Spatial Stochastic frontier models: Instructions for use

Elisa Fusco & Francesco Vidoli

August 28, 2023

In the last decade stochastic frontiers traditional models (see Kumbhakar and Lovell, 2000 for a detailed introduction to frontier analysis) have been extended with the aim to take into account firm specific heterogeneity (see *e.g.* Greene, 2004, Greene, 2005b, Greene, 2005a). If firm specific heterogeneity is not accounted, in fact, a considerable bias in the inefficiency estimates can be endogenously created.

`ssfa` package allows to include heterogeneity in a different way with respect to traditional techniques: *"instead of identifying ex-ante a multitude of determinants, often statistically and economically difficult to detect [...] this approach allow the evaluation of the conjoint effect of a multitude of determinants"* (Fusco and Vidoli, 2013) considering spatial proximities; more particularly `ssfa` package implements the Spatial Stochastic Frontier Analysis (SSFA), an original method introduced by Fusco and Vidoli (2013) with the aim to test and depurate the spatial heterogeneity in Stochastic Frontier Analysis (SFA) models by splitting the inefficiency term into three terms: the first one related to spatial peculiarities of the territory in which each single unit operates, the second one related to the specific production features and the third one representing the error term.

The main idea is that spatial dependence refers to how much the level of technical inefficiency of farm i depends on the levels set by other farms $j = 1, ..., n$, under the assumption that part of the farm i inefficiency (u_i) is linked to the neighbour DMU j's performances ($j \neq i$).
Denoting y_i as the single output of producer i, x_i the inputs vector and f a generic parametric function, the Normal / Half-Normal cross-sectional production frontier model can be respectively written[1]:

$$\begin{aligned} log(y_i) &= log(f(x_i;\beta_i)) + v_i - u_i \\ &= log(f(x_i;\beta_i)) + v_i - (1 - \rho \sum_i w_{i.})^{-1} \widetilde{u}_i \end{aligned}$$

where

$$\begin{aligned} & v_i \sim \mathcal{N}(0, \sigma_v^2) \\ & u_i \sim \mathcal{N}^+(0, (1 - \rho \sum_i w_{i.})^{-2} \sigma_{\widetilde{u}}^2) \\ & u_i \text{ and } v_i \text{ are independently distributed of each other,} \\ & \quad \text{and of the regressors} \\ & \widetilde{u_i} \sim \mathcal{N}(0, \sigma_{\widetilde{u}}^2) \\ & w_{i.} \text{ is a standardized row of the spatial weights matrix} \\ & \rho \text{ is the spatial lag parameter } (\rho \in [0, 1]) \end{aligned} \tag{1}$$

[1] For simplicity's sake and to make the notation more consistent with the SFA literature, we did not write the model in matrix form, but for each company i.

`ssfa` package allows to estimate both the "*production*" form (as shown in equation (1) and the "*cost*" form of the frontier *i.e.*:

$$log(C_i) = log(f(y_i, w_i; \beta_i)) + v_i + u_i$$

where

C_i is the cost
w_i are the input prices. (2)

Introducing a variable sc that defines the form of the frontier:

$$\begin{cases} 1 \text{ for production function} \\ -1 \text{ for cost function} \end{cases} \quad (3)$$

`ssfa` model can be written as:

$$log(y_i) = log(f(x_i; \beta_i)) + v_i - sc \cdot u_i \quad (4)$$

In order to estimate the `ssfa` model we have to install and load the package:

```
> #install.packages("ssfa")
> library(ssfa)
```

In this package, the `SSFA_example_data` and `Italian_W` datasets have been included in order to better illustrate and comment the model.

