ST451 PROJECT

A Practical Application of Hidden Markov Model to Kalman Filter-Based Pairs Trading

May 4, 2021

Abstract

The objective of this project is to implement a Bayesian updating process called the Kalman Filter in a common quantitative trading technique, which involves taking two assets that form a cointegrated relationship and utilising the mean-reverting nature between them, so called pairs trading. Furthermore, a common but difficult task for the quantitative trading participants is to detect the market regime change in the financial market. To address such challenge, we apply Hidden Markov Model for carrying out regime detection. We believe that by combining these two Bayesian approaches, a profitable while well-protected pairs trading strategy is achievable.

Keywords: State Space Model, Kalman Filter, Hidden Markov Model, Pairs Trading, Bayesian Machine Learning, Quantitative Trading

1 Introduction

Pairs trading, also referred as statistical arbitrage, is one of the most popular algorithmic trading strategies. The idea behind it is to simultaneously long and short two different assets having similar underlying factors (e.g., in the same industrial sector) that affect their price movements. The rationale behind it is that their long-term share prices are likely to be in equilibrium due to their exposure to the broad market factors that affecting their performance. A profitable arbitrage opportunity often arises when a short-term disruption happens to either of these two assets, believing that they would eventually return to their equilibrium once the disruption is resolved. To find such pairs of assets, we need a robust statistical framework for identifying the mean-reverting nature described above. Statistically speaking, this is where conintegrated time series comes in. Conintegrated time series refers to a pair of non-stationary time series forming a linear combination of each series to produce a stationary time series, which has fixed mean and variance. Due to the stationarity of such time series, its value would revert back to its mean even after short disruption. Statistical techniques to identify such conintegrated time series would be introduced in the following section.

A common approach to perform statistical arbitrage on a pairs of assets believed to have conintegrated relationship is to perform linear regression between them and use this determine the quantity to long and short at particular thresholds. This approach brings out the concern that the hedging ratio and the error term between these two assets are likely to be dyanmic - they will not stay fixed throughout the period of the strategy. It is essential to find a better technique that could adjust the hedging ratio over time. Although we can rebalance the hedging ratio by utilising a rolling linear regression with a lookback window, this would unnecessary introduces another free parameter into our strategy - the lookback window length, which should be optimised by using cross-validation. A even better method is to introduce a Bayesian approach called the Kalman Filter, a state space model that identifies the "true" hedging ratio as an unobserved hidden variable and attempts to estimate it with "noisy" observations - the price of the underlying assets.

After the 2020 coronavirus stock market crash, where unprecedentedly twice resulted in trading curbs, it is more important than ever to utilise a robust risk management system that could detect the frequent behaviour modification of the financial market. Such regime shifts are often due to the change of government policy, regulatory environment (e.g., the introduction of MiFID) and other macroeconomic effects (e.g., 2020 coronavirus and US-China trade war). As mentioned above, stationarity is key in pairs trading since it ensures the mean-reverting nature of conintegrated time series. The impact of various regimes would lead to time-varying correlation, excess kurtosis, heteroskedasticity and skewed returns, which enormously impacts the effectiveness of the pairs trading strategy. Therefore, it is essential to find a way to effectively detect these regimes. An iconic method in Bayesian machine learning for carrying out regime detection is to use Hidden Markov Model. It involves inference on "hidden" generative processes via "noisy" indirect observations correlated to these processes. In this case, the hidden process is the underlying regime state, while the asset returns are the indirect noisy observations that are influenced by these states.

The remaining part of this paper would be organised as followed: section 2 details the methodologies for the Kalman Filter, Hidden Markov Model, Cointegration test and the mechanism of our pairs trading algorithm. We endeavour to present the most essential mathematical equations in a easy-to-understand manner. Section 3 outlines the source of our data. It follows by the backtesting result and future improvement in section 4 and 5 respectively. Section 6 concludes the report.

2 Methodology

2.1 State space model and the Kalman Filter

The objective of state space model is to infer information about the hidden states (such as the hedging ratio between two cointegrated pairs) that evolve over time, given the observations, as new information arrives. A renowned Bayesian updating process for carrying out this task is called the Kalman Filter. The Kalman Filter has many practical applications in engineering control problems such as self-driving car guidance & navigation and spacecraft trajectory analysis. In quantitative finance, the most widely usage of the Kalman Filter is to dynamically update hedging ratios between assets in pairs trading.

