Fitting spatial stochastic frontier models in Stata

Kerui Du
School of Management
Xiamen University
Xiamen, China
kerrydu@xmu.edu.cn

Luis Orea
Department of Economics
School of Economics and Business
University of Oviedo
Oviedo, Spain
lorea@uniovi.es

Inmaculada C. Álvarez
Department of Economics
Universidad Autónoma de Madrid
Madrid, Spain
inmaculada.alvarez@uam.es

Abstract. In this article, we introduce a new command xtsfsp for fitting spatial stochastic frontier models in Stata. Over the last decades, an important theoretical progress of stochastic frontier models is the incorporation of various types of spatial components. Models with the ability to account for spatial dependence and spillovers have been developed for efficiency and productivity analysis, drawing extensive attention from industry and academia. Due to the unavailability of the statistical packages, the empirical applications of the new stochastic frontier models appear to be lagging. The xtsfsp command provides a routine for estimating the spatial stochastic frontier models in the style of Orea and Álvarez (2019) and Galli (2022), enabling users to handle different sources of spatial dependence. In the presented article, we introduce the spatial stochastic frontier models, describe the syntax and options of the new command, and provide several examples to illustrate its usage.

Keywords: stochastic frontier models, SFA, spatial dependence, technical efficiency, spillovers

1 Introduction

Producers might fail in optimizing their production activities, causing deviation from the maximum output or the minimum cost. Economic researchers proposed the concept of technical efficiency, which measures how well a producer is utilizing its resources to produce goods or services. A technically efficient organization makes the maximum outputs given the amount of inputs or uses the minimum amount of inputs to produce a given level of output. On the contrary, technically inefficient organization produce fewer outputs given the same inputs or uses more inputs than necessary to produce the same output. Technical efficiency is important because it allows organizations or economies to achieve their goals with the least amount of resources possible, which can lead to cost savings and increased profitability.

Aigner et al. (1977) and Meeusen and van Den Broeck (1977) introduced stochastic frontier models for evaluating technical efficiency. The essential concept behind these models is to divide the observed output of a production process into two components, namely the ”frontier” output, signifying the maximum feasible output, given the inputs utilized in the production process, and the ”residual” output, denoting the production process’s inefficiency. Following these initial works, stochastic frontier models gained extensive use as a tool for scrutinizing productivity and efficiency.

Methodologically, econometricians have expanded the horizons of stochastic frontier models in various directions. To name a few, Battese and Coelli (1995) incorporated the determinants of inefficiency. Wang (2003) developed the stochastic frontier model with scaling properties to capture the shape of the distribution of inefficiency. Greene (2005) extended the stochastic models with the random effects and the “true” fixed effects. Belotti and Ilardi (2018), Chen et al. (2014), and Wang and Ho (2010) circumvented the ”incidental parameters problem” in the fixed effects stochastic frontier model through model transformation. Karakaplan and Kutlu (2017) developed an endogenous stochastic frontier model to control for the endogeneity in the frontier or inefficiency.

In recent years, stochastic frontier models have undergone further extension to account for spatial dependence and spatial spillover effects. Glass et al. (2016) constructed a spatial Durbin stochastic model considering both global and local spatial dependence. Kutlu et al. (2020) proposed a spatial stochastic frontier model with endogenous frontier and environmental variables. Glass et al. (2016) and Kutlu et al. (2020) combine the concepts of spatial econometrics and stochastic frontier analysis by including the spatial lag of the dependent variable. On the other hand, Orea and Álvarez (2019) developed a new stochastic frontier model with spatial correlation in both noise and inefficiency terms. Galli (2022) integrated the two different modeling ideas to specify four different sources of spatial dependence fully.

With the increasing demand in the last decades to analyze technical efficiency, Stata provides official commands frontier and xtfrontier for cross-sectional and panel stochastic model estimation, respectively. Belotti et al. (2013) developed sfcross and sfpanel commands accommodating more different distribution assumptions and allowing fixed-effect and random-effect models with the consideration of heteroscedasticity. Karakaplan (2017) introduced the sfkk command for estimating endogenous stochastic frontier models. Karakaplan (2018) supplemented the xtsfkk command for fitting the endogenous panel stochastic frontier model. Kumbhakar et al. (2015) provides a practitioner’s guide to stochastic frontier analysis with a suite of Stata commands (including sfmodel, sfpan, sf_fixeff, and sfprim).

In this article, we introduce spxtsfa, a new command for fitting spatial stochastic frontier models in the style of Orea and Álvarez (2019) and Galli (2022). The proposed spxtsfa command not only allows getting more accurate inefficiency scores (see e.g. Orea et al. 2018) but also examining relevant economic issues that a non-spatial stochastic frontier model tends to overlook.

For instance, in microdata applications, the new command can be used to test whether the production/cost function can be viewed as a purely deterministic (engineering) process where the firm controls all the inputs (see e.g. Druska and Horrace 2004). A distinctive feature of the spxtsfa command is that it allows estimating a stochastic frontier model with cross-sectional correlation in the inefficiency term, a specification that is useful in applications where some firms benefit from best practices implemented in adjacent firms due to, for instance, agglomeration economies, knowledge spillovers, technology diffusion or R&D spillovers. This could especially be the case if (local) firms belong to communitarian networks (e.g. cooperatives) or common technicians (consultants) are advising all local firms. In practice, the proposed spxtsfa command can be useful to capture a kind of behavioral correlation, for instance when firms tend to "keep an eye" on the decisions of other peer firms trying to overcome the limitations caused by the lack of information or they simply emulate each other. It is finally germane to mention that the spxtsfa command also allows capturing cross-sectional effects that might be caused by non-spatial factors (e.g., the regulation environment) if we define appropriately the so-called weight (W) matrix. A proper definition of the W matrix might, for instance, allow us to examine the existence of knowledge spillovers from supplier and user firms.

As Orea and Álvarez (2019) point out, the proposed spxtsfa command can be implemented using macro-level data (e.g. data of countries, regions or industries) due to the abundant evidence of important feedback processes between neighboring or non-distant regions justify the use of SAR and Durbin frontier functions in macrodata applications. The spatial weight matrix specification commonly adopted in regional economics is based on geographical distance. However, as aforementioned, the weight matrix can be defined using a non-spatial criterion. In this sense, Liu and Sickles (2023) state that the mode of production in the world economy is characterized by the division of global value chains (GVCs) and, hence, the spatial weight matrix should be constructed using the economic distance between industries within/across national economies. In this case, the proposed spxtsfa command can be used to estimate spatial SAR and Durbin frontier functions in order to examine the diffusion of knowledge and technology among the participants in the international production network. It is also makes sense to estimate a stochastic frontier model with cross-sectional correlation in the inefficiency term using macrodata if we change the interpretation of the estimated correlation. In these applications, the spatial correlation in the inefficiency term likely captures barriers and distortions to the efficient allocation of resources across firms that are common to several regions, such as regulation, labor market trends or common institutions (see e.g. Orea et al. 2023).

