Latent variable model and identifiability

Brice Ozenne

March 8, 2021

1 A necessary condition for identifiability

One way to assess identifiability of a model is to count the number of parameters vs. the number of sufficient statistics brought by the data. If the number of parameters in the model exceed the number of sufficient statistics brought by the data the model is not identifiable.

1.1 Example in univariate linear models

Assuming normaly distributed variables:

```
set.seed(10)
data <- data.frame(Y = rnorm(10),
     X = rnorm(10))
```

the sufficient statistics are the mean, variance, and covariance. This means that knowing the mean and the variance, I can simulate new data following the same law as my observed data. When considering only the outcome, I need at least two observations to fit a linear model:

```
## can only estimate the mean of Y
eY.1 <- lm(Y ~ 1, data = data[1,, drop = FALSE])
summary(eY.1)$coef
sigma(eY.1)
```

```
               Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.01874617        NaN     NaN      NaN
[1] NaN
```

```
## can estimate both the mean and variance of Y
eY.2 <- lm(Y ~ 1, data = data[1:2,, drop = FALSE])
summary(eY.2)$coef
sigma(eY.2)
```

```
               Estimate Std. Error    t value  Pr(>|t|)
(Intercept) -0.08275319  0.1014994 -0.8153075 0.5645487
[1] 0.1435418
```

If I also want to adjust on X, I now also need to estimate the covariance between X and Y. So I need at least one more observation:

```
## can only estimate the mean of Y and its covariance with X
eXY.2 <- lm(Y ~ X, data = data[1:2,, drop = FALSE])
summary(eXY.2)$coef
sigma(eXY.2)
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.6276732        NaN     NaN      NaN
X            0.5867049        NaN     NaN      NaN
[1] NaN
```

```
## can estimate the mean, variance of Y and its covariance with X
eXY.3 <- lm(Y ~ X, data = data[1:3,, drop = FALSE])
summary(eXY.3)$coef
sigma(eXY.3)
```

```
             Estimate Std. Error    t value   Pr(>|t|)
(Intercept) -1.091006 0.09766722 -11.170647 0.05683890
X            1.072162 0.12464616   8.601644 0.07368065
[1] 0.1226233
```

2 Application to latent variable models (lvm)

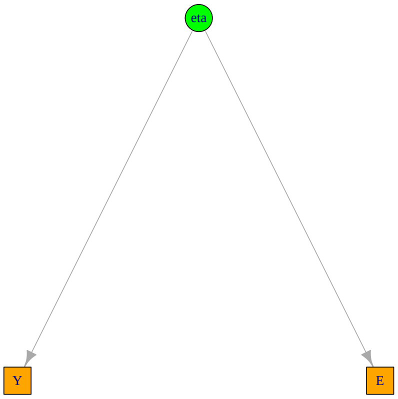
In a latent variable model things are simular. Because we are interested in modeling the relationship between variables, usually we focus on the covariance matrix: does the observed covariance matrix enable to identify the modeled covariance matrix?

2.1 Example 1: bivariate lvm

Consider the following model:

```
library(lava)

lvm.2Y <- lvm(c(Y, E) ~ eta)
latent(lvm.2Y) <- ~ eta
```



This model involves 3 variables (2 observed and 1 latent). We can write the (full) **mean vector** and **variance-covariance matrix** between all the variables:

$$\mu = \begin{bmatrix} \mu_Y \\ \mu_\eta \\ \mu_E \end{bmatrix} = \begin{bmatrix} \mathbb{E}\left[Y\right] \\ \mathbb{E}\left[\eta\right] \\ \mathbb{E}\left[E\right] \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma_{Y,Y} & \sigma_{Y,\eta} & \sigma_{Y,E} \\ & \sigma_{\eta,\eta} & \sigma_{\eta,E} \\ & & \sigma_{E,E} \end{bmatrix} = \begin{bmatrix} \mathbb{V}ar(Y) & \mathbb{C}ov(Y,\eta) & \mathbb{C}ov(Y,E) \\ & \mathbb{V}ar(\eta) & \mathbb{C}ov(\eta,E) \\ & & \mathbb{V}ar(E) \end{bmatrix}$$

