An Algorithm for Distributed Lag Estimation Subject to Piecewise Monotonic Coefficients

I. C. Demetriou and E. E. Vassiliou *

Abstract—**Linearly distributed-lag models as a time series tool have very useful applications in many disciplines. In these models, the dependent variable depends on one independent variable and its lags. The specification of the lag coefficients is a crucial question to the efficacy of a model. A new algorithm is proposed for the estimation of lag coefficients subject to the condition that the sequence of the coefficient estimates consists of a certain number of monotonic sections, where the positions of the extrema are also unknowns. The algorithm is iterative, each iteration taking a conjugate gradient step, then forming an estimate of the coefficients and finally adjusting this estimate to satisfy the given constraints. An immediate advantage is that the inversion of an ill-conditioned matrix that frequently occurs in practice is avoided. Moreover, the constraints provide a realistic representation of the prior knowledge and the calculation results in a highly efficient time series estimation. The algorithm and its convergence are described, results from simulation experiments are presented and an application of the algorithm on real annual macroeconomic data concerning the personal consumption expenditures against the GDP for the U.S.A. during 1929 - 2006 is given.**

Keywords: approximation, conjugate gradient, consumption, distributed lag model, piecewise monotonic, regression, smoothing, time series

1 Introduction

The purpose of distributed-lag models is to estimate, from time series data, values y that incorporate prior information of the independent variable x. These models have useful applications in many fields such as econometrics, engineering (see, for instance, [8], [11], [15], [21], [25]) etc. For example, in econometrics, if y_t denotes consumption expenditures and x_t income, at time period t, a change in x_t will affect not only current consumer expenditures y_t, but also future expenditures y_{t+1}, y_{t+2}, etc. Therefore we assume that y_t depends not only on x_t but also on q past values of x_t, giving the linearly distributed-lag model

$$y_t = \sum_{i=0}^{q} \beta_i x_{t-i} + \epsilon_t, \tag{1}$$

where q is a prescribed positive number representing the lag length, $\{\beta_i : i = 0, 1, \ldots, q\}$ are the unknown lag coefficients and ϵ_t is a random variable with zero mean and constant variance. The issue of the q selection depends on the data and may be decided with statistical means (see, for example, [17]:p.119). Adopting matrix notation, the unconstrained lag-distribution problem is to determine a vector $\beta = (\beta_0, \beta_1, \ldots, \beta_q)^T$ that minimizes the objective function

$$F(\beta) = (y - X\beta)^T (y - X\beta), \tag{2}$$

where $y = (y_{q+1}, y_{q+2}, \ldots, y_{q+n})^T$ is the n-vector whose components are time series observations and the $n\times(q+1)$ matrix X of current and lagged values of x_t is defined as

$$X = \begin{pmatrix} x_{q+1} & x_q & x_{q-1} & \cdots & x_1 \\ x_{q+2} & x_{q+1} & x_q & \cdots & x_2 \\ x_{q+3} & x_{q+2} & x_{q+1} & \cdots & x_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{q+n} & x_{q+n-1} & x_{q+n-2} & \cdots & x_n \end{pmatrix}.$$

Note that the components of y in (2) correspond to the last n observations of the time series data $y_t, t = 1, 2, \ldots, q, q+1, \ldots, y_{q+n}$, because we lose q degrees of freedom due to (1).

The unconstrained estimate of β, for a full rank X, is

$$\tilde{\beta} = (X^T X)^{-1} X^T y. \tag{3}$$

The main drawback with this direct least-squares estimation of β is that often there is high multicollinearity among the x_t's giving a notoriously ill-posed inverse problem, which results in imprecise estimation for the β. If, however, avoid severe distortions in the calculation of the true lag distribution, then there appear discernible patterns in the unconstrained estimate, which are affected by the nature of the observations.

So far there have been several suggestions in the literature to put some structure on the β_i's in (1). They all impose some a priori structure on the form of the lag,

*Manuscript received January 2, 2009. I. C. Demetriou is grateful to the University of Athens for research grand ELKE 70/4/4729.
Unit of Mathematics and Informatics, Department of Economics, University of Athens, 8 Pesmazoglou street, Athens 10559, Greece. Email: demetri@econ.uoa.gr, evagvasil@econ.uoa.gr

in order to combine prior and sample information in the estimation of the regression coefficients. A popular approach is the Almon polynomial lag distribution [1]. In this technique the $q+1$ coefficients of the lagged variables are assumed to lie on a polynomial whose order is predetermined. Shiller's method [23], as a variant of this model, assumes that the coefficients of the lagged variable lie close to, rather than on, a polynomial. More models are found in [6], [13], [14], [18], [19], [21], [24], etc. All these models assume rather arbitrarily that the underlying function of the lag coefficients can be approximated closely by a form that depends on a few parameters. However, over the years, literature on the subject agrees that some weak representation of the lag coefficients is a sensible requirement for a satisfactory model estimation (see, for example, [13], [21] and references therein).

In [5], we proposed a new approach to lag coefficients estimation by assuming that the lag coefficients $\beta_0, \beta_1, \ldots, \beta_q$ have at most k monotonic sections, where k is a prescribed positive number. An advantage of this approach to lag-coefficients estimation is that we impose conditions to the coefficients that give properties that occur to a wide range of underlying models. The user may try several values of k if a particular choice does not suggest itself. The conditions on $\beta_0, \beta_1, \ldots, \beta_q$ avoid any parameterization and provide a rather weak though systematic representation of the prior knowledge, as we are going to explain in Section 3.

In the case when $k = 1$ the calculation is as follows. Minimize the function (2) subject to

$$\beta_0 \geq \beta_1 \geq \cdots \geq \beta_q, \tag{4}$$

if we require monotonically decreasing coefficients, and

$$\beta_0 \leq \beta_1 \leq \cdots \leq \beta_q, \tag{5}$$

if we require monotonically increasing coefficients. Since the constraints on $\beta_0, \beta_1, \ldots, \beta_q$ are linear and consistent and since the second derivative matrix of (2) with respect to β is twice the matrix $X^T X$, the calculation of β is a convex quadratic programming problem. Given that $X^T X$ is positive semidefinite, there is a global solution and if $X^T X$ is positive definite the solution is unique. In this paper, it is assumed that $X^T X$ is positive definite. Thus several general algorithms are available for obtaining the solution (see, for example, [7]). Further, it is worth mentioning that the problem subject to the monotonic decreasing constraints (4) generalizes the method of Fisher [6], where the coefficients β_i are imposed to decline arithmetically.