Before delving into the Kalman Filter, let's first introduce all the elements in linear state-space model. In a a linear space model, we say that the state at t are a linear combination of the prior state as well as the system noise at time $t-1$. Here we assume that the system noise is drawn from multivariate normal distribution. The linear dependence of the state θ_t on the previous state θ_{t-1} is given by the matrix G_t, a matrix of known coefficients that defines systematic evolution of the state vector, which also dynamically updates over time. The multivariate time-dependent noise is given by w_t. We then can summarise the relationship:

$$\theta_t = G_t\theta_{t-1} + w_t$$

Since the states are hidden to us, we also need to define what we actually see, the observations y_t. The observations are linear combination of the current state and the measurement noise, which is also drawn from a multivariate normal distribution. Here we denote the linear dependence matrix of θ_t on y_t by F_t (a vector of known constants) and the measurement noise by v_t we have the observation equation:

$$y_t = F_t^T \theta t + v_t$$

Here we have defined the linear state space model, we then need a algorithm to solve it - this is where the Kalman Filter comes in.

To derive the Kalman Filter, we first need to introduce Bayes' Rule and conjugate priors,

The Bayes' Rule is given by:

$$P(\theta|D) = P(D|\theta)P(\theta)/P(D)$$

where θ refers to parameters and D refers to data or observations.

We denote D_t as all of the data known about the system and our current observations by y_t. We then can say that $D_t = (D_{t-1},\ y_t)$, which means that our current knowledge is a mixture of our previous knowledge and the most recent observation. By applying the Bayes' Rule, we have:

$$P(\theta_t|D_{t-1}, y_t) = \frac{P(y_t|\theta_t)P(\theta_t|D_{t-1})}{P(y_t)}$$

This equation tells us that the posterior θ_t, given our current observation y_t and previous information D_{t-1}, is equal to the likelihood of seeing an observation y_t, given the current state θ_t multiplied by the prior of the current state, given only the previous data D_{t-1}, normalised by the probability of seeing the observation at t. In a nutshell, this equation tells us that we can update our view on the state, θ_t, in a timely manner when the new information (the current observation, y_t) arises.

Let us specify the nitty-gritty that we will be using from Bayes' Rule above. Firstly we can specify the distribution form of the prior:

$$\theta_t|D_{t-1} \sim \mathcal{N}(a_t, R_t).$$

This equation tells us that the prior view of θ at time t, given our knowledge D_{t-1} is distributed as a multivariate normal distribution with mean a_t and variance-covariance R_t.

We now can denote the likelihood as:

$$y_t|\theta_t \sim \mathcal{N}(F_t^T\theta_t, V_t).$$

This says that the likelihood function of the current observation y_t is distributed as a multivariate normal distribution with mean $F_t^T\theta_t$ and variance-covariance V_t.

Eventually, we have the posterior of θ_t:

$$\theta_t|D_t \sim \mathcal{N}(m_t, C_t),$$

where we have:

$$m_t = a_t + A_t e_t = G_t m_{t-1} + A_t e_t,$$

$$C_t = R_t - A_t Q_t A_t^T,$$

$$A_t = R_t F_t Q_t^{-1},$$

$$e_t = y_t - f_t,$$

f_t is the predicted value of the observation at time t-1. The details of the above derivations could be found on (West and Harrison, 1989).

This infers that the posterior θ_t given our current knowledge D_t has a multivariate normal distribution with mean m_t and variance-covariance C_t.

Finally, we reach a point that we have an Bayesian process for updating our views on the observations and states we can use it to make predictions. We can make prediction of the next day's values by considering the mean value of the observation. It can be achieved by taking the expected value of the observation tomorrow given our knowledge of the data today:

$$\begin{aligned} E[y_{t+1}|D_t] &= E[F_{t+1}^T\theta_t + v_{t+1}|D_t] \\ &= F_{t+1}^T E[\theta_{t+1}|D_t] \\ &= F_{t+1}^T a_{t+1} \\ &= f_{t+1} \end{aligned}$$

Apart from simply calculating the mean, we must also know the variance of tomorrows observation given today's data, otherwise we cannot truly characterise the distribution on which to draw tomorrow's prediction. We have:

$$Var[y_{t+1}|D_t] = Var[F_{t+1}^T\theta_t + v_{t+1}|D_t]$$
$$= F_{t+1}^T Var[\theta_{t+1}|D_t]F_{t+1} + V_{t+1}$$
$$= F_{t+1}^T R_{t+1}F_{t+1} + V_{t+1}$$
$$= Q_{t+1}.$$

