The Journal of Open Source Software

mixComp: An R package for estimating complexity of a mixture

Anja Weigel[2*], **Fadoua Balabdaoui**[1*], **Yulia Kulagina**[1¶], **and Lilian Mueller**[2*]

1 ETH Zurich, Seminar for Statistics, Switzerland **2** ETH Zurich, Switzerland ¶ Corresponding author ***** These authors contributed equally.

DOI: 10.21105/joss.04354

Software

- Review
- Repository
- Archive

Editor: Mehmet Hakan Satman

Reviewers:

- @Athene-ai
- @zhiiiyang

Submitted: 20 April 2022
Published: 26 June 2022

Summary

Mixture models (see Lindsay, 1983a, 1983b; McLachlan & Peel, 2000; Teicher, 1963; Titterington et al., 1985) allow for modeling heterogeneous data. The number of mixture components may be known in advance, in which case the model parameters can be easily estimated (e.g., their maximum likelihood estimates (MLE) can be computed using the EM (Expectation-Maximization) algorithm (Dempster et al., 1977)). However, in many applications the number of components is unknown and has to be inferred from the data.

mixComp provides three categories of methods for estimating the unknown complexity of a (univariate) finite mixture:

- methods built upon the determinants of the Hankel matrix of moments of the mixing distribution;
- methods based on penalized minimum distance between the unknown probability density and its consistent estimator;
- likelihood ratio test (LRT) - based techniques.

All methods come with theoretical guarantees for consistency.

Statement of need

mixComp is aimed at practitioners studying phenomena that can be effectively modelled using mixture distributions. Two main features distinguish it from other mixture-related **R** (R Core Team, 2020) packages:

- while mixture component weights and parameters are often estimated as a by-product, **mixComp** methods are based on theory specifically developed to consistently estimate mixture complexity;
- **mixComp** is applicable to parametric mixtures beyond those whose component distributions are included in the **stats** package, making it more customizable than most packages for model-based clustering.

Other packages dealing with mixture models are **mclust** (Scrucca et al., 2016), which fits Gaussian mixtures using the EM algorithm, **MixSim** (Melnykov et al., 2012), which allows for simulating from mixtures and comparing the performance of clustering algorithms, and **mixdist** (Macdonald & Du, 2018), used for grouped conditional data. **mixtools** (Benaglia et al., 2009) focuses on mixture-of-regressions and non-parametric mixtures and is used to fit (multivariate) normal, multinomial or gamma mixtures with the EM algorithm, also containing routines for selecting the number of components based on information criteria and parametric bootstrapping of the LRT statistic values. However, they are limited to multinomial, normal mixtures and mixtures-of-regressions. **flexmix** (Grün & Leisch, 2007, 2008; Leisch, 2004) handles mixtures-of-regression and stands out due to its extensibility, a design principle that we also aimed for. **rebmix**(Nagode, 2018), dealing with univariate and multivariate finite mixture model for generation, estimation, clustering, classification purposes, offers a wide variety of recognized methods for mixture model estimation for both discrete and continuous variables. The approaches suggested in **mixComp** are however not among those used in **rebmix**, thus complementing it rather than providing competition.

Methods

A distribution F is called a *finite mixture* if its probability density/mass is of the form

$$f(x) = \sum_{i=1}^{p} w_i g_i(x, \theta_i),$$

$p \in \mathbb{N}$ being the mixture complexity, $(w_1, \dots w_p : \sum_{i=1}^{p} w_i = 1,\ w_i \geq 0,$ for $i = 1, \dots, p)$ - component weights and $g_i(x, \theta_i)$ - i-th component density.

Given some complexity j, the relevant parameter spaces are

$$\Theta_j = \{\theta_1, \dots, \theta_j : \theta_i \in \Theta \subseteq \mathbb{R}^d, \quad d \in \mathbb{N}, \quad \text{for } i = 1, \dots, j\}, \text{ and}$$

$$W_j = \{w_1, \dots, w_j : \sum_{i=1}^{j} w_i = 1, w_i \geq 0, \text{ for } i = 1, \dots, j\}.$$

Assume the family of the component densities $\{g(x; \theta)\}$ is known, $\theta = (\theta_1, \dots, \theta_p) \in \Theta_p$, $\mathbf{w} = (w_1, \dots, w_p) \in W_p$ and $p \in \mathbb{N}$ are unknown.

