rcd: An R Package for Estimating Robust Copula Dependence

Yi Li and Adam Ding

Abstract—The robust copula dependence (RCD) [1, 2] is recently introduced as an equitable dependence measure: it measures the dependence according to the strength of association regardless of the functional shape, treating linear and nonlinear relationships among the data equitably. It is useful to detect nonlinear relationships in data exploration. We introduce a new R package **rcd** for implementing the estimation of RCD using two methods: the kernel density estimation (KDE) and the k-nearest-neighbour (KNN) density estimation, with the latter one has smaller computational complexity in high-dimensional settings. The parallel programming with the **Rcpp** and **RcppParallel** packages is used to further speed up the estimators. The numerical performance of different estimators are evaluated with numerical experiments. The usage of functions in the **rcd** package is illustrated with numerical examples.

Keywords—Nonlinear Dependence Measures, Copula, Robust-equitability.

I. INTRODUCTION

WITH the explosion of the scale and complexity of the data, how to measure the dependence between random variables remains a fundamental problem in statistics and machine learning. The traditional measure, Pearson's correlation coefficient (ρ), is designed to detect linear relationship in the dataset. However, it fails to provide information on nonlinear relationships between the random variables. To this end, many dependence measures are proposed, such as mutual information [3], distance correlation [4] among others [5, 6, 7].

Furthermore, it is important to compare the performance of the dependence measures through theoretical properties as how they quantify the dependence for various types of relationships. Recently, the concept of equitability is first proposed by Reshef et al. [8], which states that a dependence measure should give equal importance to all relations: linear and nonlinear. This concept is further formalized by Kinney and Atwal [3] with the definition of self-equitability – a dependence measure should be invariant to any deterministic transformation of the marginal variables, under a nonlinear regression model with additive noise.

Chang et al. [1] proposed the Robust Copula Dependence (RCD), another dependence measure satisfying the self-equitability. Additionally, RCD is also robust-equitable, which means it treats equitably all types (linear and nonlinear) of deterministic signals hidden in background noises. The equitability properties help finding interesting complex relationships in large dataset. It would be useful to use this dependence measure in feature selection procedures [9, 10] to enable selecting first those features with strong but complex relationship with the response, rather than selecting first features with weak but simple relationship.

In this paper, we introduce a newly developed R [11] package **rcd**, to implement the RCD estimators. We implement the estimation of RCD with two common methods of probability density estimation, i.e. the kernel density estimation (KDE) [12, 13], and the k-nearest-neighbour (KNN) density estimation [14, 15, 16, 17]. Specifically, the bivariate version of the RCD is based on the KDE density estimation with the **Rcpp** package [18, 19]. To improve the performance of the multivariate version of the RCD, we also provide the KNN based estimator of RCD, and the calculation of the distance matrix is speeded up with the parallel programming package **RcppParallel** [20]. The comparison among different estimation methods is provided (see section IV).

The structure of this paper is as follow: In section II, we introduce definitions and theoretical properties of RCD, together with the necessary background for equitability. The estimation methods of the empirical version of RCD are also provided in this section. Section III contains the information of the **rcd** package. We discuss the structure of this package here. Several numerical examples with the real data application are presented in section IV.

II. THEORETICAL BACKGROUND

A. Equitability of RCD

In this section, we provide necessary background information for understanding the definition and the advantages of RCD. Further detailed discussion on RCD and its theoretical properties can be found in [1]. We first focus on the nonlinear regression model:

$$Y = f(X) + \epsilon, \tag{1}$$

where X and Y are two random variables, and ϵ is the independent random noise added to the regression function $f(X)$. The distribution of ϵ can only depend on X through the values of $f(X)$. Generally speaking, we use a dependence measure $D[X;Y]$ to understand the strength of the relationship between X and Y. A desirable property of $D[X;Y]$, *equitability* was first proposed by Reshef et al. [8]: $D[X;Y]$ should give similar scores to equally noisy relationships of different types in the model (1). This idea is further formalized in Kinney and Atwal [3] as:

Definition 1: A dependence measure $D[X;Y]$ is *self-equitable* if and only if $D[X;Y] = D[f(X);Y]$ whenever f is the function in model (1).

