Stochastic Frontier Analysis using SFAMB for Ox

Jonathan Holtkamp, Bernhard Brümmer*

Department of Agricultural Economics, Georg-August-Universität Göttingen

January 2016

Contents

*Department of Agricultural Economics, D-37073 Göttingen, Germany, `http://www.uni-goettingen.de/de/19255.html`, E-mail: jonathan-holtkamp@web.de

1 Introduction

SFAMB (Stochastic Frontier Analysis using ModelBase) is a class written in Ox (Doornik, 2009) and is used by writing small programs that use an object of this class. Ox Console is free for academic use and is available from www.doornik.com. Structure and format of this manual follow the documentation "Panel Data estimation using DPD for Ox" by Doornik et al. (2012).

SFAMB is a package for estimating stochastic frontier production (as well as cost, distance, and profit) functions. The basic version of this package was written by Bernhard Brümmer. The current version includes four additional models for panel data. Code of the WT model is partially adapted from Stata code by Hung-Jen Wang. We thank Hung-Jen Wang for providing data to check our CFE code.

1.1 Disclaimer

This package is functional, but no warranty is given whatsoever. An email-based forum to discuss problems and issues related to Ox or corresponding packages is the ox-users discussion list. The archive is available via www.jiscmail.ac.uk/lists/ox-users.html. Please report bugs or suggestions for improvements to Jonathan Holtkamp (jonathan-holtkamp@web.de) or Bernhard Brümmer (bbruemm@gwdg.de).

1.2 Installation

1. Install Ox on your computer. To run a sample file, go to the Ox folder (C:/Program files/OxMetrics7/Ox). Then go the folder samples and run a file such as myfirst.ox, using double click. You can also use the command prompt. Consult the general Ox documentation for usage of the Ox language and related questions, see Doornik and Ooms (2006).

2. Extract sfamb.zip into the directory Ox/packages. The respective path is now Ox/packages/sfamb.

3. Check the readme.txt file if available.

4. You should now be able to use the package from any directory. Check the command #import <packages/sfamb/sfamb> at the top of the in-

put file you want to execute (e.g. `hbest1.ox`). The command `#include <packages/sfamb/sfamb.ox>` would also work.

5. The package includes a function for graphics that can be used with the Ox package `GnuDraw`. The respective directory is again the `Ox/packages` folder.

1.3 Main files

- `sfamb.h` – the header file for the `SFAMB` class;
- `sfamb.ox` – the source code with the `SFAMB` class;
- `sfamb.oxo` – the compiled source code;
- `sfamb.pdf` – this document.

Included sample data sets:

- Sample2.xls
- USDAafrica.xls

The remaining files are sample programs.

2 Data organisation and model formulation

2.1 Data organisation

Different data file formats can be read directly into a `SFAMB` object (.xls, .dta,...), for details see the Ox manual (Doornik and Ooms, 2006).

The data has to be organized in columns where the first row holds the variable name. Each row refers to the same time period. Missing values are also called NaN (Not a Number) in Ox. In case of panel data `SFAMB` needs to recognize the structure of the data. In the data file specify the variable names of the individuals and time periods. The data has to be stacked by individual ($i = 1, 2...N$) and within individuals by time period ($t = 1, 2...T_i$). The panel may be unbalanced.

Example:

id	time	y	x1
31	1	298384	24145
31	2	333522	27725
31	3	378768	38115
37	1	62473	3401
37	2	212442	12529
37	3	295142	16734
101	1	150037	10752
101	2	158909	10418
101	3	172744	10671

2.2 Model formulation

The sequence of model formulation is sketched in figure 1.

In each input file a new object is created. This object is an instance of the `SFAMB` class and can use the functionality of the class. The function `Load` loads the data file and creates the data base. You choose the model type with `SetMethod`. Five estimators /arguments are available (see section 3.1). In case of panel data, specify the panel structure using `Ident`. If the original data is in levels you can use `PrepData` (or other functions of the Modelbase or Database classes) for transformation. There are several types of variables that can be

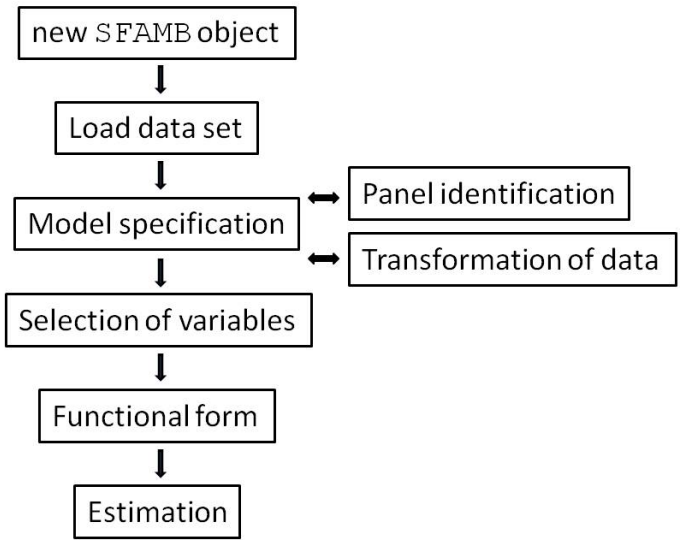

Figure 1: Model formulation.

selected according to the underlying model[1]. The respective function is called `Select` and works with the variable names.

To formulate the frontier function:

- Use `Select(Y_VAR, {".",.,.})` to select the dependent variable.
- Use `Select(X_VAR, {".",.,.})` to select the independent variable(s).

To include variables that affect the distribution of the inefficiency component:

- Use `Select(U_VAR, {".",.,.})` to select variables that shift the mean of the distribution.
- Use `Select(Z_VAR, {".",.,.})` to select the variables that are hypothesised as responsible for heteroscedasticity, i.e. those that affect the scale parameter of the distribution.

`SetTranslog` can be used to choose the functional form of the frontier function. In case of the translog specification, we recommend to normalize the variables by the respective sample means. Estimation of the model is executed via `Estimate`. For more details, see the documentation of member functions in section 4.

[1]In case of the panel models, a common constant is not identified. However, you can leave `"Constant"` in the selection because it is ignored automatically.