The remainder of this article unfolds as follows: Section 2 provides a brief description of the models in Orea and Álvarez (2019) and Galli (2022); Section 3 explain the syntax and options of spxtsfa; Section 4 and 5 present simulated data examples to illustrate the usage of the command; and section 6 concludes the article.

2 The model

In this section, we briefly describe the spatial stochastic frontier models developed by Orea and Álvarez (2019) and Galli (2022). The exposition here is only introductory. Please refer to the cited papers for more technical details.

Based on the transposed version of Wang and Ho (2010) model, Orea and Álvarez (2019) proposed a spatial stochastic frontier model which accommodates spatially-correlated inefficiency and noise terms. The model is formulated as in Eqs.(1)-(3), for $i = 1, ..., N$ and $t = 1, .., T$:

$$Y_{it} = X_{it}'\beta + \tilde{v}_{it} - s\tilde{u}_{it} \tag{1}$$

$$\tilde{v}_{it} = v_{it} + \gamma W_i^{vt}\tilde{v}_{.t} \tag{2}$$

$$\tilde{u}_{it} = u_{it} + \tau W_i^{ut}\tilde{u}_{.t} \tag{3}$$

Eq.(1) describes the stochastic frontier function where Y_{it} is the dependent variable and X_{it} is a $k \times 1$ vector of variables shaping the frontier; $s = 1$ for the production function and $s = -1$ for the cost function; $\tilde{v}_{it}$ and $\tilde{u}_{it}$ represent idiosyncratic noise and inefficiency, respectively. In Eqs.(2) and (3), $W_i^{vt} = (W_{i1}^{vt}, ..., W_{iN}^{vt})$ and $W_i^{vt} = (W_{i1}^{vt}, ..., W_{iN}^{vt})$ are two known $1 \times N$ cross-sectional weight vectors depicting the structure of the cross-sectional relationship for idiosyncratic noise and inefficiency terms, respectively; $\tilde{v}_{.t} = (\tilde{v}_{1t}, ..., \tilde{v}_{Nt})'$ and $\tilde{u}_{.t} = (\tilde{u}_{1t}, ..., \tilde{u}_{Nt})'$; v_{it} is a random variable following the distribution $N(0, \sigma_{v,it}^2)$ and $u_{it} = h(Z_{it}'\delta)u_t^*$. $h(Z_{it}'\delta) = \sqrt{exp(Z_{it}'\delta)}$ is the scaling function where Z_{it} is a $l \times 1$ vector of variables affecting individuals' inefficiency and u_t^* is a non-negative random variable following the distribution $N^+(0, 1)$. Different from the original setting in Orea and Álvarez (2019), we assume $\sigma_{v,it}^2 = exp(D_{it}'\eta)$ to account for idiosyncratic error variance which depends on a vector of variables D_{it}[1]. Furthermore, we enforce the variance of u_t^* to be equal to one, thus allowing the term Z_{it} to include a constant. Using matrix notation, we can rewrite Eqs.(2) and (3) as

$$\tilde{v}_{.t} = (I_N - \gamma W^{vt})^{-1}v_{.t} \tag{4}$$

$$\tilde{u}_{.t} = (I_N - \tau W^{ut})^{-1}h(Z_{.t}\delta)u_t^* = \tilde{h}_{.t}u_t^* \tag{5}$$

where $Z_{.t} = (Z_{1t}, ..., Z_{Nt})'$;$\tilde{h}_{.t} = (I_N - \tau W^{ut})^{-1}h(Z_{.t}\delta)$.

The above model captures the spatial correlation of the random error and inefficiency terms with the spatial autoregressive (SAR) process [2]. Referring

[1] If $D_{it} = 1$, the error term is conditional homoscedasticity.

[2] Orea and Álvarez (2019) also considered a specification of the spatial moving average process.

to Wang and Ho (2010), we can obtain the following log-likelihood function for each period t:

$$\begin{aligned}\ln L_t = & -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\Pi_t| - \frac{1}{2}\tilde{\varepsilon}_{.t}\Pi^{-1}\tilde{\varepsilon}_{.t} \\ & + \frac{1}{2}\left(\frac{\mu_*^2}{\sigma_*^2}\right) + \ln\left[\sigma_*\Phi\left(\frac{\mu_*}{\sigma_*}\right)\right] - \ln\left(\frac{1}{2}\right)\end{aligned} \tag{6}$$

where $\Pi_t = (I_N - \rho W^{yt})^{-1} diag(\sigma_{v,.t}^2)[(I_N - \rho W^{yt})^{-1}]'$ [3]; $\tilde{\varepsilon}_{.t} = (\tilde{\varepsilon}_{1t}, ..., \tilde{\varepsilon}_{Nt})', \tilde{\varepsilon}_{it} = s(Y_{it} - X_{it}'\beta)$, and

$$\mu_* = \frac{-\tilde{\varepsilon}'_{.t}\Pi_t^{-1}\tilde{h}_{.t}}{\tilde{h}'_{.t}\Pi_t^{-1}\tilde{h}_{.t} + 1} \tag{7}$$

$$\sigma_*^2 = \frac{1}{\tilde{h}'_{.t}\Pi_t^{-1}\tilde{h}_{.t} + 1} \tag{8}$$

Galli (2022) further incorporated the spatial lags of the dependent variable and the input variables into Orea and Álvarez (2019) model, which additionally measures global and local spatial spillovers affecting the frontier function. The model is expressed as

$$Y_{it} = \rho W_i^{yt} Y_{.t} + X_{it}'\beta + W_i^{xt} X_{.t}\theta + \tilde{v}_{it} + s\tilde{u}_{it} \tag{9}$$

where $W_i^{yt} = (W_{i1}^{yt}, ..., W_{iN}^{yt})$ and $W_i^{xt} = (W_{i1}^{xt}, ..., W_{iN}^{xt})$ are two known $1 \times N$ cross-sectional weight vectors [4]; $Y_{.t} = (Y_{1t}, ..., Y_{Nt})'$; $X_{.t} = (X_{1t}, ..., X_{Nt})'$. This model gives rise to the following log-likelihood function for each period t:

$$\begin{aligned}\ln L_t = & ln|I_N - \rho W^{yt}| - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\Pi_t| - \frac{1}{2}\tilde{\varepsilon}_{.t}\Pi^{-1}\tilde{\varepsilon}_{.t} \\ & + \frac{1}{2}\left(\frac{\mu_*^2}{\sigma_*^2}\right) + \ln\left[\sigma_*\Phi\left(\frac{\mu_*}{\sigma_*}\right)\right] - \ln\left(\frac{1}{2}\right)\end{aligned} \tag{10}$$

where $\tilde{\varepsilon}_{.t} = (\tilde{\varepsilon}_{1t}, ..., \tilde{\varepsilon}_{Nt})', \tilde{\varepsilon}_{it} = s(Y_{it} - X_{it}'\beta - \rho W_i^{yt} Y_{.t} - W_i^{xt} X_{.t}\theta)$.

Summing the time-specific log-likelihood functions over all periods yields the overall likelihood function for the whole sample, i.e., $lnL = \sum_{t=1}^{T} lnL_t$. Then, numerically maximize the overall log-likelihood function to obtain consistent estimates of the parameters in the above models. Specifically, we use Stata `ml model` routine with the `method-d0` evaluator to program the `spxtsfa` command. Following Gude et al. (2018), we parameterize ρ, γ, and τ as Eq.(11) to ensure the standard regularity condition for the spatial autoregressive models.

$$\begin{aligned}\eta &= \left(\frac{1}{r_{\min}}\right)(1-p) + \left(\frac{1}{r_{\max}}\right)p \\ 0 \le p &= \frac{\exp(\delta_0)}{1 + \exp(\delta_0)} \le 1\end{aligned} \tag{11}$$

[3] The term $diag(\sigma_{v,.t}^2)$ represents a diagonal matrix with $\sigma_{v,.t}^2 = (\sigma_{v,1t}^2, ..., \sigma_{v,Nt}^2)'$ as the diagonal elements

[4] We index W_i^{yt}, W_i^{xt}, W_i^{ut}, and W_i^{vt} with superscript yt, xt, ut, and vt, respectively. This indicates the spatial weight matrix can be time-varying and different across various spatial components

where η stands for one of ρ, γ, and τ; $r_{\min}$ and $r_{\max}$ are respectively the minimum and maximum eigenvalues of the corresponding spatial weight matrix.

In summary, Galli (2022) provided a fully comprehensive specification of four different types of spatial dependence: global spillovers of dependent variable Y_{it}, local spillovers of input variables X_{it}, cross-sectional correlation of idiosyncratic noise v_{it} and inefficiency u_{it}. We term this full model "*yxuv*-SAR". Some restrictions can be imposed on the specific parameters to generate the following models (summarized in Table 1), which can be estimated by the `spxtsfa` command.

Table 1: Specific models with restricted parameters

	yuv	*xuv*	*yv*	*yu*	*y*	*xuv*	*xv*	*xu*	*uv*	*u*	*v*
ρ		0				0	0	0	0	0	0
θ	0		0	0	0				0	0	0
γ				0	0			0		0	
τ			0		0		0				0

Note: 0 indicates the corresponding parameter is restricted to be zero.

3 The xtsfsp command

`xtsfsp` estimates spatial stochastic frontier models in the style of Orea and Álvarez (2019) and Galli (2022).

3.1 Syntax

Estimation syntax

```
xtsfsp depvar [indepvars], [uhet(varlist[,noconstant])
   vhet(varlist[,noconstant]) cost noconstant wy(wyspec) wx(wxspec)
   wu(wuspec) wv(wvspec) normalize(norm_method) wxvars(varlist)
   initial(matname) mlmodel(model_options) mlsearch(search_options)
   mlplot mlmax(maximize_options) nolog mldisplay(display_options)
   level(#) te(newvarname) genwxvars delmissing
   constraints(constraints) ]
```

Version syntax

```
xtsfsp , version
```

Replay syntax

```
xtsfsp [, level(#) ]
```

3.2 Options

`uhet(`*varlist*`[,noconstant])` specifies explanatory variables for scaling function depending on a linear combination of *varlist*. It is required.

`vhet(`*varlist*`[,noconstant])` specifies explanatory variables for idiosyncratic error variance function depending on a linear combination of *varlist*.

`noconstant` suppresses constant term.

`cost` specifies the frontier as a cost function. By default, the production function is assumed.

`wy(`*wyspec*`)` specifies the spatial weight matrix for lagged dependent variable. The expression is wy(W_1 [$W_2 ... W_T$] [,*mata array*]). By default, the weight matrices are `Sp` objects created by Stata offical command spmatrix. mata declares weight matrices are mata matrices. If one weight matrix is specified, it assumes a time-constant weight matrix. For time-varying cases, T weight matrices should be specified in time order. Alternatively, using array to declare weight matrices are stored in an array. If only one matrix is stored in the specified array, the time-constant weight matrix is assumed. Otherwise, the keys of the array specify time information, and the values store time-specific weight matrices.

`wx(`*wxspec*`)` specifies the spatial weight matrix for lagged independent variable. The expression is the same as `wy(`*wyspec*`)`.

`wu(`*wuspec*`)` specifies the spatial weight matrix for lagged independent variable. The expression is the same as `wy(`*wyspec*`)`.

`wv(`*wvspec*`)` specifies the spatial weight matrix for lagged independent variable. The expression is the same as `wy(`*wyspec*`)`.

`normalize(`*norm_method*`)` specifies one of the four available normalization techniques: row, col, minmax, and spectral.

`wxvars(`*varlist*`)` specifies spatially lagged independent variables.

`initial(`*matname*`)` specifies the initial values of the estimated parameters with matrix *matname*.

`mlmodel(`*model_options*`)` specifies the `ml model` options.

`mlsearch(`*search_options*`)` specifies the `ml search` options.

`mlplot` specifies using `ml plot` to search better initial values of spatial dependence parameters.

`mlmax(`*maximize_options*`)` specifies the `ml maximize` options.

`nolog` suppresses the display of the criterion function iteration log.

`mldisplay(`*display_options*`)` specifies the `ml display` options.

`level(`*#*`)` sets confidence level; default is level(95).

`te(`*newvarname*`)` specifies a new variable name to store the estimates of technical efficiency.

`genwxvars` generates the spatial Durbin terms. It is activated only when `wxvars(`*varlist*`)` is specified.

