Robust Maximum Association Between Data Sets: The R Package ccaPP

Andreas Alfons
Erasmus Universiteit Rotterdam

Christophe Croux
KU Leuven

Peter Filzmoser
Vienna University of Technology

Abstract

This package vignette is an up-to-date version of Alfons, Croux, and Filzmoser (2016), published in the *Austrian Journal of Statistics*.

An intuitive measure of association between two multivariate data sets can be defined as the maximal value that a bivariate association measure between any one-dimensional projections of each data set can attain. Rank correlation measures thereby have the advantage that they combine good robustness properties with good efficiency. The software package **ccaPP** provides fast implementations of such maximum association measures for the statistical computing environment R. We demonstrate how to use **ccaPP** to compute the maximum association measures, as well as how to assess their significance via permutation tests.

Keywords: multivariate analysis, outliers, projection pursuit, rank correlation, R.

1. Introduction

Projection pursuit allows to introduce intuitive and therefore appealing association measures between two multivariate data sets. Suppose that the data sets $\boldsymbol{X}$ and $\boldsymbol{Y}$ consist of p and q variables, respectively. A measure of multivariate association between $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be defined by looking for linear combinations $\boldsymbol{X}\boldsymbol{\alpha}$ and $\boldsymbol{Y}\boldsymbol{\beta}$ having maximal association. Expressed in mathematical terms, we define an estimator

$$\hat{\rho}_R(\boldsymbol{X}, \boldsymbol{Y}) = \max_{\|\boldsymbol{\alpha}\|=1, \|\boldsymbol{\beta}\|=1} \hat{R}(\boldsymbol{X}\boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{\beta}), \tag{1}$$

where $\hat{R}$ is an estimator of a bivariate association measure R such as the Pearson correlation, or the Spearman or Kendall rank correlation. Using the projection pursuit terminology, $\hat{R}$ is the *projection index* to maximize. The projection directions corresponding to the maximum association are called *weighting vectors* and are estimated by

$$(\hat{\boldsymbol{\alpha}}_R(\boldsymbol{X}, \boldsymbol{Y}), \hat{\boldsymbol{\beta}}_R(\boldsymbol{X}, \boldsymbol{Y})) = \operatorname*{argmax}_{\|\boldsymbol{\alpha}\|=1, \|\boldsymbol{\beta}\|=1} \hat{R}(\boldsymbol{X}\boldsymbol{\alpha}, \boldsymbol{Y}\boldsymbol{\beta}). \tag{2}$$

Alfons, Croux, and Filzmoser (2017) developed the *alternate grid algorithm* for the computation of such maximum association estimators and studied their theoretical properties for various association measures. It turns out that the Spearman and Kendall rank correlation yield maximum association estimators with good robustness properties and good efficiency. This paper is a companion paper to Alfons *et al.* (2017) that demonstrates how to apply

the maximum association estimators in the statistical environment R (R Core Team 2015) using the add-on package **ccaPP** (Alfons 2016). The package is freely available on CRAN (Comprehensive R Archive Network, http://CRAN.R-project.org).

Note that using the Pearson correlation as the projection index of the maximum association estimator corresponds to the first step of canonical correlation analysis (CCA; see, e.g., Johnson and Wichern 2002), hence the package name **ccaPP**. Since CCA is a widely applied statistical technique, various algorithms and extensions are implemented in R packages on CRAN. Two important examples are briefly discussed in the following. The package **CCA** (González, Déjean, Martin, and Baccini 2008; González and Déjean 2012) extends the built-in R function `cancor()` with additional numerical and graphical output. Moreover, it provides a regularized version of CCA for data sets containing a large number of variables. Bayesian models and inference methods for CCA are implemented in the package **CCAGFA** (Klami, Virtanen, and Kaski 2013; Virtanen, Leppaaho, and Klami 2015).

The remainder of the paper is organized as follows. In Section 2, the design and implementation of the package are briefly discussed. Section 3 demonstrates how to compute the maximum association estimators, and Section 4 illustrates how to test for their significance. A comparison of computation times is given in Section 5. The final Section 6 concludes the paper.