When $k > 1$ the piecewise monotonicity constraints are (see [4])

$$\left.\begin{array}{ll} \beta_{t_{m-1}} \leq \beta_{t_{m-1}+1} \leq \cdots \leq \beta_{t_m}, & \text{if } m \text{ is odd} \\ \beta_{t_{m-1}} \geq \beta_{t_{m-1}+1} \geq \cdots \geq \beta_{t_m}, & \text{if } m \text{ is even} \end{array}\right\}, \tag{6}$$

while the integers $\{t_m : m = 0, 1, \ldots, k\}$ satisfy the conditions

$$0 = t_0 \leq t_1 \leq \cdots \leq t_k = q. \tag{7}$$

The integers $\{t_m : m = 1, 2, \ldots, k-1\}$, namely the indices of the turning points of the estimated components of β, are not known in advance and they are variables in the optimization calculation. This raises the number of combinations of integer variables to about $O(q^k)$. Therefore it is usually quite difficult to develop efficient optimization algorithms for obtaining an optimal β by minimizing (2) subject to (6), because the combinatorial nature of the constraints defines a nonconvex calculation with very many local minima. However, we address an alternative form of the problem and develop an iterative algorithm that implements a conjugate gradient method with the piecewise monotonicity constraints on the lag coefficients, which attempts to minimize (2).

The iterative algorithm and its convergence are presented in Section 2. The piecewise monotonicity problem in distributed lag modeling is discussed in Section 3. Some numerical results from a simulation that demonstrate the performance of the method are presented in Section 4. An example of an application of our method on real data is presented in Section 5. Some concluding remarks are given in Section 6. The Fortran program that implements our algorithm when $k > 1$ for distributed-lag estimation consists of about 1400 lines including comments, which gives an idea of the size of the required calculation.

2 The algorithm and its convergence

We develop an algorithm that processes the lag coefficients iteratively. It starts from an initial estimate $\beta^{(0)}$ of β that satisfies the constraints and generates a sequence of estimates $\{\beta^{(j)} : j = 1, 2, 3, \ldots\}$ to β in two phases. In the first phase it takes a step from the current estimate to a new estimate of β by applying the Fletcher-Reeves version of the conjugate gradient algorithm with exact line searches as described by [7], for instance. In the second phase it conveys "prior knowledge" to the calculation through the replacement of the new estimate by its best piecewise monotonic approximation. The contraction mapping theorem is used as a basis for establishing convergence.

In the first phase specifically, the algorithm calculates a new estimate of the form

$$\beta^{(j+1)} = \beta^{(j)} + \alpha_j d^{(j)}, \tag{8}$$

where α_j is a step-length and $d^{(j)}$ is the search direction

$$d^{(j)} = -X^T(y - X\beta^{(j)}) + \gamma_j d^{(j-1)}, \tag{9}$$

except that the last term is omitted if $j = 1$. The value of γ_j is determined by the Fletcher-Reeves conjugacy condition. The step-length α_j is calculated to minimize the quadratic function of one variable $F(\beta^{(j)}+\alpha d^{(j)})$. Besides the advantages that we derive from the conjugate gradient algorithm (see, for example, [7], [10]), it is important that this calculation involves matrix X only multiplicatively, so ill-conditioning of X is irrelevant here.

Having calculated $\beta^{(j+1)}$, the algorithm proceeds to the second phase, which calculates a $(q+1)$-vector β that minimizes the function

$$g(\beta_0,\beta_1,\ldots,\beta_q)=\sum_{i=0}^{q}(\beta_i^{(j+1)}-\beta_i)^2 \qquad (10)$$

subject to the piecewise monotonicity constraints (6), where the integer variables $\{t_m : m=0,1,\ldots,k\}$ satisfy the conditions (7). Despite its formidable $O(q^k)$ complexity, this problem is solved by [4] in only $O(q^2+kq\log_2 q)$ computer operations, but in this paper we employ a $O(kq^2)$ version of [2] that in practice seems to require only about $O(kq)$ operations. The main properties of the problem, which provide this excellent complexity are presented in Section 3.

The algorithm finishes if the vector β found at the second phase satisfies the convergence condition

$$\|\beta-\beta^{(j)}\|_2/\|\beta\|_2 \le \varepsilon, \qquad (11)$$

where ε is a small positive tolerance. This test is applied at every estimate $\beta^{(j+1)}$ including the first iteration as well. When the test (11) fails, then the algorithm replaces $\beta^{(j+1)}$ by its best piecewise monotonic approximation vector β, increases j by one and branches to the beginning of the first phase in order to calculate at least one new vector in the sequence $\{\beta^{(j)} : j=1,2,3,\ldots\}$. Therefore the convergence test is applied at consecutive best piecewise monotonic approximations to the corresponding estimates produced by the conjugate gradient iteration. We present an outline of the algorithm discussed so far.

Algorithm 1 $(k>1)$
Step 0 Set $j=0$, $\beta^{(0)}=0$ and $\gamma_0=0$.
Step 1 Calculate $d^{(j)}=-X^T(y-X\beta^{(j)})+\gamma_j d^{(j-1)}$.
Step 2 Calculate α_j and set $\beta^{(j+1)}=\beta^{(j)}+\alpha_j d^{(j)}$.
Step 3 By employing Algorithm 2 of [2] calculate β, namely a least squares approximation with k monotonic sections to $\beta^{(j+1)}$.
Step 4 If criterion (11) is satisfied then quit, otherwise replace $\beta^{(j+1)}$ by β, calculate γ_j, increase j by one and go to Step 1. ■

We show that Algorithm 1 terminates in a finite number of iterations.

Theorem 1 Algorithm 1 meets the termination condition (11) for some finite integer j.

