Intraday Long Gamma Systematic Trading via Hidden Markov Models

Jonathan Benjamin Seror
June 30th 2020
serorjb@gmail.com

Abstract This study presents a modest Hidden Markov Models flavored extension of the tail risk mitigation strategies discussed by Ernest Chan (QTS Capital Management, L.L.C.) on his blog, and Mark Spitznagel (Universa Investments, L.P.) in an interim letter to investors dated April 7th, 2020, following the covid-19 market turmoil.

The objective is to partly replicate the performance of these long gamma strategies, and attempt to bridge a reported gap between in and out-of-tail net return on capital, by adding a Hidden Markov Model based regime detection overlay to the proposed hedging approach. The results show that a well calibrated non-biased HMM regime filter, effectively shuts down the trend following trading signal, when sub-optimal conditions would make it unprofitable.

The implementation focuses on a representative U.S. energy large cap, and relies on the R dependent mixture models depmixS4 library.

Contents

1 Regime Detection via Hidden Markov Models

1.1 Market Regimes Definition

Regimes are distinct phases of a market cycle, characterized by different market activity patterns stemming from intangible macroeconomic effects, investors confidence shifts, regulatory changes, etc. Since investors adapt and respond dynamically to such regime changes, the profitability of vanilla buy and hold strategies for example suffers in certain volatile contexts.

Identifying which regime the investor is currently facing goes a long way into assisting decision making and tail hedging a a portfolio accordingly.

1.2 Hidden Markov Models

Hidden Markov Models (HMM) are a subset of Markov memoryless transition models, assuming states are hidden and attempting to infer the probabilities of being in each state at a given point in time based on the observations set. Market regimes will be modelled as HMM using log returns observations for the purpose of this study, in order to flag a tail setting.

Given a set of discrete states $z_t \in \{1, ..., K\}$ and a conditional probability $p(X_t|z_t)$ of seeing a particular asset return observation in a market regime state, the HMM joint density function can be expressed as follows (cf. Murphy 2012).

$$p(z_t|X_t) = p(z_t)p(X_t|z_t)$$

with the probability distributions defined as transition functions for the states

$$p(z_t) = p(z_1)\prod_{t=2}^{T} p(z_t|z_{t-1})$$

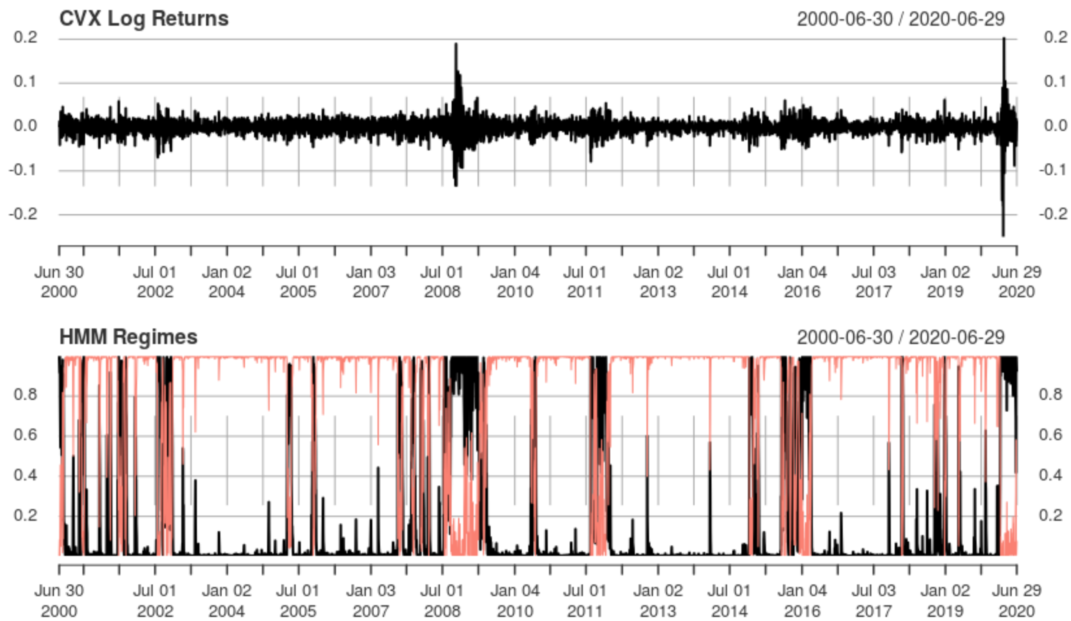
and for the conditional observations

$$p(X_t|z_t) = \prod_{t=1}^{T} p(X_t|z_t)$$

Further, this study assumes a multivariate Gaussian distribution of mean μ_k and covariance σ_k to model conditional observations.

