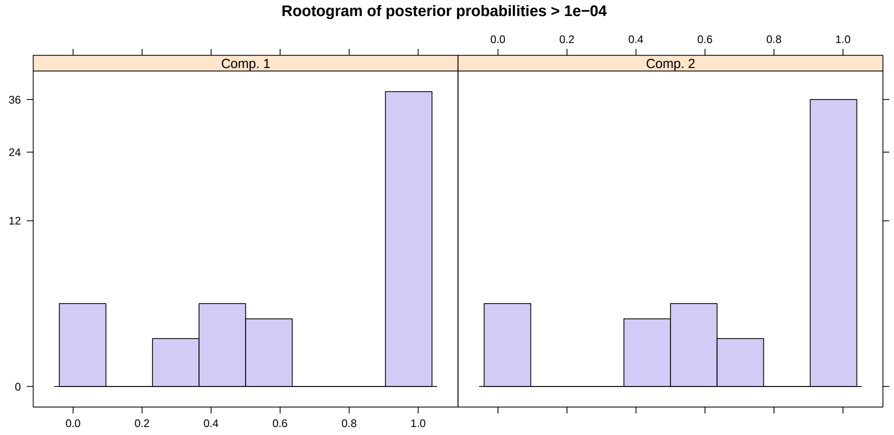

Figure 2: Rootogram of the posterior class probabilites.

Usually in each component a lot of observations have posteriors close to zero, resulting in a high count for the corresponding bin in the rootogram which obscures the information in the other bins. To avoid this problem, all probabilities with a posterior below a threshold are ignored (we again use 10^{-4}). A peak at probability 1 indicates that a mixture component is well seperated from the other components, while no peak at 1 and/or significant mass in the middle of the unit interval indicates overlap with other components.

References

McLachlan G, Peel D (2000). *Finite Mixture Models.* Wiley, New York.

Neykov N, Müller C (2003). "Breakdown Point and Computation of Trimmed Likelihood Estimators in Generalized Linear Models." *J. Statist. Plann. Inference 116*, pp. 503–519.

Neykov N, Neytchev P (1990). "A Robust Alternative of the ML Estimators." *COMPSTAT'90, Short communications*, pp. 99–100.

Vandev D, Neykov N (1998). "About regression estimators with high breakdown point." *Statistics 32*, pp. 111–129.

Affiliation:

Peter Filzmoser
Department of Statistics and Probability Theory
Vienna University of Technology
A-1040 Vienna, Austria, Wiedner Hauptstr. 8-10
E-mail: P.Filzmoser@tuwien.ac.at
URL: http://statistik.tuwien.ac.at/public/filz/

Original publication:

Journal of Statistical Software *Submitted:* yyyy-mm-dd
MMMMMM YYYY, Volume VV, Issue II. *Accepted:* yyyy-mm-dd
http://www.jstatsoft.org/