

Package ‘robustSFA’

July 23, 2025

Type Package

Title Robust Estimation of Stochastic Frontier Models with MDPDE

Version 0.2.0

Maintainer Junmo Song <junmo.song@gmail.com>

Description This provides a robust estimator for stochastic frontier models, employing the Minimum Density Power Divergence Estimator (MDPDE) for enhanced robustness against outliers. Additionally, it includes a function to recommend the optimal tuning parameter, alpha, which controls the robustness of the MDPDE. The methods implemented in this package are based on Song et al. (2017) <[doi:10.1016/j.csda.2016.08.005](https://doi.org/10.1016/j.csda.2016.08.005)>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Imports stats, truncnorm, Rcpp, frontier

NeedsCompilation yes

LinkingTo Rcpp, BH

Author Junmo Song [aut, cre],
Dong-hyun Oh [aut]

Repository CRAN

Date/Publication 2025-03-21 15:50:02 UTC

Contents

| | |
|--------------------------|---|
| bootstrap_test | 2 |
| coef.rsfa | 3 |
| efficiency | 4 |
| optimal_alpha | 5 |
| print.rsfa | 6 |
| rsfa | 7 |
| summary.rsfa | 9 |

| | |
|--------------|-----------|
| Index | 11 |
|--------------|-----------|

Description

This function performs a bootstrap test to assess the closeness of two MDPD estimates at different values of α . This test is based on the observation that outliers particularly affect the estimation of σ^2 and λ . Accordingly, the bootstrap test is conducted using the following similarity measure:

$$\text{sim}(\hat{\theta}_0, \hat{\theta}_1) = \frac{|\hat{\sigma}_1^2 - \hat{\sigma}_0^2|}{\hat{\sigma}_0^2} + \frac{|\hat{\lambda}_1 - \hat{\lambda}_0|}{\hat{\lambda}_0},$$

where $\hat{\theta}_0$ and $\hat{\theta}_1$ are the MDPD estimates corresponding to $\alpha = \alpha_0$ and $\alpha = \alpha_1$, respectively. If the two MDPD estimates are close, $\text{sim}(\hat{\theta}_0, \hat{\theta}_1)$ will be close to zero. We note that this similarity measure differs from the one used in Song et al. (2017). Apart from this, the bootstrap procedure follows the same steps described in Song et al. (2017). A low p -value indicates that the two estimates are significantly different. Note that this test may require significant computational time, as it involves numerous estimation procedures.

Usage

```
bootstrap_test(formula, data = NULL, alpha0, alpha1, B = 99)
```

Arguments

| | |
|---------|---|
| formula | A symbolic description of the model to be estimated, specified using the standard R formula syntax (e.g., $y \sim x1 + x2$). |
| data | A data frame containing the variables in the model. |
| alpha0 | First value of α . The bootstrap samples are generated using the MDPD estimates with $\alpha = \alpha_0$. |
| alpha1 | Second value of α . |
| B | A numeric value specifying the number of bootstrap replications. The default is 99. |

Value

A numeric. p -value of the bootstrap test.

Examples

```
## Example using the 'riceProdPhil' dataset from the `frontier` package
library(frontier)
data(riceProdPhil)

my.model <- log(PROD) ~ log(AREA) + log(LABOR) + log(NPK) + log(OTHER)
```

```

## Evaluate the closeness of ML estimates (alpha = 0) and
## MDPD estimates with alpha = 0.5.
bootstrap_test(my.model, data = riceProdPhil, alpha0=0.5, alpha1=0)

## Data with a single outlying observation
riceProdPhil2 <- riceProdPhil
riceProdPhil3 <- riceProdPhil

idx <- which.max(riceProdPhil$PROD)
riceProdPhil2$PROD[idx] <- riceProdPhil$PROD[idx]*10
riceProdPhil3$PROD[idx] <- riceProdPhil$PROD[idx]/100

## Evaluate the closeness of ML estimates (alpha = 0) and
## MDPD estimates with alpha = 0.5.
bootstrap_test( my.model, data = riceProdPhil2, alpha0=0.5, alpha1=0)
bootstrap_test( my.model, data = riceProdPhil3, alpha0=0.5, alpha1=0)

```

coef.rsfa

Coefficients Method for Class rsfa

Description

This function extracts the estimates from the object of class rsfa.

