**L1, L2 Filter and Momentum Strategies**

Yuxiao Cheng

Ruocheng Zhang

Jinyang Li

Instructor: Daniel H Totouom Tangho

December 12th, 2019

**Contents**

**Abstract** 3

1. Introduction 4

2. Trend Filtering Approaches and Computational Aspects: L2, L1 4

2.1 Computational aspects of L2 Filter 4

2.2 Computational aspects of L1 Filter 5

2.2.1 The L1-T filter 5

2.2.2 The L1-C filter 5

2.2.3 The L1-TC filter 6

3. Stochastic Processes Simulation and Filtering for Simulated Data 7

3.1 Model 1: Straight trend lines with a white noise perturbation 7

3.2 Model 2: Step trend lines with a white noise perturbation 8

4. Cross Validation and Trend Filtering for S&P 500 Index 9

4.1 Cross Validation Procedure 10

4.2 Trend Filtering for S&P 500c 12

5. Trend Detection and Distribution of Return 12

5.1 Trend Detection 12

5.2 Distribution of the Conditional Standardized Return 15

6. Momentum Strategies and Backtest 17

6.1 Backtest Assumptions 17

6.2 Backtest Procedures 17

6.3 Momentum Strategies 17

6.4 Results 19

7. Conclusion 20

**Challenges and Improvements** 22

1.Wavelet Filter 22

2.The possible interval for optimal 22

**References** 23

Abstract

This paper studies L1 and L2 trend filtering methods. These methods are widely used in momentum strategies, which correspond to an investment style based only on history of past prices. Various implementation of L1 filtering and L2 filtering are discussed at the beginning in order to detect some properties of noisy signals, which is using penalty condition to obtain the filtered signal composed by a set of straight trends or steps. This penalty condition is implemented in a constrained least square problem and is represented by a regularization parameter which is estimated by a cross-validation procedure. Explicit applications to momentum strategies in S&P 500 and SSE Composite Index are also discussed in detail with appropriate uses of the trend detection and configurations. The paper is concluded by listing some issues to consider when implementing a momentum strategy on index and some challenges we met and we want to improve in the future.

Keywords: Momentum strategy, L1 filtering, L2 filtering, trend extraction.

# Introduction

Trend detection is a major task of time series analysis from both mathematical and financial point of view. The trend of a time series is considered as the component containing the global change which is in contrast to the local change due to the noise. In an investment perspective, trend filtering is the core of most momentum strategies developed in the asset management industry and the hedge funds community in order to improve performance and to limit risk of portfolios.

The efficient market hypothesis illustrates that financial asset prices fully reflect all available information. It is now commonly accepted that the prices may exhibit trends or cycles, which makes it reasonable to develop momentum strategies for asset management. The momentum strategy is only based on the historical prices. In our research, the trend following momentum strategy consists of buying (or selling) the S&P 500 if it is significant that the estimated price trend is positive (or negative).

Our research is organized as follows. In section 2, we illustrate four trend filtering approaches L1-T, L1-C, L1-TC and L2. In section 3, we simulate two stochastic processes and filter the processes using aforementioned four filtering approaches. In section 4, we illustrate approaches of cross validation in order to obtain the optimal parameter for each of the filter. In section 5, we do the trend detection and figure out the distribution of the conditional standardized return. In section 6, we develop the momentum strategy and conduct backtest of the strategy by comparing the return of utilizing the strategy with the benchmark return.

# 2. Trend Filtering Approaches and Computational Aspects: L2, L1

## 2.1 Computational aspects of L2 Filter

We consider a time series which can be decomposed by a slowly varying trend and a rapidly varying noise process:

In L2 filter (Hodrick-Prescott filter), the scheme consists to determine the trend by minimizing the following function:

Let , and the D operator is the matrix:

The exact solution of this estimation is given by

## 2.2 Computational aspects of L1 Filter

### 2.2.1 The L1-T filter

The scheme for L1-T filter is as following:

It can be solved by considering the dual problem which is a QP program. Let , the problem can be rewritten as:

We construct the Lagrangian function with dual variable :

for . According to the Kuhn-Tucker theorem, the problem is equal to the dual problem:

The solution is:

### 2.2.2 The L1-C filter

The scheme for L1-C filter is as following:

or in the vectorial form:

where D operator is matrix:

The solution is similar to L1-T filter:

### 2.2.3 The L1-TC filter

The scheme for L1-TC filter is as following:

or in the vectorial form:

where operator is of L1-C filter and operator is of L2 filter. Let and , the problem can be rewritten as:

We construct the Lagrangian function with dual variable :

for -. Let and , we obtain:

with D=, , and =.

The solution is:

# Stochastic Processes Simulation and Filtering for Simulated Data

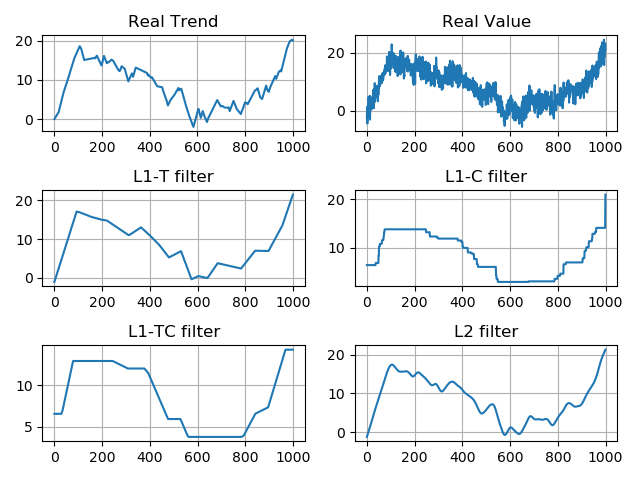
In this section, we simulate two models of stochastic processes. In each one of the model, the simulation consists of both the trend and the noise. First of all, we extract the actual trend of the simulation and plot it in time series, then add noise to the trend and plot it. Next, we filter the trend of the simulated process with noise using the aforementioned filtering approaches (L1 and L2) and plot the estimated trend. By comparing the outcomes of the estimated trend with the actual trend, we can easily acknowledge which one of the filter processes higher accuracy and thus we are able to obtain a general idea on the choice of the filtering approach towards different models.

