Stochastic Frontier Analysis

AREC 705
Alexandra E. Hill

SFA Overview

Output-oriented technical efficiency can be written as:

$$\ln y_i = \ln y_i^* - u_i, \quad u_i \geq 0$$

$$\ln y_i^* = f(\mathbf{x_i}; \beta) + v_i$$

Where u_i is production inefficiency and v_i is a zero-mean random error term.

If we rearrange this equation, we cal see that u_i is the difference between the frontier production and the observed production, i.e. $u_i = \ln y_i^* - \ln y_i$. As u_i approaches zero, the producer is becoming *more efficient*. We can also construct a measure of *efficiency*: $e^{-u_i} = \frac{y_i}{y_i^*}$. Where $e^{-u_i} \cdot 100$ gives the percentage of the maximum output that firm i produces. And $0 < e^{-u_i} \leq 1$

Distribution-Free Approaches

Today we will cover two (of the three) distribution-free approaches for measuring u_i and (later) will use this to introduce the maximum likelihood estimation methods.

These are called distribution-free approaches because they do not impose any structure on the error term (v_i). These models are deterministic (like DEA), meaning that they exclude the error term (v_i).

IO and OO Distribution-Free TE

Recall that the OO production function is:

$$y = f(\mathbf{x})e^{-u}$$

And the IO production function is:

$$y = f(\mathbf{x}e^{-\eta})$$

Then the Cobb-Douglas OO model is:

$$\ln y_i = \beta_0 + \sum_j \beta_j \ln x_j - u_i$$

And the Cobb-Douglas IO model is:

$$\ln y_i = \beta_0 + \sum_j \beta_j \ln x_j - (\sum_j \beta_j)\eta$$

Which is equivalent to the OO model with $u_i = \eta \sum_j \beta_j$

So OO technical inefficiency is equal to u_i and we can get IO technical inefficiency from: $\eta_i = \frac{u_i}{\sum \beta_j}$.

OO technical efficiency is equal to e^{-u_i} and IO technical efficiency is equal to $e^{-\eta_i}$.

Distribution-Free Estimation of SFA

Corrected OLS

Let's assume a Cobb-Douglas production function so that the above measures of OO and IO TE hold. We need to estimate a frontier function that bounds the observations ($\ln y_i$) from above. Corrected OLS (COLS) does this in a two-stage procedure where slope coefficients are estimated and the resulting production function is shifted upward until it bounds all observations in the data.

Formally:

Stage 1: OLS Regression

$$\ln y_i = \hat{\beta}_0 + \mathbf{x_i'}\hat{\tilde{\beta}} + \hat{e}_i$$

Because $E[u_i] \neq 0$, $\hat{\beta}_0$ is a biased estimate of β_0, but $\hat{\tilde{\beta}}$ is a consistent estimate of $\tilde{\beta}$

Stage 2: Adjust the OLS slope intercept upward by the amount of $\max\{\hat{e}_i\}$, so that the adjusted function bounds all observations from above. The residuals of this new estimating equation can now be written as:

$$\hat{e}_i - \max\{\hat{e}_i\} = \ln y_i - \{\left[\hat{\beta}_0 + \max\{\hat{e}_i\}\right] + \mathbf{x_i'}\hat{\tilde{\beta}}\} \leq 0$$

With

$$\hat{u}_i = -(\hat{e}_i - \max\{\hat{e}_i\}) \geq 0$$

Where $\hat{u}_i$ is our measure of OO technical inefficiency, and we can write technical efficiency as: $e^{-\hat{u}_i}$, where $e^{-u_i} \cdot 100$ gives the percentage of the maximum output that firm i produces. And $0 < e^{-u_i} \leq 1$.