CVX log returns and corresponding regimes detection Setting $K = 2$ over a 20 year period spanning from 2000 to 2020 for Chevron Corporation (CVX), gives the below sample distribution, correctly flagging among others, the 2007-2009 subprimes crisis, and the recent pandemic recession.

Whilst the R code used to produce this graph is given in annex, please note this is not the implementation used in the next section for trading simulation and backtest; instead a 15-years rolling window is used to fit the HMM, in order to avoid any look-ahead bias, cf. next section.



2 Tail Behaviour and Strategy Proposal

Trend-following is deemed more reliable in a tail context, and given correct entry and exit thresholds, very profitable. The proposed intraday strategy tests this assumption on a 1 year period starting July 2019, using a vanilla $\pm$ 1% threshold as entry point (short trading allowed) and 0% or EOD as exit, *conditional* to the market regime being flagged by the HMM deployment as tail.

2.1 Backtesting Biases

Backtesting is conducted using an event driven framework to avoid feeding future information, addressing biases is important to avoid any misinterpretation. Whilst a comprehensive drill down out of scope, look-ahead, survivorship, cognitive and optimisation biases are briefly discussed.

Look-ahead Fitting the HMM over a 20 years time span as shown in section 1, is actually inaccurate, as it uses future data to fit a model used in investing within this time frame. Hence a rolling prior 15 years window of log returns has been arbitrarily selected to fit the model, subject to optimisation.

Survivorship As the large cap equities of interest are evaluated using an HMM spanning over a 10 years rolling window, there is a bias in knowing these did not go bankrupt/acquired/merged/etc. and still exist, which is a major assumption. Taking an index or simple industry bucket as a base for fitting the HMM would partially address this issue, with a similar outcome.

Cognitive The study targets specifically an oil producer, a sector which *retrospectively* is obviously known to be deeply impacted by the pandemic crisis; that being said, all industries have been widely shaken and whilst the study only outlines a specific potential of the tail hedge strategy, an actual implementation taking a generalized approach should produce similar results.

Optimisation Excessive adjustment to fit models and parameters to a given training set, detrimental to actual test performance. Seeking *optimal* alpha with a flawless strategy implementation is not the purpose of this study, hence both sensitivity analysis and optimisation are actually left at reader's discretion. Separately the number of parameters has been kept to a minimum; worth noting the 15 year rolling window choice, conveniently captures the notable 2007-2009 financial crisis.

2.2 Implementation Assumptions

As mentioned earlier, the HMM is fit using a multivariate Gaussian distribution response family, and lookback period 15 years, which presumably captures 1 or more market cycles; the longer this time window, the more accurate the output inference, the more computationally intensive the process.

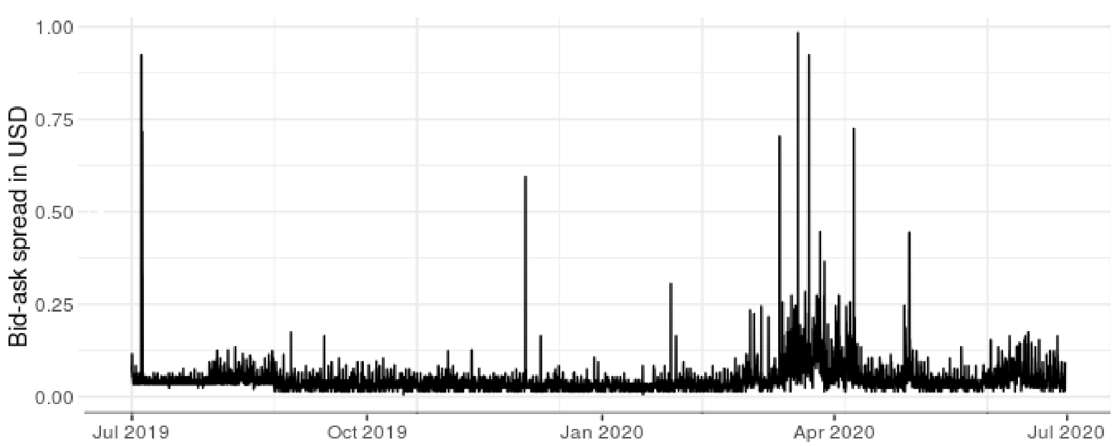
Probability Threshold A useful extension of this study could use rolling moving averages or a step function of regime probabilities, to avoid 1-day regime change spikes. A basic 50% probability *threshold* is applied to the HMM output to consider a regime change signal. A potential implementation improvement could be to adopt a dynamic sizing as a function of the HMM output tail probability.

Systematic Trading Simulation A low-frequency 15-min trading window is used, slippage, latency, liquidity and other book effects are not accounted for; outputs are presented as a function of a transaction costs range. Data quality and price quote accuracy evaluation are not in the scope of this study.