## 3.1 Model 1: Straight trend lines with a white noise perturbation

In Model 1, we construct a stochastic process which consist of data simulated by a set of straight trend lines with a white noise perturbation:

We present the simulated data and filtered trend in Figure 1[[1]](#footnote-1). The left top graph is the real trend (signal) for simulation and the right top graph is the sum of the trend and the noise (noisy signal) for simulation. The following four graphs[[2]](#footnote-2) represent the trend of noisy signal by four filtering approaches. From the outcomes of the trend, we can build up a general idea that L1-T and L2 filters perform better than L1-C and L1-TC filters for Model 1, as they are more similar to the real trend.

**Figure 1: L1-T, L1-C, L1-TC, L2 Filtering for Model 1**

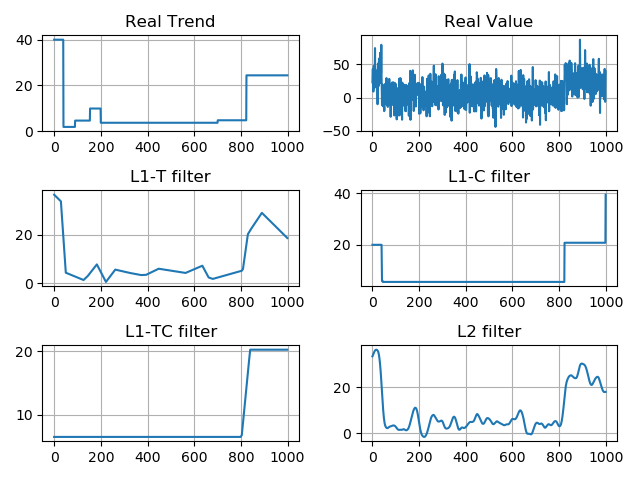


## 3.2 Model 2: Step trend lines with a white noise perturbation

In Model 2, we construct a stochastic process which consist of data simulated by a set of step trend lines with a white noise perturbation:

We present the simulated data and filtered trend in Figure 2[[3]](#footnote-3). The left top graph is the real trend (signal) for simulation and the right top graph is the sum of the trend and the noise (noisy signal) for simulation. The following four graphs[[4]](#footnote-4) represent the trend of noisy signal by four filtering approaches. From the outcomes of the trend, we can build up a general idea that L1-C and L1-TC filters perform better than L1-T and L2 filters for Model 1, as they are more similar to the real trend.

