Manual of Hung-Jen Wang's Stata Codes

Hung-Jen Wang

wangh@ntu.edu.tw

http://homepage.ntu.edu.tw/~wangh

August 2012

This manual provides explanations of the various Stata commands that perform estimations proposed in the following papers. You should also see the Stata `do` files in the demo folders for how to use them. If the papers/models you are interested in are not listed below, please check my web page and see if the program is available for separate download.

Chen, Y.-T. and Wang, H.-J. (2012) "Centered-Residuals-Based Moment Tests for Stochastic Frontier Models," *Econometric Reviews*, 31(6), 625-53.

Wang, H.-J. and Ho, C.-W. (2010) "Estimating Fixed-Effect Panel Stochastic Frontier Models by Model Transformation," *Journal of Econometrics,* 157 (2), 289-96.

Wang, H.-J. (2003) "A Stochastic Frontier Analysis of Financing Constraints on Investment: The Case of Financial Liberalization in Taiwan," *Journal of Business and Economic Statistics,* 21(3), pp.406-419.

Wang, H.-J. (2002) "Heteroscedasticity and Non-Monotonic Efficiency Effects of a Stochastic Frontier Model," *Journal of Productivity Analysis,* 18, pp.241-253.

Wang, H.-J. and Schmidt, P. (2002) "One-Step and Two-Step Estimation of the Effects of Exogenous Variables on Technical Efficiency Levels," *Journal of Productivity Analysis,* 18, pp.129-44.

Except for `ml max` which is a native Stata command, I write all other commands including `sfmodel`, `sf_init`, `sf_srch`, `sf_predict`, `sf_fixeff`, `sfmtest`, and others.

*Copyright © 2012 by Hung-Jen Wang

Contents

1.1 Cross-Sectional Models

As an overview, the main command of the model estimation is `sfmodel`, which specifies the distribution assumption, and the dependent and independent variables to be included in the model. After that, the optional commands `sf_init` and `sf_srch` can be used to give initial values of the parameters (the former) and to search for better initial values given the initial values (the latter). If initial values were not given by `sf_init` (or the native, low-level Stata command `ml init`), an arbitrary set of initial values will be selected by Stata in order to jump start the numerical optimization process. The model is estimated by numerically maximizing the log-likelihood function of the model; this is accomplished by `ml max`. After the model parameters are estimated, users can then use the command `sf_predict` to obtain observation-specific inefficiency index, confidence intervals, and marginal effects of inefficiency determinants (if applicable).

To illustrate, consider a production frontier model with a flexible specification.

$$\begin{aligned} \ln y_i &= \ln y_i^* - u_i, && (1.1)\\ \ln y_i^* &= \boldsymbol{x}_i\boldsymbol{\beta} + v_i, && (1.2)\\ u_i &\sim N^+(\mu, \sigma_u^2), && (1.3)\\ v_i &\sim N(0, \sigma_v^2). && (1.4) \end{aligned}$$

The following parameterizations are used.

$$\mu = \boldsymbol{z}_i'\boldsymbol{\delta}, \tag{1.5}$$

$$\sigma_u^2 = \exp(\boldsymbol{z}_i{}'\boldsymbol{w}), \tag{1.6}$$

$$\sigma_v^2 = \exp(c_0), \tag{1.7}$$

where c_0 is a constant, $\boldsymbol{z}_i$ is the vector of exogenous variables of observation i, and $\boldsymbol{\delta}$ and $\boldsymbol{w}$ are the corresponding coefficient vectors.

The model of Battese and Coelli (1995) parameterizes μ by a vector of variable and it keeps σ_u^2 constant ($\boldsymbol{z}$ contains only a vector of 1). The model of Wang (2002, 2003) calls for parameterizing μ and σ_u^2 by the same vector of exogenous variables. The double parameterization is not only less ad hoc, but it also accommodates non-monotonic relationships between the inefficiency and its determinants.