Proof At Step 0 the starting vector $\beta^{(0)}=0$ is not restrictive. If a more appropriate initial guess to β that satisfies the constraints (6) is available, then set this guess to $\beta^{(0)}$. Step 1 calculates the search direction $d^{(j)}$. Step 2 calculates the step-length α_j and obtains the estimate $\beta^{(j+1)}$. Step 3 calculates β, namely a least squares approximation with k monotonic sections to $\beta^{(j+1)}$. The algorithm either terminates at Step 4 or it sets β to $\beta^{(j+1)}$, calculates the parameter γ_j and then branches to Step 1. If we drop Step 3, which provides the piecewise monotonic approximation, the remaining steps provide a conjugate gradient iteration, so in this case the algorithm terminates at the minimum of (2). In view of [20]:p. 180, conjugate gradient can be regarded as a generalization of the steepest descent method. Since steepest descent is a contractive operator (see, [9]:p. 29) and the piecewise monotonic algorithm for a fixed sequence of $\{t_m : m=0,1,\ldots,k\}$ is a norm reducing operator (see, [22]:p. 376), we have a sufficient condition (see, for example, [16]) for the convergence of Algorithm 1. ■

However, it is hard to determine the convergence rate of Algorithm 1, because of the nonlinearity of the piecewise monotonic procedure.

3 The piecewise monotonicity model

In this section we discuss some properties of the piecewise monotonicity model that is employed by Step 3 of Algorithm 1. It seems appropriate to begin by noticing that the calculation of the unconstrained minimum of (2) due to (3) is highly ill-conditioned and that the unconstrained minimum allows so much freedom in the calculation of β, that model (1) is almost useless in any estimation process.

We, instead, take the view that the calculation should make the smallest change to the current estimate of β that is necessary to satisfy constraints (6). The rationale for this choice is as follows. The sequence of the lag coefficients $\{\beta_i^{(j)} : i=0,1,\ldots,q\}$ may be attended as measurements of an unknown function. Due to errors of measurement in the time series data $y_t, t=1,2,\ldots,y_{q+n}$, it is possible that the number of turning points in $\beta_i^{(j)}$'s is substantially larger than the the number of turning points in the true function values. Then the number of turning points in $\beta_i^{(j)}$'s may suggest that it would be advantageous to smooth these estimates by requiring a certain number of monotonic sections. Therefore, given a positive integer $k<q$, we seek a $(q+1)$-vector β that is closest to $\beta^{(j)}$ in the least squares sense, subject to the condition that the components of β consist of at most k monotonic sections. By specifying that the first monotonic section is increasing, we obtain the constraints (6), but the user may well define it to be decreasing.

The following properties of the piecewise monotonic approximation problem are considered by [4]. The approximation process is a projection, because if $\beta^{(j)}$ satisfies the constraints then $\beta=\beta^{(j)}$. Therefore if $\beta^{(j)}$ consists of more than k monotonic sections, as it is usually expected in practice, then the constraints prevent the equation $\beta = \beta^{(j)}$, so $\{t_m : m = 1, 2, \ldots, k-1\}$ are all different. Further at the turning points of a best fit, the equations $\beta_{t_m} = \beta^{(j)}_{t_m}, m = 1, 2, \ldots, k-1$ hold, which directs the search for the t_m's among the indices of the local maxima (i.e. $\beta^{(j)}_{m-1} < \beta^{(j)}_m \geq \beta^{(j)}_{m+1}$) of $\beta^{(j)}$ if m is odd and among the indices of the local minima (i.e. $\beta^{(j)}_{m-1} > \beta^{(j)}_m \leq \beta^{(j)}_{m+1}$) of $\beta^{(j)}$ if m is even, which reduces the amount of required computation at least by a factor of four. The most important property, however, is that the monotonic sections in a best piecewise monotonic fit are distinct. Indeed, the components $\{\beta_i : i = t_{m-1}, t_{m-1}+1, \ldots, t_m\}$ on $[t_{m-1}, t_m]$ minimize the sum of the squares $\sum_{i=t_{m-1}}^{t_m} (\beta^{(j)}_i - \beta_i)^2$ subject to the constraints $\beta_i \leq \beta_{i+1}$, $i = t_{m-1}, \ldots, t_m - 1$, if m is odd, and subject to the constraints $\beta_i \geq \beta_{i+1}$, $i = t_{m-1}, \ldots, t_m - 1$, if m is even. In the former case the sequence $\{\beta_i : i = t_{m-1}, t_{m-1}+1, \ldots, t_m\}$ is the best monotonic increasing fit to $\{\beta^{(j)}_i : i = t_{m-1}, t_{m-1}+1, \ldots, t_m\}$ and on the latter case the best monotonic decreasing one. Therefore, provided that $\{t_m : m = 1, 2, \ldots, k-1\}$ are known, the components of β can be generated by solving a separate monotonic problem on each section $[t_{m-1}, t_m]$ in the cost of only $O(t_m - t_{m-1})$ computer operations. It follows that an optimal fit β associated with the integer variables $\{t_m : m = 1, 2, \ldots, k-1\}$ can split at t_{k-1} into two optimal sections. One section that provides a best fit on $[t_0, t_{k-1}]$, which in fact is similar to β with one monotonic section less, and one section on $[t_{k-1}, t_k]$ that is a single monotonic fit to the remaining data. Hence the optimization problem of phase two can be replaced by a problem, whose variables are the positions of the turning points of the required fit and which is solved by dynamic programming. The implementation of this idea includes several options that are considered by [4] and [2], while a versatile computer code has been written by [3].

4 Numerical results

This section presents results from simulation experiments in order to demonstrate the performance of Algorithm 1. The data were produced in two steps. First, the values x_t were chosen to be the daily U.S. Dollar/Euro Foreign Exchange Rate derived from the Board of Governors of the Federal Reserve System for the period 1/4/1999 - 5/8/2007, which amounts to 2099 observations. Second, each component y_t was generated from (1) after a function value $\phi(z_i)$ was substituted for β_i and a number from the uniform distribution on $[-r, r]$ was substituted for ϵ_t, where $r = 0.05, 0.1$, and

$$\phi(z) = \frac{\frac{9\pi}{2} - z}{10} \sin(z), \quad z \in [\pi/2, 9\pi/2]. \qquad (12)$$

Function (12) was chosen because it is a sine wave that quickly decays towards zero and its measurements appear particularly suitable for simulating possible seasonal effects of the lag coefficients. It has five monotonic sections. For the underlying function, we let the lag length q have the values 25, 50 and 100 and for each q the data points have the equally spaced values $\{z_i = \pi/2 + (4\pi i/q) : i = 0, 1, \ldots, q\}$. Of course the values of r were selected to provide substantial differences in the final form of the coefficients $\beta_i, i = 0, 1, \ldots, q$.

