Mini-Project 5
Max Median Rule
Group 5
Dobelman
STAT 486 - Market Models

Connor Barnhill, Brian Graff, Frank Portman, Reid Minto

March 29, 2013

1 Introduction to the Strategy

The Max Median Rule is a stock picking strategy that was developed for the common man, as people began planning their own investing strategies like their 401k. It is easily used because all you need to be able to do is pull data from a source and model the data using an application like excel. The actual process involves pulling the daily returns for the SP500, taking the day-to-day ratios and looking at the 500 median returns of the stocks. Then you choose the stocks with the 20 highest median returns over the year hold them for one year and liquidate at the years end. This process is then repeated year after year. This strategy beats the SP 500 by 50%, but on down years is seen to have a downside of 151.4% compared to 123.1%. We decided to further expand on the already successful Max Median Strategy. We did this by looking at the 20 stocks with the highest geometric and harmonic means. The geometric mean is the nth root of the product of returns. A positive attribute of using this metric is it allows you normalize the range so no part of the range dominates the weighting. The harmonic mean is an average that is found by taking the sum of the reciprocals of all the returns and dividing them by the number of returns. This allows each data point to give an equal weight.

2 Data

To retrieve our data, we referenced the file found in Owlspace under projects which gives the constituent list for the SP500. From that file, we copy and pasted the PERMNOs into a .txt file to upload into the WRDS search. We used the relevant dates of 01/01/1969-12/31/2012, and got the data sorted by company name. To clean and process the data, we also referenced the SP500 constituent list. From our original data set, we eliminated observations that were outside of the date range for a given PERMNO. This gave us our final subset, sorted by date, onto which we applied our algorithms.

3 Introduce The Harmonic Mean Criterion

In order to improve upon the results, let's first demonstrate why the median is used in the first place as opposed to the arithmetic mean. The median and mean of a sample are both measures of the center of our data. The arithmetic mean, however, can be heavily influenced by the presence of just a few large outliers. For simplicity, suppose we have a sample of integers of 1, 2, 3, 4 and 20. The sample arithmetic mean will be 6, and the median will be 3. The reason we perform the median operator or take the arithmetic mean is to obtain a single statistic that gives us a measure of the central tendency or "average" behavior of the underlying data generating process of our data. The 20 in our sample above may be interpreted as an anomaly, since its distance to the center of the data (by any measure of the center) is much greater than all the other data points. Therefore in some situations we may prefer other measures of central tendency than the arithmetic mean that are more robust to large outliers.

When working with stock returns, we usually look at median or mean returns because the measure of mean behavior implicitly gives us an indication of the performance of a company which we hope could tell us what performance we may expect in the future from that company. Stock prices are arguably more susceptible to outside manipulations than many other kinds of data, so we may wish to discount daily returns that are anomalies, or that are too far away from the center of a stock's yearly return for other reasons than strong company performance. This is the reason that the median was chosen in the original max-median trading strategy as opposed to the arithmetic average. Therefore, when searching for a strategy to improve upon the max-median results, we searched for robust measures of central behavior and arrived at the geometric mean and the harmonic mean. The geometric and harmonic means, like the median, are measures of central tendency but are not too heavily influenced by large outliers. For instance, in the example data above of 1, 2, 3, 4, and 20, the arithmetic average is 6, the median 3, the geometric mean 3.44, and the harmonic mean 2.34. Instead of equally investing in the top 20 median return stocks from the previous year, we will attempt a trading strategy that equally invests in the top 20 stocks by geometric or harmonic means from the daily returns of the previous year.

4 Methodology

To compute daily returns, we first checked to make sure that the PERMNO for the previous entry was the same as the current entry to prevent creating false returns across different stocks. We then set today's return equal to today's price divided by yesterday's price for all days. Next we wrote a function that, given a year, returned the 20 PERMNOs with the highest median return. We also created a similar function that returned the 20 PERMNOs with the highest harmonic mean. We then created two functions to backtest using either the Max Median or Max Harmonic Mean criteria.

5 Commentary

As indicated in our performance tables, our Max Median portfolio did not do as well as we expected. In fact, the total return for the portfolio was 0.269. The Max Harmonic Mean portfolio did considerably better than Max Median, earning a total return of 5.684, but still underperforming the SP500 and the Dow. It is important to not that the Max Median portfolio is extremely volatile. In good years it seems to do fairly well, while in bad year it does very poorly. Volatility aside, this result is still not what we were expecting as all of the available literature states that the Max Median portfolio outperforms the SP500 by some margin. While we are fairly certain that we are procedurally correct, possible sources of errors in our analysis may lie in how we cleaned the data, or possibly even the data itself. Errors arising in the early years would compound and lead to even larger errors in later years.