The scaling property model of Wang and Schmidt (2002) does not nest within the above framework. For illustration purposes, we list it here.

$$\ln y_i = \ln y_i^* - u_i, \tag{1.8}$$

$$\ln y_i^* = \boldsymbol{x}'\boldsymbol{\beta} + v_i, \tag{1.9}$$

$$\begin{aligned} u_i &\sim h(\boldsymbol{z}_i,\, \boldsymbol{\delta}) \cdot N^+(\tau,\, \sigma^2) \\ &\equiv \exp(\boldsymbol{z}_i'\boldsymbol{\delta}) \cdot N^+(\tau,\, \exp(c_u)) \end{aligned} \tag{1.10}$$

$$\begin{aligned} v_i &\sim N(0, \sigma_v^2) \\ &\equiv N(0, \exp(c_0)), \end{aligned} \tag{1.11}$$

where τ, c_u, and c_0 are unconstrained constant parameters, and $\boldsymbol{z}_i$ is a variable vector which does *not* contain a constant. In this setup, the distribution of u_i is based on the basic distribution $N^+(\mu,\, \sigma^2)$ and the scale is stretched by the non-stochastic and non-negative scaling function $\exp(\boldsymbol{z}_i'\boldsymbol{\delta})$.

1.1.1 sfmodel

```
sfmodel depvar [if] [in],  distribution(halfnormal | truncated | exponential)
                           production | cost
                           frontier(varlist_f [, noconstant])
                           mu([varlist_m [, noconstant]]) etas([varlist_e [, noconstant]])
                           usigmas([varlist_u [, noconstant]])
                           vsigmas([varlist_v [, noconstant]])
                           scaling hscale(varlist_h) tau cu robust cluster(variable)
                           show [technique(NR | DFP | BFGS | BHHH)]
```

Description

The command `sfmodel` selects the distribution assumption of u for the model, specifies the dependent and independent variables of the frontier function, and optionally specifies the exogenous determinants of inefficiency. In effect, this command set up the appropriate likelihood function, but it does *not* carry out the estimation. The estimation is put into action by `ml max`.

Options

distribution(*halfnormal*|*truncated*|*exponential*) indicates the distribution assumption of u_i. The possible choices are `halfnormal` for the model of half-normal distribution, `truncated` for the truncated-normal distribution model and the scaling-property model, and `exponential` for exponential distribution model.

production|cost indicates whether the model is a production-type model (production) or a cost-type model (cost).

frontier($varlist_f$[, noconstant]) specifies variables to be included in the deterministic part of the frontier function; i.e., the $\boldsymbol{x}$ variables in the discussions of the previous sections.

Note that for the variables specified in the various functions, including `frontier()`, `usigmas()`, `vsigmas()`, and many others to be discussed later, only *non-constant* variables should be specified. It is because, by default, a constant of 1 will be appended to the specified variable list by Stata, therefore if the $varlist_f$ (i.e., variable list) includes a constant, there will be a collinearity problem.

There are three different possibilities in specifying the $varlist_f$. (1) Leave it blank (ex., `frontier()`). Then the function will contain only a constant automatically appended by Stata. (2) Specify a list of existing variables (ex., `frontier(x1 x2)`). Then the function will contain those specified plus a constant. (3) Specify a list of existing variables and ask Stata not to automatically include the constant (ex., `frontier(x1 x2, noconstant)`). Then the function will contain only the specified variables, and no constant will be appended to the list by Stata.

mu([$varlist_m$[, noconstant]]) is used only when u_i is assumed to have a truncated-normal distribution. It specifies variables used to parameterize the pre-truncation mean of the distribution of u_i, i.e., the $\boldsymbol{z}_i$ in places such as (1.5). If specifies `mu()` without arguments, the pre-truncation mean is a constant. If variables are specified in the argument, such as `mu(z1, z2)`, then the pre-truncation mean is a function of these variables.

etas([$varlist_e$[, noconstant]]) is used only when u_i is assumed to have an exponential distribution. Note that it parameterizes the η^2 (which is the variance of u), not η (which is the standard deviation). This is emphasized by the last letter "*s*" (for *square*) of this syntax. If specifies `etas()` without arguments, then η^2 is assumed to be a constant.