In the case when $k = 1$, vector β was obtained by minimizing the objective function (2) subject to the constraints (4). We have developed a special quadratic programming method for this problem that takes account of the fact that each of the constraint functions depends on only two adjacent components of β, but we do not enter into the details of our computation. In the cases when $k = 2, 3, 4$ and 5, vector β was obtained by employing Algorithm 1, where the first monotonic section in (6) was let to be decreasing.

The actual values of q, the convergence tolerance ε in (11), the number of monotonic sections k and the following list of calculated parameters are given in Tables 1 and 2, one for each value of r:

1. $S_{\phi\hat{\beta}} = (\sum_{i=1}^{m} (\phi(z_i) - \hat{\beta}_i)^2)^{1/2}$, the distance between the function values $\phi(z_i), i = 0, 1, \ldots, q$ and the estimated lag coefficients $\hat{\beta}_i, i = 0, 1, \ldots, q$.
2. $P_{RelError} = \max_{q+1 \leq i \leq q+n} |y_i - \underline{\hat{\beta}}^T \underline{\xi}^{(i-q)}| / (\max_{q+1 \leq i \leq q+n} y_i - \min_{q+1 \leq i \leq q+n} y_i) \times 100$, the percent relative error of the time series estimation, which relates the error to the scale of values taken by the data, where $\underline{\xi}^{(i-q)}$ is the $(i-q)$th column of matrix X.
3. The number of iterations required by Algorithm 1 for $k = 2, 3, 4$ and 5 to calculate the lag coefficients.
4. The CPU time in seconds to perform the calculations in single precision arithmetic using the standard Fortran 77 compiler of Compaq Visual Fortran 6.1 on a Personal Computer with an Intel 2.4 GHz processor operating in Microsoft Windows XP with 32 bits word length.

The parameter 1 requires the a-priori knowledge of the underlying function of the lag coefficients, so it can be used only for testing purposes. The parameter 2 is the actual time series smoothing quality indicator that the user has available at the end of the calculation. The parameters 3 and 4 present the computational effort of the method. A direct comparison of the number of iterations and the CPU time indicates the work required by a single iteration of Algorithm 1.

Each of the Tables 1 and 2 consists of a triplex of rows for the cases $q = 25, 50$ and 100. Each row-triplex presents

Table 1: Performance and CPU times when $r = 0.05$

q	ε	k	$S_{\phi\hat{\beta}}$	$P_{RelError}$	Iterations	CPU time (sec)
25		1	2.2833	26.3521	-	0.07
	10^{-3}	2	1.3342	24.4036	5	0.01
		3	0.9283	17.6173	5	0.00
		4	0.2166	11.1945	9	0.01
		5	0.4196	12.7486	7	0.03
	10^{-5}	2	1.3847	27.3186	26	0.04
		3	0.7618	15.1199	26	0.04
		4	0.5019	11.1966	28	0.06
		5	0.6264	11.6611	28	0.06
50		1	3.2477	34.5435	-	0.20
	10^{-3}	2	1.8148	32.2325	12	0.03
		3	0.7418	12.9000	12	0.03
		4	0.3555	6.7254	14	0.04
		5	0.3731	6.7840	14	0.06
	10^{-5}	2	1.8918	34.2661	45	0.12
		3	0.9533	14.1700	37	0.10
		4	0.6830	7.1592	41	0.10
		5	0.7258	7.1424	39	0.14
100		1	4.8319	42.7780	-	0.79
	10^{-3}	2	2.6640	43.0436	10	0.06
		3	1.0613	14.2788	10	0.07
		4	0.1483	3.2309	10	0.06
		5	0.3112	3.2183	17	0.10
	10^{-5}	2	2.7581	44.3071	55	0.31
		3	1.2796	14.9423	41	0.23
		4	0.6952	4.0038	46	0.26
		5	0.7301	3.4768	48	0.28

numerical results for $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-5}$ in order to show the performance of Algorithm 1 at different stages of the computation. For each q, a horizontal line separates the results when $k = 1$ from those when $k > 1$. Some smaller values of ε were tried too, but we do not consider them, because they made little difference to the solution.

Moving along the rows of each row-triplex, for a specific ε, we can see the individual performance of Algorithm 1 for each value of k. Moving down the columns of each row-triplex, for a specific ε, we can compare the performance of Algorithm 1 for different values of k. In all cases, the algorithm converged to the solution rapidly, in a number of iterations that seldom exceeded q. It is worthy of note that these results are by far better than those of [5] that implements an analogue version to Algorithm 1 with the steepest descent method.

The values of the coefficient error $S_{\phi\hat{\beta}}$ and the time series error reduction $P_{RelError}$, when $k = 1$, provided upper bounds to the corresponding parameters when $k > 1$. It seems that these parameters showed a tendency to decrease as k increased, because the calculated lag coefficients tend to follow the trend of the underlying function (12) for an appropriate value of k.

As a general remark, the calculated lag coefficients for $\varepsilon = 10^{-5}$ get closer to the unconstrained coefficients than the coefficients for $\varepsilon = 10^{-3}$. In contrast, the latter coefficients provide smoother estimates. This remark is illustrated in Figs. 1 and 2, one figure for each value of ε, where the calculated lag coefficients with $k = 4$, denoted by ($\circ$) on a continuous curve, do remarkably well in capturing the shape of the underlying function. The data for these figures are associated with the cases $q = 5$ and $k = 4$, for $\varepsilon = 10^{-3}, 10^{-5}$, of Table 2.