Here we have the expectation and variance of tomorrow's observation, given today's information, we then predict the observation on the next day by drawing it from the distribution:

$$y_{t+1}|D_t \sim \mathcal{N}(f_{t+1}, Q_{t+1})$$

2.2 Regime detection with Hidden Markov Model

Hidden Markov Model is such a model that is fully autonomous but only partially observable. What does it mean? In such a model, there are underlying latent states and probability transitions between them but they are not directly observable. These latent states instead influence the observations, though the observations do not possess the Markov Property.

Before delving into Hidden Markov Model, we need to formulate the Markov Chain Models into a probabilistic framework, which is the joint density function for the probability of seeing the observations, can be written as:

$$p(X_{1:T}) = p(X_1)p(X_2|X_1)p(X_3|X_2)...$$
$$= p(X_1)\Pi_{t=2}^T p(X_t|X_{t-1}).$$

This tells us that the probability of seeing a sequence of observations is given by the probability of the initial observation multiplied T-1 times by the conditional probability of seeing the subsequent observation, given the previous observation.

Hidden Markov Models are Markov Models where the states are hidden from view instead of being directly observable. For each state, we have a set of output observations, which are directly visible. For Hidden Markov Models, it is necessary to create a set of discrete states $s_t \in 1,...,K$ (In regime detection, K is often equal to 2 or 3) and to model the observations with an additional probability model, i.e., $p(x_t|s_t)$. This states that the conditional probability of seeing a particular observation (asset return) given that the state (market regime) is currently equal to s_t. It should be noted that the Hidden Markov Model will tend to stay in a particular state for some time before suddenly jumping to a new state and repeating the same behaviour.

This is desirable because the market regimes are not expected to change too frequently. When they do change, they are likely to persist for a period of time.

We now define the Hidden Markov Model in a joint probability distribution form as:

$$p(s_{1:T}|x_{1:T}) = p(s_{1:T})p(x_{1:T}|s_{1:T})$$

$$= [p(s_1)\Pi_{t=2}^{T} p(s_t|s_{t-1})][\Pi_{t=1}^{T} p(x_t|s_t)]$$

The first line tells us that the joint probability of seeing the full set of hidden states and observations is equal to the probability of seeing the hidden states multiplied by the probability of seeing the observations, conditional on the states. This is intuitively since the observations can not affect the states, but the hidden states do indirectly affect the observations. Figure 1 visually presents the dynamic process of the states s_t and how they dynamically lead to the observations x_t. Our objective in this project is to apply Hidden Markov Models to detect the market regime. It is a task to determine which market regime state the asset that we are interested in is currently in by utilising the asset returns we acquire up to date. It is a filtering problem, which is to estimate the current values of the state from past and current observations (Murphy, 2012).

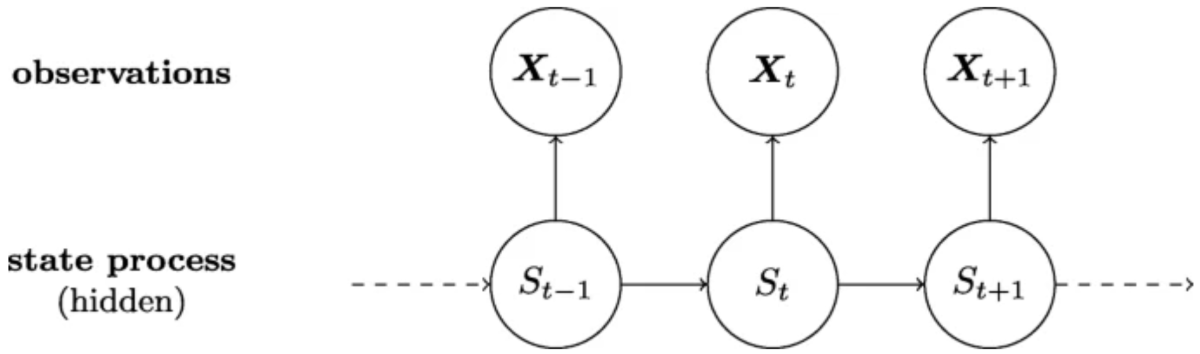

Figure 1: Hidden Markov Model

Two algorithms could be used to recursively compute filtered marginals $p(s_t = j|x_{1:t})$ in an HMM, they are the Forward Algorithm (Murphy, 2012) and the Viterbi Algorithm (Forney, 1973). Here our choice is the Viterbi Algorithm. It is an algorithm to compute the most probable sequence of states in a chain-structured graphical model, which is to compute

$$s^* = argmax_{s_{1:T}} p(s_{1:T}|x_{1:T}).$$

The details of how the algorithm works can be found on (Murphy, 2012).