1. Functions using Hankel matrices

The basic Hankel approach (Dacunha-Castelle & Gassiat, 1997) estimates (based on $\mathbf{X} = \{X_1, \dots, X_n\}$, an i.i.d. n-sample from F)

$$\hat{p} := \text{argmin}_{j \in \mathbb{N}} \Big\{ |\det H(\hat{\mathbf{c}}_{2j+1})| + A(j)l(n) \Big\},$$

with positive function $l(n) \to 0$ as $n \to \infty$; positive, strictly increasing function $A(j)$; $H(\hat{\mathbf{c}}_{2j+1})$ - Hankel matrix built on $\hat{\mathbf{c}}_{2j+1}$, the consistent estimator of the first $2j+1$ moments of the mixing distribution.

mixComp offers several methods for calculating $\hat{\mathbf{c}}_{2j+1}$ and provides extensions of the basic approach.

2. Functions using distances

Consider the parametric family

$$\mathcal{F}_j = \{f_{j,\mathbf{w},\theta} : (\mathbf{w}, \theta) \in W_j \times \Theta_j\},$$

$f_{j,\mathbf{w},\theta}(x) = \sum_{i=1}^{j} w_i g(x; \theta_i), \quad \{g(x; \theta) : \theta \in \Theta\}$. Note: $\mathcal{F}_j \subseteq \mathcal{F}_{j+1}, \forall j = 1, 2, \dots$.

These methods search for the ‘best’ estimate (e.g. MLE) $(\hat{\mathbf{w}}^j, \hat{\theta}^j) \in W_j \times \Theta_j$ for a given j and thereby specified density/mass function $\hat{f}_j(x) = f_{j,\hat{\mathbf{w}}^j,\hat{\theta}^j}(x)$, and the non-parametric density/mass estimate $\tilde{f}_n(x)$. Then

$$\hat{p} = \min_j \{D(\hat{f}_j, \tilde{f}_n) - D(\hat{f}_{j+1}, \tilde{f}_n) \leq t(j, n)\},$$

where D denotes the distance measure, $t(j, n)$ - a suitable penalty function.

mixComp offers several distance-based procedures following (Umashanger & Sriram, 2009; Woo & Sriram, 2006; Woo & Sriram, 2007).

3. Functions using LRTS

These methods obtain the MLE for the mixture density/mass with j and $j+1$ components ($j = 1, 2, \dots$), yielding $(\hat{\mathbf{w}}^j, \hat{\theta}^j) \in W_j \times \Theta_j$ and $(\hat{\mathbf{w}}^{j+1}, \hat{\theta}^{j+1}) \in W_{j+1} \times \Theta_{j+1}$,

$$\text{LRTS} = -2 \ln \left(\frac{L_{\mathbf{X}}(\hat{\mathbf{w}}^j, \hat{\theta}^j)}{L_{\mathbf{X}}(\hat{\mathbf{w}}^{j+1}, \hat{\theta}^{j+1})} \right), \text{ with}$$

$L_{\mathbf{X}}$ being the likelihood function given $\mathbf{X}$.

A parametric bootstrap sampling is applied to generate `B` n-samples from a j-component mixture given $(\hat{\mathbf{w}}^j, \hat{\theta}^j)$. For each bootstrap sample, the method computes the MLEs and the LRTS (Likelihood Ratio Test Statistic) corresponding to the mixture densities with j and $j+1$ components. The decision is to reject $H_0 : p = j$, setting $j \leftarrow j+1$ if the LRTS is larger than the specified quantile of its bootstrapped counterparts; otherwise $\hat{p}$ is set to j (Xekalaki & Karlis, 1999).

Examples

Hellinger distance method with bootstrap applied to the Shakespeare data (viewed as a mixture of geometrics) (Balabdaoui & Kulagina, 2020; Chee & Wang, 2016; Efron & Thisted, 1976; Spevack, 1968).

```r
# apply the shift:
shakespeare.obs <- unlist(shakespeare) - 1
# define the MLE function:
MLE.geom <- function(dat) 1 / (mean(dat) + 1)
# create the datMix object:
Shakespeare.dM <- datMix(shakespeare.obs, dist = ”geom”, discrete = TRUE,
             MLE.function = MLE.geom, theta.bound.list = list(prob = c(0, 1)))
# estimate the complexity:
set.seed(0)
(res <- hellinger.boot.disc(Shakespeare.dM, B = 50, ql = 0.025, qu = 0.975))
> The estimated order is 3.
# plot:
plot(res, breaks = 100, xlim = c(0, 20))
```