2.2.1 Bid-Ask Spread Widening

By nature, long gamma strategies are exposed to volatile trading conditions, and this study accounts for the bid-ask spread widening, by using the actual bid-ask spread when entering the market as an acceptable proxy considering the strategy is intraday.

Below for reference the Chevron Corporation CVX historical share price, and corresponding sample bid-ask spread chart observed at a major institutional broker, expressed in USD, divided in 15 minutes time periods, over the 1 year trading window considered for this study starting July 2019.

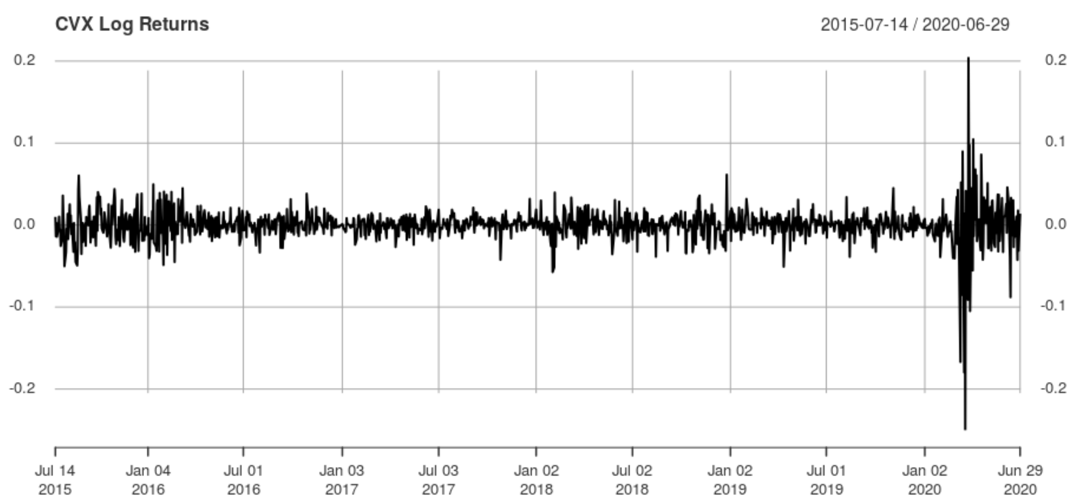


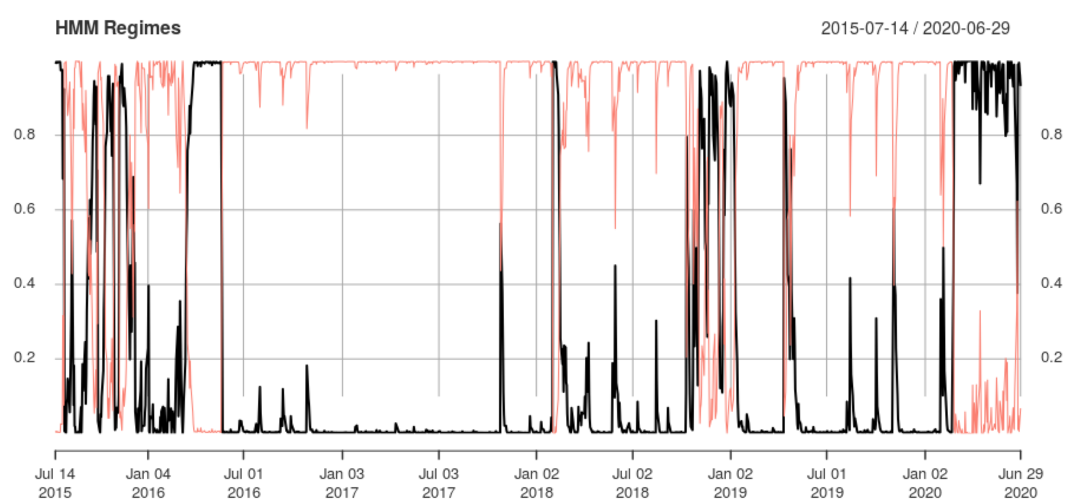
3 US energy Large Cap Amid Covid-19 Turmoil Case Study

3.1 HMM Output Probabilities

Setting $K = 2$ regimes over a prior 15 year rolling window for Chevron Corporation (CVX), gives the below sample distribution for a 5 year period starting July 2015; whilst the fit looks consistently good, only the output from the 1 year period starting July 2019 is used to test the Long Gamma strategy, as it is interestingly divided in roughly ⅓ non-tail (the first 165 days) ⅔ tail (the last 88 days), hence very relevant.

The HMM chart below provides the detected probabilities of each regime, in pink the non-volatile, non-tail market and in black the tail part during which the strategy is allowed to perform. For reference, the covid-19 pandemic actually started shaking the global financial markets mid-Feburary 2020, which is accurately captured by the model.





