

A Hill Climbing Algorithm for Maximum Likelihood Estimation of the Gamma Distributed-lag Model with Multiple Explanatory Variables

Alessandro Magrini

Dep. Statistics, Computer Science, Applications – University of Florence, Italy

Abstract

Linear regression with distributed-lags is a consolidated methodology in time series analysis to assess the impact of several explanatory variables on an outcome that may persist over several periods. Finite polynomial distributed-lags have a long tradition due to a good flexibility accompanied by the advantage of a linear representation, which allows parameter estimation through Ordinary Least Squares (OLS). However, they require to specify polynomial degree and lag length, and entail the loss of some initial observations. Gamma distributed-lags overcome these problems and represents a good compromise between flexibility and number of parameters, however they have not a linear representation in the parameters and currently available estimation methods, like OLS-based grid search and non-linear least squares, are unsatisfactory in the case of multiple explanatory variables. For these reasons, the Gamma lag distribution has not been able to replace finite polynomial lags in applied time series analysis, and it has been mostly employed in the case of a single explanatory variable. In this paper, we propose a hill climbing algorithm for maximum likelihood estimation of multiple linear regression with Gamma distributed-lags. The proposed algorithm is applied to assess the dynamic relationship between Bitcoin's price and three composite indices of the US stock market.

Keywords: infinite distributed-lags, grid search, multiple dynamic impacts, ordinary least squares, time series regression.

1. Introduction

Linear regression with distributed-lags is a consolidated methodology in time series analysis to assess the impact of several explanatory variables on an outcome that may persist over several periods. In order to achieve efficiency in the estimation and to eventually comply with scientific theory, the distributed-lags are often constrained to a functional form in empirical applications.

Polynomial distributed-lags (Almon 1965; Andrews and Fair 1992) have a long tradition in applied time series analysis due to a good flexibility accompanied by the advantage of a linear representation, which allows the regression model to be estimated through Ordinary Least Squares (OLS). However, they require the specification of the polynomial degree and of the

maximum lag length, and entail the loss of some initial observations. These limitations are overcome by infinite distributed-lags. The first proposal of infinite lag distribution was the geometric lag, assuming exponentially decreasing weights. [Koyck \(1954\)](#) showed that the geometric lag can be represented as a linear model with first order autoregressive term and first order autocorrelated errors. Unfortunately, despite its desirable statistical properties, the geometric lag distribution has been criticized to be too restrictive for several applications, mainly because the weights do not increase towards a peak before decreasing to zero. In fact, the existence of a peak in the lag distribution is particularly motivated by several scientific theories, for instance, in the economic field, the reaction to factors like income, price level, investments and money supply.

The Pascal lag ([Solow 1960](#)) was proposed as an extension of the geometric lag, involving an additional integer parameter which allows to represent unimodal and right-skewed lag distributions. The family of rational distributed-lags ([Jorgenson 1966](#)) was suggested as a further extension of both the geometric and the Pascal lag, also including polynomial distributed-lags as a special case. A rational lag is defined by the ratio of two finite polynomial lags, and can be represented as an autoregressive distributed-lag linear model with autocorrelated errors, for which [Dhrymes, Klein, and Steiglitz \(1970\)](#) proposed a procedure for Maximum Likelihood Estimation (MLE). Rational lags overcome the problem of specifying the number of lags, but still maintain the problem of specifying the degrees of the polynomials, which cannot be treated as parameters to be estimated, thus requiring the comparison among models specified with different degrees.

In order to overcome the limitations of rational distributed-lags, [Tsurumi \(1971\)](#) proposed a single-parameter Gamma lag, extending the geometric lag to unimodal and right-skewed lag distributions. The model was applied to estimate an investment function for the Canadian manufacturing industry, with capital investments as response variable and capital stock as the only explanatory variable with lags. Since the model is not linear in the parameters, Non-linear Least Squares (NLS, [Hartley and Booker 1965](#)) estimation was employed.

[Schmidt \(1974\)](#) re-elaborated the Gamma lag proposed by [Tsurumi \(1971\)](#) into a formulation with two parameters varying between 0 and 1, and, as an alternative to NLS estimation, proposed an OLS-based Grid Search (OLS-GS) for MLE estimation, consisting in applying OLS to a grid of values for the two parameters and in selecting the estimate with the lowest residual sum of squares. The model was applied to estimate the impact of capital appropriations (the only explanatory variable with lags) on research expenditure (response variable), using the dataset on 602 US companies analyzed by [Almon \(1965\)](#).

[Tserkezos \(1998\)](#) exploited the Gamma-distributed lag model to perform quarterly disaggregation of the gross national product in some OECD countries. This is the first application considering more than one explanatory variable with lags, specifically the index of industrial production, retail sales and total employment. However, it is not clear whether the Gamma lag distributions are specified a-priori or estimated.

The problem of estimating the impact of public research expenditure on agricultural productivity was the field where the Gamma distributed-lag model found the major employment. [Thirtle, Piesse, and Schimmelpfennig \(2008\)](#) applied several families of lag distributions to the UK case, including the Gamma distributed-lag, that was estimated through NLS. [Balcombe and Rapsomanikis \(2010\)](#), focusing on United States of America (USA), assumed a Gamma lag distribution for public research expenditure and proposed a Bayesian model averaging approach for its estimation. [Alston, Andersen, James, and Pardey \(2011\)](#) inspired a long series of empirical works on USA leading to a consolidation of a Gamma lag for public research expenditure with length equal to 50 years and peak at 24 years ([Andersen and Song 2013](#); [Wang, Heisey, Huffman, and Fuglie 2013](#); [Jin and Hiuffman 2016](#); [Baldos, Viens, Hertel, and Fuglie 2018](#)). [Bervejillo, Alston, and Tumber \(2012\)](#) applied the same lag distribution to Uruguayan data and found similar results. Instead, using Czech data, [Ratinger and Kristkova \(2015\)](#) found a lag distribution for public research expenditure with length equal to 15 years

and peak at 5 years. All these applications considered a multi-factor productivity index as response variable, and, as explanatory variables with lags, public research expenditure and an estimated foreign knowledge stock. Further drivers of productivity like meteorological conditions were also included in the model, but with no lags. OLS-GS estimation was used in all of these applications, excepting Thirtle *et al.* (2008), who employed NLS estimation, and Balcombe and Rapsomanikis (2010); Baldos *et al.* (2018), who exploited Markov Chain Monte Carlo simulation.

From the analysis of the literature it is apparent that, since its first proposal, the Gamma lag distribution has found relatively little application compared to the finite polynomial one. Also, it has been employed mostly in the case of a single explanatory variable, with a predominant use of OLS-GS compared to NLS estimation and Markov Chain Monte Carlo (MCMC) simulation. We believe that the Gamma lag outperforms existing lag distributions for what concerns flexibility, but linear regression with Gamma-distributed lags, known as Gamma-distributed lag model, has been mostly used in the case of a single explanatory variable due to the unavailability of effective estimation methods. In fact, on one hand, OLS-based grid search proposed by Schmidt (1974) has an exponential time complexity with respect to the number of explanatory variables, thus it may become infeasible for more than two of them. On the other hand, NLS estimation represents an alternative with good properties, but, as noted by Tsurumi (1971), it may entail problems of convergence and particular care should be placed in the selection of initial values to avoid getting stuck at a local maximum.

