

Empirical and non-parametric copula models with the `cort` R package

Oskar Laverny^{*1, 2}

1 Institut Camille Jordan, Université Lyon 1, Lyon, France 2 SCOR SE, Paris, France

DOI: [10.21105/joss.02653](https://doi.org/10.21105/joss.02653)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Pierre de Buyl](#) ↗

Reviewers:

- [@coatless](#)
- [@agisga](#)

Submitted: 02 September 2020

Published: 04 December 2020

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

The R package `cort` implements object-oriented classes and methods to estimate, simulate and visualize certain types of non-parametric copulas.

Copulas are functions that describe dependence structure of a given dataset, or of a given multivariate random event, without describing the univariate events, the marginals. In statistics, it is sometimes useful to separate the marginal distributions (inflation of the money, mortality of the population) from the dependence structure between them, since estimating everything separately is usually easier. Copulas are broadly used in finance, actuarial science, geostatistics, biostatistics, and many other fields, when dealing with dependence.

Copulas are distribution functions on the unit hypercube that have uniform margins (what we call the ‘copula constraints’), and hence this package can be classified in ‘density estimation software.’ Although the estimation of copulas is a widely-treated subject, most performing estimators available in the literature are based on restricted, parametric estimation: vine copulas ([Nagler & Czado, 2016](#)) and graphical models ([Li et al., 2019](#)) for example are potential solutions but under restrictive assumptions. Classical density estimators such as kernels or wavelets do not satisfy marginal copula constraints. There also exist several tree-structured piece wise constant density estimators, but they do not always lead to proper copulas when applied on pseudo-observations or true copula samples. The new models that are implemented in this package try to solve these issues.

We note that a lot of tools are available in R for copula modeling through the excellent package `copula` ([Hofert et al., 2020](#); [Ivan Kojadinovic & Jun Yan, 2010](#); [Jun Yan, 2007](#); [Marius Hofert & Martin Mächler, 2011](#)). Most of these tools however focus on parametric estimation. We start to bridge the gap by providing some tools for non-parametric estimation.

Statement of need

The Copula recursive tree, or Cort, designed by ([Laverny et al., 2020](#)) is a flexible, consistent, piece wise linear estimator for a copula ([Sklar, 1959](#)), leveraging the patchwork copula formalization ([Durante et al., 2015](#)) and a specific piece wise constant density estimator, the density estimation tree ([Ram & Gray, 2011](#)). While the patchwork structure imposes the grid, this estimator is data-driven and constructs the grid recursively from the data, minimizing a chosen distance on the copula space. Furthermore, while the addition of the copula constraints makes the available solutions for density estimation unusable, our estimator is only concerned with dependence and guarantees the uniformity of margins. The R package `cort` provides a useful implementation of this model and several potential refinements, allowing for fast computations of Cort trees, and parallel computations of Cort forests.

*Corresponding author

The main feature implemented in the package is the Cort algorithm, a non-parametric, piece wise constant, copula density estimator. The implementation is recursive and hence quite efficient. The `cort` package is a statistical package that allows to estimate several non-parametric copula models in R. Although the state of the art `copula` package has functions to estimate the empirical copula, we provide a structured set of S4 classes that allows estimation of empirical copulas, checkerboard copulas, Cort copula and bagging of all of these. A specific class exists for bagging Cort models, which implementation runs in parallel, to fasten the computations, using the `future` package (Bengtsson, 2020). Most of the underlying machinery and computations are written in C++, through the `Rcpp` (Eddelbuettel, 2013; Eddelbuettel & Balamuta, 2017; Eddelbuettel & François, 2011; Eddelbuettel & Sanderson, 2014) package.

The `cort` package was designed to be used by statisticians who need a non-parametric view of the dependence structure of a given dataset. It features a rich and extensive API to call the statistical fitting procedures, plotting functions and tools to assess the quality of the fits, while complying with the R standards. Examples datasets are included in the package, and the many vignettes give examples of use cases. The package is available on the Comprehensive R Archive Network (CRAN).

Acknowledgements

We acknowledge contributions from Véronique Maume-deschamps, Didier Rullière and Esterina Masiello, who all gave meaningful insights about performances of the code. We are thankful to the reviewers for the fruitful review of this article and of the package code.

References

- Bengtsson, H. (2020). *A unifying framework for parallel and distributed processing in r using futures*. <https://arxiv.org/abs/2008.00553>
- Durante, F., Fernández-Sánchez, J., Quesada-Molina, J. J., & Ubeda-Flores, M. (2015). Convergence results for patchwork copulas. *European Journal of Operational Research*, 247(2), 525–531. <https://doi.org/10.1016/j.ejor.2015.06.028>
- Eddelbuettel, D. (2013). *Seamless r and c++ integration with Rcpp*. Springer. <https://doi.org/10.1007/978-1-4614-6868-4>
- Eddelbuettel, D., & Balamuta, J. J. (2017). Extending R with C++: A Brief Introduction to `extitRcpp`. *PeerJ Preprints*, 5, e3188v1. <https://doi.org/10.7287/peerj.preprints.3188v1>
- Eddelbuettel, D., & François, R. (2011). `Rcpp`: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8), 1–18. <https://doi.org/10.18637/jss.v040.i08>
- Eddelbuettel, D., & Sanderson, C. (2014). `RcppArmadillo`: Accelerating r with high-performance c++ linear algebra. *Computational Statistics and Data Analysis*, 71, 1054–1063. <https://doi.org/10.1016/j.csda.2013.02.005>
- Hofert, M., Kojadinovic, I., Maechler, M., & Yan, J. (2020). *Copula: Multivariate dependence with copulas*. <https://CRAN.R-project.org/package=copula>
- Ivan Kojadinovic, & Jun Yan. (2010). Modeling multivariate distributions with continuous margins using the copula R package. *Journal of Statistical Software*, 34(9), 1–20. <https://doi.org/10.18637/jss.v034.i09>
- Jun Yan. (2007). Enjoy the joy of copulas: With a package copula. *Journal of Statistical Software*, 21(4), 1–21. <https://doi.org/10.18637/jss.v021.i04>

- Laverny, O., Maume-Deschamps, V., Masiello, E., & Rullière, D. (2020). *Dependence structure estimation using copula recursive trees*. <https://arxiv.org/abs/2005.02912>
- Li, Y., Liu, X., & Liu, F. (2019). *PANDA: AdaPtive noisy data augmentation for regularization of undirected graphical models*. <https://arxiv.org/abs/1810.04851>
- Marius Hofert, & Martin Mächler. (2011). Nested archimedean copulas meet R: The nacopula package. *Journal of Statistical Software*, 39(9), 1–20. <https://doi.org/10.18637/jss.v039.i09>
- Nagler, T., & Czado, C. (2016). Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas. *Journal of Multivariate Analysis*, 151, 69–89. <https://doi.org/10.1016/j.jmva.2016.07.003>
- Ram, P., & Gray, A. G. (2011). Density estimation trees. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 627–635. <https://doi.org/10.1145/2020408.2020507>
- Sklar, A. (1959). Fonction de répartition dont les marges sont données. *Inst. Stat. Univ. Paris*, 8, 229–231.