Udacity MLND Capstone Project

Josh Nielsen

Project Overview

The LPPLS model provides a flexible framework to detect bubbles and predict regime changes of a financial asset (Sornette, 2013). A bubble is defined as a faster-than-exponential increase in asset price, that reflects positive feedback loop of higher return anticipations competing with negative feedback spirals of crash expectations. It models a bubble price as a power law with a finite-time singularity decorated by oscillations with a frequency increasing with time.

Here is the model and a description of its 7 parameters:

$$E[ln\,p(t)] = A + B(t_c - t)^m + C(t_c - t)^m cos(\omega ln(t_c - t) - \varphi)$$

where:

$E[ln\,p(t)]$	expected log price at the date of the termination of the bubble
t_c	critical time (date of termination of the bubble and transition in a new regime)
A	expected log price at the peak when the end of the bubble is reached at t_c
B	amplitude of the power law acceleration
C	amplitude of the log-periodic oscillations
m	degree of the super exponential growth
ω	scaling ratio of the temporal hierarchy of oscillations
φ	time scale of the oscillations

In this project I will use Keras to fit the 7 parameter LPPLS model to the S&P 500 index adjusted daily close time series data. Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, among other frameworks. We will be using TensorFlow. I will then compute a buy/sell indicator based on the model's parameters and compare a trading strategy against the S&P 500.

Problem Statement

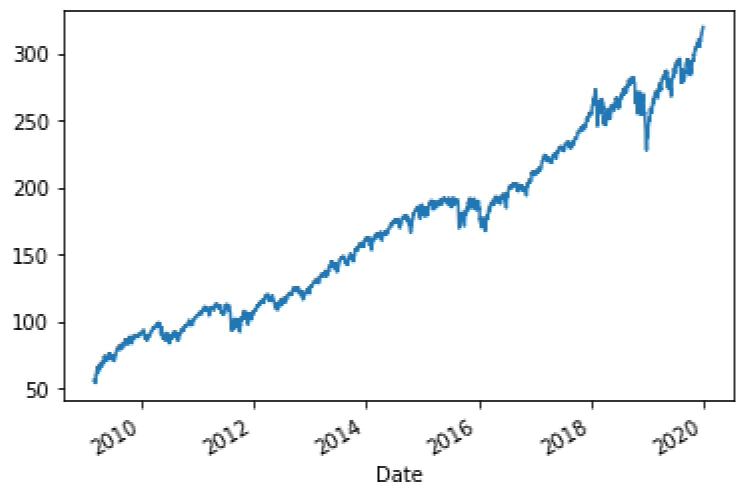
In the current academic literature, the LPPLS model is fit to data using an ordinary least squares method (Sornette, et al 2013). Using a neural network to fit a parametric function like the lppls model, has not been documented. I would like to present that approach here.

Metrics

To determine the viability of the neural network's fitting of the LPPLS model to the empirical data, I will measure the standard error of the loss function over the course many fits (8000). This will tell us if the model is converging on an optimum over time.

Data Exploration & Visualizations

I will use the adjusted daily closing price for the S&P 500 for a generalized overview and demonstration of the fitting procedure. The data was collected from yahoo finance. It can be retrieved here:
https://finance.yahoo.com/quote/SPY/history?period1=1235862000&period2=1577401200&interval=1d&filter=history&frequency=1d Any empty values are forward filled. See below for a visualization of the time-series data.



