Block Versions of R functions in 'generalCorr' for Generalized Correlations and Causal Paths

Hrishikesh D. Vinod*

October 30, 2019

Abstract

Karl Pearson developed the correlation coefficient r(X,Y) in 1890s. Vinod (2014) develops new generalized correlation coefficients so that when $r^*(Y|X) > r^*(X|Y)$ then X is the "kernel cause" of Y. Vinod (2017) reports simulations favoring kernel causality. An R software package called 'generalCorr' (at `www.r-project.org`) computes generalized correlations, partial correlations, and plausible causal paths. This short paper describes the block versions of various R functions newly added to the 'generalCorr' package in October 2019. Newly published Vinod (2019) has the latest rendering of the theory behind causal paths including theorems with proofs. The function 'causeSummBlk' is recommended.
Keywords: generalized measure of correlation, non-parametric regression, partial correlation, observational data, endogeneity.

1 Pearson correlation as a measure of dependence

An R package in Vinod (2016) called 'generalCorr' provides software tools for computing generalized correlation coefficients and preliminary determination of causal directions among a set of variables. The package is accessed by R commands (always in the red font for copy and paste):

```
if(!"generalCorr"%in%installed.packages()) {
install.packages("generalCorr",
repos = "http://cran.case.edu/")} ; library(generalCorr)
x=1:20
y=sin(x)
cor.test(x,y)
```

*Vinod: Professor of Economics, Fordham University, Bronx, New York, USA 104 58. E-mail: vinod@fordham.edu. I thank Fred Viole of Fordham for pointing out a limitation of my dependence measure for large sample sizes.

Note that in this example y is perfectly dependent on x.

The output of the above code is as follows

```
> cor.test(x,y)
Pearson's product-moment correlation
data:  x and y
t = -0.40418, df = 18, p-value = 0.6908
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.5157148  0.3629142
sample estimates:
cor
-0.0948372
```

The above output clearly shows that Pearson correlation coefficient r_{xy} is close to zero and that it is statistically insignificant. Since y perfectly depends on x, a proper measure of dependence should somehow reveal this 100% dependence. Yet we find that $r_{xy} = -0.095$ underestimates dependence by $1-0.095 = 0.905$, implying a staggering 91% underestimation. The underestimation occurs because the usual correlation coefficient measures only linear dependence and the functional dependence of y on x by the relation $y = sin(x)$ is highly nonlinear.

Using the 'generalCorr' package a general non-symmetric matrix of generalized correlation coefficients is computed as follows.

```
gmcmtx0(cbind(x,y))
```

The matrix output by 'gmcmtx0' is non-symmetric with the column name as the most plausible "cause" and row name as its effect.

```
> gmcmtx0(cbind(x,y))
   x          y
x  1 -0.04847292
y -1  1.00000000
```

The first column (headed x) has the generalized correlation of -1 suggesting that the independent variation in x is 100% responsible for causing the variation in y the variable named along the second row. On the other hand, if we consider the independent variation in the second column variable y, has only about 5% effect on x named along the first row.

As a measure of dependence (upon ignoring the causal direction) we can consider the larger of the two generalized correlations. The version 1.1.4 of 'generalCorr' has a new function 'depMeas' for this purpose. Even though the code for 'depMeas' can compute the blocking case the default is no blocking. Explicit choice of 'blksiz' is needed to get the blocking version. All other block versions of functions in 'generalCorr' package have the default of blksiz=10. In other words, one gets the block version by simply using the function without having to explicitly specify block size.

```
depMeas(x,y)
```

It is used pairwise, and correctly recognizes that $y = sin(x)$ has perfect 100% dependence or -1 as the signed dependence measure.

```
> depMeas(x,y)
[1] -1
```

Unfortunately, this measure fails to work so well when we use a larger sample size. For examples, the following R output shows (R input is suppressed for brevity) that a block version is needed to give a correct measure of dependence (close to perfect 100%)

```
> x=1:40; y=sin(x); depMeas(x,y)
[1] 0.321893
> x=1:40; y=sin(x); depMeas(x,y,blksiz=10)
[1] 1

> x=1:60; y=sin(x); depMeas(x,y)
[1] -0.2507747
> x=1:60; y=sin(x); depMeas(x,y, blksiz=10)
[1] -1
```

The default value of block size is 10. The kernel regression used for the three criteria of causality as explained in Vinod (2019) chooses one bandwidth per variable for the entire sample. The blocking adds one more bandwidth for each block. This allows greater flexibility in nonlinear kernel fitting at the cost of adding more bandwidth parameters, depending on the sample size

Let us report an illustrative complicated function relating x and y, where we expenct a good measure of dependence to be close to unity.

```
x=1:40
y= sin(x)+3*(cos(x))^3
depMeas(x,y, blksiz=10)
```

It is interesting that many complicated functions relating x and y do give high dependence result.

```
depMeas(x,y, blksiz=10)
[1] 0.9685486
```

If one tries $y = exp(x) * (cos(x))^2 + 99sqrt(x)$ we find

```
depMeas(x,y, blksiz=10)
[1] 0.9932447
```

We can safely conclude that strange but exact nonlinear relations between x and y yield near unity measures of dependence, avoiding the extreme underestimation by the traditional correlation coefficient.

