UNIVERSITY OF COPENHAGEN

Estimating CES functions in R
package micEconCES

Henningsen, Geraldine; Henningsen, Arne

Published in:
Symposium i Anvendt Statistik 2011

Publication date:
2011

Document version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Henningsen, G., & Henningsen, A. (2011). Estimating CES functions in R: package micEconCES. In P. Linde (Ed.), *Symposium i Anvendt Statistik 2011* (pp. 242-253). Danmarks Statistik.

Estimating the CES Function in R: Package micEconCES

Géraldine Henningsen, Systems Analysis Division, Risø National Laboratory for Sustainable Energy, Technical University of Denmark

Arne Henningsen, Institute of Food and Resource Economics, Faculty of Life Sciences, University of Copenhagen

1 Introduction

The Constant-Elasticity-of-Substitution (CES) function was developed as a generalisation of the Cobb-Douglas function by the Stanford group around Arrow et al. (1961). In recent years the CES has gained in importance in macroeconomics (e.g. Amras, 2004; Bentolila and Gilles, 2006) and growth theory (e.g. Caselli, 2005; Caselli and Coleman, 2006) as an alternative to the Cobb-Douglas function and it can be applied in many other fields.

The formal specification of a CES production function[1] with two inputs is

$$y = \gamma\left(\delta x_1^{-\rho} + (1-\delta)x_2^{-\rho}\right)^{-\frac{\nu}{\rho}}, \tag{1}$$

where y is the output quantity, x_1 and x_2 are the input quantities, and γ, δ, ρ, and ν are parameters. Parameter $\gamma \in (0,\infty)$ determines the productivity, $\delta \in (0,1)$ determines the optimal distribution of the inputs, $\rho \in (-1,0) \cup (0,\infty)$ determines the (constant) elasticity of substitution, which is $\sigma = 1/(1+\rho)$, and $\nu \in (0,\infty)$ is equal to the elasticity of scale.[2]

The CES function includes three special cases: for $\rho \to 0$, σ approaches 1 and the CES turns to the Cobb-Douglas form; for very large ρ, σ approaches 0 and the CES turns to the Leontief production function; and for $\rho \to -1$, σ approaches infinity and the CES turns to a linear function if ν is equal to 1.

As the CES function is non-linear in parameters and cannot be linearised analytically, it is not possible to estimate it with the usual linear estimation techniques.

[1]The CES functional form can be used to model different economic relationships (e.g. as production function or utility function). However, as the CES functional form is mostly used to model production technology, we name the independent (right-hand side) variables "inputs" and the dependent (left-hand side) variable "output" to keep the notation simple.

[2]Originally, the CES function of Arrow et al. (1961) could model only constant returns to scale but later Kmenta (1967) added the parameter ν, which allows for variable returns to scale if $\nu \neq 1$.

Therefore, the CES is often approximated by the so-called "Kmenta approximation" (Kmenta, 1967) or estimated by non-linear least-squares using different optimization algorithms. In this paper, we describe and compare these estimation approaches, explain how we implemented them in the R package *micEconCES* (Henningsen and Henningsen, 2010), and show how they can be used for economic analysis and modelling. The *micEconCES* package is available for download from the Comprehensive R Archive Network (CRAN, `http://CRAN.R-Project.org/package=micEconCES`).

2 Estimation of the CES production function

If the R package *micEconCES* (Henningsen and Henningsen, 2010) is installed, it can be loaded with the command

```
> library("micEconCES")
```

We demonstrate the usage of this package by estimating a CES model with an artifical data set, because this avoids several problems that usually occur with real-world data.

```
> cesData <- data.frame(x1 = rchisq(200, 10), x2 = rchisq(200,
+     10))
> cesData$y <- cesCalc(xNames = c("x1", "x2"), data = cesData,
+     coef = c(gamma = 1, delta = 0.6, rho = 0.5, nu = 1.1))
> cesData$y <- cesData$y + 2.5 * rnorm(200)
```

The first line creates a data set with two input variables (called `x1` and `x2`) that have 200 observations each and are generated from a random χ^2 distribution with 10 degrees of freedom. The second line uses the command `cesCalc` that is included in the *micEconCES* package and calculates the deterministic output variable (called `y`) given the CES production function with the two input variables `x1` and `x2` and the coefficients $\gamma = 1$, $\delta = 0.6$, $\rho = 0.5$, and $\nu = 1.1$. The last line generates the stochastic output variable by adding normally distributed random errors to the deterministic output variable.