Since we don't observed η, we cannot estimate its mean and variance. A common convention is to set its mean to 0 and variance to 1 [1]. Using this latent variable model, we also assumed here that Y is independent of E given η (i.e. $\mathbb{C}ov(Y,\eta) = \frac{\mathbb{C}ov(Y,\eta)\mathbb{C}ov(E,\eta)}{\mathbb{V}ar[\eta]}$). So at the end of the day, we only have 4 parameters to estimate:

$$\theta = (\mu_Y, \mu_E, \sigma_{Y,Y}, \sigma_{Y,\eta}, \sigma_{\eta,E}, \sigma_{E,E})$$

[1] By default, `lava` do something else: it sets the mean of Y, the reference outcome, to 0 and fix its covariance such that $\mathbb{C}ov(Y,\eta) = \mathbb{V}ar(\eta)$

The **empirical mean vector** contains two parameters but the **empirical variance-covariance matrix** only contains three different parameters:

$$m = \begin{bmatrix} \overline{Y} \\ \overline{E} \end{bmatrix} \qquad S = \begin{bmatrix} s_{Y,Y} & s_{Y,E} \\ & s_{E,E} \end{bmatrix}$$

We can check that in R:

```
n <- 1e3
df.2Y <- sim(lvm.2Y, n, latent = FALSE)
cbind(mu=colMeans(df.2Y), vcov = cov(df.2Y))
```

```
           mu         Y         E
Y -0.01663862 2.0403977 0.9651849
E  0.06287055 0.9651849 1.9554924
```

Overall:

- ✔ for the mean parameters: the full expectation vector would contain 3 parameters, one for each variable. We only observe two of them (Y and E) and by default lava fix the intercept of Y to be 0 so there are only two mean parameters.

- ✘ for the variance-covariance parameters: we have 6 parameters to estimate (3 variances, 3 covariances) which after applying some necessary restriction reduces to 4 parameters. However we only observe 3 moments. The model is therefore not identifiable.

This means that the lvm won't properly converge

```
estimate(lvm(Y ~ eta, E ~ eta, eta[0:1] ~ 1),
  data = df.2Y)
```

```
                    Estimate Std. Error  Z-value  P-value
Measurements:
   Y~eta             1.03678    0.04082 25.39900   <1e-12
   E~eta             0.93001    0.04310 21.57969   <1e-12
Intercepts:
   Y                -0.01664    0.04515 -0.36853   0.7125
   E                 0.06287    0.04420  1.42245   0.1549
Residual Variances:
   Y                 0.96344    0.04249 22.67723
   E                 1.08861    0.05162 21.08771
Warning messages:
1: In estimate.lvm(lvm(Y ~ eta, E ~ eta, eta[0:1] ~ 1), data = df.2Y) :
  Near-singular covariance matrix, using pseudo-inverse!
2: In print.lvmfit(x) : Small singular value: 0
3: In print.lvmfit(x) : Singular covariance matrix. Pseudo-inverse used.
```

The non identifiability come from the fact that the only equation defining the parameters $\sigma_{E,eta}$ and $\sigma_{\eta,\eta}$ is:

$$\mathbb{C}ov(Y,\eta) = \frac{\mathbb{C}ov(Y,\eta)\mathbb{C}ov(E,\eta)}{\mathbb{V}ar\left[\eta\right]} = \mathbb{C}ov(Y,\eta)\mathbb{C}ov(E,\eta)$$

which is not identifiable because we only observe $\mathbb{C}ov(Y,\eta)$ so $\mathbb{C}ov(Y,\eta)$ and $\mathbb{C}ov(E,\eta)$ can take any value as soon as their product remain constant and equal to $\mathbb{C}ov(Y,\eta)$. One solution is to constraint them to be equal:

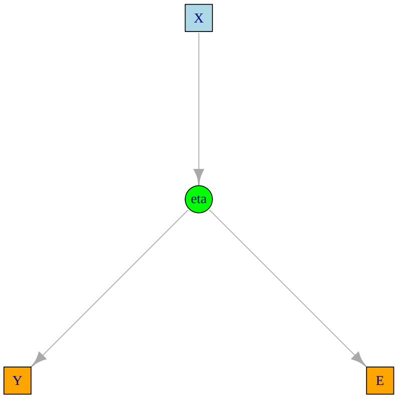
```
estimate(lvm(Y ~ lambda*eta, E ~ lambda*eta, eta[0:1] ~ 1),
  data = df.2Y)
```

```
                    Estimate Std. Error  Z-value  P-value
Measurements:
   Y~eta             0.98195    0.03569 27.51623   <1e-12
Intercepts:
   Y                -0.01664    0.04515 -0.36853   0.7125
   E                 0.06287    0.04420  1.42245   0.1549
Residual Variances:
   Y                 1.07414    0.07321 14.67183
   E                 0.98932    0.07078 13.97739
```

