Volatility Estimator

Zhaokang Yu

2020-08-08

Problem Description

I was given price data for approximately a year, sampled at 1 minute for 6 stocks and was asked to produce a fair estimator of the volaility over the next month.

Summary

Data cleaning is performed in a way to mitigate the negative impact of missing data or sudden price moves on volatility estimation, modelled via the covariance of an autoregressed return time series of high frequency. A volatility signature is then obtained for each trading day, producing a new data set (one sample per day) ready for PCA-based unsupervised learning method. The learning method produce estimated volatilities of different time horizon, from which we chose the longest to be a candidate extrapolated for even longer horizon. The model also provided us with the confidence interval based on the explained variance ratio by the PCA components.

Submissions:

See README.md for details of how to execute the program and visualize data.

- Python code base

1. Executable: src/main.py
2. Configuration: src/rv/datamodels/config.py
3. Data cleaning: src/rv/preprocessing.py
4. Calculation: src/rv/calculator.py

- Jupyter Notebook for analyais and visualization

1. visaulize_prices.ipynb
2. visualize_returns.ipynb

Data Characteristics

- Features

The given data only comprises of stock prices, thus containing very scarce features. I could have elected to create from prices more features such as lags, moving averages of different horizon but I deemed the created features too highly correlated to the prices themselves and more inclined to cause over-fitting.

- Frequency

The task is to estimate a monthly volatility which is for a long term horizon, whereas the given data is sampled every minute which is a very short term horizon and thus of much higher frequency. It would be inappropriate to resample the raw dataset and use the prices every month apart to calculate the estimated volatility as this method doesn't count for intraday/overnight session properly. Nor is it sensible to reduce the dataset by only keeping daily closing prices. Plus, the information of the dataset will be under-utilized if we decide not to exploit the higher frequency, 1-minute data. So I have decided to stick to the original sample frequency and try to extract useful information for the monthly volatility estimation. As I will elaborate in "Chossen Approach" section, 1-minute time series exhibits more stationariness than longer horizon such as daily or monthly , making them more suitable to be applied to the Random Walk Model.

- Price jumps

Some stocks clearly exhibit overnight price jump, which presumably were caused by corporate actions such as dividends, stock split or earnings announcment. To remove the impact of these actions adjustments were applied . More details in the data cleaning section.

Chosen Approach

- Data cleaning: Prices series are adjusted when overnight change exceeds certain relative threshold and the return is more than certain standard deviation away from the mean.
- Additionally I observed that some prices are marked as empty entries and others being zero. These prices are repaired by interpolating from its neighbouring prices.
- For some stocks (such as stock d), occasional outliers such as 1.0 are also observed and they are removed from the intraday analyais.
- Separation of intraday and overnight sessions: It's clear these stocks exhibit different intraday vs overnight variance distribution. Thus it's important to separate these two sessions so subsequent analysis is performed in a consistent manner, i.e. mainly focus on the returns from the intraday sessions. However, the variance distribution between intraday and overnight can be regardes as volatility time modulation so we need to calculate the day fraction of the intraday session over an entire 24-hour period, in a sense of representing the volatility time used to convert an intraday volatility

measure to a monthly volatility measure. Winsorization is used to avoid the impact from returns far away from the distribution bulk.

- Learning method: As mention in the Data characteristics section, the given dataset is composed of scarce features and lacks target values to be treated with supervised learning methods. So I have adopted the classical PCA-based unsupervised learning method. For each trading day, by applying Bachelier's first law on correlated returns calculated from 1-minute price time series, I obtain volatility as a function of the lag variable (in minute unit), the same varible one would use in autocorrelation function (ACF). This is known as the volatility signature plot for a stock. See Appendix A for math derivation details. I then sample this signature plot at certain typical lag values such as 1, 5, 10, 15, 30, 60, 120 and 195 minutes, to form one observation of the function output I would like to estimate. Repeating the same operation of all trading days I obtain the actual dataset to apply my learning method on. The dataset represents N observations of data in $\mathbb{R}^p$ where N is the number of trading days with valid data and p is the number of typical lag values I have chosen before. Consider a rank q linear model for representing them:

$$f(\lambda) = \mu + \mathbf{V}_q \lambda$$

- For this new dataset I apply PCA decomposition via SVD method. By choosing the first three components making up more than 95% of explained variance. That means $q = 3$.
- Estimator: Now the problem is reduced to choose the appropriate λ to be plugged in the function learnt from the model. For each observation i there is a corresponding optimal λ_i. I have chosen to apply an EWMA of a span of 22 trading days (corresponding to one calendar month).
- Confidence level: I chose to use components making up more than 95% of the explained variance so my confidence inverterval is 95%

Appendix A: Calculate volatility signature from returns

Assume that a price series is described by:

$$p_t = p_0 + p_{average} \sum_{u=0}^{t-1} r_u$$

Where the return series r_t is time-stationary with mean

$$\mathbb{E}[r_t] = 0$$

and variance

$$\mathbb{E}[r_t^2] - \mathbb{E}[r_t]^2 = \sigma_r^2$$

The price variogram $\mathcal{V}(\tau)$, defined by

$$\mathbb{E}[(p_{t+\tau} - p_t)^2]$$

links the return series and the volatility by the equation

$$\sigma^2(\tau) = \frac{\mathcal{V}}{\tau._{average}^2}$$

As a results, volatility as a function of the lag τ is given by:

$$\sigma^2(\tau) = \sigma_r^2\Big[1 + 2\sum_{u=1}^{\tau}(1 - \frac{u}{\tau})Correl_r(u)\Big] = \sigma_r^2 + +2\sum_{u=1}^{\tau}(1 - \frac{u}{\tau})Cov(r_t, r_{t+u})$$

References

- "Trades, Quotes and Prices", by Jean-Philippe Bouchaud, Jullius Bonart, Jonathan Donier and Martin Gould
- "The Elements of Statistical Learnings", by Trevo Hastie, Robert Tibshirani and Jerome Friedman