hscale($varlist_h$) is used only for the scaling property model of Wang and Schmidt (2002). It specifies the variables in the scaling function; i.e., the $\boldsymbol{z}_i$ variables in (1.10). Since the scaling function of (1.10) needs to be specified as a function of variable, empty string (`hscale()`) is not allowed. Also, because the function does not have a constant by construction, the `noconstant` option is not allowed.

tau is used only for the scaling property model of Wang and Schmidt (2002). It indicates the τ parameter in (1.10).

cu is used only for the scaling property model of Wang and Schmidt (2002). It indicates the c_u parameter in (1.10).

usigmas([*varlist*$_u$ [, *noconstant*]]) specifies the variables used to parameterize the pre-truncation variance of the inefficiency u_i. Note that it parameterizes the σ^2 (variance), not σ (standard deviations). This is emphasized by the last letter "*s*" (for *square*) of this syntax. If `usigmas()` is specified without arguments, a constant variance of u_i is assumed.

vsigmas([*varlist*$_v$ [, *noconstant*]]) specifies the variables used to parameterize the variance of the random error v_i. Note that it parameterizes the σ_v^2, not σ_v. This is emphasized by the last letter "*s*" (for *square*) of this syntax. If `vsigmas()` is specified without arguments, a constant variance of v_i is assumed.

show prints the likelihood function set up by `sfmodel` in Stata's `ml model` syntax. It is mainly for debugging purposes. It might also be useful if, for example, users want to supply initial values using Stata's `ml init` in lieu of `sf_init` and need to know the order of equations and variables in the likelihood function.

A Note: Users are reminded that to estimate the model of Wang (2002), variable list specified in `mu()` and `usigmas()` should be identical. They should be identical in two aspects: all the specified variables should be the same, *and* that the order of their appearances in `mu()` and `usigmas()` should also be the same. The requirement that the same variables be included in both places *is* the basic idea of the model. The requirement of the same ordering, on the other hand, is a purely technical one. If same variables are specified but they appear in different orders, the model can still be estimated, but the marginal effects cannot be computed by the `sf_predict` command to be discussed later.

1.1.2 sf_init

`sf_init`, `frontier(`*numlist*$_f$`)` [`mu(`*numlist*$_m$`)` `eta(`*numlist*$_e$`)` `usigmas(`*numlist*$_u$`)`
`vsigmas(`*numlist*$_v$`)` `hscale(`*number*$_h$`)` `zvar(`*number*$_z$`)`
`tau(`*number*$_t$`)` `cu(`*numlist*$_c$`)` `show`]

Description

The `sf_init` command is used following `sfmodel` or `sf_fixeff`. It allows users to supply initial values for the model parameters, which are later used by the numerical maximization routines to maximize the likelihood function. This is an optional command. Stata will pick an arbitrary set of initial values to begin the maximization process if no initial value is given by the user. However, since stochastic frontier models are numerically difficult, particularly for the more elaborated models, *good* initial values have better chances of successful convergence and would certainly speed up convergence.

The sf_init is essentially a wrapper of the Stata command `ml init`. Unlike `ml init`, users using sf_init do not need to worry about orders of the equations (`frontier`, `mu`, etc.) used in the model. For example, users can put `vsigmas(`*numlist*$_v$`)` before `frontier(`*numlist*$_f$`)` or vice versa. Experienced users may specify `show` option in `sfmodel` to know the order of equations/variables, and use `ml init` to provide initial values directly. See [R] `ml` for information on `ml init`.

It is important to note that if the user supplies initial values using sf_init (or for this matter, `ml init` as well), he needs to give a *complete* set of initial values for the model. That means initial values for each variable (including the constant) in each of the equations. Users cannot specify only a subset of initial values.

Options

The options, `frontier(`*numlist*$_f$`)`, `mu(`*numlist*$_m$`)`, etc., correspond to those in `sfmodel` sf_fixeff are used in a similar way: It specifies a list of numbers or a $1 \times k$ matrix of numerical values. The `show` option, which is mainly for debugging purposes, prints the initial value vector set up by sf_init in Stata's `ml init` syntax.