As the CES function is non-linear in its parameters, the most straightforward way to estimate the CES function in R would be to use `nls`, which performs non-linear least-squares estimations.

```
> cesNls <- nls(y ~ gamma * (delta * x1^(-rho) + (1 - delta) *
+      x2^(-rho))^(-phi/rho), data = cesData, start = c(gamma = 0.5,
+      delta = 0.5, rho = 0.25, phi = 1))
> print(cesNls)

Nonlinear regression model
  model:  y ~ gamma * (delta * x1^(-rho) + (1 - delta) * x2^(-rho))^(-phi/rho)
   data:  cesData
 gamma  delta    rho    phi
1.0102 0.6271 0.6398 1.0955
 residual sum-of-squares: 1175

Number of iterations to convergence: 6
Achieved convergence tolerance: 4.147e-07
```

While the `nls` routine works well in this ideal artificial example, it does not perform well in many applications with real data, either because of non-convergence, convergence to a local minimum, or theoretically unreasonable parameter estimates. Therefore, we show alternative ways of estimating the CES function in the following subsections.

2.1 Kmenta approximation

Given that non-linear estimation methods are often troublesome—particularly during the 1960s and 1970s when computing power was very limited—Kmenta (1967) linearised the classical two-input CES production function by a Taylor series approximation at the point $\rho = 0$. This so-called Kmenta-approximation can be estimated by ordinary least-squares techniques.

$$\log y = \log \gamma + \nu\, \delta \log x_1 + \nu\, (1-\delta) \log x_2 - \frac{\rho\, \nu}{2}\, \delta\, (1-\delta)\, (\log x_1 - \log x_2)^2 \qquad (2)$$

The Kmenta approximation of the CES function can be estimated by the function `cesEst`, which is included in the *micEconCES* package. If argument `method` of this function is set to `"Kmenta"`, `cesEst` estimates the Kmenta approximation by OLS and calculates the parameters of the original CES function as well as their

covariance matrix using the delta method. This is done by the following command, where argument `yName` specifies the dependent variable, argument `xNames` specifies the explanatory variables, argument `data` specifies the data set, argument `method` specifies the estimation method, and argument `vrs` allows for variable returns to scale.

```
> cesKmenta <- cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData,
+     method = "Kmenta", vrs = TRUE)
> print(cesKmenta)

Estimated CES function

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE,
    method = "Kmenta")

Coefficients:
 gamma  delta    rho     nu
0.7425 0.6086 0.7153 1.2187
```

The estimation of the CES using the Kmenta approximation encounters several problems. First, it is a truncated Taylor series, whose remainder term must be seen as an omitted variable. Second, the Kmenta approximation converges to the underlying CES function only in a region of convergence, that is depending of the true parameters of the CES function (Thursby and Lovell, 1978). Although ν and δ can be estimated with small bias and mean squared error (MSE), results for γ and ρ are estimated with generally large bias and MSE (Maddala and Kadane, 1967; Thursby and Lovell, 1978; Thursby, 1980). More reliable results can only be obtained if $\rho \to 0$, and thus, $\sigma \to 1$ which increases the convergence region, i.e. if the underlying CES is of the Cobb-Douglas form. This is a major drawback of the Kmenta approximation as its purpose is to facilitate the estimation of functions with non-unitary σ.

2.2 Non-linear least-squares estimation

Various non-linear optimization algorithms have been used for estimating the parameters of the CES function by non-linear least-squares. Initially, the Levenberg-

Marquardt algorithm (Marquardt, 1963) was most commonly used. In a Monte Carlo study by Thursby (1980), the Levenberg-Marquardt algorithm outperforms the other methods and gives the best estimates of the CES parameters. However, the Levenberg-Marquardt algorithm performs as poorly as the other methods in estimating the elasticity of substitution (σ), meaning that the estimated σ tends to be biased towards infinity, unity, or zero.

