Frontier Analysis with R

Summer School on Mathematical Methods in Finance and Economy

Thibault LAURENT

Toulouse School of Economics

June 2010 (slides modified in August 2010)

Introduction

A first simulated example

Analysis of the real data

The packages about frontier analysis

- **FEAR**: Frontier Efficiency Analysis with R
- Available at http://www.clemson.edu/economics/faculty/wilson/Software/FEAR/fear.html
- install the package from local zip file
- Other packages: **DEA** (Data Envelopment Analysis) (no more available in August 2010), **frontier** (Stochastic Frontier Analysis)
- Soon on CRAN: package **frontiles**, exploratory frontier analysis and measures of efficiency.

Introduction

A first simulated example

Analysis of the real data

Simulation with R

1. Factor variable

Random generation of a vector of size 100 following a binomial distribution with $p = 0.2$:

```
> x <- rbinom(100, 1, 0.2)
> plot(table(x), main = "frequency")
```

Other distributions: Poisson $\mathcal{P}(\lambda)$ (function `rpois`), etc.

2. Numeric variable

Random generation of a vector of size 100 following a gaussian distribution $\mathcal{N}(\mu = 1, \sigma = 1)$:

```
> x <- rnorm(100, 1, 1)
> hist(x, main = "")
```

Other distributions: Uniform $\mathcal{U}_{[a,b]}$ (function `runif`), etc.

Simulate the data (1)

See Simar-Zelenyuk (*Journal of Applied Econometrics*, 2007)

- one output y and one input x both of size $n = 15$
- The true frontier is defined by the function $f : x \to \sqrt{x}$
- For simulating the data:
  1. define the vector of input as $x \sim \mathcal{U}_{[0,1]}$
  2. define a vector $u \sim \mathcal{N}^+(\mu = 0.25, \sigma = 0.2)$
  3. the vector of input is defined as $y = \frac{\sqrt{x}}{1+u}$

Simulate the data (2)

The function `set.seed` allows us to keep the same simulated data

```
> require(tmvtnorm)
> ns = 15
> set.seed(121181)
> x = runif(ns, 0, 1)
> ybar = x^(1/2)
> set.seed(121181)
> u = rtmvnorm(n = ns, mean = c(0.25), sigma = c(0.2),
+     lower = c(0))
> y = ybar/(1 + u)
```

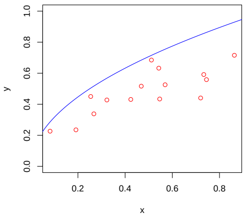
Representation of the data

Representation of the simulated data:

```
> plot(y ~ x, type = "p",
+      col = "red", ylim = c(0,
+          1))
```

Representation of the true frontier:

```
> x.seq <- seq(0, 1, by = 0.01)
> t.fr <- x.seq^(1/2)
> lines(t.fr ~ x.seq, col = "blue")
```



"True frontier" efficiency measurement

- Output oriented measure:

$$\lambda(x, y) = \frac{y}{f(x)}$$

- Input oriented measure:

$$\theta(x, y) = \frac{f^{-1}(y)}{x}$$

- Shepard measure:

$$\delta(x, y) = \frac{1}{\theta(x,y)}$$

```
> lambda = y/sqrt(x)
> theta = y^2/x
> delta = 1/theta
```

Reproducible research

```
> require(xtable)
> tab1 <- data.frame(lambda, theta, delta)
> matable <- xtable(tab1[1:5, ], digits = 3, align = "l|ccc",
+     caption = "True Frontier Efficiency measures")
> print(matable, hline.after = c(0), file = "V.tex",
+     size = "tiny")
```

	lambda	theta	delta
1	0.648	0.419	2.385
2	0.792	0.627	1.595
3	0.958	0.917	1.090
4	0.770	0.594	1.685
5	0.753	0.567	1.765

Table: True Frontier Efficiency measures

Stochastic frontier (1)

1. adjust a linear model with function `lm` and keep the coefficient β of the regression line: $y = \alpha + \beta x$
2. find the firm k which maximises $(y_i - \hat{y}_i)$, $i = 1, \ldots, n$. Notice that the firm k can be found and detected manually with function `identify`
3. calculate α' such that the regression line $y = \alpha' + \beta x$ goes through firm k and represent the stochastic frontier

