Optimize the log-periodic power law (LPPL) parameters using GREY WOLF optimizer for predicting stock price crisis.

Dataset used : Daily data collected from National Stock exchange India from 2019-08-01 to 2020-05-14. Close price of N50 is used for evaluation.

Python programing language is used to Optimize the log-periodic power law (LPPL) parameters using Grey wolf optimizer for Predicting the stock price crisis. Libraries used: scipy's optimize library,numpy,pandas,matplotlib

First install the requirements using pip install -r requirements.txt

Methodology

How can we predict crashes in financial markets with LPPL

The bubbles seen in financial markets show some similarities in the way they evolve and grow. Log periodic power law can capture this particular oscillating movement. The ending crash of a speculative bubble is the climax of this Log-Periodic oscillation. The most probable time of a crash is given by a parameter in the LPPL equation. By fitting the Log-Periodic Power Law equation to a financial time series, it is possible to predict the crash.
A crash in the financial market can be explained as a sudden and dramatic decline of price of index in a very short duration. A crash may occur when the traders panic and sell their assets at the same time.
In 1996 two independent works done by Feigenheim && Freund and Sornette,Johansen && Bouchaud proposed that at the time of bubble the economic index increases as a power law with Log periodic oscillation leading to a critical point that describes the beginning of the market crash. The Log-periodic power law as a function of t is given by,

$$y(t) = A + B\left(t_c - t\right)^z + C\left(t_c - t\right)^z \cos\left(\omega \log\left(t_c - t\right) + \Phi\right)$$

where tc denotes the most probable time of crash, z is the exponential growth and ω controls ampitude of oscillations and A,B,C and ϕ are units carrying no structural details. When t approaches tc, the oscillations occur more frequently with decreasing amplitude.

Grey wolf algorithm

Grey wolf algorithm proposed by Seyedali Mirjalilia, Seyed Mohammad Mirjalilib, Andrew Lewis in 2014 is inspired by the leadership hierarchy and hunting mechanism of grey wolves in nature. Alpha, beta, delta, and omega are the four types of grey wolves in the hierarchy. The algorithm consists of 3 main steps of hunting such as searching for prey, encircling prey, and attacking prey.

Social hierarchy

Fittest solution is considered as alpha. Beta, gamma are second and third best solutions respectively. The remaining solutions are taken as omega. The optimization is guided by alpha,beta and gamma. Omega wolves follows these 3.

Encircling prey

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}$$

where t indicates the current iteration, A and C are coefficient vectors, Xp is the position vector of the prey, and X indicates the position vector of a grey wolf. A grey wolf can update its position inside the space around the prey in any random location by using these equations.

Hunting

In this step using the three best solutions obtained so far(alpha,beta,gamma) and we update the position of other search agents (including the omegas) according to the position of the best search agents.

Attacking prey

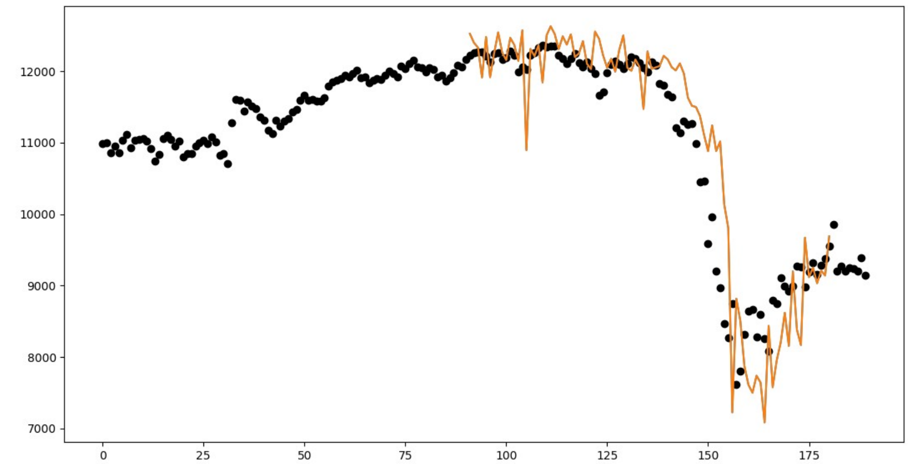
The fluctuation range of A is decreased by a . A is a random value in the interval [-2a, 2a] where a is decreased from 2 to 0 over the course of iterations.
When |A| < 1 wolves to attack towards the prey.