3 Stochastic frontier production function estimation

3.1 Econometric methods

SFAMB provides frontier models of Aigner et al. (1977); and Meeusen and van den Broeck (1977), respectively, with extensions; Schmidt and Sickles (1984); Greene (2005); Wang and Ho (2010) as well as Chen et al. (2014). The available estimators are:

	SetMethod	Ineff.distribution	Example
SFA - cross section	POOLED	μ and/or σ_u	hbest1.ox
Least squares with dummies	LSDV		hbest2.ox
SFA - with dummies	TFE		hbest2.ox
SFA - within-transformation	WT	μ or σ_u^2	hbest3.ox
SFA - consistent fixed effects	CFE		hbest2.ox

The Stochastic Frontier model

This section is intended as a short, concise introduction to Stochastic Frontier Analysis (SFA) techniques. A more detailed introduction can be found in Coelli et al. (2005). More advanced material is covered in Kumbhakar and Lovell (2000). The basic problem in efficiency analysis lies in the estimation of an unobservable frontier (production, distance or cost) function from observable input and output data, together with price data when necessary. Standard estimation techniques like OLS are inappropriate in this setting since they aim at the identification of average relationships, which are not in the focus of an efficiency model.

The basic approach was simultaneously developed by Aigner et al. (1977), and Meeusen and van den Broeck (1977). An exposition in terms of the production function highlights its most important characteristics. The basic production function model is given by:

$$y_i = \alpha + \beta' x_i + v_i - u_i \tag{1}$$

On the left hand side, y_i is the output (or some transformation of the output)

of observation i (i=1...N). On the right hand side, x_i is a matrix of inputs that produce output y_i, and the vector β describes technology parameters to be estimated. The most commonly used transformation of the variables is the natural logarithm. The crucial part of this formulation is the composed error term given by $\epsilon_i = v_i - u_i$, where v_i represents statistical noise and u_i represents inefficiency. Estimation is possible by means of Maximum Likelihood Estimation (MLE) where distributional assumptions concerning the error components are required. The noise component is a conventional two-sided error, distributed as $v_i \sim N(0, \sigma_v^2)$. The inefficiency component is a non-negative disturbance that can be modelled using several distributions. However, the *truncated normal* and *half normal* distributions are most frequently used and are implemented in `SFAMB`. Accordingly, the random variable u_i is distributed as $u_i \sim N^+(\mu, \sigma_u^2)$. If μ equals zero the model is labelled as the *normal-half normal* SF model; *normal-truncated normal* SF model otherwise.

The independence assumption for the inefficiency distribution can be changed by introducing covariates into the distribution, thereby accounting for differences in inefficiency between individuals. The corresponding covariates are often labelled as Z-variables. These can be used to model either the mean (location) or variance (scale) of the underlying distribution or both, cf. Alvarez et al. (2006). An useful overview is given by Lai and Huang (2010) who summarize and categorize several well-known models. A model describing μ by means of covariates is labelled as the KGMHLBC model[2], $u_i \sim N^+(\mu_0 + \delta' z_i, \sigma_u^2)$. If μ is set to zero and the scale is modelled using an exponential form it is the RSCFG model[3], $u_i \sim N^+(0, exp(2\,(\delta' z_i))$. Combination leads to $u_i \sim N^+(\mu_i = \mu_0 + \delta' z_i, \sigma_{u,i}^2 = exp(2\,(\delta' z_i))$, that can be labelled as a generalized linear mean model (Lai and Huang, 2010). *Note*: In `SFAMB`, the respective parameter modelled in the `POOLED` model is (the natural logarithm of) $\sigma_{u,i}$, not $\sigma_{u,i}^2$![4]

In addition to standard results, the estimation output of the `POOLED` model provides three other figures:

`gamma` is given by $\gamma = \sigma_u^2/\sigma^2 = \sigma_u^2/(\sigma_v^2 + \sigma_u^2)$
where (in case of the half-normal specification) $\sigma_u^2 = \frac{1}{n}\sum_i exp(2\,\delta' z_i)$.

`VAR(u)/VAR(total)` describes the "correct" variance decomposition of the com-

[2]Kumbhakar et al. (1991); Huang and Liu (1994); Battese and Coelli (1995);
[3]Reifschneider and Stevenson (1991); Caudill et al. (1995);
[4]While σ_u^2 is often used, the original formulation of CFG involved σ_u.

posed error (recall that given u is a one-sided disturbance, σ_u^2 is not the variance $var[u]$ of the one-sided error). The share of the variance of u in the total variance of the composed error is given by $var[u]/var[\epsilon] = [(\pi-2)/\pi]\sigma_u^2/[(\pi-2)/\pi]\sigma_u^2 + \sigma_v^2$, cf. Greene (2008, p.118).

`Test of one-sided err` provides a likelihood ratio test statistic for the presence of inefficiency, i.e. for the null hypothesis H_0: γ=0. The critical value cannot be taken from a conventional χ^2-table, see Kodde and Palm (1986).

A point estimator of inefficiency is given by $E(u_i|\epsilon_i)$, see Jondrow et al. (1982). If the dependent variable is in logarithms, a more appropriate estimator is the point estimator of technical efficiency $TE_i = E(exp(-u_i)|\epsilon_i)$, see Battese and Coelli (1988).

Unobserved heterogeneity - `LSDV` model

With panel data, additional information on each individual is available. Each cross section i is observed over a certain period of time T_i (t=1...$\mathrm{T_i}$):

$$y_{it} = \alpha_i + \beta' x_{it} + v_{it} \tag{2}$$

This formulation differs from equation (1) in that it involves time dimension t, only one error component (two-sided) and an individual intercept α_i. The model is estimated by OLS, and hence, $v_{it} \sim N(0, \sigma_v^2)$. Its virtue lies in the identification of the N time-invariant individual ("fixed") effects. These effects may capture unmeasured attributes, and hence, this approach is one way to deal with (unobserved) heterogeneity. The model has different names in the literature; one that is commonly used is "Least squares with dummy variables" (LSDV). Instead of estimating all N dummies, the usual approach is to employ a transformation: for each panel i, the respective variables (e.g. x_{it}) are transformed by subtracting the individual mean (out of T_i) from the observation in period t, i.e. $\tilde{x}_{it} = x_{it} - \bar{x}_i$. This procedure (within-transformation) removes the individual effects (because $\tilde{\alpha}_i = \alpha_i - \alpha_i = 0$) and estimation works only with deviations from means, i.e. with the transformed variables. Estimates of the individual effects are calculated as:

$$\hat{\alpha}_i = \bar{y}_i - \hat{\beta}' \bar{x}_i \tag{3}$$

Schmidt and Sickles (1984) use the model in a frontier context. They interpret the individual with the highest intercept as 100% technically efficient. The inefficiency of the remaining groups is assessed by $u_i = max(\hat{\alpha}) - \hat{\alpha}_i$; efficiency estimates are time-invariant and are given by $TE_i = E(exp(-u_i))$. Estimation output of the `LSDV` model differs from the other models to some extent:

`sigma_e` describes σ_v that is the square root of the corrected estimate of the error variance $\sigma_v^2 = \frac{SSR}{N(T-1)-K}$. This estimate is also used to compute the standard errors.

