Constant Elasticity of Substitution Production Function with Multiple Inputs

Fan Wang

2022-08-11

Contents

1 Constant Elasticity of Substitution with Multiple Inputs

Go to the **RMD**, **PDF**, or **HTML** version of this file. Go back to Python Code Examples Repository (bookdown site) or the pyfan Package (API).

```
import numpy as np
```

1.1 Expenditure Minimization Problem (CRS) with Three Inputs

In this section, we will solve the expenditure minimization problem with multiple inputs given a constant elasticity of substitution production function with constant returns to scale.

Let X, Y, and Z be three inputs. p_x, p_y, and p_z are the prices of these inputs. Let Q be the output requirement, A be aggregate productivity, ρ be the elasticity of substitution across the inputs, and α_x, α_y, and $\alpha_z = 1 - \alpha_x - \alpha_y$ be the relative productivity of the three inputs.

This problem has analytical solutions.

First, we write down the problem where we choose x, y, and z inputs jointly to minimize overall cost of production, while achieving the output target Q,

$$\min_{x,y,z} (p_x X + p_y Y + p_z Z)$$
$$\text{s.t., } Q = f(X,Y,Z) = A \cdot (\alpha_x X^\rho + \alpha_y Y^\rho + \alpha_z Z^\rho)^{\frac{1}{\rho}}$$

Second, we write down the Lagrangian.

$$\mathcal{L}(X,Y,Z) = (p_x X + p_y Y + p_z Z) - \lambda \left(A \cdot (\alpha_x X^\rho + \alpha_y Y^\rho + \alpha_z Z^\rho)^{\frac{1}{\rho}} - Q \right)$$

Third, we take derivatives. Note that $\frac{1}{\rho} - 1 = \frac{1-\rho}{\rho}$.

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial X} &= p_x - \lambda \frac{A}{\rho} f(X,Y,Z)^{1-\rho} \cdot \left(\alpha_x \rho X^{\rho-1}\right) \\
\frac{\partial \mathcal{L}}{\partial Y} &= p_y - \lambda \frac{A}{\rho} f(X,Y,Z)^{1-\rho} \cdot \left(\alpha_y \rho Y^{\rho-1}\right) \\
\frac{\partial \mathcal{L}}{\partial Z} &= p_z - \lambda \frac{A}{\rho} f(X,Y,Z)^{1-\rho} \cdot \left(\alpha_z \rho Z^{\rho-1}\right) \\
\frac{\partial \mathcal{L}}{\partial \lambda} &= Q - A \cdot \left(\alpha_x X^{\rho} + \alpha_y Y^{\rho} + \alpha_z Z^{\rho}\right)^{\frac{1}{\rho}}
\end{aligned}$$

Fourth, at optimal choices, the derivatives are equal to zero.

$$\begin{aligned}
\frac{1}{\lambda} &= \frac{1}{p_x} A f(X,Y,Z)^{1-\rho} \cdot \left(\alpha_x X^{\rho-1}\right) \\
\frac{1}{\lambda} &= \frac{1}{p_y} A f(X,Y,Z)^{1-\rho} \cdot \left(\alpha_y Y^{\rho-1}\right) \\
\frac{1}{\lambda} &= \frac{1}{p_z} A f(X,Y,Z)^{1-\rho} \cdot \left(\alpha_z Z^{\rho-1}\right) \\
Q &= A \cdot \left(\alpha_x X^{\rho} + \alpha_y Y^{\rho} + \alpha_z Z^{\rho}\right)^{\frac{1}{\rho}}
\end{aligned}$$

Fifth, we combine the first and the second, as well as the first and the third equations together.

$$\begin{aligned}
\frac{1}{p_x} A f(X,Y,Z)^{1-\rho} \cdot \left(\alpha_x X^{\rho-1}\right) &= \frac{1}{p_y} A f(X,Y,Z)^{1-\rho} \cdot \left(\alpha_y Y^{\rho-1}\right) \\
\frac{1}{p_x} A f(X,Y,Z)^{1-\rho} \cdot \left(\alpha_x X^{\rho-1}\right) &= \frac{1}{p_z} A f(X,Y,Z)^{1-\rho} \cdot \left(\alpha_z Z^{\rho-1}\right) \\
Q &= A \cdot \left(\alpha_x X^{\rho} + \alpha_y Y^{\rho} + \alpha_z Z^{\rho}\right)^{\frac{1}{\rho}}
\end{aligned}$$

Sixth, we simplify the optimality conditions above to cancel out duplicative terms.