Although it is true that even MLE of linear regression with finite polynomial distributed-lags requires to compare several different models with an exponential time complexity in order to select degree and lag length, several shortcuts are in use to effectively reduce the number of comparisons, like assuming that degree and lag length are the same for all the explanatory variables. We believe that this is the main reason that made finite polynomial distributed-lags very popular, despite their limitations. In this paper, we propose a hill climbing algorithm for MLE of the Gamma distributed-lag model, which is able to find a local maximum of the likelihood function in a reasonable time for an arbitrary number of explanatory variables. Performing multiple restarts from different random initial values allows to hopefully find the global maximum.

This paper is structured as follows. In Section 2, we formalize the Gamma distributed-lag model and review its currently available parameter estimation methods. In Section 3, we present our algorithm for MLE of the Gamma distributed-lag model with multiple explanatory variables. Section 4 includes a Monte Carlo experiment to assess the statistical properties of our algorithm. In Section 5, the proposed algorithm is applied to assess the dynamic relationship between Bitcoin's price and three composite indices of the US stock market. Section 6 includes concluding remarks and considerations on future developments.

2. The Gamma distributed-lag model

In this section, we provide a formalization of the linear regression model with Gamma distributed-lags, known as Gamma distributed-lag model (Subsection 2.1) and review its currently available parameter estimation methods (Subsection 2.2).

2.1. Formalization

Let Y be the response variable and X be an explanatory variable. Also, let y_t and x_t be the value of Y and X , respectively, at time t . Denote x_{t-k} as the value of X before k periods from t ; k is called *time lag*. We assume that time is discrete ($t = 1, \dots, T$) and that the time series of Y and X are weakly stationary, i.e., their expected value and autocorrelation function do not depend on time. The (simple) linear regression model with infinite distributed-lags

explaining Y from X is defined as:

$$y_t = \alpha + \theta \sum_{k=0}^{\infty} w_k x_{t-k} + \varepsilon_t \quad (1)$$

where α is the intercept, θ is a coefficient defining the scale of the lag distribution of X , w_k is the weight of X at time lag k , and ε_t is the random error at time t , which is assumed to have null expected value and to be linearly uncorrelated with X at any time lag. Note that $\beta_k = \theta w_k$ represents the linear regression coefficient of X at time lag k , often called dynamic coefficient, while $\beta^* = \theta \sum_{k=0}^{\infty} w_k$ represents the cumulative linear regression coefficient of X after a large number of time lags, often called long-term coefficient.

In the original formulation of the Gamma distributed-lag (Schmidt 1974), the weights w_k are defined as:

$$w_k = (k+1)^{\delta/(1-\delta)} \lambda^k \quad (2)$$

where $0 \leq \delta < 1$ and $0 \leq \lambda < 1$ are two real-valued parameters. In order to ease the interpretation of parameter θ , we force the weights in (2) to have sum 1:

$$w_k = \frac{(k+1)^{\delta/(1-\delta)} \lambda^k}{c(\delta, \lambda)} \quad (3)$$

$$c(\delta, \lambda) = \sum_{l=0}^{\infty} (l+1)^{\delta/(1-\delta)} \lambda^l$$

Note that the normalization constant $c(\delta, \lambda)$ has not an analytical expression, but it can be easily computed through numerical approximation. The weights in (3) are the same proposed by Alston *et al.* (2011), with the difference that the summation is not truncated to a finite time lag. Thus, the (simple) linear regression model with Gamma distributed-lags explaining Y from X has the form:

$$y_t = \alpha + \frac{\theta}{c(\delta, \lambda)} \sum_{k=0}^{\infty} (k+1)^{\delta/(1-\delta)} \lambda^k x_{t-k} + \varepsilon_t \quad (4)$$

where, since the weights w_k sum to 1, the long-term coefficient $\beta^* = \theta \sum_{k=0}^{\infty} w_k$ is equal to parameter θ .

Some remarks about the lag distribution implied by the weights in (3) are needed. First, it has a unique maximum (peak), which occurs at time lag:

$$k = \begin{cases} \frac{\delta}{(\delta-1) \log \lambda} - 1 & \lambda > 0 \\ 0 & \lambda = 0 \end{cases} \quad (5)$$

Second, if $\lambda = 0$ we get $w_0 = 1$ and $w_k = 0 \forall k \neq 0$ whichever the value of δ , thus a contemporaneous relationship between X and Y is obtained. Third, the geometric lag distribution is obtained for $\delta = 0$. Figure 1 shows the weights implied by three Gamma lag distributions, while Table 1 reports peak and quantiles of several Gamma lag distributions.

Now we generalize model (4) to the case of multiple explanatory variables. Let Y be the response variable and X_1, \dots, X_J be a set of J explanatory variables. Also, let y_t and $x_{j,t}$ ($j = 1, \dots, J$) be the value of Y and X_j , respectively, at time t . The multiple linear regression model with infinite distributed-lags explaining Y from X_1, \dots, X_J is defined as:

$$y_t = \alpha + \sum_{j=1}^J \theta_j \sum_{k=0}^{\infty} w_{j,k} x_{j,t-k} + \varepsilon_t \quad (6)$$

where θ_j is a coefficient defining the scale of the lag distribution of X_j , $w_{j,k}$ is the weight of X_j at time lag k , and ε_t is the random error at time t , which is assumed to have null expected

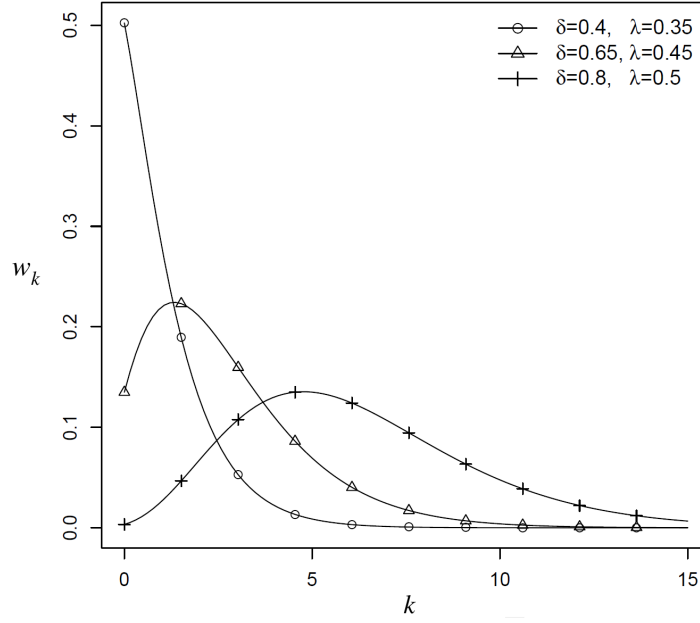


Figure 1: Weights implied by several Gamma lag distributions. Line with dots: $\delta = 0.4$, $\lambda = 0.35$. Line with triangles: $\delta = 0.65$, $\lambda = 0.45$. Line with crosses: $\delta = 0.8$, $\lambda = 0.5$.

