The Stata Journal (2009)
9, Number 1, pp. 161–165

Evaluating concavity for production and cost functions

Christopher F. Baum
Department of Economics
Boston College
Chestnut Hill, MA
baum@bc.edu

Teresa Linz
Center for Development Research
University of Bonn
Bonn, Germany
tlinz@uni-bonn.de

Econometric estimation of firms' cost functions is a well-known technique, generally involving the joint estimation of the cost function with a set of cost share equations, one for each factor of production. Under common assumptions about the structure of production and cost, these equations are linear and can be consistently estimated with `sureg` subject to linear constraints. See Greene (2008, sec. 10.4.2).

A crucial assumption derived from the analytics of production and cost functions is that the cost function should be globally concave in each of the factors. Although it may not be possible to establish global concavity over all possible ranges of factor prices and levels of output, researchers have often been dissatisfied with evaluating concavity at a single point in the multidimensional space such as the multivariate point of means. Empirical rejection of concavity may indicate a misspecification of the underlying model of production and cost.

An alternative that might provide stronger evidence on the suitability of the chosen production and cost functions involves evaluating whether the estimated function is concave at each point in the sample space. In this note, we illustrate how this can readily be computed in Stata.

We illustrate with an example of a five-factor production function and consider the dual of the production problem. The cost function provides an estimate of the minimum cost (C) of producing the level of output (y) given the factor prices (P) of water (w), electricity (e), labor (l), capital (c), and diesel (d). A transcendental logarithmic, or "translog", cost function is used (Berndt and Christensen 1973), yielding the specification

$$\begin{aligned}
\log C = \alpha_0 + \alpha_w \log P_w + \alpha_e \log P_e + \alpha_l \log P_l + \alpha_c \log P_c + \alpha_d \log P_d + \alpha_y \log y + \\
\frac{1}{2}\gamma_{ww} \log P_w^2 + \frac{1}{2}\gamma_{ee} \log P_e^2 + \frac{1}{2}\gamma_{ll} \log P_l^2 + \frac{1}{2}\gamma_{cc} \log P_c^2 + \frac{1}{2}\gamma_{dd} \log P_d^2 + \frac{1}{2}\gamma_{yy} \log y^2 + \\
\gamma_{we} \log P_w \log P_e + \gamma_{wl} \log P_w \log P_l + \gamma_{wc} \log P_w \log P_c + \gamma_{wd} \log P_w \log P_d + \\
\gamma_{wy} \log P_w \log P_y + \gamma_{el} \log P_e \log P_l + \gamma_{ec} \log P_e \log P_c + \gamma_{ed} \log P_e \log P_d + \\
\gamma_{ey} \log P_e \log y + \gamma_{lc} \log P_l \log P_c + \gamma_{ld} \log P_l \log P_d + \gamma_{cy} \log P_c \log y + \\
\gamma_{cd} \log P_c \log P_d + \gamma_{dy} \log P_d \log y
\end{aligned}$$

The log(total cost) equation includes the log levels of each factor, the level of output, their squares, and their cross-products. Given differentiability of the cost function, the cost shares of all inputs can be expressed as elasticities of the cost function with respect to the input prices. Using Shephard's lemma (McFadden 1978), the cost share equations can be derived by logarithmic differentiation of the cost function:

st0160

$$S_{it}(P_i, y) = \frac{\partial \log C}{\partial \log P_i} = \alpha_i + \sum_{j}^{n} \gamma_{ij} \log P_j + \gamma_{iy} \log y \tag{1}$$

Although there are five factors of production, the adding-up conditions across factors imply that the residual covariance matrix assembled by `sureg` would be rank deficient. As discussed in Greene (2008, 278), consistent estimates can be derived by estimating any four of the five factor share equations in the system, because they sum up to unity. The `isure` option (iterated seemingly unrelated regression) ensures that the resulting estimates are insensitive to the choice of the omitted factor share.

