Can we ”infer” causality with a single linear model in the case of a non-linear causal path?

Timothée Bonnet

November 17, 2014

After the talk of this pervert mouse rapist, we were wondering whether a linear model is enough to identify correctly causality when there are direct and indirect effects of a cause.

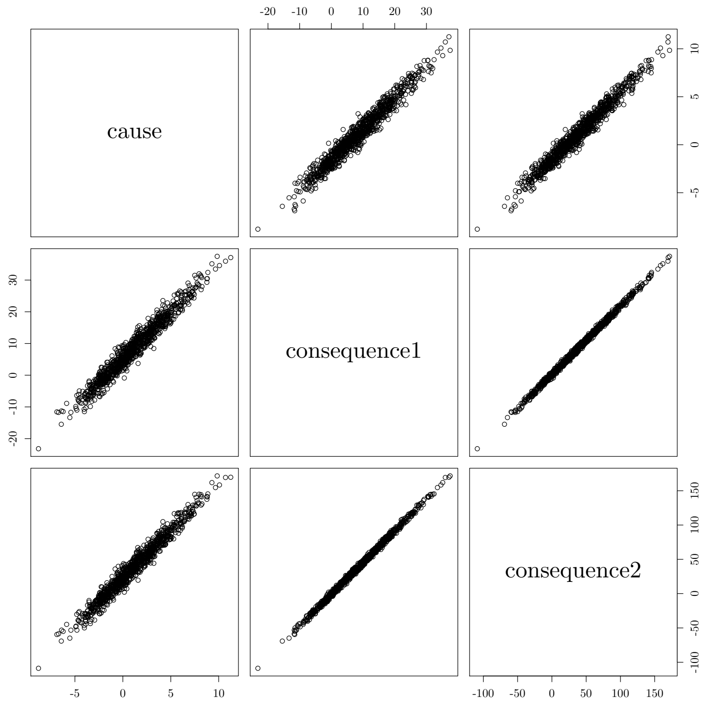
Let’s figure this out, by first creating some data using this function:

```r
fiat_datum<-function(AB=3,BC=4,AC=2,nbpt=1000,noise=0, nastiness=0)
{
unobservedProcess1<-runif(n=nbpt,min = -1,max=1)
unobservedProcess2<-runif(n=nbpt,min = -1,max=1)
cause<-rnorm(n = nbpt,mean=1,sd=3)
consequence1<-(5+cause*AB+rnorm(nbpt,mean=0,sd = 2)-noise*1.5*unobservedProcess1)^(1+nastiness)
consequence2<-2+cause*AC+consequence1*BC+rnorm(nbpt,mean=0,sd=2)+noise*10*unobservedProcess2

df<-data.frame(cause, consequence1,consequence2)
}

df<-fiat_datum()
```

When we keep it very simple, it looks like this:



The idea behind it, is that a cause A affects a variable B and a variable C, while B also affects C.

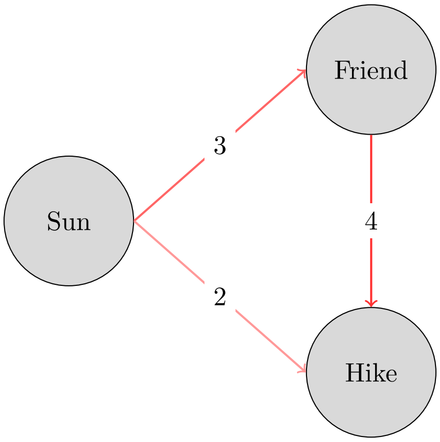


Figure 1: Causal path between the possibility to go hiking during the weekend, the weather, and a lovely friend going hiking. The arrows indicate the direction of causality. Their color and the number in their middle indicates the slope beween the variables.

Can we find back the causal coefficients Sun → Hike and Friend → Hike, using only a multiple regression?