Implementation

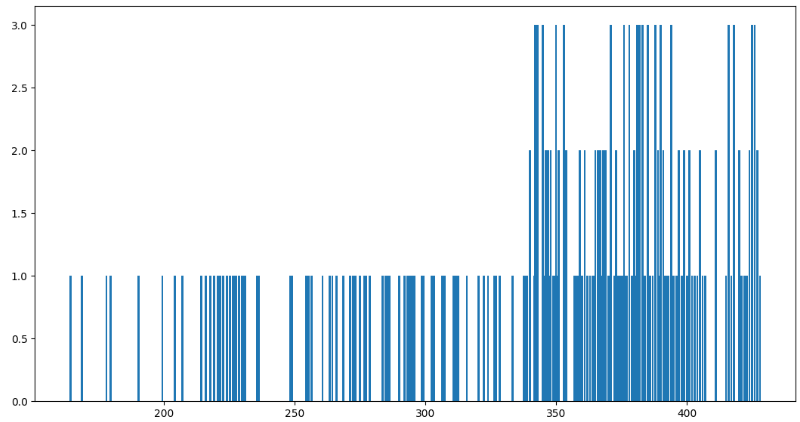
1. Initially choose the number of repeatations of experiment. To get meaningful statistical result NumOfRuns is chosen as 90. Along with that population size, number of iterations must also be passed to optimizer. Number of iterations for each run of optimization is taken as 300.
2. While initialising scores of alpha,beta,delta to inf , make the size of alpha_pos, beta_pos, delta_pos to be the dimension of LPPL parameter set.
3. Randomly choose the positions of all the search agents, but within the bounds specified for each LPPL fit function parameters.
4. At the start of every iteration return back the search agents that go beyond the boundaries of the search space.
5. Calculate objective function(LPPL) for each search agent in every iteration.
6. Based on the score obtained by fitness function update the value of alpha,beta,delta
7. Adjust the value of a by decreasing it linearly from 2 to 0.
8. At the end of each iteration update the positions of search agents .
9. After completing all the iteration return the position of alpha,beta,delta.
10. With the results(position of alpha,beta,delta along with time) from all the runs , estimate the fitness value for predicted parameters.
11. Get the upper/lower guesses (excluding the extream ones) of prediction and plot the result.