There is a difference in specifying the number of initial values for `sfmodel` and sf_fixeff. The explanations are in the follows.

Consider a model estimated by `sfmodel`. Suppose in the `sfmodel` command line you specify `frontier(x1 x2)`. This implies that the deterministic part of the frontier equation contains two variables, $x1$ and $x2$, and a constant. The constant is automatically appended to the equation by Stata unless `noconstant` is also specified. The corresponding initial values in sf_init would be specified as `frontier(0.1 0.2 0.5)`. In this example, 0.1 and 0.2 are the initial values of $x1$ and $x2$, respectively, and the value 0.5 (the *last* value in the list) is the initial value of the constant of the deterministic frontier function. The above rule applies to almost all the equations except for the `hscale()` function which does not have a constant by construction.

Now consider a model estimated by sf_fixeff. Suppose in the sf_fixeff command line you specify `frontier(x1 x2)`. Unlike in the previous case, Stata will not automatically append a constant in this case. That is, `frontier()` in sf_fixeff does not (and cannot) have a constant by construction. Therefore, the corresponding initial values in sf_init would be specified as `frontier(0.1 0.2)`. The same is also true for `zvar()`; that is, no initial value for the *additional constant* since no constant is added to the equation.

1.1.3 sf_srch

sf_srch, [`n(`*number*`)` `frontier([`*varlist*$_f$`])` `mu([`*varlist*$_m$`])` `eta([`*varlist*$_e$`])`
`usigmas([`*varlist*$_u$`])` `vsigmas([`*varlist*$_v$`])` `hscale([`*varlist*$_h$`])` `zvar([`*varlist*$_z$`])`
`nograph` `fast`]

Description

The `sf_srch` searches for better initial values for the variables in the specified functions. The `sf_srch` will do the search for all the constants in the model by default. This is an optional command, and it can be used regardless of whether `sf_init` has been previously issued. If `sf_init` is used to provide initial values before issuing `sf_srch`, `sf_srch` will perform the search using those initial values as starting points. Otherwise, the search starts at initial values chosen by the internal algorithm of Stata.

Unlike `sf_init`, users do not need to specify a complete set of variables with `sf_srch`. That is, users can choose to perform the search on only a subset of variables from all or part of the equations.

The `sf_srch` is essentially a wrapper of Stata's `ml plot` command, which graphs the likelihood profile for a specified parameter and then replaces the value of the parameter according to the maximum value of the likelihood function. The command is useful for fine-tuning the initial value of the specified parameter, holding other parameter values unchanged.

Options

n(*number***)** specified the number of times the search is to be performed on the specified parameters. For example, `n(1)` will do the search once for each of the specified variables, and `n(2)` will allow the search to cycle through the variables once again. There is no upper limit on the number, but it has to be an integer and greater than 0.

frontier([$varlist_f$]**)** specifies the variables of the frontier function to which the search is to be performed. The specified variables should be the same or be a subset of the variables specified in the `frontier` function of the `sfmodel` command. If only a subset of the variables is specified, the search will be performed on those variables only, and the initial values of other variables not specified will not be altered.

By default of `sf_srch`, all the constants of the model, such as the constants in functions of `frontier`, `mu`, etc. and the `tau` and `cu` parameters, will be searched automatically for better initial values.

mu() eta() usigmas() vsigmas() hscale() zvar() they are used in the similar way as described above. Note that `zvar()` is used only for models estimated by `sf_fixeff`.

nograph asks Stata to perform the search silently without showing graphs of the likelihood function profile in a graph window. Although the graphs are sometimes informative, the graph rendering

fast asks Stata to draw graphs of variables' likelihood profiles using Stata 7 style, which is much faster than the default. By default, graphs are drawn using newer (Stata 8.0 or higher) graph styles, which generate good-looking graphs at the expense of longer time. For the purpose of initial value search, pretty graphs are not of much value. The time-look tradeoff is significant particularly when many initial values are to be searched by `sf_srch`.

Note that the `nograph` option does not cut the search time as `fast` would do. The `nograph` option only suppresses graph renderings in the screen.