Yi Li and Adam Ding are with the Department of Mathematics, Northeastern University, Boston, MA, 02115, USA e-mail: a.ding@northeastern.edu

Manuscript submitted May, 2017.

As a first step to construct such measure, we consider the functional space where all f's are strictly monotone continuous deterministic functions. This motivates us to make use of the probability integral transformation (PIT) and the copula theory [21] to separate the dependence information from the marginal distributions, by taking advantage of the Sklar's theorem [21]. Specifically, for any joint distribution function $F_{X,Y}(x,y) = Pr(X \leq x, Y \leq y)$, there exists a copula C – a probability distribution on the unit square $\mathcal{I}^2 = [0,1] \times [0,1]$ – such that

$$F_{X,Y}(x,y) = C[F_X(x), F_Y(y)] \qquad \text{for all } x, y. \qquad (2)$$

Here $F_X(x) = Pr(X \leq x)$ and $F_Y(y) = Pr(Y \leq y)$ are the marginal cumulative distribution functions (CDFs) of X and Y respectively. Similar results could be generalized to multivariate joint distributions. In other words, the copula C is the CDF of probability integral transformed, uniformly distributed, variables $U = F_X(X)$ and $V = F_Y(Y)$. We further denote the derivatives of the copula function $\frac{\partial^2}{\partial u \partial v} C(u,v)$ as $c(u,v)$, the copula density. In this way, the copula decomposition separates the dependence from any marginal effects, and the copula C and its density c capture all the dependence between X and Y.

In many applications, the noise scheme is not limited by the regression form in (1). In applications with sensor data, the deterministic signal is often hidden in continuous background noise. The copula function, i.e. the dependence structure in such case, could be modeled as the combination between the singular copula C_s, which representing the deterministic signal, and the independence copula $\Pi = uv$, which is the uniform distribution on the unit square. An equitable dependence measure would reflect the signal strength p, regardless of the relationship shapes reflected by C_s in such circumstance. Therefore, the following definition of *robust equitability* is proposed in [1]:

Definition 2: A dependence measure $D[X;Y]$ is robust-equitable if and only if $D[X;Y] = p$ whenever (X,Y) follows a distribution whose copula is $C = pC_s + (1-p)\Pi$, for a singular copula C_s.

Our robust copula dependence, RCD, is based on the distance between the copula density $c(u,v)$ and independence case $c \equiv 1$.

Definition 3: [1] Let X and Y be two random variables, and $U = F_X(X), V = F_Y(Y)$, where F_X and F_Y are the CDFs of X and Y. The copula density for the joint random variable (U,V) is denoted by $c(u,v)$. The *robust copula dependence* between X and Y is

$$RCD = \frac{1}{2} \iint_{I^2} |c(u,v) - 1| du dv. \qquad (3)$$

RCD has a value between 0 and 1, bigger value indicates stronger dependence. $RCD = 0$ when the variables are independent of each other, $RCD = 1$ when there is an deterministic relationship.

Proposition 4: [1] RCD is both self-equitable and robust-equitable.

B. Estimation of RCD

In this part, we present the two estimators for RCD. Since the RCD is essentially a functional of the copula density function, the estimators are inherently from the two well-known density estimation methods, i.e. the KDE estimator [12, 13] and the KNN estimator [14, 15, 16, 17].