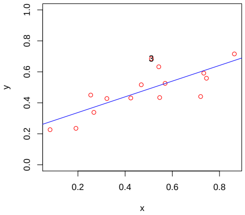
Stochastic frontier (2)

1. OLS model

```
> res.lm <- lm(y ~ x)
> beta.lm <- coefficients(res.lm)
```

2. Use of the function `identify`

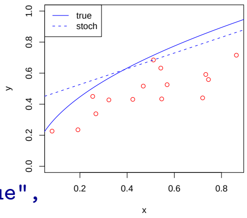
```
> plot(x, y, col = "red")
> abline(beta.lm, col = "blue")
> identify(x, y)
```



Stochastic frontier (3)

3. Find α' and representation

```
> alpha2 <- y[3] - beta.lm[2] *
+     x[3]
> plot(y ~ x, type = "p",
+     col = "red", ylim = c(0,
+         1))
> lines(t.fr ~ x.seq, col = "blue")
> abline(alpha2, beta.lm[2],
+     col = "blue", lty = 2)
> legend("topleft", legends = c("true",
+     "stoch"), lty = 1:2,
+     col = "blue")
```



Stochastic frontier efficiency measurement

Let us define $f_1 : x \to \alpha' + \beta x$

```
> f1 = function(x) alpha2 + beta.lm[2] * x
```

$f_1^{-1} : x \to \frac{x-\alpha'}{\beta}$

```
> f1.inv = function(x) (x - alpha2)/beta.lm[2]
```

- Output oriented measure:

$$\lambda(x, y) = \frac{y}{f_1(x)}$$

- Input oriented measure:

$$\theta(x, y) = \frac{f_1^{-1}(y)}{x}$$

- Shepard measure:

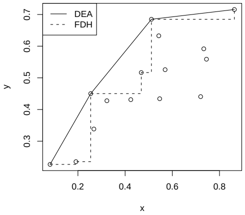
$$\delta(x, y) = \frac{1}{\theta(x,y)}$$

```
> lambda1 = y/f1(x)
> theta1 = f1.inv(y)/x
> delta1 = 1/theta1
```

DEA - FDH representation

Manual detection of the firms located on the two frontiers with the `identify()` function

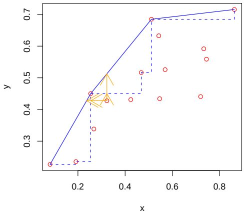
```
> plot(y ~ x)
> identify(x, y)
> lines(x[c(2, 9, 3, 4)],
+     y[c(2, 9, 3, 4)])
> lines(x[c(2, 12, 12,
+     9, 9, 8, 8, 3, 3,
+     4, 4)], y[c(2, 2,
+     12, 12, 9, 9, 8,
+     8, 3, 3, 4)], lty = 2)
> legend("topleft", legend = c("DEA",
+     "FDH"), lty = 1:2)
```



DEA - FDH efficiency frontiers/measures

Let consider firm number 5

1. On which part of the frontier would this firm be located if it were efficient
   in the ouput direction ?
   in the input direction ?
2. Using this position on the estimated frontiers, calculate the measures of efficiency

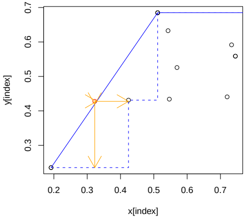


Naive Bootstrap

Repeat B times (with the loop `for`)

1. sampling among the 15 observations with function `sample`
2. calculate new estimators of the frontiers
3. calculate new measures of efficiency
4. stock the results

Calculate Biais, Variance, Confidence interval



Introduction

A first simulated example

Analysis of the real data

Exploratory Data Analysis

The data sets

- one output and three input observed on 62 farms in Spain

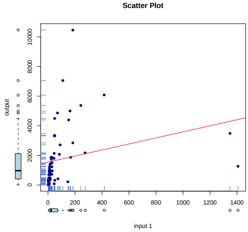
```
> spain <- read.table("spain.txt", header = TRUE)
> summary(spain)
```

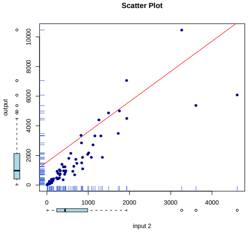
- Fore more details, see the section 5.2. of Aragon-Daouia-Thomas (*Annales d'économie et de statistique*, 2006).