F**igure 2: L1-T, L1-C, L1-TC, L2 Filtering for Model 2**

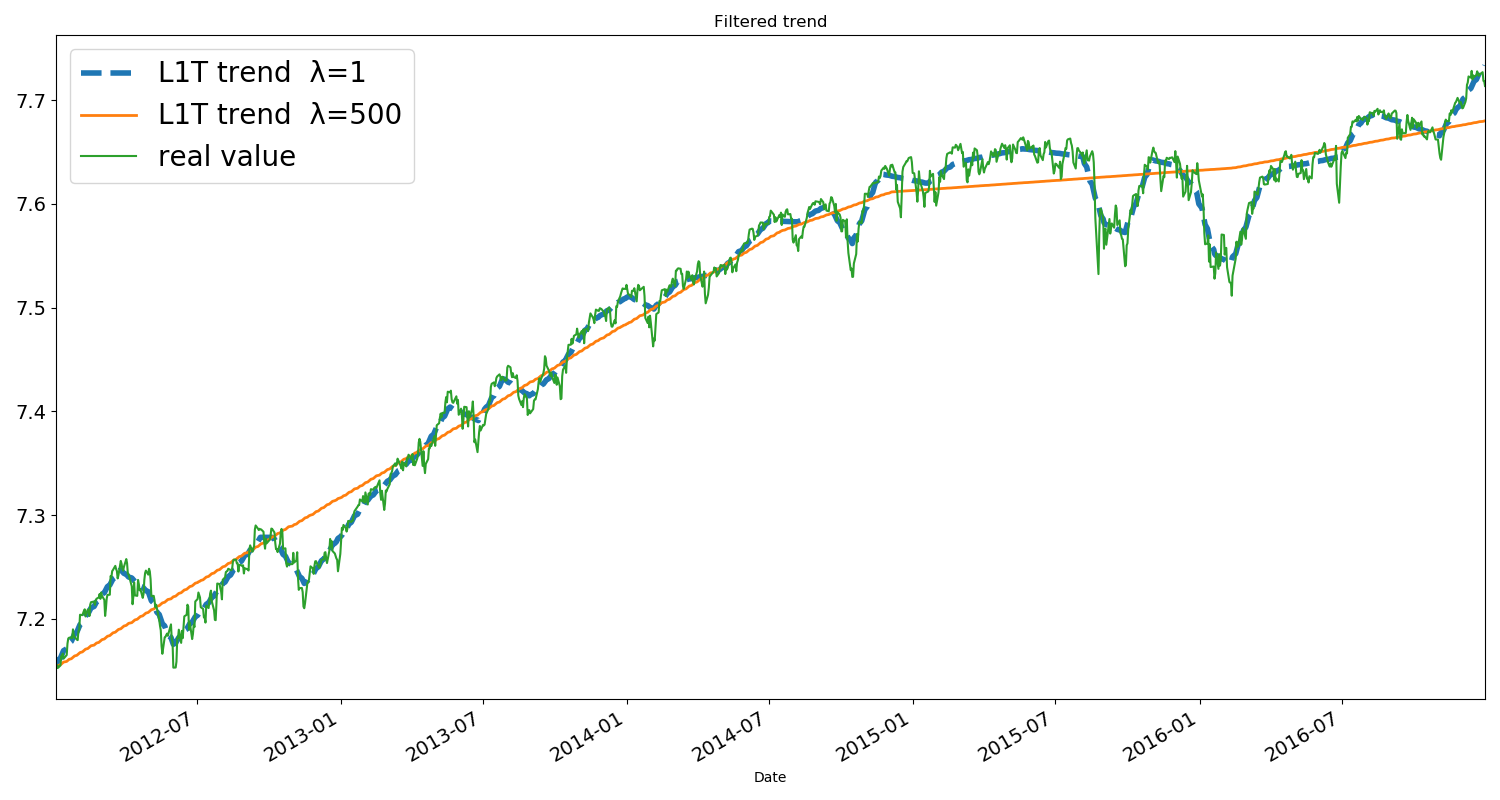


We will conduct trend filter and momentum strategies on S&P 500 index in the later sections and the stock price is a stochastic process which is similar to Model 1 we have discussed. This will be checked by computing the error of different filters for S&P 500 later.

# Cross Validation and Trend Filtering for S&P 500 Index

Previously, we assume that the parameter for filter L1-T and L1-C 5, for filter L2 is 1000. In this section, we illustrate the method of cross validation based on the historical data to determine an optimal for each filter. Before cross validation, let’s have a look at the trend of S&P 500 index from 2012 to 2016 using L1-T filter. We take and in this case. It is obvious that when is very small, the filtered trend would be over-fitted with too much noise, while when is very large, the filtered trend would be too smooth to fit the original data.

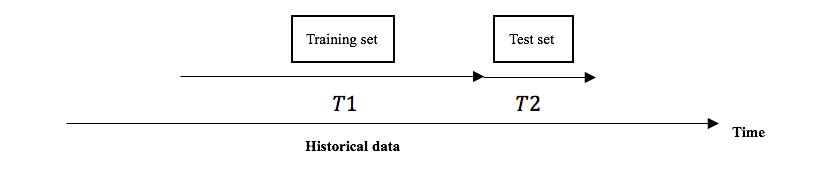
**Figure 3: L1-T Filtered Trend for S&P 500 from 2012 to 2016**



## 4.1 Cross Validation Procedure

We define two parameters T1 and T2 which characterize the trend detection mechanism. The first parameter T1 denotes the length (time period) of the training set of original data and T2 denotes the length (time period) of the test set. The cross validation procedure is conducted for determine the optimal .

**Figure 4: Cross Validation Procedure**



We implemented the the cross validation in the codes below. In practice, the optimal for L1-T and L1-C (for L1-TC is a combination of and ) is between 0 and 20, and the optimal for L2 is quite big. Thus, we use (0,20) as the original interval for L1 filter and utilize exponential method to find a such interval for L2 filter. Based on the original interval, we conduct cross validation to figure out optimal . We set the interval of cross validation as 250.

**Cross Validation Implemented in Codes**

**def cross\_validation(df,lower\_bound,higher\_bound,step\_length,leng\_of\_training,leng\_of\_test,type):**

**Interval=250 #denotes the length of interval in each round of validation**

**error = []**

**for delta in np.linspace(lower\_bound,higher\_bound,step\_length): # delta denotes**

**oneerror = 0**

**for count in range(1, 9):**

**x1 = df["Close"][(Interval \* (count - 1)):(Interval \* (count - 1) + leng\_of\_training)]**

**# x1 denotes the training set**

**if type=="L1T": filtered = l1tf(x1, (delta)) # cross validation for different filters**

**elif type=="L1C": filtered = l1ctf(x1, (delta))**

**else :filtered = hp(x1, (delta))**

**estimate = filtered[(leng\_of\_training-leng\_of\_test): leng\_of\_training]**

**# y1 denotes the test set**

**y1 = list(df["Close"][(Interval \* (count - 1) + leng\_of\_training):(Interval \* (count - 1) + (leng\_of\_training+leng\_of\_test))])**

**newerror = [pow(y1[i] - estimate[i], 2) for i in range(0, len(y1))]**

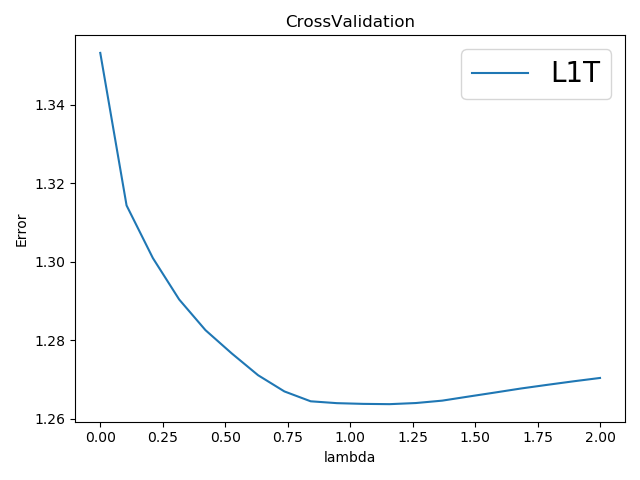
**oneerror = oneerror + sum(newerror)**

**error.append(oneerror) # when varies, error varies, we choose which minimize error**

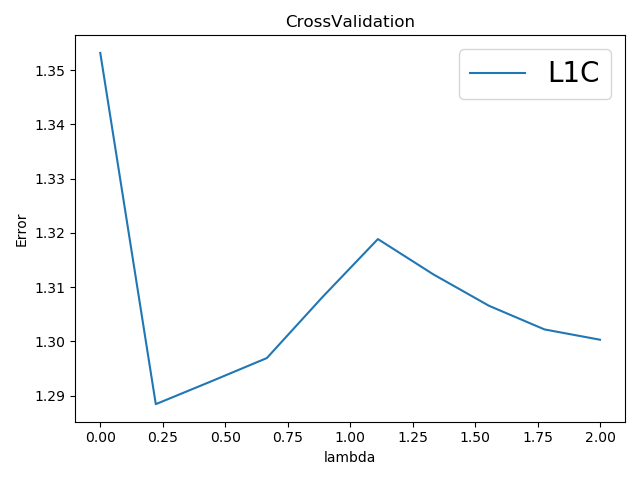
**return np.linspace(lower\_bound,higher\_bound,step\_length)[error.index(min(error))] # optimal**

The follow three figures[[5]](#footnote-5) show the relationship between and error[[6]](#footnote-6) of different filters. The x axis denotes the index of error array in the program and y axis denotes the error. We can easily figure out the optimal which minimizes the error in each one of the filter.