1) The KDE-based Estimator: We first consider the bivariate dependence estimation. In other words, we study the dependence between two univariate random variables X and Y. Such bivariate pairwise dependence are required in many common applications. Let $\{Z_i = (U_i, V_i)\}_{i=1}^n$ be the n realization of random variables $Z = (U,V)$ on the copula scale with the density $c(u,v)$. The KDE estimator of the copula density is given by

^{n}

(4)

wh

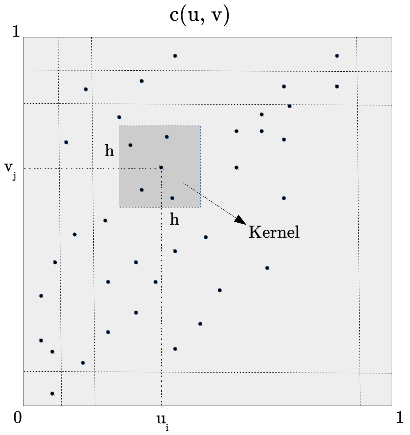

Fig. 1. The KDE density estimator. The small square illustrate the kernel function that is used to smooth the estimated density. The grid is divided in m by m for numerical integration purpose. The default value for m is set to be 200.

For computational simplicity, we use the kernel function K as the square centered at origin, with unit half-length: $K(z) = \mathbb{I}\{z : \|z\|_\infty \leq 1\}$. Here $\|z\|_\infty$ is the l_∞ norm, that it, it equals the maximum absolute value among all the coordinates of the vector z.

The density estimator $\hat{c}_{kde}(Z)$, is further plugged in equation (3). The numerical integration is approximated by summation over the $m \times m$ grid, by dividing each coordinate equally into m (with a default value of 200) intervals.

2) The KNN Estimator: For higher dimensional Z, the size of the grid grows exponentially with the dimension. Inspired by [14] and [15], we incorporate the KNN density estimation $\hat{c}$ and estimate RCD by $RCD(\hat{c}) = \sum_{\hat{c}(Z_i)>1} [1 - 1/\hat{c}(Z_i)]/n$. Here, $\{Z_i\}_{i=1}^n$ denotes the sample of high dimensional $Z = (U,V)$ where U and V can be multivariate vectors. The KNN estimator is

$$\hat{c}_{knn}(Z) = \frac{k/n}{A_{r(k,n)}}, \qquad (5)$$

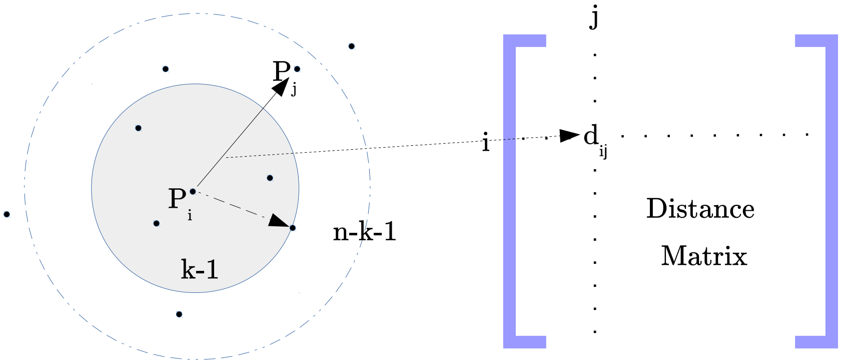

Fig. 2. The KNN density estimator.

where $r(k,n)$ is the distance from Z to its k-th closest of $Z_1, Z_2, \cdots, Z_n$ and $A_{r(k,n)}$ is the volume of the d-dimensional hyper-ball with radius r as illustrated in Figure 5.

This methods, however, requires the calculation of the distance matrix between each data point. To speed up the process, we apply the **RcppParallel** package to calculate the distance matrix. The overall consistency of the above two estimators is guaranteed in [1]. Due to the difference between the quadrature methods, for d-dimensional Z, the computational cost for the KDE based estimator is $O(nm^d)$ while the computational cost for the KNN based estimator is $O(n^2 d)$.

III. Structure of the Package

The structure of the package is presented in Figure 3. Most of the central algorithms are coded in C++. The source code is connected to R with the help of the **Rcpp** package by creating the corresponding R wrapper functions.