To estimate IO technical inefficiency, we can adjust this parameter using the regression coefficients:

$$\hat{\eta}_i = \frac{\hat{u}_i}{\sum_{\forall j} \beta_j}$$

The SFA Package

```
/* install data and ado files from Kumbhakar, S.C. Wang, H-J,
and Horncastle, A.P. (2014) */

net install sfbook_install, ///
from("https://sites.google.com/site/sfbook2014/home/install/") replace

sfbook_install

* Load dataset
use dairy, clear
```

Note that this will install (user written) SFA ado files to your computer in your PLUS directory and will install the accompanying datasets in your c:\sfbook_demo for Windows and /users/c(username)/sfbook_demo for Mac.

COLS in Stata

```
* Let's start with single input, single output
global xvar llabor

* Stage 1: OLS regression
        regress ly $xvar

* recall that the OLS coefficient on llabor is consistent,
* but the _constant is not

* Stage 2: adjust intercept and estimate efficiency
        * store residuals
        predict e, resid

        * get max(resid)
        sum e

        * generate inefficiency and efficiency
        gen double u = -(e -r(max))
        gen double eff = exp(-u)

        sum u eff

* Can plot IO and OO efficiency
        gen eff_io_round = round(eff_io, 0.01)
        gen eff_oo_round = round(eff_oo, 0.01)

        twoway (scatter ly llabor, mlabel(eff_io_round)), ///
        legend(off) ytitle("log of milk production") name(eff_io, replace) ///
```

```
        title("IO Efficiency Scores")

        twoway (scatter ly llabor, mlabel(eff_oo_round)), ///
        legend(off) ytitle("log of milk production") ///
        name(eff_oo, replace) title("OO Efficiency Scores")


* can plot OLS and COLS lines:
        regress ly $xvar

        predict y_hat, xb

        qui sum e
        gen double y_hat_cols = y_hat + r(max)

        twoway (scatter ly llabor, mlabel(eff_io_round)) (line y_hat llabor) ///
        (line y_hat_cols llabor), legend(pos(6) col(2) label(2 "OLS") ///
        label(3 "COLS") order(2 3)) ytitle("log of milk production") ///
        name(cols, replace)
```

Corrected Mean Absolute Deviation

Follow the same two step procedure, but use a regression through the median, rather than mean in the first stage. (CMAD) This is equivalent to running a quantile regression, but rather than running multiple quantile regressions, you will just run one through the middle (medians).

$$Q_{y_i}(\tau|x_i) = \alpha(\tau) + \beta(\tau)x_i$$
$$Q_{y_i}(\frac{1}{2}|x_i) = \alpha_{\frac{1}{2}} + \beta_{\frac{1}{2}}x_i$$

Where

$$\hat{\beta(\tau)} = argmin_{\beta \in \mathbb{R}^p} \sum \rho_{1/2}(y_i - x'\beta)$$

And

$$\rho_\tau(u) = u(\tau - I(u < 0))$$

Evaluated at the median, the quantile regression estimator is equivalent to:

$$\hat{\beta(\tau)} = \min_{\beta \in \mathbb{R}^p} \sum |y_i - x'\beta|$$

CMAD in Stata

```
* Stage 1: quantile regression
        qreg ly $xvar, quant(0.5) /* note that quant(0.5) is default */

* Stage 2: adjust intercept and estimate efficiency
        * store residuals
        predict e_cmad, resid

        * get max(resid)
        sum e_cmad

        * generate inefficiency and efficiency
        gen double u_cmad = -(e_cmad -r(max))
        gen double eff_cmad = exp(-u_cmad)

        sum u_cmad eff_cmad

        * can plot OLS and COLS:
        reg ly $xvar, quant(0.5)

        predict y_hat_mad, xb

        qui sum e_cmad
        gen double y_hat_cmad = y_hat_mad + r(max)

        twoway (scatter ly llabor) (line y_hat_mad llabor) ///
        (line y_hat_cmad llabor), legend(pos(6) col(2) ///
        label(2 "Mean Absolute Deviation") ///
        label(3 "Corrected Mean Absolute Deviation") ///
        order(2 3)) ytitle("log of milk production") name(cmad, replace)
```