5 An example on consumption data

To illustrate our method we present an application on real annual macroeconomic data derived from the Bureau of Economic Analysis of the U.S. Department of Commerce for the period 1/1/1929 - 1/1/2006. The dependent variable is the Real Personal Consumption Expenditures (PCE) and the independent variable is the Real Gross Domestic Product (GDP) for U.S.A., both measured in billions of chained 2000 dollars. The data of our application are given explicitly in the relevant columns of

Table 2: Performance and CPU times when $r = 0.1$

q	ε	k	$S_{\phi\hat{\beta}}$	$P_{RelError}$	Iterations	CPU time (sec)
25		1	2.3327	31.9851	-	0.07
	10^{-3}	2	1.3688	26.6767	5	0.01
		3	1.0594	21.7691	5	0.00
		4	1.0274	21.5515	5	0.01
		5	1.0274	21.5515	5	0.01
	10^{-5}	2	1.6847	32.7452	42	0.07
		3	1.2848	21.7943	36	0.06
		4	1.3455	19.9306	36	0.07
		5	1.3460	20.0997	36	0.07
50		1	3.2624	35.6179	-	0.23
	10^{-3}	2	1.8262	35.7775	12	0.04
		3	0.7780	16.1560	12	0.04
		4	0.2296	11.0936	12	0.04
		5	0.2842	11.3168	12	0.04
	10^{-5}	2	2.0411	39.4993	51	0.15
		3	1.2809	18.1565	30	0.09
		4	0.9544	12.5077	30	0.09
		5	1.1047	13.1880	51	0.15
100		1	4.8494	41.2454	-	0.83
	10^{-3}	2	2.6736	42.3965	12	0.07
		3	1.0804	16.0402	12	0.07
		4	0.1696	5.6775	12	0.07
		5	0.1603	5.5867	12	0.07
	10^{-5}	2	2.8730	45.6307	99	0.56
		3	1.5432	18.4490	51	0.29
		4	0.9891	7.1667	51	0.28
		5	1.0542	6.6939	51	0.29

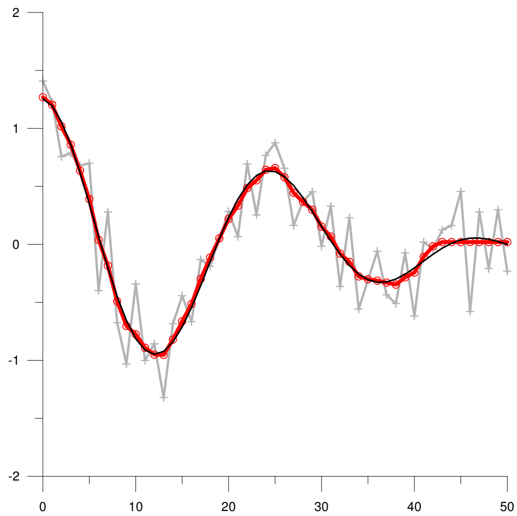

Figure 1: The unconstrained (+) and the piecewise monotonic lag coefficients with $k = 4$ ($\circ$), when $q = 50$, $r = 0.1$ and $\varepsilon = 10^{-3}$. Function $\phi(.)$ is denoted by the thin line

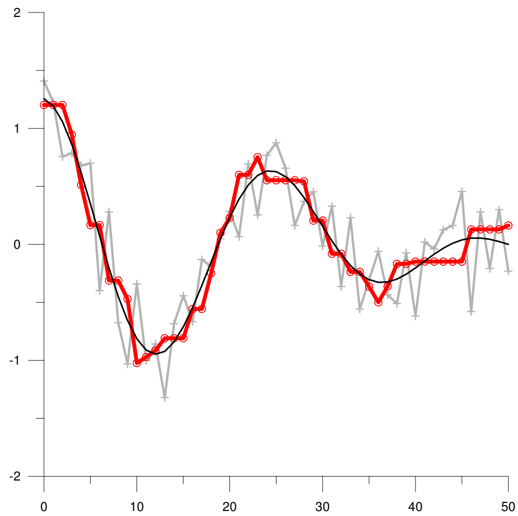

Figure 2: As in Figure 1, but $\varepsilon = 10^{-5}$

Table 3. We assume that a change in the GDP will affect not only current consumption, but also future consumption for seven time periods. Therefore we estimate the coefficients of the distributed-lag model with lag length $q = 7$ subject to the piecewise monotonicity constraints (6) on the components of β by considering separately the

Table 3: The values of GDP and PCE for U.S.A. during the years 1929-2006 and the least squares estimates to PCE from GDP when the lag coefficients consist of one, two and four monotonic sections