$$\begin{aligned}
\frac{p_y}{p_x} &= \frac{\alpha_y}{\alpha_x} \cdot \left(\frac{Y}{X}\right)^{\rho-1} \\
\frac{p_z}{p_x} &= \frac{\alpha_z}{\alpha_x} \cdot \left(\frac{Z}{X}\right)^{\rho-1} \\
Q &= A \cdot \left(\alpha_x X^{\rho} + \alpha_y Y^{\rho} + \alpha_z Z^{\rho}\right)^{\frac{1}{\rho}}
\end{aligned}$$

Seventh, we write optimal Y and Z choices as functions of X.

$$\begin{aligned}
Y &= \left(\frac{p_x}{p_y} \cdot \frac{\alpha_y}{\alpha_x}\right)^{\frac{1}{1-\rho}} \cdot X \\
Z &= \left(\frac{p_x}{p_z} \cdot \frac{\alpha_z}{\alpha_x}\right)^{\frac{1}{1-\rho}} \cdot X \\
Q &= A \cdot \left(\alpha_x X^{\rho} + \alpha_y Y^{\rho} + \alpha_z Z^{\rho}\right)^{\frac{1}{\rho}}
\end{aligned}$$

Eighth, we eliminate Y and Z from the quantity constraint, and simplify the resulting equation as a function of X.

$$Q = A \cdot \left(\alpha_x X^{\rho} + \alpha_y \left(\left(\frac{p_x}{p_y} \cdot \frac{\alpha_y}{\alpha_x} \right)^{\frac{1}{1-\rho}} \cdot X \right)^{\rho} + \alpha_z \left(\left(\frac{p_x}{p_z} \cdot \frac{\alpha_z}{\alpha_x} \right)^{\frac{1}{1-\rho}} \cdot X \right)^{\rho} \right)^{\frac{1}{\rho}}$$

$$Q = A \cdot \left(\alpha_x + \alpha_y \left(\frac{p_x}{p_y} \cdot \frac{\alpha_y}{\alpha_x} \right)^{\frac{\rho}{1-\rho}} + \alpha_z \left(\frac{p_x}{p_z} \cdot \frac{\alpha_z}{\alpha_x} \right)^{\frac{\rho}{1-\rho}} \right)^{\frac{1}{\rho}} \cdot X$$

Ninth, we have completed deriving the optimal input choice equation that minimizes the cost of production while achieving the target level of production. $X^* \left(Q, p_x, p_y, p_z \right)$ denotes the cost minimizing optimal X input choice given prices and the output quantity target Q.

$$X^* \left(Q, p_x, p_y \right) = \frac{Q}{A} \cdot \left(\alpha_x + \alpha_y \left(\frac{p_x}{p_y} \cdot \frac{\alpha_y}{\alpha_x} \right)^{\frac{\rho}{1-\rho}} + \alpha_z \left(\frac{p_x}{p_z} \cdot \frac{\alpha_z}{\alpha_x} \right)^{\frac{\rho}{1-\rho}} \right)^{-\frac{1}{\rho}}$$

1.2 Expenditure Minimization Problem (CRS) with N Inputs

Now, we generalize the solutions we derived above to the N inputs CES expenditure minimization problem.

Let $\{X_i\}_{i=1}^N$ be N inputs, and $\{p_i\}_{i=1}^N$ are prices on these N inputs. Let Q be the output requirement, A be aggregate productivity, ρ be the elasticity of substitution across the inputs, and $\{\alpha_i\}_{i=1}^N$ are the relative productivity of the three inputs, with $\sum_i^N \alpha_i = 1$.

First, we write down the N inputs problem.

$$\min_{\{X_i\}_{i=1}^N} \left(\sum_{i=1}^N p_i X_i \right)$$

$$\text{s.t., } Q = f\left(\{X_i\}_{i=1}^N \right) = A \cdot \left(\sum_{i=1}^N \alpha_i X_i^{\rho} \right)^{\frac{1}{\rho}}$$

Second, the relative optimality condition remains the same as prior.

$$\forall i, j \in \{1, ..., N\}, \frac{p_i}{p_j} = \frac{\alpha_i}{\alpha_j} \cdot \left(\frac{X_i}{X_j} \right)^{\rho - 1}$$

Third, the resulting optimal expenditure minimizing input choice X_j^* is shown below.

$$X_j^* \left(Q, \{p_i\}_{i=1}^N \right) = \frac{Q}{A} \cdot \left(\sum_{i=1}^N \alpha_i \cdot \left(\frac{p_j}{p_i} \cdot \frac{\alpha_i}{\alpha_j} \right)^{\frac{\rho}{1-\rho}} \right)^{-\frac{1}{\rho}}$$

Given model parameters, prices and quantity requirements, the optimal choice equation above can be easily implemented.

