

DTSC 5301 Group Project Data Science for Finance

Tactical Asset Allocation with a Bubble Signal

Authors

- Josh Nielsen, josh.nielsen@colorado.edu
- Charan Kanwal Preet Singh chch4792@colorado.edu
- Nikhil Madhu Belavinakodige, nibe7119@colorado.edu
- Om Sai Madala, oomma5510@colorado.edu
- Anup Bhutada, anbh1796@colorado.edu

Introduction

Tactical asset allocation (TAA) is an active portfolio management practice where a portfolio manager allocates percentages of their assets toward certain categories of financial instruments based on some signal. In this notebook, we compare the performance of two portfolios: an equally weighted portfolio and a portfolio whose weights are determined by a bubble signal. When there is a strong bubble signal we allocate to a risk-off portfolio comprised fully of debt instruments. When there is a strong negative bubble signal we allocate to a risk-on portfolio comprised fully of equity instruments. When there is neither a strong positive nor a negative bubble signal, we allocate to a 60/40 portfolio where 60% of our assets are in equity and 40% are in debt instruments. Below are the exchange traded funds we use for our analysis. We compare the cumulative returns for our bubble portfolio to a 60/40 portfolio from `START_DATE` to `END_DATE`.

ETF Ticker	Allocation	Underlying Asset	Description
VTI	20%	Equity	Seeks to track the performance of the CRSP US Total Market Index. Large- mid- and small-cap equity diversified across growth and value styles.
VTV	20%	Equity	Seeks to track the performance of the CRSP US Large Cap Value Index which measures the investment return of large-capitalization value stocks.
VBR	20%	Equity	Seeks to track the performance of the CRSP US Small Cap Value Index which measures the investment return of small-capitalization value stocks.
TLT	20%	Debt	The iShares 20+ Year Treasury Bond ETF seeks to track the investment results of an index composed of U.S. Treasury bonds with remaining maturities greater than twenty years.
AGG	20%	Debt	The iShares Core U.S. Aggregate Bond ETF seeks to track the investment results of an index composed of the total U.S. investment-grade bond market.

General Imports and Constants

```
library('tidyverse')

-- Attaching packages ----- tidyverse 1.3.2 --
v ggplot2 3.3.6      v purrr   0.3.4
v tibble  3.1.8      v dplyr   1.0.10
v tidyr   1.2.0      v stringr 1.4.1
v readr   2.1.2      v forcats 0.5.2
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()

library('ggplot2')
library('dplyr')

# constants
POS_BUBBLE_THRESHOLD = 0.15
NEG_BUBBLE_THRESHOLD = 0.15
UNIVERSE_REMOTE_URL = 'https://raw.githubusercontent.com/Joshwani/DataScience5301GroupProject/main/universe.csv'
CONF_REMOTE_URL = 'https://raw.githubusercontent.com/Joshwani/DataScience5301GroupProject/main/conf.csv'
```

Fetching Universe Data (Yahoo! Finance)

Yahoo! Finance computes an Adjusted Closing price that factors in all splits and dividends, see [here](#). For the sake of simplicity and reproduce-ability, in our analysis, we are simply loading the data sets that were previously fetched from Yahoo! Finance.

```
# read universe.csv into df var, set Date col as row name
df <- read.csv(UNIVERSE_REMOTE_URL, row.names = 'Date')
```