It should be noted that for the application considered here, the observations of asset returns are continuous. A common choice is to make use of a conditional multivariate Gaussian distribution with mean μ_k and covariance σ_k, which can be formulated as:

$$p(x_t|s_t = k, \theta) = \mathcal{N}(x_t|\mu_k, \sigma_k).$$

This tells that if the state s_t is currently equal to k, then the probability of seeing observation x_t, given the parameters of the model θ, is distributed as a multivariate Gaussian.

2.3 Cointegration

We define x_t and y_t (t = 1,..,T) as two non-stationary time series, with a, b ∈ R. We say x_t and y_t are cointegrated if the combined series ax_t + by_t is stationary.

A model is said to be considered stationary if all of the roots of the equation exceed unity. In order to detect whether a time series is stationary or not, we need to construct a statistical hypothesis test for the presence of a unit root in a time series sample. We consider Augmented Dickey-Fuller (ADF) for testing the stationarity of the time series (Mushtaq, 2011).

The testing procedure for the ADF test is applied to the model:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + ... + \delta_{p-1} \Delta_{t-p+1} + \epsilon_t,$$

where α is a constant, β is the coefficient on a time trend and p is the lag order of the autoregressive process. When $\alpha = 0$ and $\beta = 0$, it corresponds to modelling a random walk; when only $\beta = 0$, it corresponds to modelling a random walk with a drift. The unit root test is then carried out under the null hypothesis $\gamma = 0$ against the alternative hypothesis of $\gamma < 0$. Once a value for the test statistic

$$DF_\gamma = \frac{\hat{\gamma}}{SE(\hat{\gamma})}$$

is computed, it can be compared to the relevant critical value for the Dickey-Fuller test. If the calculated test statistic is less than the critical value, then the null hypothesis of $\gamma = 0$ is rejected and no unit root is present.

2.4 Pairs Trading Strategy

Here we choose the price of asset A as the "observed" variables y_t, and the price of the asset B to be given by x_t. which forms the linear regression formulation as:

$$y_t = \theta_{\mathbf{t}} \mathbf{x_t} + v_t$$

$$= (\theta_t^0, \theta_t^1) \begin{pmatrix} x_t \\ 1 \end{pmatrix} + v_t$$

We define the dynamic hedge ratio by one component of the hidden state vector θ_t, which we will denote as θ_t^0. The goal here is to build a mean-reverting strategy from this pair of assets.

The synthetic "spread" between asset A and asset B is the time series that we are actually interested in longing or shorting. The Kalman Filter is used to dynamically track the hedging ratio between the two in order to keep the spread stationary.

Here it is essential to determine when the spread has moved too far from its expected value, assuming when the short-term disruption occurs. How can we determine the spread has moved too far? One approach is to consider a multiple of the standard deviation of the spread and use these as the bounds. Therefore, we can "long the spread" of the forecast error moves below the negative standard deviation of the spread. Respectively we can "short the spread" if the forecast error moves above the positive standard deviation of the spread. The exist rules are the opposite of the entry rules.

"Longing the spread" means that purchasing (longing) N units of asset A and selling (shorting) $\lfloor \theta_t^0$ N $\rfloor$ units of asset B, where $\lfloor$ x $\rfloor$ represents the largest integer less than x. The later is necessary since we must transact a whole number of units of the ETFs. "Shorting the spread" is the opposite of this. N controls the size we long or short.

e_t represents the forecast error or residual error of the prediction at time t, while Q_t represents the variance of this prediction at time t.