2. Design and implementation

Various bivariate association measures and the alternate grid algorithm for the maximum association estimators are implemented in C++, and integrated into R via the package **RcppArmadillo** (Eddelbuettel and Sanderson 2014; Eddelbuettel, François, and Bates 2015).

The following bivariate association measures are available in the package **ccaPP**:

`corPearson()`: Pearson correlation

`corSpearman()`: Spearman rank correlation

`corKendall()`: Kendall rank correlation, also known as Kendall's τ

`corQuadrant()`: Quadrant correlation (Blomqvist 1950)

`corM()`: Association based on a bivariate M-estimator of location and scatter with a Huber loss function (Huber and Ronchetti 2009)

It should be noted that these are barebones implementations without proper handling of missing values. Hence the first three functions come with a substantial speed gain compared to R's built-in function `cor()`. Moreover, the fast $O(n \log(n))$ algorithm for the Kendall correlation (Knight 1966) is implemented in `corKendall()`, whereas `cor()` uses the naive $O(n^2)$ algorithm.

The alternate grid algorithm for the maximum association estimators is implemented in the function `maxCorGrid()`. Any of the bivariate association measures above can be used as projection index, with the Spearman rank correlation being the default. We do not recommend to use the quadrant correlation since its influence function is not smooth, which may result

in unstable estimates of the weighting vectors. For more details on the theoretical properties of the maximum association estimators, the reader is referred to Alfons *et al.* (2017).

To assess the significance of a maximum association estimate, a permutation test is provided via the function `permTest()`. Parallel computing to increase computational performance is implemented via the package **parallel**, which is part of R since version 2.14.0.

3. Maximum association measures

In this section, we show how to apply the function `maxCorGrid()` from the package **ccaPP** to compute the maximum association estimators. We thereby use the classic `diabetes` data (Andrews and Herzberg 1985, page 215), which are included as example data in the package.

First we load the package and the data. All measurements are taken for a group of $n = 76$ persons.

```
library("ccaPP")
data("diabetes")
x <- diabetes$x
y <- diabetes$y
```

Component `x` consists of $p = 2$ variables measuring *relative weight* and *fasting plasma glucose*, while component `y` consists of $q = 3$ variables measuring *glucose intolerance*, *insulin response to oral glucose* and *insulin resistance*. It is of medical interest to establish a relation between the two data sets.

The function `maxCorGrid()` by default uses the Spearman rank correlation as projection index.

```
spearman <- maxCorGrid(x, y)
spearman
##
## Call:
## maxCorGrid(x = x, y = y)
##
## Maximum correlation:
## [1] 0.5346995
```

The estimated weighting vectors can be accessed through components `a` and `b` of the returned object, respectively.

```
spearman$a
## [1] -0.2560459  0.9666646
spearman$b
## [1] 9.999999e-01 3.671395e-04 4.571134e-05
```

With the argument `method`, another bivariate association measure can be set as projection index, e.g., the Kendall rank correlation, the M-association or the Pearson correlation.

```
maxCorGrid(x, y, method = "kendall")
##
## Call:
## maxCorGrid(x = x, y = y, method = "kendall")
##
## Maximum correlation:
## [1] 0.3912216
maxCorGrid(x, y, method = "M")
##
## Call:
## maxCorGrid(x = x, y = y, method = "M")
##
## Maximum correlation:
## [1] 0.5328512
maxCorGrid(x, y, method = "pearson")
##
## Call:
## maxCorGrid(x = x, y = y, method = "pearson")
##
## Maximum correlation:
## [1] 0.4887634
```

Note that the Spearman and Kendall rank correlation estimate different population quantities than the Pearson correlation. Thus the above values of the different maximum association measures are not directly comparable. The argument `consistent` can be used for the former two methods to get consistent estimates of the maximum correlation under normal distributions.

```
maxCorGrid(x, y, consistent = TRUE)
##
## Call:
## maxCorGrid(x = x, y = y, consistent = TRUE)
##
## Maximum correlation:
## [1] 0.5526498
maxCorGrid(x, y, method = "kendall", consistent = TRUE)
##
## Call:
## maxCorGrid(x = x, y = y, method = "kendall", consistent = TRUE)
##
## Maximum correlation:
## [1] 0.5765741
```

The M-association measure is consistent at the normal model and estimates the same population quantity as the Pearson correlation.