- The first dataset contains the simulated data used by Fusco and Vidoli (2013) to test the model. Data Generating Process (DGP) follows the construction criteria proposed by Banker and Natarajan (2008), also used by Johnson and Kuosmanen (2011), with the addition of a strong spatial correlation ($\rho = 0.80$) in the inefficiency term through a spatial lag parameter and the contiguity matrix `Italian_W`.
- The second dataset is the Italian provinces contiguity matrix for the year 2008 containing 107 x 107 row-standardized distances.

```
> data(SSFA_example_data)
> data(Italian_W)
> names(SSFA_example_data)
```

```
[1] "DMU"   "log_y" "log_x"
```

The variable `log_y` is the log-transformed output, `log_x` is the log-transformed input and `DMU` is the Decision Making Unit name.

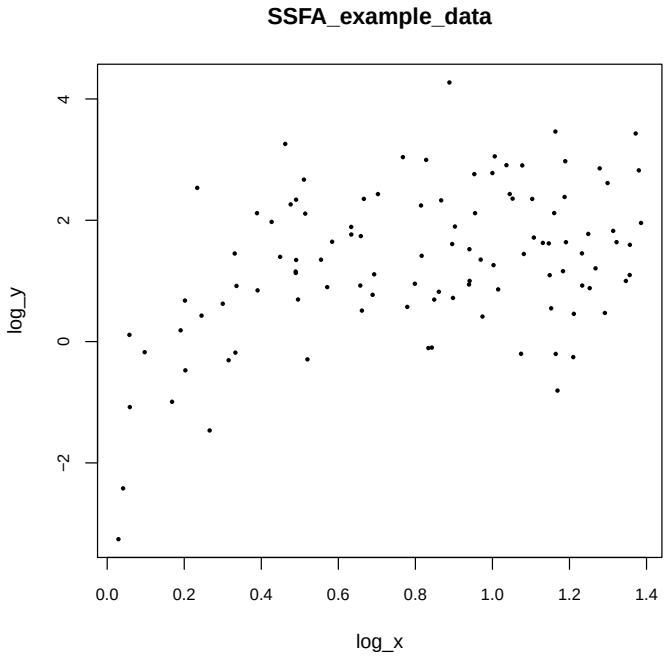


Figure 1: Example simulated data

`ssfa` package allows to easily compare the Spatial Stochastic Frontier (SSFA) with the classical Stochastic Frontier (SFA) by setting the parameter `par_rho` as `TRUE` to estimate the SSFA or `FALSE` to estimate the classical SFA.

In order to compare the SSFA estimation versus the SFA one, a standard SFA production frontier has been first estimate by setting, into the `ssfa` function, command `form="production"` and `par_rho="FALSE"`:

```
> sfa <- ssfa(log_y ~ log_x , data = SSFA_example_data, data_w=Italian_W,
+             form = "production", par_rho=FALSE)
> summary(sfa)

Stochastic frontier analysis model

          Estimate Std. Error z value Pr(>|z|)
Intercept 1.185847   0.441450 2.68626 0.007226 **
log_x     1.273394   0.301340 4.22577  2.4e-05 ***
sigmau2   1.319261   0.892472 1.47821 0.139352
sigmav2   0.779320   0.307384 2.53533 0.011234 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

sigma2 = 2.098581

Inefficiency parameter Lambda (sigmau/sigmav): 1.30109

Moran I statistic: 0.457094
```

```
Mean efficiency: 0.485295

LR-test: sigmau2 = 0 (inefficiency has no influence to the model)
H0: sigmau2 = 0 (beta_ssfa = beta_ols)

     Value Log-Lik
ssfa     -163.6215
ols      -164.1653

Value LR-Test: 1.088 p-value 0.148

AIC: 335.2431, (AIC for lm: 332.3306)
```

In the standard SFA framework (`par_rho="FALSE"`), `ssfa` function returns, in addition to the *intercept* and the `log_x` coefficient, the estimation of the variance of the two error components `sigmau2` and `sigmav2`. Other useful information about efficiency estimation are reported:

- `sigma2`: the estimate of the *total variance* where $\sigma^2 = \sigma_u^2 + \sigma_v^2$;
- `lambda`: the ratio of the standard deviation of the inefficiency term to the standard deviation of the stochastic term *i.e.* $\frac{\sigma_u}{\sigma_v}$;
- the mean of efficiency estimated;
- the results of the test on the influence of the inefficiency on the model. This is a test of the null hypothesis $H_0 : \sigma_u^2 = 0$ against the alternative hypotheses $H_1 : \sigma_u^2 > 0$. If the null hypothesis is true, the stochastic frontier model is reduced to an OLS model with normal errors. For this example, the output shows $LR = 1.088$ with a p-value of 0.148. There are several possible reasons for the failure to this test, including for example the uncontrolled spatial dependence of the inefficiency term.