Table 1: Peak and quantiles of several Gamma lag distributions

δ	λ	Peak	50%	95%	99%	δ	λ	Peak	50%	95%	99%
0.60	0.25	0.1	1	3	4	0.50	0.90	8.5	15	44	62
0.55	0.50	0.8	2	6	9	0.55	0.90	10.6	17	47	66
0.45	0.65	0.9	3	9	14	0.60	0.90	13.2	20	52	71
0.50	0.70	1.8	4	12	18	0.50	0.95	18.5	32	91	128
0.60	0.70	3.2	5	15	20	0.65	0.95	35.2	48	118	158
0.55	0.75	3.2	6	17	23	0.70	0.95	44.5	58	131	174
0.60	0.75	4.2	7	18	25	0.75	0.95	57.5	71	150	195
0.55	0.80	4.5	8	22	31	0.80	0.95	77.0	90	177	225
0.60	0.80	5.7	9	24	33	0.95	0.85	115.9	120	171	195
0.55	0.85	6.5	11	30	42	0.95	0.90	179.3	186	264	301
0.60	0.85	8.2	12	33	45	0.95	0.95	369.4	382	543	620

value and to be linearly uncorrelated with X_1, \dots, X_J at any time lag. The multiple linear regression with Gamma distributed-lags is obtained for:

$$w_{j,k} = \frac{(k+1)^{\delta_j/(1-\delta_j)} \lambda_j^k}{c(\delta_j, \lambda_j)} \quad (7)$$

$$c(\delta_j, \lambda_j) = \sum_{l=0}^{\infty} (l+1)^{\delta_j/(1-\delta_j)} \lambda_j^l$$

leading to the following formulation:

$$y_t = \alpha + \sum_{j=1}^J \frac{\theta_j}{c(\delta_j, \lambda_j)} \sum_{k=0}^{\infty} (k+1)^{\delta_j/(1-\delta_j)} \lambda_j^k x_{j,t-k} + \varepsilon_t \quad (8)$$

where α is the intercept, $\beta_{j,k} = \theta_j w_{j,k}$ is the linear regression coefficient of X_j at time lag k , and $\beta_j^* = \theta_j \sum_{w=0}^{\infty} w_{j,k} = \theta_j$ is the linear regression coefficient of X_j after a large number of time lags, i.e., the long-term coefficient of X_j .

2.2. Parameter estimation methods

Maximum Likelihood Estimation (MLE) of model (8) is challenging because, for $j = 1, \dots, J$, parameters δ_j and λ_j enter non-linearly. First of all, note that model (6) can be written as:

$$y_t = \alpha + \sum_{j=1}^J \theta_j Z_{j,t} + \sum_{j=1}^J W_{j,t} + \varepsilon_t$$

$$Z_{j,t} = \sum_{k=0}^{t-1} w_{j,k} x_{j,t-k}$$

$$W_{j,t} = \theta_j \sum_{k=t}^{\infty} w_{j,k} x_{j,t-k}$$
(9)

where, for $j = 1, \dots, J$, the term $W_{j,t}$ is asymptotically negligible and its omission does not affect the asymptotic properties of MLE, although estimates in small samples are significantly more efficient when $W_{j,t}$ is estimated rather than dropped (Schmidt 1975; Baltagi and Ferry 1985). The decomposition in (9) allows to estimate any infinite distributed-lag model in finite samples without losing observations for the sake of considering several lags of the explanatory variables.

We begin the discussion on parameter estimation from the case of one explanatory variable, thus, for the moment, we focus on model (4). Observing that it is linear in both α and θ if both δ and λ are held constant, Schmidt (1974) proposed to apply OLS to a grid of values for δ and λ and to select the estimate with the lowest residual sum of squares. This method, here referred as OLS-based Grid Search (OLS-GS), returns the maximum likelihood estimate given the considered grid resolution, i.e., the number of equally-spaced values of δ and λ . OLS-GS in the case of one explanatory variable requires to run OLS q^2 times, where q is the grid resolution, while, in the general case of J explanatory variables, OLS should be run q^{2J} times. Actually, if value 0 is considered for δ and λ , we have a total of $[q(q-1)+1]^J$ unique combinations, as all the Gamma lag distributions with $\lambda = 0$ are identical. For example, if q is set to 20, then all the values of δ and λ from 0 to 0.95 differing by 0.05 are considered, and the case $J = 1$ requires 381 runs of OLS, while the cases $J = 2$ and $J = 3$ require, respectively, 145161 and 55.3 million runs. It is clear that OLS-GS may result practically infeasible for $J > 2$ even for a low grid resolution.

Non-linear Least Squares (NLS) estimation represents an alternative with good properties, as it equates to maximize a Gaussian likelihood, thus it has the property of consistency and asymptotic normality (Hartley and Booker 1965). However, as noted by Tsurumi (1971), it may entail problems of convergence and great effort should be placed in the selection of initial values to avoid getting stuck at a local maximum. Among the empirical applications of the Gamma distributed-lag model, Thirtle *et al.* (2008) complained the lack of convergence of NLS, and concluded that the Gamma lag distribution was not appropriate for their problem. MCMC simulation is an alternative to MLE methods which also allows to take into account prior information on the parameters, if available, but it may take very much time to achieve convergence (see Balcombe and Rapsomanikis 2010; Baldos *et al.* 2018).

3. Hill climbing algorithm for maximum likelihood estimation

Since model (8) is not linear in the parameters, the likelihood may have several local maxima. OLS-GS is able to discover the global maximum, but, when the number of explanatory variables increases, it becomes infeasible and only a heuristic search is practicable. At this purpose, we exploit the hill climbing scheme (Russell and Norvig 2003, pages 111–114), a widely adopted strategy in artificial intelligence to heuristically explore high dimensional spaces. The hill climbing is an iterative algorithm which starts from an arbitrary solution, then attempts to find a better one by making all the possible incremental changes. If a change produces a

better solution, then the procedure is repeated until no further improvements are possible. In our case, the space to be explored is composed of all the models that can be generated by instantiating δ and λ parameters in model (8), and the objective function is their residual sum of squares (to be minimized).

Let q be the grid resolution. As a consequence, $1/q$ is the distance between the values of the grid. Given a combination of δ and λ parameters, the proposed algorithm considers, for one explanatory variable X_j at a time, all the possible changes by $\pm 1/q$ to the pair (δ_j, λ_j) . Given a combination of δ and λ parameters, there are 8 possible changes that can be made for one explanatory variable (see Table 2), for a total of $8J$ possible changes. OLS estimation is performed for all of them and the one with the lowest residual sum of squares is selected. Then, if such residual sum of squares is lower than the one of the current position, then the change is performed and the procedure is repeated; otherwise, a local maximum of the likelihood is found and the algorithm stops. Full details are provided in Algorithm 1.

Table 2: All the possible changes by $\pm 1/q$ to the pair (δ_j, λ_j) allowed by Algorithm 1

Change	δ_j	λ_j
1	+0	+1/q
2	+0	-1/q
3	+1/q	+0
4	+1/q	+1/q
5	+1/q	-1/q
6	-1/q	+0
7	-1/q	+1/q
8	-1/q	-1/q

Algorithm 1 Maximum likelihood estimation of the Gamma distributed-lag model with multiple explanatory variables.

- Take as inputs vector `delta.best` containing the initial values for $\delta_1, \dots, \delta_J$, and vector `lambda.best` containing the initial values for $\lambda_1, \dots, \lambda_J$.
 - Define `delta.current` as a copy of `delta.best` and `lambda.current` as a copy of `lambda.best`. Initialize `theta.best`, `theta.current` and `theta.test` as empty vectors of length J . Initialize `rss.best` and `rss.current` as $+\infty$.
 - For $j = 1, \dots, J$:
 - For $k = 1, \dots, 8$:
 - * Define `delta.test` as a copy of `delta.current`, and `lambda.test` as a copy of `lambda.current`. Set position j of `delta.test` and `lambda.test` according to Change k in Table 2.
 - * Use OLS to fit a linear regression model with Gamma distributed-lags given `delta.test` and `lambda.test`. Set `theta.test` as the estimates of $\theta_1, \dots, \theta_J$, and `rss.test` as the residual sum of squares.
 - * If `rss.test` is less than `rss.current`, then set `theta.current` as `theta.test`, `delta.current` as `delta.test`, `lambda.current` as `lambda.test` and `rss.current` as `rss.test`.
 - If `rss.current` is less than `rss.best`, then set `theta.best` as `theta.current`, `delta.best` as `delta.current`, `lambda.best` as `lambda.current` and `rss.best` as `rss.current`.
 - Return `theta.best`, `delta.best` and `lambda.best`.
-

In order to discover eventual other local maxima and hopefully converge to the global one (the maximum likelihood estimate), the algorithm is restarted several times from different random initial values. We implemented the algorithm in an R (R Core Team 2020) package available on Github at <https://github.com/alessandromagrini/gammadlm>. The package can be installed from the R console by typing `install_github("alessandromagrini/gammadlm")` after loading the `devtools` package.