4. Permutation tests

To assess the significance of maximum association estimates, permutation tests can be performed with the function `permTest()`. The number of random permutations to be used can be set with the argument `R`, which defaults to 1000. On machines with multiple processor cores, only the argument `nCores` needs to be set to take advantage of parallel computing in order to reduce computation time. If `nCores` is set to `NA`, all available processor cores are used.

In the examples in this section, we use 2 processor cores. To keep computation time minimal, we set the number of random permutations to 100. Furthermore, we set the seed of the random number generator via the argument `seed` for reproducibility of the results. Since we employ parallel computing, **ccaPP** uses random number streams (L'Ecuyer, Simard, Chen, and Kelton 2002) from the package **parallel** rather than the default R random number generator.

```
permTest(x, y, R = 100, nCores = 2, seed = 2016)
##
## Permutation test for no association
##
## r = 0.534699, p-value = 0.000000
## R = 100 random permuations
## Alternative hypothesis: true maximum correlation is not equal to 0
```

Again, the Spearman rank correlation is used as projection index by default. A different bivariate association measure can be specified via the argument `method`, which is passed down to the function `maxCorGrid()`.

```
permTest(x, y, R = 100, method = "kendall", nCores = 2, seed = 2016)
##
## Permutation test for no association
##
## r = 0.391222, p-value = 0.000000
## R = 100 random permuations
## Alternative hypothesis: true maximum correlation is not equal to 0
permTest(x, y, R = 100, method = "M", nCores = 2, seed = 2016)
##
## Permutation test for no association
##
## r = 0.532851, p-value = 0.000000
## R = 100 random permuations
## Alternative hypothesis: true maximum correlation is not equal to 0
permTest(x, y, R = 100, method = "pearson", nCores = 2, seed = 2016)
##
## Permutation test for no association
##
## r = 0.488763, p-value = 0.000000
## R = 100 random permuations
## Alternative hypothesis: true maximum correlation is not equal to 0
```

Clearly, all four tests strongly reject the null hypothesis of no association between the two data sets.

Since the focus of **ccaPP** is on robustness, we introduce an outlier into the `diabetes` data as in Taskinen, Kankainen, and Oja (2003). More precisely, we replace the value 0.81 of the first observation of variable *glucose intolerance* by 8.1, i.e., by a simple shift of the comma.

```
y[1, "GlucoseIntolerance"] <- 8.1
```

Now we repeat the four permutation tests with the contaminated data.

```
permTest(x, y, R = 100, nCores = 2, seed = 2016)
##
## Permutation test for no association
##
## r = 0.487536, p-value = 0.010000
## R = 100 random permuations
## Alternative hypothesis: true maximum correlation is not equal to 0
permTest(x, y, R = 100, method = "kendall", nCores = 2, seed = 2016)
##
## Permutation test for no association
##
## r = 0.361116, p-value = 0.000000
## R = 100 random permuations
## Alternative hypothesis: true maximum correlation is not equal to 0
permTest(x, y, R = 100, method = "M", nCores = 2, seed = 2016)
##
## Permutation test for no association
##
## r = 0.509973, p-value = 0.000000
## R = 100 random permuations
## Alternative hypothesis: true maximum correlation is not equal to 0
permTest(x, y, R = 100, method = "pearson", nCores = 2, seed = 2016)
##
## Permutation test for no association
##
## r = 0.267837, p-value = 0.350000
## R = 100 random permuations
## Alternative hypothesis: true maximum correlation is not equal to 0
```

The test based on the maximum Pearson correlation is highly influenced by the outlier and no longer rejects the null hypothesis. The tests based on the maximum Spearman and Kendall rank correlation, as well as the test based on maximum M-association, remain stable.

5. Computation times

This section analyzes the computation times of the methods implemented in **ccaPP**. All computations are performed in R version 3.2.2 on a machine with an Intel Xeon X5670 CPU. The computation times are recorded with the R package **microbenchmark** (Mersmann 2014).

Table 1: Average computation time (in milliseconds) of the bivariate association measures in base R and the package **ccaPP**.