1.2.1 Marginal Cost of Output

Each of the input for the production function has a separate input price p_i. Given the optimal expenditure minimizing input choices $\{X_i^*\}_{i=1}^B (Q)$, when the output requirement Q increases, how does the overall cost of production increase?

This is a question about the marginal cost, which is constant since the production function has constant returns to scale.

First, we write down the objective of the minimizing problem (production cost), given optimal choices.

$$\begin{aligned} COST^*(Q) =& \sum_{j=1}^{N} p_j \cdot X_j^* \left(Q, \{p_i\}_{i=1}^N\right) \\ =& \sum_{j=1}^{N} p_j \cdot \frac{Q}{A} \cdot \left(\sum_{i=1}^{N} \alpha_i \cdot \left(\frac{p_j}{p_i} \cdot \frac{\alpha_i}{\alpha_j}\right)^{\frac{\rho}{1-\rho}}\right)^{-\frac{1}{\rho}} \\ =& \frac{Q}{A} \cdot \left(\sum_{j=1}^{N} p_j \cdot \left(\sum_{i=1}^{N} \alpha_i \cdot \left(\frac{p_j}{p_i} \cdot \frac{\alpha_i}{\alpha_j}\right)^{\frac{\rho}{1-\rho}}\right)^{-\frac{1}{\rho}}\right) \end{aligned}$$

Second, take derivative of the minimal cost function with respect to Q. The resulting derivative is not a function of Q.

$$\text{MCofQ} = \frac{\partial \text{COST}^*(Q)}{\partial Q} = \frac{1}{A} \cdot \left(\sum_{j=1}^{N} p_j \cdot \left(\sum_{i=1}^{N} \alpha_i \cdot \left(\frac{p_j}{p_i} \cdot \frac{\alpha_i}{\alpha_j}\right)^{\frac{\rho}{1-\rho}}\right)^{-\frac{1}{\rho}}\right)$$

Third, note that the resulting equation is a productivity-scaled "weighted sum" of the cost-minimizing level of X_j required per unit of Q, $\frac{X_j^*\left(Q,\{p_i\}_{i=1}^N\right)}{Q}$, where the weights are cost for each j, p_j and where productivity-scaled means to divide by A.

1.2.2 Matrix Representation of Solutions

Now we represent the solutions to the problem in matrix form.

First, note that we need to divide each price by every other price. Which means we are taking the kronecker product (outer product) of two vectors.

$$\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \otimes \begin{bmatrix} \frac{1}{p_1} \\ \frac{1}{p_2} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{p_1} & \frac{1}{p_2} \end{bmatrix} = \begin{bmatrix} \frac{p_1}{p_1} & \frac{p_1}{p_2} \\ \frac{p_2}{p_1} & \frac{p_2}{p_2} \end{bmatrix}$$

Second, combining all operations, the matrix representation of the optimal choice formula is as shown below. Note that $\circ$ denotes element-wise operations (Hadamard product), $^{\circ(-1)}$ denotes element-wise inverse operations (Hadamard Inverse), and $\cdot$ denotes matrix multiplication. Note that Q and A are scalar values as prior, but P and α are N by 1 vectors.

$$X^* = \underbrace{\frac{Q}{A}\left(\left(\underbrace{\overbrace{\left(P \cdot \left(P^{\circ(-1)}\right)^T\right)}^{N \text{ by } N} \circ \overbrace{\left(\alpha \cdot \left(\alpha^{\circ(-1)}\right)^T\right)^T}^{N \text{ by } N}}_{N \text{ by } N}\right)^{\frac{\rho}{1-\rho}} \cdot \overbrace{\alpha}^{N \text{ by } 1}\right)^{\frac{-1}{\rho}}}_{N \text{ by } 1}$$

Third, the matrix representation of the aggregate marginal cost equation is shown below.

$$\text{MCofQ} = \frac{1}{A} \underbrace{\left(\overbrace{P^T}^{1 \text{ by } N} \cdot \left(\underbrace{\left(P \cdot \left(P^{\circ(-1)} \right)^T \right)}_{N \text{ by } N} \circ \overbrace{\left(\alpha \cdot \left(\alpha^{\circ(-1)} \right)^T \right)^T}^{N \text{ by } N} \right)^{\frac{\rho}{1-\rho}} \cdot \overbrace{\alpha}^{N \text{ by } 1} \right)^{\frac{-1}{\rho}}}_{1 \text{ by } 1}$$