`AIC1 (all obs)` is given by $AIC1 = -2\,lnL + 2\,(K+1)$; it uses the likelihood function $lnL = -\frac{NT}{2}ln(2\pi) - \frac{NT}{2}ln(\sigma^2) - \frac{\sum_i \sum_t \tilde{v}_{it}^2}{2\sigma^2}$ with the uncorrected $\sigma^2 = \frac{SSR}{NT}$.

`AIC2` uses a different formula for the criterion, $AIC2 = ln(\frac{SSR}{NT}) + (2\,\frac{K+N}{NT})$; that does not need the likelihood function and considers the number of individuals in the penalty term.

Unobserved heterogeneity in SFA

Dummy variables - `TFE` model

The approach outlined above does not distinguish between inefficiency and unobserved heterogeneity because there is no one-sided error component. The respective ("true") specification of the SF model for panel data is given by:

$$y_{it} = \alpha_i + \beta' x_{it} + v_{it} - u_{it} \tag{4}$$

This model was proposed by Greene (2005) and is known as the "true fixed effects" (TFE) frontier model. Estimation involves all N individual effects, and hence, the model suffers from the incidental parameters problem. In micro panels (T fixed), σ^2 is inconsistent as the sample size increases.

The point estimators for inefficiency and technical efficiency are the same as for the `POOLED` model. Output of the `TFE` model provides `lambda`, given by $\lambda = \sigma_u/\sigma_v$.

Elimination of dummies - `WT` model

To overcome the incidental parameters problem Wang and Ho (2010) propose an extension that is based on deviations from means[5]:

$$\tilde{y}_{it} = \beta' \tilde{x}_{it} + \tilde{v}_{it} - \tilde{u}_{it} \tag{5}$$

This within-transformation (`WT`) model is estimated by MLE. The transformed noise component is distributed as multivariate normal, i.e. $\tilde{v}_{it} \sim MN(0, \Pi)$. However, simple transformation of the one-sided error component would result in an unknown distribution. Therefore, time-varying inefficiency is specified as $u_{it} = u_i^* \times h_{it}$. The persistent part u_i^* (inefficiency) is assumed to follow a half-normal or truncated-normal distribution, i.e. $u_i^* \sim N^+(\mu, \sigma_u^2)$, where μ is equal to zero in case of a half-normal distribution. The scaling function $h_{it} = f(\delta' z_{it})$ includes firm- and time-specific variables (z_{it}) that might affect the inefficiency distribution. The vector δ describes the corresponding parameters and the function takes an exponential form, i.e. $f(\delta' z_{it}) = exp(\delta' z_{it})$. The use of within-transformation does not affect the component u_i^* but the function value of h_{it} is transformed: $\tilde{u}_{it} = u_i^* \times \tilde{h}_{it}$. Wang and Ho (2010) present the conditional expectation of u_{it} in their equation (30); efficiency estimates are given by $TE_{it} = E(exp(-u_{it}|\tilde{\epsilon}_{it})$. Estimation output of the `WT` model additionally provides:

`lambda` is given by $\lambda = \sigma_u / \sigma_v$.

Although the individual effects are not directly estimated with the `WT` model, they can be recovered, assuming that $\bar{v}_i = 0$:

$$\hat{\alpha}_i = \bar{y}_i - \hat{\beta}' \bar{x}_i + \bar{u}_i \tag{6}$$

Consistent estimation with time-varying inefficiency - `CFE` model

Consistent estimation of the fixed effects SF model given in equation (4) is demonstrated by Chen et al. (2014). Their solution is also based on deviations from means so that the transformed model looks like equation (5). However, the respective likelihood function is derived only from the first *T-1* deviations, i.e. from $\tilde{\epsilon}_i^* = (\tilde{\epsilon}_{i1}, ..., \tilde{\epsilon}_{i,T-1})'$. This procedure has two advantages. First, within-transformation removes the incidental parameters. Second, an implicit correction

[5] In addition they demonstrate how the model can be estimated by first-differencing.

of the error variance is achieved by means of the first *T-1* deviations.[6] Wang and Ho (2010) use a multivariate normal distribution to model v_{it} but have to accept a persistent basic inefficiency component u_i^*. The current model is based on a more general distributional theory and allows for firm-specific and time-varying inefficiency u_{it}.

The composed error, $\epsilon = v - u$, has a skewed distribution (to the left) due to the non-negativeness of u. Accordingly, the standard (half-normal) SF model has a skew normal distribution, with skewness parameter λ and density:

$$f(\epsilon) = \frac{2}{\sigma}\,\phi\left(\frac{\epsilon}{\sigma}\right)\,\Phi\left(-\lambda\frac{\epsilon}{\sigma}\right) \tag{7}$$

While the skew normal distribution is a generalization of the normal distribution, it can be generalized itself by using the closed skew normal (CSN) distribution.[7] The composed error has a CSN distribution; what is written as:

$$\epsilon_{it} \sim CSN_{1,1}(0, \sigma^2, -\frac{\lambda}{\sigma}, 0, 1) \tag{8}$$