To estimate a CES function by non-linear least-squares using the Levenberg-Marquardt algorithm, one can call the `cesEst` function with argument `method` set to `"LM"` or without this argument, as the Levenberg-Marquardt algorithm is the default estimation method used by `cesEst`. Argument `start` can be used to specify a vector of starting values, where the order must be γ, δ, ρ (only if ρ is not fixed, e.g. during grid search), and ν (only if the model has variable returns to scale). If no starting values are provided, they are determined automatically (see below). We estimate the same example as before now by the Levenberg-Marquardt algorithm.

```
> cesLm <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE)
> print(cesLm)
```

```
Estimated CES function

Call:
cesEst(yName = "y", xNames = c("x1", "x2"), data = cesData, vrs = TRUE)

Coefficients:
 gamma  delta    rho     nu
1.0102 0.6271 0.6398 1.0955
```

Function `cesEst` can use also three further gradient-based optimization algorithms for non-linear least-squares estimations of the CES function. If argument `method` of `cesEst` is set to `"CG"`, the "Conjugate Gradients" method based on Fletcher and Reeves (1964) is used. If it is set to `"Newton"`, an improved Newton-type algorithm as described in Dennis and Schnabel (1983) and Schnabel, Koontz, and Weiss (1985) is used. And if it is `"BFGS"`, the quasi-Newton method developed independently by Broyden (1970), Fletcher (1970), Goldfarb (1970), and Shanno (1970) is used.

While the gradient-based (local) optimization algorithms described above are designed to find local minima, global optimization algorithms, which are also known

as direct search methods, are designed to find the global minimum. They are more tolerant to not well-behaved objective functions but they usually converge more slowly than the gradient-based methods. However, increasing computing power has made these algorithms suitable for day-to-day use.

If argument `method` is set to `"NM"`, the so-called Nelder-Mead algorithm (Nelder and Mead, 1965) is used. If it is set to `"SANN"` a variant of the "Simulated Annealing" algorithm (Kirkpatrick, Gelatt, and Vecchi, 1983; Cerny, 1985) given in Bélisle (1992) is used. Finally, if argument `method` is set to `"DE"`, the Differential Evolution algorithm (Storn and Price, 1997) is used. This algorithm has proven to be effective and accurate on a large range of optimisation problems, inter alia the CES function (Mishra, 2007).

As a meaningful analysis based on a CES function requires that this function is consistent with economic theory, it is often desirable to constrain the parameter space to the economically meaningful region. If argument `method` is set to `"L-BFGS-B"`, function `cesEst` estimates a CES function under parameter constraints using a modification of the BFGS algorithm suggested by Byrd et al. (1995). If argument `method` is set to `"PORT"`, the so-called PORT routines (Gay, 1990) are used. They include a quasi-Newton optimisation algorithm that allows for box constraints on the parameters and has several advantages over traditional Newton routines.

2.3 Grid search for ρ

As the objective function for estimating the CES by non-linear least-squares shows a tendency to "flat surfaces" around the minimum—in particular for a wide range of values for ρ—many optimization algorithms have problems in finding the minimum of the objective function. This problem can be alleviated by performing a one-dimensional grid search, where a sequence of values for ρ is pre-selected and the remaining parameters are estimated by non-linear least-squares holding ρ fixed at each of the pre-defined values. Later, the estimation with the value of ρ that results in the smallest sum of squared residuals is chosen.

