INSTRUMENTAL VARIABLES IN CANONICAL DEMAND & SUPPLY MODEL

ETHAN LIGON

1. Introduction

The father-son team of P.G. and Sewall Wright provide the earliest examples of instrumental variables strategies (P. G. Wright, 1928; S. Wright, 1934) in economics (and really in any field). S. Wright (1934) provides a very clear discussion (it's still very worth reading) of instrumental variables using both structural equations (anticipating the approach of the Cowles Commission) and graphical methods (anticipating the approach of Pearl and coauthors).

The application developed by S. Wright (1934) had to do with the prices and quantities of hogs. Identification is a problem here even in a (log-) linear model: with data on prices and quantities (p, q) we can estimate only five useful moments $(\mathrm{E}\,p, \mathrm{E}\,q, \mathrm{E}\,pq, \mathrm{E}\,p^2, \mathrm{E}\,q^2)$, but even the simplest model, say,

$$q_D = \alpha p + u; \quad q_S = \beta p + v; \quad q_D = q_S,$$

with (u, v) normally distributed has seven quantities that the Wrights assumed were unknowns in the model $(\alpha, \beta, \mathrm{E}\,u, \mathrm{E}\,v, \mathrm{E}\,uv, \mathrm{E}\,u^2, \mathrm{E}\,v^2)$.

Question

Why are there "only five useful moments"? Why aren't higher order moments involved? Is there additional information that could be exploited here? Under what conditions?

Wright's solution to this problem was to argue that there were additional variables one could observe that bore on the problem. In particular, he claimed that the size of the previous year's corn crop z affected hog *supply*, but not demand, so that $\mathrm{Cov}(v, z) \neq 0$, while $\mathrm{Cov}(u, z) = 0$; he further assumes that $\mathrm{Cov}(u, v) = 0$. These additional moments suffice to achieve identification, in that the data can pin down

Date: April 7, 2024.

the quantities (a combination of parameters and moments) which are left free in the model.

Below, we modify the data-generating process we've described previously so that the supply shock process $v = v(z, w)$. Note the notational convention that indicates that all the randomness in the shock v is derived from the fact that z and w are random. (There's no loss of generality in this, since w isn't further restricted, and in particular could be correlated with z).

2. Data-Generating Process

```python
import numpy as np
import pandas as pd
from scipy.stats import distributions as iid

# Unobservable component of supply shock z
# Can have any distribution one pleases
w = iid.beta(1,2,loc=-iid.beta(1,2).mean()) # Centered for
↪  convenience

# Structural parameters;
(alpha,beta) = (-1,2)
sigma = {'u':1/2,'v':1/3}
mu = {'u':2,'v':-1}

# u,v assumed independent
u = iid.norm(loc=mu['u'], scale=sigma['u'])  # Demand shocks
v = iid.norm(loc=mu['v'], scale=sigma['v'])  # Supply shocks

# Reduced form coefficients
pi = [[-beta/(alpha - beta), -1/(alpha - beta)],
      [ alpha/(alpha - beta), 1/(alpha - beta)]]

# Generate N realizations of system
# Outcomes have columns (p,q,z)
def wright_dgp(N):
    """
    Generate data consistent with Wright (1934) hog demand and
    ↪  supply.

    Returns a pandas dataframe with N observations on (p,q,z),
    ↪  where
    z is understood to be a supply shock.
    """

    # Arrange shocks into an Nx2 matrix
    U = np.c_[u.rvs(N), v.rvs(N)]
    # Matrix product gives [q,p]; label by putting into df
    df = pd.DataFrame(U@pi,columns=['q','p'])

    Udf = pd.DataFrame(U,columns=['u','v']) # For future reference

    # Relate v and z (need not be linear)
    unobserved_shock = w.rvs(N)/10
    df['z'] = (1-unobserved_shock)*np.exp(4*Udf['v'] -
    ↪  unobserved_shock)
    df['Constant'] = 1

    # Include a constant term in both X & Z
    return df[['q']],df[['Constant','p']],df[['Constant','z']]
```

3. Identification

From the data generating process above we explore the use of z as an instrument. Note that in pursuing our estimation strategy we're going to pretend to be entirely ignorant of the funky non-linear, non-separable relationship between z and the "total" supply shock v, simply maintaining the possibility that the two are related. We're also going to pretend that we don't know the parameters (α, β) or the parameters governing the distribution of u and v. Our goal is to devise a strategy that *identifies* these unknown parameters with things we can estimate from data on (q, p, z).