2.2 Example 2: bivariate lvm with group effect

Let's modify the previous model by adding an exogenous variable affecting the latent variable:

```
lvm.2YX <- lvm(Y[0.5:0.5] ~ eta, E[1.25:2] ~ 3*eta,
        eta[0:1] ~ 0.25*X, X[0:1] ~ 1)
latent(lvm.2YX) <- ~ eta
```



This model involves 4 variables (3 observed and 1 latent). We can write the (full) **mean vector** and **variance-covariance matrix** between all the variables:

$$\mu = \begin{bmatrix} \mu_Y \\ \mu_\eta \\ \mu_E \\ \mu_X \end{bmatrix} = \begin{bmatrix} \mathbb{E}\left[Y\right] \\ \mathbb{E}\left[\eta\right] \\ \mathbb{E}\left[E\right] \\ \mathbb{E}\left[X\right] \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma_{Y,Y} & \sigma_{Y,\eta} & \sigma_{Y,E} & \sigma_{Y,X} \\ & \sigma_{\eta,\eta} & \sigma_{\eta,E} & \sigma_{\eta,X} \\ & & \sigma_{E,E} & \sigma_{E,X} \\ & & & \sigma_{X,X} \end{bmatrix} = \begin{bmatrix} \mathbb{V}ar(Y) & \mathbb{C}ov(Y,\eta) & \mathbb{C}ov(Y,E) & \mathbb{C}ov(Y,X) \\ & \mathbb{V}ar(\eta) & \mathbb{C}ov(\eta,E) & \mathbb{C}ov(\eta,X) \\ & & \mathbb{V}ar(E) & \mathbb{C}ov(E,X) \\ & & & \mathbb{V}ar\left[X\right] \end{bmatrix}$$

As before we will constrain the mean of the latent variable to be 0 and its variance to be 1. Furthermore, conditional on the latent variable, Y, E, and η are independent of each other. So $\mathbb{C}ov(Y,E)$, $\mathbb{C}ov(Y,X)$, and $\mathbb{C}ov(E,X)$ are not "real" parameters. So we only need to estimate 9 parameters:

$$\theta = (\mu_Y, \mu_E, \mu_X, \sigma_{Y,Y}, \sigma_{E,E}, \sigma_{X,X}, \sigma_{Y,\eta}, \sigma_{E,\eta}, \sigma_{\eta,X})$$

The **empirical mean vector** and **empirical variance-covariance matrix** also contain 9 parameters:

$$m = \begin{bmatrix} \overline{Y} \\ \overline{E} \\ \overline{X} \end{bmatrix} \qquad S = \begin{bmatrix} s_{Y,Y} & s_{Y,E} & s_{Y,X} \\ & s_{E,E} & s_{E,X} \\ & & s_{X,X} \end{bmatrix}$$

or in R:

```
set.seed(10)
df.2YX <- sim(lvm.2YX, n = 1e4, latent = FALSE)
cbind(mu=colMeans(df.2YX), vcov = cov(df.2YX))
```

```
           mu         Y          E         X
Y  0.47355535 1.5787540  3.2020009 0.2519949
E  1.17255908 3.2020009 11.5908969 0.7315259
X -0.02028518 0.2519949  0.7315259 0.9781732
```

So the model satisfy one necessary condition for being identifiable. This condition is however not sufficient to ensure identifiability but is easier to check than the NSC (nessary and sufficient condition). To check the NSC we need to write down the equations relating the empirical and the theoretical moments. To make things a little simpler we will assume that X has mean 0 and variance 1. We obtain the model:

$$\begin{aligned}
Y &= \mu_Y + \sigma_{Y,\eta}\eta + \varepsilon_Y \qquad \varepsilon_Y \sim \mathcal{N}\left(0, \sigma_{Y,Y}\right) \\
E &= \mu_E + \sigma_{E,\eta}\eta + \varepsilon_E \qquad \varepsilon_Y \sim \mathcal{N}\left(0, \sigma_{E,E}\right) \\
\eta &= \sigma_{\eta,X}X + \xi_\eta \qquad \varepsilon_\eta \sim \mathcal{N}\left(0, 1\right)
\end{aligned}$$

we get:

$$\begin{cases}
s_{Y,Y} = \sigma^2_{Y,\eta}(1+\sigma^2_{\eta,X}) + \sigma_{Y,Y} \\
s_{E,E} = \sigma^2_{E,\eta}(1+\sigma^2_{\eta,X}) + \sigma_{E,E} \\
s_{Y,E} = \sigma_{Y,\eta}\sigma_{E,\eta}(1+\sigma^2_{\eta,X}) \\
s_{Y,X} = \sigma_{Y,\eta}\sigma_{\eta,X} \\
s_{E,X} = \sigma_{E,\eta}\sigma_{\eta,X}
\end{cases}
\qquad
\begin{cases}
\sigma_{\eta,X} = \sqrt{\frac{s_{Y,X}s_{E,X}}{s_{Y,E}-s_{Y,X}s_{E,X}}} \\
\sigma_{Y,\eta} = \frac{s_{Y,X}}{\sigma_{\eta,X}} \\
\sigma_{E,\eta} = \frac{s_{E,X}}{\sigma_{\eta,X}} \\
\sigma_{Y,Y} = s_{Y,Y} - \sigma^2_{Y,\eta}(1+\sigma^2_{\eta,X}) \\
\sigma_{E,E} = s_{E,E} - \sigma^2_{E,\eta}(1+\sigma^2_{\eta,X})
\end{cases}$$

which we can solve and therefore the model is identifiable. This is confirmed by the fact that lava is able to estimate the model:

```
e.lvm.2XY <- estimate(lvm(Y~eta,E~eta,eta[0:1]~X), data = df.2YX)
e.lvm.2XY
```

```
                    Estimate Std. Error  Z-value  P-value
Measurements:
   Y~eta             1.01882    0.01931 52.76797   <1e-12
   E~eta             2.95758    0.05605 52.76797   <1e-12
Regressions:
   eta~X             0.25286    0.01119 22.59794   <1e-12
Intercepts:
   Y                 0.47878    0.01231 38.90703   <1e-12
   E                 1.18773    0.03324 35.73453   <1e-12
```

```
Residual Variances:
   Y                  0.47569    0.03712 12.81639
   E                  2.29545    0.30930  7.42139
```

We can in fact manually estimat the coefficients (here we do REML estimation instead of ML so the estimated variance will be a bit larger compared to lava):

```
s_YY <- var(df.2YX$Y)
s_EE <- var(df.2YX$E)
s_YX <- cov(df.2YX$Y,df.2YX$X)
s_EX <- cov(df.2YX$E,df.2YX$X)
s_YE <- cov(df.2YX$Y,df.2YX$E)

ratio <- s_EX / (s_YE - s_YX * s_EX)

manual <- c("eta~X" = sqrt(s_YX * ratio),
     "Y~eta" = s_YX/sqrt(s_YX * ratio),
     "E~eta" = s_EX/sqrt(s_YX * ratio),
     "Y~~Y" = s_YY - s_YX^2/(s_YX * ratio) * (1 + s_YX * ratio),
     "E~~E" =  s_EE - s_EX^2/(s_YX * ratio) * (1 + s_YX * ratio)
     )
rbind(manual = manual,
      lava = coef(e.lvm.2XY)[names(manual)]
      )
```

```
           eta~X    Y~eta    E~eta      Y~~Y     E~~E
manual 0.2471585 1.019568 2.959744 0.4757339 2.295681
lava   0.2528586 1.018822 2.957578 0.4756863 2.295452
```

Note that the model is exactly identifiable in the sense that we have exactly the same number of parameters and moments. Adding an additional link between age and one outcome would make the model non identifiable since we would increase by one the number of parameters (p=10) while still having only 9 moments:

```
estimate(lvm(Y~eta+Age,E~eta,eta[0:1]~X), data = df.2YX)
```

```
                    Estimate Std. Error  Z-value  P-value
Measurements:
   Y~eta              1.01882    0.01931 52.76797   <1e-12
   E~eta              2.95758    0.05605 52.76797   <1e-12
Regressions:
   eta~X              0.25286    0.01119 22.59794   <1e-12
Intercepts:
   Age                0.47878    0.01231 38.90703   <1e-12
   E                  1.18773    0.03324 35.73453   <1e-12
Residual Variances:
   Y                  0.36434
   Age               0.11135
   E                 2.29545    0.30930  7.42139
Warning messages:
1: In sqrt(diag(asVar)) : production de NaN
2: In print.lvmfit(x) : Small singular value: 2.409293e-13
```