Optimize the log-periodic power law (LPPL) parameters using Differential Evolution for predicting stock price crisis

By Kushal Mondal
Roll No. 192IT008, M.Tech, 2nd Semester
Department of Information Technology

Submitted to
Dr. Biju R. Mohan

Introduction

LPPL(Log-periodic Power Law) models have been widely used to describe the behaviour of stock prices during an endogenous bubble and to predict the most probable time of crash or bubble formation, which is also called critical time. LPPL model fits with the logarithm of the dependent variable data and the best fit can indicate a critical time when a crash in the values of dependent variables may occur. In this problem the dependent variable is stock price. So, in effect here LPPL is used to predict a crash in stock price, thus answering the problem. I have taken reference from research papers written by Kwangwon Ahn et al, Vincenzo Liberatore and Daniel Traian PELE for design and implementation of the solution.

Dataset preparation

I have taken closing price data of HDFC bank dated from June 1, 2018 to June 4, 2020. There are 490 data points in the set and it includes a major crash in price during the COVID-19 time starting from March, 2020. This dataset is modified to include a column that serially numbers the date starting from 1 to 490. I have splitted the dataset into 2 parts. The first part consists of 70% of the data and it is used for training the model. The second part consisting of 30% of data is taken for judging the critical time predicted by the model. The modified dataset is included in the folder.

Dataset:

Training data:

Overview of approach

The LPPL model is fit into the training dataset to predict a crash in future. Following is the equation for LPPL model:

$$Y_t = A + B(t_c - t)^{\beta}\{1 + C\cos[\omega \ln(t_c - t) + \phi]\},$$

Here Yt is the log of stock price and t is the equivalent of timestamp in time series. Hence, t is the independent variable and Yt is the dependent variable. There are 7 parameters in LPPL which are A, B, β, C, tc, ω and ϕ. These seven parameters are optimized using Differential Evolution technique and for the optimized values the RMSE is calculated using the following formula:

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - Y_t)^2},$$

Here yt is the observed value, Yt is the predicted value and T is the number of observations. Among the parameters of LPPL tc gives us the critical time when a crash may occur. Which is then plotted with the dataset to justify its correctness.

Methodology and Implementation

Role of Optimizer: The optimizer used here is Differential Evolution optimizer which is a method in Evolutionary computing that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality . The optimizer takes the following parameters:

1. Objective function: RMSE on LPPL which needs to be minimized.
2. Lower bound of parameters: Lower bound of the 7 parameter. How to find the lower bound is discussed later.
3. Upper bound of parameters: It is set in proportion to the lower bound values after some trial and analysis.
4. Dimension: the number of parameters, here it is 7
5. Population size: Population of candidate solutions can be arbitrary but according to study standard value is 10 times the dimension. Hence it is 70.
6. Maximum iteration: Number of iteration before it stops. It is basically a stopping criteria and can be set arbitrarily.

Once the optimizer gets all these values it tries to optimize the value so that objective function goal is achieved. In other words it gives the parameters within the lower and upper bound for which it generates the lowest RMSE.

The crucial part of modelling LPPL is finding several sets of initial values for parameters and consequently the upper bound of them so that RMSE can be minimized. The goal is to get the value of critical time(tc) for which the RMSE is lowest (mentioned in *An LPPL algorithm for estimating the critical time of a stock market bubble by Daniel Traian PELE*). That critical time is the answer to the problem. Obviously the values of these parameters which includes tc are sensitive to the lower and upper bound set for them, therefore I used the following approach to find several sets of initial values (or lower bound) and then employ the optimizer to do its job.

Parameter initialization: In the paper named '*Forecasting Financial Crashes: Revisit to Log-Periodic Power Law'* the author has mentioned using peak values in the dataset along with a system of regression and price gyration to get initial values of parameters for LPPL. This approach employ different window sizes for finding the peaks and the approach is as below:

1. **Identify three consecutive stock price peaks, that is, i, j, and k for a window size W**: I have used an automatic peak finding algorithm that takes the window size W and for each datapoint Ti it looks to its left and right till the distance of W and declare Ti as peak if it is maximum among the values in window of both sides.
2. **Assign the distance-based weight to each peak:** It is achieved by using the values in the ti column in the dataset. It is strictly increasing and a peak in distance has bigger value than a peak that is closer.
3. **Estimate the initial values of tc, ω, and ϕ from the price gyration as $tc = \rho k - j / \rho - 1$, $\omega = 2\pi/\ln \rho$, and $\phi = \pi - \omega \log(tc - k)$ with $\rho = j - i/ k - j$:** Here the values are carefully calculated to avoid negative and zero values inside log function while calculating ϕ. For this I have omitted values of tc which are negative.
4. **Set the initial values of other two parameters, that is, β = 1 and C = 0.**
5. **Estimate the initial values of A and B using a regression algorithm on the equation $yt = A + B(tc - t)$.** Here, yt is the log of values of the closing price and t is time index in the dataset. I have used Linear regression and A, B are the intercept

value and coefficient value respectively. At this point I have a set of 7 parameters for one set of i,j,k in peak values(step 1) for a single window size W.

6. **Repeat the steps in 1 to 5 for each possible set of i,j,k and get several sets of parameters:** I have used three consecutive peaks as i, j, k. It can be taken in random fashion too, as stated by *Daniel Traian PELE* in his paper *'An LPPL algorithm for estimating the critical time of a stock market bubble'.*
7. **Find the LPPL parameters using the constrained optimization, DE, with the initial population of the parameters from step 6:** Here I have used DE optimizer and noted the rmse and tc values.
8. **Repeat steps 1 to 7 with changing the window size W:** For each window size as well I have noted the RMSE and tc values.