To set up the estimation, imposing adding-up constraints and constraints of symmetry on the matrix of second partial derivatives, we use the following commands:

```
global eq1 ("lnTC lnw lne lnl lnc lnd lny lnww lnee lnll lncc lndd lnyy ///
   lnwe lnwl lnwc lnwd lnel lnec lned lnlc lnld lncd lnwy lney lnly lncy lndy")
global eq2 ("Sw lnw lne lnl lnc lnd lny")
global eq3 ("Se lnw lne lnl lnc lnd lny")
global eq4 ("Sl lnw lne lnl lnc lnd lny")
global eq5 ("Sc lnw lne lnl lnc lnd lny")
constraint define 1 [Sw]_cons+[Se]_cons+[Sl]_cons+[Sc]_cons+[lnTC]lnd=1
constraint define 2 [Sw]lnw+[Sw]lne+[Sw]lnl+[Sw]lnc+[lnTC]lnwd=0
constraint define 3 [Sw]lne+[Se]lne+[Se]lnl+[Se]lnc+[lnTC]lned=0
constraint define 4 [Sw]lnl+[Se]lnl+[Sl]lnl+[Sl]lnc+[lnTC]lnld=0
constraint define 5 [Sw]lnc+[Se]lnc+[Sl]lnc+[Sc]lnc+[lnTC]lncd=0
constraint define 6 [lnTC]lnwd+[lnTC]lned+[lnTC]lnld+[lnTC]lncd+[lnTC]lndd=0
constraint define 7 [Sw]lny+[Se]lny+[Sl]lny+[Sc]lny+[lnTC]lndy=0
constraint define 8 [Sw]lne=[Se]lnw
constraint define 9 [Sw]lnl=[Sl]lnw
constraint define 10 [Sw]lnc=[Sc]lnw
constraint define 11 [Se]lnl=[Sl]lne
constraint define 12 [Se]lnc=[Sc]lne
constraint define 13 [Sl]lnc=[Sc]lnl
constraint define 14 [Sw]lnw=[lnTC]lnww
constraint define 15 [Sw]lne=[lnTC]lnwe
constraint define 16 [Sw]lnl=[lnTC]lnwl
constraint define 17 [Sw]lnc=[lnTC]lnwc
constraint define 18 [Sw]lnd=[lnTC]lnwd
constraint define 19 [Sw]_cons=[lnTC]lnw
constraint define 20 [Se]lne=[lnTC]lnee
constraint define 21 [Se]lnl=[lnTC]lnel
constraint define 22 [Se]lnc=[lnTC]lnec
constraint define 23 [Se]lnd=[lnTC]lned
constraint define 24 [Se]_cons=[lnTC]lne
constraint define 25 [Sl]lnl=[lnTC]lnll
constraint define 26 [Sl]lnc=[lnTC]lnlc
constraint define 27 [Sl]lnd=[lnTC]lnld
constraint define 28 [Sl]_cons=[lnTC]lnl
constraint define 29 [Sc]lnc=[lnTC]lncc
constraint define 30 [Sc]lnd=[lnTC]lncd
constraint define 31 [Sc]_cons=[lnTC]lnc
constraint define 32 [Sw]lny=[lnTC]lnwy
constraint define 33 [Se]lny=[lnTC]lney
constraint define 34 [Sl]lny=[lnTC]lnly
constraint define 35 [Sc]lny=[lnTC]lncy
sureg $eq1 $eq2 $eq3 $eq4 $eq5, constr(1-35) isure nolog small
  (output omitted)
```

Because the cost shares, S_i, are themselves derivatives of $\log C$ with respect to $\log P_i$, the parameters of the cost share equations are second derivatives of the cost function and can be assembled into a Hessian matrix of second partial derivatives. For concavity, the Hessian must be negative (semi)definite. Diewert and Wales (1987) show that this condition will be satisfied if and only if the matrix $\mathbf{M}$ is negative (semi)definite, where $\mathbf{M}$ is defined as

$$\mathbf{M} = H - S^k + SS'$$

where S is defined in (1) and S^k is a $k \times k$ diagonal matrix of shares. A real symmetric matrix will have negative eigenvalues if it is negative definite and nonpositive eigenvalues if it is negative semidefinite.