`delmissing` allows estimation when missing values are present by removing the corresponding units from spatial matrix.

`constraints(`*constraints)* specifies linear constraints for the estimated model.

3.3 Dependency of xtsfsp

`xtsfsp` depends on the *moremata* package contributed by Jann (2005). If not already installed, you can install it by typing ssc install moremata.

4 Examples

In this section, we use simulated data to exemplify the use of the *xtsfsp* command. Referring to Galli (2022), we first consider the *yxuv*-SAR model specified by the following data-generating process (DGP 1) with $i = 1, ..., 300$ and $t = 1, .., 20$,

$$y_{it} = 0.3W_i y_{.t} + 2x_{it} + 0.5W_i x_{.t} + \tilde{v}_{it} - \tilde{u}_{it} \tag{12}$$

where $\tilde{v}_{it}$ and $\tilde{u}_{it}$ are defined as in Eqs.(2) and (3) with $\gamma = 0.3$, $\tau = 0.3$, $Z_{it} = (z_{it}, 1)'$,$\delta = (2, ln(0.2))'$, $D_{it} = 1$ and $\eta = ln(0.2)$. All the spatial matrices for the four spatial components are the same and time-invariant, created from a binary contiguity spatial weight matrix. We generate the exogenous variables X_{it} and z_{it} from the standard normal distribution, respectively. With the sample generated by DGP 1, we can fit the model in the following syntax.

```
. use xtsfsp_ex1.dta

. xtset id t

Panel variable: id (strongly balanced)
 Time variable: t, 1 to 20
         Delta: 1 unit

. * importing spatial weight matrix from xtsfsp_wmat1.mmat
. mata mata matuse xtsfsp_w1.mmat,replace
(loading w1[300,300])

. * fitting the model
. xtsfsp y x, uhet(z) wu(w1,mata) wy(w1,mata) wv(w1,mata) wx(w1,mata) wxvars(x) te(te) nolog

Spatial frontier model(yxuv-SAR)                     Number of obs =     6,000
                                                     Wald chi2(2)  = 113733.00
Log likelihood = -3953.6034                          Prob > chi2   =    0.0000

------------------------------------------------------------------------------
           y | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
frontier     |
           x |   2.006736   .0082953   241.91   0.000     1.990477    2.022994
         W_x |   .5457545   .0709898     7.69   0.000     .4066171    .6848919
       _cons |   1.008833   .0460347    21.91   0.000     .9186065    1.099059
-------------+----------------------------------------------------------------
    /lnsigv2 |  -1.566886   .0184019   -85.15   0.000    -1.602953   -1.530819
-------------+----------------------------------------------------------------
uhet         |
           z |   .9858524   .0385881    25.55   0.000      .910221    1.061484
       _cons |  -1.533232   .3361919    -4.56   0.000    -2.192156   -.8743081
-------------+----------------------------------------------------------------
Wy           |
       _cons |     .57773    .062202     9.29   0.000     .4558163    .6996436
-------------+----------------------------------------------------------------
Wv           |
       _cons |   .6066382   .0743834     8.16   0.000     .4608494     .752427
------------------------------------------------------------------------------
```

Wu						
_cons	.6584973	.0938095	7.02	0.000	.474634	.8423606
rho	.2810617	.0286408	9.81	0.000	.2240201	.3361839
gamma	.2943177	.033966	8.67	0.000	.2264087	.3593786
tau	.3178137	.042162	7.54	0.000	.2329366	.3978845

```
Note: Wy:_cons, Wv:_cons and Wu:_cons are the transfromed parameters;
      rho, gamma and tau are their origin metrics in spatial components, repsectively.
      W_(x) represent Spatial Durbin terms W(x)
```

The output shows that the command fits six equations with `ml model`. The frontier equation has three explanatory variables x_{it}, $W_i x_{.t}$ and constant. The scaling function uhet() has two explanatory variables Z_{it} and constant. The equation /lnsigv2 is constructed for the variance parameter σ_v^2 which is transformed by the function $exp(\cdot)$. Three Equations (Wy, Wu, and Wv) handle the spatial dependence parameters ρ, τ, and γ, which are parameterized as Eq.(11). We directly include the spatial Durbin term $W_i x_{.t}$ in the frontier equation (represented as W__x) such that we do not need to fit a separate equation. The bottom of the table reports the transformed parameters in the original metric.

Secondly, we consider a restricted model *xuv*-SAR with different spatial weight matrices, one of which is time-varying, and the others are time-constant. The model is described as DGP 2:

$$y_{it} = 1 + 2x_{it} + 0.5W_i^{xt} x_{it} + \tilde{v}_{it} + \tilde{u}_{it}, i = 1, .., 300; t = 1, .., 10 \qquad (13)$$

where the other parameters are set the same as the DGP 1 except for $W_i^{ut} = W_i^u$, $W_i^{vt} = W_i^v$ and $\delta = (4, ln(0.2))'$. Different from DGP 1, which set the production function frontier, DGP 2 specifies a cost function. The estimation of the model is shown as follows.

```
. * importing spatial weight matrices from xtsfsp_w2.mmat
. mata mata matuse xtsfsp_w2.mmat,replace
(loading w1[300,300], w10[300,300], w2[300,300], w3[300,300], w4[300,300], w5[300,300], w6[300,300], w7[300

. local w w1 w2 w3 w4 w5 w6 w7 w8 w9 w10

. use xtsfsp_ex2.dta

. xtset id t

Panel variable: id (strongly balanced)
 Time variable: t, 1 to 10
         Delta: 1 unit

. * initial values for estimated parameters
. mat b=(2,0.5,1,-1.5,4,-1.5,0.6,0.6)

. * fitting the model
. xtsfsp y x, cost uhet(z) wu(w2,mata) wv(w1,mata) wxvars(x) wx(`w´,mata) init(b) genwxvars nolog

Spatial frontier model(xuv-SAR)                        Number of obs =    3,000
                                                       Wald chi2(2)  = 57441.23
Log likelihood = -1911.966                             Prob > chi2   =   0.0000
```

