Nonlinear Granger Causal Paths, Dependence Measures and Canonical Correlations

Hrishikesh D. Vinod*

November 29, 2020

Abstract

A popular F test of Granger-causality relies on normally distributed errors of ordinary least squares (OLS) linear regressions. There is a long-standing need for a user-friendly algorithm replacing the OLS by kernel regressions, and the F test by a bootstrap. This paper introduces a version (1.1.6) of the R package 'generalCorr' which offers (`bootGcRsq`) to satisfy the need. Granger causality requires the 'cause' to occur at a time before the 'effect' occurs, ruling out concurrent causality. The R command (`causeSummary`) for assessing concurrent causality possibly in cross-sectional data is now enhanced by (`causeSummBlk`). The command (`gmcmtx0`) for the non-symmetric matrix of generalized correlation coefficients R^* is enhanced by (`gmcmtxBlk`). The asymmetric R^* leads to two new concepts, (i) measures of dependence implemented by the command (`depMeas`), and (ii) generalized canonical correlations which explicitly incorporate pairwise non-linear dependence between linear combinations of variables. The latter needs a new Lagrangian maximization implemented by the command (`canonRho`). We illustrate its application using joint production of wool and mutton by capital and labor.

Keywords: generalized measure of correlation, non-parametric regression, observational data, endogeneity.

1 Introduction: Improving correlation as a measure of dependence

The threefold aim of this paper is to discuss usage of three software tools or functions newly available in the R package 'generalCorr' versions (1.1.6) and later.

(i) Newer measures of dependence in Section 2.

*Vinod: Professor of Economics, Fordham University, Bronx, New York, USA 104 58. E-mail: vinod@fordham.edu. `https://www.ssrn.com/abstract=3737870` also has this paper which is a vignette number 5 of the R package 'generalCorr.'

(ii) Granger-causality using kernel regressions and maximum entropy bootstrap in Section 3

(iii) Canonical correlations defined for non-symmetric matrix of generalized correlation coefficients R^* in Section 4.

We include examples with R code illustrating the usage of these new tools. The reader is invited to copy any, if not all, material in the red font and paste it on their own R console. The response by R should match the material in blue font. We have shortened the response output of the code for brevity.

2 Newer measures of dependence

Karl Pearson developed the correlation coefficient r(X, Y) in the 1890s. We show that it can underestimate the dependence between two variables by as much as 91%. Expanding on Zheng et al. (2012) Vinod (2014) develops new generalized correlation coefficient which avoids underestimation, furthermore showing that when $r^*(Y|X) > r^*(X|Y)$, 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 added to the 'generalCorr' package in versions 1.1.5 and later. Vinod (2019) describes the latest rendering of the underlying theory behind causal paths including formal theorems with proofs. The newer R function 'causeSummBlk(.)' is recommended for practitioners.

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. Newer versions provide further enhancements. In the sequel we use the red font to identify code which is ready for copy and paste to the reader's own R console. For example consider the code:

```
library(generalCorr)
x=1:20; y=sin(x)
cor.test(x,y)
```

