Estimating the Linfoot correlation in R

Sören Möller (joint with Jacob Hjelmborg)
moeller@health.sdu.dk

Epidemiology, Biostatistics and Biodemography, Department of Public Health,
University of Southern Denmark, Denmark

July 2nd, 2015
useR! 2015, Aalborg

Aim

- Investigate the Linfoot correlation
- Simulate data (in R)
- Estimate Linfoot correlation (in R)
  - Various estimators
  - Use existing R-packages when possible
- Compare estimates

Multiple Information and the Linfoot Correlation

Definition

The **mutual information** I of a copula with density c is given by

$$I = \int_0^1 \int_0^1 c(u, v) \log c(u, v) du dv.$$

Definition

The **Linfoot correlation** L (Linfoot, 1957) is given by

$$L = \sqrt{1 - \exp(-2 \cdot I)}.$$

Rényi's criterions

A dependence measure δ fulfils **Rényi's criterions (1959)** if

1. $\delta(X, Y)$ is defined for any pair of random variables X and Y neither of them being constant with probability 1.
2. $\delta(X, Y) = \delta(Y, X)$.
3. $0 \leq \delta(X, Y) \leq 1$.
4. $\delta(X, Y) = 0$ if and only if X and Y are independent.
5. $\delta(X, Y) = 1$ if there is a strict dependence between X and Y, i. e. either $X = g(Y)$ or $Y = f(X)$ where $g(x)$ and $f(x)$ are Borel-measurable functions.
6. $\delta(f(X), g(Y)) = \delta(X, Y)$ if the Borel-measurable functions $f(x)$ and $g(x)$ map the real axis in a one-to-one way onto itself.
7. $\delta(X, Y) = |R(X, Y)|$ if the joint distribution of X and Y is bivariate Gaussian. Here $R(X, Y)$ denotes the correlation coefficient of X and Y.

The Linfoot correlation fulfils all of Rényi's criterions.

Linfoot Correlation of the Clayton Copula

Theorem

For an bivariate Clayton copula (independent of the marginals) with parameter $\theta \geq 0$ *we have* **mutual information**

$$MI = \log(1+\theta) - (2\theta+1)\left(1+\frac{1}{1+\theta}\right) + 2(1+\theta)$$

and **Linfoot correlation**

$$L = \left(1 - \exp\left(-2\left(\log(1+\theta) - (2\theta+1)\left(1+\frac{1}{1+\theta}\right) + 2(1+\theta)\right)\right)\right)^{-1/2}$$

For the case $\theta = 0$ *this corresponds to the independence copula with* $I = 0$ *and* $L = 0$.

Estimation: Parametric

For (bivariate) Gaussian and Clayton distributed data, we know how to calculate the Linfoot correlation from distribution parameters.

- Fit parametric Clayton or Gaussian copula to the data using `fitCopula` from `library(copula)`
- Calculate the Linfoot correlation from the parameter

```
fit <- fitCopula(claytonCopula(2,dim=2),data,method="ml")
theta <- attr(fit,"estimate")
MI <- log(1+theta)  - (2*theta+1)*(1+1/(1+theta))
     + 2*(1+theta)
L <- sqrt(1-exp(-2*MI))
fit <- fitCopula(normalCopula(dim=2),data,method="ml")
L <- coef(fit)
```

In the Gaussian case, integrate, use plugin estimator or just calculate ρ.

Estimation: Kernel estimators

We use beta and Gaussian kernel estimators followed by Riemann integration or a plugin estimator to calculate the Linfoot correlation. Either `kde2d` from `library(MASS)` for Gaussian kernel estimator or

```
f <- function(p){
  sum(dbeta(p[1],u/h,(1-u)/h)*dbeta(p[2],v/h,(1-v)/h))
        /length(u)
}
g <- expand.grid(1:N/(N+1),1:N/(N+1))
mat <- matrix(apply(g,1,f),N,N)
MI <- sum(mat*log(mat))/N^2
L <- sqrt(1-exp(-2*MI))
```

Estimation: k-nearest neighbor

We use a k-nearest neighbot estimator to estimate the mutual information, and thus the Linfoot correlation.

- Use `mutinfo` from `library(FNN)` to calculate the mutual information
- Calculate the Linfoot correlation

```
MI <- mutinfo(x,y,k)
L <- sqrt(1-exp(-2*MI))
```

It is known from literature that this estimator can be improved by adding correction terms, which we intend to do in the future. Furthermore it is yet unclear, how to chose k. We use $k = 25$ for now.