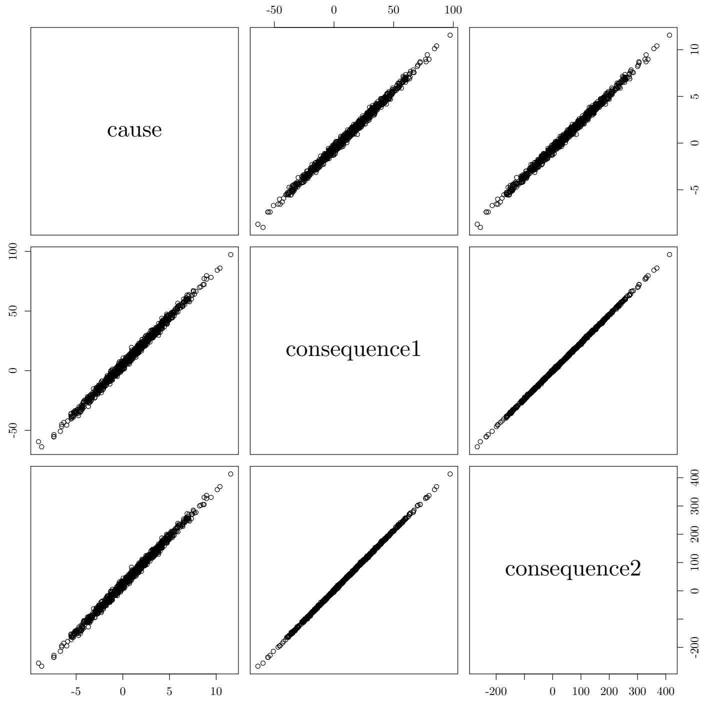
```
m0<-lm(consequence2~1+cause+consequence1,data=df)
summary(m0)

##
## Call:
## lm(formula = consequence2 ~ 1 + cause + consequence1, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.597 -1.331 -0.050  1.362  6.336
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.28725    0.17783   12.86   <2e-16 ***
## cause         2.05100    0.09851   20.82   <2e-16 ***
## consequence1  3.96755    0.03231  122.78   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.992 on 997 degrees of freedom
## Multiple R-squared:  0.9979,Adjusted R-squared:  0.9978
## F-statistic: 2.316e+05 on 2 and 997 DF,  p-value: < 2.2e-16
```

Yes, the two independent variables do control one each other, the estimation is not biased. However, maybe the accuracy of the estimation is affected by the correlation between them.

Let's generate data with a stronger dependency between the cause and the first consequence, keeping the other two causal path constants.

```
df2<-fiat_datum(AB=8)
plot(df2)
```

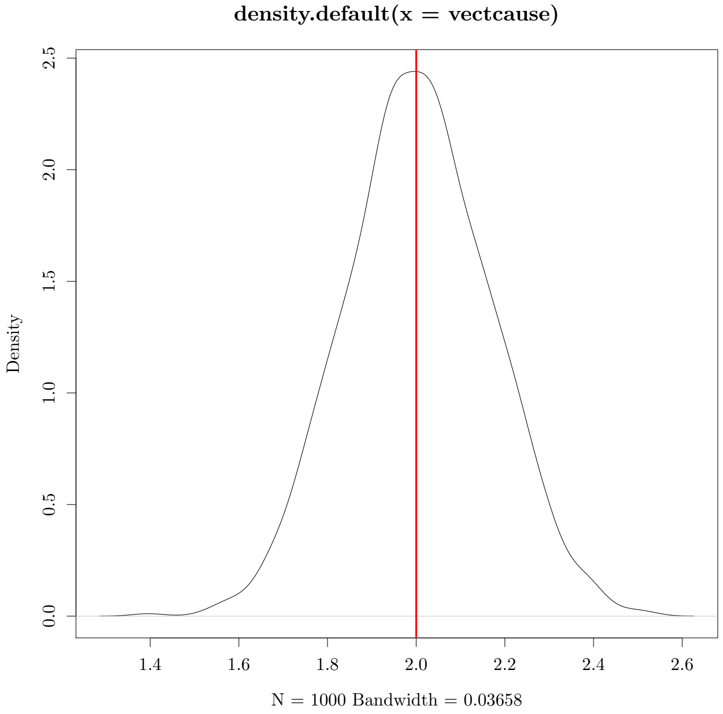


```
m1<-lm(consequence2~1+cause+consequence1,data=df2)
summary(m1)
```

```
##
## Call:
## lm(formula = consequence2 ~ 1 + cause + consequence1, data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3074 -1.3492 -0.0112  1.4763  6.9590
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.23758    0.17646   12.68   <2e-16 ***
## cause         2.36869    0.26173    9.05   <2e-16 ***
## consequence1  3.95427    0.03274  120.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.006 on 997 degrees of freedom
## Multiple R-squared:  0.9996,Adjusted R-squared:  0.9996
## F-statistic: 1.371e+06 on 2 and 997 DF,  p-value: < 2.2e-16
```

The estimate does not seem biased, let's check it seriously:

```
vectcause<-replicate(n = 1000,expr = {lm(consequence2~1+cause+consequence1,
                                         data=fiat_datum(AB=5))$coefficients["cause"]})
plot(density(vectcause))
abline(v=2,lwd=3,col="red")
```



```
mean(vectcause);sd(vectcause)/(sqrt(1000))

## [1] 2.004635
## [1] 0.005116896
```

Which seems very reasonable.