Note that $y = sin(x)$ is perfectly dependent on x in this example. Next, we shall see that the usual correlation coefficient underestimates dependence. The output of the above code is as follows

```
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 the 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 nonlinear.

A general non-symmetric matrix of generalized correlation coefficients is computed by using the 'generalCorr' package as follows.

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

The matrix output produced by the R function 'gmcmtx0(.)' is non-symmetric with the column name as the most plausible "cause" and row name as its "effect." The conditioning variable may be stated as $r^*_{row|column}$, with an asterisk to denote the generalized correlation. The matrix is expected to be generally non-symmetric, similar to causal paths.

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

The second row (named y) of the first column (named x) reports the generalized correlation of $r^*_{y|x} = -1$. This suggests that the independent variation in x is 100% responsible for causing the variation in $y = sin(x)$. On the flipped side (across the diagonal of the reported matrix), we have the effect of independent variation in y (the second column name), $r^*_{x|y} = -0.0485$. That is, only about 5% effect on $x = 1, 2, \ldots 20$ named along the first row can be attributed to independent variation in $y = sin(x)$.

Upon ignoring the causal direction, a symmetric measure of dependence can be defined as the larger absolute value between the two generalized correlations.

$$depMeas = \text{sign}(r_{ij})\, max(|r^*_{i|j}|, |r^*_{j|i}|), \tag{1}$$

where we use the sign of the original correlation coefficient. For our simple problem with $n = 20$ we have $depMeas = -max(|-1|, |0.0485|) = -1$ implying perfect dependence of $y = sin(x)$ on x.

An alternate symmetric version uses the geometric mean (gm) or the square root of the cross products of two correlations defined as

$$R^{gm} = \{R^{gm}_{ij}\} = \{\text{sign}(r_{ij})\, \sqrt{(|(r^*_{i|j} * r^*_{j|i})|}\}, \tag{2}$$

where $\text{sign}(r_{ij})$ denotes the sign of the Pearson coefficient as before.

2.1 Block versions of some functions

The 'gmcmtx0(.)' uses nonparametric kernel regression of y on x and also of x on y via the R package 'np' Hayfield and Racine (2008), which in turn uses a single bandwidth parameter for each variable for the entire data range of n observations. Block version subdivides the data range into k non-overlapping intervals of block length defined by a parameter blksiz=10 as the default. For example, if we have $n = 10k$ observations, we have k blocks with the first block of the first 10 and the next block of the next 10 and so on for the k blocks. In general, n need not be an integer k times 'blksiz' chosen by the researcher. Since we cannot have less than blksiz=10 observations for fitting our nonparametric regressions (needed by 'gmcmtx0(.)' using the np package), we must let the last block become larger than the designated blksiz. For example, when $n = 25$, we have $k = 2$ blocks, with the first block having the first 10 observations, and the next block having all remaining 15 observations.

The block version of `gmcmtx0(.)` called `gmcmtxBlk(.)` is included in versions 1.1.5 or higher of my 'generalCorr' package. Considerable care is needed to create nonlinear nonparametric conditional expectation functions (fitted values) for each block and to further string them together into an overall vector of fitted values. The reader can know the exact algorithm by simply typing the name of the function on the R console.

A new R function `depMeas(x,y)` does not automatically implement the block version. The default blksiz is 'length(x),' that is entire sample size n is the block size, that is no blocking is done. A block version requires setting a smaller block size as: `depMeas(x, y, blksiz=10)` assuming $n > 10$.

Most block versions of functions in 'generalCorr' package use the letters 'Blk' or 'B' in their names and have the default of blksiz=10. For example, 'causeSummBlk' is a block version of one of the most useful and easy to use functions 'causeSummary' in 'generalCorr.' Also see 'gmcmtxBlk' for generalized correlation coefficients, whereas 'parcorBijk' and 'parcorBMany' provide their partial correlation versions. Causality scores with blocking are available in 'siPairsBlk.'

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

The depMeas(.) expects two vectors as inputs. The following output shows that it correctly recognizes that $y = sin(x)$ has perfect 100% dependence or -1 as the signed dependence measure.

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

Now we show the need for a block version for larger sample sizes, such as when $n > 20$ at say $n = 40$.

```
x=1:40; y=sin(x); depMeas(x,y)
```

The output below shows that the dependence is now wrongly estimated to be only 0.32, whereas the correct level is 1.

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

Now we show that a block version does indeed give a correct measure of dependence (close to perfect 100%) for two examples with $n = 40, n = 60$.

```
x=1:40; y=sin(x); depMeas(x,y,blksiz=10)
x=1:60; y=sin(x); depMeas(x,y, blksiz=10)
```

The output below shows that the dependence is correctly estimated to be 1.

```
> x=1:40; y=sin(x); depMeas(x,y,blksiz=10)
[1] 1
> x=1:60; y=sin(x); depMeas(x,y, blksiz=10)
[1] -1
```

The blocking works by implicitly adding an extra bandwidth parameter for each block allowing 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 expect 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)
```

Interestingly, many complicated functions relating x and y do give high dependence results.

