The Journal of Open Source Software

latentcor: An R Package for estimating latent correlations from mixed data types

Mingze Huang[1, 2], **Christian L. Müller**[3, 4, 5], **and Irina Gaynanova**[1]

1 Department of Statistics, Texas A&M University, College Station, TX **2** Department of Economics, Texas A&M University, College Station, TX **3** Ludwig-Maximilians-Universität München, Germany **4** Helmholtz Zentrum München, Germany **5** Flatiron Institute, New York

DOI: 10.21105/joss.03634

Software

- Review
- Repository
- Archive

Editor: Chris Vernon

Reviewers:

- @corybrunson
- @rmflight

Submitted: 11 August 2021
Published: 21 September 2021

Summary

We present `latentcor`, an R package for correlation estimation from data with mixed variable types. Mixed variables types, including continuous, binary, ordinal, zero-inflated, or truncated data are routinely collected in many areas of science. Accurate estimation of correlations among such variables is often the first critical step in statistical analysis workflows. Pearson correlation as the default choice is not well suited for mixed data types as the underlying normality assumption is violated. The concept of semi-parametric latent Gaussian copula models, on the other hand, provides a unifying way to estimate correlations between mixed data types. The R package `latentcor` comprises a comprehensive list of these models, enabling the estimation of correlations between any of continuous/binary/ternary/zero-inflated (truncated) variable types. The underlying implementation takes advantage of a fast multilinear interpolation scheme with an efficient choice of interpolation grid points, thus giving the package a small memory footprint without compromising estimation accuracy. This makes latent correlation estimation readily available for modern high-throughput data analysis.

Statement of need

No R software package is currently available that allows accurate and fast correlation estimation from mixed variable data in a unifying manner. The popular `cor` function within R package `stats` (Team & others, 2013), for instance, allows to compute Pearson's correlation, Kendall's τ and Spearman's ρ, and a faster algorithm for calculating Kendall's τ is implemented in the R package `pcaPP` (Croux et al., 2013). Pearson's correlation is not appropriate for skewed or ordinal data, and its use leads to invalid inference in those cases. While the rank-based Kendall's τ and Spearman's ρ are more robust measures of *association*, they cannot directly be used as subsitutes for statistical methods that require Pearson correlation as input (a prominent example is, e.g., graphical model estimation (Xue & Zou, 2012; Yoon et al., 2019)). The R package `polycor` (Fox, 2019) is designed for ordinal data and allows to computes polychoric (ordinal/ordinal) and polyserial (ordinal/continuous) correlations based on the latent Gaussian model. However, the package does not have functionality for zero-inflated data, nor can it handle skewed continuous measurements as it does not allow for copula transformation. The R package `correlation` (Makowski et al., 2020) in the `easystats` collection provides 16 different correlation measures, including polychoric and polyserial correlations. However, functionality for correlation estimation from zero-inflated data is lacking. The R package `mixedCCA` (Yoon et al., 2020) is based on the latent Gaussian copula model and can compute latent correlations between continuous/binary/zero-inflated variable types as an intermediate step for canonical correlation analysis. However, `mixedCCA` does not allow for ordinal data types. The R package `latentcor`, introduced here, thus represents

the first stand-alone R package for computation of latent correlation that takes into account all variable types (continuous/binary/ordinal/zero-inflated), comes with an optimized memory footprint, and is computationally efficient, essentially making latent correlation estimation almost as fast as rank-based correlation estimation.

Estimation of latent correlations

The general estimation workflow

The estimation of latent correlations consists of three steps:

- computing Kendall's τ between each pair of variables,
- choosing the bridge function $F(\cdot)$ based on the types of variable pairs; the bridge function connects the Kendall's τ computed from the data, $\widehat{\tau}$, to the true underlying correlation ρ via moment equation $\mathbb{E}(\widehat{\tau}) = F(\rho)$;
- estimating latent correlation by calculating $F^{-1}(\widehat{\tau})$.

We summarize the references for the explicit form of $F(\cdot)$ for each variable combination as implemented in `latentcor` below.