y	Coefficient	Std. err.	z	P>|z|	[95% conf. interval]	
frontier						
x	1.995755	.0083855	238.00	0.000	1.97932	2.01219

```
         W_x     .5069274   .0221354    22.90   0.000     .4635428    .5503121
       _cons     .9917396   .0127299    77.91   0.000     .9667894     1.01669
-------------+----------------------------------------------------------------
    /lnsigv2    -1.614693   .0260288   -62.03   0.000    -1.665708   -1.563677
-------------+----------------------------------------------------------------
uhet         |
           z     3.999532    .002155  1855.90   0.000     3.995308    4.003755
       _cons    -1.699077   .4472546    -3.80   0.000     -2.57568   -.8224746
-------------+----------------------------------------------------------------
Wv           |
       _cons     .5877882   .0584159    10.06   0.000     .4732951    .7022812
-------------+----------------------------------------------------------------
Wu           |
       _cons     .6214134   .0008532   728.34   0.000     .6197412    .6230857
-------------+----------------------------------------------------------------
         tau     .3010498   .0003879   776.12   0.000     .3002893    .3018098
       gamma     .2856864   .0268209    10.65   0.000     .2323035     .337353
------------------------------------------------------------------------------
Note: Wv:_cons and Wu:_cons are the transformed parameters;
      gamma and tau are their origin metrics in spatial components, repsectively.
      W_(x) represent Spatial Durbin terms W(x)
```

In the second example, we use `cost` option to specify the type of frontier. The matrix `b` is used as the initial value for the maximum likelihood estimation. The likelihood function of spatial stochastic frontier models is complicated, and generally difficult to obtain the optimal global solutions. Thus, good initial values would be helpful for fitting spatial stochastic models. Practitioners might fit the non-spatial stochastic models using `fronteir` and `sfpanel` commands to obtain the initial values of the parameters involved in the frontier and the scaling function and then use the `mlplot` option to search initial values for spatially-correlated parameters.

To show the usage of the *delmissing* option, we replace the two observation of Y_{it} with missing values and re-run the above codes which gives rise to error information "*missing values found. use delmissing to remove the units from the spmatrix*". The inclusion of the *delmissing* option addresses this issue and the generated variable ___e_sample___ records the regression sample.

```
. * replace some observations of y to be missing
. replace y=. if _n==1 | _n==100
(2 real changes made, 2 to missing)

. local w w1 w2 w3 w4 w5 w6 w7 w8 w9 w10

. * estimation is aborted
. xtsfsp y x, cost uhet(z) wu(w2,mata) wv(w1,mata) wxvars(x) wx(`w´,mata) init(b) nolog
missing values found. use delmissing to remove the units from the spmatrix
invalid syntax
```

```
. * re-estimation with delmissing option
. local w w1 w2 w3 w4 w5 w6 w7 w8 w9 w10

. xtsfsp y x, cost uhet(z) wu(w2,mata) wv(w1,mata) wxvars(x) wx(`w´,mata) init(b) delmissing nolog
missing values found. The corresponding units are deleted from the spmatrix

Spatial frontier model(xuv-SAR)                      Number of obs =    2,998
                                                     Wald chi2(2)  = 57427.27
```

```
Log likelihood = -1909.3843                              Prob > chi2   =   0.0000

------------------------------------------------------------------------------
           y | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
frontier     |
           x |   1.996019   .0083865   238.00   0.000     1.979582    2.012456
         W_x |   .5065641   .0221317    22.89   0.000     .4631868    .5499415
       _cons |   .9914628   .0127155    77.97   0.000     .9665409    1.016385
-------------+----------------------------------------------------------------
    /lnsigv2 |  -1.615516   .0260354   -62.05   0.000    -1.666545   -1.564488
-------------+----------------------------------------------------------------
uhet         |
           z |   3.999476   .0021537  1857.00   0.000     3.995254    4.003697
       _cons |   -1.69889   .4472536    -3.80   0.000    -2.575491   -.8222893
-------------+----------------------------------------------------------------
Wv           |
       _cons |   .5864895   .0584601    10.03   0.000     .4719098    .7010692
-------------+----------------------------------------------------------------
Wu           |
       _cons |   .6214135   .0008526   728.87   0.000     .6197425    .6230845
-------------+----------------------------------------------------------------
         tau |   .3010498   .0003876   776.69   0.000     .3002899    .3018093
       gamma |     .28509   .0268512    10.62   0.000     .2316482    .3368159
------------------------------------------------------------------------------
Note: Wv:_cons and Wu:_cons are the transformed parameters;
      gamma and tau are their origin metrics in spatial components, repsectively.
      W_(x) represent Spatial Durbin terms W(x)
      Missing values found
      The regression sample recorded by variable __e_sample__
```

Thirdly, we consider the restricted model *uv*-SAR with time-varying spatial weight matrices and conditional heteroscedasticity of random errors. The DGP 3 is described as

$$\begin{aligned} &y_{it} = 1 + 2x_{it} + \tilde{v}_{it} - \tilde{u}_{it}, i = 1, .., 300; t = 1, .., 10 \\ &\sigma^2_{v,it} = exp(1 + d_{it}) \\ &h(Z'_{it}\delta) = \sqrt{exp(1 + z_{it})} \end{aligned} \tag{14}$$

where the other parameters are set the same as the DGP 1 except for $W_i^{ut} = W_i^{vt} = W_i^t$. The following syntax estimates the model alongside the results.

```
. * importing spatial weight matrices from xtsfsp_w3.mmat
. mata mata matuse xtsfsp_w3,replace
(loading w1[300,300], w10[300,300], w2[300,300], w3[300,300], w4[300,300], w5[300,300], w6[300,300], w7[300

. local w w1 w2 w3 w4 w5 w6 w7 w8 w9 w10

. use xtsfsp_ex3.dta

. xtset id t

Panel variable: id (strongly balanced)
 Time variable: t, 1 to 10
         Delta: 1 unit

. xtsfsp y x, wu(`w´,mata) wv(`w´,mata) uhet(z) vhet(d) nolog

Spatial frontier model(uv-SAR)                            Number of obs =   3,000
                                                          Wald chi2(1)  = 7472.38
Log likelihood = -5803.0516                               Prob > chi2   =  0.0000
```

```
------------------------------------------------------------------------------
           y | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
frontier     |
           x |   1.988108   .0229991    86.44   0.000     1.943031    2.033185
       _cons |   .8997916   .0737763    12.20   0.000     .7551927    1.044391
-------------+----------------------------------------------------------------
lnsigv2      |
           d |   .9897965   .0259185    38.19   0.000     .9389972    1.040596
       _cons |   .9950927   .0259519    38.34   0.000      .944228    1.045957
-------------+----------------------------------------------------------------
uhet         |
           z |   1.019351   .0458701    22.22   0.000     .9294473    1.109255
       _cons |   1.122574   .4629594     2.42   0.015     .2151904    2.029958
-------------+----------------------------------------------------------------
Wv           |
       _cons |   .6171649   .0431991    14.29   0.000     .5324961    .7018337
-------------+----------------------------------------------------------------
Wu           |
       _cons |   .6471479   .0703691     9.20   0.000      .509227    .7850688
-------------+----------------------------------------------------------------
         tau |    .299117   .0196647    15.21   0.000     .2601042    .3371547
       gamma |   .3127036   .0317402     9.85   0.000     .2492256    .3735057
------------------------------------------------------------------------------
Note: Wv:_cons and Wu:_cons are the transformed parameters
      gamma and tau are their origin metrics in spatial components, repsectively.
```