For the problem at hand, matrix $\mathbf{M}$ can be computed as follows:

$$\begin{bmatrix} \gamma_{ww} + S_w^2 - S_w & & & & \\ \gamma_{we} + S_w S_e & \gamma_{ee} + S_e^2 - S_e & & & \\ \gamma_{wl} + S_w S_l & \gamma_{el} + S_e S_l & \gamma_{ll} + S_l^2 - S_l & & \\ \gamma_{wc} + S_w S_c & \gamma_{ec} + S_e S_c & \gamma_{lc} + S_l S_c & \gamma_{cc} + S_c^2 - S_c & \\ \gamma_{wd} + S_w S_d & \gamma_{ed} + S_e S_d & \gamma_{ld} + S_l S_d & \gamma_{cd} + S_c S_d & \gamma_{dd} + S_d^2 - S_d \end{bmatrix}$$

Using the coefficients' point estimates available in `ereturn list` computed from 45 time-series observations, we can calculate the elements of matrix $\mathbf{M}$ for each time period:

```
generate double h11 = _b[lnTC:lnww] + Sw^2 - Sw
generate double h21 = _b[lnTC:lnwe] + Sw * Se
generate double h31 = _b[lnTC:lnwl] + Sw * Sl
generate double h41 = _b[lnTC:lnwc] + Sw * Sc
generate double h51 = _b[lnTC:lnwd] + Sw * Sd
generate double h22 = _b[lnTC:lnee] + Se^2 - Se
generate double h32 = _b[lnTC:lnel] + Se * Sl
generate double h42 = _b[lnTC:lnec] + Se * Sc
generate double h52 = _b[lnTC:lned] + Se * Sd
generate double h33 = _b[lnTC:lnll] + Sl^2 - Sl
generate double h43 = _b[lnTC:lnlc] + Sl * Sc
generate double h53 = _b[lnTC:lnld] + Sl * Sd
generate double h44 = _b[lnTC:lncc] + Sc^2 - Sc
generate double h54 = _b[lnTC:lncd] + Sc * Sd
generate double h55 = _b[lnTC:lndd] + Sd^2 - Sd
mkmat h*, mat(M)
```

This creates matrix $\mathbf{M}$ as $T \times 15$, where T is the number of observations in the sample and there are 15 elements in the lower triangle of the matrix for a specific period. We now use Mata to calculate the eigenvalues of the matrix for each time period:

```
mata:
M = st_matrix("M")'
lambda = J(cols(M), 5, .)
for(t=1; t<=cols(M); t++) {
        mt = invvech(M[., t])
        lambda[t, .] = symeigenvalues(mt)
}
lambda
end
```