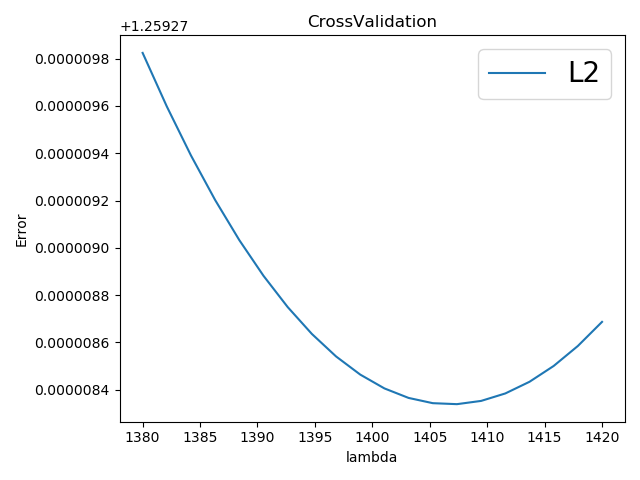
**Figure 5: versus error for filter L1-T**



**Figure 6: versus error for filter L1-C**



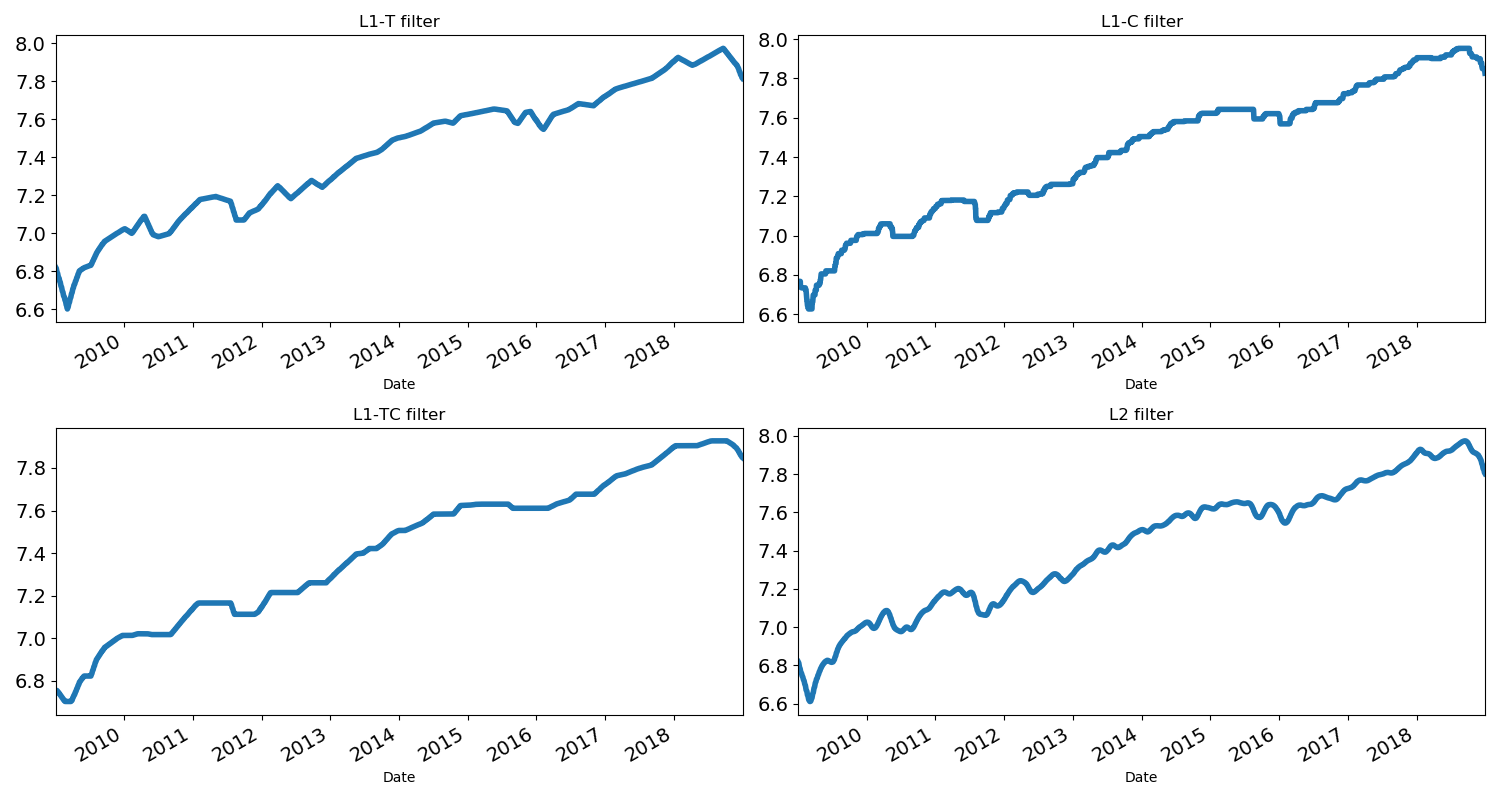
**Figure 7: versus error for filter L2**



## 4.2 Trend Filtering for S&P 500

We conduct cross validation on S&P 500 index from 2010 to 2018 and obtain optimal for four filters. The optimal for filter L1-T, L1-C and L2 are 1.158, 0.222 and 1407.368. We filter the trend using the optimal and get the figure below.