The function `cesEst` carries out this grid search procedure, if the user sets its argument `rho` to a numeric vector containing the values of ρ that should be used in the grid search. The estimation of the other parameters during the grid search can use all non-linear optimization algorithms described above. Since the "best" value of ρ that was found in the grid search is not known but estimated (as the other parameters but with a different method), the covariance matrix of the estimated parameters includes ρ and is calculated as if ρ was estimated as usual. The following command estimates the CES function by a one-dimensional grid search for ρ, where the pre-selected values for ρ are the values from -0.3 to 1.5 with an increment of 0.1 and the default optimisation method, the Levenberg-Marquardt algorithm is used to estimate the remaining parameters.

```
> cesGrid <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE, rho = seq(from =
+      to = 1.5, by = 0.1))
```

An overview of the relationship between the pre-selected values of ρ and the corresponding sums of the squared residuals can be obtained by applying the plot method.[3] The resulting graph is shown in figure 1.

```
> plot(cesGrid)
```

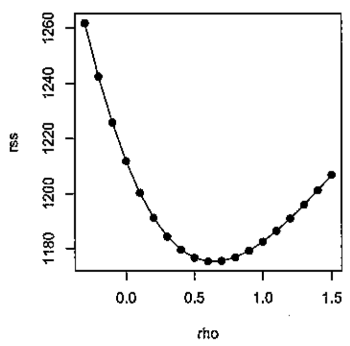

Figure 1: Values of ρ and corresponding sums of squared residuals

The results of this grid search algorithm can be either used directly or used as starting values for a non-linear least-squares estimation so that also ρ values between the grid points can be estimated. Starting values can be set by argument startVal.

```
> cesStartGrid <- cesEst("y", c("x1", "x2"), cesData, vrs = TRUE,
+      start = coef(cesGrid))
```

[3]This plot method can be applied only if the model was estimated by grid search.

3 Implementation

The function cesEst is the primary user interface of the *micEconCES* package (Henningsen and Henningsen, 2010). However, the actual estimations are carried out by internal helper functions or functions from other packages.

The non-linear least-squares estimations are carried out by various optimisers from other packages. Estimations with the Levenberg-Marquardt algorithm are performed by function nls.lm of the *minpack.lm* package (Elzhov and Mullen, 2009), which is an R interface to the FORTRAN package *MINPACK* (Moré, Garbow, and Hillstrom, 1980). Estimations with the Conjugate Gradients (CG), BFGS, Nelder-Mead (NM), Simulated Annealing (SANN), and L-BFGS-B algorithms use the function optim from the *stats* package (R Development Core Team, 2009). Estimations with the Newton-type algorithm are performed by function nlm from the *stats* package (R Development Core Team, 2009), which uses the FORTRAN library *UNCMIN* (Schnabel, Koontz, and Weiss, 1985) with line search as step selection strategy. Estimations with the Differential Evolution (DE) algorithm are performed by function DEoptim from the *DEoptim* package (Ardia and Mullen, 2009). Estimations with the PORT routines use function nlminb from the *stats* package (R Development Core Team, 2009), which uses the FORTRAN library *PORT* (Gay, 1990).

The grid search procedure is implemented in the internal function cesEstGridRho. This function consecutively calls cesEst for each of the pre-selected values of ρ, where argument rho of cesEst is set to one of the pre-selected values at each call. If argument rho of cesEst is a single scalar value, cesEst does not perform a grid search but estimates the CES function by non-linear least-squares with parameter ρ fixed at the value of argument rho.

Function cesCalc can be used to calculate the output quantity of the CES function given input quantities and parameters. An example of using cesCalc is shown in the beginning of section 2, where the output variable of an artificial data set that is used to demonstrate the usage of cesEst is generated with this function. Furthermore, the cesCalc function is used by the internal function cesRss that calculates and returns the sum of squared residuals, which is the objective function in the non-linear least-squares estimations. As the CES function is not defined for $\rho = 0$, cesCalc calculates in this case the output quantity with the limit of the CES function for $\rho \to 0$, which is the Cobb-Douglas function.

We noticed that the calculations with cesCalc using equation (1) are imprecise when ρ is close to 0. This is caused by rounding errors that are unavoidable on digital computers but are usually negligible. However, rounding errors can get large in specific circumstances, e.g. in the CES function with very small ρ, when very small (in absolute terms) exponents $(-\rho)$ are applied first and then a very large (in absolute terms) exponent $(-\nu/\rho)$ is applied. Therefore, `cesCalc` uses a first-order Taylor series approximation at the point $\rho = 0$ for calculating the output of the CES function, if the absolute value of ρ is smaller than or equal to argument `rhoApprox`, which is $5 \cdot 10^{-6}$ by default.