As we mentioned, the distance matrix is calculated with the **RcppParallel** package, which provides a complete toolkit for creating portable, high-performance parallel algorithms without requiring direct manipulation of operating system threads. Particularly, we utilize the high level parallel function (**parallelFor**), which uses Intel TBB[1] as a back-end on systems that support it and TinyThread[2] on other platforms.

Note that the parallel algorithm will use all the available cores on the machine. You can change the setting based on the following code.

```
R> require(RcppParallel)
R> setThreadOptions(numThreads =
R>    defaultNumThreads() - 1)
```

IV. Numerical Examples

In this section, we first compare several functions that are used in the **rcd** package and the pure R implementation of these functions. Then, we compare the estimation performance by the two estimators on a set of bivariate cases. Finally, we provide some examples on how to use the **rcd** function in the package.

A. *Example I: Performance Comparison*

Five functions are compared in two groups, the KDE and the KNN. We randomly generate data sets with sample sizes $n = 500, 1000, 2000, 3000, 4000$ and 5000 respectively. The elapsed time is calculated with the **rbenchmark** package [22]. The result is displayed in the first two panels of Figure 4.

The left panel of Figure 4 is the comparison between the KDE versions of the estimator for RCD with pure R and the **Rcpp** package. Result shows that the later version significantly accelerates the run time. Meanwhile, the middle panel of Figure 4 compares three functions, the pure R, **Rcpp** with serial calculation of distance matrix, and paralleling with the **RcppParallel** package. Result shows that the **Rcpp** and the parallel versions improve the performance.

The above comparison is based on the bivariate(d=2) case. A multivariate case with $d = 1000$, for the KNN based estimator only, is considered in the right panel of Figure 4. As we can see from this plot, as the computation cost becoming more expensive in each loop, the advantage of the parallelism becomes more obvious.

B. *Example II: Comparison of the Two Estimators*

Here we compare the estimation results of the KDE-based and KNN-based estimators, in the linear mixture model:

$$Y = pX + (1-p)\Pi, \tag{6}$$

where Π is the uniform noise on the unit support, and the p is the portion of the signal X (which follows a univariate uniform distribution). The theoretical RCD value equals to the portion p. As p varies between zero and one, the two sets of the estimated RCD values are displayed in left panel of Figure 5. The blue square is the estimation results for the KDE estimator, while the red circle is for the KNN estimator. Both of them fits the theoretical value, which is the diagonal line.

C. *Example III: Code Usage*

The development version of the **rcd** package could be installed with the **devtools** package [23] from GitHub. Note that for Windows machine users, in order to install the **rcd** package, the *Rtools* [3] needs to be installed properly.

[1] A C++ template library that provides portable (visa-vi instuction-sets and compilers) access to SIMD extensions.

[2] A C++ library for portable use of operating system threads.

[3] https://cran.r-project.org/bin/windows/Rtools/

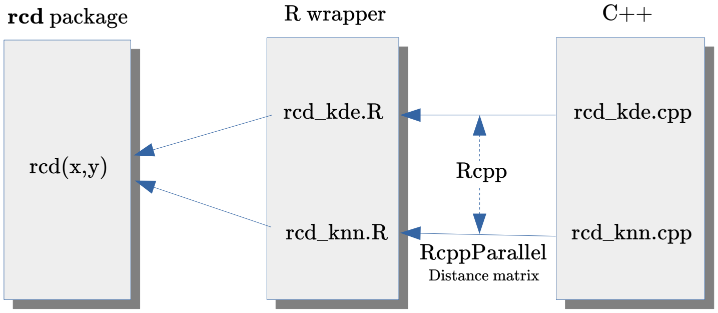

Fig. 3. The structure of the **rcd** package.