**Figure 8: Cooperation of Filtered Trend for S&P 500 from 2010 to 2018**



# Trend Detection and Distribution of Return

In this section, we will introduce the method of detecting the trend. We apply the aforementioned filtering approaches and the trend detection to historical data of S&P 500 (year 2018) to see the trend and the its significance of the stock prices, which are essential for our momentum strategies in the next step. In addition, we figure out the distribution of the conditional standardized return of the stock.

## 5.1 Trend Detection

Previously, we filtered the trend of stochastic processes. However, in some cases the trend may not be significant, so we first need to detect whether there is a significant trend. The method of trend detection is as following:

Considering the following statistic:

With if andif

We obtain[[7]](#footnote-7):

It can be shown that:

The normalized score should be:

takes the value +1(or -1) if we have a perfect positive (or negative) trend. If there is no trend, then . Under this null hypothesis, we obtain:

with

In our program, we create the function “*trend\_detection*” to detect if there is a significant trend at each point of the S&P 500 index from 2009 to 2018 by calculating and ploting the normalized score .

**Trend Detection Implemented in Codes**

**def trend\_detection(n,dataset,average\_filtered\_derivative[[8]](#footnote-8)):**

**n = 10**

**# judge whether there is a trend**

**st = [0] \* (len(dataset))**

**for date in range(n, len(dataset)):**

**total = 0**

**for i in range(0, n - 1):**

**for j in range(i + 1, n):**

**value = judge(dataset["Close"][date - i], dataset["Close"][date - j])**

**total = total + value**

**st[date ] = total**

**st\_normal = list(map(lambda num: num \* 2 / (n \* n + 1), st))**

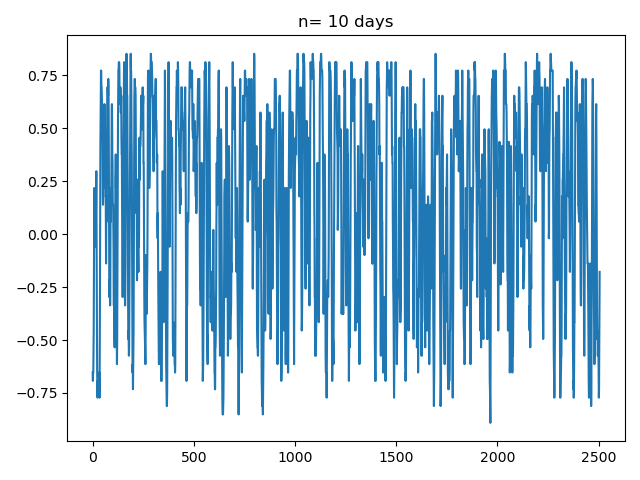
**std = math.sqrt((n \* (n - 1) \* (2 \* n + 5)) / 18)**

**zt = list(map(lambda num: num / std, st))**

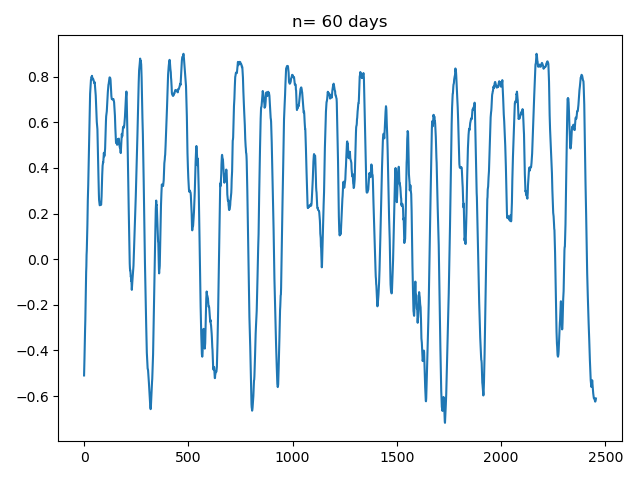
**print(len(list(filter(lambda num: num > 1.645 or num < -1.645, zt))) / len(df))**

The follow figures show the outcome of trend detection for S&P 500 from 2009 to 2018. We compare the results by setting n=10, 60, 100 and make a plot of trend detection versus L1-T filter when n=10. From the graphs and frequency table, we can conclude that we tend to reject the hypothesis that there is no trend when we consider a long period of time.

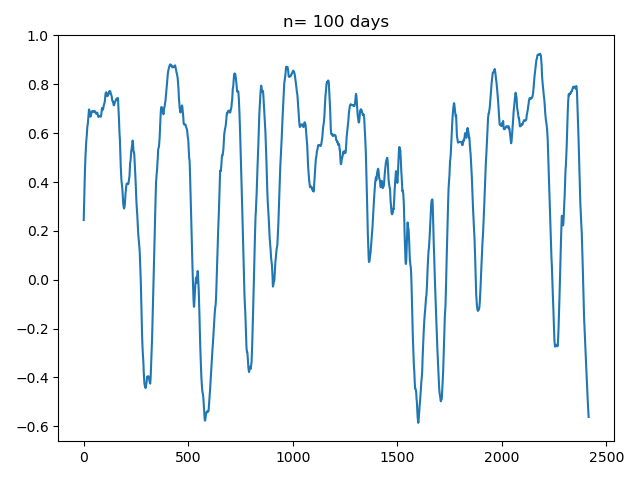
**Figure 9: Trend Detection for the S&P 500 from 2009 to 2018 (n=10 days)**