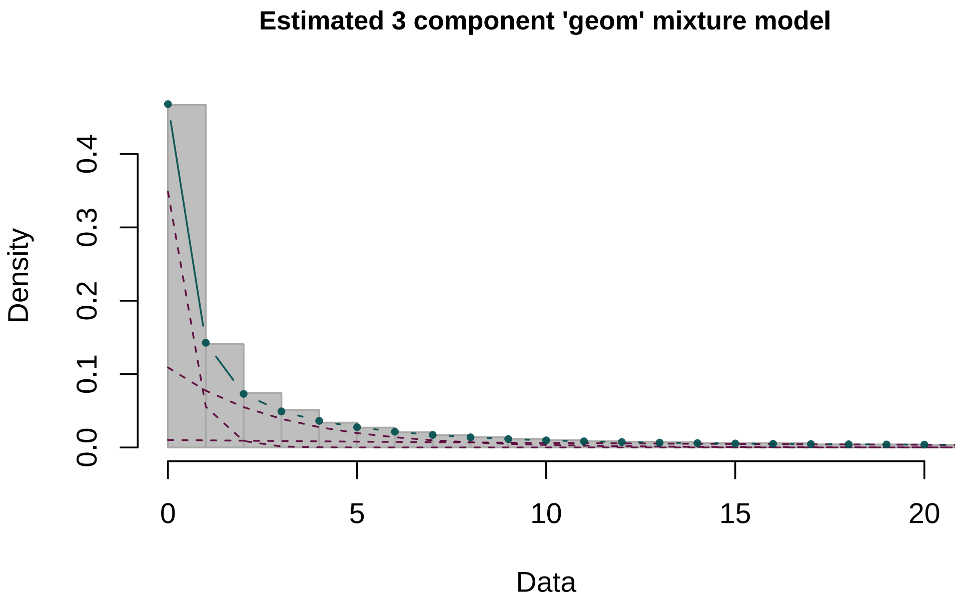

Figure 1: Hellinger distance method with bootstrap for the Shakespeare data

LRT method applied to the Acidity data (Crawford et al., 1992; Crawford, 1994; Richardson & Green, 1997).

```
#
# define the MLE functions:
MLE.norm.mean <- function(dat) mean(dat)
MLE.norm.sd <- function(dat){
  sqrt((length(dat) - 1) / length(dat)) * sd(dat)
}
MLE.norm.list <- list("MLE.norm.mean" = MLE.norm.mean,
                      "MLE.norm.sd" = MLE.norm.sd)
# define parameter bounds:
norm.bound.list <- list("mean" = c(-Inf, Inf), "sd" = c(0, Inf))

acidity.obs <- unlist(acidity)
# create the datMix object:
acidity.dM <- datMix(acidity.obs, dist = "norm", discrete = FALSE,
                     MLE.function = MLE.norm.list, theta.bound.list = norm.bound.list)
# estimate the complexity:
set.seed(0)
res <- mix.lrt(acidity.dM, B = 100, quantile = 0.95)
```

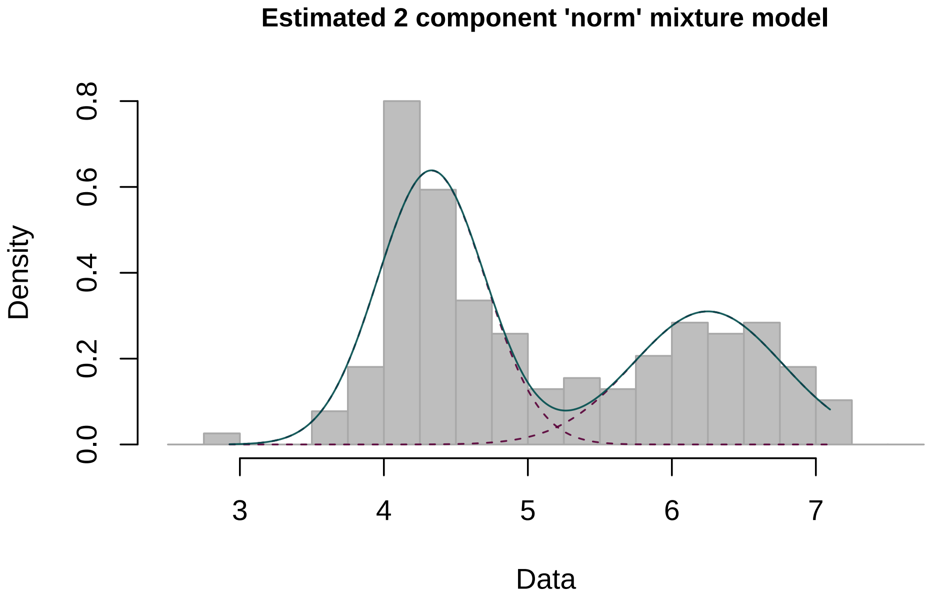

Figure 2: LRT method for the Acidity data

References

Balabdaoui, F., & Kulagina, Y. (2020). Completely monotone distributions: Mixing, approximation and estimation of number of species. *Computational Statistics and Data Analysis*, *150*, 107014, 26. https://doi.org/10.1016/j.csda.2020.107014