In addition to previous statistics, `summary` function displays information about the spatial autocorrelation of the SFA residuals, the Moran's I statistic. For example, in this application $I = 0.457$ showing a positive and significant ($p - value < 2.2e - 16$) global autocorrelation among residuals.

```
>  moran.test(residuals(sfa), listw=sfa$list_w)


        Moran I test under randomisation

data:  residuals(sfa)
weights: sfa$list_w

Moran I statistic standard deviate = 8.3329, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.457093892      -0.009433962       0.003134475
```

Autocorrelation among residuals can be tested also locally thanks to `plot_moran` function that enables you to assess how similar an observed value is to its neighbouring observations; its horizontal axis is based on the values of the observations and is also known as the response axis, while the vertical Y axis is based on the weighted average or spatial lag of the corresponding observation on the horizontal X axis. This function need a neighbours list: it can be easily calculate thanks to the `nb2listw` function of `spdep` package from the contiguity matrix `Italian_W`.

```
>  plot_moran(sfa, listw=sfa$list_w)
```

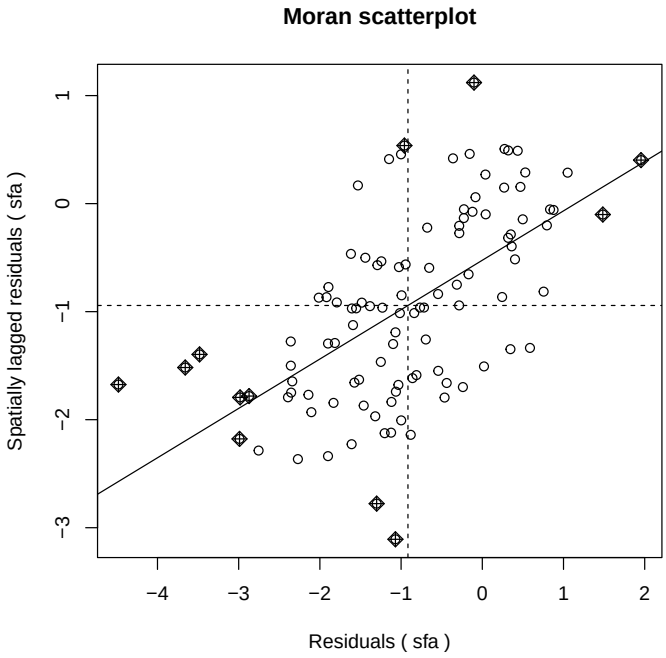


Figure 2: SFA Moran scatterplot

Finally, `summary` function reports the AIC value for the `ssfa` model and the `lm` model.

Having estimated the SFA model as baseline, the spatial production frontier SSFA can be carried on by setting command `form="production"` and `par_rho="TRUE"`:

```
> ssfa <- ssfa(log_y ~ log_x , data = SSFA_example_data, data_w=Italian_W,
+              form = "production", par_rho=TRUE)
> summary(ssfa)


Spatial Stochastic frontier analysis model

            Estimate Std. Error z value Pr(>|z|)
Intercept   3.445040   1.855917 1.85625 0.063418 .
log_x       1.633247   0.226941 7.19679  < 2e-16 ***
sigmau2_dmu 0.596074   0.604825 0.98553 0.324363
sigmav2     0.474248   0.203866 2.32627 0.020004 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Pay attention:
1 - classical SFA sigmau2 = sigmau2_dmu + sigmau2_sar: 0.882803 where sigmau2_sar: 0.286729
2 - sigma2 = sigmau2_dmu + sigmau2_sar + sigmav2: 1.357051

Inefficiency parameter Lambda = sigmau_dmu/sigmav: 1.256882
```

```
Spatial parameter Rho: 0.778393

Moran I statistic: -0.189043

Mean efficiency: 0.571884

LR-test: sigmau2_dmu = 0 (inefficiency has no influence to the model)
H0: sigmau2_dmu = 0 (beta_ssfa = beta_ols)

     Value Log-Lik
ssfa     -138.9479
ols      -164.1653

Value LR-Test: 50.435 p-value 0

AIC: 297.8958, (AIC for lm: 332.3306)
```