Creating the Benchmark

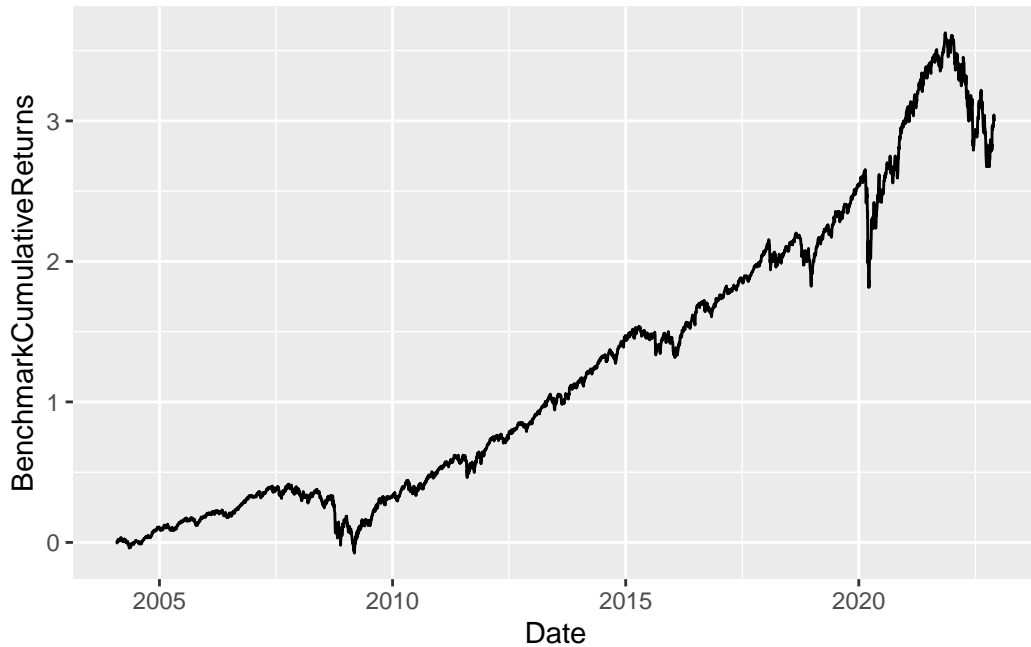
```
# compute daily percent change
df_pct_chg <- na.omit((df-lag(df, 1))/lag(df, 1))

# create a 60/40 portfolio
df_pct_chg$Benchmark <- apply(df_pct_chg, 1, mean)
```

```
df_pct_chg$BenchmarkCumulativeReturns <- cumprod(1+df_pct_chg$Benchmark)-1

# add date as a col of type Date
df_pct_chg$Date = as.Date(rownames(df_pct_chg))

# visualize it
ggplot(df_pct_chg, aes(x=Date)) + geom_line(aes(y=BenchmarkCumulativeReturns))
```



Bubble Model

Is it possible to avoid the large draw-downs and capitalize on the run-ups? Here we use the [Log-Periodic Power Law Singularity \(LPPLS\) Model](#). It describes a bubble 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. A bubble has a distinct signature that resembles a power law with a finite-time singularity decorated by oscillations with a frequency increasing with time.

If we can identify bubbles, then we can use them as an indicator for our tactical asset allocation strategy.

The first step is to compute the bubble indicators for the benchmark. Here we are importing the signal pre-computed from Boulder Investment Technologies GitHub Repository.