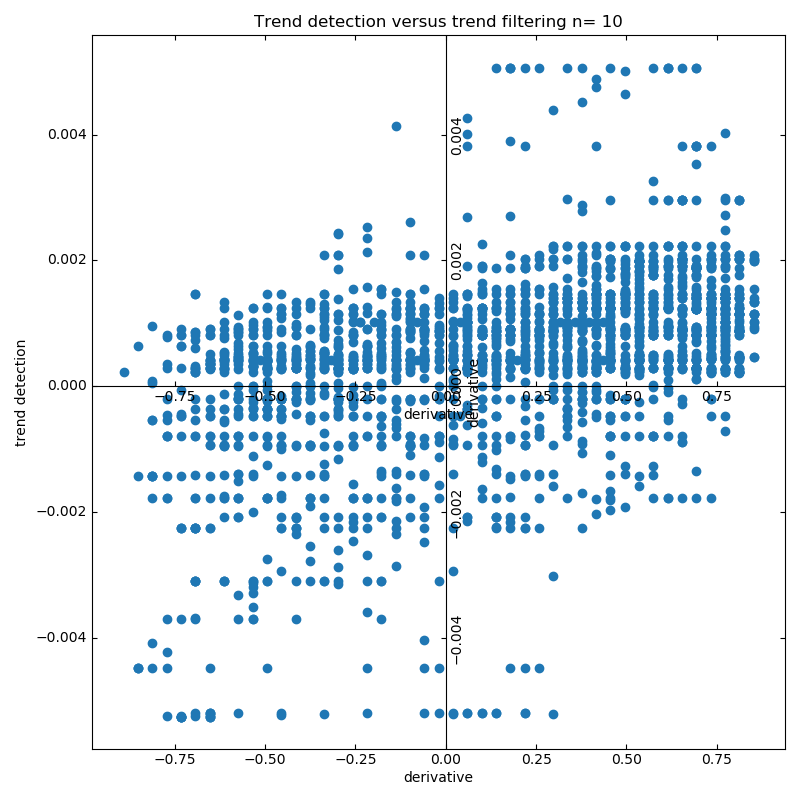
**Figure 10: Trend Detection for the S&P 500 from 2009 to 2018 (n=60 days)**



**Figure 11: Trend Detection for the S&P 500 from 2009 to 2018 (n=60 days)**



**Figure 12: Trend Detection Versus L1-T Filter (n=10)**



**Frequencies of Rejecting the Null Hypothesis with Confidence Level**

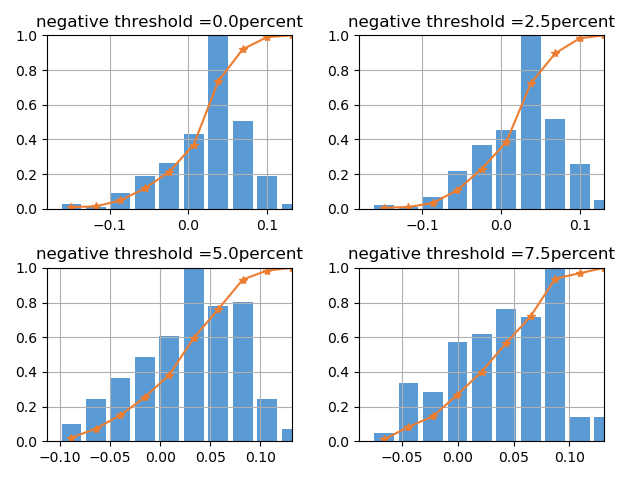
|  |  |  |  |
| --- | --- | --- | --- |
| Confidence Level | 90% | 95% | 99% |
| n=10 days | 59.98% | 49.92% | 38.39% |
| n=60 days | 88.12% | 85.33% | 82.07% |
| n=100 days | 87.88% | 85.93% | 84.49% |

## 5.2 Distribution of the Conditional Standardized Return

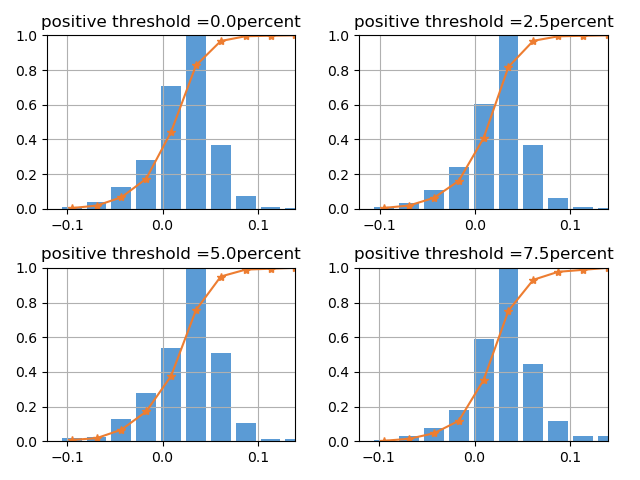
The persistence of trends is tested here in a simple framework for major financial indices.

For each of these indices the average one-month returns are separated into two sets. The first set includes one-month returns that immediately follow a positive three-month return, the second one follows a negative three-month return. Figure 13 illustrates the case of the S&P 500 index with positive trend and Figure 14 shows the negative case. The cumulative distributions of these two sets are shown. In the first quadrant we consider the distribution of the one-month returns following a three-month return above 0 or below 0 in two figures. In the second quadrant, we consider the distribution of one-month returns following a three-month return above 2.5% and, on the other hand, the distribution of returns below -2.5%. The same procedure is repeated in the other quadrants, for a 5% and a 7.5% threshold.

**Figure 13:**



**Figure 14:**



The figures above show that S&P 500 performed well from 2009 to 2018. Whatever the past three-month return was, the following one month could have a large probability (about 70 percent) to achieve a positive return. So, we cannot say the trend of S&P 500 is very persistent since if it has a negative return before, it is still very possible to reverse to positive one in the future.

# Momentum Strategies and Back test

## 6.1 Backtest Assumptions

In the back test section, we have the following major assumptions:

1. We first hold 100000 cash and we invest these cash in the underlying we want.
2. There is no subsequential financial crisis within the back test years.
3. The underlying market is the only market that we may invest.
4. We hold the stock for a constant period.
5. Every time we invest all the money in each trade.

f. There is no transaction cost.

g. When we short we only short the amount of our cash/price of underlying

h. We assume we can buy fractions of shares, E.g. 1.5 shares

## 6.2 Backtest Procedures

We go through the following steps:

1. Select one filter to figure out the trend.
2. Apply cross validation to find the best for the model.
3. Use filter to derive the average derivative and judge if there is a trend.
4. If there is trend we do the decision based on momentum explored before and average derivatives mentioned above.