Results

Best prediction of stock prices



Critical time distribution



Critical time points

{353.54064986205117: 1, 168.52710035379948: 1, 226.2995618341215: 1, 264.4709502282683: 1, 380.6995890306016: 1, 283.5900077612652: 1, 231.221
86651894978: 1, 303.2270584846711: 1, 366.9384847826029: 1, 367.62508274004045: 1, 286.0019645793189: 1, 406.0711695129924: 1, 207.08629454295
294: 1, 350.0: 3, 387.91480781516947: 1, 299.25907612096955: 1, 312.10755862289363: 1, 254.51673103109997: 1, 312.7237703624803: 1, 214.437325
00187892: 1, 338.6026787335088: 1, 356.9899444532494: 1, 222.68384780098026: 1, 217.78187058435324: 1, 295.7426967808181: 1, 249.1605950785968
8: 1, 268.6828345947961: 1, 293.37478047254103: 1, 310.836084923601: 1, 229.77851657123696: 1, 322.2490499987625: 1, 302.4360619452444: 1, 405
.14480355193524: 1, 379.0649878038391: 1, 302.31875955867355: 1, 248.5146659757711: 1, 284.71944773739983: 1, 303.4503532134527: 1, 204.140948
40509145: 1, 350.70524751130046: 1, 214.36946436937163: 1, 286.3988421293933: 1, 190.27227092367625: 1, 274.94498619077996: 1, 294.67606551151
04: 1, 323.9871020419945: 1, 248.8324183903269: 1, 372.59672758491615: 1, 358.3574643168551: 1, 178.07173783977842: 1, 311.05150072345526: 1,
357.04130404594326: 1, 235.7262190186071: 1, 306.41039872379764: 1, 373.491745583007: 1, 168.56819987418135: 1, 358.2869127724375: 1, 260.6851
064268588: 1, 278.7761715838563: 1, 315.90324774118045: 1, 225.31649225542785: 1, 356.97870720629: 1, 381.93950529765533: 1, 289.9475623849346
: 1, 284.7113184740442: 1, 389.5476630516834: 1, 293.04156095250073: 1, 228.7282390627909: 1, 417.0: 1, 298.7789896300492: 1, 302.610899250842
7: 1, 333.40667882245214: 1, 385.0: 3, 256.4109518797183: 1, 216.07937790697852: 1, 418.0: 3, 263.3610498754852: 1, 374.233586635241: 1, 328.4
876169056877: 1, 311.7874223431008: 1, 273.249336517684: 1, 367.616158391897: 1, 293.82381335472263: 1, 199.47208180390984: 1, 272.42853918265
69: 1, 179.73178674283227: 1, 221.50308843732032: 1, 326.4437991674871: 1, 293.75446853870466: 1, 368.6676771689559: 1, 292.1072532316764: 1,
375.54257463759467: 1, 295.28584286148583: 1, 345.4162140170724: 1, 395.0: 1, 320.26985335766716: 1, 326.8048880852892: 1, 403.0: 1, 266.10444
95917862: 1, 379.1372873625608: 1, 339.0: 1, 277.3240557730644: 1, 342.0: 3, 343.0: 3, 345.0: 3, 346.0: 2, 347.0: 2, 348.0: 2, 349.0: 1, 322.4
547954755178: 1, 351.0: 2, 353.0: 3, 354.0: 2, 376.2436508522018: 1, 373.51892499799186: 1, 358.0: 1, 359.0: 2, 360.0: 1, 361.0: 2, 362.0: 1,
363.0: 1, 364.0: 1, 365.0: 2, 366.0: 2, 367.0: 2, 368.0: 2, 369.0: 2, 370.0: 1, 371.0: 3, 373.0: 2, 375.0: 1, 376.0: 3, 377.0: 1, 378.0: 3, 37
9.0: 1, 380.0: 2, 381.0: 3, 382.0: 3, 383.0: 3, 384.0: 1, 341.7020418460176: 1, 386.0: 1, 387.0: 1, 388.0: 3, 389.0: 2, 390.0: 3, 391.0: 2, 39
2.0: 1, 393.0: 1, 394.0: 3, 364.77841458832245: 1, 396.0: 1, 397.0: 2, 398.0: 1, 399.0: 2, 401.0: 2, 402.0: 1, 271.22482774373077: 1, 404.0: 1
, 405.0: 2, 407.0: 1, 400.22214322586086: 1, 227.11851805300248: 1, 411.0: 2, 320.206031391175: 1, 415.0: 1, 416.0: 3, 312.62992226019355: 1,
272.51523642186754: 1, 349.22019296972866: 1, 421.0: 1, 373.99643856691887: 1, 423.0: 1, 424.0: 2, 425.0: 3, 219.19212843015785: 1, 427.0: 2,
428.0: 1, 391.7966120609272: 1, 224.0584014582464: 1, 227.7353424530013: 1, 327.15318135444454: 1, 230.44762349610468: 1, 254.90093214293404:
1, 276.767564402009: 1, 307.27603948351197: 1, 394.1815054830383: 1, 284.7839229966382: 1, 255.4068542165802: 1, 400.7971071049842: 1, 220.769
0117110873: 1, 420.0: 2, 349.91625084085325: 1, 164.34603191053665: 1, 341.8635173818695: 1, 236.20092387725413: 1, 255.30114385223382: 1, 422
.0: 1, 337.78624891332794: 1, 345.40236317121844: 1, 276.5927677301671: 1, 322.4245558190833: 1, 372.2823479929357: 1, 285.13622724746733: 1,
347.7802705869287: 1, 340.0: 2, 426.0: 3}