Load Bubble Data

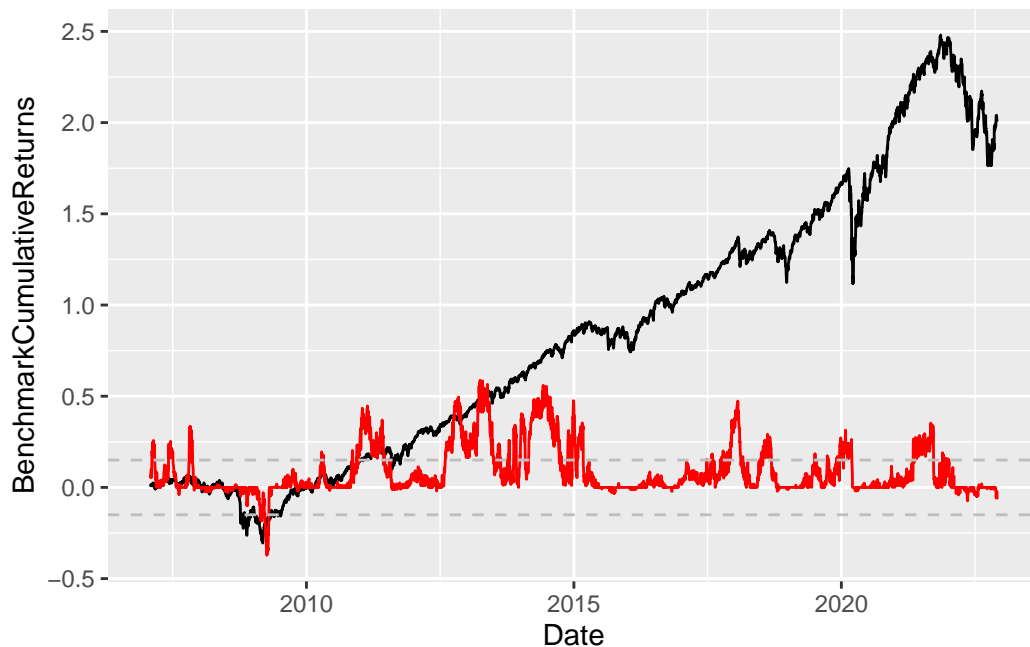
```
bubble_df<- read.csv(CONF_REMOTE_URL, row.names='time')
bubble_df$Date <- as.Date(rownames(bubble_df))

# join data frames. df + bubble df on date
m_df <- merge(df_pct_chg, bubble_df, by='Date')
m_df$BenchmarkCumulativeReturns <- cumprod(1+m_df$Benchmark)-1
```

Visualize Bubble Data

Here we plot the positive and negative bubble indicators along with the thresholds at which they will trigger an allocation signal for our strategy. Note, the thresholds have been chosen rather arbitrarily.

```
ggplot(m_df, aes(x=Date)) +
  geom_line(aes(y=BenchmarkCumulativeReturns)) +
  geom_line(aes(y=conf), color='red') +
  geom_hline(yintercept=c(POS_BUBBLE_THRESHOLD, -NEG_BUBBLE_THRESHOLD), color='grey', line
```