4. Monte Carlo experiment

A Monte Carlo experiment was performed to assess the statistical properties of our hill climbing algorithm. We considered seven data-generating models obtained by combining the three lag distributions shown in Figure 1 under the case of one, two and three explanatory variables (see Table 3). The choice of these three lag distributions was motivated by their representativeness for the population of Gamma lag distributions with length between 10 and 20, which is expected to cover a good part of real-world applications. We tested three different numbers of periods $T \in \{30, 50, 100\}$ and two values for the first-order autocorrelation of the random errors $\rho \in \{0, 0.25\}$. For each combination of T , and ρ , we performed 500 Monte Carlo replications. For each replication, explanatory variables were simulated from a multivariate Normal distribution with null mean, unit variance and 0.5 correlation, and the algorithm was run two times: one time with 100 and the other time with 500 random restarts, both with grid resolution $q = 20$. Throughout the experiment, we set $\alpha = 0$ and error variance equal to 1 for all the data-generating models. The Monte Carlo experiment was run on a laptop with Intel[®] i5 processor. The average processing time for a single run of Algorithm 1 is reported in Table 4, while the mean squared error of the estimates are shown in Tables 5 and 6.

Table 3: Data-generating models considered in the Monte Carlo experiment

Model	J	Parameters		
\mathcal{M}_{1a}	1	$\delta_1 = 0.4$	$\lambda_1 = 0.35$	$\theta_1 = 1$
\mathcal{M}_{1b}	1	$\delta_1 = 0.65$	$\lambda_1 = 0.45$	$\theta_1 = 1$
\mathcal{M}_{1c}	1	$\delta_1 = 0.8$	$\lambda_1 = 0.5$	$\theta_1 = 1$
\mathcal{M}_{2a}	2	$\delta_1 = 0.4$ $\delta_2 = 0.65$	$\lambda_1 = 0.35$ $\lambda_2 = 0.45$	$\theta_1 = 1$ $\theta_2 = 1$
\mathcal{M}_{2b}	2	$\delta_1 = 0.4$ $\delta_2 = 0.8$	$\lambda_1 = 0.35$ $\lambda_2 = 0.5$	$\theta_1 = 1$ $\theta_2 = 1$
\mathcal{M}_{2c}	2	$\delta_1 = 0.65$ $\delta_2 = 0.8$	$\lambda_1 = 0.45$ $\lambda_2 = 0.5$	$\theta_1 = 1$ $\theta_2 = 1$
\mathcal{M}_3	3	$\delta_1 = 0.4$ $\delta_2 = 0.65$ $\delta_3 = 0.8$	$\lambda_1 = 0.35$ $\lambda_2 = 0.45$ $\lambda_3 = 0.5$	$\theta_1 = 1$ $\theta_2 = 1$ $\theta_3 = 1$

From Table 4, we see that the processing time increases less than proportionally with T and, at constant T , it is practically the same for all the tested values of ρ . This holds for both 100 and 500 random restarts of the algorithm, with the difference that the processing time in the latter case is higher with an almost proportional relationship, as it is reasonable. Instead, the number of explanatory variables J is a great source of computation burden: the processing time for $J = 2$ is almost 7 times the one for $J = 1$, and the processing time for $J = 3$ is almost 2.5 times the one for $J = 2$. Note that the processing time is bounded to the minimum between the number of possible models and the number of random restarts, case in which Algorithm 1 equates to OLS-GS. Thus, the case $T = 30$ and $J = 1$ represents an exception where 500 random restarts reduce to 381, according to the calculus made in Section 2.

Table 4: Average processing time (seconds) for a single run of Algorithm 1 on a laptop with Intel[®] i5 processor

100 random restarts								
T	ρ	\mathcal{M}_{1a}	\mathcal{M}_{1b}	\mathcal{M}_{1c}	\mathcal{M}_{2a}	\mathcal{M}_{2b}	\mathcal{M}_{2c}	\mathcal{M}_3
30	0.00	3.25	3.26	3.27	24.27	23.59	23.67	56.81
50	0.00	3.28	3.30	3.30	25.48	24.59	25.18	57.07
100	0.00	3.36	3.36	3.36	28.03	27.45	27.69	59.39
30	0.25	3.26	3.26	3.26	23.73	23.65	23.57	56.86
50	0.25	3.29	3.28	3.29	25.37	25.16	25.27	57.61
100	0.25	3.35	3.38	3.36	28.10	27.07	27.55	59.11
500 random restarts								
T	ρ	\mathcal{M}_{1a}	\mathcal{M}_{1b}	\mathcal{M}_{1c}	\mathcal{M}_{2a}	\mathcal{M}_{2b}	\mathcal{M}_{2c}	\mathcal{M}_3
30	0.00	3.47	3.47	3.48	131.36	132.26	132.44	283.83
50	0.00	3.49	3.50	3.50	140.64	141.80	141.37	324.18
100	0.00	3.57	3.57	3.58	148.54	147.76	148.46	348.23
30	0.25	3.46	3.47	3.49	131.80	132.25	132.72	286.58
50	0.25	3.49	3.49	3.50	141.42	141.68	141.09	326.02
100	0.25	3.56	3.57	3.57	147.21	148.02	147.69	346.98

From Tables 5 and 6, we see that the root mean squared error of the estimates always decreases with the increase of the sample size T , as expected, and it is generally higher when the first order autocorrelation of the errors is not null. Also, root mean squared errors based on 500 random restarts are near to those based on 100 restarts for most data-generating models and experimental combinations, suggesting a similar performance of the algorithm in the two cases. Thus, we aimed at assessing whether the algorithm with 100 random restarts has good chances to find the local maximum. At this purpose, we used the root mean squared errors based on 500 random restarts for data-generating models with one explanatory variable (\mathcal{M}_{1a} , \mathcal{M}_{1b} and \mathcal{M}_{1c}) as benchmark for the precision of the estimates (Table 6, first block). This is justified by the fact that, in the case $J = 1$, our algorithm with 500 random restarts equates OLS-GS estimation. For instance, the root mean squared errors from \mathcal{M}_{1a} can be compared with those for the first explanatory variable in \mathcal{M}_{2a} , \mathcal{M}_{2c} and \mathcal{M}_3 (see Table 3 for a detailed correspondence). Clearly, the comparison is only indicative, because the precision of the estimates is expected to further decrease when the number of explanatory variables increases. By observing the results, it is apparent that root mean squared errors for data-generating models with two and three explanatory variables are comparable to those for data-generating models with one explanatory variable, suggesting a good ability of the algorithm to find the global maximum with 100 random restarts.