We specify the complete rules as follows:

1. e_t < - 1.5 x $\sqrt{Q_t}$ – Long the spread: Go long N shares of asset A and go short $\lfloor \theta_t^0$ N $\rfloor$ units of asset B
2. e_t ≥ - 0.5 x $\sqrt{Q_t}$ – Exit long: Close all long positions of asset A and asset B
3. e_t > 1.5 x $\sqrt{Q_t}$ – Short the spread: Go short N shares of asset A and go long $\lfloor \theta_t^0$ N $\rfloor$ units of asset B
4. e_t ≤ 0.5 x $\sqrt{Q_t}$ – Exit short: Close all short positions of asset A and asset B

The role of the Kalman filter is to help us calculate θ_t, as well as e_t and Q_t. θ_t represents the vector of the intercept and slope values in the linear regression between asset A and asset B at time t. It is estimated by Kalman filter. The forecast error $e_t = y_t - \hat{y}$ is defined by the difference between the value of asset A today and the Kalman filter's estimate of asset A today. Q_t is the variance of the predictions and hence $\sqrt{Q_t}$ is the standard deviation of the prediction.

We train the Hidden Markov Models on the training sets for both asset A and asset B. The trained models are then used to predict the hidden regime states for each asset respectively, where we have states indexed by 0 and 1.

The implementation of the strategy involves the following steps:

1. Receive the daily market price for both asset A and asset B.

2. Utilise the "online" Kalman Filter to dynamically estimate the price of asset A today boased on yesterday's observations of asset B.

3. Take the difference e_t between the Kalman estimate of asset A and the actual value - a measure of how much the spread of asset A and asset B moves away from its expected value.

4. Long the spread when it moves below the negative threshold we define above and short the spread correspondingly when it moves above the positive threshold.

5. Check the regime states for both asset A and asset B, if they are both at regime state 0, then we execute the above step; otherwise, we stop executing and wait for new opportunities.

6-i. Exit the long and short positions when the series reverts to its expected value.

6-ii. Close the open positions if regime states for either asset A or asset B move from the regime state 0 to the regime state 1.

QSTrader (Halls-Moore, 2016) will carry the infrastructure "work" such as the position tracking, portfolio handling and data ingestion, while we mainly focus on the code for signals generation and risk management.

3 Data

We use a pair of two ETFs, with ticker symbols UPRO and VOO, both representing a set of US equities tracking the S&P500 index. We are interested in finding out the performance in this pair of ETFs during the coronavirus market crash. We obtain the backward-adjusted closing prices from Yahoo Finance for UPRO and VOO from April 29th 2011 to April 30th 2021. We train the Hidden Markov Model for both ETF from April 29th 2011 to April 29th 2019. The backtest period is from April 30th 2019 to April 30th 2021.