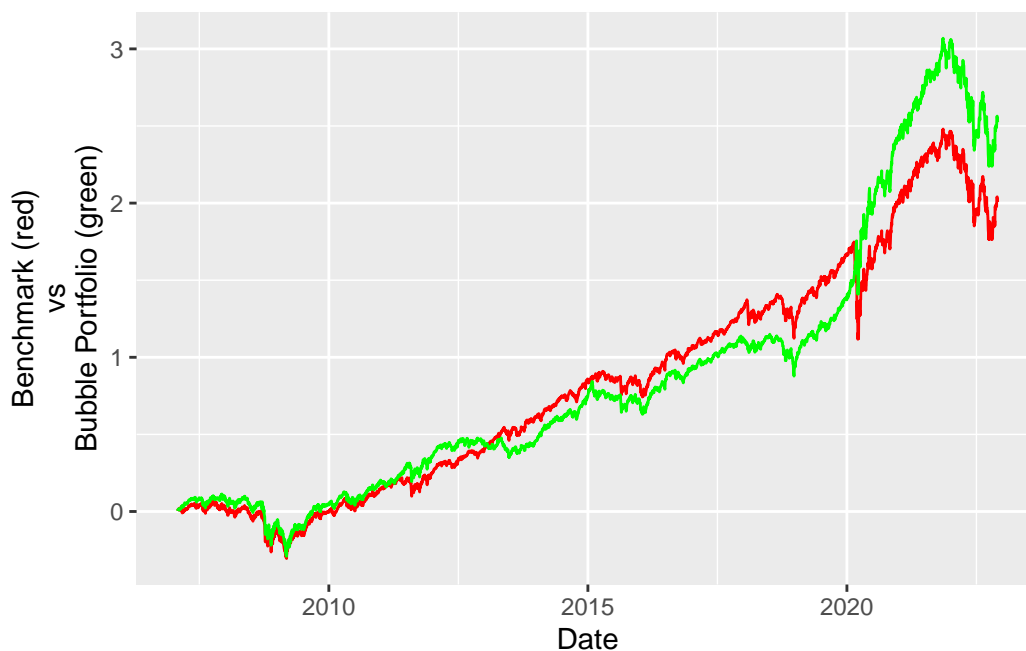


Construct Bubble Portfolio

```
m_df <- m_df %>% mutate(BubblePortfolio = case_when(
  conf >= POS_BUBBLE_THRESHOLD ~ ((m_df$TLT.Adj.Close * .5) + (m_df$AGG.Adj.Close * .5)),
  conf <= -NEG_BUBBLE_THRESHOLD ~ ((m_df$VTI.Adj.Close * .333) + (m_df$VBR.Adj.Close * .333)),
  -NEG_BUBBLE_THRESHOLD < conf & conf < POS_BUBBLE_THRESHOLD ~ ((m_df$VTI.Adj.Close * .2) + (m_df$VBR.Adj.Close * .2))
))

m_df$BubblePortfolioCumulativeReturns <- cumprod(1+m_df$BubblePortfolio)-1

ggplot(m_df, aes(x=Date)) +
  geom_line(aes(y=BenchmarkCumulativeReturns), color='red') +
  geom_line(aes(y=BubblePortfolioCumulativeReturns), color='green') +
  labs(x="Date", y="Benchmark (red)\nvs\nBubble Portfolio (green)")
```



Conclusion

The bubble portfolio does a nice job of avoiding the covid crash of 2020. However, it doesn't really add much more value other than that. There are a variety of parameters we could play with but that would likely just leave us overfit in some regard. This brings up an interesting point about bias when working on trading strategies. I think we could loosely call it p-hacking

where we fiddle with certain parameters until we have a very nice looking backtest that is grossly overfit.

```
sessionInfo()
```

```
R version 4.2.1 (2022-06-23)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Big Sur ... 10.16

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats      graphics  grDevices  utils      datasets  methods    base

other attached packages:
[1] forcats_0.5.2  stringr_1.4.1  dplyr_1.0.10  purrr_0.3.4
[5] readr_2.1.2    tidyr_1.2.0    tibble_3.1.8  ggplot2_3.3.6
[9] tidyverse_1.3.2

loaded via a namespace (and not attached):
 [1] tidyselect_1.1.2  xfun_0.32        haven_2.5.1
 [4] gargle_1.2.0      colorspace_2.0-3 vctrs_0.4.1
 [7] generics_0.1.3    htmltools_0.5.3  yaml_2.3.5
[10] utf8_1.2.2        rlang_1.0.4      pillar_1.8.1
[13] glue_1.6.2        withr_2.5.0      DBI_1.1.3
[16] dbplyr_2.2.1      modelr_0.1.9     readxl_1.4.1
[19] lifecycle_1.0.1   munsell_0.5.0    gtable_0.3.0
[22] cellranger_1.1.0  rvest_1.0.3      evaluate_0.16
[25] labeling_0.4.2    knitr_1.40       tzdb_0.3.0
[28] fastmap_1.1.0     fansi_1.0.3      broom_1.0.0
[31] scales_1.2.1      backports_1.4.1  googlesheets4_1.0.1
[34] jsonlite_1.8.0    farver_2.1.1     fs_1.5.2
[37] hms_1.1.2         digest_0.6.29    stringi_1.7.8
[40] grid_4.2.1        cli_3.4.1        tools_4.2.1
[43] magrittr_2.0.3    crayon_1.5.1     pkgconfig_2.0.3
[46] ellipsis_0.3.2    xml2_1.3.3       reprex_2.0.2
[49] googledrive_2.0.0 lubridate_1.8.0  assertthat_0.2.1
```

[52]	rmarkdown_2.16	httr_1.4.4	R6_2.5.1
[55]	compiler_4.2.1		