Compare COLS and CMAD

```
* compare COLS and CMAD frontiers
        twoway (scatter ly llabor) (line y_hat_mad llabor) ///
        (line y_hat_cmad llabor) (line y_hat llabor) ///
        (line y_hat_cols llabor), legend(pos(6) col(2) ///
        label(2 "Mean Absolute Deviation") ///
        label(3 "Corrected Mean Absolute Deviation") ///
        label(4 "OLS") label(5 "COLS") order(4 5 2 3 )) ///
        ytitle("log of milk production") name(cmad_cols, replace)

* create DEA efficiency score for one-input one-output
        gen ratio = ly/llabor
        qui sum ratio
```

```
        gen eff_dea = ratio/r(max)
        sum eff_dea

* scatter plot with labels of efficiency score (round score so plot looks nicer)
        gen eff_dea_round = round(eff_dea, 0.01)
        twoway scatter ly llabor , legend(off) mlabel(eff_dea_round) ///
        ytitle("log of milk production") name(dea_eff, replace)

* create a line that goes through the origin and the most efficient point
        sum ly if eff_dea==1
        local delta_y = r(max)
        sum llabor if eff_dea==1
        local delta_x = r(max)

        local slope = `delta_y'/`delta_x'

        gen y_hat_dea = `slope'*llabor


* compare all
        twoway (scatter ly llabor) (line y_hat_mad llabor) ///
        (line y_hat_cmad llabor) (line y_hat llabor) ///
        (line y_hat_cols llabor) (line y_hat_dea llabor, lcolor(blue)),///
        legend(pos(6) col(2) label(2 "MAD") label(3 "CMAD") ///
        label(4 "OLS") label(5 "COLS") label(6 "DEA") order(4 5 2 3 6 )) ///
        ytitle("log of milk production") name(cmad_cols_dea, replace)

        * compare measures of efficiency:
                order eff_dea eff_cmad eff farmid
                gsort -eff_dea

        * compare ranksings:
                foreach var of varlist eff eff_cmad eff_dea {
                        gsort -`var'
                        gen rank_`var' = _n
                }

                order eff_dea eff_cmad eff rank_* farmid
                gsort -eff_dea
```

SFA Distribution Approaches and Maximum Likelihood Estimation

Overview of Distribution Approaches in Cross-sectional Data

In cross-sectional data we cannot separately identify u_i and v_i. To do so, we need to make assumptions on the distributions and independence of both. To begin, we will assume that the two distributions are independent (we probably won't get to distributions that are related in this course). v_i is typically assumed to be normally distributed with mean 0. It is not as clear what distribution is most appropriate for u_i.

Before investing a bunch of time in doing this, we might want to first apply a simple "check" to see if it is necessary.

Skewness Test

Schmidt and Lin (1984) propose an OLS residual test for the validity of the SFA specification. We will use this test on the OLS residuals to determine whether the current SFA specification is a good approach.

The basic idea: we have a composite error term $v_i - u_i, u_i \geq 0$ and v_i distributed symmetrically around zero $\rightarrow$ the distribution of the error terms should be negatively skewed. This test constructs a test statistic for skewness of the error terms. *If skewness has the correct sign, we reject H_0 and have evidence of the existence of a one-sided error.*

The tests:

Schmidt and Lin (1984): sample-moment based test

$$\sqrt{b_1} = \frac{m_3}{m_2\sqrt{m_2}},$$

Where m_2 and m_3 are the second and third sample moments of the OLS residuals. If $\widehat{\sqrt{b_1}} < 0$ OLS residuals are skewed to the left, and if $\widehat{\sqrt{b_1}} > 0$ they are skewed to the right. In a somewhat angry Stata journal article Royston (1991) proposes an improvement to this estimator that accounts for both skewness and kurtosis.