The logic of our identification strategy, then, is that changes in z move the supply curve, while changes in the demand curve are unrelated to these movements. This "unrelation" delivers a form of independence that allows us to estimate the demand curve.

To see why this works algebraically, re-write just the demand schedule as

$$q_D(p^*(u, v), u) = \mu_u + \alpha p^*(u, v) + (u - \mu_u),$$

where we've written $q_D(p^*, u)$ to remind ourselves that the demand schedule is defined as a function of nothing other than the price p and demand shocks u, and written $p^*(u, v)$ to remind ourselves that the market-clearing price can be written as a function of nothing other than demand and supply shocks.

Next, write a *regression* function that mimics the functional form of the model demand schedule, say

$$q = m + ap + e;$$

taking expectations conditional on z,

$$\mathrm{E}(q|z) = m + a\,\mathrm{E}(p|z) + \mathrm{E}(e|z);$$

Conditional on the model assumption that (i) the demand schedule is linear in p and u, we have $e = u - \mu_u$; so conditional on the second model assumption that z is unrelated to u, we conclude that $\mathrm{E}(e|z) = 0$, so that for the regression equation

$$\mathrm{E}(q|z) = m + a\,\mathrm{E}(p|z).$$

Similarly taking conditional expectations of the model expression for the demand schedule, we have

$$\begin{aligned}\mathrm{E}(q_D(p^*(u,v),u)|z) &= \mu_u + \alpha\,\mathrm{E}(p^*(u,v)|z) + \mathrm{E}(u - \mu_u|z)\\ &= \mu_u + \alpha\,\mathrm{E}(p^*(u,v)|z).\end{aligned}$$

Now, the form of the conditional expectation of the regression function exactly matches that of the model equation, so that the model parameters (μ_u, α) are **identified** with the regression parameters (m, a).

4. Estimator

We now want to take our regression equation to realized data. Let

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{e} \qquad \text{where } \boldsymbol{y} = [q],\ \boldsymbol{X} = [\boldsymbol{\imath}, \boldsymbol{p}], \text{ and } \boldsymbol{e} = [\boldsymbol{u} - \bar{\boldsymbol{u}}];$$

finally, let $\boldsymbol{Z} = [\boldsymbol{\imath}, \boldsymbol{z}]$, where all matrices are conformable.

Then in matrix form, we have the regression

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{e} \Rightarrow \boldsymbol{Z}^\top\boldsymbol{y} = \boldsymbol{Z}^\top\boldsymbol{X}\boldsymbol{b} + \boldsymbol{Z}^\top\boldsymbol{e};$$

The final term involves a sum over products ze and the sum of regression error terms e. A (just identified) linear regression will *set* $Z^\top\boldsymbol{e} = 0$. Provided then only that $\boldsymbol{Z}^\top\boldsymbol{X}$ has full (column) rank, a least squares IV estimator of $\boldsymbol{b}$ is

$$\boldsymbol{b} = (\boldsymbol{Z}^\top\boldsymbol{X})^{-1}\boldsymbol{Z}^\top\boldsymbol{y}.$$