The density of a $CSN_{p,q}$-distribution includes a p-dimensional pdf and a q-dimensional cdf of a normal distribution. The five associated parameters describe location, scale and skewness, as well as the mean vector and covariance matrix in the cdf. With panel data, the T-dimensional vector $\epsilon_i = (\epsilon_{i1}, ..., \epsilon_{iT})'$ is distributed as:

$$\epsilon_i \sim CSN_{T,T}(0_T, \sigma^2 I_T, -\frac{\lambda}{\sigma} I_T, 0_T, I_T) \tag{9}$$

where I is the identity matrix. Chen et al. (2014) partition the vector ϵ_i into linear combinations: its mean $\bar{\epsilon}_i$ and its first *T-1* deviations $\tilde{\epsilon}_i^*$. The CSN distribution is "closed under linear combinations"(p.10). The density and respective log likelihood function for the model are derived from $\tilde{\epsilon}_i^*$. Accordingly, the likelihood function is free of incidental parameters and the parameters to be estimated are β, λ and σ^2 –as in the basic SF model. $\bar{\epsilon}_i$ and $\tilde{\epsilon}_i^*$ are not independent, unless $\lambda = 0$. If $\lambda = 0$ the model is the fixed effects model with normal error.

In order to obtain the inefficiency index, the composed error has to be re-

[6] With regards to the degrees of freedom, the correction accounts for the N individuals: $df = NT - N - K = N(T-1) - K$.

[7] Chen et al. (2014) explain how the SF model is related to the CSN distribution and present the required properties of CSN distributed random variables. Another plain introduction to the CSN distribution in the SF context is provided by Brorsen and Kim (2013).

covered:

$$\epsilon_{it} = y_{it} - \hat{y}_{it} = y_{it} - \hat{\beta}' x_{it} - \hat{\alpha}_i \tag{10}$$

There are two ways to calculate $\hat{\alpha}_i$. The one used here is labelled as the mean-adjusted estimate by Chen et al. (2014):

$$\hat{\alpha}_i^M = \bar{y}_i - \hat{\beta}' \bar{x}_i + \sqrt{\frac{2}{\pi}}\,\hat{\sigma}_u \tag{11}$$

The point estimators for inefficiency and technical efficiency are the same as for the `POOLED` model. The output provides `lambda`, given by $\lambda = \sigma_u/\sigma_v$.

3.2 Examples

Example: hbest1.ox

The first example is a generalized linear mean model (cf. Lai and Huang (2010)) where $u_i \sim N^+(\mu\,, \sigma_{u,i} = exp(\delta' z_i))$. The original data are in levels and are transformed using member function `PrepData` to accommodate the translog functional form. The data is a subset of USDA data prepared by Fuglie (2012) including the regions Sub-Saharan Africa and South Africa.

`hbest1.ox`

```
#include <oxstd.h>
#include <packages/gnudraw/gnudraw.h>
#import  <packages/sfamb/sfamb>

main(){
/*new object of class 'Sfa', Load data*/
decl fob = new Sfa();fob.Load("USDAafrica.xls");
//fob.Info(); exit(1);

/*Model specification*/
fob.SetMethod(POOLED);fob.SetConstant();
/*Data preparation*/
decl inorm = 1;//choose logs; or normalization and logs
fob.Renew(fob.PrepData(fob.GetVar("output"), inorm), "lny");
fob.Renew(fob.PrepData(fob.GetVar("labour"), inorm), "lnlab");
fob.Renew(fob.PrepData(fob.GetVar("land"), inorm), "lnland");
fob.Renew(fob.PrepData(fob.GetVar("machinery"), inorm), "lnmac");
fob.Renew(fob.PrepData(fob.GetVar("fertilizer"), inorm), "lnfert");
```

```
fob.Renew(fob.GetVar("time")-meanc(fob.GetVar("time")),"trend");
//fob.Info(); exit(1);

/*Set up model*/
fob.Select(Y_VAR, {"lny",0,0}); //Select dependent variable

fob.Select(X_VAR, { //Select regressors
"Constant",0,0,
"lnlab",0,0,
"lnland",0,0,
"lnmac",0,0,
"lnfert",0,0,
"trend",0,0
});

fob.Select(U_VAR, { //Shifting mu
"Constant",0,0
});

fob.Select(Z_VAR, { //Scaling sigma_u
"Constant",0,0,
"lnlab",0,0,
"lnland",0,0,
"lnmac",0,0,
"lnfert",0,0
});

// Select estimation sample
fob.SetSelSample(-1, 1, -1, 1);//full sample
fob.SetPrintSfa(TRUE);
MaxControl(1000,10,TRUE);
fob.SetTranslog(inorm);

//Estimate the model, get TE scores
fob.Estimate();
fob.SetConfidenceLevel(0.05);
fob.Renew(fob.TEint(0.05), {"TE", "lower", "upper"});
fob.Renew(fob.Ineff(),{"jlms"});

//fob.Save("out.xls");
fob.TestGraphicAnalysis();

delete fob;
```

```
}
```

The output of this program looks like follows. Note that `Constant` appears several times. The first refers to the production function, the second refers to the parameter σ_u (always after the parameter of σ_v) whereas the last one refers to μ. A direct visual assessment of TE scores is possible when creating a graph using `TestGraphicAnalysis` (see figure 2).

`hbest1.out`

```
Sfa package version 1.0, object created on 19-02-2014
Constructing Squares and Cross-Products...done.
-Pooled model-

---- Sfa ----
The estimation sample is:  1 - 2400
The dependent variable is: lny
The dataset is: USDAafrica.xls
```