5. Real-world application

We apply our hill climbing algorithm to estimate the parameters of a Gamma distributed-lag model explaining Bitcoin's exchange rate (BTC) based on three composite indices of the US stock market: Dow Jones Industrial Average (DJA), Nasdaq Composite (IXIC) and Standard&Poor 500 (GSPC). The correlation between Bitcoin's exchange rate and the US stock market is of great interest for financial analysts, as a negative sign could indicate that investors consider cryptocurrencies as a safe-haven in times of financial turmoil. Our purpose is neither to provide new findings nor to develop a new theory on the relationship between Bitcoin's exchange rate and the US stock market, but rather to illustrate the application of Algorithm 1 in the case of multiple explanatory variables. Thus, the relationship between Bitcoin's exchange rate and the US stock market assessed here should be interpreted in a predictive sense, rather than causally.

Table 5: Root mean squared errors for Algorithm 1 with 100 random restarts (500 Monte Carlo replications)

		\mathcal{M}_{1a}			\mathcal{M}_{1b}			\mathcal{M}_{1c}		
T	ρ	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$
30	0.00	0.378	0.259	1.113	0.281	0.293	1.895	0.277	0.334	1.914
50	0.00	0.367	0.243	0.646	0.258	0.275	0.904	0.263	0.300	1.132
100	0.00	0.352	0.219	0.307	0.255	0.240	0.417	0.206	0.266	0.538
30	0.25	0.392	0.253	1.887	0.282	0.282	2.304	0.267	0.322	2.395
50	0.25	0.371	0.247	0.936	0.266	0.263	1.330	0.196	0.298	1.454
100	0.25	0.350	0.234	0.405	0.205	0.237	0.775	0.189	0.276	0.766

		\mathcal{M}_{2a}					
T	ρ	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$	$\hat{\delta}_2$	$\hat{\lambda}_2$	$\hat{\theta}_2$
30	0.00	0.367	0.224	2.316	0.302	0.242	2.125
50	0.00	0.351	0.217	0.798	0.242	0.231	0.923
100	0.00	0.337	0.206	0.316	0.217	0.216	0.468
30	0.25	0.360	0.234	2.381	0.280	0.249	2.441
50	0.25	0.329	0.222	1.049	0.257	0.233	1.616
100	0.25	0.330	0.212	0.500	0.222	0.219	0.723

		\mathcal{M}_{2b}					
T	ρ	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$	$\hat{\delta}_2$	$\hat{\lambda}_2$	$\hat{\theta}_2$
30	0.00	0.351	0.237	2.277	0.279	0.271	2.689
50	0.00	0.330	0.219	0.741	0.208	0.256	1.116
100	0.00	0.331	0.211	0.305	0.200	0.219	0.501
30	0.25	0.368	0.223	3.182	0.247	0.257	2.860
50	0.25	0.346	0.227	1.260	0.229	0.262	1.482
100	0.25	0.327	0.215	0.462	0.186	0.221	0.809

		\mathcal{M}_{2c}					
T	ρ	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$	$\hat{\delta}_2$	$\hat{\lambda}_2$	$\hat{\theta}_2$
30	0.00	0.281	0.234	2.920	0.261	0.279	2.795
50	0.00	0.237	0.227	1.268	0.250	0.266	1.344
100	0.00	0.200	0.210	0.548	0.205	0.250	0.601
30	0.25	0.251	0.225	3.254	0.225	0.269	3.193
50	0.25	0.226	0.218	1.818	0.203	0.256	1.670
100	0.25	0.204	0.209	0.767	0.156	0.245	0.889

		\mathcal{M}_3								
T	ρ	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$	$\hat{\delta}_2$	$\hat{\lambda}_2$	$\hat{\theta}_2$	$\hat{\delta}_3$	$\hat{\lambda}_3$	$\hat{\theta}_3$
30	0.00	0.340	0.225	2.599	0.246	0.208	2.929	0.298	0.244	2.949
50	0.00	0.320	0.215	1.135	0.233	0.206	1.491	0.273	0.231	1.529
100	0.00	0.315	0.198	0.441	0.207	0.207	0.585	0.191	0.205	0.684
30	0.25	0.344	0.233	2.625	0.250	0.216	3.033	0.234	0.234	3.239
50	0.25	0.329	0.214	1.431	0.241	0.209	1.849	0.220	0.225	1.981
100	0.25	0.320	0.205	0.569	0.200	0.198	0.798	0.210	0.207	0.879

We considered daily close exchange rates in US dollars from April to September 2020 available on Yahoo Finance, for a total of 127 observations. All the time series appear definitely non-stationary in level (Figure 2, left column), as confirmed by the augmented Dickey-Fuller test (Dickey and Fuller 1981) that was unable to reject the hypothesis of unit root for all of them. Instead, the hypothesis of unit root was rejected for all the time series taken in log return, i.e., after applying first order difference of logarithmic values (Figure 2, right column). Data summaries are shown in Table 7.

Table 6: Root mean squared errors for Algorithm 1 with 500 random restarts (500 Monte Carlo replications)

		\mathcal{M}_{1a}			\mathcal{M}_{1b}			\mathcal{M}_{1c}		
T	ρ	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$
30	0.00	0.376	0.248	1.295	0.285	0.288	1.891	0.277	0.323	2.057
50	0.00	0.364	0.243	0.512	0.266	0.274	0.904	0.234	0.304	1.151
100	0.00	0.351	0.214	0.278	0.255	0.246	0.487	0.204	0.259	0.575
30	0.25	0.394	0.257	2.284	0.305	0.284	2.338	0.259	0.315	2.567
50	0.25	0.365	0.246	0.878	0.263	0.262	1.401	0.216	0.301	1.373
100	0.25	0.353	0.232	0.440	0.237	0.251	0.704	0.216	0.271	0.786
		\mathcal{M}_{2a}								
T	ρ	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$	$\hat{\delta}_2$	$\hat{\lambda}_2$	$\hat{\theta}_2$			
30	0.00	0.394	0.252	2.209	0.337	0.277	2.242			
50	0.00	0.371	0.245	1.150	0.277	0.258	0.946			
100	0.00	0.359	0.239	0.326	0.270	0.241	0.443			
30	0.25	0.398	0.266	2.320	0.285	0.281	2.449			
50	0.25	0.388	0.257	1.267	0.274	0.269	1.689			
100	0.25	0.364	0.248	0.518	0.226	0.250	0.724			
		\mathcal{M}_{2b}								
T	ρ	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$	$\hat{\delta}_2$	$\hat{\lambda}_2$	$\hat{\theta}_2$			
30	0.00	0.360	0.216	2.431	0.323	0.329	2.373			
50	0.00	0.350	0.235	0.817	0.307	0.300	1.151			
100	0.00	0.337	0.240	0.324	0.285	0.268	0.537			
30	0.25	0.378	0.229	2.601	0.276	0.309	3.081			
50	0.25	0.360	0.250	1.281	0.247	0.291	1.834			
100	0.25	0.341	0.211	0.669	0.185	0.215	1.046			
		\mathcal{M}_{2c}								
T	ρ	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$	$\hat{\delta}_2$	$\hat{\lambda}_2$	$\hat{\theta}_2$			
30	0.00	0.285	0.274	3.149	0.326	0.314	2.775			
50	0.00	0.245	0.261	1.452	0.224	0.294	1.335			
100	0.00	0.214	0.255	0.571	0.199	0.273	0.657			
30	0.25	0.252	0.275	3.271	0.288	0.292	2.983			
50	0.25	0.237	0.263	1.977	0.221	0.267	1.907			
100	0.25	0.201	0.245	0.793	0.171	0.247	0.587			
		\mathcal{M}_3								
T	ρ	$\hat{\delta}_1$	$\hat{\lambda}_1$	$\hat{\theta}_1$	$\hat{\delta}_2$	$\hat{\lambda}_2$	$\hat{\theta}_2$	$\hat{\delta}_3$	$\hat{\lambda}_3$	$\hat{\theta}_3$
30	0.00	0.361	0.286	2.641	0.293	0.226	3.191	0.338	0.297	3.208
50	0.00	0.343	0.263	1.410	0.271	0.219	1.666	0.330	0.272	1.616
100	0.00	0.327	0.255	0.530	0.239	0.205	0.766	0.216	0.257	0.730
30	0.25	0.372	0.297	2.978	0.309	0.259	3.281	0.386	0.276	3.369
50	0.25	0.356	0.267	1.647	0.264	0.234	1.726	0.353	0.260	2.028
100	0.25	0.335	0.263	0.635	0.205	0.214	0.881	0.231	0.245	0.926