Scatter plot (1)

```
> op <- par()
> layout(matrix(c(2, 1, 0, 3), 2, 2, byrow = T),
+     c(1, 6), c(4, 1))
> par(mar = c(1, 1, 5, 2))
> plot(y ~ x1, data = spain, pch = 16, col = "darkblue")
> abline(lm(y ~ x1, data = spain), col = "red")
> title(main = "Scatter Plot")
> rug(spain$x1, side = 1, col = "royalblue")
> rug(spain$y, side = 2, col = "royalblue")
> par(mar = c(1, 2, 5, 1))
> boxplot(spain$y, axes = F, col = "lightblue")
> title(ylab = "output", line = 0)
> par(mar = c(5, 1, 1, 2))
> boxplot(spain$x1, horizontal = T, axes = F, col = "lightblue")
> title(xlab = "input", line = 1)
> par(op)
```

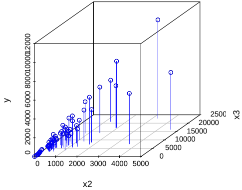
Scatter plot (2)





Scatter plot 3-d

```
> require(scatterplot3d)
> with(spain, scatterplot3d(x1,
+      x2, y))
```



Structure of the data in **FEAR**

- the p inputs are included in a $p \times n$ `matrix`

```
> input <- t(cbind(spain$x1, spain$x2, spain$x3))
```

- the q outputs are included in a $q \times n$ `matrix`

```
> output <- t(matrix(spain$y))
```

Measures of efficiency

- function `dea` computes DEA Efficiency estimates
- function `fdh` computes FDH efficiency estimates
- function `orderm` computes m-order efficiency estimates ($m = 25$ by default)
- function `hquan` computes non parametric conditional and unconditional α-quantile estimates ($\alpha = 0.95$ by default)

NB: argument `ORIENTATION` indicates the direction in which efficiency is to be evaluated (equal to 1 for input direction, 2 for output direction, 3 for hyperbolic)

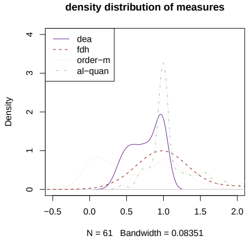
Measures of efficiency (2)

```
> require(FEAR)

FEAR (Frontier Efficiency Analysis with R) 1.13 installed
Copyright Paul W. Wilson 2010
See file LICENSE for license and citation information

> res.dea <- dea(input, output, ORIENTATION = 2)
> res.fdh <- fdh(input, output, ORIENTATION = 2)
> res.orderm <- orderm(input, output, ORIENTATION = 2)
> res.hquan <- hquan(input, output, ORIENTATION = 2)
> res.measures <- rbind(res.dea, res.fdh[1, ], res.orderm[1, ],
+     res.hquan)
> row.names(res.measures) <- c("dea", "fdh", "orderm", "al-quan")
```

You can use the functions `order` or `sort` to compute the ranks of the firms depending on the efficiency measure.

Comparison of the measures of efficiency

```
> plot(density(res.dea),
+     xlim = c(-0.5, 2),
+     ylim = c(0, 4), col = colors()[99],
+     lty = 1, main = "density distribution of measures")
> lines(density(res.fdh),
+     col = colors()[100],
+     lty = 2)
> lines(density(res.orderm),
+     col = colors()[101],
+     lty = 3)
> lines(density(res.hquan),
+     col = colors()[102],
+     lty = 4)
> legend("topleft", legend = c("dea",
+     "fdh", "order-m",
+     "al-quan"), lty = 1:4,
+     col = colors()[99:102])
```



Bootstrap

Function `boot.sw98` implements the bootstrap method of Simar and Wilson (1998) for estimating confidence intervals for Shepard (1970) input and output distance functions.
NB: may take time

```
> boot.sw98(input, output)
```