	Base R			Package **ccaPP**			
n	Spearman	Kendall	Pearson	Spearman	Kendall	Pearson	M
100	0.20	0.40	0.08	0.03	0.03	0.01	0.11
1 000	0.41	18.73	0.08	0.18	0.16	0.01	0.32
10 000	3.38	1761.71	0.19	2.06	1.84	0.05	2.42
100 000	54.88	176431.15	1.34	25.21	22.46	0.41	27.42

First, we compare the barebones implementations of the Pearson, Spearman and Kendall correlations (functions `corPearson()`, `corSpearman()` and `corKendall()` in **ccaPP**) with their counterparts from the base R function `cor()`. We also include the M-association measure from the function `corM()` in the comparison. The bivariate association measures are computed for 10 random draws from a bivariate normal distribution with true correlation $\rho = 0.5$ and sample size $n = 100, 1\,000, 10\,000, 100\,000$. For each random sample, computation times from 10 independent runs are recorded.

Table 1 contains the average computation times of the bivariate association measures. Clearly, the fast $O(n \log(n))$ algorithm for the Kendall correlation (Knight 1966) in **ccaPP** is a huge improvement over the naive $O(n^2)$ algorithm in base R. Time savings for the Spearman and Pearson correlation are also substantial, considering that they are only due to a lack of missing data handling. For the M-association, the computation time is somewhat higher than that of the Spearman and Kendall correlation.

Since the projection pursuit algorithm for the maximum association measures involves computing a large number of bivariate associations (see Alfons *et al.* 2017), the faster barebones implementations are crucial to keep the computation of the maximum association feasible.

We employ the same procedure as above to record the computation time of the maximum association measures, except that each of the random samples is drawn from a multivariate normal distribution such that the true maximum correlation is $\rho = 0.5$ and the corresponding weighting vectors are $\alpha = (1, 0, \ldots, 0)'$ and $\beta = (1, 0, \ldots, 0)'$. The sample size is set to $n = 100, 1\,000, 10\,000$, the dimension of $\boldsymbol{X}$ is $p = 5, 10, 50$, and the dimension of $\boldsymbol{Y}$ is $q = 1, 5, 10, 50$.

Inspired by canonical correlation analysis (CCA), we also compute other association measures for comparison. In CCA, the first canonical correlation is given by the square root of the largest eigenvalue of the matrix

$$\boldsymbol{\Sigma}_{XX}^{-1} \boldsymbol{\Sigma}_{XY} \boldsymbol{\Sigma}_{YY}^{-1} \boldsymbol{\Sigma}_{YX}, \tag{3}$$

where $\boldsymbol{\Sigma}_{XX} = \mathrm{Cov}(\boldsymbol{X}), \boldsymbol{\Sigma}_{YY} = \mathrm{Cov}(\boldsymbol{Y})$, $\boldsymbol{\Sigma}_{XY} = \mathrm{Cov}(\boldsymbol{X}, \boldsymbol{Y})$ and $\boldsymbol{\Sigma}_{YX} = \boldsymbol{\Sigma}_{XY}'$ (see, e.g., Johnson and Wichern 2002). This is of course identical to the maximum association measure with the Pearson correlation as projection index. Other association measures are obtained by plugging different scatter matrices into (3). However, such a measure is in general different from the maximum association measure based on the corresponding bivariate association, with the maximum association being much easier to interpret. Here we plug in scatter matrices

Table 2: Average computation time (in seconds) of the maximum association measures in package **ccaPP**, as well as association measures based on corresponding full correlation matrix.