Estimation: Empirical copula

We would like to use the empirical compula density to directly estimate the Linfoot correlation

- Use `empiricalCopula` from `library(copula)`
- Riemann-integrate to estimate the mutual information
- Calculate the Linfoot correlation

```
dens <- dempiricalCopula(x,y,N)$z*N^2
MI <- sum(dens*log(dens))/N^2
L <- sqrt(1-exp(-2*MI))
```

In practice this in most cases gives a total mass far below 1, a negtive mutual information and hence no estimate for L.
Hence we did not use this estimator in the simulation study.

Simulated data

Simulate Clayton data with Linfoot correlation L using `rMvdc` from `library(copula)`

```
if(L==0){theta=0}
else if(L==1){theta=Inf}
else{
 f <- function(x){Linfoot.Clayton.algebraic(x)$L-L}
 theta <- uniroot(f,interval=c(0,1000))$root
}
pmarg <- list(list(mean=0,sd=1),list(mean=0,sd=1))
rMvdc(N,mvdc(claytonCopula(theta,dim=2),
      c("norm","norm"),paramMargins=pmarg)
```

or simulate Gaussian data using `mvrnorm` from `library(MASS)`

```
mvrnorm(n=N,c(0,0),matrix(c(1,L,L,1),2,2))
```

Simulation results (mean and standard deviation)

Generating copula	Independence	Clayton	Clayton	Clayton
Number of simulation runs	1000	1000	1000	1000
True Linfoot correlation	0	0.25	0.50	0.75
Clayton	0.024 (0.020)	0.252 (0.033)	0.501 (0.026)	0.749 (0.016)
Gaussian + RI	0.024 (0.019)	0.210 (0.031)	0.434 (0.026)	0.672 (0.018)
Gaussian + PE	0.042 (0.032)	0.379 (0.057)	0.779 (0.040)	0.986 (0.005)
Gaussian + ρ	<0.001 (0.030)	0.211 (0.031)	0.435 (0.028)	0.672 (0.019)
Beta kernel + RI	$\dagger_{\mathbf{1000}}$	0.073 (0.035) $\dagger_{\mathbf{958}}$	0.331 (0.033)	0.618 (0.019)
Beta kernel + PE	0.081 (0.012)	0.186 (0.035)	0.418 (0.048)	0.660 (0.027)
Gaussian kernel + RI	0.110 (0.017)	0.226 (0.026)	0.432 (0.024)	0.664 (0.015)
k-nearest neighbour ($k = 25$)	0.100 (0.043) $\dagger_{\mathbf{524}}$	0.235 (0.047) $\dagger_{\mathbf{2}}$	0.499 (0.029)	0.754 (0.016)
Generating copula	Independence	Gaussian	Gaussian	Gaussian
Number of simulation runs	1000	1000	1000	1000
True Linfoot correlation	0	0.25	0.50	0.75
Clayton	0.024 (0.020)	0.229 (0.030)	0.050 (0.024)	0.680 (0.017)
Gaussian + RI	0.024 (0.019)	0.250 (0.029)	0.499 (0.023)	0.749 (0.014)
Gaussian + PE	0.042 (0.032)	0.433 (0.051)	0.826 (0.035)	0.956 (0.028)
Gaussian + ρ	<0.001 (0.030)	0.250 (0.029)	0.500 (0.023)	0.750 (0.013)
Beta kernel + RI	$\dagger_{\mathbf{1000}}$	0.072 (0.035) $\dagger_{\mathbf{871}}$	0.355 (0.028)	0.634 (0.016)
Beta kernel + PE	0.081 (0.012)	0.161 (0.019)	0.321 (0.027)	0.565 (0.031)
Gaussian kernel + RI	0.110 (0.017)	0.239 (0.024)	0.446 (0.021)	0.676 (0.013)
k-nearest neighbour ($k = 25$)	0.100 (0.043) $\dagger_{\mathbf{524}}$	0.248 (0.041)	0.502 (0.026)	0.753 (0.015)

$\dagger_n$: n runs did not converge

Future plans

- Evaluate more estimators from litterature
- Develop new estimators
- Use better numerical integration strategies
- Extend to more than two dimensions
- Test on larger simulations and more parameter values
  - Algoritmically faster implementations
  - Parallelization
- Apply these estimators on real data
  - Twin survival
  - Epi-genetic data
- Combine into an R-package

Conclusions

- Linfoot correlation is hard to estimate
- Even on simulated data
- Many estimators possible
- Most available in existing packages on CRAN
- Interesting and different (but not very good) results

Thank you for your attention!