---

title: 'pivmet: an `R` package proposing pivotal methods for consensus clustering and mixture modelling' tags:

- R
- statistics
- consensus clustering
- mixture models authors:
- name: Leonardo Egidi orcid: 0000-0003-3211-905X corresponding: yes equal-contrib: yes affiliation: 1
- name: Roberta Pappad&#xFFFD; equal-contrib: yes affiliation: 1
- name: Francesco Pauli affiliation: 1
- name: Nicola Torelli affiliation: 1 affiliations:
- name: Department of Economics, Business, Mathematics, and Statistics *Bruno de Finetti*, University of Trieste index: 1 date: 6 February 2024 bibliography: paper.bib aas-doi: null aas-journal: null

---

Summary

We introduce the `R` package `pivmet`, a software that performs different pivotal methods for identifying, extracting, and using the so-called pivotal units of a dataset that are chosen to represent the groups of data points to which they belong. These algorithms turn out be very useful in many unsupervised and supervised learning frameworks such as clustering, classification and mixture modelling.

More specifically, applications of pivotal methods could cover, among the others: a Markov-Chain Monte Carlo (MCMC) relabelling procedure to deal with the well-known label-switching problem [@stephens2000dealing; @richardson1997bayesian; @fruhwirth2001markov; @egidi2018relabelling] occurring during Bayesian estimation of mixture models; model-based clustering through sparse finite mixture models (SFMM) [@malsiner2016model; @fruhwirth2019here]; consensus clustering [@JMLR02], which may allow to improve classical clustering techniques---e.g. the classical \(k\)-means---via a careful seeding; and Dirichlet process mixture models (DPMM) [@ferguson1973bayesian; @escobar1995bayesian; @neal2000markov] in Bayesian nonparametrics.

Installation

The stable version of the package can be installed from the Comprehensive R Archive Network (CRAN) (http://CRAN.R-project.org/package=pivmet):

```
install.packages("pivmet")
library(pivmet)
```

Statement of need

In the modern *big-data* and *machine learning* age, summarizing some essential information from a data pattern is often relevant and can help simplifying the data pre-processing steps. The advantage of identifying representative units of a group---hereafter *pivotal units* or *pivots*---somehow chosen to be as far as possible from units in the other groups and as similar as possible to the units in the same group is that they may convey relevant information about the group they belong to while saving wasteful operations.

Despite the lack of a strict theoretical framework behind their characterization, the pivots may be beneficial in many machine learning frameworks, such as clustering, classification, and mixture modelling to derive reliable estimates and/or a better grouping partition.

A deep and theoretical detail around the package's supported pivotal methods is provided in [@egidi2018relabelling].

The `pivmet` package [@pivmet] for `R` , available from the Comprehensive `R` Archive Network (CRAN) at http://CRAN.R-project.org/package=pivmet (http://CRAN.R-project.org/package=pivmet)„ implements various pivotal selection criteria to deal with, but not limited to: (i) mixture model Bayesian estimation---either via the JAGS software [@rjags] using Gibbs sampling or the Stan [@rstan] software performing Hamiltonian Monte Carlo (HMC)---to tackle the so-called *label switching* problem; (ii) consensus clustering, where a variant of the \(k\)-means algorithm is available; (iii) Dirichlet Process Mixture Models (DPPM).

Overview and main functions

The package architecture strongly relies on three main functons:

- The function `piv_MCMC()` is used to fit a Bayesian Gaussian mixture model with underlying Gibbs sampling or Hamiltonian Monte Carlo algorithm. The user can specify distinct prior distributions with the argument `priors` and the selected pivotal criterion via the argument `piv.criterion`.

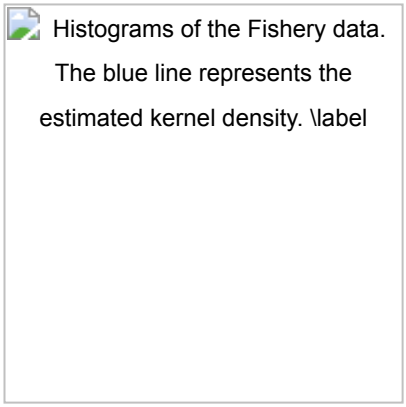
- The function `piv_rel()` takes in input the model fit returned by `piv_MCMC` and implements the relabelling step as outlined by [@egidi2018relabelling].

- The function `piv_KMeans()` performs a robust consensus clstering based on distinct \(k\)-means partitions. The user can specify some options, such as the number of consensus partitions.