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

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

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

We can safely conclude this section by claiming that great many arbitrary (complicated) but exact nonlinear relations between x and y yield near unity measures of dependence, avoiding the extreme underestimation by the traditional correlation coefficient.

3 Granger Causality Testing

Causality is generally difficult to test by statistical methods alone. Denote two time series as X and Y and one wants to predict the next value of Y from past values of Y and past values of X. If past values of X are useful for predicting Y, Economics Nobel laureate Clive Granger suggested in Granger (1969) that we can say that $X \to Y$ or X causes Y. Granger's

view of causality is popular and well covered in textbooks, (Hansen, 2020, sec. 14.43), under the name 'Granger causality' to be distinguished from true causation. It is usually a one-step-ahead predictive causality, admittedly subject to the famous fallacy of "post hoc ergo propter hoc" translated as "after this, therefore because of this."

Granger's tool is implemented in some R packages, including Zeileis and Hothorn's 'lmtest,' which contains a bivariate version of the test. Granger's non-causality null hypothesis is as follows. The past p-values of $X(X_{t-1}$, to $X_{t-p})$ do *not* help in predicting the next value of Y beyond the prediction by its own past values. We regress:

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} \ldots \alpha_p Y_{t-p} + \beta_1 X_{t-1}, \ldots, \beta_p X_{t-p} + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma^2). \tag{3}$$

An F-test in the 'lmtest' package tests the Granger non-causality null hypothesis that all coefficients of the past values of X are jointly zero: $(\beta_1 = \beta_2 = \ldots = \beta_p = 0)$.

The R package 'lmtest' comes with data for an intuitively attention-grabbing question: What came first: chicken or egg? It uses slightly changed Thurman & Fisher's 1988 annual data for years (1930 to 1983) on US production of these commodities. The following code shows how to use the 'grangertest' function on these data.

```
#Example: Did Chicken came first or Eggs?
library(lmtest); data(ChickEgg)
attach(data.frame(ChickEgg))
grangertest(egg ~ chicken, order = 3, data = ChickEgg)
```

The output of 'grangertest' (omitted for brevity) shows that F statistic is 0.5916, which is very small, with a p-value= $0.6238 > 0.05$, the usual 5% type I error. Granger non-causality null is not rejected (accepted), or the RHS variable 'chicken' does not Granger-cause the LHS variable 'egg,' implying the causal path (egg→ chicken).

Apart from the so-called 'post hoc' fallacy problem associated with relying on time-precedence to infer causality mentioned above, an important limitation of the Granger test is that it assumes linear relations among the variables in (3) with normal errors $\epsilon \sim N(0, \sigma^2)$. Newer versions of generalCorr ($\geq$ 1.1.6) use nonparametric kernel regressions as well as the bootstrap for statistical inference. The following code illustrates by using the chicken-egg data.

```
#Chicken/Eggs data above, meboot and kernel regressions
library(generalCorr);options(np.messages=FALSE)
b1=bootGcRsq(x1=chicken,x2=egg,pwanted=3,px1=3,px2=3,n999=999)
bb=b1[,3]; print(length(bb[bb>0])/999)
Fn=function(x) quantile(x,prob=c(0.025, 0.975))#confInt
apply(b1,2,Fn)#reports 95 percent confidence interval
```

Kernel regressions and maximum entropy bootstraps for realistic estimation and statistical inference are admittedly computer intensive. Even with only 54 data points, the above code takes 122 minutes on a Windows 10 Dell computer with i5-7500 CPU @ 340GHz with 8GB RAM and 64 bit operating system. The function 'bootGcRsq' uses 'local linear' fits and

Akaike Information Criterion (AIC) is used for bandwidth selection in the 'np' package. A default setting giving 'local constant' fit with least squares bandwidth selection is faster taking 61 minutes for the same job, with slightly less accurate estimates. Abridged output of the above code follows.

```
[1] "dif>0 means Rsq-X1-on-X2 > Rsq-X2-on-X1 with causal path X2-->X1"
[1] "95 percent confidence intervals for each column"
> bb=b1[,3]; print(length(bb[bb>0])/999)
[1] 0.982983
> Fn=function(x) quantile(x,prob=c(0.025, 0.975))#confInt
> apply(b1,2,Fn)#reports 95 percent confidence interval
          Rsq-X1-on-X2 Rsq-X2-on-X1     dif
2.5%     0.9455809    0.8088933 0.003659968
97.5%    0.9925729    0.9703288 0.166225143
```