The internal function `cesDerivCoef` returns the partial derivatives of the CES function with respect to all coefficients at all provided data points. These derivatives are not defined for $\rho = 0$ and are imprecise if ρ is close to zero (similar to the output variable of the CES function, see above). Therefore, we calculate these derivatives by first-order Taylor series approximations at the point $\rho = 0$ if ρ is zero or close to zero. Function `cesDerivCoef` has an argument `rhoApprox` that can be used to set the threshold levels for defining when ρ is "close" to zero. This argument must be a numeric vector with exactly four elements that define the thresholds for the derivatives with respect to the four parameters. By default, these thresholds are $5 \cdot 10^{-6}$ for $\partial y/\partial \gamma$, $\partial y/\partial \delta$, and $\partial y/\partial \nu$, and 10^{-3} for $\partial y/\partial \rho$.

Function `cesDerivCoef` is used to provide analytical gradients for the gradient-based optimization algorithms, i.e. Levenberg-Marquardt, Conjugate Gradients, Newton-type, BFGS, L-BFGS-B, and PORT. Furthermore, this function is used to obtain the gradient matrix for calculating the asymptotic covariance matrix of the non-linear least-squares estimator.

The asymptotic covariance matrix of the non-linear least-squares estimator obtained by the various iterative optimisation methods is calculated by (Greene, 2008, p. 292)

$$\hat{\sigma}^2 \left(\left(\frac{\partial y}{\partial \theta} \right)^\top \frac{\partial y}{\partial \theta} \right)^{-1}, \tag{3}$$

where $\partial y/\partial \theta$ denotes the matrix of the derivatives of the dependent variable with respect to all parameters evaluated at each observation, and $\hat{\sigma}^2$ denotes the estimated variance of the residuals. As equation (3) is only valid asymptotically, we calculate the estimated variance of the residuals by

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^{N} u_i^2, \tag{4}$$

i.e. without correcting for degrees of freedom.

If no starting values are provided by the user, function `cesEstStart` determines the starting values automatically. The starting value of δ is always set to 0.5. If the coefficient ρ is estimated (not fixed as, e.g., during grid search), the starting value of ρ is set to 0.25, which corresponds to an elasticity of substitution of 0.8. If the estimation allows for a model with variable returns to scale, the starting value of ν is set to 1, which corresponds to constant returns to scale. Finally, the starting value of γ is set to a value so that the mean of the endogenous variable is equal to the mean of its fitted values, i.e.

$$\gamma = \frac{\frac{1}{N} \sum_{i=1}^{N} y_i}{\frac{1}{N} \sum_{i=1}^{N} \left(0.5\, x_{1i}^{-\rho_0} + 0.5\, x_{2i}^{-\rho_0} \right)^{-\frac{1}{\rho_0}}}, \tag{5}$$

where ρ_0 is either the pre-selected value of ρ (if ρ is fixed) or the starting value of ρ, i.e. 0.25 (if ρ is estimated).

4 Conclusion

The R package *micEconCES* provides tools for economic analysis with the CES function, particularly the econometric estimation using the Kmenta approximation and non-linear least squares based on various optimization algorithms. A special feature is the grid search for the parameter ρ. A Monte Carlo simulation performed by the authors proved satisfying results for all estimation methods.

In addition to the traditional CES function with two inputs, the latest version of the *micEconCES* package supports also the nested CES functions for 3 and four inputs.

References

Amras, P. 2004. "Is the U.S. Aggregate Production Function Cobb-Douglas? New Estimates of the Elasticity of Substitution." *Contribution in Macroeconomics* 4:Article 4.

Ardia, D., and K. Mullen. 2009. *DEoptim: Global Optimization by Differential Evolution.* R package version 2.0-3.

Arrow, K.J., B.H. Chenery, B.S. Minhas, and R.M. Solow. 1961. "Capital-labor substitution and economic efficiency." *The Review of Economics and Statistics* 43:225–250.