## 6.3 Momentum Strategies

The strategy we apply is straight forward and naïve. First we detect if the underlying has momentum, we continue only if we see the momentum is significant enough. Then we calculate two indicators to make our judgment regarding to sell or buy. The first indicator is the filtered average derivatives on one single day, the second indicator is the trend indicator which would show if that single day has trend or not. When there is positive (or negative) momentum, which means there is high empirical probability that a period of increase (or decrease) will lead to an increase (or decrease), we may want to buy (or sell) the underlying if we see the positive (or negative) trend and positive (or negative) derivative. Based on this judgment list, now each day in the forecast set could has its buy or sell or do-nothing indicator which is predicted by the data which are supposed to be known.

Consistent with the assumption, we invest 100000 on the underlying market, we choose to long or short the underlying for a predetermined constant period T, which means we only make judgment every T days and can do no transactions within the time interval even if there is buy or sell indicators between.

Now the only challenge remains is to determine what is the best holding period. A loop is run from 1 to 30 days for this. The similar technique of cross validation is applied here to decide the best holding period but the difference is we use the mean best holding period as the parameter for the forecast set.

def Best\_Period(new\_close\_price, average\_filtered\_derivative, confidence):

Max=0

Best\_Holding\_Period=0

for i in range(1,30):

# 0 is the starting date below

# confidence is a 0,1 list which indicates if the trend is significant

Action\_list = strategy(average\_filtered\_derivative, 0, i, confidence)

Final\_revenue = calculate\_revenue(Action\_list, new\_close\_price, i)

if Final\_revenue > Max:

Max = Final\_revenue

Best\_Holding\_Period = i

print("Best holding period:"+str(Best\_Holding\_Period)+"return:"+str(Max))

return Best\_Holding\_Period

The revenue is computed by the absolute cash amount which means if we choose to long the stock, R= shares \*(price of the stock after T days – price of the stock now). Cumulating all the revenue we have the final revenue.

**Standardized Score Implemented in Codes**

**#updating zt  
n=60  
size=len(close\_all)  
st = [0] \* size  
for date in range(n, size):  
 total = 0  
 for i in range(0, n - 1):  
 for j in range(i + 1, n):  
 value = judge(close\_all[date - i], close\_all[date - j])  
 total = total + value  
 st[date] = total  
std = math.sqrt((n \* (n - 1) \* (2 \* n + 5)) / 18)  
zt = list(map(lambda num: num / std, st))  
confidence=[0]\*size  
for i in range(len(zt)):  
 if(zt[i]>1.96) :confidence[i]=1  
 elif (zt[i]<-1.96):confidence[i]=-1**

The Benchmark is set as hold the stock for all the period, the revenue of holding the benchmark is the difference of the portfolio value for the starting time and ending time.

## 6.4 Results

The result is mush as expected, we now use ^GSPC, MSCI, and [000001.SS](file:///C:\Users\yuxiao1997\Documents\WeChat%20Files\railgun0007\FileStorage\File\2019-12\000001.SS) to illustrate when the strategy work and when it won’t work.

The project is using the stock data from 2015-01-04 to 2018-12-31 as the training and test set, and the whole year data of 2019 as the forecast set.

For ^GSPC and MSCI, they share the same pricing pattern, almost increasing consistently from 2015 to 2019, which means they probably increase no matter there is a decrease or increase before. Thus it has no momentum. Imposing the momentum strategy, we find that it never runs better than just hold the position because once we trade on period, under the assumption, if no trend is detected after one period, we liquidate the position and do nothing for the next period. So chances are that we may lose some potential opportunities for the upward trend in these two case. However, if we only focus on average derivatives but no trend detection, we might be able to win over the benchmark because of the high frequency of positive average derivatives.

For SSE Composite Index, we see more up and downs and there is no obvious trend for both sides. After doing the momentum analysis, we detected the positive momentum on ten days level and earn profit of 10000 over benchmark for each 100000 cash we put into the market.

There are plenty of limitations in this strategy. For example, the time level of momentum could be explored further to see what is the best and our restriction is too strict like there is no transaction cost and no money market. We are using the average best holding period for the forecast year which is not rigorous and making much sense.

# Conclusion

Momentum strategies are efficient ways to use the market tendency for building trading strategies. Hence, a good estimator of the trend is essential from this perspective. Of course, the selection of the trend filtering method may lead to a single procedure or to a pool of methods based on the feature of the dataset. In this paper, we show that we can use L1 and L2 filters to forecast the trend of the market index in a very simple way. We also propose a cross-validation procedure to calibrate the optimal regularization parameter where the only information to provide is the investment time horizon. Finally, we consider several back tests for different filters on the S&P 500 and SSE Composite Index and obtain competing results. Intuitively, an investor should buy assets with positive return forecasts and sell assets with negative forecasts following the Momentum strategies. But the result in our test shows that if the index keeps rising during several years, the momentum strategies may not perform better than just holding it in the hand.

This paper does not consider very clearly about the momentum strategy, for example the size of each long or short position and how to deal with money when strategy asks not to buy or sell index. This paper also does not take into account anything about volatility and risk. Actually, individual risks, their correlations and the expected return should be considered to design strategies.

These are just a few questions relating to trend following strategies. Many more arise in practical cases, such as execution policies and transaction cost management. Each of these issues must be studied in depth, and re-examined on a regular basis. This is the essence of quantitative management processes and we want to improve this part in the future.