Type	continuous	binary	ternary	zero-inflated (truncated)
continuous	Liu et al. (2009)	-	-	-
binary	Fan et al. (2017)	Fan et al. (2017)	-	-
ternary	Quan et al. (2018)	Quan et al. (2018)	Quan et al. (2018)	-
zero-inflated (truncated)	Yoon et al. (2020)	Yoon et al. (2020)	See `latentcor` vignette for derivation	Yoon et al. (2020)

Efficient inversion of the bridge function

In `latentcor`, the inversion of the bridge function $F(\cdot)$ can be computed in two ways. The original approach (`method = "original"`) relies on numerical inversion for each pair of variables based on uni-root optimization (Yoon et al., 2020). Since each pair of variables requires a separate optimization run, the original approach is computationally expensive when the number of variables is large. The second approach to invert $F(\cdot)$ is through fast multilinear interpolation of pre-calculated F^{-1} values at specific sets of interpolation grid points (`method = "approx"`). This construction has been proposed in (Yoon et al., 2021) and is available for continuous/binary/truncated pairs in the current version of `mixedCCA`. However, that implementation lacks the ternary variable case and relies on an interpolation grid with a large memory footprint. `latentcor` includes the ternary case and provides an optimized interpolation grid by redefining the bridge functions on a rescaled version of Kendall's τ. Here, the scaling adapts to the smoothness of the underlying type of variables by simultaneously controlling the approximation error at the same or lower level. As a result, `latentcor` has significantly smaller memory footprint (see Table below) and smaller approximation error compared to `mixedCCA`.

Memory footprints (in KB):

case	mixedCCA	latentcor
binary/continuous	10.08	4.22
binary/binary	303.04	69.1
truncated/continuous	20.99	6.16
truncated/binary	907.95	92.25
truncated/truncated	687.68	84.33
ternary/continuous	-	125.83
ternary/binary	-	728.3
ternary/truncated	-	860.9
ternary/ternary	-	950.61

Illustrative examples

To illustrate the excellent performance of latent correlation estimation on mixed data, we consider the simple example of estimating correlations between continuous and ternary variables.

First, we use `latentcor` to generate synthetic data with two variables of sample size 500, and true latent correlation value of 0.5. We then estimate the correlation using the original method, the approximation method (default), and standard Pearson correlation.

```
library(latentcor)

# The first variable is ternary
# The second variable is continuous.
# No copula transformation is applied.
set.seed(2346)
X = gen_data(n = 500, types = c("ter", "con"), rhos = 0.5)$X
# Estimate correlations
latentcor(X = X, types = c("ter", "con"), method = "original")$R
latentcor(X = X, types = c("ter", "con"))$R
cor(X)
```

The original method estimates the latent correlation to be equal to 0.4766 (and the approximation method is very close with the value 0.4762). By contrast, applying Pearson correlation gives an estimate of 0.4224, which is further from the true value 0.5.

To illustrate the bias induced by Pearson correlation estimation, we consider the truncated/continuous case for different values of the true correlation. Figure 1A displays the values obtained by using standard Pearson correlation, revealing a significant estimation bias with respect to the true correlations. Figure 1B displays the estimated latent correlations using the original approach versus the true values of the underlying ternary/continuous correlations. The alignment of points around $y = x$ line confirms that the estimation is empirically unbiased. Figure 1C displays the estimated latent correlations using the approximation approach (`method = "approx"`) versus true values of underlying latent correlation. The results are almost indistinguishable from Figure 1B at a fraction of the computational cost.

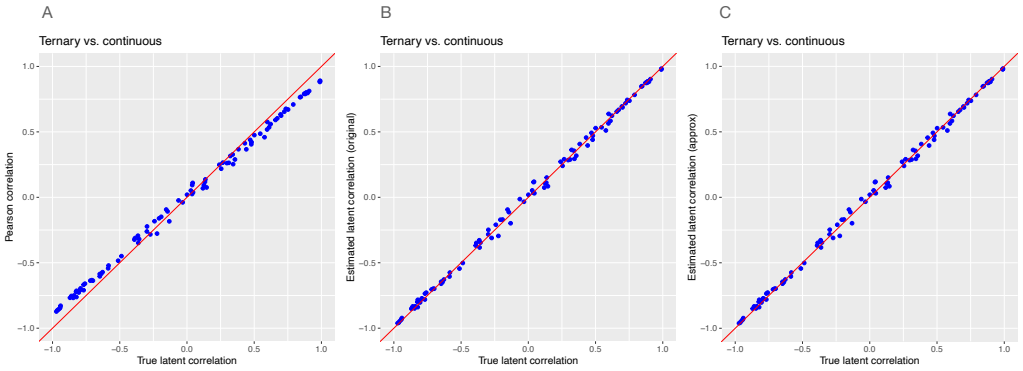

Figure 1: Scatter plots of estimated Pearson correlation (panel A) and latent correlations (`original` in panel B, `approx` in panel C) vs. ground truth correlations

The script to reproduce the displayed results is available at latentcor_evaluation.