In order to avoid spurious regression due to non-stationarity (Granger and Newbold 1974), we worked with the variables in log return. We applied model (8) and estimated the parameters using Algorithm 1 with grid resolution $q = 20$ and the constraint that the peak of each lag distribution should not occur before time lag 0. Such constraint equates to disregard, for $j = 1, \dots, J$, the combinations of δ_j and λ_j values implying a negative peak according to (5). Under this constraint, the number of possible models is almost 13.6 millions, a quite large number that clearly cannot be handled by OLS-GS estimation. We ran the algorithm with 100, 500 and 1000 random restarts, obtaining the same result. The processing time for 100,

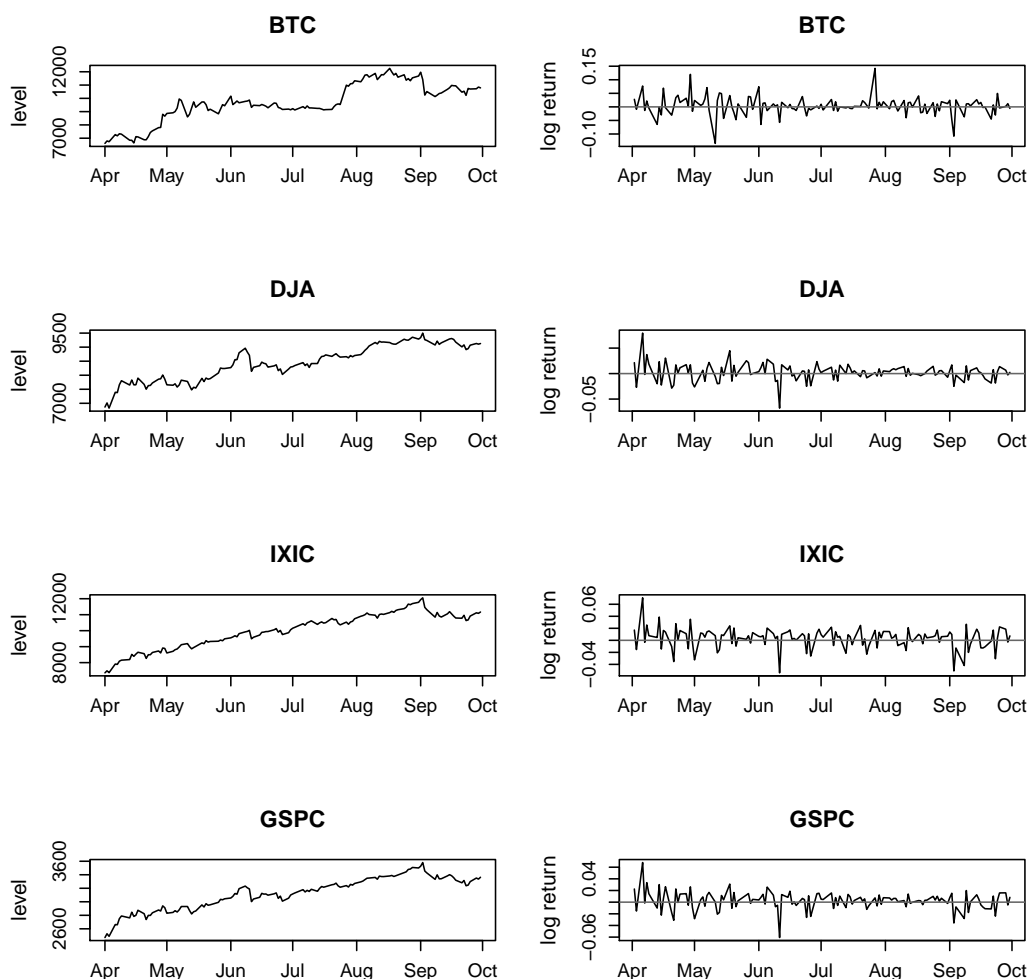


Figure 2: Daily close exchange rate in US dollars of BTC, DJA, IXIC and GSPC from April to September 2020. Left column: time series in level. Right column: time series in log return.

Table 7: Data summaries: daily close exchange rates in US dollars from April to September 2020 in level and in log return

	BTC		DJA		IXIC		GSPC	
	level	log return	level	log return	level	log return	level	log return
Minimum	6607	-0.135	6828	-0.068	7361	-0.054	2470	-0.061
1st quartile	9148	-0.011	7869	-0.005	9169	-0.006	2930	-0.006
Median	9528	0.002	8421	0.004	10095	0.006	3149	0.004
Mean	9668	0.004	8435	0.002	9989	0.003	3122	0.002
3rd quartile	10826	0.019	9048	0.011	10905	0.013	3322	0.011
Maximum	12254	0.142	9500	0.079	12056	0.071	3581	0.068

500 and 1000 restarts was 87, 452 and 980 seconds, respectively, on a laptop with Intel[®] i5 processor.

After parameter estimation, we checked the presence of auto-correlation and heteroscedasticity in the residuals by performing Durbin-Watson (Durbin and Watson 1971) and White (White 1980) tests. They did not reject the null hypothesis of no auto-correlation and of homoscedasticity, respectively, thus no correction for consistency was applied to the estimated standard errors. The summary of parameter estimation is provided in Table 8, while the implied dynamic coefficients are shown in Table 9 and displayed in Figure 3.

According to the R-squared, the model explains 21.8% of the variability of BTC on the logarithmic scale, with a mean absolute percentage error of prediction on the original scale equal to 2.8% (Figure 4), confirming the predictive relevance of the three US stock market indices, even if other predictors may exist besides them. Our results show that: (i) the lag distribution of DJA has negative sign and 99th percentile equal to 4; (ii) the lag distribution of IXIC has negative sign and 99th percentile equal to 9; (iii) the lag distribution of GSPC has positive sign and 99th percentile equal to 8. Since all the variables are in log return, the dynamic coefficients approximate dynamic elasticities. For instance, $\beta_{j,k} = \theta_j w_{j,k}$ represents the expected percentage variation in Bitcoin's exchange rate after k time lags (days) that variable X_j increased its value by 1%. Thus, based on our results, we deduce that, in the long-term: (i) a 1% increase in DJA is expected to decrease BTC by 2.72%; (ii) a 1% increase in IXIC is expected to decrease BTC by 3.45%; (iii) a 1% increase in GSPC is expected to increase BTC by 6.22%. The estimated effects of the considered US stock market indices on Bitcoin's exchange rate are quite large, but they appear plausible because financial time series are typically characterized by high variability, thus variations tend to have different sign and to generate effects which counterbalance each other rapidly. Also, note that the effects of the three composite indices here considered have different sign, thus, in the typical case when they vary in the same direction (as the most stock market indices are positively correlated in the long-term), the total effect on Bitcoin's exchange rate may be low or null. For instance, if DJA, IXIC and GSPC increase all by 1%, then BTC is expected to vary only by $-2.72\% - 3.45\% + 6.22\% = +0.05\%$ in the next 11 days.