# Challenges and Improvements

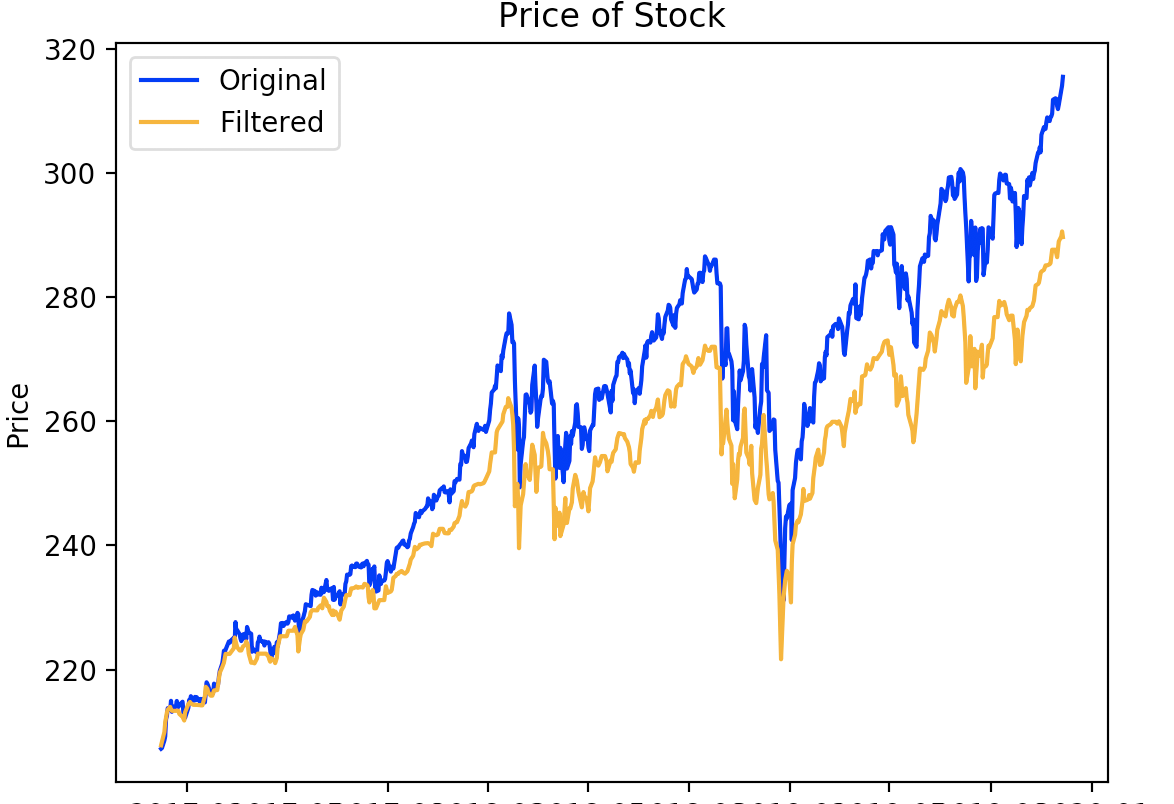
## 1.Wavelet Filter

We extend to the Wavelet Filter to get the trend of S&P 500 from 2017 to 2019. This corresponds to the wavelet analysis which consists of adopting a double dimension analysis, both in tine and frequency. The method of denoising can be divided into three steps:

Firstly, compute the wavelet transform of the original signal to obtain the wavelet coefficients ; Secondly, modify the wavelet coefficients according to a denoising rule D: ; Thirdly, convert the modified wavelet coefficients into a new signal using the inverse wavelet transform .

However, after implementing the wavelet filter upon data using python functions, we find out that the denoising procedure performs bad as the trend contains too much noise in figure. Thus, we need to think about the reasons of failure and further discuss how to make the trend smother.

**Figure 15: The Trend of S&P 500 Using Wavelet Filter (2017-2019)**



## The possible interval for optimal

We conduct cross validation to figure out the optimal which minimizes error for each filter. However, the original interval of is based on the practical experience, for L1-T and L1-C filter, the interval is (0,10) and for L2 filter, the value should be quite large. We did not find a way to compute the interval based on mathematics. Thus, we can improve the method of choosing the interval in further studies.

# References

[1] Benjamin Bruder, Tung-Lam Dao, Jean-Charles Richard, Thierry Roncalli (2011), Trend Filtering Methods for Momentum Strategies, University of Evry.

[2] Beveridge S. and Nelson C.R. (1981), A New Approach to the Decomposition of Economic Time Series into Permanent and Transitory Components with Particular Attention to Measurement of the Business Cycle, *Journal of Monetary Economics.*

[3] Daubechies I., Defrise M. and De Mol C. (2004), An Iterative Thresholding Al- gorithm for Linear Inverse Problems with a Sparsity Constraint, *Communications on Pure and Applied Mathematics*.

[4] Hodrick R.J. and Prescott E.C. (1997), Postwar U.S. Business Cycles: An Empirical Investigation, *Journal of Money, Credit and Banking*.

[5] Kalaba R. and Tesfatsion L. (1989), Time-varying Linear Regression via Flexible Least Squares, *Computers & Mathematics with Applications.*

[6] Kim S-J., Koh K., Boyd S. and Gorinevsky D. (2009), L1 Trend Filtering, *SIAM Review.*

[7] Tibshirani R. (1996), Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society B*.

[8] Tung-Lam Dao (2014), Momentum Strategies with L1 Filter, *Journal of Investment Strategies.*

1. In Model 1, we consider n=1000 observations. The parameters of the simulation are , , and . [↑](#footnote-ref-1)
2. Here, we assume , and we will discuss how to choose an optimal in the cross validation section. [↑](#footnote-ref-2)
3. In Model 2, we consider n=1000 observations. The parameters of the simulation are , , . [↑](#footnote-ref-3)
4. Here, we assume ,. [↑](#footnote-ref-4)
5. The L1-TC filter is based on L1-T and L1-C filters. Here we just drop L1-TC filter and use the other three filters. [↑](#footnote-ref-5)
6. Error is calculated as follows:

   where h denotes the length of the test set. [↑](#footnote-ref-6)
7. If there are some tied sequences , the formula becomes:

   with the number of tied sequences and the number of data points In the tied sequence.htied sequence.puter byr. We can ay in the program and y axis denotes the error. We can [↑](#footnote-ref-7)
8. The average filtered derivative is computed with a uniform moving average of 10 days. [↑](#footnote-ref-8)