The 95% confidence interval in the last column entitled 'dif' for the difference [R^2(x1-on-x2)$-R^2$(x2-on-x1)] between coefficients of determination of the two flipped kernel regressions does not contain the zero. The number of positive realizations of 'dif' are over 98%, suggesting that x2=egg Granger-causes x1=chicken, supporting the path (egg$\rightarrow$ chicken), with over 98% chance and agreeing with 'lmtest.' Thus, upon allowing for nonlinearity our bootstrap inference based on 999 resamples agrees with the F-test-determined path (egg$\rightarrow$ chicken) of Granger causality.

Real-world relations are rarely linear, and regression errors (ϵ_t) are rarely bell shaped. The above R code illustrates how to use the new function 'bootGcRsq' for any application where one wants to test Granger-causality using modern 21-st century tools without assuming (a) linear regressions and (b) Normally distributed errors needed to justify F tests.

4 Generalized Canonical Correlations

Now we turn to our third objective mentioned in the introduction. Hotelling (1936) is concerned with dependence relations between two sets of variables in his canonical correlation analysis (CCA). He places at least two variables in the left-hand side (LHS) set and at least two in the right-hand side (RHS) set. Hotelling's motivating examples include (i) relations between the human body and mind measured by physical and mental measurements on individuals, and (ii) jointly dependent agricultural product demands and prices based on sets jointly demanded products such as wheat, potatoes, barley, and rye. The dimension reduction methods of CCA are widely applied in many areas of science attested by many R packages we mention later.

This section extends his methods to a generalized non-symmetric correlation matrix R^* produced by the R command `generalCorr::gmcmtx0` partitioned as:

$$R^* = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}, \tag{4}$$

involving two sets of standardized (zero mean unit variance) variables, X_1 and X_2 with r and $(p-r)$ elements, respectively, with $r \leq (p-r)$. We define

$$X_1' = (x_1, x_2, \ldots x_r), \tag{5}$$
$$X_2' = (x_{r+1}, x_{r+2}, \ldots x_p), \tag{6}$$
$$w_1 = \alpha' X_1, \tag{7}$$
$$w_2 = \beta' X_2, \tag{8}$$

where w_1 and w_2 are two sets of linear combinations of variables with an $r \times 1$ vector containing LHS weights $\alpha = [\alpha_i, i = 1, 2, \ldots r]$ and a similar $(p-r) \times 1$ vector of RHS weights $\beta = [\beta_j, j = 1, 2 \ldots (p-r)]$. The estimation of CCA is feasible only when the larger set with coefficients denoted by β is on the RHS.

The traditional CCA maximizes the bilinear form $\alpha' \Sigma_{12} \beta (= \beta' \Sigma_{21} \alpha)$, the covariance between w_1 and w_2. It is a sum of cross products written as a bilinear quadratic form. Our generalized CCA proposed here replaces the usual correlation matrix $R = \{r_{ij}\}$ by a non-symmetric $R^* = \{r^*_{i|j}\}$, whose elements are signed square roots of R^2 of two flipped regressions between LHS and RHS sets (Result 3) consistent with the partition in (4). While traditional CCA can begin with a partition of variance-covariance matrices between $X_j, j = 1, 2, \ldots p$ variables, the first step of generalized CCA must be a partitioning of generalized correlation matrix R^*, not a covariance matrix.

Since R^* is non-symmetric with $\Sigma_{12} \neq \Sigma_{21}$, it makes a difference whether the maximand is $\alpha' \Sigma_{12} \beta$ or $\beta' \Sigma_{21} \alpha$. Hence, we let us maximize an average of the two bilinear forms with a new maximand $(1/2)[\alpha' \Sigma_{12} \beta + \beta' \Sigma_{21} \alpha]$.