Finally, we use the real data to show how to conduct the empirical studies with the models and the command described above. We collect the raw data on China's provinces from the CSMAR database including GDP (denoted as Y), investment, labor force (denoted as L), the ratio of government expenditure to GDP (denoted as fiscal), the ratio of FDI to GDP (denoted as fdi),, and the trade as a share of GDP (denoted as trade). All nominal variables are deflated into the constant price in 1997. The capital stocks by provinces are estimated with the PIM method. The production function is approximated by the translog function and the scaling function of the inefficiency term is assumed to be determined by the ratio of government expenditure to GDP, the ratio of FDI to GDP, and the trade as a share of GDP .

```
. * Translate shapefile to Stata format
. spshape2dta province
.
. use province
. drop if _ID == 26 | _ID>31
(4 observations deleted)
. spset
      Sp dataset: province.dta
Linked shapefile: province_shp.dta
            Data: Cross sectional
 Spatial-unit ID: _ID
     Coordinates: _CX, _CY (planar)
. * Create spatial contiguity matrix
. spmatrix create contiguity w_con, normalize(none)
```

```
  weighting matrix in w_con contains 1 island
. * Obtain spatial matrix as Mata matrix wm from w_con
. spmatrix matafromsp wm id = w_con
. * Match the iland (_ID = 21) with the nearest province(_ID =19)
. mata: wm[19,21]=1
. mata: wm[21,19]=1
. * Create spmatrix w_con from Mata matrix wm and rwo-normalized the matrix
. spmatrix spfrommata w_con= wm id, normalize(row) replace
. * Obtain the new spatial matrix as Mata matrix wm from w_con
. spmatrix matafromsp wm id = w_con
.
. use chnempirical.dta,clear
. * Generate varables for the translog function
. qui translog Y K L , time(year) norm
. global x  lnK lnL _t lnK_lnL _t_lnK _t_lnL _t_2 lnK_2 lnL_2
. global z fiscal trade fdi
. * Fit the model with frontier command
. frontier lnY $x,uhet($z) nolog
                (...omitted outputs...)
. * Predict the inefficiency term and efficiency scores
. predict double uhat, u
. gen double te0 = exp(-u)
. * Store the estimated parameters
. mat b0=e(b)
.
. * Fit the model with xtsfsp command
. xtset _ID year
Panel variable: _ID (strongly balanced)
 Time variable: year, 1997 to 2017
         Delta: 1 unit
. mat b1 = b0,0.6,0.6,0.6
. xtsfsp lnY $x, uhet($z) wy(wm,mata) wu(wm,mata) wv(wm,mata) init(b1) te(tesp1) nolog
Spatial frontier model(yuv-SAR)                        Number of obs =      630
                                                       Wald chi2(9)  = 16704.73
Log likelihood = 328.51183                             Prob > chi2   =   0.0000
```

lnY	Coefficient	Std. err.	z	P>|z|	[95% conf. interval]	
frontier						
lnK	.6720836	.0259441	25.91	0.000	.6212341	.7229331
lnL	.3390601	.0185192	18.31	0.000	.302763	.3753571
_t	.0141493	.0051647	2.74	0.006	.0040267	.0242719
lnK_lnL	.0075504	.051805	0.15	0.884	-.0939855	.1090864
_t_lnK	.0425662	.0115137	3.70	0.000	.0199998	.0651326
_t_lnL	-.0034571	.0079979	-0.43	0.666	-.0191327	.0122186
_t_2	-.0069112	.0013048	-5.30	0.000	-.0094686	-.0043539
lnK_2	-.0875718	.0344022	-2.55	0.011	-.154999	-.0201446
lnL_2	-.0388746	.0377404	-1.03	0.303	-.1128443	.0350952
_cons	.4968023	.043107	11.52	0.000	.4123142	.5812903
/lnsigv2	-4.025468	.0581939	-69.17	0.000	-4.139526	-3.91141
uhet						
fiscal	.0098788	.0056888	1.74	0.082	-.001271	.0210287

```
       trade   -.0911438   .0213384    -4.27   0.000    -.1329663   -.0493212
         fdi   -.0185831   .0100212    -1.85   0.064    -.0382243    .0010582
       _cons   -1.963808   .3894655    -5.04   0.000    -2.727146   -1.200469
-------------+----------------------------------------------------------------
Wy           |
       _cons   -.0467557   .0362548    -1.29   0.197    -.1178138    .0243023
-------------+----------------------------------------------------------------
Wv           |
       _cons   -.0671769   .1600686    -0.42   0.675    -.3809057    .2465518
-------------+----------------------------------------------------------------
Wu           |
       _cons    1.280422   .2367245     5.41   0.000       .81645    1.744393
-------------+----------------------------------------------------------------
         rho   -.0233713   .0181157    -1.29   0.197     -.058833    .0121493
       gamma   -.0335725   .0799361    -0.42   0.674    -.1881642     .122643
         tau    .5649866   .0805643     7.01   0.000     .3869258    .7024182
------------------------------------------------------------------------------
Note: Wy:_cons, Wv:_cons and Wu:_cons are the transfromed parameters;
      rho, gamma and tau are their origin metrics in spatial components, repsectively.

. scalar loglikehood1 =  e(ll)

. mat b1 = b0,0.6

. xtsfsp lnY $x, uhet($z)  wu(wm,mata)  init(b1) te(tesp2)  nolog

Spatial frontier model:u-SAR                           Number of obs =      630
                                                       Wald chi2(9)  = 16174.91
Log likelihood = 327.60816                             Prob > chi2   =   0.0000

------------------------------------------------------------------------------
         lnY | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
frontier     |
         lnK     .672366   .0264662    25.40   0.000     .6204932    .7242388
         lnL    .3392661   .0184417    18.40   0.000     .3031209    .3754113
          _t    .0114801    .004803     2.39   0.017     .0020664    .0208937
     lnK_lnL    .0128618   .0467126     0.28   0.783    -.0786932    .1044169
      _t_lnK    .0422561   .0108054     3.91   0.000     .0210779    .0634344
      _t_lnL   -.0027543   .0074782    -0.37   0.713    -.0174113    .0119027
        _t_2   -.0066348   .0012629    -5.25   0.000      -.00911   -.0041595
       lnK_2   -.0915034   .0326073    -2.81   0.005    -.1554126   -.0275942
       lnL_2   -.0395255   .0328292    -1.20   0.229    -.1038696    .0248185
       _cons    .4644181   .0315665    14.71   0.000     .4025489    .5262874
-------------+----------------------------------------------------------------
    /lnsigv2   -4.013496   .0575493   -69.74   0.000     -4.12629   -3.900701
-------------+----------------------------------------------------------------
uhet         |
      fiscal    .0097757      .0055     1.78   0.076     -.001004    .0205555
       trade   -.0869182   .0204711    -4.25   0.000    -.1270408   -.0467956
         fdi   -.0133135   .0090702    -1.47   0.142    -.0310908    .0044638
       _cons   -1.986602   .3822161    -5.20   0.000    -2.735732   -1.237472
-------------+----------------------------------------------------------------
Wu           |
       _cons     1.07423   .1922999     5.59   0.000     .6973292    1.451131
-------------+----------------------------------------------------------------
         tau     .490752   .0729815     6.72   0.000     .3351572    .6202829
------------------------------------------------------------------------------

. scalar loglikehood2 =  e(ll)

. local lrtest = -2*(loglikehood2-loglikehood1)

. local pvalue = 1- chi2(2,`lrtest´)

. display "Likelihood-ratio test: LR chi2(2) = `lrtest´, Prob > chi2 = `pvalue´"
Likelihood-ratio test: LR chi2(2) = 1.807357699424301, Prob > chi2 = .4050766989379054
```

```
.
. * Plot the density of estimates of technical efficiency from different models
. twoway (kdensity te0, color(black) lpattern(solid)) ///
>        (kdensity tesp1,color(red) lpattern(dash))   ///
>            (kdensity tesp2,color(blue) lpattern(longdash)),  ///
>         legend(pos(10) ring(0) label(1 Non-spatial Stoc. Frontier) ///
>         label(2 Spatial Stoc. Frontier:yuv) label(3 Spatial Stoc. Frontier:u)) ///
>         xtitle("Technical Efficiency") ytitle("Density")
```