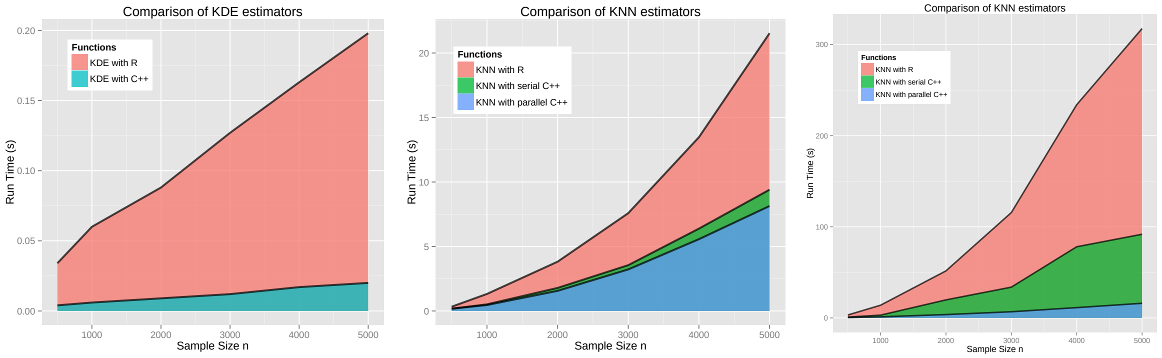

Fig. 4. Comparison of run time (in seconds) among the functions in the **rcd** package and the functions coded with pure R. The first two plots is based the bivariate case with sample size $n = 5000$, while the last plot is for multivariate case with dimension $d = 1000$.

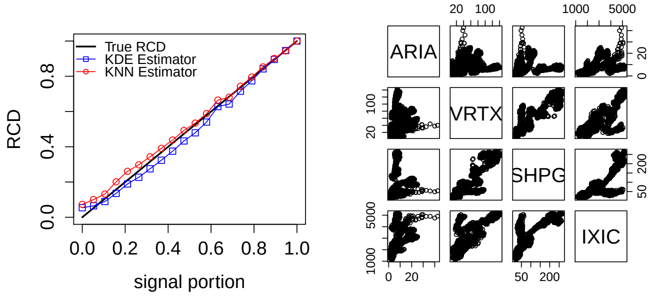

Fig. 5. Left panel: Comparison of the two estimators used in the **rcd** package with the linear relationship $Y = pX + (1-p)\Pi$. The blue square is for the KDE estimator, and the red circle is for the KNN estimator. Right Panel: The example stock price dataset **pharm** in the **rcd** package. This is a pair-wise plot for the price-price or the price-index combination of the columns in **pharm**.

```
R> require(devtools)
R> install_github("liyi-1989/rcd")
```

The basic usage of the **rcd** function is as follow:

```
R> require(rcd)
R> n <- 1000
R> x <- runif(n)
R> y <- x^2 + 2*runif(n)
R> rcd(x, y, method = "knn")
```

Note that the **method** argument specifies the estimation method we discussed above. For the KDE estimation, the bandwidth can be set with the **bandwidth** argument, while for the KNN method, the parameter k (number of nearest neighbors) can be set with the **k** augment. Without explicit setting by the users, both parameters use the default values from [1].

D. *Example IV: Data Set Example*

The **rcd** package comes with a dataset **pharm**, which include the stock prices of three pharmaceutical companies (ARIAD Pharmaceuticals, Vertex Pharmaceuticals, and Shire Plc) and one market index (Nasdaq Composite) from Jan. 3, 2000 to Feb. 12, 2006[4].

```
R> require(rcd)
R> data(pharm)
R> pairs(pharm[,2:5])
```

The empirical RCD values for each pair of the price (or index) are displayed in Table 1. According to the RCD values, the Shire Plc (SHPG) is most closely related to the market (IXIC) movement.

```
R> pharm.rcd<-diag(4)
R> colnames(pharm.rcd)<-names(pharm[2:5])
R> rownames(pharm.rcd)<-names(pharm[2:5])
R> for (i in 1:3) {
```

[4] The data is publicly available from Yahoo Finance.