			Package **ccaPP**				Full scatter matrix			
n	p	q	Spearman	Kendall	Pearson	M	Spearman	Kendall	Pearson	MCD
100	5	1	0.014	0.011	0.001	0.036	0.001	0.005	0.001	0.020
100	5	5	0.073	0.049	0.006	0.244	0.002	0.010	0.001	0.038
100	10	1	0.030	0.023	0.003	0.088	0.002	0.012	0.001	0.044
100	10	5	0.114	0.083	0.012	0.473	0.002	0.021	0.001	0.075
100	10	10	0.180	0.107	0.021	0.658	0.003	0.037	0.001	0.130
100	50	1	0.174	0.137	0.047	0.641	0.007	0.224	0.003	0.926
100	50	5	0.588	0.429	0.365	5.777	0.008	0.259	0.003	1.096
100	50	10	0.692	0.435	0.426	8.249	0.009	0.307	0.003	1.348
100	50	50	1.257	0.824	0.993	33.368	0.013	0.839	0.005	
1 000	5	1	0.189	0.152	0.005	0.219	0.002	0.324	0.001	0.075
1 000	5	5	1.143	0.961	0.035	1.280	0.003	0.860	0.001	0.143
1 000	10	1	0.408	0.342	0.018	0.532	0.004	1.034	0.001	0.165
1 000	10	5	1.837	1.620	0.072	2.239	0.005	1.890	0.001	0.271
1 000	10	10	2.567	2.145	0.110	3.693	0.006	3.320	0.001	0.459
1 000	50	1	2.285	2.055	0.293	3.567	0.019	21.126	0.005	2.805
1 000	50	5	8.728	7.611	1.188	14.019	0.020	24.544	0.006	3.264
1 000	50	10	10.264	8.661	1.271	16.524	0.024	29.184	0.006	3.938
1 000	50	50	21.192	16.785	3.448	39.227	0.038	80.656	0.011	14.740
10 000	5	1	1.933	1.895	0.043	1.472	0.018	32.153	0.002	0.115
10 000	5	5	12.136	10.695	0.251	8.958	0.036	85.527	0.004	0.214
10 000	10	1	4.783	4.113	0.140	3.223	0.032	102.857	0.003	0.234
10 000	10	5	19.922	19.365	0.539	17.111	0.043	188.259	0.004	0.369
10 000	10	10	32.188	24.658	0.856	22.533	0.063	330.891	0.006	0.618
10 000	50	1	28.747	26.078	3.150	29.440	0.153	2107.029	0.028	3.374
10 000	50	5	116.614	100.885	9.538	114.121	0.160	2448.142	0.032	3.917
10 000	50	10	134.916	103.590	10.014	123.863	0.179	2910.402	0.035	4.706
10 000	50	50	244.389	209.834	20.318	224.293	0.320	8045.749	0.082	16.556

corresponding to the Pearson, Spearman and Kendall correlation. For the Pearson correlation, the corresponding scatter matrix is the sample covariance matrix. For the Spearman and Kendall correlation, the scatter matrices are given by the respective pairwise associations multiplied with scale estimates of the corresponding variables. Furthermore, since a multivariate M-estimator of the covariance matrix is not robust, we instead use the minimum covariance determinant estimator (MCD; see Rousseeuw and Van Driessen 1999).

Table 2 lists average computation times for various values of n, p and q. The function `maxCorGrid()` is thereby used with the default values for all control parameters of the algorithm (see the corresponding R help file). For the maximum association measures, the number of bivariate associations that have to be computed clearly takes a toll on computa-

tion time compared to the association measures based on the full scatter matrices. Note that the Kendall correlation is the exception, as the computation of the full scatter matrix uses R's built-in `cor()` function, and therefore the naive $O(n^2)$ algorithm. Also note that computing the full MCD scatter matrix requires more observations than variables, i.e., $n > p + q$, hence it cannot be computed for $n = 100$ and $p = q = 50$.

For the Pearson correlation, the projection pursuit algorithm to find the maximum association cannot be recommended since the first canonical correlation is much faster to compute. However, the focus of **ccaPP** is on the Spearman and Kendall rank correlation, for which the maximum association measures are much more intuitive than the association measures based on the full scatter matrix. In our opinion, the gain of easy interpretability outweighs the increased computational cost. In any case, the maximum association measures are still reasonably fast to compute for many problem sizes due to our C++ implementation.

It is also worth noting that the association measures based on a full scatter matrix require the number of observations to be larger than the number of variables in each of the two data sets, i.e., $n > \max(p, q)$. The maximum association measures do not have this limitation, although computation time increases considerably in high dimensions.

