mixR: An R package for Finite Mixture Modeling for Both Raw and Binned Data

Youjiao Yu

1 Department of Statistical Science, Baylor University

Statement of need

R (R Core Team, 2020) provides a rich collection of packages for building and analyzing finite mixture models, which are widely used in unsupervised learning, such as model-based clustering and density estimation. For example, mclust (Scrucca et al., 2016) can be used to build Gaussian mixture models with different covariance structures, mixtools (Benaglia et al., 2010) implements parametric and non-parametric mixture models as well as mixtures of Gaussian regressions, flexmix (Leisch, 2004) provides a general framework for finite mixtures of regression models, mixdist (Macdonald et al., 2018) fits mixture models for grouped and conditional data (also called binned data). To our knowledge, almost all R packages for finite mixture models are designed to use raw data as the modeling input except mixdist. However, the popular model selection methods based on information criteria or bootstrapping likelihood ratio test (bLRT) (Feng & McCulloch, 1996; McLachlan, 1987; Yu & Harvill, 2019) are not implemented in mixdist. To bridge this gap and to unify the interface for finite mixture modeling for both raw and binned data, we implement mixR package that provides the following primary features.

- `mixfit()` performs maximum likelihood estimation (MLE) for finite mixture models for Gaussian, Weibull, Gamma, and Log-normal distributions via EM algorithm (Dempster et al., 1977). The model fitting is accelerated via package Rcpp (Eddelbuettel et al., 2011).
- `select()` selects the best model from a series of mixture models with a different number of mixture components by using Bayesian Information Criterion (BIC).
- `bs.test()` performs bLRT for two mixture models from the same distribution family but with a different number of components.

mixR also contains the following additional features.

- Visualization of the fitted mixture models using ggplot2 (Wickham, 2011).
- Functions to generate random data from mixture models.
- Functions to convert parameters of Weibull and Gamma mixture models between shape-scale representation used in probability density functions and mean-variance representation which is more intuitive for people to understand the distribution.

Examples

We demonstrate how to use mixR for fitting finite mixture models and selecting mixture models using BIC and bLRT.

Model fitting

We fit the following four mixture models to a data set that consists of 1000 random data points generated from a Weibull mixture model with two components.

- Gaussian mixture with two components (mod1)
- Gaussian mixture with two components to the binned data (mod2)
- Gaussian mixture with three components (mod3)
- Weibull mixture with two components (mod4)

The fitted coefficients in mod1 and mod2 and the top two plots in Figure 1 show that binning does not cause much information loss, and we get similar fitted results using either raw data or binned data. This is usually the case when we have at least moderate data size, and the underlying mixture model is not too complex (e.g., too many mixture components). A benefit of binning is that it reduces the computation burden significantly for large data, especially when conducting bLRT, which is computationally intensive. From Figure 1 we also observe that Gaussian mixture models can provide a good fit for non-Gaussian data though the number of mixture components tends to be overestimated because more Gaussian components are needed to model the asymmetry and long tails that usually exist in non-Gaussian data.

```r
library(mixR)
set.seed(101)
x <- rmixweibull(1000, c(0.4, 0.6), c(0.6, 1.3), c(0.1, 0.1))
x_binned <- bin(x, brks = seq(min(x), max(x), length = 30))
mod1 <- mixfit(x, ncomp = 2)
mod2 <- mixfit(x_binned, ncomp = 2)
mod3 <- mixfit(x, ncomp = 3)
mod4 <- mixfit(x, ncomp = 2, family = 'weibull')
mod1
## Normal mixture model with 2 components
## comp1  comp2
## pi 0.4210604 0.5789396
## mu 0.6014690 1.3084871
## sd 0.1092375 0.0932826
##
## EM iterations: 5 AIC: -406.65 BIC: -382.11 log-likelihood: 208.32

mod2
## Normal mixture model with 2 components
## comp1  comp2
## pi 0.4213019 0.5786981
## mu 0.6018737 1.3091224
## sd 0.1084973 0.0916267
##
## EM iterations: 9 AIC: 5813.09 BIC: 5837.63 log-likelihood: -2901.54

p1 <- plot(mod1, title = 'Gaussian Mixture (2 components)')
p2 <- plot(mod2, title = 'Gaussian Mixture (binned data 2 components)')
p3 <- plot(mod3, title = 'Gaussian Mixture (3 components)')
p4 <- plot(mod4, title = 'Weibull Mixture (2 components)')
gridExtra::grid.arrange(p1, p2, p3, p4, nrow = 2)
```
Model selection

Figure 2 shows that the best Gaussian mixture model selected by BIC has three components and unequal variances for each component, while the best Weibull mixture model has two components. The bLRT with \( H_0 : g = 2 \) versus \( H_a : g = 3 \) for Gaussian mixture models (using the default 100 bootstrap iterations) returns a p-value of zero, showing that Gaussian mixture with three components is significantly better than that with two components. Similarly, the same test for Weibull mixture models returns an insignificant p-value of 0.82, indicating that the Weibull mixture with three components is no better than it with two components.

```r
b1 <- select(x, ncomp = 2:4)
b2 <- select(x, ncomp = 2:4, family = 'weibull')
b3 <- bs.test(x, ncomp = c(2, 3))
b4 <- bs.test(x, ncomp = c(2, 3), family = 'weibull')

b3$pvalue
## [1] 0

b4$pvalue
## [1] 0.82
```

```r
par(mfrow = c(2, 2))
plot(b1)
plot(b2, main = "Weibull Mixture Model Selection by BIC")
plot(b3, main = "Bootstrap LRT for Gaussian Mixture Models\n(g = 2 vs. g = 3)", xlab = 'Bootstrap Test Statistics')
plot(b4, main = "Bootstrap LRT for Weibull Mixture Models\n(g = 2 vs. g = 3)", xlab = 'Bootstrap Test Statistics')
```
Figure 2: (top left) Gaussian mixture model selection using BIC (UV stands for unequal variances for each mixture component and EV stands for equal variance); (top right) Weibull mixture model selection using BIC; (bottom left) bLRT with \(H_0: g = 2\) versus \(H_a: g = 3\) for Gaussian mixture models; (bottom right) bLRT with \(H_0: g = 2\) versus \(H_a: g = 3\) for Weibull mixture models.

Summary

mixR unifies the interface for fitting and comparing finite mixture models for both raw data and binned data for distributions including Gaussian, Weibull, Gamma, and Log-normal. The package also provides features for generating random data from mixture models, conversion of parameters for Weibull and Gamma models, and model visualization in ggplot2. The heavy computation in mixR is completed in C++ using Rcpp.

mixR is actively used by researchers and practitioners in various fields (Buckland et al., 2021; Büchel & Corman, 2021; Jung et al., 2020; Korne et al., 2021; Ogana, 2020; Sylvestre et al., 2020; S.-I. Yang et al., 2021; Z. R. Yang, 2021).

References