3.2 Systematic Trading Simulation

As mentionned earlier, a low-frequency 15-min trading window is used, slippage, latency, liquidity, market impact and other book effects are not accounted for; outputs are presented as a function of a transaction costs spread ranging from 0bp to 50 bps. The bid-ask spread has been accounted for as described in the dedicated subsection. A basic 50% probability *threshold* is applied to the HMM output to consider a regime change signal. Also a static 'all-in' sizing is taken without leverage for simplicity's sake, leaving out potentially more optimal approaches as function of the HMM output tail probability. An approximated risk free rate of 1% is used to compute performance ratios, which is approximately the averaged 1 year U.S. Treasury yield over the period.

As mentionned in the backtesting biases section, seeking *optimal* alpha with a flawless strategy implementation is not the purpose of this study. The simple round numbers for parameters listed above have been chosen for illustrative purposes, both sensitivity analysis and optimisation are actually left at reader's discretion.

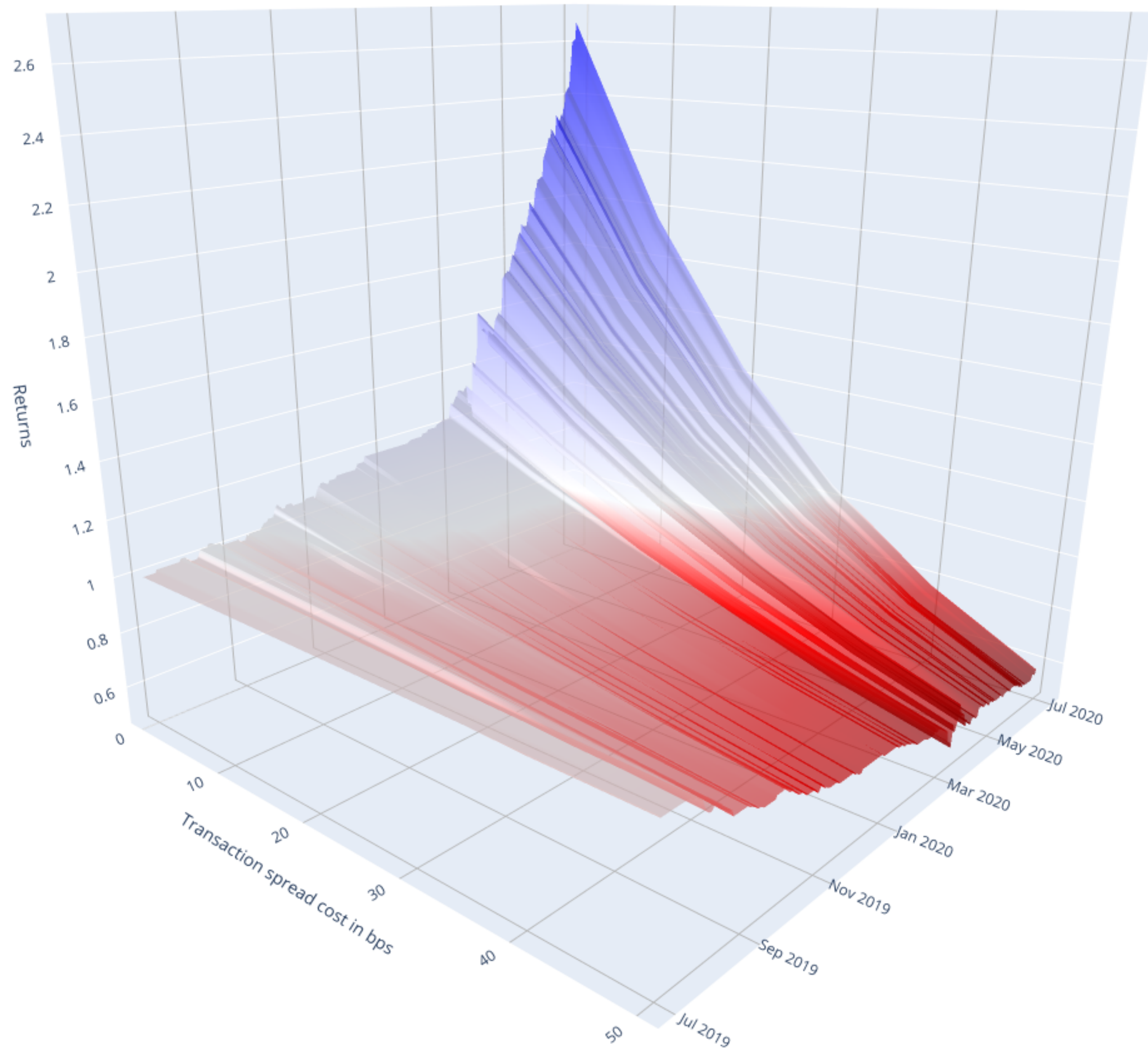
With the use of a Hidden Markov Model filter, 87 trades were triggered during the 253 trading days period, of which 52 were profitable. Comparatively the same trend following strategy without any filtering, generated 154 signals, with 96 profitable ones. However, a key difference is that these positive signals generated on average 2.4% gross profit for the filtered ones, against 1.6% for the unfiltered, thereby piling up losses once transaction costs are accounted for.