	ARIA	VRTX	SHPG	IXIC
ARIA	1.00	0.48	0.54	0.55
VRTX		1.00	0.66	0.60
SHPG			1.00	0.70
IXIC				1.00

TABLE I
THE PAIR-WISE RCD FOR THE **PHARM** DATASET IN THE **RCD** PACKAGE.

```
R>    for (j in (i+1):4) {
R>        pharm.rcd[j,i]<-rcd(pharm[i+1],
R>             pharm[j+1])
R>        pharm.rcd[i,j]<-pharm.rcd[j,i]
R>     }
R> }
R> pharm.rcd
```

V. Conclusions

We introduced a newly developed R package **rcd** that implement the estimation of robust copula dependence measure. We review the definition and theoretical properties of RCD: it is equitable and measures the dependence strength equitably for different functional relationships. We implemented two estimation methods, KDE and KNN, where the KNN estimator has good computational speed in high-dimensional settings. We discussed and compared the usage of parallel programming tools for improving the performance speed. We demonstrate that the **rcd** package can efficiently calculate RCD. This enables the application of RCD as an effective way for calculating the nonlinear dependence strength in data exploration.

Acknowledgment

This project is partially supported by the NSF grant CCF-1442728.

References

[1] Y. Chang, Y. Li, A. A. Ding, and J. Dy. A robust-equitable copula dependence measure for feature selection. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Statistics*, pages 84–92, 2016.

[2] A. A. Ding, J. Dy, Y. Li, and Y. Chang. A robust-equitable measure for feature ranking and selection. In *Journal of Machine Learning Research, in press*, 2017.

[3] J. B. Kinney and G. S. Atwal. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences*, 111(9):3354–3359, 2014.

[4] G. J. Székely, M. L. Rizzo, et al. Brownian distance covariance. *The annals of applied statistics*, 3(4):1236–1265, 2009.

[5] D. Lopez-Paz, P. Hennig, and B. Schölkopf. The randomized dependence coefficient. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1–9. Curran Associates, Inc., 2013.

[6] D. Mari, M. S, and S. Kotz. *Correlation and Dependence*. World Scientific Publishing Company Pte Limited, 2001.

[7] K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In *NIPS*, volume 20, pages 489–496, 2007.

[8] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti. Detecting novel associations in large data sets. *science*, 334(6062):1518–1524, 2011.

[9] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1226–1238, 2005.

[10] M. Mandal and A. Mukhopadhyay. An improved minimum redundancy maximum relevance approach for feature selection in gene expression data. *Procedia Technology*, 10:20–27, 2013.

[11] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[12] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

[13] M. P. Wand and M. C. Jones. *Kernel smoothing*, volume 60. Crc Press, 1994.

[14] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[15] D. O. Loftsgaarden and C. P. Quesenberry. A nonparametric estimate of a multivariate density function. *Ann. Math. Statist.*, 36(3):1049–1051, 06 1965.

[16] D. S. Moore and J. W. Yackel. Large sample properties of nearest neighbor density function estimators. *Statistical Decision Theory and Related Topics*, II:269–279, 1977.

[17] Y. P. Mack and M. Rosenblatt. Multivariate k-nearest neighbor density estimates. *Journal of Multivariate Analysis*, 9(1):1–15, 1979.

[18] D. Eddelbuettel and R. François. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011.

[19] D. Eddelbuettel. *Seamless R and C++ Integration with Rcpp*. Springer, New York, 2013. ISBN 978-1-4614-6867-7.

[20] J. Allaire, R. Francois, G. Vandenbrouck, M. Geelnard, and Intel. *RcppParallel: Parallel Programming Tools for 'Rcpp'*, 2016. R package version 4.3.15.

[21] R. B. Nelsen. *An introduction to copulas (Springer series in statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[22] W. Kusnierczyk. *rbenchmark: Benchmarking routine for R*, 2012. R package version 1.0.0.

[23] H. Wickham and W. Chang. *devtools: Tools to Make Developing R Packages Easier*, 2015. R package version 1.9.1.