The output of `ssfa` (with `par_rho="FALSE"`) returns the *intercept*, the `log_x` coefficient and the estimation of the variance of the two error components not spatially correlated *i.e.* `sigmau2_dmu` and `sigmav2`.

In this case, the model decomposes the inefficiency variance `sigmau2` into `sigmau2_dmu` and `sigmau2_sar`, respectively the part of inefficiency variance due to DMU's specificities and to the spatial dependence, *i.e.* $\sigma_u^2 = \sigma_{u_{dmu}}^2 + \sigma_{u_{sar}}^2$. Consequently, the *total variance* is given by $\sigma^2 = \sigma_{u_{dmu}}^2 + \sigma_{u_{sar}}^2 + \sigma_v^2$.

In this application, (`lambda` = 1.257) is smaller than the SFA one (`lambda` = 1.301) because the production unit inefficiency is sterilized from the influence of the neighbourhood performances.

In addition, the `summary` function reports the estimated spatial parameter ρ that in this case is 0.778 very close to the true simulation parameter (0.80); Moran's $I = -0.189$ is no more significant ($p-value = 0.9993$).

```
> moran.test(residuals(ssfa), listw=ssfa$list_w)


        Moran I test under randomisation

data:  residuals(ssfa)
weights: ssfa$list_w

Moran I statistic standard deviate = -3.2046, p-value = 0.9993
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     -0.189042898      -0.009433962       0.003141349


> plot_moran(ssfa, listw=sfa$list_w)
```

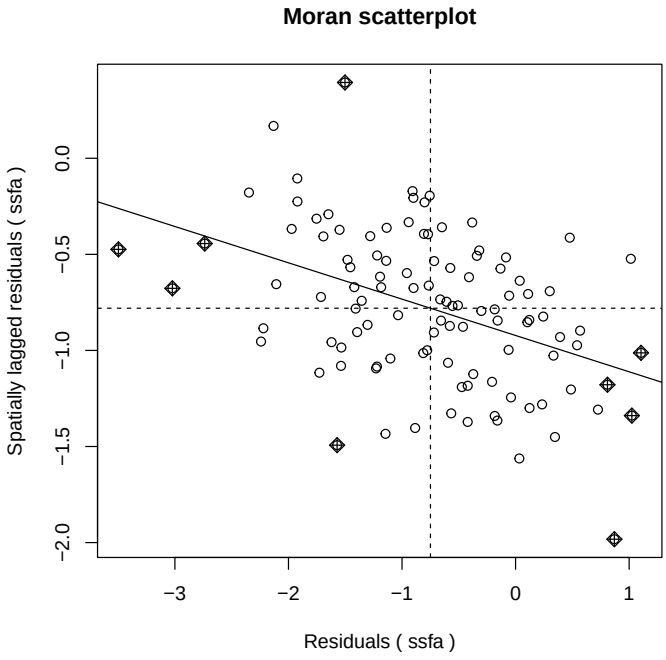


Figure 3: SSFA Moran scatterplot

In this application it can be easily note that the likelihood-ratio test is highly significant ($LR = 50.435$ with a p-value $= 0.000$); these findings, support the conclusion that the SSFA model is able to correctly estimate the inefficiency component of the error term.