As shown further below, the HMM regime filter *effectively shuts down* the trend following trading signal when suboptimal conditions would make it unprofitable.

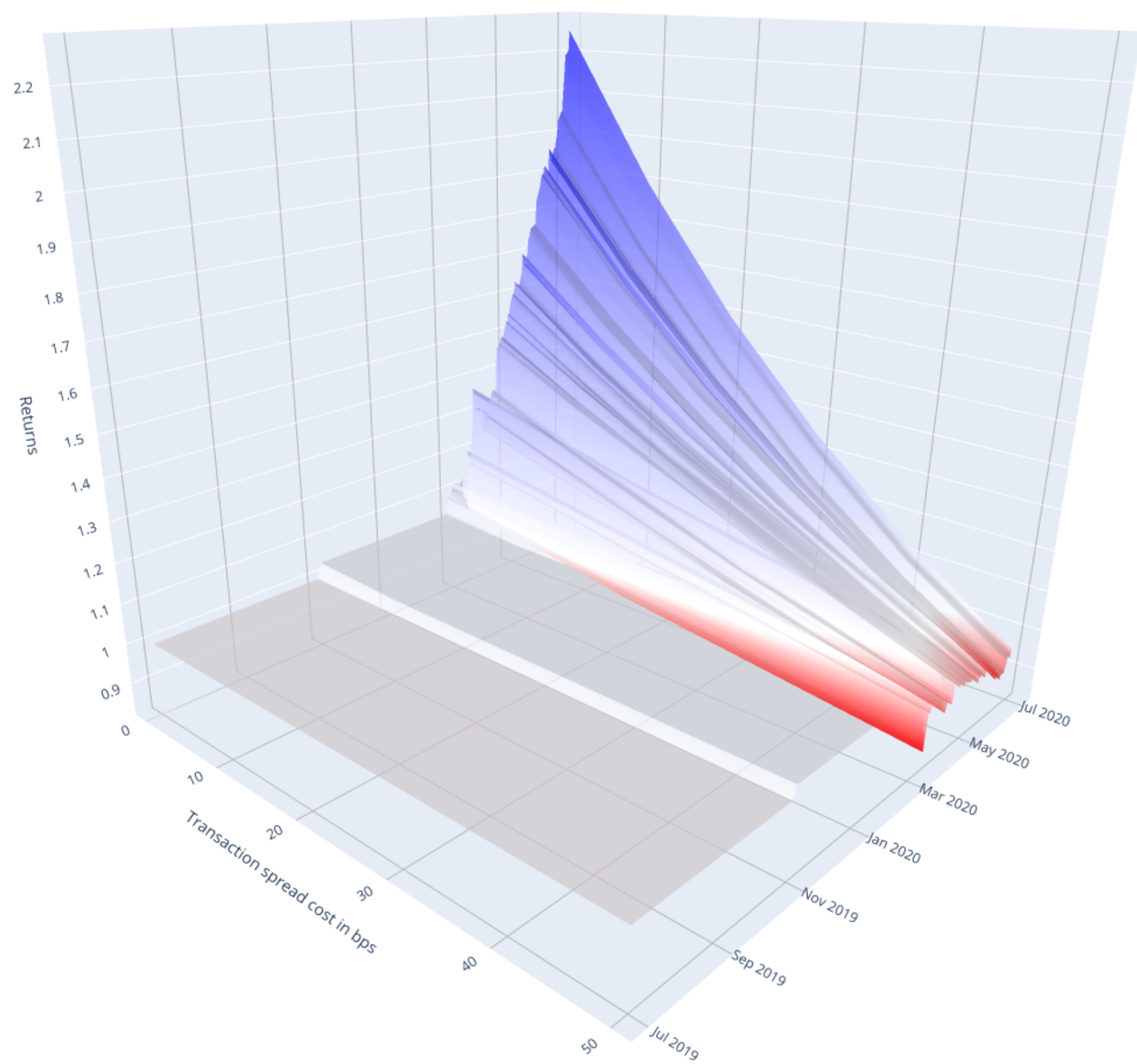
3.2.1 Returns

Below two comparative, comprehensive charts displaying *returns over time as a function of transaction costs* to account for costs, commissions, slippage and market impact effects, etc. Taking for example a *10 basis point* transaction pair cost, the result gives a 189% total return over the 253 trading days period ranging from July 1, 2019 to June 30, 2020. Note considering the extremely low rate environment over that period, and the relatively short investment period of the study, discounting has been left out, only accounted for in the next section on performance ratios.