Royston, P. 1991. sg3.5: Comment on sg3.4 and an improved D'Agostino test. Stata Technical Bulletin 3: 23-24. Reprinted in Stata Technical Bulletin Reprints, vol. 1, pp. 110-112. College Station, TX: Stata Press.

Coelli (1995) suggests another variant of this test:

$$M3T = \frac{m_3}{\sqrt{\frac{6m_2^3}{N}}}$$

Note that this comes fro the fact that the third moment of the OLS residuals are asymptotically distributed as normal with mean zero and variance $\frac{6m_2^3}{N}$. A convenience of this test is that it relies

on the normal distribution critical values, whereas the critical values in the above are a source of contention.

Skewness Tests in Stata

Let's start with looking at the distribution of OLS residuals:

```
* Look at OLS resids for normality
label var e_oo "OLS Residuals"

* get std. deviation to plot density
sum e_oo
local sd = r(sd)

graph twoway histogram e_oo, bin(100) xlabel(-.6(.2).6)  ///
xtitle("OLS Residuals") legend(pos(6) col(3) label(2 "Normal Distribution")///
label(3 "Kernel Density") order(3 2))|| function normalden(x,0,`sd'), ///
range(-.6 .6) || kdens e_oo

* Formal tests for skewness
sum e_oo, d

* Check that that this is the test statistic we are interested in
qui sum e_oo
local e_mean = r(mean)
egen double m2 = mean((e_oo-`e_mean')^2)
egen double m3 = mean((e_oo-`e_mean')^3)
local sqrt_b1 = m3/(m2*m2^(1/2))
display `sqrt_b1'
* -.73772692
```

So we have the test statistic, but the critical values for this are somewhat controversial. There are two main ones we will use that are both conducted using sktest in stata:

```
sktest e_oo, noadj /* unaltered test */
sktest e_oo /*Royston (1991) altered test */
```

Both tests indicate that we can reject the null of no skewness.

Finally, we can manually estimate the Coelli M3T statistic:

```
* Coelli
local N = _N
local m3t = m3/sqrt((6*(m2^3))/`N')
display `m3t'
* -4.2164605
* compare this with normal distribution critical value of (e.g.) -1.96.
* Confirms our rejection of the null of no skewness
```

Rejecting the Null Hypothesis of no skewness provides support for our current production function specification, which means we should proceed with the ML estimation!

- If we could not reject H0, we might try a different production specification.
- We could also try DEA, but if we find a normal distribution of errors, we should be careful in attributing differences across firms to efficiency rather than random noise.

The Half-Normal Distribution

Aigner (1977)

$$\ln y_i = \ln y_i^* - u_i$$
$$\ln y_i^* = \mathbf{x}'\beta + v_i$$
$$u_i \sim i.i.d.\ N^+(0, \sigma_u^2)$$
$$v_i \sim i.i.d.\ N(0, \sigma_v^2)$$

Where u_i and v_i are independent.

The half-normal distribution can be expressed as either a truncated normal distribution or the absolute value of the normal distribution (this is called a folded zero-mean normal distribution.

For the truncated-normal distribution, assume that a random variable $z \sim N(\mu, \sigma_z^2)$, with probability density function $g(z)$. If it is truncated from above at the point α so that $z \geq \alpha$, then the density function is:

$$f(z) = \frac{g(z)}{1 - \Phi\left(\frac{\alpha-\mu}{\sigma_z}\right)} = \frac{\frac{1}{\sigma_z}\phi\left(\frac{z-\mu}{\sigma_z}\right)}{1 - \Phi\left(\frac{\alpha-\mu}{\sigma_z}\right)}, \quad z \geq \alpha$$

The density function of u_i can be obtained by setting $\mu = \alpha = 0$:

$$f(u_i) = \frac{\frac{1}{\sigma_u}\phi\left(\frac{u_i}{\sigma_u}\right)}{1 - \Phi(0)} = 2(2\pi\sigma^2)^{-\frac{1}{2}} \exp\left(-\frac{u_i^2}{2\sigma^2}\right)$$