One caveat of the `fast` option is that the code is based on Stata 8.2 when the book is written. New features added to `ml plot` in the future versions of Stata will not be available if `fast` is specified.

1.1.4 sf_predict

`sf_predict [if] [in],` `bc(`*newvarname*$_1$`)` `jlms(`*newvarname*$_2$`)` `[ci(#)` `marginal ]`

Description

The command `sf_predict` computes the observation-specific efficiency index of Battese and Coelli (1988) (`bc`) and/or Jondrow et al. (1982) (`jlms`), and the confidence intervals of these index (`ci`). When appropriate, it can also calculate the marginal effects of the exogenous determinants on the mean and the variance of inefficiency u_i (`marginal`). If marginal effects are requested, the sample mean of the marginal effects will be printed on Stata's result window, and the observation-specific marginal effects will be saved in the dataset.

The syntax of `sf_predict` is the same for all the models, regardless of the distributional assumptions on u_i or whether exogenous determinant variables are included in the model.

Options

bc(*newvarname*$_1$**)** calculates the technical efficiency index $E(\exp(-u_i)|\epsilon_i)$ of Battese and Coelli (1988), and save the observation-specific values in the variable *newnvarname*$_1$.

jlms(*newvarname*$_2$**)** calculates the inefficiency index $E(u_i|\epsilon_i)$ of Jondrow et al. (1982), and save the observation-specific values in the variable *newvarname*$_2$.

ci(#) calculates, for each observation, the lower and upper bounds of the confidence intervals of the efficiency score specified by either `bc(`*newvarname*$_1$`)`, or `jlms(`*newvarname*$_2$`)`, or both. The number (#) indicates the coverage of the confidence interval. For example, ci(95) indicates that the lower and upper bounds of a 95% confidence intervals are to be calculated. Values of the bounds are saved in new variables, with the names being *newvarname*$_1$`_#L` and *newvarname*$_1$_#U, or *newvarname*$_2$_#L and *newvarname*$_2$_#U, respectively, for the case of `bc` and `jlms`.

For example, if `bc(r1)` and `ci(95)` are specified, then three variables will be created. One is `r1` created per the option `bc(r1)`, which takes values of the point estimates of $E(\exp(-u_i)|\epsilon_i)$ of each observation. The other two are `r1_95L` and `r1_95U`. The former contains values of the lower bound of the 95% confidence interval of `r1` and the latter contains values of the upper bound of the 95% confidence interval of `r1` for each observation. If `jlms(k2)` is also specified, then, in addition to the variable `k2`, two other variables are created that contains values of the lower and upper bounds of the associated confidence interval: `k2_95L` and `k2_95U`.

marginal calculates the marginal effects of the exogenous determinants on inefficiency. The marginal effects are observation-specific, and those values are saved in the variables *variable*_M and *variable*_V for the marginal effects on the mean and the variance, respectively, of the inefficiency, where *variable* is the variable name of the exogenous determinant. In addition, the sample mean of the variable's marginal effects will be printed on the Stata's result window.

For example, if `mu(z1)` is specified in `sfmodel` for a truncated normal model and `marginal` is specified in `sf_predict`, then variables z1_M and z1_V are created, taking values of the marginal effects of z1 on the mean and the variance, respectively, of expected inefficiency ($E(u_i)$). The sample means of z1_M and z1_V are also printed on the result window of Stata.

1.1.5 sf_mixtable

`sf_mixtable, dof(`*num*`)`

Description

The command `sf_mixtable` tabulates critical values of a mixed Chi-square distribution at different significance levels with the degree of freedom equal to `-dof-`. The values are used for hypothesis testings of the LR test. The values are taken from Table 1 of Kodde and Palm (1986).

Options

dof(*num***)** specifies the degrees of freedom of the test statistic, which is usually the number of restrictions in the test. The degrees of freedom is restricted to values between 1 and 40 (inclusive).