Usage

```

## S3 method for class 'rsfa'
coef(object, ...)

```

Arguments

| | |
|--------|--------------------------|
| object | An object of class rsfa. |
| ... | Unused. |

Value

A named vector of MDPD estimates.

Examples

```

## Example using the 'riceProdPhil' dataset from the `frontier` package
library(frontier)
data(riceProdPhil)

my.model <- log(PROD) ~ log(AREA) + log(LABOR) + log(NPK) + log(OTHER)

```

```
fit.ml <- rsfa(my.model, data = riceProdPhil)
coef(fit.ml)
fit.mdpde<- rsfa(my.model, data = riceProdPhil, alpha = 0.1)
coef(fit.mdpde)
```

efficiency

Function for Estimating Technical Efficiencies for Class rsfa

Description

This function returns the individual efficiencies calculated from the MDPD or ML estimates. The efficiencies are calculated using the estimator of Battese and Coelli (1988).

Usage

```
efficiency(object, ...)
```

Arguments

| | |
|--------|--------------------------|
| object | An object of class rsfa. |
| ... | Unsed. |

Value

A column vector of the estimated individual efficiencies.

References

Battese, G.E. and Coelli, T. (1988). Prediction of firm-level technical efficiencies with a generalized frontier production function and panel data. *Journal of Econometrics*, 38, 387-399.

Examples

```
## Example using the 'riceProdPhil' dataset from the `frontier` package
library(frontier)
data(riceProdPhil)

my.model <- log(PROD) ~ log(AREA) + log(LABOR) + log(NPK) + log(OTHER)
fit.ml <- rsfa(my.model, data = riceProdPhil)
efficiency(fit.ml)
fit.mdpde<- rsfa(my.model, data = riceProdPhil, alpha = 0.1)
efficiency(fit.mdpde)
```

| | |
|---------------|---|
| optimal_alpha | <i>Selecting an Optimal α for MDPDE</i> |
|---------------|---|

Description

This function recommends an optimal α for performing MDPD estimation. The function repeatedly calls `bootstrap_test` and selects the optimal α from the set $\{0, 0.05, 0.1, \dots, \text{specified } \alpha\}$.

Usage

```
optimal_alpha(formula, data = NULL, base_alpha = 0.5, B = 99, s_level = 0.1)
```

Arguments

| | |
|------------|---|
| formula | A symbolic description of the model to be estimated, specified using the standard R formula syntax (e.g., $y \sim x1 + x2$). |
| data | A data frame containing the variables in the model. |
| base_alpha | A numeric value. This selection procedure includes a bootstrap test. The bootstrap samples are generated using the MDPD estimates with $\alpha = \text{base_alpha}$, and the procedure selects an optimal α from the set $\{0, 0.05, 0.1, \dots, \text{base_alpha}\}$. |
| B | A numeric. This selecting procedure includes a bootstrap test, where B is the number of the bootstrap replications required for calculating a critical value. The default value is 99. |
| s_level | A numeric value representing the significance level for the bootstrap test. The default value is 0.1. |

Value

A character string indicating the recommended optimal α for MDPD estimation, selected from the set $\{0, 0.05, 0.1, \dots, \text{base_alpha}\}$.

Examples

```
## Example using the 'riceProdPhil' dataset from the `frontier` package
library(frontier)
data(riceProdPhil)

## It takes to time to get the result of the optimal_alpha() function.
my.model <- log(PROD) ~ log(AREA) + log(LABOR) + log(NPK) + log(OTHER)
optimal_alpha(my.model, data = riceProdPhil, base_alpha=0.5)

## Data with a single outlying observation
riceProdPhil2 <- riceProdPhil
riceProdPhil3 <- riceProdPhil

idx <- which.max(riceProdPhil$PROD)
```

```

riceProdPhil2$PROD[idx] <- riceProdPhil$PROD[idx]*10
riceProdPhil3$PROD[idx] <- riceProdPhil$PROD[idx]/100

optimal_alpha(my.model, data = riceProdPhil2, base_alpha=0.5)
optimal_alpha(my.model, data = riceProdPhil3, base_alpha=0.5)

```

print.rsfa

Print Method for Class rsfa

Description

This function prints the results of the rsfa estimation.