The analytical implications of the new maximand are explored next. Our modified Lagrangian maximand is

$$L = (1/2)[\alpha' \Sigma_{12} \beta + \beta' \Sigma_{21} \alpha - \mu_1 (\alpha' \Sigma_{11} \alpha - 1) - \mu_2 (\beta' \Sigma_{22} \beta - 1)]. \tag{9}$$

The first-order condition from matrix derivatives from matrix algebra texts (Vinod, 2011, p. 256) for $\partial L / \partial \alpha$ yields

$$\beta' [\Sigma_{12}' + \Sigma_{21}] = \mu_1 \alpha' (\Sigma_{11} + \Sigma_{11}'), \tag{10}$$

which differentiates a bilinear form involving a non-symmetric matrix. After incorporating $\partial L / \partial \beta$ and algebraic manipulations

$$\beta' [\Sigma_{12}' + \Sigma_{21}] (\Sigma_{11} + \Sigma_{11}')^{-1} [\Sigma_{12} + \Sigma_{21}'] (\Sigma_{22} + \Sigma_{22}')^{-1} = \mu_2^2 \beta'.$$

Changing the notation μ_2^2 by ρ^2 and computing a transpose of both sides gives the equation

$$(\Sigma_{22} + \Sigma_{22}')^{-1} [\Sigma_{12} + \Sigma_{21}]' (\Sigma_{11} + \Sigma_{11}')^{-1} [\Sigma_{12}' + \Sigma_{21}]' \beta - \rho^2 \beta = 0. \tag{11}$$

It is convenient to simplify (11) by defining

$$A^* = (\Sigma_{22} + \Sigma_{22}')^{-1} [\Sigma_{21} + \Sigma_{12}'] (\Sigma_{11} + \Sigma_{11}')^{-1} [\Sigma_{12} + \Sigma_{21}'], \tag{12}$$

which is observable provided the two matrix inverses exist.

Substituting A^* in (11) we have $A^*\beta - \rho^2\beta = 0$ or $(A^* - \rho^2 I)\beta = 0$. We rule out the trivial solution $\beta = 0$ and get the determinantal equation of the usual eigenvalue-eigenvector problem. It yields our estimate of β as the eigenvector associated with the largest eigenvalue ρ^2. We maximize of the Lagrangian by choosing the eigenvector associated with the largest ordered eigenvalue ρ^2.

Traditional CCA makes the LHS coefficients α observable by solving

$$\alpha = [\Sigma_{11}^{-1}\Sigma_{12}\beta]/\rho, \tag{13}$$

where the square root ρ of the eigenvalue has an arbitrary sign, suggesting that canonical correlation estimates of α, β are known only up to their signs. The freedom to change all signs (from negative to positive, say) can be useful in interpreting the estimated α coefficients to make sense in applications. Using the package 'generalCorr,' the command `canonRho(mtx, nLHS = 2, sgn = 1, verbo = FALSE, ridg = c(0, 0))` readily implements this.

In our non-symmetric case, we estimate α by:

$$\alpha = [(\Sigma_{11} + \Sigma_{11}')^{-1}[\Sigma_{12} + \Sigma_{21}']\beta]/\rho. \tag{14}$$

Our derivation above generalizes the following equation from the traditional CCA,

$$\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\beta - \rho^2\beta = 0 = A\beta - \rho^2\beta, \tag{15}$$

well-defined only when the two matrix inverses exist. Traditional CCA algorithm uses the familiar eigenvalue-eigenvector problem $A\beta = \rho^2\beta$ to estimate β as the eigenvector associated with the largest eigenvalue ρ^2. Generalized CCA simply replaces A by A^*.

Vinod (1976) confronts some numerical issues with the traditional CCA and solves them by adding a matrix λI before the matrix inversions in (15), where λ is a small constant based on measurement errors. The version with

$$A_r = (\Sigma_{22} + \lambda I)^{-1}\Sigma_{21}(\Sigma_{11} + \lambda I)^{-1}\Sigma_{12}$$

is referred to as 'regularized' CCA in many computer 'deep learning' applications. For example, Tuzhilina et al. (2020) discuss group regularization, and Garcia-Medina and Gonzalez (2020) make their estimates less sensitive to small perturbations in the data by using $\lambda = 0.01$.