1.2 Panel Data Models

The panel data model program estimates a fixed-effect panel stochastic frontier model by within-transformation as proposed by Wang and Ho (2010). The main program is `sf_fixeff` which sets up the likelihood function. The `sf_init` and `sf_srch` commands described on p.7 and p.8 can be optionally used following `sf_fixeff` to provide and refine initial values. The model is then estimated by the standard `ml max` command. After the model is estimated, the (in)efficiency index is obtained by `sf_effindex`.

We introduce `sf_fixeff` and `sf_effindex` commands in the rest of this section. Readers should refer to previous sections for `sf_init` and `sf_srch`.

Consider a fixed-effects panel stochastic frontier model with the following specifications:

$$y_{it} = \alpha_i + \boldsymbol{x}_{it}\boldsymbol{\beta} + \varepsilon_{it}, \tag{1.12}$$

$$\varepsilon_{it} = v_{it} - u_{it}, \tag{1.13}$$

$$v_{it} \sim N(0, \sigma_v^2), \tag{1.14}$$

$$u_{it} = h_{it} \cdot u_i^*, \tag{1.15}$$

$$h_{it} = f(\boldsymbol{z}_{it}\boldsymbol{\delta}), \tag{1.16}$$

$$u_i^* \sim N^+(\mu, \sigma_u^2), \tag{1.17}$$

$$\sigma_v^2 = \exp(C_v), \tag{1.18}$$

$$\sigma_u^2 = \exp(C_u), \qquad i = 1, \ldots, N, \quad t = 1, \ldots, T. \tag{1.19}$$

In this setup, α_i is individual i's fixed unobservable effect, and other variables are defined as usual.

1.2.1 sf_fixeff

```
sf_fixeff depvar [if] [in], distribution(halfnormal | truncated)
                            production | cost
                            frontier(varlist_f) zvar(varlist_z) id(varname)
                            time(varname)[ mu usigmas vsigmas show ]
```

Description

The command `sf_fixeff` estimates the panel data model described in (1.12) to (1.19) using the within-transformation method. It handls balanced and unbalanced panels automatically. The command specifies the dependent and independent variables of the frontier function, the exogenous determinants of inefficiency, and selects the distribution assumption of u^* of the model. In effect, this command set up the appropriate likelihood function, but it does *not* carry out the estimation. The estimation is put into action by `ml max`.

Wang and Ho (2010) show that the within-transformed and the first-differenced models are algebraically the same. This command uses only the within-transformation method. Users do not need to transformed the variables prior to using the command; the program will do it for you. The command will create a list of within-transformed variables after the estimation.

Options

distribution(*halfnormal*|*truncated*) indicates the distribution assumption of u_i^*. The possible choices are `halfnormal` for the model of half-normal distribution and `truncated` for the truncated-normal distribution mode. If `halfnormal` is chosen, $\mu = 0$ in (1.17).

production | **cost** indicates whether the model is a production-type model (production) or a cost-type model (cost).

frontier(*varlist*$_f$) specifies variables to be included in the frontier function, i.e., $\boldsymbol{x}_{it}$ in (1.12). It cannot be empty.

Note that individual-specific and time-invariant variables, such as gender and regional dummies, cannot be specified. Unlike most of other equation specification, a constant *will not be* automatically added to the equation.

Note also that variables specified here should *not* be transformed by the within. The transformation will be done by the program.

zvar(*varlist*$_z$) specifies variables to be included in the scaling function, i.e., $\boldsymbol{z}_{it}$ in (1.16). It cannot be empty.

Similar to the `frontier()`, variables included here should not be transformed by the within or the first-difference method. A constant will NOT be automatically added to this equation.

id(*varname*) specifies the variable that identifies each panel.

time(*varname*) specifies the time variable for the panels.

mu is used only when u_i^* is assumed to have a truncated-normal distribution. It indicates the μ parameter in (1.17).

vsigmas indicates the C_v parameter in (1.18). It is a constant.

usigmas indicates the C_u parameter in (1.19). It is a constant.