PCE	GDP	$k=1$	$k=2$	$k=4$	PCE	GDP	$k=1$	$k=2$	$k=4$
661.4	865.2				2310.5	3652.7	2474.8	2363.1	2361.3
626.1	790.7				2396.4	3765.4	2576.1	2460.5	2455.6
606.9	739.9				2451.9	3771.9	2642.7	2535.7	2533.2
553.0	643.7				2545.5	3898.6	2734.0	2635.9	2637.3
541.0	635.5				2701.3	4105.0	2848.2	2765.1	2761.8
579.3	704.2				2833.8	4341.5	2977.0	2910.0	2903.3
614.8	766.9				2812.3	4319.6	3033.6	2954.0	2946.1
677.0	866.6	580.7	603.0	599.9	2876.9	4311.2	3081.7	2993.7	2995.3
702.0	911.1	602.1	598.4	594.0	3035.5	4540.9	3188.5	3095.3	3096.6
690.7	879.7	605.9	582.4	580.4	3164.1	4750.5	3303.2	3222.6	3215.6
729.1	950.7	639.3	599.8	601.4	3303.1	5015.0	3447.7	3378.0	3371.7
767.1	1034.1	684.1	645.1	643.2	3383.4	5173.4	3568.7	3482.3	3473.2
821.9	1211.1	763.5	734.9	732.5	3374.1	5161.7	3634.4	3533.0	3527.3
803.1	1435.4	867.1	831.0	824.8	3422.2	5291.7	3727.8	3602.5	3604.2
826.1	1670.9	987.7	937.7	929.4	3470.3	5189.3	3755.2	3631.2	3628.4
850.2	1806.5	1088.7	1008.0	999.4	3668.6	5423.8	3871.3	3789.1	3795.2
902.7	1786.3	1146.2	1034.1	1029.9	3863.3	5813.6	4039.3	3972.6	3965.4
1012.9	1589.4	1139.7	1018.7	1020.8	4064.0	6053.7	4184.5	4136.2	4122.3
1031.6	1574.5	1163.4	1064.6	1074.1	4228.9	6263.6	4322.5	4238.8	4230.0
1054.4	1643.2	1203.4	1151.6	1154.2	4369.8	6475.1	4466.6	4329.6	4322.3
1083.5	1634.6	1218.0	1219.1	1217.9	4546.9	6742.7	4635.8	4480.0	4473.2
1152.8	1777.3	1267.5	1288.5	1288.9	4675.0	6981.4	4808.0	4631.3	4624.0
1171.2	1915.0	1317.9	1329.6	1323.4	4770.3	7112.5	4958.3	4789.0	4782.9
1208.2	1988.3	1356.5	1348.7	1342.8	4778.4	7100.5	5051.8	4895.0	4892.2
1265.7	2079.5	1407.8	1360.9	1358.3	4934.8	7336.6	5196.3	5052.3	5055.0
1291.4	2065.4	1437.7	1374.2	1371.2	5099.8	7532.7	5330.8	5206.0	5199.3
1385.5	2212.8	1512.2	1451.5	1452.8	5290.7	7835.5	5501.9	5402.3	5397.1
1425.4	2255.8	1561.6	1497.7	1493.3	5433.5	8031.7	5646.8	5539.5	5529.4
1460.7	2301.1	1612.1	1565.1	1564.5	5619.4	8328.9	5822.2	5705.1	5698.6
1472.3	2279.2	1634.1	1585.4	1584.3	5831.8	8703.5	6027.3	5881.2	5871.2
1554.6	2441.3	1704.5	1663.6	1665.0	6125.8	9066.9	6250.4	6079.8	6067.9
1597.4	2501.8	1750.5	1712.4	1706.8	6438.6	9470.3	6502.5	6314.3	6302.8
1630.3	2560.0	1798.5	1761.7	1760.2	6739.4	9817.0	6749.5	6534.1	6521.4
1711.1	2715.2	1874.7	1831.1	1829.3	6910.4	9890.7	6917.6	6686.8	6676.3
1781.6	2834.0	1946.1	1896.4	1890.9	7099.3	10048.8	7091.6	6851.9	6852.4
1888.4	2998.6	2036.6	1979.6	1975.7	7295.3	10301.0	7284.2	7061.8	7059.5
2007.7	3191.1	2144.6	2068.3	2063.1	7577.1	10703.5	7521.9	7345.4	7339.4
2121.8	3399.1	2267.9	2186.7	2180.2	7841.2	11048.6	7749.5	7592.3	7580.1
2185.0	3484.6	2359.3	2260.2	2253.4	8091.4	11415.3	7987.0	7828.8	7817.7

cases $k = 1$ and $k > 1$.

a) Monotonic lag coefficients ($k = 1$)
We require the estimated lag coefficients to be monotonically decreasing, so the problem is to minimize (2) subject to the constraints (4). The optimal lag coefficients are shown in the second column ($k = 1$) of Table 4, while the unconstrained lag coefficients due to (3) are shown in the fifth column ($\tilde{\beta}$) of this table. Fig. 3 displays these coefficients. Although the fluctuation of the unconstrained coefficients make them generally inadequate to the estimation problem, the first unconstrained coefficient seems to be more significant than the others. On the other hand, the optimal monotonic decreasing coefficients follow the main trend of the unconstrained coefficients and maintain the importance of the first coefficient. Therefore, with the monotonicity assumption, the resultant estimated values of PCE are given in the third column ($k = 1$) of Table 3 and displayed in Fig. 4 together with the provided GDP values. We see that the estimated PCE values via formula (1), which actually involves the GDP values, fall close to the observed PCE values. In particular, the current GDP

Table 4: The estimated ($k = 1$, 2 and 4) and the unconstrained ($\tilde{\beta}$) lag coefficients in Section 5

	$k=1$	$k=2$	$k=4$	$\tilde{\beta}$
β_0	0.2527	0.3097	0.3098	0.3105
β_1	0.0688	0.0453	0.0073	0.0037
β_2	0.0688	0.0453	0.0831	0.0862
β_3	0.0688	-0.0454	-0.0452	-0.0455
β_4	0.0688	0.0866	0.0883	0.1112
β_5	0.0688	0.0866	0.0884	0.0915
β_6	0.0688	0.0963	0.0967	0.0707
β_7	0.0688	0.1167	0.1126	0.1127

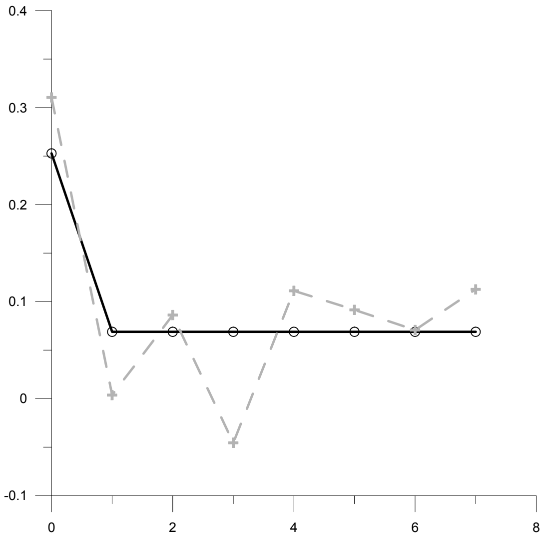

Figure 3: The unconstrained (+) and the monotonically decreasing (o) lag coefficients of Table 4

value rather than past ones affects mainly the associated PCE value. Indeed, in view of the monotonically decreasing lag coefficients, GDP affects strongly consumption at the beginning of the lags, while its action subsequently is reduced and kept at a low level. Thus the constraints (4) provide a plausible choice for the lag coefficients that leads to a satisfactory estimation of the true PCE values.

b) Piecewise monotonic lag coefficients ($k > 1$)
In order to illustrate some features of Algorithm 1 we performed two experiments. In the first experiment we calculated the lag coefficients by employing Algorithm 1 with $k = 2$ and $k = 4$, while the first monotonic section was let to be decreasing. The tolerance for the termination criterion (11) in Step 4 was set to 10^{-5} and the estimated values of β are shown in the third and fourth column of Table 4 and displayed in Fig. 5 together with the unconstrained lag coefficients. The algorithm terminated in 13 and 18 iterations, with $P_{RelError} = 3.5416$ and $P_{RelError} = 3.6911$ with respect to $k = 2$ and $k = 4$. As can be seen, these estimates follow the trend of the unconstrained lag coefficients, although the latter are not