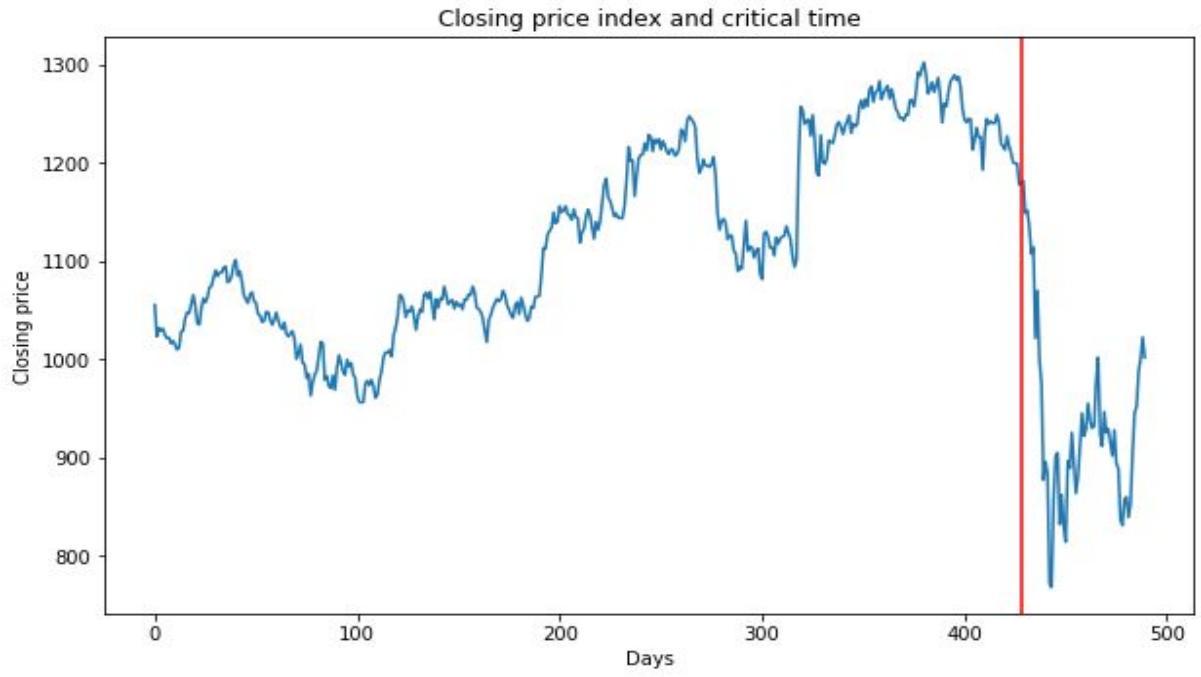
Among all the noted rmse values I took the minimum rmse and the corresponding tc value. This tc is the predicted critical time for a crash. The algorithm should be run multiple times to get minimum rmse since parameters keep changing.

Result and analysis

The following screenshot shows the best case of set of minimized rmse and tc values for varying window sizes and peak values:

Index	Window size	RMSE	tc
0	10	0.0450687	635.70673
1	12	0.0450771	629.5844
2	14	0.0307959	428.22945
3	16	0.0419637	895.98002
4	18	0.0359035	1425.2326
5	20	0.0358909	1453.2645
6	22	0.0357895	1438.3049
7	24	0.0359192	1410.0665
8	26	0.0359381	1414.785
9	28	0.035888	1411.5549
10	30	0.0359421	1417.5824
11	32	0.0358673	1409.9

The marked line indicates the minimum rmse and the corresponding tc. It has happened for window size 14. Hence 428th day has very high chances of a crash. Here I have increased the window size by 2 in each iteration. The following plot puts the tc value in the dataset:



The red line is the values of tc.

Meta heuristic process of parameter optimization is volatile and requires a lot of combination of factors to get a good value. I have tried several window sizes for parameters initialization and several sizes of population in optimizer on my dataset. On checking all the rmse values 0.030 is the least I have found and the corresponding tc for it provides a reliable result. The whole procedure should be run multiple times with several combinations of parameters tuning for the optimizer to get a good result.

A brief comparison of DE with GWO:

It should be observed that Differential Evolution optimization has performed better than the Grey Wolf optimization process. I have used 12 different window sizes in DE as compared to 24 in GWO to get the comparable lowest RMSE values. Though rmse is slightly better in DE. The tc values in both cases are comparatively similar. The population size, however, is smaller for GWO standing at 50 whereas in DE it is 70. Both optimizers hit the lowest rmse at similar window sizes hence similar peak values.

Both optimizers have a high density of tc values with good rmse in the interval of 1400 to 1450. In the paper '*An LPPL algorithm for estimating the critical time of a stock market bubble' by Daniel Traian PELE,* the author used the distribution of tc as an indicator of crash which can be interesting in this behavior on my data.

Reference

1. *Forecasting Financial Crashes: Revisit to Log-Periodic Power Law by Bingcun Dai, Fan Zhang, Domenico Tarzia and Kwangwon Ahn*
2. *An LPPL algorithm for estimating the critical time of a stock market bubble by Daniel Traian PELE*
3. *Computational LPPL Fit to Financial Bubbles by Vincenzo Liberatore*
4. *EvoloPy: An Open-Source Nature-Inspired Optimization Framework in Python by Ibrahim Aljarah, Hossam Faris, Seyedali Mirjalili*
5. *Differential Evolution optimization article*