After the estimation, `sf_transform` can be used to obtain $\hat{\sigma}_v^2$ and $\hat{\sigma}_u^2$.

show prints the likelihood function set up by `sf_fixeff` in Stata's `ml model` syntax. It is mainly for debugging purposes. It might also be useful if, for example, users want to supply initial values using Stata's `ml init` in lieu of `sf_init` and need to know the order of equations and variables in the likelihood function.

1.2.2 sf_effindex

`sf_effindex , bc(`*newvarname*$_1$`) jlms(`*newvarname*$_2$`)`

Description

The command `sf_effindex` computes both of the JLMS inefficiency index and the BC efficiency index for the panel data model estimated by `sf_fixeff`.

Options

bc(*newvarname*$_1$**)** calculates the technical efficiency index $E(\exp(-u_i)|\tilde{\epsilon}_{i.})$ of Battese and Coelli (1988), and save the observation-specific values in the variable *newnvarname*$_1$. See Wang and Ho (2010) for details.

jlms(*newvarname*$_2$**)** calculates the inefficiency index $E(u_i|\tilde{\epsilon}_{i.})$ of Jondrow et al. (1982), and save the observation-specific values in the variable *newvarname*$_2$. See Wang and Ho (2010) for details.

1.3 Specification Tests I: Moment-Based Test

Consider the following model.

$$\ln y_i = \ln y_i^* - u_i, \tag{1.20}$$

$$\ln y_i^* = \boldsymbol{x}_i\boldsymbol{\beta} + v_i, \tag{1.21}$$

$$v_i \sim N(0, \sigma_v^2), \tag{1.22}$$

$$u_i \sim N^+(0, \sigma_u^2), \tag{1.23}$$

or

$$u_i \sim \text{exponential}(\theta). \tag{1.24}$$

The θ is the parameter for the exponential distribution with the mean of the distribution being $1/\theta$.

The skewness test of Schmidt and Lin (1984) and the likelihood ratio tests have been used by researchers to test the existence of u_i and the associated distribution assumption of it. These tests, however, are conditional on the model's other specifications being correct. The specification includes the functional form of the frontier function, the variables in the frontier and/or the inefficiency functions, and the distribution assumption of v_i.

The tests we introduced here and in the next section are general specification tests. They can be particularly useful in testing the distribution assumption of the composed error of the model, i.e., $v_i - u_i$ for a production frontier model and $v_i + u_i$ for a cost frontier model.

1.3.1 sfmtest

```
sfmtest depvar [if] [in],  udistribution(halfnormal | exponential)
                           production | cost
                           frontier(varlist_f [, noconstant])
                           [ omega(num)
                             bc(newvarname_1) jlms(newvarname_2) ]
```

Description

This command carries out the moment-based estimation and test for stochastic frontier models proposed by Chen and Wang (2012). It does three things: (1) It estimates the main model parameters using the method of moment. (2) It optionally estimates the JLMS and BC (in)efficiency index. (3) It performs the specification test of the model.

The specification test may be particularly useful in testing the distribution assumption of the composed error of the model ($v_i - u_i$ for production frontier model and $v_i + u_i$ for cost frontier model). The test can be performed against one of the following two null hypothesis: (A) v_i is normal and u_i is half-normal (`udist(h)`), and (B) v_i is normal and u_i is exponential (`udist(e)`).

The test is based on the moment generating function of the assumed distribution and it can take the form of a sine or a cosine test. The forms of the tests, are, respectively:

$$\phi_2(\epsilon_{c,i}, \theta) = \begin{bmatrix} \sin(\omega_1 \epsilon_{c,i}) - \mathrm{E}[\sin(\omega_1 \epsilon_{c,i})] \\ \vdots \\ \sin(\omega_q \epsilon_{c,i}) - \mathrm{E}[\sin(\omega_q \epsilon_{c,i})] \end{bmatrix} \tag{1.25}$$

and

$$\phi_2(\epsilon_{c,i}, \theta) = \begin{bmatrix} \cos(\omega_1 \epsilon_{c,i}) - \mathrm{E}[\cos(\omega_1 \epsilon_{c,i})] \\ \vdots \\ \cos(\omega_q \epsilon_{c,i}) - \mathrm{E}[\cos(\omega_q \epsilon_{c,i})] \end{bmatrix}, \tag{1.26}$$

where $\epsilon_{c,i} = \epsilon_i - \mathrm{E}[\epsilon_i]$ is the centered composed error, and ω_k is a pre-determined real number with $k = 1, \ldots, q$. In `sfmtest`, we choose $q = 1$. It is very difficult to determine the optimal choice of the ω from a theoretical viewpoint. Nonetheless, as shown by Chen and Wang (2012), $\omega = 1$ or around 1 usually work well. Their simulation results also show that the cosine test works better than the sine test in some of the scenarios.