The half-normal log likelihood function for each observation can be written:

$$L_i = -\ln\left(\frac{1}{2}\right) - \frac{1}{2}\left(\sigma_v^2 + \sigma_u^2\right) + \ln\phi\left(\frac{\epsilon_i}{\sqrt{\sigma_v^2 + \sigma_u^2}}\right) + \ln\Phi\left(\frac{\mu_{*i}}{\sigma_*}\right)$$

$$\mu_{*i} = \frac{-\sigma_u^2 \epsilon_i}{\sigma_v^2 + \sigma_u^2}$$

$$\sigma_* = \frac{\sigma_v^2 \sigma_u^2}{\sigma_v^2 + \sigma_u^2}$$

And the sum of all L_i gives us our maximization problem.

Note that we cannot restrict the variance parameters to be positive, so to ensure non-negative variance, we set:

$$\sigma_u^2 = \exp(w_u)$$
$$\sigma_v^2 = \exp(w_v)$$

Where w_u and w_v are the unrestricted (constant) parameters estimated in the ML maximization. - In other words, we must transform the estimates of variance as written in the equation above into (guaranteed) non-negative values using exp

In Stata:

```
* Use (user-written) sfmodel command
sfmodel ly, prod dist(h) frontier($xvar) usigmas() vsigmas() show ///
ml max, difficult gradient gtol(1e-5) nrtol(1e-5)
* note that variance parameters are exponentials,
* to get variance parameters:
sf_transform

* Compare with OLS
reg ly $xvar
```

Question: How can we interpret the coefficients in the regressions?

Question: What can we say about returns to scale?

Aside on Maximum Likelihood Estimation in Stata:

- Can usually get estimators to converge faster if provide initial values
- A good place to start with SFA is to use the OLS coefficients (because they should be consistent estimates)

```
* OLS regression
reg ly $xvar
* store coefs as vector
matrix b_ols = e(b)
* look at vector
matrix list b_ols

* Use (user-written) sfmodel command
sfmodel ly, prod dist(h) frontier($xvar) usigmas() vsigmas() show
```

```
* sf_init to set initial values, need inputs for each parameter
sf_init, frontier(b_ols) usigmas(0.1) vsigmas(0.1)

/* Optional: wrapper for ml plot to search for better initial values
before starting
sf_srch, frontier($xvar) usigmas() vsigmas() n(2)

* note that this will flash a bunch of graphs because it is in fact running ml plot
ml plot lcattle
*/

ml max, difficult gradient gtol(1e-5) nrtol(1e-5)
* note that the ml estimation converged in only 9 iterations
* (vs 13 previously)
sf_transform
```

Validation

Central to SFA is that we have a one-sided error term which represents inefficiency. We can use a likelihood ratio test to test for the presence of a one-sided error term.

The general LR test statistic is:

$$-2[L(H_0) - L(H_1)]$$

Where $L(H_0)$ is the log-likelihood value of the restricted (OLS) and unrestricted (SF) model, degrees of freedom = number of restrictions (here 1)

```
* get log likelihood from SFA half normal
sfmodel ly, prod dist(h) frontier($xvar) usigmas() vsigmas()
qui ml max, difficult gradient gtol(1e-5) nrtol(1e-5)
* store log likelihood
scalar ll_hn = e(ll)

* log likelihood from OLS
qui regress ly $xvar
scalar ll_ols = e(ll)

* get LR stat:
display -2*(ll_ols - ll_hn)

* look at critical values
sf_mixtable, dof(1)
```

Our critical value of 16.4262 is well above significance at any conventional level, thus we reject the null hypothesis of no one sided error (i.e. we reject no technical inefficiency)

Estimating Technical Efficiency

Now that we have imposed distributional assumptions on u_i , it is not so straightforward to estimate individual technical inefficiencies. Jondrow et al. (1982) develop a conditional density function. $E(u_i|\epsilon_i)$ Horrace and Schmidt (1996) develop a confidence interval for the (conditional) estimate.