	Coefficient	Std.Error	robust-SE	t-value	t-prob
Constant	0.418511	0.01734	0.01604	26.1	0.000
lnlab	0.128542	0.01338	0.01105	11.6	0.000
lnland	0.747665	0.01552	0.01301	57.5	0.000
lnmac	-0.0103591	0.009488	0.008851	-1.17	0.242
lnfert	0.0753081	0.006573	0.006243	12.1	0.000
trend	0.0104214	0.0007006	0.0006763	15.4	0.000
.5*lnlab^2	-0.0555308	0.02432	0.02387	-2.33	0.020
.5*lnland^2	-0.170596	0.02547	0.02843	-6.00	0.000
.5*lnmac^2	-0.0152330	0.005151	0.004632	-3.29	0.001
.5*lnfert^2	0.0611979	0.003107	0.003063	20.0	0.000
.5*trend^2	0.000420185	6.481e-005	6.132e-005	6.85	0.000
lnlab*lnland	0.189014	0.02492	0.02557	7.39	0.000
lnlab*lnmac	-0.125613	0.008138	0.007344	-17.1	0.000
lnlab*lnfert	-0.0294984	0.006109	0.005248	-5.62	0.000
lnlab*trend	-0.000443247	0.0007217	0.0006231	-0.711	0.477
lnland*lnmac	0.137893	0.008829	0.008381	16.5	0.000
lnland*lnfert	-0.0633866	0.006748	0.006383	-9.93	0.000
lnland*trend	-0.000495269	0.0007838	0.0007483	-0.662	0.508
lnmac*lnfert	-0.0135746	0.002997	0.002857	-4.75	0.000
lnmac*trend	0.000810360	0.0002892	0.0002743	2.95	0.003
lnfert*trend	0.000898462	0.0002366	0.0002062	4.36	0.000
ln{\sigma_v}	-2.64680	0.1459	0.1361	-19.4	0.000
Constant	-1.04439	0.04104	0.04791	-21.8	0.000

lnlab	0.232693	0.04300	0.05044	4.61	0.000
lnland	-0.146195	0.04176	0.05050	-2.90	0.004
lnmac	-0.00976602	0.01491	0.01671	-0.584	0.559
lnfert	-0.0149101	0.01372	0.01647	-0.905	0.365
Constant	0.454143	0.02926	0.03249	14.0	0.000

log-likelihood	-458.928611		
no. of observations	2400	no. of parameters	28
AIC.T	973.857222	AIC	0.405773842
mean(lny)	-1.14273	var(lny)	2.98932
\gamma:	0.9618	VAR(u)/VAR(total)	0.9016
Test of one-sided err	172.93	mixed Chi^2 !!	

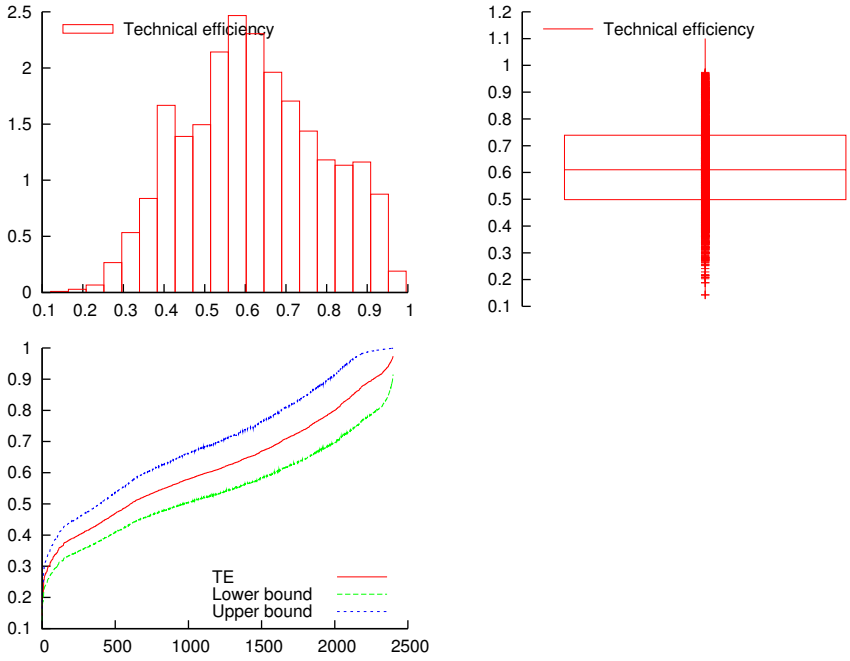

Figure 2: TE scores of the POOLED model.

Example: hbest2.ox

In this example, the `CFE` model of Chen et al. (2014) is specified. You can immediately switch to the `LSDV` or `TFE` model, respectively, by changing the argument of `SetMethod`. The panel structure is identified with `Ident`.

hbest2.ox

```
#include <oxstd.h>
#include <packages/gnudraw/gnudraw.h>
#import  <packages/sfamb/sfamb>

main(){
/*new object of class 'Sfa', Load data*/
decl fob = new Sfa(); fob.Load("USDAafrica.xls");
/*Model specification*/
fob.SetMethod(CFE); fob.SetConstant();
/*Identification of panel structure*/
fob.Ident(fob.GetVar("ID"), fob.GetVar("time"));
/*Data preparation*/
decl inorm = 1;//choose logs; or normalization and logs
fob.Renew(fob.PrepData(fob.GetVar("output"), inorm), "lny");
fob.Renew(fob.PrepData(fob.GetVar("labour"), inorm), "lnlab");
fob.Renew(fob.PrepData(fob.GetVar("land"), inorm), "lnland");
fob.Renew(fob.PrepData(fob.GetVar("machinery"), inorm), "lnmac");
fob.Renew(fob.PrepData(fob.GetVar("fertilizer"), inorm), "lnfert");
fob.Renew(fob.GetVar("time")-meanc(fob.GetVar("time")),"trend");

/*Set up model*/
fob.Select(Y_VAR, {"lny",0,0}); //Select dependent variable

fob.Select(X_VAR, { //Select regressors
"Constant",0,0,
"lnlab",0,0,
"lnland",0,0,
"lnmac",0,0,
"lnfert",0,0,
"trend",0,0
});

//Select estimation sample
fob.SetSelSample(-1, 1, -1, 1);//full sample
fob.SetPrintSfa(TRUE);
MaxControl(1000,10,TRUE);
```

```
fob.SetTranslog(inorm);

//Estimate the model, get TE scores
fob.Estimate();
fob.Renew(fob.TE(),{"TE"});
fob.Renew(fob.Ineff(),{"jlms"});
fob.TestGraphicAnalysis();

delete fob;
}
```

The output of this program looks like follows. The data is again USDAafrica.xls. As one would expect the average TE score is higher under this model.