Usage

```

## S3 method for class 'rsfa'
print(x, ...)

```

Arguments

| | |
|-----|--------------------------|
| x | An object of class rsfa. |
| ... | Unused. |

Value

A named vector of MDPD estimates.

Examples

```

## Example using the 'riceProdPhil' dataset from the `frontier` package
library(frontier)
data(riceProdPhil)

my.model <- log(PROD) ~ log(AREA) + log(LABOR) + log(NPK) + log(OTHER)
fit.ml <- rsfa(my.model, data = riceProdPhil)
print(fit.ml)
fit.mdpde <- rsfa(my.model, data = riceProdPhil, alpha = 0.1)
print(fit.mdpde)

```

Description

This function performs the robust estimation for stochastic frontier models using the Minimum Density Power Divergence Estimator (MDPDE). The MDPDE is particularly useful when outliers in a dataset distort estimation results, especially regarding technical efficiencies. The parameter α in the MDPDE plays a crucial role in mitigating the impact of outliers when estimating coefficients and technical efficiencies. It actually controls the trade-off between robustness and efficiency of the MDPDE. The current estimation process is based on the following model:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + V - U := g(X, \beta) + V - U,$$

where V follows the normal distribution with mean 0 and variance σ_v^2 , and U follows the half-normal distribution with variance σ_u^2 . The MDPDE with α for the model above is given as follows:

$$\operatorname{argmin}_{\theta \in \Theta} \frac{1}{\sigma^\alpha} \left\{ \frac{\sqrt{2}}{\sqrt{\pi}} \int e^{-(1+\alpha)\frac{z^2}{2}} \Phi^{1+\alpha}(-z\lambda) dz - \left(1 + \frac{1}{\alpha}\right) \frac{1}{n} \sum_{i=1}^n e^{-\alpha \frac{(Y_i - g(X_i, \beta))^2}{2\sigma^2}} \Phi^\alpha\left(-\frac{Y_i - g(X_i, \beta)}{\sigma} \lambda\right) \right\},$$

where $\theta = (\beta_0, \beta_1, \dots, \beta_p, \sigma^2, \lambda)$ with $\sigma^2 = \sigma_v^2 + \sigma_u^2$ and $\lambda = \sigma_u/\sigma_v$. For more details, refer to Song et al. (2017).

Usage

```
rsfa(formula, data = NULL, alpha = 0, se = TRUE)
```

Arguments

| | |
|---------|---|
| formula | A symbolic description of the model to be estimated, specified using the standard R formula syntax (e.g., $y \sim x1 + x2$). |
| data | A data frame containing the variables in the model. |
| alpha | A numeric value controlling the robustness of the MDPDE. When alpha is zero (the default), 'rsfa' returns the ML estimates. It is not necessary for α to be large. Values of α close to zero are known to be efficient and sufficiently robust. The usually recommended range of α is 0.1 to 0.3. A function for selecting the optimal α is also provided in this package. For this purpose, see the <code>optimal_alpha()</code> function. |
| se | Logical. If TRUE (the default), standard errors are presented. |

Value

A list containing robust estimation results: estimates, standard errors, and residuals, as well as technical efficiencies.

| | |
|--------------|---|
| call | The matched call. |
| coefficients | A named vector of parameter estimates. |
| se | A vector of estimated standard errors. |
| residuals | A vector of residuals. |
| efficiencies | A vector of estimated technical efficiencies calculated using the estimator of Battese and Coelli (1988). |
| sigma2_V | Numeric. Estimated value of σ_V^2 . |
| sigma2_U | Numeric. Estimated value of σ_U^2 . |
| alpha | Numeric. α used in the MDPDE. |

References

Battese, G.E. and Coelli, T. (1988). Prediction of firm-level technical efficiencies with a generalized frontier production function and panel data. *Journal of Econometrics*, 38, 387-399.

Coelli, T. and Henningsen, A. (2020). *Stochastic Frontier Analysis*. R package version 1.1-8.

Song, J., Oh, D-h., and Kang, J. (2017). Robust estimation in stochastic frontier models. *Computational Statistics & Data Analysis*, 105, 243-267.