Without HMM tail regime filter



With HMM tail regime filter



3.2.2 Performance Ratios

$$\text{Sharpe } \frac{\mathbb{E}(r_p - r_b)}{\sqrt{Var(r_p - r_b)}} \;\Big|\; \text{Sortino } \frac{\mathbb{E}(r_p - r_b)}{\sqrt{Var(r_p - r_b)_{downward}}} \;\Big|\; \text{CALMAR } \frac{\mathbb{E}(r_p - r_b)}{maximum\, drawdown}$$

Formulas for the performance ratios are given above, with r_p the portfolio returns and r_b the benchmark return. An *indicative* estimate can be obtained under the assumptions outlined above, considering for instance a *10 basis point* flat reference commission spread on each transaction, a 1 year investment period and using a 1% risk free reference rate r_f as benchmark, the resulting (annualized) Sharpe ratio is 52, the Sortino ratio is 81 and the CALMAR is 7.

Conveniently the investment period is exactly one year, which allows performance analysis on an annualized basis. Whilst the Sharpe ratio is industry standard, it is relevant in this case to consider the Sortino as most of the volatility is upward, and CALMAR as the maximum drawdown is 12.56% and maximum drawdown duration is 16 days.

Again whilst the study attemps to capture the bid-ask widening spread and transaction costs, it is worth emphasizing these attractive numbers can differ *substantially* in a live implementation, due to execution variability, slippage, latency, market impact and other effects.

3.2.3 Summary

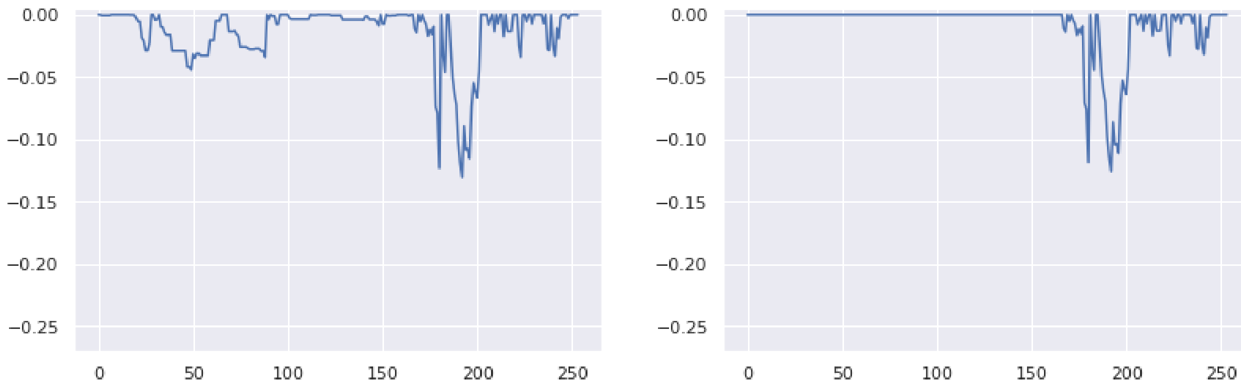
Without HMM tail regime filter

Cost spread (bps)	Sharpe	Sortino	Max. drawdown	Drawdown days	CALMAR
0	95.53	178.83	-13.50%	26	12.27
10	55.57	108.29	-13.06%	32	7.30
20	25.56	50.25	-13.46%	164	3.23
30	3.14	6.14	-26.41%	231	0.20
40	-13.47	-25.94	-37.70%	252	-0.60
50	-25.65	-49.08	-47.27%	252	-0.92

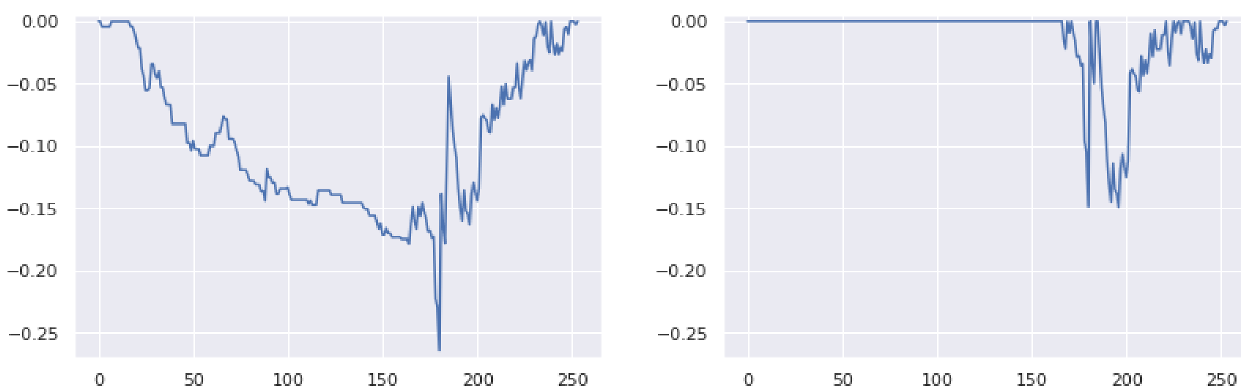
With HMM tail regime filter

Cost spread (bps)	Sharpe	Sortino	Max. drawdown	Drawdown days	CALMAR
0	72.74	113.75	-11.39%	16	10.87
10	52.67	81.28	-12.57%	16	7.02
20	35.24	53.51	-13.60%	17	4.29
30	20.18	30.29	-14.90%	35	2.22
40	7.26	10.53	-16.91%	68	0.70
50	-3.74	-5.39	-20.58%	68	-0.3