Since the generalized correlation coefficients $r^*_{i|j}$ comprising non-symmetric R^* matrix are generally larger in magnitude than the usual correlation matrix containing r_{ij}, the diagonal blocks Σ_{11}, Σ_{22} are likely to be collinear. Hence, the regularization adding the λI matrix appears desirable in applications using our extension.

We cannot guarantee that the eigenvalue-eigenvector problem $A^*\beta = \rho^2\beta$ from asymmetric R^* matrices will always yield real eigenvalues. If eigenvalues contain imaginary numbers, one can use our symmetric versions, R^{max} or R^{gm}, based on nonlinear dependence defined in (1) or (2), guaranteed to avoid imaginary eigenvalues, while admitting nonlinear dependence among all variable pairs.

Recall that generalized $r^*_{i|j}$ values avoid severe underestimation of dependence by the usual correlation coefficients r_{ij} in cases where the dependence is nonlinear. Hence the users of traditional CCA need an option to use a superior measure of pairwise dependence. We have demonstrated that one can indeed replace all partitions of Σ matrices in derivations of traditional CCA. We replace Σ by a generalized non-symmetric matrix of correlations R^*, whose elements represent superior measures of dependence between all variable pairs. Next, we use a similar partitioning, making several appropriate modifications of Hotelling's CCA derivation to obtain new estimates of coefficients (α, β), while relaxing some of his assumptions. These new coefficients use the modified maximand from (9).

González and Déjean's R package 'CCA' cites Vinod (1976) and contains data sets on which one can try our function `canonRho`. The package 'ccaPP, incorporates projection pursuit, 'CCP' provides significance testing, 'nscancor' imposes non-negativity constraints. Much work needs to be done in this area with numerous practical applications.

4.1 Numerical examples of new CCA

In an attempt to explore the implications of the new maximand from (9) this subsection considers an example of joint production of wool and mutton from capital and labor inputs from (Vinod, 2008, Sec. 5.2). Table 1 reports the usual correlation matrix R in the upper panel and generalized R^* in the lower panel. We partition these matrices using $r = 2, p = 4$, as shown in eq. (4).

```
rm(list=ls()) #clean out memory of R new problem
options(prompt = " ", continue = "  ", width = 68)
print(date())
labor=c(17703, 18552, 18814, 13490, 13390, 12710, 13100,
13350, 14280, 14710, 13290, 12560)
cap=c(193649, 227548, 206165, 261888, 260120, 158990,
262540, 286690, 307320, 309180, 302420, 304810)
wool=c(228091, 233309, 232258, 235807, 241284, 242177,
239101, 243713, 259939, 166563, 261249, 248538)
mutton=c(11416, 14304, 16321, 16255, 16553, 16328, 15292,
14495, 15528, 16239, 17536, 17171)
labor=ts(labor, frequency=1, start=c(1951,1))
cap=ts(cap, frequency=1, start=c(1951,1))
wool=ts(wool, frequency=1, start=c(1951,1))
mutton=ts(mutton, frequency=1, start=c(1951,1))
nn=10000  #used to re-scale all large data numbers
cap=cap/nn; labor=labor/nn; wool=wool/nn; mutton=mutton/nn
mtx=cbind(wool,mutton,cap,labor)
mtx2=apply(mtx,2,log)#log all variables
```

Note that `mtx2` contains data on logs of wool, mutton, capital, and labor inputs. Now, we compute the two correlation matrices.