However, the accuracy is lower. Interestingly, only the direct path estimation (the variance, on the slot [”cause”;”cause”]) is blured by the correlation between the cause and the first consequence:

```
vcov(m0)

##              (Intercept)        cause consequence1
## (Intercept)   0.031624712  0.015386584 -0.005322435
## cause         0.015386584  0.009704506 -0.003112043
## consequence1 -0.005322435 -0.003112043  0.001044205

vcov(m1)

##              (Intercept)        cause consequence1
## (Intercept)   0.031139826  0.042393309 -0.005362294
## cause         0.042393309  0.068504917 -0.008542076
## consequence1 -0.005362294 -0.008542076  0.001071723
```

Probably, modelling explicitly the causal path would help to deal with the loss of accuracy.

I tried with a simple bayesian model. For some reasons, KnitR does not want to write this sink on the source folder so the model must be put by hand in a .txt file.

```
library(R2jags)

## Loading required package:  rjags
## Loading required package:  coda
## Loading required package:  lattice
## Linked to JAGS 3.4.0
## Loaded modules:  basemod,bugs
##
## Attaching package:  ’R2jags’
##
## The following object is masked from ’package:coda’:
##
##    traceplot
```

```
#
# sink("simpleB.txt")
# cat("
#     model {
#     #priors and constraints
#     AB~dunif(-10,10)
#     AC~dunif(-10,10)
#     BC~dunif(-10,10)
#     intB~dunif(-10,10)
#     intC~dunif(-10,10)
#     tauB<-pow(sdB,-2)
#     tauC<-pow(sdC,-2)
#     sdB~dunif(0,10)
#     sdC~dunif(0,10)
#     #likelihood
#     for (i in 1:S)
#       {
#         oB[i]~dnorm(B[i],tauB)
#         B[i]<-intB+AB*oA[i]
#         oC[i]~dnorm(C[i],tauC)
#         C[i]<-intC+AC*oA[i]+BC*oB[i]
#       }
#
#     }
#     ",fill = TRUE)
# sink()
```

If I run it for the first (not too correlated) data set:

```
dataB<-list(oB=df$consequence1,oA=df$cause,S=nrow(df),oC=df$consequence2)
initsB <- function() list(AB =runif(n=1,-10,10),AC=runif(n=1,-10,10),BC=runif(n=1,-10,10))
paramsB <- c("AB","AC","BC")
# MCMC settings
ni <- 5000 ; nt <- 10 ; nb <- 1000 ; nc <- 3
B1<-jags(dataB,initsB,paramsB,"simpleB.txt",
         n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, working.directory = getwd())
```

```
## module glm loaded
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
##    Graph Size: 8015
##
## Initializing model
```

```
print(B1)

## Inference for Bugs model at "simpleB.txt", fit using jags,
##  3 chains, each with 5000 iterations (first 1000 discarded), n.thin = 10
##  n.sims = 1200 iterations saved
##           mu.vect sd.vect     2.5%      25%      50%      75%    97.5%
## AB          2.980   0.020    2.942    2.967    2.979    2.994    3.020
## AC          2.044   0.105    1.843    1.970    2.049    2.119    2.250
## BC          3.969   0.035    3.902    3.946    3.968    3.994    4.036
## deviance 8393.553   3.731 8388.144 8390.806 8392.850 8395.766 8401.866
##           Rhat n.eff
## AB       1.001  1200
## AC       1.005   350
## BC       1.005   360
## deviance 1.000  1200
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 7.0 and DIC = 8400.5
## DIC is an estimate of expected predictive error (lower deviance is better).
```

And now for the second (much more correlated) data set:

```
dataB<-list(oB=df2$consequence1,oA=df2$cause,S=nrow(df2),oC=df2$consequence2)
initsB <- function() list(AB =runif(n=1,-10,10),AC=runif(n=1,-10,10),BC=runif(n=1,-10,10))
paramsB <- c("AB","AC","BC")
# MCMC settings
ni <- 5000 ; nt <- 10 ; nb <- 1000 ; nc <- 3
B2<-jags(dataB,initsB,paramsB,"simpleB.txt",
         n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, working.directory = getwd())

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
##    Graph Size: 8015
##
## Initializing model

print(B2)

## Inference for Bugs model at "simpleB.txt", fit using jags,
##  3 chains, each with 5000 iterations (first 1000 discarded), n.thin = 10
##  n.sims = 1200 iterations saved
##           mu.vect sd.vect     2.5%      25%      50%      75%    97.5%
## AB          7.970   0.020    7.932    7.956    7.969    7.983    8.009
## AC          2.322   0.241    1.844    2.187    2.323    2.468    2.856
## BC          3.960   0.030    3.895    3.942    3.960    3.977    4.020
## deviance 8395.541   3.822 8390.197 8392.721 8394.981 8397.561 8404.895
##           Rhat n.eff
## AB       1.004   470
## AC       1.091    37
## BC       1.107    34
## deviance 1.011   230
```

```
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 7.3 and DIC = 8402.8
## DIC is an estimate of expected predictive error (lower deviance is better).
```

The uncertainty around AC is much larger in the second case... so it does not help.

hmmm, so there is nothing to do about this correlation between the causal paths?