Polynomial Distributed Lags

by

R.E. Hall

Number 7 - July 28, 1967

Econometricks Working Paper # 3

R.E. Hall, July 28, 1967

Revised October 5, 1967

Polynomial Distributed Lags

In general, a distributed lag model can be written

$$(1) \qquad y_t = \sum_{\tau=0}^{p-1} \beta_\tau x_{t-\tau} \qquad ;$$

y_t and x_t are time series and β_τ are the coefficients of the lag function. Often the number of periods, p, covered by the lag function is so large that the individual coefficients β_τ cannot be estimated with sufficient accuracy. In this case we usually seek to estimate the coefficients subject to some restrictive hypothesis _a priori_. The best-known hypothesis is that of Koyck and many others, requiring that β_τ have the special form

$$(2) \qquad \beta_\tau = \lambda^\tau \qquad .$$

A great deal of effort has been devoted to the econometric aspects of estimating Koyck distributed lags, without achieving any general agreement about how to estimate them. The purpose of this paper is to describe an alternative lag specification which is both more flexible and easier to estimate. Credit for the discovery and introduction of this method goes to Shirley Almon (_1_); this paper merely restates her method (with some modifications due to Charles Bischoff (_2_)) in a somewhat simpler form, and goes on to describe the computer implementation of the method.

The basic hypothesis of the method is that the lag distribution β_τ is a smooth function of the lag τ. Our interpretation of the notion of smoothness is that the function can be approximated closely by a polynomial of fairly low order; that is, we suppose that

$$(3) \qquad \beta_\tau = \alpha_1 + \alpha_2\tau + \alpha_3\tau^2 + \ldots + \alpha_N\tau^{N-1} \quad ,$$

where N is usually less than 6. Polynomial approximations of this kind have a long history in numerical analysis, but the methods of polynomial interpolation developed in connection with them do not appear, in retrospect, to have any usefulness in approximating distributed lags. It is Mrs. Almon's use of Lagrangian interpolation polynomials rather than ordinary polynomials that makes her presentation of the method somewhat obscure. The simpler approach of the present paper is formally equivalent to Mrs. Almon's approach.

The polynomial approximation gives rise to a straightforward linear estimation problem. We begin by substituting equation (3) into (1):

$$(4) \qquad y_t = \sum_{\tau=0}^{p-1} (\alpha_1 + \alpha_2\tau + \ldots + \alpha_N\tau^{N-1})x_{t-\tau}$$

$$= \alpha_1 \left(\sum_{\tau=0}^{p-1} x_{t-\tau}\right) + \alpha_2 \left(\sum_{\tau=0}^{p-1} \tau x_{t-\tau}\right) + \ldots$$

$$+ \alpha_N \left(\sum_{\tau=0}^{p-1} \tau^{N-1} x_{t-\tau}\right) \quad .$$

By defining new variables $z_{t,j}$, which are moving averages of the original variables, as follows:

$$(5) \qquad z_{t,j} = \sum_{\tau=0}^{p-1} \tau^{j-1} x_{t-\tau} \quad ,$$

we have a linear model of ordinary form:

$$(6) \qquad y_t = \alpha_1 z_{t,1} + \alpha_2 z_{t,2} + \dots + \alpha_N z_{t,N-1} \quad .$$

All estimation methods which are appropriate for linear equations are available to the investigator of distributed lags if the method of polynomial approximation is used.

In practice, the method proceeds as follows. First, a polynomial weighting matrix, A, is generated; it is convenient to normalize it so that the lag interval lies between $\frac{1}{p+1}$ and $\frac{p}{p+1}$ (instead of between 0 and p-1) -- this is done by taking powers of $\frac{\tau+1}{p+1}$ instead of powers of τ . The matrix A has N columns and p rows ; each column gives the weights applied to the lagged x's in generating one z-variable while each row gives the weights applied to the estimates of α in calculating the lag function β .

Next the z-variables are generated from A and x using the product relation

$$(7)\qquad \left[z_{t,1} \cdots z_{t,N} \right] = \left[x_t, x_{t-1}, \ldots, x_{t-p+1} \right] \cdot \begin{bmatrix} 1 & \frac{1}{p+1} & \left(\frac{1}{p+1}\right)^2 & \cdots & \left(\frac{1}{p+1}\right)^{N-1} \\ 1 & \frac{2}{p+1} & \left(\frac{2}{p+1}\right)^2 & \cdots & \left(\frac{2}{p+1}\right)^{N-1} \\ \vdots & & & & \\ 1 & \frac{p}{p+1} & \left(\frac{p}{p+1}\right)^2 & \cdots & \left(\frac{p}{p+1}\right)^{N-1} \end{bmatrix} .$$

or,

$$(8)\qquad z_t = \bar{x}_t A \quad .$$

This process is called "scrambling" at MIT.

The next step is to obtain estimates α of the coefficients of the z-variables by whatever method is appropriate for the stochastic specification chosen for the model. Finally, estimates $\hat{\beta}$ of the lag coefficients can be obtained from $\hat{\alpha}$ using the relation

$$(9)\qquad \hat{\beta} = A\hat{\alpha} \quad .$$

An estimate of the variance-covariance matrix $V(\hat{\beta})$ can be calculated as

$$(10)\qquad V(\hat{\beta}) = A\, V(\hat{\alpha})\, A' \quad .$$

Two additional statistics may be of interest. These are s, the sum of the lag coefficients, and μ , the mean lag. If u denotes a vector of p 1's, we can calculate s from

$$(11) \qquad s = u'\hat{\beta} = u'A\hat{\alpha} \qquad ,$$

and its variance from

$$(12) \qquad V(s) = u'AV(\hat{\alpha})A'u \qquad .$$

Second, if we define the vector v by

$$(13) \qquad v = \begin{bmatrix} 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ p-1 \end{bmatrix} \qquad ,$$

then

$$(14) \qquad \mu = \frac{1}{s} v'\beta \qquad .$$

Its asymptotic variance is

$$(15) \qquad V(\mu) = \left(\frac{1}{s} v' - \frac{1}{s^2} u' \right) AV(\hat{\alpha}) A' \left(\frac{1}{s} v - \frac{1}{s} v - \frac{1}{s^2} u \right) \qquad .$$

The process of calculating the lag coefficients and these statistics is called "unscrambling" at MIT.