hbest2.out

```
Sfa package version 1.0, object created on 10-02-2014
#groups:   #periods(max):  avg.T-i:
  48.000       50.000       50.000
Constructing Squares and Cross-Products...done.
-CFE model-

---- Sfa ----
The estimation sample is:  1 - 2400
The dependent variable is: lny
The dataset is: USDAafrica.xls

                  Coefficient  Std.Error  t-value  t-prob
lnlab              0.00883654    0.03048    0.290   0.772
lnland               0.677192    0.02304     29.4   0.000
lnmac                0.106177   0.009083     11.7   0.000
lnfert              0.0837343   0.007086     11.8   0.000
trend               0.00920993  0.0006800     13.5   0.000
.5*lnlab^2            0.138565    0.02083     6.65   0.000
.5*lnland^2           0.177254    0.02047     8.66   0.000
.5*lnmac^2           0.0121082   0.003350     3.61   0.000
.5*lnfert^2          0.0245012   0.002852     8.59   0.000
.5*trend^2         0.000407978 3.744e-005     10.9   0.000
lnlab*lnland         -0.138300    0.02024    -6.83   0.000
lnlab*lnmac         -0.0247345   0.007611    -3.25   0.001
lnlab*lnfert         0.00218990   0.005678    0.386   0.700
lnlab*trend       -0.000134440  0.0005109   -0.263   0.792
lnland*lnmac         0.0243333   0.008190     2.97   0.003
lnland*lnfert       -0.0319551   0.006178    -5.17   0.000
```

```
lnland*trend       0.000212194  0.0004843     0.438    0.661
lnmac*lnfert        0.00379000   0.001959      1.93    0.053
lnmac*trend        0.000346355  0.0001844      1.88    0.060
lnfert*trend      -0.000171510  0.0001308     -1.31    0.190
ln{\sigma_v^2}        -4.94563     0.1464     -33.8    0.000
ln{\sigma_u^2}        -3.44008     0.1125     -30.6    0.000

log-likelihood      1476.81739
no. of observations       2400  no. of parameters          22
AIC.T               -2909.63478 AIC                -1.21234782
mean(lny)         7.55183e-018  var(lny)              0.127523
lambda                   2.123
```

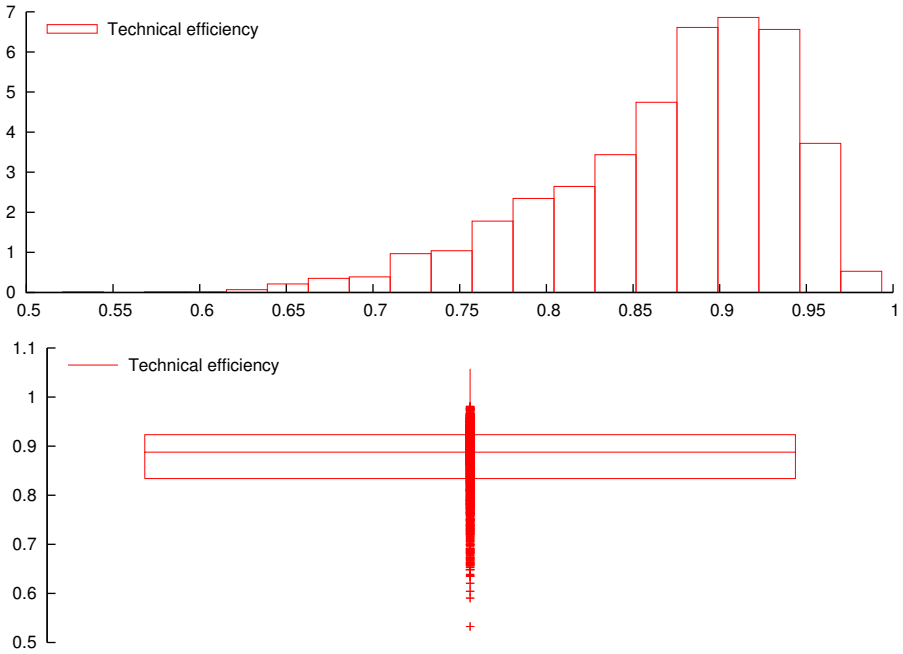

Figure 3: TE scores of the `CFE` model.

Example: member functions `SetTranslog` and `Elast`

The member function `SetTranslog` allows for convenient specification of a translog functional form. In the following, we refer to the current instance of the class as `fob`. Suppose your selection of regressors looks like this:

```
fob.Select(X_VAR, {
"Constant",0,0,
"lnx1",0,0,
"lnx2",0,0,
"lnx3",0,0,
"trend",0,0});
```

The default specification is Cobb-Douglas, i.e. `SetTranslog(0)`, changing the argument to 1 invokes construction of the respective square and cross terms of `X_VAR`. In general notation:

$$ln\, y_i = \beta_0 + \sum_{j=1}^{K} \beta_j\, ln\, x_{ji} + \frac{1}{2} \sum_{j=1}^{K} \sum_{l=1}^{K} \beta_{jl}\, ln\, x_{ji}\, ln\, x_{li} \tag{12}$$

If your selection includes dummies the variables should be ordered like this:

```
fob.Select(X_VAR, {
"Constant",0,0,
"lnx1",0,0,
"lnx2",0,0,
"lnx3",0,0,
"trend",0,0,
"dummy1",0,0,
"dummy2",0,0});
```

Specification of a translog form is then possible by means of `SetTranslog(4)` because only the first four regressors are used (`"Constant"` is ignored automatically).

After estimation the member function `Elast` can be used to calculate the output elasticity (ϵ_{ji}) of each input for each observation:

$$\epsilon_{ji} = \beta_j + \sum_{l=1}^{K} \beta_{jl}\ \ln\ x_{li} \tag{13}$$

The following example illustrates one possible way the function may be used. Here, results are plotted as histograms (see figure 4). Note that indexing starts at 0 in Ox.

```
decl vEps1 = fob.Elast("lnx1");//eps~tval
decl vEps2 = fob.Elast("lnx2");
decl vEps3 = fob.Elast("lnx3");
decl vEpst = fob.Elast("trend");

DrawDensity(0, vEps1[][0]',{"eps1"},1,1,0,0,0,0,1,0,1);
DrawDensity(1, vEps2[][0]',{"eps2"},1,1,0,0,0,0,1,0,1);
DrawDensity(2, vEps3[][0]',{"eps3"},1,1,0,0,0,0,1,0,1);
DrawDensity(3, vEpst[][0]',{"epst"},1,1,0,0,0,0,1,0,1);
ShowDrawWindow();
```