6. Conclusions

The package **ccaPP** provides functionality for the statistical computing environment R to compute intuitive measures of association between two data sets. These maximum association measures seek the maximal value of a bivariate association measure between one-dimensional projections of each data set. We recommend the maximum Spearman and Kendall rank correlation measures because of their good robustness properties and efficiency. For details on the theoretical properties of the estimators, as well as the alternate grid algorithm and extensive numerical results, the reader is referred to Alfons *et al.* (2017).

Due to our C++ implementation, the maximum association measures are reasonably fast to compute. The significance of maximum association estimates can be assessed via permutation tests, which allow for parallel computing to decrease computation time. In addition, the corresponding functions in **ccaPP** are easy to use.

References

Alfons A (2016). ***ccaPP**: (Robust) Canonical Correlation Analysis via Projection Pursuit.* R package version 0.3.2, URL http://CRAN.R-project.org/package=ccaPP.

Alfons A, Croux C, Filzmoser P (2016). "Robust maximum association between data sets: The R package **ccaPP**." *Austrian Journal of Statistics*, **45**(1), 71–79.

Alfons A, Croux C, Filzmoser P (2017). "Robust Maximum Association Estimators." *Journal of the American Statistical Association*, **112**(517), 436–445.

Andrews D, Herzberg A (1985). *Data.* Springer-Verlag, New York. ISBN 978-1-4612-5098-2.

Blomqvist N (1950). "On a Measure of Dependence Between Two Random Variables." *The Annals of Mathematical Statistics*, **21**(4), 593–600.

Eddelbuettel D, François R, Bates D (2015). ***RcppArmadillo**: **Rcpp** Integration for **Armadillo** Templated Linear Algebra Library*. R package version 0.6.200.2.0, URL http://CRAN.R-project.org/package=RcppArmadillo.

Eddelbuettel D, Sanderson C (2014). "**RcppArmadillo**: Accelerating R with High-Performance C++ Linear Algebra." *Computational Statistics & Data Analysis*, **71**, 1054–1063.

González I, Déjean S (2012). ***CCA**: Canonical Correlation Analysis*. R package version 1.2, URL http://CRAN.R-project.org/package=CCA.

González I, Déjean S, Martin P, Baccini A (2008). "**CCA**: An R Package to Extend Canonical Correlation Analysis." *Journal of Statistical Software*, **23**(12), 1–14.

Huber P, Ronchetti E (2009). *Robust Statistics*. 2nd edition. John Wiley & Sons, New York. ISBN 978-0-470-12990-6.

Johnson R, Wichern D (2002). *Applied Multivariate Statistical Analysis*. 5th edition. Prentice Hall, Upper Saddle River, New Jersey. ISBN 978-0-130-92553-4.

Klami A, Virtanen S, Kaski S (2013). "Bayesian Canonical Correlation Analysis." *Journal of Machine Learning Research*, **14**(Apr), 965–1003.

Knight W (1966). "A Computer Method for Calculating Kendall's Tau with Ungrouped Data." *Journal of the American Statistical Association*, **61**(314), 436–439.

L'Ecuyer P, Simard R, Chen E, Kelton W (2002). "An Object-Oriented Random-Number Package with Many Long Streams and Substreams." *Operations Research*, **50**(6), 1073–1075.

Mersmann O (2014). ***microbenchmark**: Accurate Timing Functions*. R package version 1.4-2, URL http://CRAN.R-project.org/package=microbenchmark.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Rousseeuw PJ, Van Driessen K (1999). "A Fast Algorithm for the Minimum Covariance Determinant Estimator." *Technometrics*, **41**(3), 212–223.

Taskinen S, Kankainen A, Oja H (2003). "Sign Test of Independence Between Two Random Vectors." *Statistics and Probability Letters*, **62**(1), 9–21.

Virtanen S, Leppaaho E, Klami A (2015). ***CCAGFA**: Bayesian Canonical Correlation Analysis and Group Factor Analysis*. R package version 1.0.7, URL http://CRAN.R-project.org/package=CCAGFA.

Affiliation:

Andreas Alfons
Erasmus Universiteit Rotterdam
PO Box 1738, 3000DR Rotterdam
E-mail: alfons@ese.eur.nl
URL: http://people.few.eur.nl/alfons/