Bélisle, C.J.P. 1992. "Convergence Theorems for a Class of Simulated Annealing Algorithms on R^d." *Journal of Applied Probability* 29:885–895.

Bentolila, S.J., and S.P. Gilles. 2006. "Explaining Movements in the Labour Share." *Contributions to Macroeconomics* 3:Article 9.

Broyden, C.G. 1970. "The Convergence of a Class of Double-rank Minimization Algorithms." *Journal of the Institute of Mathematics and Its Applications* 6:76–90.

Byrd, R., P. Lu, J. Nocedal, and C. Zhu. 1995. "A limited memory algorithm for bound constrained optimization." *SIAM Journal for Scientific Computing* 16:1190–1208.

Caselli, F. 2005. "Accounting for Cross-Country Income Differences." In P. Aghion and S. N. Durlauf, eds. *Handbook of Economic Growth*. North Holland, pp. 679–742.

Caselli, F., and I. Coleman, Wilbur John. 2006. "The World Technology Frontier." *American Economic Review* 96:499–522.

Cerny, V. 1985. "A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm." *Journal of Optimization Theory and Applications* 45:41–51.

Dennis, J.E., and R.B. Schnabel. 1983. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs (NJ, USA): Prentice-Hall.

Elzhov, T.V., and K.M. Mullen. 2009. *minpack.lm: R Interface to the Levenberg-Marquardt Nonlinear Least-Squares Algorithm Found in MINPACK*. R package version 1.1-4.

Fletcher, R. 1970. "A New Approach to Variable Metric Algorithms." *Computer Journal* 13:317–322.

Fletcher, R., and C. Reeves. 1964. "Function minimization by conjugate gradients." *Computer Journal* 7:48–154.

Gay, D.M. 1990. "Usage Summary for Selected Optimization Routines." Computing Science Technical Report No. 153, AT&T Bell Laboratories.

Goldfarb, D. 1970. "A Family of Variable Metric Updates Derived by Variational Means." *Mathematics of Computation* 24:23–26.

Greene, W.H. 2008. *Econometric Analysis*, 6th ed. Prentice Hall.

Henningsen, A., and G. Henningsen. 2010. *micEconCES: Analysis with the Con-*

stant Elasticity of Scale (CES) Function. R package version 0.8, `http://CRAN.R-project.org/package=micEconCES`.

Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi. 1983. "Optimization by Simulated Annealing." *Science* 220:671–680.

Kmenta, J. 1967. "On Estimation of the CES Production Function." *International Economic Review* 8:180–189.

Maddala, G., and J. Kadane. 1967. "Estimation of Returns to Scale and the Elasticity of Substitution." *Econometrica* 24:419–423.

Marquardt, D.W. 1963. "An Algorithm for Least-Squares Estimation of Non-linear Parameters." *Journal of the Society for Industrial and Applied Mathematics* 11:431–441.

Mishra, S.K. 2007. "A Note on Numerical Estimation of Sato's Two-Level CES Production Function." MPRA Paper No. 1019, North-Eastern Hill University, Shillong.

Moré, J.J., B.S. Garbow, and K.E. Hillstrom. 1980. *MINPACK*. : Argonne National Laboratory.

Nelder, J.A., and R. Mead. 1965. "A Simplex Algorithm for Function Minimization." *Computer Journal* 7:308–313.

R Development Core Team. 2009. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, ISBN 3-900051-07-0.

Schnabel, R.B., J.E. Koontz, and B.E. Weiss. 1985. "A Modular System of Algorithms for Unconstrained Minimization." *ACM Transactions on Mathematical Software* 11:419–440.

Shanno, D.F. 1970. "Conditioning of Quasi-Newton Methods for Function Minimization." *Mathematics of Computation* 24:647–656.

Storn, R., and K. Price. 1997. "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces." *Journal of Global Optimization* 11:341–359.

Thursby, J.G. 1980. "CES Estimation Techniques." *The Review of Economics and Statistics* 62:295–299.

Thursby, J.G., and C.A.K. Lovell. 1978. "In Investigation of the Kmenta Approximation to the CES Function." *International Economic Review* 19:363–377.