2 Block version of matrix of generalized correlation coefficients

For larger sample sizes we have seen that more parameters in the form of bandwidths are needed to get a good fit to nonlinear relations. Consider

```
x=1:30;y=sin(x)
```

Now the new function 'gmcmtxBlk,' (a block version of the function 'gmcmtx0') gives the following output, with the same interpretation of column names as cause and row names are response or effect. The output correctly identifies x as the cause with a larger absolute value of the generalized correlation coefficient $r^*_{y|x} = -1$ along the second row.

```
> gmcmtxBlk(cbind(x,y),blksiz=10)
   x          y
x  1 -0.9433333
y -1  1.0000000
```

Since the magnitudes satisfy the inequality $|r^*_{x|y} = -0.943| < |r^*_{y|x} = -1|$, our algorithm treats this as evidence favoring the conditioning $y|x$ implying the correct causal path $x \rightarrow y$ for our artificial example. Vinod (2017) and Vinod (2019) treat the inequality among cross diagonal generalized correlation coefficients as Cr3, the third criterion. Our experience suggests that sometimes it can fail to identify the correct cause and must be supplemented with additional criteria Cr1 based on implementation of Hausman-Wu test for exogeneity and Cr2 based on absolute values of residuals. Both Cr1 and Cr2 involve inequalities quantified by numerical integrals based on four orders of stochastic dominance.

Overall causality is conveniently revealed by the function 'causeSummary(cbind(x,y))'. All three criteria Cr1 to Cr3 are dependent on the 'np' package giving good kernel regression estimates of conditional expectation functions. The use of only one bandwidth parameter for the entire range of sample data may not be enough for the task of identifying the causal paths as seen from the following example:

When the sample size $n = 20$ holds the 'causeSummary' function of 'generalCorr' reports the cause and response along with the Pearson correlation 'corr' and its traditional p-value.

```
x=1:20;y=sin(x)
causeSummary(cbind(x,y))
```

The above code for $n = 20$ correctly identifies the causal path $x \rightarrow y$ with strength 100.

```
     cause response strength corr.     p-value
[1,] "x"   "y"      "100"    "-0.0948" "0.69084"
```

Now consider the case where $n = 30$.

```
x=1:30;y=sin(x)
causeSummary(cbind(x,y))
```

The output of the above code is as follows.

```
     cause response strength corr.    p-value
[1,] "y"   "x"      "100"    "-0.131" "0.4903"
```

The above output suggests the wrong causal path $y \to x$, mostly due to the single bandwidth limitation of 'np' package when sample size is large.

The October 2019 version (1.1.4) of 'generalCorr' has a new function 'causeSummBlk' to allow a new bandwidth after every blksiz=10 block of observations. Again, the block version is found to work better for $n = 30$.

```
x=1:30;y=sin(x)
causeSummBlk(cbind(x,y),blksiz=10)
```

The output shows that the block version gives the correct causal path: $x \to y$.

```
> causeSummBlk(cbind(x,y), blksiz=10)
    cause response strength corr.    p-value
[1,] "x"   "y"      "100"    "-0.131" "0.4903"
```

3 Block versions of partial correlations

The newer version 1.1.4 of the package 'generalCorr' adds block version of 'parcor_ijk,' the partial correlation between x and y after removing the effect of z. The block version is called 'parcorBijk'. The following code illustrates its use.

```
x=1:30;y=sin(x)
set.seed(99);z=runif(30)
parcor_ijk(x,y,z)
parcorBijk(x,y,z)
```

Blocking obviously makes a difference as seen in the following output.

```
> parcor_ijk(x,y,z)
$ouij
[1] -0.0827093
$ouji
[1] -0.5205535

> parcorBijk(x,y,z)
$ouij
[1] -0.249118
$ouji
[1] -0.4539861
```

4 Conclusion

It appears that blocking does improve the performance of generalized correlations and causal path algorithms. The new functions 'causeSummBlk(cbind(x,y,z))', 'gmcmtxBlk(cbind(x,y,z)) along with a measure of dependence 'depMeas(x,y,blksiz=10)' can be recommended for general use. It would be interesting to use simulations to guide the practitioner in the choice of block size set by the optional parameter 'blksiz.'

References

Vinod, H. D. (2014), "Matrix Algebra Topics in Statistics and Economics Using R," in "Handbook of Statistics: Computational Statistics with R," , eds. Rao, M. B. and Rao, C. R., New York: North Holland, Elsevier Science, vol. 34, chap. 4, pp. 143–176.

— (2016), *generalCorr: Generalized Correlations and Initial Causal Path*, Fordham University, New York, R package version 1.1.2, 2018, has 3 vignettes, URL `https://CRAN.R-project.org/package=generalCorr`.

— (2017), "Generalized correlation and kernel causality with applications in development economics," *Communications in Statistics - Simulation and Computation*, 46, 4513–4534, posted online: 29 Dec 2015, URL `https://doi.org/10.1080/03610918.2015.1122048`.

— (2019), "New Exogeneity Tests and Causal Paths," in "Handbook of Statistics: Conceptual Econometrics Using R," , eds. Vinod, H. D. and Rao, C. R., New York: North Holland, Elsevier, vol. 41, chap. 2, pp. 33–64, URL `https://doi.org/10.1016/bs.host.2018.11.011`.