Table 8: Summary of parameter estimation. Standard errors are shown within brackets.

X_j	$\hat{\delta}_j$	$\hat{\lambda}_j$	$\hat{\theta}_j$	Peak	50%	95%	99%
DJA	0.85	0.05	-2.718 (0.579)	0.9	1	3	4
IXIC	0.75	0.35	-3.455 (0.747)	1.9	2	6	9
GSPC	0.55	0.45	6.228 (1.114)	0.5	1	5	8
$\hat{\alpha} = 0.006$ (0.003). Residual std. dev.: 0.031. R-squared: 0.218							

Table 9: Estimated dynamic coefficients. Standard errors are shown within brackets.

Time lag	DJA	IXIC	GSPC
0	-0.523 (0.112)	-0.245 (0.053)	1.570 (0.281)
1	-1.329 (0.283)	-0.685 (0.148)	1.649 (0.295)
2	-0.661 (0.141)	-0.809 (0.175)	1.218 (0.218)
3	-0.169 (0.036)	-0.671 (0.145)	0.779 (0.139)
4	-0.030 (0.006)	-0.459 (0.099)	0.460 (0.082)
5	-0.004 (0.001)	-0.277 (0.060)	0.259 (0.046)
6	-0.001 (0.000)	-0.154 (0.033)	0.141 (0.025)
7	0.000 (0.000)	-0.081 (0.017)	0.075 (0.013)
8	0.000 (0.000)	-0.040 (0.009)	0.039 (0.007)
9	0.000 (0.000)	-0.019 (0.004)	0.020 (0.004)
10	0.000 (0.000)	-0.009 (0.002)	0.010 (0.002)
11	0.000 (0.000)	-0.004 (0.001)	0.005 (0.001)
0 to 11	-2.718 (0.579)	-3.452 (0.746)	6.223 (1.113)
0 to ∞	-2.718 (0.579)	-3.455 (0.747)	6.228 (1.114)

6. Concluding remarks

We have proposed a hill climbing algorithm for maximum likelihood estimation of the Gamma distributed-lag model with multiple explanatory variables. The Gamma lag is a very flexible

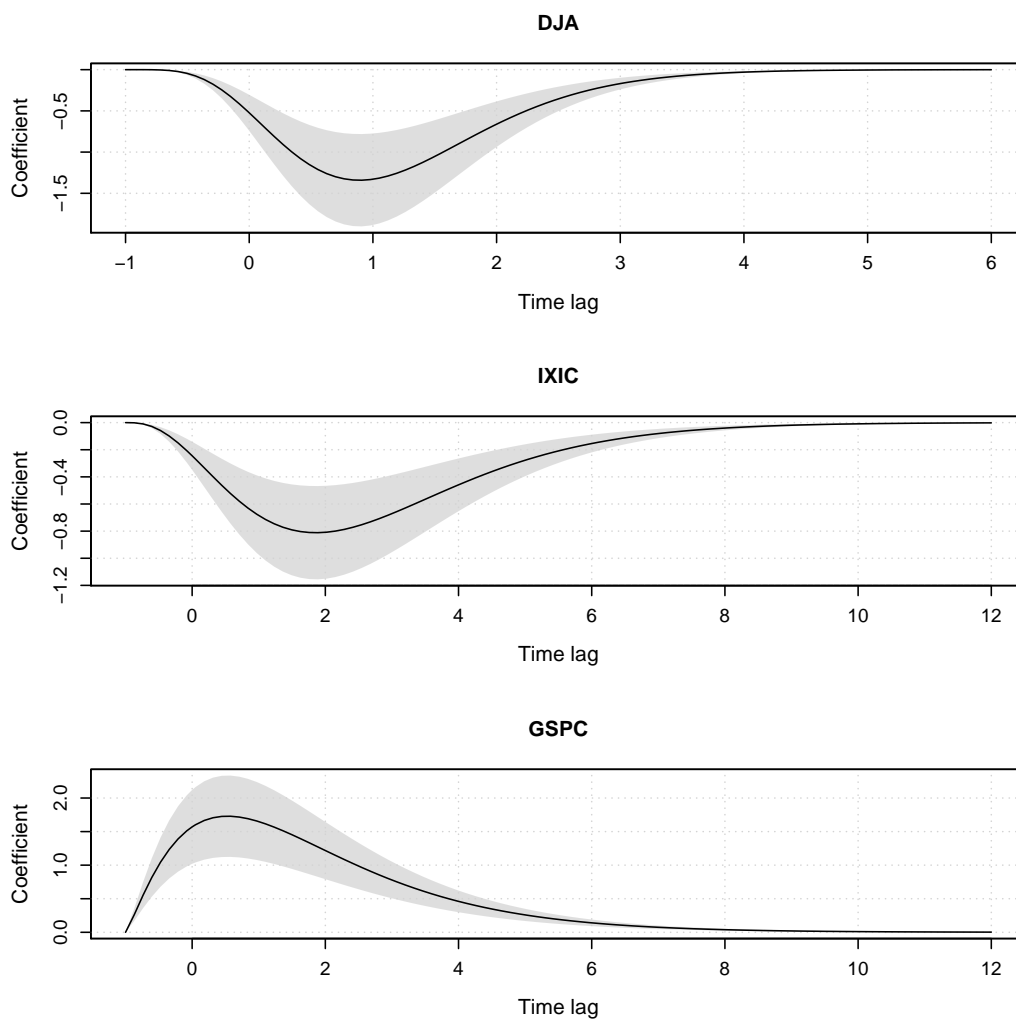


Figure 3: Estimated lag distributions with interpolation on non-integer time lags. Shaded regions indicate 95% asymptotic confidence bands.

infinite lag distribution, but it has found little popularity in empirical applications mostly due to the difficulty of performing parameter estimation in the case of multiple explanatory variables. Existing applications exploit OLS-based grid search, but such approach may result infeasible for more than two explanatory variables. Our algorithm guarantees only a local maximum, but, compared to OLS-based grid search, it makes maximum likelihood estimation feasible for an arbitrary number of explanatory variables, and can be restarted several times from different random initial values to hopefully find the global maximum.

In principle, the number of random restarts providing good chances to find the global maximum depends on the length of lag distributions and on the number of explanatory variables. The Monte Carlo experiment reported in this paper suggests that 100 random restarts are sufficient in the case of three lag distributions of length between 10 and 20. The experiment could be extended to longer lag distributions and to higher numbers of explanatory variables at the purpose of deriving rules of thumb for the choice of the number of restarts. A fruitful development of our algorithm is represented by the implementation of a sampling scheme generating overdispersed starting values, which could improve the effectiveness of the restart strategy.

Using our algorithm, constraints on peak and length of the lag distributions can be easily imposed by restricting the search on a subspace of the possible values of δ and λ parameters. This feature has a high applicative relevance, as it makes possible to take into account prior

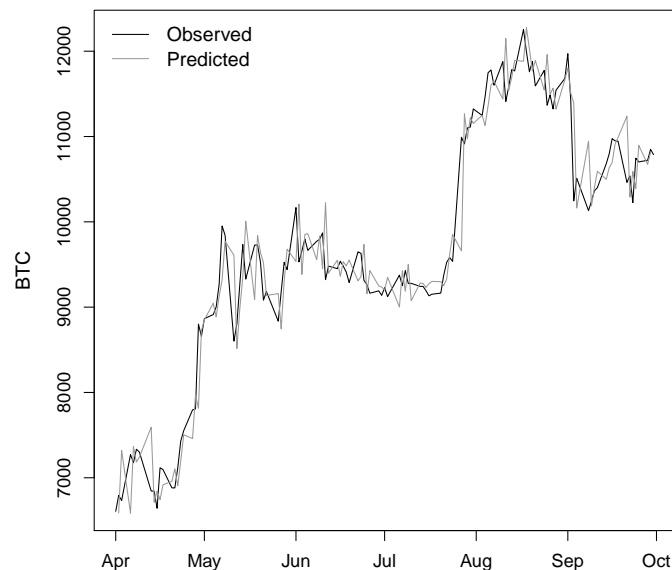


Figure 4: Observed (black line) versus predicted (grey line) values of BTC. The mean absolute percentage error of prediction is 2.8%.

knowledge in the estimation.