Examples

```
## Example using the 'riceProdPhil' dataset from the `frontier` package
library(frontier)
data(riceProdPhil)
summary(riceProdPhil)
```

```
## ML estimates and MDPD estimates with alpha=0.1
my.model <- log(PROD) ~ log(AREA) + log(LABOR) + log(NPK) + log(OTHER)
fit.ml <- rsfa(my.model, data = riceProdPhil)
summary(fit.ml)
```

```
fit.mdpde<- rsfa(my.model, data = riceProdPhil, alpha = 0.1)
summary(fit.mdpde)
```

```
## Comparison of the technical efficiencies from MLE and MDPDE
eff.ml<-efficiency(fit.ml)
eff.mdpde<-efficiency(fit.mdpde)
plot(eff.ml, eff.mdpde, pch=20,
      xlab="efficiency from MLE", ylab="efficiency from MDPDE")
abline(0,1, lty=2)
```

```
## Data with a single outlying observation
```



```

riceProdPhil2 <- riceProdPhil
riceProdPhil3 <- riceProdPhil
idx <- which.max(riceProdPhil$PROD)
riceProdPhil2$PROD[idx] <- riceProdPhil$PROD[idx]*10
riceProdPhil3$PROD[idx] <- riceProdPhil$PROD[idx]/100

## The technical efficiencies for "riceProdPhil2" data
fit.ml2<- rsfa(my.model, data = riceProdPhil2)
eff.ml2 <- efficiency(fit.ml2)
fit.mdpde2<- rsfa(my.model, data = riceProdPhil2, alpha = 0.1)
eff.mdpde2 <- efficiency(fit.mdpde2)

plot(eff.ml, eff.ml2, xlim = c(0, 1), ylim = c(0, 1), col="red", pch=20,
      xlab = "efficiency from MLE (riceProdPhil)",
      ylab = "efficiency (riceProdPhil2)")
points(eff.ml, eff.mdpde2, col="blue", pch=20)
abline(0,1,lty=2)
legend("topleft", legend = c("MLE", "MDPDE"), col = c("red", "blue"), pch = 20, bty="n")
title(main =
      "Efficiency comparison: MLE(no outlier) vs MLE and MPDPE (one over-performing producer)")

## The technical efficiencies for "riceProdPhil3" data
fit.ml3<- rsfa(my.model, data = riceProdPhil3)
eff.ml3 <- efficiency(fit.ml3)
fit.mdpde3<- rsfa(my.model, data = riceProdPhil3, alpha = 0.1)
eff.mdpde3 <- efficiency(fit.mdpde3)

plot(eff.ml, eff.ml3, xlim = c(0, 1), ylim = c(0, 1), col="red", pch=20,
      xlab = "efficiency from MLE (riceProdPhil)",
      ylab = "efficiency (riceProdPhil3)")
points(eff.ml, eff.mdpde3, col="blue", pch=20)
abline(0,1,lty=2)
legend("topleft", legend = c("MLE", "MDPDE"), col = c("red", "blue"), pch = 20, bty="n")
title(main =
      "Efficiency comparison: MLE(no outlier) vs MLE and MPDPE (one under-performing producer)")

```

summary.rsfa

Summary Method for Class rsfa

Description

This function provides a summary of the MDPD estimation, including coefficient estimates, standard errors, z-values, p-values, and significance codes.

Usage

```

## S3 method for class 'rsfa'
summary(object, ...)

```

Arguments

| | |
|--------|--------------------------|
| object | An object of class rsfa. |
| ... | Unused. |

Value

A summary table with estimates, standard errors, z-values, *p*-values, and significance codes.

Examples

```
## Example using the 'riceProdPhil' dataset from the `frontier` package
library(frontier)
data(riceProdPhil)

my.model <- log(PROD) ~ log(AREA) + log(LABOR) + log(NPK) + log(OTHER)
fit.ml <- rsfa(my.model, data = riceProdPhil)
summary(fit.ml)
fit.mdpde <- rsfa(my.model, data = riceProdPhil, alpha = 0.1)
summary(fit.mdpde)
```

Index

`bootstrap_test`, 2

`coef.rsfa`, 3

efficiency, 4

`optimal_alpha`, 5

`print.rsfa`, 6

`rsfa`, 7

`summary.rsfa`, 9