Table 1: Correlation coefficients r_{ij} in the upper panel and their generalized values $r^*_{i|j}$ in the lower panel use logs of all variables

	wool	mutton	capital	labor
wool	1.0000	0.0883	-0.0622	-0.2411
mutton	0.0883	1.0000	0.3745	-0.5381
capital	-0.0622	0.3745	1.0000	-0.3794
labor	-0.2411	-0.5381	-0.3794	1.0000
wool	1.0000	0.5153	-0.0331	-0.2464
mutton	0.1280	1.0000	0.3769	-0.8974
capital	-0.6573	0.6023	1.0000	-0.6228
labor	-0.9621	-0.4929	-0.9994	1.0000

The R functions to compute CCA using correlation coefficients included only in the new version (>1.1.5) is called `canonRho()`.

```
library(generalCorr)
options(np.messages=FALSE)
library(xtable)
xtable(cor(mtx2),digits=4)
g1=gmcmtx0(mtx2) #define generalized corr matrix as g1
xtable(g1,digits=4)
```

Note that the correlation coefficient between log wool and log mutton is only 0.0883 in the upper panel of Table 1. The lower panel has the larger magnitudes with $r^*_{1|2} = 0.5153$ and $r^*_{2|1} = 0.1280$ when nonlinear fits are considered.

Now we are ready to compute CCA coefficient vectors and the value of the maximized eigenvalue ρ^2 leading to a measure of dependence between two sets ρ as the positive square root of the maximized eigenvalue.

```
c1=canonRho(cor(mtx2),verbo=TRUE)
print("Pearson corr. mtx version")
print(c1)
c2=canonRho(g1,verbo=TRUE)
print("gmcmtx0 generalized corr. mtx version")
print(c2)
c3=canonRho(symmze(g1),verbo=TRUE)
print("symmetrized gmcmtx0 generalized corr. mtx version")
print(c3)#R* version
c4=canonRho(g1,verbo=TRUE, ridg=c(0.01,0.01))
print("gmcmtx0 generalized corr. mtx with ridge regularization version")
print(c4) #not symmetrized but generalized R*
c5=canonRho(symmze(g1),verbo=TRUE, ridg=c(0.01,0.01))
print("gmcmtx0 generalized and symmetrized corr. mtx with ridge regularization version
```

```
print(c5)#symmetrized R* ridge version
fbet=rbind(c1$bet,c2$bet,c3$bet, c4$bet, c5$bet)
falp=cbind(c1$alp,c2$alp,c3$alp, c4$alp, c5$alp)
frho=rbind(c1$rho,c2$rho,c3$rho, c4$rho, c5$rho)
ans=cbind(t(falp),fbet,frho)

colnames(ans)=c("LnWool", "LnMutton", "LnCap", "LnLab","rho")
rownames(ans)=c("CCA","genCCA","SymGenCCA","GenRidgCCA","SymGenRidgCCA")
print(ans)
xtable(ans,digits=4)
```

We report the output of above code in Table 2. If eigenvalues are imaginary numbers for the generalized correlation matrix R^* denoted in the R code as g1, the coefficient estimates for α, β are not meaningful and we need to use the `symmze()` function to create a symmetric matrix of generalized correlation coefficients and then use `canonRho()`.

We report α, β estimates of all five versions of CCA along rows of Table 2. Since eigenvector signs are arbitrary, our reported values change all signs to have a majority of positive signs. The row labeled 'CCA' refers to the Hotelling version. The row labeled 'genCCA' uses the R^* matrix from the lower panel of Table 1. The row labeled 'SymGenCCA' reports the results for R^d defined in eq. (1), using symmetrized R^* without any ridge-type adjustment. The row labeled 'GenRidgCCA' contains the version with the ridge-type adjustment with $\lambda = 0.01$ for both LHS and RHS. The row labeled 'SymGenRidgCCA' reports the symmetrized R^d version with the ridge-type adjustment with $\lambda = 0.01$ for both LHS and RHS. Logs of variables are denoted by the prefix 'Ln'.

Note that the symmetrized R^d based on nonlinear dependence defined in eq. (1) needs the R function `symmze()`. This operation is not really needed for the wool-mutton example since eigenvalues are real. It is meant for the imaginary eigenvalue cases and the rho values reported in the last column of Table 2 do not have the usual interpretation, since the symmetry of R^d matrix is an artificial construct.

This section has demonstrated the feasibility of generalized CCA and its potential benefit in improving the dependence index as well as the fit based on estimated α, β weights on the two sets of variables.