We next illustrate an application of `latentcor` to the `mtcars` dataset, available in standard R. The `mtcars` dataset comprises eleven variables of continuous, binary, and ternary data type. The function `get_types` can be used to automatically extract these types from the data. After the types are determined, the correlation matrix can be estimated using either the original method or the approximation method.

```
library(latentcor)
X = mtcars
# Extract variable types
type = get_types(X)
# Estimate correlations
latentcor(mtcars, types = types, method = "original")$R
latentcor(mtcars, types = types)$R
```

Figure 2 shows the 11×11 matrices with latent correlation estimates (with default `approx` method, left panel), Pearson correlation estimates (middle panel), and their difference in estimation (right panel). Even on this small dataset, we observe absolute differences exceeding 0.2.

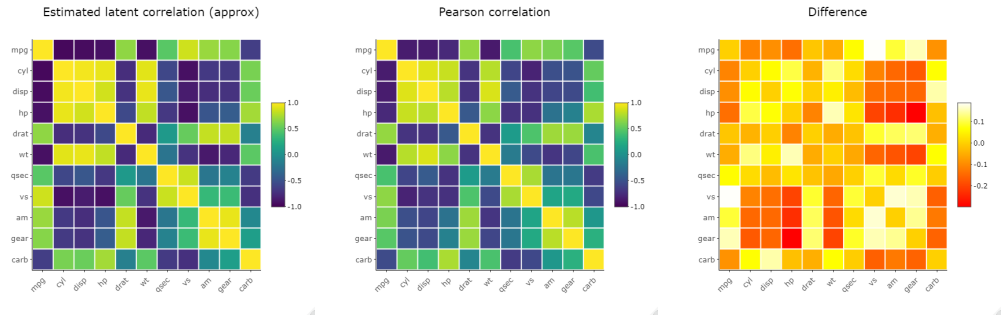

Figure 2: Heatmap of latent correlations (`approx`, left panel), Pearson correlation (middle panel), and difference between the two estimators (latent correlation - Pearson correlation) on the mtcars dataset

The script to reproduce Figure 2 is available here. We also provide interactive heatmaps for estimated latent correlations, Pearson correlations, and their differences (estimated latent correlations minus Pearson correlations) for the `mtcars` data set.

Basic Usage and Availability

The R package `latentcor` is available on Github. A getting started vignette with basic examples is available here. A vignette with mathematical background of estimation process as well as effect of optional parameters is available here.

Acknowledgments

We thank Dr. Grace Yoon for providing implementation details of the `mixedCCA` R package.

References

Croux, C., Filzmoser, P., & Fritz, H. (2013). Robust sparse principal component analysis. *Technometrics*, *55*(2), 202–214. https://doi.org/10.1080/00401706.2012.727746

Fan, J., Liu, H., Ning, Y., & Zou, H. (2017). High dimensional semiparametric latent graphical model for mixed data. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, *79*(2), 405–421. https://doi.org/10.1111/rssb.12168

Fox, J. (2019). *Polycor: Polychoric and polyserial correlations*. https://CRAN.R-project.org/package=polycor

Liu, H., Lafferty, J., & Wasserman, L. (2009). The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, *10*(10).

Makowski, D., Ben-Shachar, M. S., Patil, I., & Lüdecke, D. (2020). Methods and algorithms for correlation analysis in r. *Journal of Open Source Software*, *5*(51), 2306. https://doi.org/10.21105/joss.02306

Quan, X., Booth, J. G., & Wells, M. T. (2018). Rank-based approach for estimating correlations in mixed ordinal data. *arXiv Preprint arXiv:1809.06255*.

Team, R. C., & others. (2013). *R: A language and environment for statistical computing*.

Xue, L., & Zou, H. (2012). Regularized rank-based estimation of high-dimensional nonparanormal graphical models. *The Annals of Statistics*, *40*(5), 2541–2571. https://doi.org/10.1214/12-aos1041

Yoon, G., Carroll, R. J., & Gaynanova, I. (2020). Sparse semiparametric canonical correlation analysis for data of mixed types. *Biometrika*, *107*(3), 609–625. https://doi.org/10.1093/biomet/asaa007

Yoon, G., Gaynanova, I., & Müller, C. L. (2019). Microbial networks in SPRING-semiparametric rank-based correlation and partial correlation estimation for quantitative microbiome data. *Frontiers in Genetics*, *10*, 516. https://doi.org/10.3389/fgene.2019.00516

Yoon, G., Müller, C. L., & Gaynanova, I. (2021). Fast computation of latent correlations. *Journal of Computational and Graphical Statistics*, 1–8. https://doi.org/10.1080/10618600.2021.1882468