Other functions are available into `ssfa` package:

- `fitted.ssfa`: this function calculates the fitted values of the original data used to estimate the SSFA model.

```
> ssfa_fitted <- fitted.ssfa(ssfa)
> sfa_fitted <- fitted.ssfa(sfa)
```

- `plot_fitted`: plots the original data, the SSFA fitted frontier and optionally the SFA fitted frontier with the aim to compare models colouring points according to the efficiency values.

```
> plot_fitted(SSFA_example_data$log_x, SSFA_example_data$log_y, ssfa, pch=16, cex=0.5,
+               xlab="X", ylab="Y", cex.axis=0.8 )
> points(SSFA_example_data$log_x, SSFA_example_data$log_y, pch=16, cex=0.5,
+   col= ifelse(eff.ssfa(ssfa)<=quantile(eff.ssfa(ssfa), 0.20) , "#D7191C",
+         ifelse(eff.ssfa(ssfa)>quantile(eff.ssfa(ssfa), 0.20)
+           &eff.ssfa(ssfa)<=quantile(eff.ssfa(ssfa), 0.4) ,"#FF8C00",
+         ifelse(eff.ssfa(ssfa)>quantile(eff.ssfa(ssfa), 0.4)
+           &eff.ssfa(ssfa)<=quantile(eff.ssfa(ssfa), 0.6) ,"#FFFF00",
+         ifelse(eff.ssfa(ssfa)>quantile(eff.ssfa(ssfa), 0.6)
+           &eff.ssfa(ssfa)<quantile(eff.ssfa(ssfa), 0.8) ,"#ADFF2F",
+         ifelse(eff.ssfa(ssfa)>quantile(eff.ssfa(ssfa), 0.8)
+           &eff.ssfa(ssfa)<=quantile(eff.ssfa(ssfa), 1),"#008B00", "#2F4F4F"))))))
> lines(sort(SSFA_example_data$log_x),sfa_fitted[order(SSFA_example_data$log_x)],
+             col="red")
```

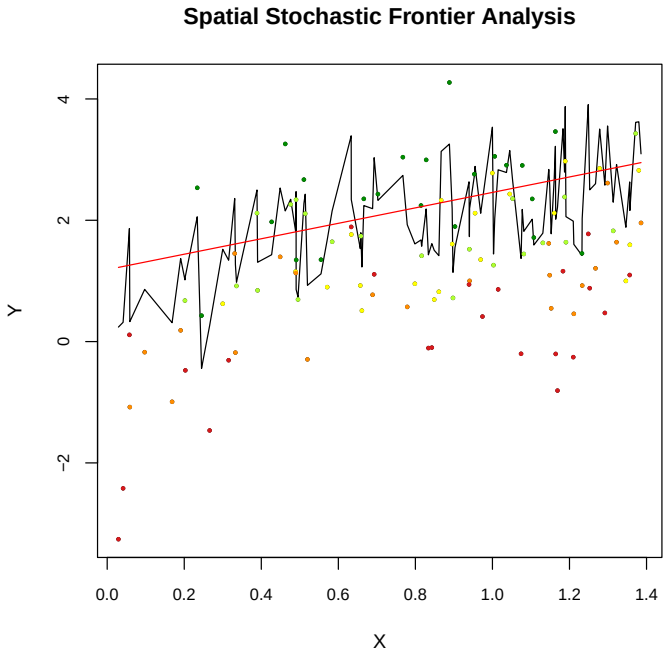
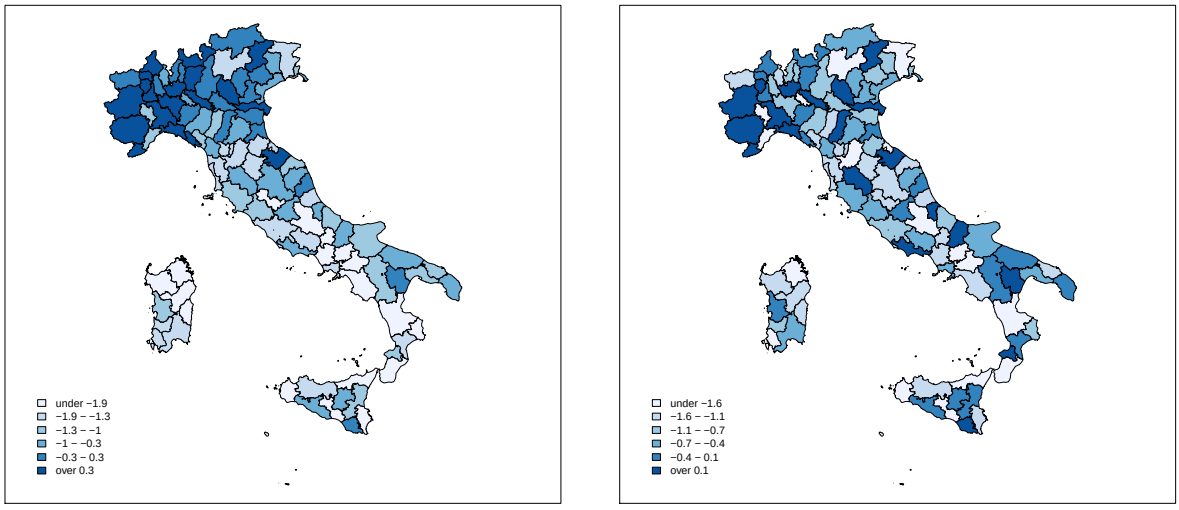