I am not going to go into these details, but recommend reading and citing the following papers if you ever include this in a paper:

Jondrow, J., Lovell, C.A.K., Materov, I.S., and Schmidt, P. (1982) "On the estimation of technical inefficiency in the stochastic frontier production function model," *Journal of Econometrics, 19*: 233-238.

Battese, G.E. and Coelli, T.J. (1988). "Prediction of Firm-Level Technical Efficiencies with a Generalized Frontier production Function and Panel Data," *Journal of Econometrics, 38*: 387-399.

Horrace, W.C. and Schmidt, P. (1996). "Confidence Statements for Efficiency Estimates from Stochastic Frontier Models," *Journal of Productivity Analysis, 7*: 275-282.

Bera, A.K. and Sharma, S.C. (1999). "Estimating Production Uncertainty in Stochastic Frontier Production Function Models," *Journal of Productivity Analysis, 12*: 187-210.

```
* use sf_predict to get efficiency scores
sfmodel ly, prod dist(h) frontier($xvar) usigmas() vsigmas()
qui ml max, difficult gradient gtol(1e-5) nrtol(1e-5)

sf_predict, bc(eff_hn) jlms(u_hn) ci(95)

* look at summary stats
sum eff_hn u_hn
```

How can we interpret these results? How are they different from what we have done so far?

The Half-Normal Distribution with Heteroscedasticity

The above model assumes that variances are constant. (i.e. we have σ_u not $\sigma_{u,i}$. But ignoring heteroscedasticity gives consistent estimates of βs, but inconsistent estimates of technical efficiency. Heteroscedasticity can be parameterized with some observed exogenous variables:

$$\sigma^2_{u,i} = \exp(\mathbf{z'_{u,i}w_u})$$

$$\sigma_{v,i}^2 = \exp(\mathbf{z'_{v,i}} \mathbf{w_v})$$

We can use exogenous variables z_i to (1) construct heteroskedastic error terms that are a function of the exogenous variable, and (2) this implicitly assumes that the exogenous variables are *inefficiency explanatory variables*. We do this through a single step procedure by writing the variance as a function of u_i:

$$E(u_i) = \exp\{\frac{1}{2}\ln(2/\pi) + (\mathbf{z_{u,i}}' w_u)\}$$

```
* half-normal with heterogeneity
* we are going to use the variable comp
* (IT expenditure as a percentage of total expenditure)
* as the exogenous determininant of inefficiency.
* Note this is a somewhat 'leanient' use of the word exogenous :)

sfmodel ly, prod dist(h) frontier($xvar) usigmas(comp) vsigmas() show
sf_init, frontier(b_ols) usigmas(0.1 0.1) vsigmas(0.1)
sf_srch, frontier($xvar) usigmas(comp) n(1) nograph fast
ml max, difficult  gtol(1e-5) nrtol(1e-5)

* recover the variance parameters
sf_transform

* efficiency index (and marginal effect)
sf_predict, bc(eff_hn2) jlms(u_hn2) marginal
gsort -eff_hn2
order eff_hn2 eff_hn
```

A Few Other Models

1. The Truncated-Normal Distribution (with and without heteroscedasticity)

```
* in the above code, replace dist(h) with:
dist(t)
*  (other than that use the same options)
```

2. The Truncated-Normal with Scaling

```
* in the above code, use the following options (with exogenous determinant)
prod frontier($xvar) dist(t) scaling hscale(exogenous_var)///
tau cu vsigmas() show
```

3. Exponential Distribution

```
* in the above code, replace dist(h) with:
```

```
dist(e)
* replace usigmas() with:
estas()
/* this will be etas(exogenous_var) in the heteroskedastic model */
```