Options

udistribution(*halfnormal*|*exponential*) indicates the distribution assumption of u_i. The possible choices are `halfnormal` for the model of half-normal distribution and `exponential` for the truncated-normal distribution mode.

production|**cost** indicates whether the model is a production-type model (production) or a cost-type model (cost).

frontier(*varlist*$_f$**)** specifies variables to be included in the frontier function, i.e., $\boldsymbol{x}_i$ in (1.20).

omega(*num***)** is ω in (1.25) and (1.26) and it contains a single constant parameter. Chen and Wang (2012) show that the value of 1 (the default) or around 1 usually works well in their simulation experiments.

bc(*newvarname*$_1$**)** calculates the technical efficiency index $E(\exp(-u_i)|\epsilon_i)$ of Battese and Coelli (1988), and save the observation-specific values in the variable *newnvarname*$_1$.

jlms(*newvarname*$_2$**)** calculates the inefficiency index $E(u_i|\epsilon_i)$ of Jondrow et al. (1982), and save the observation-specific values in the variable *newvarname*$_2$.

1.4 Specification Test II: SICM Test (preliminary)

Sorry, the paper has not been published yet.

Bibliography

[1] 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**, pp. 387-399.

[2] Battese, G.E., and Coelli, T.J. (1995). "A Model for Technical Inefficiency Effects in a Stochastic Frontier Production Function for Panel Data," *Empirical Economics* **20**, pp. 325-32.

[3] Bierens, H.J., and Wang, L. (2012). "Integrated Conditional Moment Tests for Parametric Conditional Distributions," *Econometric Theory* **28**, pp. 328-362.

[4] Chen, Y.-T., and Wang, H.-J. (2012). "Centered-Residuals-Based Moment Tests for Stochastic Frontier Models," *Econometric Reviews* **31**, pp. 625-53.

[5] Jondrow, J., Knox Lovell, C.A., Materov, I.S., and Schmidt, P. (1982). "On the Estimation of Technical Inefficiency in the Stochastic Frontier Production Function Model," *Journal of Econometrics* **19**, pp. 233-238.

[6] Kodde, D.A., and Palm, F.C. (1986). "Wald Criteria for Jointly Testing Equality and Inequality Restrictions," *Econometrica* **54**, pp. 1243-48.

[7] Schmidt, P., and Lin, T.-F. (1984). "Simple Tests of Alternative Specifications in Stochastic Frontier Models," *Journal of Econometrics* **24**, pp. 349-361.

[8] Wang, H.J. (2002). "Heteroscedasticity and Non-Monotonic Efficiency Effects of a Stochastic Frontier Model," **18**, pp. 241-253.

[9] Wang, H.J., and Schmidt, P. (2002). "One-Step and Two-Step Estimation of the Effects of Exogenous Variables on Technical Efficiency Levels," **18**, pp. 129-144.

[10] Wang, H.-J. (2003). "A Stochastic Frontier Analysis of Financing Constraints on Investment: The Case of Financial Liberalization in Taiwan," *Journal of Business & Economic Statistics* **21**, pp. 406-19.

[11] Wang, H.-J., and Ho, C.-W. (2010). "Estimating Fixed-Effect Panel Stochastic Frontier Models by Model Transformation," *Journal of Econometrics* **2**, pp. 289-96.

[12] Chen, Y.-T., Su, H.-W., and Wang, H.J. (2012). "The SICM Specification Tests of Stochastic Frontier Models," manuscript, National Taiwan University.