Figure 4: Plot data, SSFA and SSFA frontiers

- `residuals.ssfa`: calculates the SSFA model residuals.

```
> ssfa_residuals <- residuals.ssfa(ssfa)
> sfa_residuals <- residuals.ssfa(sfa)
```

With residuals estimation we can compare SFA and SSFA results, for example, with maps like the following:



(a) SFA

(b) SSFA

Figure 5: Spatial residuals distribution by method

Figure 5 shows that the spatial dependence present in SFA residuals (a) is fully neutralized by the SSFA model (b).

- `eff.ssfa`: calculates the efficiency (Battese and Coelli (1988) formulation) and inefficiency (Jondrow et al. (1982) formulation) estimated.

```
> ssfa_eff <-  eff.ssfa(ssfa)
> #sfa_eff <-  eff.ssfa(sfa)
>
> #summary(sfa_eff)
> #summary(ssfa_eff)
>
> ssfa_u <- u.ssfa(ssfa)
> #sfa_u <- u.ssfa(sfa)
>
> #summary(ssfa_u)
> #summary(sfa_u)
```

References

Banker, R. and Natarajan, R. (2008). Evaluating contextual variables affecting productivity using data envelopment analysis. *Operations research*, 56(1):48–58.

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(3):387–399.

Fusco, E. and Vidoli, F. (2013). Spatial stochastic frontier models: controlling spatial global and local heterogeneity. *International Review of Applied Economics*, 27(5):679–694.

Greene, W. (2004). Distinguishing between heterogeneity and inefficiency: stochastic frontier analysis of the world health organization's panel data on national health care systems. *Health Economics*, 30:959–980.

Greene, W. (2005a). Fixed and random effects in stochastic frontier models. *Journal of Productivity Analysis*, 23:7–32.

Greene, W. (2005b). Reconsidering heterogeneity in panel data estimators of the stochastic frontier model. *Journal of Econometrics*, 126(2):269—-303.

Johnson, A. and Kuosmanen, T. (2011). One-stage estimation of the effects of operational conditions and practices on productive performance: asymptotically normal and efficient, root-n consistent stonezd method. *Journal of Productivity Analysis*, 36:219–230.

Jondrow, J., Knox Lovell, C. A., Materov, I. S., and Schmidt, P. (1982). On the estimation of technical inefficiency in the stochastic frontier production function model. *Journal of Econometrics*, 19(2-3):233–238.

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