The Gamma lag distribution can be extended by adding a positive offset specified a-priori to represent a gestation lag, as in [Alston *et al.* \(2011\)](#). Our algorithm can be straightforwardly applied to the general situation where the offset is greater than zero, and such functionality is still implemented in our R package. Note that, specifying an offset greater than zero, the Gamma lag distribution can also be applied to the response itself in order to model the autoregressive effect.

Inference on θ parameters and on dynamic coefficients $\beta_{j,k}$ performed in the real-world application is conditioned to δ and λ parameters, i.e., it is assumed that they are known without uncertainty. As a consequence, the standard errors may be underestimated. This is a typical issue of estimation methods based on model selection, which clearly affects OLS-GS estimation and our hill climbing algorithm, as well. The problem was raised for the first time by [Schmidt \(1974\)](#), who proposed to derive standard errors from the full information matrix, but, to our knowledge, no empirical application of the Gamma-distributed lag model went beyond inference conditioned to δ and λ parameters. Block bootstrap methods ([Radovanov and Marcikić 2014](#)) represent a potential solution to correctly take into account uncertainty on δ and λ parameters. Unfortunately, the application of bootstrap techniques to our algorithm could be very time consuming, thus it has not been performed here, but we plan to direct future work towards its implementation.

Due to the flexibility of the Gamma lag distribution, our proposal may ease time series regression analysis involving multiple explanatory variables, for both assessment and predictive purposes. Also, our proposal naturally applies to dynamic simultaneous equation models.

Acknowledgements

We thank reviewers for valuable comments that helped improving the presentation of our work. The authors acknowledge the financial support provided by the "Dipartimenti Eccellenti 2018–2022" ministerial funds, Italy.

References

- Almon S (1965). "The Distributed Lag between Capital Appropriations and Expenditures." *Econometrica*, **33**(1), 178–196.
- Alston JM, Andersen MA, James JS, Pardey PG (2011). "The Economic Returns to U.S. Public Agricultural Research." *American Journal of Agricultural Economics*, **93**(5), 1257–1277.
- Andersen MA, Song W (2013). "The Economic Impact of Public Agricultural Research and Development in the United States." *Agricultural Economics*, **44**, 287–295.
- Andrews WK, Fair RC (1992). "Estimation of Polynomial Distributed Lags and Leads with End Point Constraints." *Journal of Econometrics*, **53**, 123–139.
- Balcombe K, Rapsomanikis G (2010). "An Analysis of the Impact of Research and Development on Productivity Using Bayesian Model Averaging with a Reversible Jump Algorithm." *American Journal of Agricultural Economics*, **92**(4), 985–998.
- Baldos ULC, Viens FG, Hertel TW, Fuglie KO (2018). "R&D Spending, Knowledge Capital, and Agricultural Productivity Growth: A Bayesian Approach." *American Journal of Agricultural Economics*, **101**(1), 291–310.
- Baltagi H, Ferry JR (1985). "Small Sample Results on Schmidt's Truncation Remainder to the Gamma Distributed Lag." *Statistische Hefte*, **26**, 87–95.
- Bervejillo JE, Alston JM, Tumber KP (2012). "The Benefits from Public Agricultural Research in Uruguay." *The Australian Journal of Agricultural and Resource Economics*, **56**, 475–497.
- Dhrymes PJ, Klein LR, Steiglitz K (1970). "Estimation of Distributed Lags." *International Economic Review*, **11**(2), 235–250.
- Dickey DA, Fuller WA (1981). "Likelihood Ratio Statistics for Autoregressive Time Series with a Unit Root." *Econometrica*, **49**(4), 1057–1072.
- Durbin J, Watson GS (1971). "Testing for Serial Correlation in Least Squares Regression III." *Biometrika*, **58**, 1–19.
- Granger CWJ, Newbold P (1974). "Spurious Regressions in Econometrics." *Journal of Econometrics*, **2**(2), 111–120.
- Hartley HO, Booker A (1965). "Nonlinear Least Squares Estimation." *The Annals of Mathematical Statistics*, **36**(2), 638–650.
- Jin Y, Hiuffman WE (2016). "Measuring Public Agricultural Research and Extension and Estimating Their Impacts on Agricultural Productivity: New Insights from U.S. Evidence." *Agricultural Economics*, **47**, 15–31.
- Jorgenson DW (1966). "Rational Distributed Lag Functions." *Econometrica*, **34**(1), 135–149.
- Koyck LM (1954). *Distributed Lags and Investment Analysis*. North Holland, Amsterdam, NL.
- R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, AT. URL <https://www.R-project.org/>.
- Radovanov B, Marcikić A (2014). "A Comparison of Four Different Block Bootstrap Methods." *Croatian Operational Research Review*, **189**(5), 189–202.

- Ratinger T, Kristkova Z (2015). “R&D Investments, Technology Spillovers and Agricultural Productivity, Case of the Czech Republic.” *Agricultural Economics – Czech*, **61**(7), 297–313.
- Russell JS, Norvig P (2003). *Artificial Intelligence: A Modern Approach*. 2nd edition. Prentice Hall, Upper Saddle River, US-NJ.
- Schmidt P (1974). “An Argument for the Usefulness of the Gamma Distributed Lag Model.” *International Economic Review*, **15**(1), 246–250.
- Schmidt P (1975). “A Note on the Treatment of the Truncation Remainder in the Gamma Distributed Lag.” *International Economic Review*, **16**(3), 800–801.
- Solow RM (1960). “On a Family of Lag Distributions.” *Econometrica*, **28**(2), 393–406.
- Thirtle C, Piesse J, Schimmelpfennig D (2008). “Modeling the Length and Shape of the R&D Lag: An Application to UK Agricultural Productivity.” *Agricultural Economics*, **39**, 73–85.
- Tserkezos D (1998). “GNP Quarterly Disaggregation of Selected OECD Countries Using a Related Time Series Approach.” *Applied Economics Letters*, **5**, 733–736.
- Tsurumi H (1971). “A Note on Gamma Distributed Lags.” *International Economic Review*, **12**(2), 317–324.
- Wang SL, Heisey PW, Huffman WE, Fuglie KO (2013). “Public R&D, Private R&D, and U.S. Agricultural Productivity Growth: Dynamic and Long-run Relationships.” *American Journal of Agricultural Economics*, **95**(5), 1287–1293.
- White H (1980). “A Heteroskedasticity-consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity.” *Econometrica*, **48**(4), 817–838.

Affiliation:

Alessandro Magrini

Dep. Statistics, Computer Science, Applications ‘G. Parenti’ – University of Florence

Viale Morgagni 59, IT-50134 Florence, Italy

E-mail: alessandro.magrini@unifi.it

ORCID: [0000-0002-7278-5332](https://orcid.org/0000-0002-7278-5332)