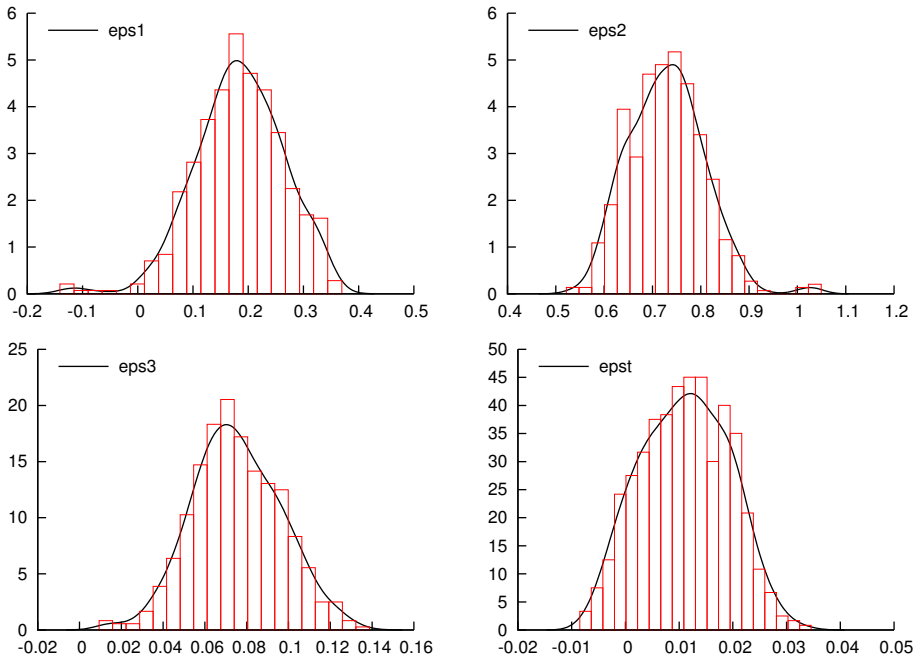

Figure 4: Histograms of calculated elasticities (by observation).

4 SFAMB member functions

These functions (user interface) together with the data members and several other functions build up the SFAMB class. These other functions are not listed here. The interested user may consult the package's header file and source code file.

AiHat AiHat();

Return value

Returns the calculated individual effects $\hat{\alpha}_i$, N x 1 vector.

Description

-Only panel data- These values can be obtained after estimation, see section 3.1 for the respective formulas.

Different functions to extract data:

Return value

Different vectors or matrices.

Description

These functions can be used with convenient (Database) functions such as Save, Renew or savemat.

IDandPer(); is a NT x 2 matrix holding the number of the individual (e.g. 1,1,1,2,...N,N) as well as the individual group size T_i. -Only panel data-

GetLLFi(); returns the individual log-likelihood values. It is a NT x 1 vector for models POOLED and LSDV but a N x 1 vector for the other models.

GetResiduals(); returns the (composed) residual of the respective observation, NT x 1 vector.

GetTldata(); returns the corresponding vectors of Y, X, square and cross terms of X. -Use with SetTranslog()-

GetMeans(); returns the means of Y- and X-variables, N x (k+1) matrix. -Only panel data-

GetWithins(); returns the within-transformed Y- and X-variables, NT x (k+1) matrix. -Only panel data-

DropGroupIf DropGroupIf(const mifr);

No return values

Description

-Only panel data- Allows to exclude a whole individual from the sample if the condition in one (single) period is met. Call after function `Ident`.

mifr is the condition that specifies the observation to be dropped, see the general documentation of `selectifr`.

Elast Elast(const sXname);

Return value

Returns the calculated output elasticity as well as the respective t-value.

Description

-Use with `SetTranslog()`- Only if a translog functional form is used. The observation-specific output elasticity of input k is $\delta \, ln \, y_i / \delta \, ln \, x_{ki}$.

sXname is the name of the corresponding input variable (string).

GetResults GetResults(const ampar, const ameff, const avfct, const amv);

No return values

Description

-Only `POOLED` model- This function can be used to store the results of the estimation procedure for further use. All four arguments should be addresses of variables.

mpar consists of a Npar X 3 matrix, where Npar is the number of parameters in the model. The first column contains the coefficient estimates, the second column the standard errors, and the last the appropriate probabilities.

eff consists of a Nobs X 3 matrix, where Nobs is the number of total observations. The first column holds the point estimate for technical efficiency, the second and third columns contain the upper and lower bound of the (1-alpha) confidence interval.

fct Holds some likelihood function values (OLS and ML), as well as some information on the correct variance decomposition of the composed error term.

v Variance-Covariance-Matrix.

Ident Ident(const vID, const vPer);

No return values

Description

-Only panel data- Identifies the structure of the panel.

vID is a NT x 1 vector holding the identifier (integer) of the individual.

vPer is a NT x 1 vector holding the identifier (integer) of the time period.

Ineff Ineff();

Return value

Returns point estimates of technical inefficiency, NT x 1 vector.

Description

These predictions are given by the conditional expectation of u (MLE), see section 3.1 for details.

PrepData PrepData(const mSel, iNorm);

Return value

Returns logs of the specified variables, either normalized or not.

Description

This function expects your data in levels and can do two things: It takes logs of your specified variables (if `iNorm` =0) or it normalizes your data (by the sample mean if `iNorm` =1) before taking logs. The transformed variable should receive a new name.

mSel is a NT x k matrix holding the respective Y- and X-variables.

iNorm is an integer: 0=no normalization; 1=normalization;

SetConfidenceLevel SetConfidenceLevel(const alpha);

No return values

Description

-Only `POOLED` model- This function expects a double indicating the error probability for the construction of confidence bounds (default 0.05).

SetPrintDetails SetPrintDetails(const bool);

No return values

Description

-Not for `LSDV` model- Prints starting values, warnings and elapsed time if `bool` $\neq 0$.

SetRobustStdErr SetRobustStdErr(const bool);

No return values

Description

-Only `POOLED` model- By default, robust standard errors are used for the cross-sectional model. Use `FALSE` to switch off this setting.

SetStart SetStart(const vStart);

No return values

Description

This function expects a column vector of appropriate size containing starting values for the maximum likelihood iteration[8]. If the function is not called at all, OLS values are used in conjunction with a grid search for the SFA specific parameters σ_u and λ.