I will use the adjusted daily closing price for each stock in the S&P 500 for the trading strategy simulation. The data was collected from the IEX https://iexcloud.io/. The data is updated daily but I will only use data from 2015-04-29 to 2020-02-11. The dataset is stored in amazon aws s3 bucket here: https://boulderinvestmenttech.s3.amazonaws.com/iex_etl/data/agg/master.csv

Algos and Techniques

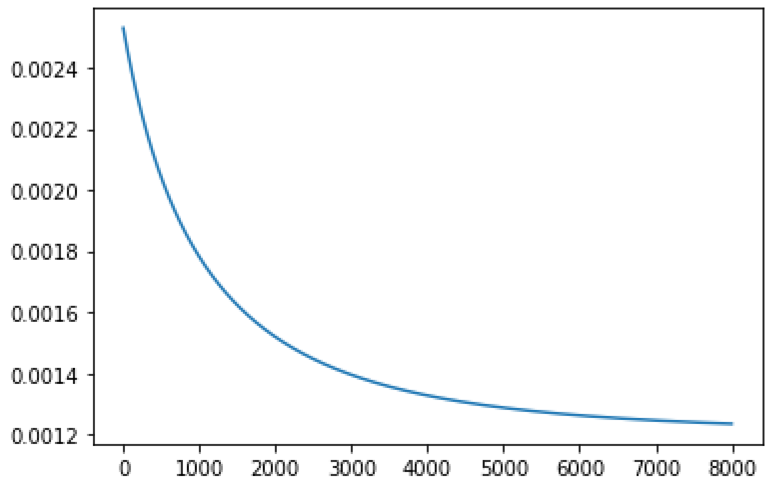
In Keras, each layer of a neural network takes an input tensor and multiplies it with a weight tensor and applies an activation function before the result is returned. Next, the result is compared to the label data (in our case the empirical S&P 500 price data) such that a loss function can be calculated and optimized. We will need to create a custom layer that takes an input tensor, then we can apply our objective/loss function and return the result. The objective/loss function will ensure we get the optimal fit to the empirical price data.

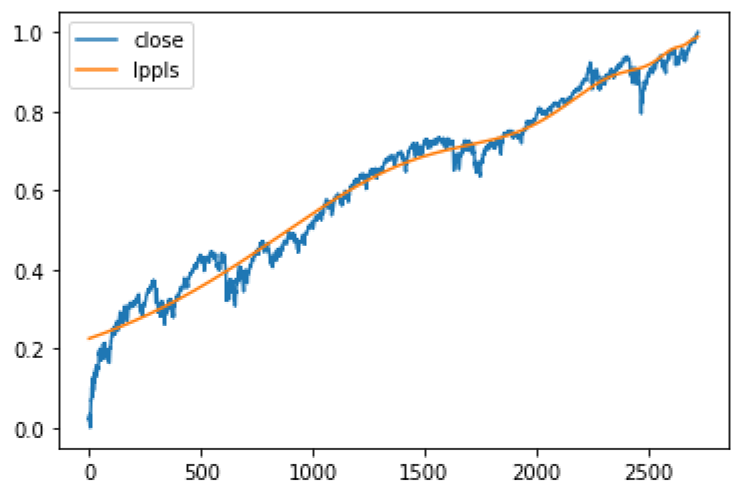
Benchmark

In the context of this project, it makes most sense to compare the portfolio constructed by the LPPLS indicators to a benchmark portfolio or a broad stock index. I will use the S&P 500 index.

Implementation

Earlier in this course I created a python package titled `lppls`. It implements an ordinary least squares fitting procedure for the LPPLS model. For this project I created a new branch within the lppls repository that attempts to fit the model via a neural network using Keras and Tensorflow. The fitting procedure can be viewed here:
https://github.com/Joshwani/mlnd/tree/master/Capstone
I used a custom layer from Keras that is capable of fitting any parametric function. In this specific project I fit the LPPLS model. Below are the loss function plots and the actual fit.