4 Result

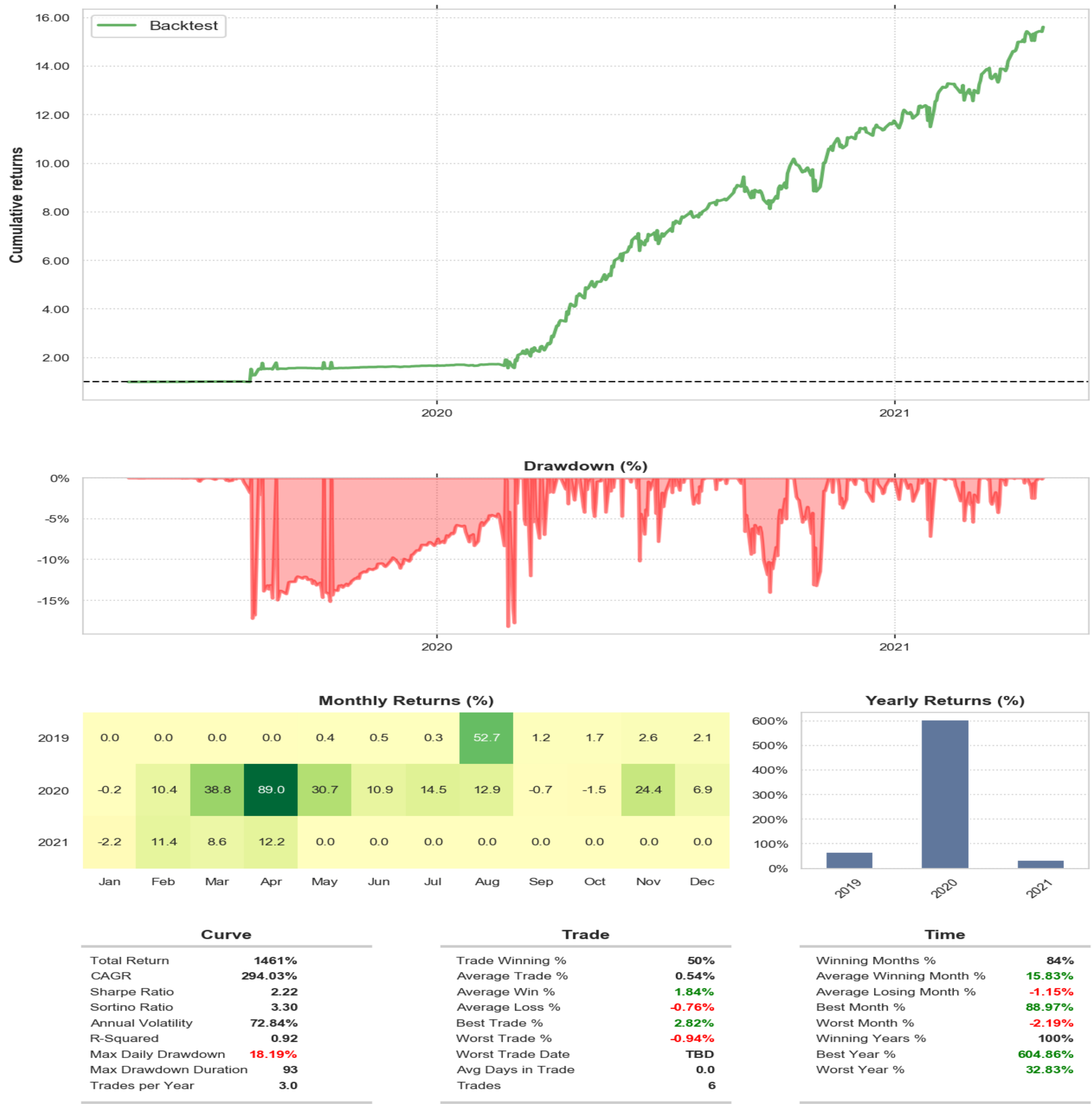

Figure 2: Backtest Result with HMM Risk Manager

Beyond our expectation, the Kalman Filter-Based Pairs Trading strategy with the Hidden Markov Model as a risk manager performed astonishing well during the 2020 coronavirus US stock crash in 2020. According to the chart shown above, we can see that the strategy achieved a Sharpe Ratio at 2.22 and total return at 1461% return with a maximum daily drawdown at 18.19%. Surprisingly, it resulted in 604.86% return in 2020. These results indicates that not only did the market-neutral strategy could avoid from the market impact due to COVID-19 pandemic, but it further reinforced its performance due to the dynamic Kalman Filter and the Hidden Markov Model as a risk manager.