5. Estimation

Let's write some code to estimate the parameters of the regression model using the estimator devised above (the "simple IV estimator"):

```python
import numpy as np

def draw_b(N,dgp):
    """
    Generate a random variate $b$ from a sample of $N$ draws from
    ↪  a function dgp.
    """
    y,X,Z =  dgp(N)
    b = pd.Series(np.linalg.solve(Z.T@X,Z.T@y.squeeze()),index=X.columns) # Solve normal eqs

    u = y.squeeze() - (X@b).squeeze()
    sigma2 = np.var(u)

    avarb = sigma2*(X.T@Z)@np.linalg.inv(Z.T@Z)@(Z.T@X).values/N

    ase = pd.Series(np.sqrt(np.diag(avarb)),index=X.columns)

    return b,ase

b,ase = draw_b(10000,wright_dgp)

print(f"b=\n{b.T}")
print()
print(f"sigma(b)=\n{ase}")
```

6. Inference

Now consider the point that the estimator b is a random variable. Under the assumptions of the *model* a Central Limit Theorem applies, so it's asymptotically normal, with

$$\begin{aligned}\text{avar}(b) = \text{E}[(X^\top Z)(Z^\top Z)^{-1}(Z^\top X)]^{-1}(X^\top Z)(Z^\top Z)^{-1}uu^\top(Z^\top Z)^{-1}(Z^\top X) \\ \cdot [(Z^\top X)(Z^\top Z)^{-1}(Z^\top Z)]^{-1},\end{aligned} \tag{1}$$

which in the homoskedastic case simplifies to

$$\text{avar}(b) = \text{E}[(X^\top Z)(Z^\top Z)^{-1}(Z^\top X)]^{-1}\sigma^2.$$

But in any finite sample the just identified linear IV estimator can be feisty. Let's explore using a little Monte Carlo experiment. Let's begin by constructing a slightly more transparent data-generating process, in which Z and X have a linear relationship:

```
from scipy.stats import distributions as iid

def linear_dgp(N,beta,gamma,pi,sigma_u,sigma_v):
    u = iid.norm(scale=sigma_u).rvs(N)
    v = iid.norm(scale=sigma_v).rvs(N)
    Z = iid.norm().rvs(N)

    X = Z*pi + v
    y = X*beta + u
    df = pd.DataFrame({'y':y,'x':X,'z':Z,'Constant':1})

    return df[['y']],df[['Constant','x']],df[['Constant','z']]
```

The next bit of code *repeatedly* draws new random samples and calculates b from them; we then construct a histogram of the resulting estimates.

```
from matplotlib import pyplot as plt

B = pd.DataFrame([draw_b(100,lambda N:
↪  linear_dgp(N,1,0,.01,1,1))[0]['x'] for i in range(1000)])
B.hist(bins=int(np.ceil(np.sqrt(B.shape[0]))))
```

7. Evaluation

Consider the $p-p$ plot of the empirical distribution of our estimates b against the normal distribution:

```
def ppplot(data,dist):
    data = np.array(data)

    # Theoretical CDF, evaluated at points of data
    P = [dist.cdf(x) for x in data.tolist()]

    # Empirical CDF, evaluated at points of data
    Phat = [(data<x).mean() for x in data.tolist()]

    fig, ax = plt.subplots()

    ax.scatter(P,Phat)
    ax.plot([0,1],[0,1],color='r') # Plot 45
    ax.set_xlabel('Theoretical Distribution')
    ax.set_ylabel('Empirical Distribution')
    ax.set_title('p-p Plot')

    return ax

ppplot(B,iid.norm(loc=B.mean(),scale=ase['p']))
```

8. Questions

Question

The above Monte Carlo exercise repeatedly estimates the model using a sample size of 100. Given the specification of the model, what is the asymptotic distribution of b (i.e., of $\sqrt{N}(b - \beta)$)?

Question

The above code repeatedly estimates the model using a sample size of 100. What does the histogram of estimates b suggest about the adequacy of this sample size, when comparing to the asymptotic distribution? Experimenting with the sample size above, what size seems to be adequate?

References

Wright, P. G. (1928). *Tariff on animal and vegetable oils.* Macmillan Company, New York.

Wright, S. (1934). The method of path coefficients. *The annals of mathematical statistics*, *5*(3), 161–215.