1.2.3 Computational Implementation

Now we write a small program with the solutions for the cost minimization problem as well as the aggregate marginal cost equation.

```
# The CRS CES solver function
def ffi_ces_crs_solver(ar_price, ar_share,
                       fl_elasticity,
                       fl_Q, fl_A):
    # Price Ratio: divide each price by every other price
    # PRICES x trans(1/PRICE)
    # SHARES x trans(1/SHARES)
    # fl_elasticity = 0.5
    # fl_Q = 1
    # fl_A = 1

    # np.random.seed(8888)
    # ar_price = np.random.uniform(size=3)
    # # ar_price = np.array([1,1,1])
    # # ar_price = np.array([0.6965, 0.2861, 0.2269])

    # ar_share = np.random.uniform(size=3)
    # ar_share = np.array([1,1,1])/3
    # ar_share = np.array([0.3255, 0.4247, 0.2498])
    ar_share = ar_share/sum(ar_share)

    # Each column divides by a different number
    # [a;b;c] x [1/a, 1/b, 1/c] : (3 by 1) x (1 by 3)
    mt_rela_price = np.outer(ar_price, np.reciprocal(ar_price))
    np.reciprocal(mt_rela_price)
    mt_rela_share = np.outer(ar_share, np.reciprocal(ar_share))

    # (p_j/p_i) x (alpha_i/alpha_j)
    mt_rela_prcshr = np.multiply(mt_rela_price, np.reciprocal(mt_rela_share))
    mt_rela_prcshr_rho = mt_rela_prcshr**(fl_elasticity/(1-fl_elasticity))

    # alpha_i x [(p_j/p_i) x (alpha_i/alpha_j)]^(rho/(1-rho))
    # 1/p_i are columns
    # (N by N) x (N by 1)
    ar_sum = np.matmul(mt_rela_prcshr_rho, ar_share)
    ar_sum_rho = ar_sum**(-1/fl_elasticity)

    # rescale
    ar_opti_costmin_x = ar_sum_rho * fl_Q/fl_A
    ar_opti_costmin_x
```

```
    # weighted average of prices
    fl_mc_aggprice = (np.dot(ar_sum_rho, ar_price)/fl_A)
    fl_mc_aggprice

    return ar_opti_costmin_x, fl_mc_aggprice


# Test function
fl_elasticity = 0.5
fl_Q = 1
fl_A = 1

# Test 1 fixed inputs
ar_price = np.array([0.6965, 0.2861, 0.2269])
ar_share = np.array([0.3255, 0.4247, 0.2498])
ffi_ces_crs_solver(ar_price, ar_share, fl_elasticity, fl_Q, fl_A)
# Test 2 common price
```

```
## (array([0.19527139, 1.97018808, 1.08366427]), 0.9455607576080614)
```

```
ar_price_common = np.array([1, 1, 1])
ffi_ces_crs_solver(ar_price_common, ar_share, fl_elasticity, fl_Q, fl_A)

# Test 3 common share
```

```
## (array([0.8712591 , 1.48323465, 0.51313331]), 2.8676270655589446)
```

```
ar_share_common = np.array([1, 1, 1])/3
ffi_ces_crs_solver(ar_price, ar_share_common, fl_elasticity, fl_Q, fl_A)

# Test 4 fixed price and shares
```

```
## (array([0.21274943, 1.2608834 , 2.00466427]), 0.9637770393078176)
```

```
np.random.seed(8888)
ar_price_rand = np.random.uniform(size=3)
ar_share_rand = np.random.uniform(size=3)
ffi_ces_crs_solver(ar_price_rand, ar_share_rand, fl_elasticity, fl_Q, fl_A)

# relative price function
```

```
## (array([2.49155320e-04, 3.99020825e-03, 1.24045273e+00]), 0.551620115706425)
```

```
def ffi_ces_rela_opti(ar_inputs_x, ar_share,
                      fl_elasticity, it_idx_base=0):

    # relative prices, with it_index_price_1 price normalized to 1.
    ar_rela_prices = np.multiply((ar_share * 1/ar_share[it_idx_base]),
                                 (ar_inputs_x * 1/ar_inputs_x[it_idx_base])**(fl_elasticity - 1))

    return(ar_rela_prices)


# Test
# Input optimal quantities
fl_elasticity = 0.5
```

```
fl_Q = 1
fl_A = 1
ar_price = np.array([0.6965, 0.2861, 0.2269])
ar_share = np.array([0.3255, 0.4247, 0.2498])
ar_opti_costmin_x, fl_mc_aggprice = ffi_ces_crs_solver(ar_price, ar_share,
                                                       fl_elasticity,
                                                       fl_Q, fl_A)
# Solve for relative prices given demand optimality
it_index_price_1 = 0
ar_rela_prices = ffi_ces_rela_opti(ar_opti_costmin_x, ar_share, fl_elasticity,
                                   it_idx_base=it_index_price_1)
ar_price/ar_price[it_index_price_1]
```

```
## array([1.        , 0.41076813, 0.32577172])
```