To construct the spatial weight matrix, we use the geographic data [5] and use the Stata command shshape2dta for conversion. We drop some units with heavily missing values (drop if _ID==26 | _ID>31). Based on the generated province.dta, we construct the spatial contiguity matrix with Stata Official spmatrix routines [6]. As _ID 26 is an iland so that we define its contiguity unit with the nearest one. As such, we extract the spatial weight matrix as wm in the Mata environment from the spmatrix object w_con and assign the elements wm[19,21] and wm[21,19] with one. Furthermore, we use the spmatrix routine to standardize the matrix. Now, the spmatrix object w_con can be used for our command xtsfsp. Alternatively, we can put the new w_con into the Mata environment as wm. After the preparation of the spatial weight matrix, we fit the models with the empirical data.

We begin with fitting a non-spatial stochastic frontier model with Stata official comamnd frontier. We predict the efficiency scores stored in the new variable te0 and extract the estimates of the parameters to serve as the initial values in spatial stochastic frontier models. Then we consider three source of spatial dependence. The estimated results show that the spatial correlation coefficients for the lag of dependent variable and the random errors are statistically insignificant at the 10% level. As such, we consider a restricted model with spatial dependence in the inefficient term. Since the restricted model is nested in the previous model, we also conduct the likelihood ratio test. The result documents that the null hypothesis the restricted model can not be rejected. The significant of τ indicates the positive spatial spillovers of technical efficiency. Figure 1 shows the distribution of efficiency scores estimated by different models.

5 Conclusion

Geospatial units are not isolated or separated but connected. For example, the economic trade, social activities, and cultural exchange between different regions affect each other. Such spatial interdependence challenges the traditional econometric methods, which generally assume cross-sectional independence. Spatial econometrics is developed to handle spatial correlation. This article presented a community-contributed command for fitting spatial stochastic frontier models with different sources of spatial dependence (Galli 2022; Orea and Álvarez 2019).

[5] They include province.shp and province.dbf.

[6] The community-contributed command spwmatrix by (Jeanty 2010) is also a powerful tool for creating spatial weight matrix in Stata

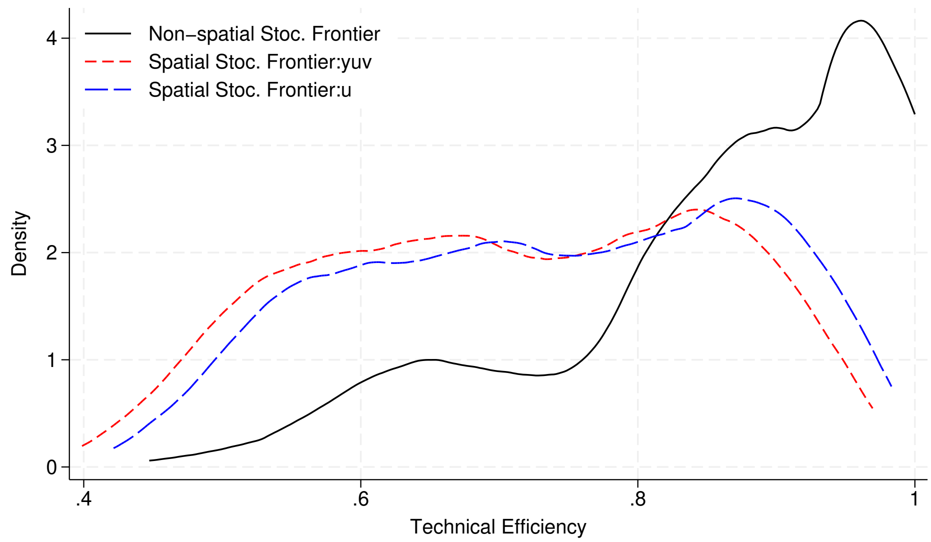

Figure 1: Distribution of efficiency scores

We hope the developed command can provide some convenience to practitioners and reduce the difficulty of model applications, thereby promoting sound empirical research.