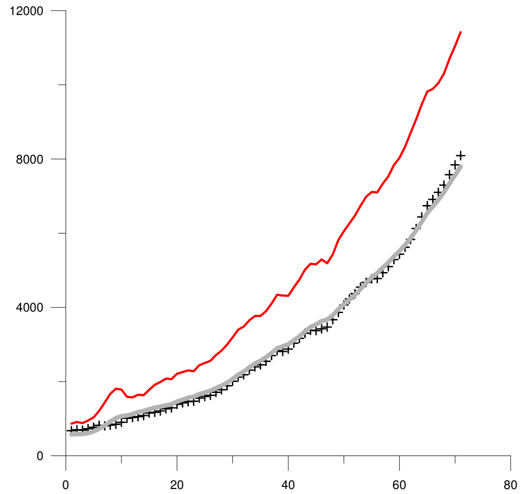

Figure 4: Least squares estimation (grey line) to the PCE values (+) with the monotonically decreasing lag coefficients ($k = 1$) of Table 4 on the GDP values (thin line) of Table 3

explicitly available. The $k = 2$ case has introduced one turning point at β_3, but the extra turning points of the $k = 4$ case that were added automatically at the 2nd and 3rd data point gave an estimate that is closer to the unconstrained one. The result suggests that the user may apply Algorithm 1 with increasing values of k until a satisfactory time series estimation is obtained (which can be checked by the change of $P_{RelError}$ as k changes). Of course, the prior knowledge parameter k may give the calculation valuable information. Further, the corresponding estimated values of PCE are shown in columns $k = 2$ and $k = 4$ of Table 3 and displayed in Fig. 6 together with the GDP values. Again, it is remarked that the estimated PCE values fall close to the observed PCE values providing very satisfactory estimations of the true PCE values.

In the second experiment we derived approximations to the lag coefficients by employing Algorithm 1 with $k = 2$, for $\varepsilon = 10^{-3}, 10^{-4}$ and 10^{-5}. The calculated values of β, which are identical for the last two values of ε, are shown in Table 5 and displayed in Fig. 7. As before, these approximations capture the pattern of the unconstrained lag coefficients. Moreover, the smaller the value of ε, the closer the approximation components are to the unconstrained coefficients, while these components cannot be worse than the unconstrained lag coefficients, due to the employed piecewise monotonicity constraints. Thus the user may decide to monitor the smoothing performance of the method by means of the tolerance magnitude in (11). In addition to smoothness, this is a helpful consideration for the convergence of the method, because due to degeneracy or near degeneracy of matrix X, a line search, sometimes, may have to choose a tiny steplength α_j, which implies that the algorithm may make slow progress to the solution.

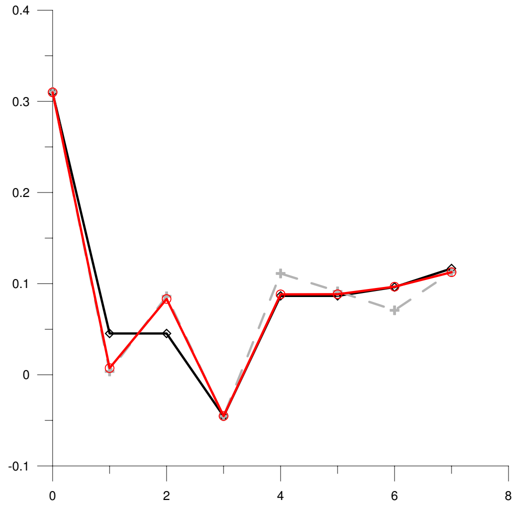

Figure 5: The unconstrained (+) and the piecewise monotonic lag coefficients with $k = 2$ ($\diamond$) and $k = 4$ ($\circ$) of Table 4

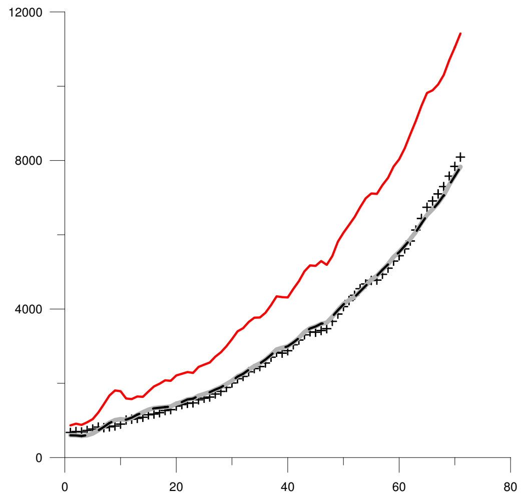

Figure 6: Least squares estimations (grey and dashed line) to the PCE values (+) with the piecewise monotonic lag coefficients ($k = 2$ and $k = 4$) of Table 4 on the GDP values (thin line) of Table 3

6 Conclusions and future work

We have developed a new method for calculating distributed-lag coefficients in time series data subject to the condition that the coefficient estimates are composed of a certain number of monotonic sections.