We turn now to variations of this basic method; these take the form of what Bischoff (2) calls zero restrictions. A zero restriction is used to impose *a priori* the hypothesis that the lag distribution approaches zero at one or both ends. If the method of polynomial approximation is used in estimating the lag distribution, zero restrictions are imposed by limiting the components $z_{t,j}$ to those corresponding to polynomials which meet the restrictions.

Now in normalized form, the basic polynomial lag function (3) is

$$(16) \qquad \beta_\tau = \alpha_1 + \alpha_2 \left(\frac{\tau+1}{p+1}\right) + \alpha_3 \left(\frac{\tau+1}{p+1}\right)^2 + \dots + \alpha_N \left(\frac{\tau+1}{p+1}\right)^{N-1} .$$

If a zero restriction is imposed at the near end, formula (16) is modified by eliminating the constant term:

$$(17) \qquad \beta_\tau = \alpha_1 \left(\frac{\tau+1}{p+1}\right) + \alpha_2 \left(\frac{\tau+1}{p+1}\right)^2 + \dots + \alpha_{N-1} \left(\frac{\tau+1}{p+1}\right)^{N-1} .$$

This form of the distributed lag function always has small coefficients for the shortest lags. The name "zero restriction" is derived from the fact that if a hypothetical β_{-1} were calculated from formula (17), it would be zero no matter what values the α-coefficients had.

A zero restriction is imposed at the far end in a similar way. Instead of formula (16), we use

$$(18) \quad \beta_\tau = \alpha_1 \left[1 - \frac{\tau+1}{p+1}\right] + \alpha_2 \left[\left(\frac{\tau+1}{p+1}\right)^2 - \frac{\tau+1}{p+1}\right] + \ldots + \alpha_{N-1} \left[\left(\frac{\tau+1}{p+1}\right)^{N-1} - \frac{\tau+1}{p+1}\right] .$$

In this case, a hypothetical β_p is always zero, so that the lag function is constrained to be close to zero for the longest lags.

Finally, zero restrictions may be imposed at both ends by dropping the first term from equation (18):

$$(19) \quad \beta_\tau = \alpha_1 \left[\left(\frac{\tau+1}{p+1}\right)^2 - \frac{\tau+1}{p+1}\right] + \alpha_2 \left[\left(\frac{\tau+1}{p+1}\right)^3 - \frac{\tau+1}{p+1}\right] + \ldots + \alpha_{N-2} \left[\left(\frac{\tau+1}{p+1}\right)^{N-1} - \frac{\tau+1}{p+1}\right]$$

Equation (19) is the form which Mrs. Almon proposed.

Note that all three kinds of zero restrictions are simply modifications of the weighting matrix A -- all of the formulas given earlier for the unscrambling phase still hold for the modified versions of A.

Programs for Polynomial Distributed Lags

1. AMAT

This routine generates the A-matrix.

Calling sequence: CALL AMAT(NPER, NDEG, JZERO,A)

NPER	Number of periods, p.
NDEG	Number of terms in approximating polynomial. On input it should be the highest power of $\frac{\tau+1}{p-1}$ plus one. On return it will be reduced to take account of zero restrictions so that it is equal to the number of columns in A.
JZERO	Zero restriction code: 1 for both. 2 for far only. 3 for near only. 4 for neither.
A	A-matrix, packed by columns; length = NPER*NDEG.

2. SCRAMB

This routine generates the z-variables, given the matrix A and the variable x.

Calling sequence: CALL SCRAMB(NOBX,NPER,NDEG,A,X,Z)

NOBX	Number of observations in X. Number of observations in Z is NOBX - NPER.

NPER	Number of periods, p.
NDEG	Degree as returned by AMAT.
A	A-matrix.
X	Input vector; length = NOBX.
Z	Output matrix. Length = (NOBX - NPER)*NDEG.

3. UNSCRM

This routine calculates various statistics which are useful in interpreting distributed lag estimates.

Calling sequence: CALL UNSCRM(NPER,NDEG,NOV,LOC,DMEAN,SMEAN,SUM,SSUM,B,D,
V,VS,A,S,W)

NPER	Number of periods, p.
NDEG	Degree as returned by AMAT.
NOV	Total number of variables in regression including those not associated with the distributed lag.
LOC	Position in B of the set of estimates for the distributed lag being unscrambled.
DMEAN	Mean lag.
SMEAN	Standard error of DMEAN.
SUM	Sum of lag coefficients.
SSUM	Standard error of sum of lag coefficients.
B	Vector of regression estimates. Length = NOV.
D	Vector of estimates of the distributed lag. Length = NPER.

V	Estimate of variance-covariance matrix of B. Length = NOV**2.
VS	Variance-covariance matrix of estimates of α for this distributed lag, extracted from V by the program. Length = NDEG**2.
A	Polynomial weighting matrix A. Length = NPER*NDEG.
S	Vector of standard errors of D. Length = NPER.
W	Scratch vector. Length = NPER.

References

(1) S. Almon, "The Distributed Lag Between Capital Appropriations and Expenditures" Econometrica 33: pp.178-196, January 1965

(2) C.W. Bischoff, MIT Ph.D. Dissertation, September 1967