Finally, there are some limitations that should be duly noted. First, despite the excellent flexibility of the spatial stochastic frontier models and our command introduced above, they fully parameterize the spatial structure, the frontier function and the distribution of random errors and inefficiency term in the spirit of stochastic frontier models and spatial econometrics. The model mispecifications are the potential costs. Some theoretical works devoted into relax the parametric assumptions are particularly valuable. Secondly, the models require prior information on the spatial weight matrices. The different choices of the spatial weight matrices might lead to various results. Internalizing the spatial weight matrices. in the spatial stochastic frontier models is still an open issue. Thirdly, the numerical computation for the MLE of the spatial stochastic frontier model is very complicated. When the spatial weight matrix is large and the time-varying spatial weight matrix is taken into account, the estimation is very time consuming because the matrix has to be inverted repeatedly.

6 Acknowledgments

Kerui Du thanks the financial support of the National Natural Science Foundation of China (72074184). We are grateful to Federica Galli for his Matlab codes, Federico Belotti, Silvio Daidone, Giuseppe Ilardi and Vincenzo Atella for the sfcross/sfpanel package, Mustafa U. Karakaplan for the sfkk package, and

Jan Ditzen, William Grieser and Morad Zekhnini for the nwxtregress package which inspired our design of the `xtsfsp` command.

7 References

Aigner, D., C. Lovell, and P. Schmidt. 1977. Formulation and Estimation of Stochastic Frontier Production Function Models. *Journal of Econometrics* 6(1): 21–37.

Battese, G. E., and T. J. Coelli. 1995. A Model for Technical Inefficiency Effects in a Stochastic Frontier Production Function for Panel Data. *Empirical Economics* 20(2): 325–332.

Belotti, F., S. Daidone, G. Ilardi, and V. Atella. 2013. Stochastic Frontier Analysis Using Stata. *The Stata Journal: Promoting communications on statistics and Stata* 13(4): 719–758.

Belotti, F., and G. Ilardi. 2018. Consistent Inference in Fixed-Effects Stochastic Frontier Models. *Journal of Econometrics* 202(2): 161–177.

Chen, Y.-Y., P. Schmidt, and H.-J. Wang. 2014. Consistent Estimation of the Fixed Effects Stochastic Frontier Model. *Journal of Econometrics* 181(2): 65–76.

Druska, V., and W. C. Horrace. 2004. Generalized moments estimation for spatial panel data: Indonesian rice farming. *American Journal of Agricultural Economics* 185–198.

Galli, F. 2022. A Spatial Stochastic Frontier Model Including Both Frontier and Error-Based Spatial Cross-Sectional Dependence. *Spatial Economic Analysis* 1–20.

Glass, A. J., K. Kenjegalieva, and R. C. Sickles. 2016. A Spatial Autoregressive Stochastic Frontier Model for Panel Data with Asymmetric Efficiency Spillovers. *Journal of Econometrics* 190(2): 289–300.

Greene, W. 2005. Fixed and Random Effects in Stochastic Frontier Models. *Journal of Productivity Analysis* 23(1): 7–32.

Gude, A., I. Álvarez, and L. Orea. 2018. Heterogeneous Spillovers among Spanish Provinces: A Generalized Spatial Stochastic Frontier Model. *Journal of Productivity Analysis* 50(3): 155–173.

Jann, B. 2005. MOREMATA: Stata module (Mata) to provide various functions. Statistical Software Components, Boston College Department of Economics. URL https://ideas.repec.org/c/boc/bocode/s455001.html.

Jeanty, P. W. 2010. SPWMATRIX: Stata module to generate, import, and export spatial weights. Statistical Software Components, Boston College Department of Economics. URL https://ideas.repec.org/c/boc/bocode/s457111.html.

Karakaplan, M. 2018. XTSFKK: Stata Module to Estimate Endogenous Panel Stochastic Frontier Models in the Style of Karakaplan and Kutlu. *Boston College Department of Economics* Statistical Software Components S458445.

Karakaplan, M., and L. Kutlu. 2017. Handling Endogeneity in Stochastic Frontier Analysis. *Economics Bulletin* 37(2): 889–901.

Karakaplan, M. U. 2017. Fitting Endogenous Stochastic Frontier Models in Stata. *The Stata Journal: Promoting communications on statistics and Stata* 17(1): 39–55.

Kumbhakar, S. C., H. Wang, and A. P. Horncastle. 2015. *A practitioner's guide to stochastic frontier analysis using Stata.* Cambridge University Press.

Kutlu, L., K. C. Tran, and M. G. Tsionas. 2020. A Spatial Stochastic Frontier Model with Endogenous Frontier and Environmental Variables. *European Journal of Operational Research* 286(1): 389–399.

Liu, W., and R. C. Sickles. 2023. Industry-Specific Productivity and Spillovers Through Input–Output Linkages: Evidence from Asia-Pacific Value Chain. *Review of Income and Wealth* .

Meeusen, W., and J. van Den Broeck. 1977. Efficiency Estimation from Cobb-Douglas Production Functions with Composed Error. *International Economic Review* 18(2): 435.

Orea, L., and I. C. Álvarez. 2019. A New Stochastic Frontier Model with Cross-Sectional Effects in Both Noise and Inefficiency Terms. *Journal of Econometrics* 213(2): 556–577.

Orea, L., I. C. Álvarez, and T. Jamasb. 2018. A spatial stochastic frontier model with omitted variables: Electricity distribution in Norway. *The Energy Journal* 39(3).

Orea, L., I. C. Álvarez, and L. Servén. 2023. The structural and productivity effects of infrastructure provision in developed and developing countries. *in Parmeter, C. and Wang, H.J. (Eds.). Essays in Honor of Subal Kumbhakar (Advances in Econometrics), Emerald Publishing Limited* .

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(3): 406–419.

Wang, H.-J., and C.-W. Ho. 2010. Estimating Fixed-Effect Panel Stochastic Frontier Models by Model Transformation. *Journal of Econometrics* 157(2): 286–296.

About the authors

Kerui Du is an associate professor at the School of Management, Xiamen University. His primary research interests include applied econometrics, energy and environmental economics.

Luis Orea is a full professor at the School of Economics and Business, University of Oviedo. His primary research interests include Efficiency and productivity analysis, econometric modelling, agricultural economics, energy economics, regulation and competition, spatial economics.

Inmaculada C. Álvarez is a full professor at the Department of Economics, Universidad Autónoma de Madrid. Her primary research interest include infrastructures, efficiency and productivity, economic growth and development, spatial economics and quantitative methods.