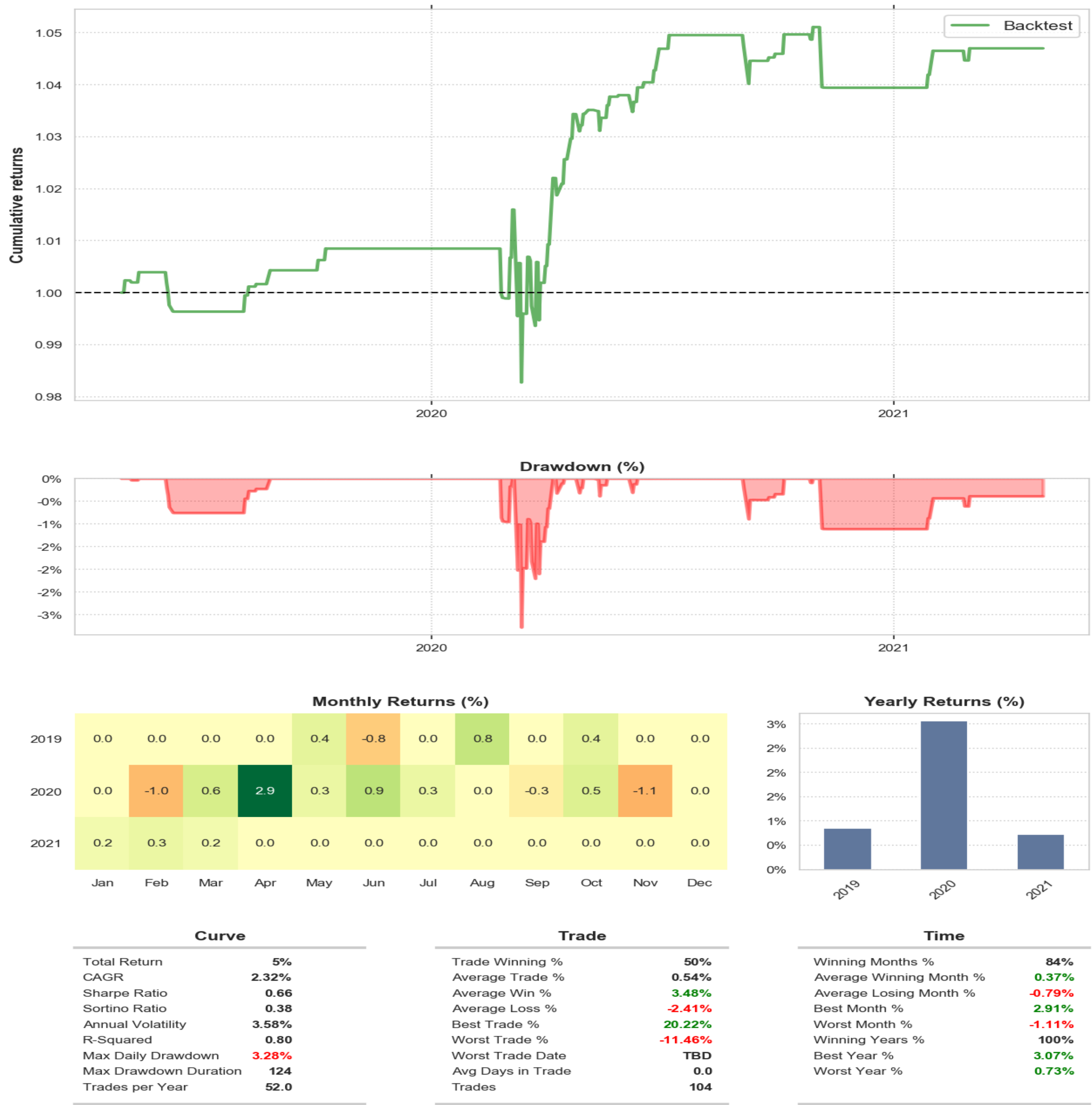

Figure 3: Backtest Result without HMM Risk Manager

How would the Kalman Filter-Based Pairs Trading strategy perform without the Hidden Markov Model as a risk manager? From the chart shown above, we can see that the strategy achieved a Sharpe Ratio at 0.66 and total return at only 5%, though its maximum daily drawdown at only 3.28%. Comparing these two strategy, we can see that the one with HMM as a risk manager is a more cautious strategy, with only 6 trades made throughout the period;

while the one without HMM as a risk manager is more aggressive strategy in general, with more than 100 trades made throughout the whole period. By comparing these two strategies, it could further verify that the Hidden Markov Model indeed "protected" the strategy from the volatile market in 2020. Moreover, the Kalman-Based Pairs Trading strategies achieve positive returns with or without the Hidden Markov Model as a risk manager.

5 Future Improvement

To make this strategy even better, there is still a lot of work could be done. Potential areas for further improvements are:

Portfolio Construction – We could add more pairs of conintegrated ETFs to diversify the portfolio and increase the overall complexity of the strategy.

Parameter Optimisation – The parameters of the Kalman Filter such as the system noise w_t and the measurement noise v_t could be optimised via cross-validation, though this might result in overfitting the historical data.

6 Conclusion

Our proposed Kalman Filter-based Pairs Trading strategy with the Hidden Markov Model as a risk manager achieved amazingly well results during the US stock market crash due to the COVID-19 pandemic. This dynamic Bayesian strategy tied with Kalman Filter and the Hidden Markov model not only achieved fabulous overall returns, but it also protected the strategy from the extreme volatile market condition during the stock market crash.

References

[1] G David Forney. “The viterbi algorithm”. In: *Proceedings of the IEEE* 61.3 (1973), pp. 268–278.

[2] Michael L Halls-Moore. “Advanced Algorithmic Trading”. In: (2016).

[3] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[4] Rizwan Mushtaq. “Augmented dickey fuller test”. In: (2011).

[5] Mike West and Jeff Harrison. “Subjective intervention in formal models”. In: *Journal of Forecasting* 8.1 (1989), pp. 33–53.