Each row of matrix `lambda` contains the five eigenvalues of **M** for that time period:[1]

```
. mata: mm_matlist(lambda, "%9.5f")
                 1           2           3           4           5
     ┌───────────────────────────────────────────────────────────────┐
   1 │     0.13259     0.01068     0.00000    -0.00288    -0.06144   │
   2 │     0.10441     0.01027     0.00000    -0.00155    -0.05348   │
   3 │     0.10709     0.01655    -0.00000    -0.00247    -0.03358   │
   4 │     0.14514     0.00823    -0.00000    -0.00418    -0.03001   │
   5 │     0.10316     0.01127    -0.00000    -0.00150    -0.05334   │
   6 │     0.13224     0.01689    -0.00000    -0.00401    -0.02283   │
   7 │     0.21684     0.01486    -0.00000    -0.00653    -0.00872   │
   8 │     0.16139     0.01703    -0.00000    -0.00524    -0.01880   │
   9 │     0.17416     0.00939     0.00000    -0.00516    -0.03479   │
  10 │     0.12426     0.01145    -0.00000    -0.00339    -0.02840   │
  11 │     0.12944     0.01133    -0.00000    -0.00341    -0.03716   │
  12 │     0.13342     0.00035     0.00000    -0.00388    -0.01101   │
  13 │     0.21015    -0.00000    -0.00135    -0.00489    -0.01590   │
  14 │     0.16252     0.01153     0.00000    -0.00391    -0.02644   │
  15 │     0.14300     0.01701     0.00000    -0.00333    -0.02644   │
  16 │     0.12358     0.01396     0.00000    -0.00240    -0.02924   │
  17 │     0.15912     0.00914    -0.00000    -0.00324    -0.03468   │
  18 │     0.10364     0.01426    -0.00000    -0.00096    -0.03496   │
  19 │     0.14656     0.01169     0.00000    -0.00288    -0.03357   │
  20 │     0.13079     0.00000    -0.00070    -0.00126    -0.03093   │
  21 │     0.09045     0.00195     0.00000    -0.00147    -0.05131   │
  22 │     0.15502     0.00835     0.00000    -0.00242    -0.02290   │
  23 │     0.12108    -0.00000    -0.00010    -0.00087    -0.02347   │
  24 │     0.21675     0.00000    -0.00009    -0.00389    -0.00497   │
  25 │     0.17269     0.00972     0.00000    -0.00663    -0.02061   │
  26 │     0.15330     0.00000    -0.00541    -0.00736    -0.01757   │
  27 │     0.14008     0.00215    -0.00000    -0.00522    -0.01880   │
  28 │     0.15688     0.00500    -0.00000    -0.00601    -0.01692   │
  29 │     0.16513    -0.00000    -0.00518    -0.00675    -0.03441   │
  30 │     0.13266    -0.00000    -0.00345    -0.00481    -0.04113   │
  31 │     0.15934     0.00000    -0.00026    -0.00590    -0.04476   │
  32 │     0.16523    -0.00000    -0.00512    -0.01528    -0.03634   │
  33 │     0.14735    -0.00000    -0.00474    -0.01327    -0.02475   │
  34 │     0.16027    -0.00000    -0.00495    -0.01862    -0.03200   │
  35 │     0.15523    -0.00000    -0.00516    -0.00763    -0.01635   │
  36 │     0.18057     0.00266    -0.00000    -0.00269    -0.00712   │
  37 │     0.29339     0.01262     0.00365     0.00171     0.00000   │
  38 │     0.15251     0.00572     0.00000    -0.01060    -0.01939   │
  39 │     0.21218     0.00466    -0.00000    -0.00716    -0.01275   │
  40 │     0.19018     0.00507    -0.00000    -0.00293    -0.01339   │
  41 │     0.18717     0.00527     0.00052     0.00000    -0.01114   │
  42 │     0.20440     0.00533     0.00146    -0.00000    -0.02895   │
  43 │     0.17757     0.00648     0.00324     0.00000    -0.03411   │
  44 │     0.24303     0.00480    -0.00000    -0.00217    -0.03588   │
  45 │     0.18789     0.00551    -0.00000    -0.00134    -0.02136   │
     └───────────────────────────────────────────────────────────────┘
```

The presence of positive eigenvalues indicates that the assumption of concavity of the cost function is not satisfied by the estimates. To comply with economic theory, either another functional form has to be chosen or necessary concavity conditions must be imposed prior to estimation by restricting matrix **M** to be negative semidefinite.

---

[1] We use Ben Jann's `mm_matlist()` function from the `moremata` Statistical Software Components package to produce formatted matrix output.

However, in the case of the translog function, global imposition of concavity will destroy its flexibility and it should instead be imposed at a chosen reference point. According to Ryan and Wales (2000), the imposition of concavity at a single point will often lead to satisfaction of concavity at most or all data points in the sample. This can again be checked with the method described above.

References

Berndt, E. R., and L. R. Christensen. 1973. The translog function and the substitution of equipment, structures, and labor in U.S. manufacturing 1929–68. *Journal of Econometrics* 1: 81–113.

Diewert, W. E., and T. J. Wales. 1987. Flexible functional forms and global curvature conditions. *Econometrica* 55: 43–68.

Greene, W. H. 2008. *Econometric Analysis*. 6th ed. Upper Saddle River, NJ: Prentice Hall.

McFadden, D. 1978. Part I.1: Cost, revenue, and profit functions. In *Contributions to Economic Analysis*, ed. J. Tinbergen, D. W. Jorgenson, and J. Waelbroeck. Amsterdam: North-Holland.

Ryan, D. L., and T. J. Wales. 2000. Imposing local concavity in the translog and generalized Leontief cost functions. *Economics Letters* 67: 253–260.