3.2.4 Drawdown Analysis



High water mark curve plotting drawdown over 253 trading days, without (left) and with (right) Hidden Markov Model tail regime identification and filtering, using a 10bps transaction cost spread

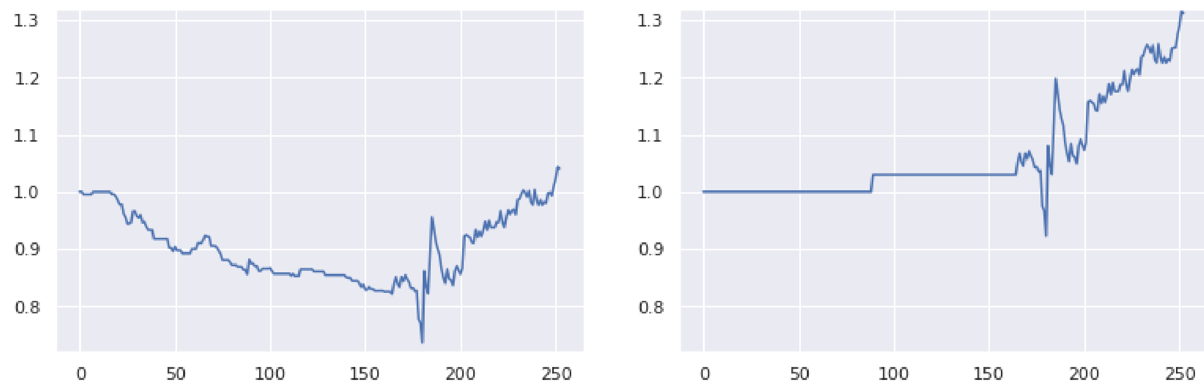


Increasing the spread to 30bps, the HMM filtered method continues to display drawdown resilience

Whilst the unflavored base strategy can somewhat sustain the comparison against the HMM enhanced version at low transaction costs, and even deliver additional returns, it should be acknowledged and emphasised that a large proportion of the selected time period for this study does show volatility, allowing the natural strategy to thrive.

Over a longer sustained investment period, the returns of the base strategy would naturally erode over time, whilst the HMM filtered strategy would stay away from unprofitable, non-tail markets.

4 Conclusion



Long gamma equity curve plotting returns (based on a 30bps transaction cost), over 253 trading days, without (left) and with (right) Hidden Markov Model tail regime identification and filtering

The Hidden Markov Model regime detection filter effectively shuts down the trend following trading signal when suboptimal conditions would make it unprofitable; the difference is more pronounced using a conservative 30 basis point flat reference spread for each transaction, where the HMM filtered strategy returns 34%, against a modest 6% (and a 231 days drawdown) without, as shown on the two charts above.

Appendices

A Python

A.1 Python Graphing Insert

Code sample used to produce the surface plots, leveraging the plotly library.

```
import plotly.graph_objects as go
import pandas as pd

z_data = pd.read_csv("/home/js/cvx_trading_HMM.csv")

fig = go.Figure(data=[go.Surface(z=z_data.values, colorscale=[[0, 'red'], [0.1,
 ↪'white'], [1.0, 'blue']], opacity = 0.65, legendgroup = "Returns")])

fig.update_layout(scene = dict(
                    xaxis_title='Transaction-pair cost spread in bps',
                    yaxis_title='Number of trading days',
                    zaxis_title='Returns'))

fig.update_layout(margin=dict(l=0, r=0, b=0, t=0),
                  scene = dict(
                    xaxis = dict(
                        nticks=9,
                        range= [0,7],
                        gridcolor="rgb(180, 180, 180)"
                        ),
                    yaxis = dict(
                        gridcolor="rgb(180, 180, 180)",
                        ),
                    zaxis = dict(
                        nticks=18
                        )
                    )
                  )

fig.show()
```

A.2 Python Backtest Insert

Simplistic event-driven backtest implementing the conditions laid out for the scope of this study, leveraging a 15 minute pricing feed, sample input .csv extract below.