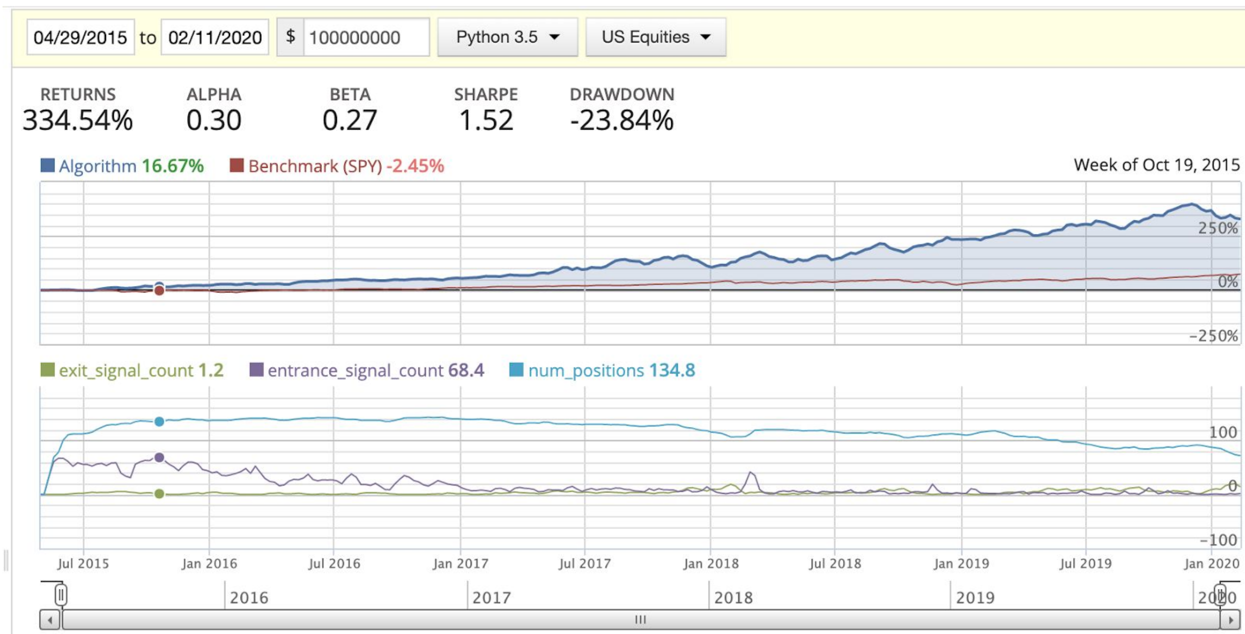




Next I needed to compute the DS LPPLS confidence indicator for the individual equities in the S&P 500. Computing the DS LPPLS Confidence indicator was quite computationally expensive. Crunching the numbers for the backtest cost ~80 USD using an AWS c5.metal EC2 instance with 96 vCPU and took ~20 hours. The model could probably benefit from greater vectorization/parallelization, but I digress; time and money are the reasons why I didn't run a longer initial test.

Results

To run the backtest, I output the historical DS LPPLS Confidence indicators in a specific csv format so as to comply with Quantopian's Self-Serve data upload. That way I could use it within their backtesting environment. Below are the results.



Refinement

An adagrad optimizer was chosen to be used in the model because it has parameter-specific learning rates, which are adapted inversely proportional to how frequently a parameter is updated. A learning rate of .11 was chosen because it appeared to perform best. More involved analysis of implementing various learning rates or a custom optimizer could improve the outcome.

References

Filimonov, V. and Sornette, D. A Stable and Robust Calibration Scheme of the Log-Periodic Power Law Model. Physica A: Statistical Mechanics and its Applications. 2013

Nhat Quang Pham Huu, Back-Testing of Trading Strategies Based on the Financial Crisis Observatory Output. 2018.

Tinatin Mamageishvili, Back-Testing of Trading Strategies Using Financial Crisis Observatory Output. 2019.

Sornette, D. Why Stock Markets Crash: Critical Events in Complex Financial Systems. Princeton University Press. 2002.

Sornette, D. Demos, G. Zhang, Q. Cauwels, P. Filimonov, V. Zhang, Q. Real-time prediction and
post-mortem analysis of the Shanghai 2015 stock market bubble and crash. Journal of Investment Strategies. 2015