Example 1: relabelling for label switching

The Fishery dataset in the `bayesmix` [@bayesmix] package has been previously used by @titterington1985statistical and @papastamoulis2016label. It consists of 256 snapper length measurements---see left plot of Figure \autoref for the data histogram, along with an estimated kernel density. Analogously to some previous works, we assume a Gaussian mixture model with \(k=5\) groups, where \(\mu_j\), \(\sigma_j\) and \(\eta_j\) are the mean, the standard deviation and the weight of group \(j\), respectively. We fit our model by simulating \(15000\) samples from the posterior distribution of \((\mathbf{z}, \mathbf{\mu}, \mathbf{\sigma}, \mathbf{\eta})\), by selecting the default argument `software="rjags"`; for

univariate mixtures, the MCMC Gibbs sampling is returned by the function `JAGSrun` in the package `bayesmix`. Alternatively, one could fit the model according to HMC sampling and with underlying Stan ecosystem by typing `software="rstan"`. By default, the burn-in period is set equal to half of the total number of MCMC iterations.


Histograms of the Fishery data. The blue line represents the estimated kernel density. \label


Fishery dataset: traceplots of the parameters <span class=

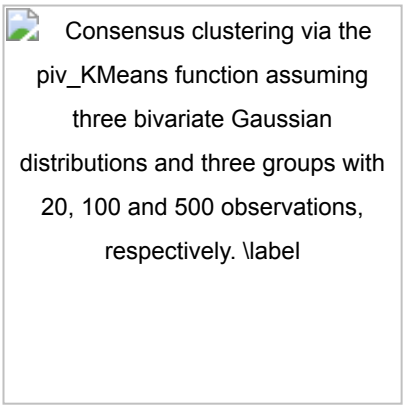
\((\mathbf{\mu}, \mathbf{\sigma}, \mathbf{\eta})\) obtained via the rjags option for the piv_MCMC function (Gibbs sampling, 15000 MCMC iterations). Top row: Raw MCMC outputs. Bottom row: relabelled MCMC samples. \label" />

Figure \autoref displays the traceplots for the parameters \((\mathbf{\mu}, \mathbf{\sigma}, \mathbf{\eta})\). From the first row showing the raw MCMC outputs as given by the Gibbs sampling, we note that label switching clearly occurred. Our algorithm is able to fix label-switching and reorder the means \(\mu_j\) and the weights \(\eta_j\), for \(j=1,\ldots,k\), as emerged from the second row of the plot.

Example 2: consensus clustering

As widely known, one of the drawbacks of the \(k\)-means algorithm is represented by its inefficiency in distinguishing between groups of unbalanced sizes. For thesere reasos, the clustering scientific literature claims that a better robust clustering solution is usually obtained if more partitions are obtained, in such a way the final partition works as a sort of *consensus*. We perform here a consensus clustering technique based on single \(k\)-means consifurations, where each of these has been obtained through a careful initial pivotal seeding.

For illustration purposes, we simulate three bivariate Gaussian distributions with 20, 100 and 500 observations, respectively---see Figure \autoref. The plots with titles 'piv KMeans' refer to the pivotal criteria `MUS`, (i) or `maxsumint`, (ii) or `maxsumdiff`, where the labels 1, 2, and 4 follow the order used in the `R` function; moreover, we consider Partitioning Around Medoids (PAM) method via the `pam` function of the `cluster` package and agglomerative hierarchical clustering (agnes), with average, single, and complete linkage. The partitions from the classical \(k\)-means are obtained using multiple random seeds. Group centers and pivots are marked via asterisks and triangles symbols, respectively. As we may notice, pivotal \(k\)-means methods are able to satisfactorily detect the true data partition.



Consensus clustering via the piv_KMeans function assuming three bivariate Gaussian distributions and three groups with 20, 100 and 500 observations, respectively. \label

Conclusion

The `pivmet` package proposes various methods for identifying pivotal units in datasets with a grouping structure and using them for improving inferential conclusions and clustering partitions. The package suits well for both supervised and unsupervised problems, by providing a valid alternative to existing functions for similar applications, and keeping low the computational effort. It is of future interest to include additional aspects in the software, such as the estimation of the number of components in the data when this information is latent/unknown and provide more graphical tools to diagnose pivotal selection.

Reproducibility

The `R` code required to generate the examples is available at https://github.com/LeoEgidi/pivmet/tree/master/paper/rcode (https://github.com/LeoEgidi/pivmet/tree/master/paper/rcode).

Acknowledgements

We want to thank Ioannis Ntzoufras and Dimitris Karlis from Athens University of Economics and Business (AUEB) for their valuable suggestions about the package structure.

References