GmtTime	Date	DayOpen	Open	Close	HMMTailProbability	DailyReturn
13:30	01.07.2019	124.988	124.988	125.597	0.26%	0.49%
13:45	01.07.2019	124.988	125.647	125.677	0.26%	0.55%
14:00	01.07.2019	124.988	125.678	125.528	0.26%	0.43%
14:15	01.07.2019	124.988	125.528	125.718	0.26%	0.58%
14:30	01.07.2019	124.988	125.718	125.748	0.26%	0.61%
14:45	01.07.2019	124.988	125.747	125.508	0.26%	0.42%
15:00	01.07.2019	124.988	125.527	125.548	0.26%	0.45%
15:15	01.07.2019	124.988	125.548	125.078	0.26%	0.07%
15:30	01.07.2019	124.988	125.078	125.218	0.26%	0.18%
⋮	⋮	⋮	⋮	⋮	⋮	⋮
19:45	30.06.2020	86.598	89.218	89.208	100%	3.01%

```python
import matplotlib.pyplot as plt
import seaborn as sns
import pandas

df=pandas.read_csv('/home/js/cvx_trading_15min.csv')
df['DailyReturn'] = pandas.to_numeric(df['DailyReturn'])
df['BidAskSpread'] = pandas.to_numeric(df['BidAskSpread'])

Date = ''
DatePrevious =''
Invested = False
AlreadyInvestedToday = False
TransactionCostsSpread = 0.001 #e.g. 10bps
BidAskSpreadPaid = 0.0
Direction=''
TotalReturn = 1.0
DailyReturnsSeries = []
TradeCount=0

for index, row in df.iterrows():
    Date = row['Date']
    if Date != DatePrevious:
        if AlreadyInvestedToday == True:
            TotalReturn = TotalReturn * pow(1 - TransactionCostsSpread ,2) *␣
↪(1-BidAskSpreadPaid)
```

```
        DailyReturnsSeries.append(TotalReturn)
        Invested= False
        AlreadyInvestedToday = False

    if Invested == False and AlreadyInvestedToday == False and␣
↪row['HMMTailProbability']>0.5:
    #if Invested == False and AlreadyInvestedToday == False: #without HMM filter
        if row['DailyReturn'] >= 0.01:
            Direction = 'Long'
            Invested = True
            AlreadyInvestedToday = True
            TradeCount += 1
        if row['DailyReturn']<=-0.01:
            Direction = 'Short'
            Invested = True
            AlreadyInvestedToday = True
            TradeCount += 1
        BidAskSpreadPaid = row['BidAskSpread']/row['Open'] #approximation

    if Invested == True:
        if Direction == 'Long':
            TotalReturn = TotalReturn * (1+((row['Close']-row['Open'])/
↪row['Open']))
        else:
            TotalReturn = TotalReturn * (1+((row['Open']-row['Close'])/
↪row['Open']))
        if (row['DailyReturn']<=0 and Direction=='Long') or␣
↪(row['DailyReturn']>=0 and Direction=='Short'):
            Invested=False

    DatePrevious = Date

DailyReturnsSeries.append(TotalReturn)

print ('Final trade count: %s' % TradeCount)
print ('Final returns over 1 year: %s' % TotalReturn)
sns.set(style="darkgrid")
plt.plot(DailyReturnsSeries)
```

B R HMM Insert

Sample HMM vanilla implementation for the above example. Note this is not the implementation used in the next section for trading simulation and backtest; instead a 15-years rolling window is used to fit the HMM, in order to avoid any look-ahead bias, cf. next section.

```
library('depmixS4')
library('quantmod')

getSymbols("CVX",from="2000-06-30", to= "2020-06-30")
CVX$CVX.Returns = as.numeric(diff(log(CVX$CVX.Close)))
CVX$CVX.Returns[is.na(CVX$CVX.Returns)] = 0

HiddenMarkovModel = depmix(CVX$CVX.Returns~1.,family=gaussian(), nstates=2,
 ↪data=CVX$CVX.Returns)
HiddenMarkovModelFit = fit(HiddenMarkovModel,verbose = FALSE)
PosteriorProbabilities = posterior(HiddenMarkovModelFit)

CVX$CVX.P1=PosteriorProbabilities[,2]
CVX$CVX.P2=PosteriorProbabilities[,3]

layout(3:1)
plot(CVX$CVX.Returns ,type='l', main = 'CVX Log Returns')
plot( CVX$CVX.P1, type="l", col="black", main = "HMM Regimes")
lines(CVX$CVX.P2, type="l", col="salmon")
```

References

[1] Ernest Chan, *Why does our Tail Reaper program work in times of market turmoil?*, Mar.2020.

[2] Mark Spitznagel, *Interim Decennial Letter*, Apr.2020.

[3] Jack Fan, Isaac Kleshchelski, *Trend Following and Volatility Regimes*, Oct.2017.

[3] Michael Halls-Moore, *Advanced Algorithmic Trading*, 2016.