Benaglia, T., Chauveau, D., Hunter, D. R., & Young, D. (2009). Mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software*, *32*(6), 1–29. http://www.jstatsoft.org/v32/i06/

Chee, C.-S., & Wang, Y. (2016). Nonparametric estimation of species richness using discrete k-monotone distributions. *Computational Statistics and Data Analysis*, *93*, 107–118. https://doi.org/10.1016/j.csda.2014.10.021

Crawford, S. L. (1994). An application of the laplace method to finite mixture distributions. *Journal of the American Statistical Association*, *89*(425), 259–267. https://doi.org/10.1080/01621459.1994.10476467

Crawford, S. L., DeGroot, M. H., Kadane, J. B., & Small, M. J. (1992). Modeling lake-chemistry distributions: Approximate bayesian methods for estimating a finite-mixture model. *Technometrics*, *34*(4), 441–453. https://doi.org/10.1080/00401706.1992.10484955

Dacunha-Castelle, D., & Gassiat, E. (1997). The estimation of the order of a mixture model. *Bernoulli*, *3*(3), 279–299. https://doi.org/10.2307/3318593

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, *39*(1), 1–38. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

Efron, B., & Thisted, R. A. (1976). Estimating the number of unseen species: How many words did Shakespeare know? *Biometrka*, *63*, 435–447. https://doi.org/10.1007/978-0-387-75692-9_5

Grün, B., & Leisch, F. (2007). Fitting finite mixtures of generalized linear regressions in R. *Computational Statistics and Data Analysis*, *51*(11), 5247–5252. https://doi.org/10.1016/j.csda.2006.08.014

Grün, B., & Leisch, F. (2008). FlexMix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, *28*(4), 1–35. https://doi.org/10.18637/jss.v028.i04

Leisch, F. (2004). FlexMix: A general framework for finite mixture models and latent class regression in R. *Journal of Statistical Software*, *11*(8), 1–18. https://doi.org/10.18637/jss.v011.i08

Lindsay, B. G. (1983a). The geometry of mixture likelihoods: A general theory. *The Annals of Statistics*, *11*(1), 86–94. https://doi.org/10.1214/aos/1176346059

Lindsay, B. G. (1983b). The geometry of mixture likelihoods: The exponential family. *The Annals of Statistics*, *11*(3), 783–792. https://doi.org/10.1214/aos/1176346245

Macdonald, P., & Du, J. (2018). *Mixdist: Finite mixture distribution models*. https://CRAN.R-project.org/package=mixdist

McLachlan, G., & Peel, D. (2000). *Finite mixture models* (p. xxii+419). John Wiley; Sons. https://doi.org/10.1002/0471721182

Melnykov, V., Chen, W.-C., & Maitra, R. (2012). MixSim: An r package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, *51*(12), 1–25. https://doi.org/10.18637/jss.v051.i12

Nagode, M. (2018). Multivariate normal mixture modeling, clustering and classification with the rebmix package. *arXiv Preprint arXiv:1801.08788*.

R Core Team. (2020). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Richardson, S., & Green, P. J. (1997). On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society B*, *59*(4), 731–792. https://doi.org/10.1111/1467-9868.00095

Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). Mclust 5: Clustering, classification and density estimation using gaussian finite mixture models. *The R Journal*, *8*(1), 289–317. https://doi.org/10.32614/rj-2016-021

Spevack, M. (1968). *A complete and systematic concordance to the works of shakespeare*.

Teicher, H. (1963). Identifiability of finite mixtures. *The Annals of Mathematical Statistics*, *34*, 1265–1269. https://doi.org/10.1214/aoms/1177703862

Titterington, D. M., Smith, A. F. M., & Makov, U. E. (1985). *Statistical analysis of finite mixture distributions* (p. x+243). John Wiley; Sons. https://doi.org/10.2307/2531224

Umashanger, T., & Sriram, T. N. (2009). L2E estimation of mixture complexity for count data. *Computational Statistics and Data Analysis*, *53*(12), 4243–4254. https://doi.org/10.1016/j.csda.2009.05.013

Woo, M.-J., & Sriram, T. N. (2006). Robust estimation of mixture complexity. *Journal of the American Statistical Association*, *101*(476), 1475–1486. https://doi.org/10.1198/016214506000000555

Woo, M.-J., & Sriram, T. N. (2007). Robust estimation of mixture complexity for count data. *Computational Statistics and Data Analysis*, *51*(9), 4379–4392. https://doi.org/10.1016/j.csda.2006.06.006

Xekalaki, E., & Karlis, D. (1999). On testing for the number of components in a mixed poisson model. *The Annals of the Institute of Statistical Mathematics*, *51*, 149–162. https://doi.org/10.1023/A:1003839420071