SetTranslog SetTranslog(const iTl);

No return values

Description

This function expects an integer to control the construction of additional regressors from the selected X-variables.

- A value of zero indicates no further terms to be added, e.g., for a log-linear model, this corresponds to the Cobb-Douglas form.
- A value of one indicates that all square and cross terms of all independent variables should be constructed, e.g., for a log-linear model, this corresponds to the full translog form.
- An integer value of $k > 1$ indicates that the square and cross terms should be constructed for only the first k independent variables (useful when the regressor matrix contains dummy variables).

TE TE();

Return value

Returns point estimates of technical efficiency, NT x 1 vector.

Description

These predictions are given by the conditional expectation of $exp(-u)$ (MLE), see section 3.1 for details.

[8]Corresponding to the technology parameters. In case of the `TFE` model, a vector of zeros is used for the alphas.

TEint TEint(const dAlpha);

Return value

Returns point estimates of technical efficiency as well as lower and upper bounds.

Description

-Only `POOLED` model- This function expects a double indicating the error probability for the construction of confidence bounds (default 0.05), for details see Horrace and Schmidt (1996), for an application Brümmer (2001). It returns a NT x 3 matrix structured as (point estimate-lower bound-upper bound).

TestGraphicAnalysis TestGraphicAnalysis();

No return values

Description

Only useful in conjunction with the (free) Ox package `GnuDraw`. This function draws two or three graphs, respectively: A histogram of the efficiency point estimates and a boxplot of these estimates. In case of the `POOLED` model: in addition, a sorted graph depicting the interval estimates for technical efficiency at the specified significance.

References

Aigner, D., Lovell, C. A. K. and Schmidt, P. (1977). Formulation and Estimation of Stochastic Frontier Production Function Models. *Journal of Applied Econometrics* 6: 21–37.

Alvarez, A., Amsler, C., Orea, L. and Schmidt, P. (2006). Interpreting and Testing the Scaling Property in Models where Inefficiency Depends on Firm Characteristics. *Journal of Productivity Analysis* 25: 201–212, doi:10.1007/s11123-006-7639-3.

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.

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: 325–332.

Brorsen, B. W. and Kim, T. (2013). Data aggregation in stochastic frontier models: the closed skew normal distribution. *Journal of Productivity Analysis* 39: 27–34, doi:10.1007/s11123-012-0274-2.

Brümmer, B. (2001). Estimating confidence intervals for technical efficiency : the case of private farms in Slovenia. *European Review of Agricultural Economics* 28: 285–306.

Caudill, S. B., Ford, J. M. and Gropper, D. M. (1995). Frontier estimation and Firm-Specific Inefficiency Measures in the Presence of Heteroscedasticity. *Journal of Business & Economic Statistics* 13: 105–111.

Chen, Y.-Y., Schmidt, P. and Wang, H.-J. (2014). Consistent estimation of the fixed effects stochastic frontier model. *Journal of Econometrics* 181: 65–76, doi:10.1016/j.jeconom.2013.05.009.

Coelli, T. J., Rao, P. D., O'Donnell, C. J. and Battese, G. E. (2005). *An Introduction to Efficiency and Productivity Analysis*. Springer, New York.

Doornik, J. A. (2009). *An Object-Oriented Matrix Language Ox 6*. Timberlake Consultants Press.

Doornik, J. A., Arellano, M. and Bond, S. (2012). Panel Data estimation using DPD for Ox. Boston College, http://fmwww.bc.edu/ec-p/software/ox/dpd.pdf.

Doornik, J. A. and Ooms, M. (2006). Introduction to Ox. Http://www.doornik.com/ox/OxIntro.pdf.

Fuglie, K. O. (2012). Productivity Growth and Technology Capital in the Global Agricultural Economy. In Fuglie, K. O., Wang, S. L. and Ball, V. E. (eds), *Productivity Growth in Agriculture: An International Perspective*. CABI.

Greene, W. H. (2005). Reconsidering heterogeneity in panel data estimators of the stochastic frontier model. *Journal of Econometrics* 126: 269–303, doi: 10.1016/j.jeconom.2004.05.003.

Greene, W. H. (2008). The Econometric Approach to Efficiency Analysis. In Fried, H. O., Lovell, C. A. K. and Schmidt, S. S. (eds), *The Measurement of Productive Efficiency and Productivity Growth*. Oxford University Press.

Horrace, W. C. and Schmidt, P. (1996). Confidence Statements for Efficiency Estimates from Stochastic Frontier Models. *Journal of Productivity Analysis* 7: 257–282.

Huang, C. J. and Liu, J.-T. (1994). Estimation of a non-neutral stochastic frontier production function. *Journal of Productivity Analysis* 5: 171–180, doi: 10.1007/BF01073853.

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.

Kodde, D. A. and Palm, F. C. (1986). Wald criteria for jointly testing equality and inequality restrictions. *Econometrica* 54: 1243–1248.

Kumbhakar, S. C., Gosh, S. and McGuckin, J. T. (1991). A Generalized Production Frontier Approach for Estimating Determinants of Inefficiency in U.S. Dairy Farms. *Journal of Business and Economic Statistics* 9: 279–286.

Kumbhakar, S. C. and Lovell, C. A. K. (2000). *Stochastic Frontier Analysis*. Cambridge University Press.

Lai, H.-p. and Huang, C. J. (2010). Likelihood ratio tests for model selection of stochastic frontier models. *Journal of Productivity Analysis* 34: 3–13, doi: 10.1007/s11123-009-0160-8.

Meeusen, W. and Broeck, J. van den (1977). Efficiency Estimation from Cobb-Douglas production functions with composed error. *International Economic Review* 18: 435–444.

Reifschneider, D. and Stevenson, R. (1991). Systematic Departures from the Frontier: A Framework for the Analysis of Firm Inefficiency. *International Economic Review* 32: 715–723.

Schmidt, P. and Sickles, R. C. (1984). Production Frontiers and Panel Data. *Journal of Business & Economic Statistics* 2: 367–374.

Wang, H.-J. and Ho, C.-W. (2010). Estimating fixed-effect panel stochastic frontier models by model transformation. *Journal of Econometrics* 157: 286–296, doi:10.1016/j.jeconom.2009.12.006.