Table 2: Estimated coefficients α_1, α_2 and β_1, β_2 for Hotelling CCA, generalized CCA, and a version of the generalized CCA using ridge-type adjustment with $\lambda = 0.01$

	LnWool	LnMutton	LnCap	LnLab
CCA	-0.2779	-1.0329	0.2663	-0.9639
genCCA	-0.4466	0.0412	0.6952	0.7189
SymGenCCA	0.0270	-0.0099	-0.7071	-0.7071
GenRidgCCA	-0.4510	0.0404	0.6946	0.7194
SymGenRidgCCA	0.0853	-0.0311	-0.7074	-0.7068

We do not claim that the new Lagrangian for asymmetric generalized correlation matrices

is always useful. Since this is an active area of research in many fields of science, we encourage readers to try our algorithm 'canonRho' on their data sets.

5 Final Remarks

For perfectly dependent $X_j = sin(X_i)$ example, Pearson's product-moment correlation coefficient developed for the computing facilities of the 1890s, $r_{ij} = -0.17$, underestimates the absolute magnitude of dependence by 83%. The generalized correlation coefficient, designed for the modern computing environment, $r^*_{j|i} = -1$, correctly reveals the perfect dependence of the two series, even if the dependence is nonlinear. There is little reason to live with potentially vast underestimation in ubiquitous applications of correlation coefficients. Hence readily computed generalizations of Pearson correlation and partial correlation coefficients available in a free R package 'generalCorr' are worthy of consideration.

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.

Our generalization of the Granger-causality test statistic as the difference between two R^2 values of two flipped kernel regressions allow for nonlinear and nonparametric causal dependence between two time series. The function 'bootGcRsq' is shown to be easy to use.

We have analytically derived a generalization of Hotelling's canonical correlation analysis (CCA) to allow nonlinearities. Our example shows that the function 'canonRho' to implement the new CCA can be readily implemented. We are unable to demonstrate its practical usefulness at this time.

References

Garcia-Medina, A. and Gonzalez, F. G. (2020), "Transfer entropy as a variable selection methodology of cryptocurrencies in the framework of a high dimensional predictive model," *PLoS ONE*, 15(1), URL `https://doi.org/10.1371/journal.pone.0227269`.

Granger, C. W. J. (1969), "Investigating Causal Relations by Econometric Methods and Cross Spectral Methods," *Econometrica*, 37, 424–438.

Hansen, B. E. (2020), *Econometrics*, Madison, WI: Department of Economics, University of Wisconsin, URL `http://www.ssc.wisc.edu/~bhansen/econometrics/Econometrics.pdf`.

Hayfield, T. and Racine, J. S. (2008), "Nonparametric Econometrics: The np Package," *Journal of Statistical Software*, 27, 1–32, URL `http://www.jstatsoft.org/v27/i05/`.

Hotelling, H. (1936), "Relations Between Two Sets of Variables," *Biometrika*, 28, 321–327.

Tuzhilina, E., Tozzi, L., and Hastie, T. (2020), "Canonical Correlation Analysis in high dimensions with structured regularization," *arXiv:2011.01650*, stat.ME, 467–479, URL https://arxiv.org/abs/2011.01650.

Vinod, H. D. (1976), "Canonical ridge and econometrics of joint production," *Journal of Econometrics*, 4, 147–166.

— (2008), *Hands-on Intermediate Econometrics Using R: Templates for Extending Dozens of Practical Examples*, Hackensack, NJ: World Scientific, ISBN 10-981-281-885-5, URL https://www.worldscientific.com/worldscibooks/10.1142/6895.

— (2011), *Hands-on Matrix Algebra Using R: Active and Motivated Learning with Applications*, Hackensack, NJ: World Scientific, ISBN 978-981-4313-68-1, URL http://www.worldscibooks.com/mathematics/7814.html.

— (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, available 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.

Zheng, S., Shi, N.-Z., and Zhang, Z. (2012), "Generalized Measures of Correlation for Asymmetry, Nonlinearity, and Beyond," *Journal of the American Statistical Association*, 107, 1239–1252.