The method seems to be both effective in computation and competent to its modelling task. Three distinctive features of this process are to be noted: (1) The process is designed to overcome the multicollinearity problem that frequently occurs in practice, (2) the piecewise monotonicity model provides a rather weak, nonetheless

Table 5: The lag coefficients with $k = 2$ monotonic sections, for $\varepsilon = 10^{-3}, 10^{-4}$ and 10^{-5} in Step 4 of Algorithm 1

	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}, 10^{-5}$
β_0	0.2129	0.3097
β_1	0.1312	0.0453
β_2	0.0596	0.0453
β_3	0.0182	-0.0454
β_4	0.0343	0.0866
β_5	0.0690	0.0866
β_6	0.0980	0.0963
β_7	0.1198	0.1167

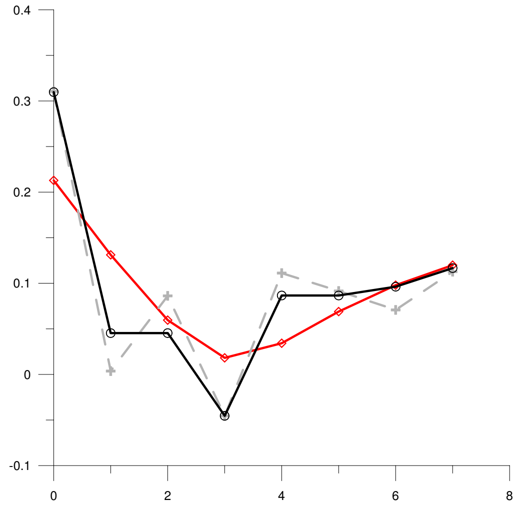

Figure 7: The unconstrained (+) and the lag coefficients with $k = 2$ monotonic sections, for $\varepsilon = 10^{-3}$ ($\diamond$), 10^{-4} ($\circ$) and 10^{-5} ($\circ$) of Table 5

realistic representation of the lag coefficients and, (3) the calculation benefits from the excellent complexity of the piecewise monotonicity method and the fast convergence of the conjugate gradient technique. In particular, the choice of the prior knowledge parameter k gives the time series estimation valuable flexibility.

For the piecewise monotonic model we have used a Fortran package that has been developed recently [3], which indeed is a major part of our calculation. For the special problem that minimizes (2) subject to the monotonic constraints (4) we have used Fortran codes developed by one of the authors (EEV).

The calculations performed so far on real data show that our method overcomes some severe shortcomings of traditional lag estimation techniques, while it provides a weak representation of the lag coefficients. Still, there is plenty of room for much empirical analysis on this method. The algorithm due to employing the conjugate gradient technique seems to be very fast for interactive computation.

It is expected that the algorithm will find useful applications to real problems, so work is under way in order to provide a Fortran package that will be accessible through a public software library.

References

[1] Almon, S., "The Distributed Lag between Capital Appropriations and Expenditures", *Econometrica*, V33, N1, pp. 178-196, 1965.

[2] Demetriou, I. C., "Discrete Piecewise Monotonic Approximation by a Strictly Convex Distance Function", *Mathematics of Computation*, V64, N209, pp. 157-180, 1995.

[3] Demetriou, I. C., "Algorithm 863: L2WPMA, a Fortran 77 Package for Weighted Least Squares Piecewise Monotonic Data Approximation", *ACM Transactions on Mathematical Software*, V33, N1, pp. 1-19, 2007.

[4] Demetriou, I. C., Powell, M. J. D., "Least Squares Smoothing of Univariate Data to Achieve Piecewise Monotonicity", *IMA Journal of Numerical Analysis*, V11, pp. 411-432, 1991.

[5] Demetriou, I. C., Vassiliou, E. E., "A Distributed Lag Estimator with Piecewise Monotonic Coefficients". In: *Proceedings of the World Congress on Engineering*, V II (Eds. S. I. Ao, L. Gelman, D. W. L. Hukins, A. Hunter, A. M. Korsunsky), London, U.K., 2-4 July, 2008, International Association of Engineers, pp. 1088-1094.

[6] Fisher, I., "Note on a Short-Cut Method for Calculating Distributed Lags", *International Statistical Institute Bulletin*, pp. 323-327, 1937.

[7] Fletcher, R., *Practical Methods of Optimization, Second Edition*, Chichester: J. Wiley and Sons, 2003.

[8] Gershenfeld, N., *The Nature of Mathematical Modelling*, Cambridge: Cambridge University Press, 1999.

[9] Goldstein, A. A., *Constructive Real Analysis*, New York: Harper, 1967.

[10] Golub, G., van Loan, C. F., *Matrix Computations, Second Edition*, Baltimore and London: The John Hopkins University Press, 1989.

[11] Gray, R. M., *Toeplitz and Circulant Matrices: A Review*, Stanford: Department of Engineering, http://ee.stanford.edu/~gray/toeplitz.pdf, 2006.

[12] Griliches, Z., "Distributed Lags: A Survey", *Econometrica*, V35, N1, pp. 16-49, 1967.

[13] Hannan, E. J., "The Estimation of Relations Involving Distributed Lags", *Econometrica*, V33, N1, pp. 206-224, 1965.

[14] Jorgenson, D., "Rational Distributed Lag Functions", *Econometrica*, V34, N1, pp. 135-149, 1966.

[15] Kailath, T. (editor), *Linear Least-Squares Estimation*, Stroudsburg, Pennsylvamia: Dowden, Hutchinson and Ross, Inc., Benchmark Papers in Electrical Engineering and Computer Science, V17, 1977.

[16] Katsaggelos, A. K., "Iterative Image Restoration Algorithms", *Optical Engineering*, V28, N7, pp. 735-748, 1989.

[17] Koop, G., *Analysis of Economic Data*, Chichester: J. Wiley and Sons, 2000.

[18] Koyck, L., *Distributed Lags and Investment Analysis*, Amsterdam: North-Holland Pub. Co., 1954.

[19] DeLeeuw, F., "The Demand for Capital Goods by Manufactures: A Study of Quartely Time Series", *Econometrica*, V30, 407-423, 1962.

[20] Luenberger, D. G., *Introduction to Linear and Nonlinear Programming*, Reading, Massachusetts: Addison-Wesley Pub. Co., 1973.

[21] Maddala, G. S., *Econometrics*, London: McGraw-Hill Book Co., 1977.

[22] Robertson, T., Wright, F. T., Dykstra, R. L. *Order Restricted Statistical Inference*, New York: J. Wiley and Sons, 1988.

[23] Shiller, R. J., "A Distributed Lag Estimator Derived from Smothness Priors", *Econometrica*, V41, N4, pp. 775-788, 1973.

[24] Solow, R. M., "On a Family of Lag Distributions", *Econometrica*, V28, N2, pp. 393-406, 1960.

[25] Tsay, R. S., *Analysis of Financial Time Series*, New York: J. Wiley and Sons, 2002.

[26] Zellner, A., Geisel, M. S., "Analysis of Distributed Lag Models with Applications to Consumption Function Estimation", *Econometrica*, V38, N6, pp. 865-888, 1970.