6 Performance Tables

	Type	Max-Med	SPDiv	SP	DowNoDiv	Max HM
1	Total Return	0.269	58.828	15.875	15.616	5.684
2	CAGR	0.969	1.102	1.068	1.068	1.042

	Year	SPDiv	SP	DowNoDiv	Max-Med	Max HM
1	1971	1.14	1.11	1.06	1.10	1.06
2	1972	1.19	1.16	1.15	0.92	1.14
3	1973	0.85	0.83	0.83	0.65	0.82
4	1974	0.74	0.70	0.72	0.69	0.79
5	1975	1.37	1.31	1.38	1.35	1.34
6	1976	1.24	1.19	1.18	0.98	1.24
7	1977	0.93	0.88	0.83	0.81	0.86
8	1978	1.06	1.01	0.97	0.98	0.99
9	1979	1.19	1.12	1.04	1.25	1.01
10	1980	1.33	1.26	1.15	1.25	1.01
11	1981	0.95	0.90	0.91	0.66	0.86
12	1982	1.22	1.15	1.20	1.44	1.34
13	1983	1.22	1.17	1.20	1.01	1.15
14	1984	1.07	1.02	0.96	0.83	1.08
15	1985	1.32	1.27	1.28	1.18	1.21
16	1986	1.18	1.14	1.23	0.82	0.98
17	1987	1.05	1.02	1.02	0.80	0.93
18	1988	1.17	1.12	1.12	0.99	1.01
19	1989	1.31	1.27	1.27	1.10	0.88
20	1990	0.97	0.93	0.96	1.00	0.74
21	1991	1.31	1.27	1.20	1.16	1.12
22	1992	1.08	1.05	1.04	0.99	1.11
23	1993	1.10	1.07	1.14	0.94	1.00
24	1994	1.01	0.99	1.02	0.97	1.07
25	1995	1.38	1.34	1.33	1.20	1.24
26	1996	1.23	1.21	1.26	1.07	1.19
27	1997	1.34	1.31	1.23	0.99	0.85
28	1998	1.29	1.27	1.16	0.96	0.94
29	1999	1.21	1.20	1.25	1.12	1.02
30	2000	0.92	0.91	0.94	0.49	1.04
31	2001	0.88	0.87	0.93	0.77	1.00
32	2002	0.78	0.77	0.83	0.64	0.85
33	2003	1.29	1.26	1.25	1.43	1.50
34	2004	1.11	1.09	1.03	1.00	1.30
35	2005	1.05	1.03	0.99	1.22	0.97
36	2006	1.16	1.14	1.16	1.02	1.36
37	2007	1.06	1.04	1.06	1.19	1.08
38	2008	0.64	0.62	0.66	0.41	1.07
39	2009	1.26	1.23	1.19	1.26	1.12
40	2010	1.15	1.13	1.11	1.20	0.88
41	2011	1.02	1.00	1.06	1.11	1.14
42	2012	1.16	1.13	1.07	1.08	1.03

7 Code

```
################################################################
##              Mini Project 5 - Max Median Rule              ##
##  Connor Barnhill, Brian Graff, Frank Portman, Reid Minto   ##
################################################################

## Load Libraries
library(reshape2)
library(ggplot2)
library(plyr)
library(lubridate)
library(xtable)

## Load the .csv data from WRDS/COMPUSTAT
mmr.data <- read.csv("data.csv", stringsAsFactors = F)
load("mmr.RData")
load("mmr_clean.RData")
load("mmr_clean_returns.RData")

## Clean dates
mmr.data$date <- ymd(mmr.data$date)

## Save as .Rdata for faster loading
save(mmr.data, file = "mmr.RData")

## Load constituent list
constituents <- read.csv("constituents.csv", stringsAsFactors = F)

## Clean constituent list
constituents$start <- mdy(constituents$start)
constituents$ending <- mdy(constituents$ending)
constituents$start <- year(constituents$start)
constituents$ending <- year(constituents$ending)

## Remove stocks that werent in the S&P for whatever years
mmr.data$year <- year(mmr.data$date)

for(i in 1:length(constituents$PERMNO)) {

  x <- constituents$PERMNO[i]
  w <- which(mmr.data$year <= constituents$start[i])
  y <- which(mmr.data$year > constituents$ending[i])
  z <- which(mmr.data$PERMNO == x)

  remove <- c(w, y)
  remove <- unique(remove)

  remove <- intersect(remove, z)


  if(length(remove) != 0) {
```

```
    mmr.data <- mmr.data[-remove, ]
  }
}

## Save as .Rdata for faster loading
save(mmr.data, file = "mmr_clean.RData")


## Add returns
mmr.data$returns <- NA
returns.vector <- rep(NA, length(mmr.data$returns))
permnos.vector <- mmr.data$PERMNO
prc.vector <- mmr.data$PRC

for (i in 2:length(returns.vector)) {

  x <- permnos.vector[i]
  y <- permnos.vector[i - 1]

  if(x == y) {

    returns.vector[i] <- prc.vector[i] / prc.vector[i - 1]

  }

}

mmr.data$returns <- returns.vector

## Save as .Rdata for faster loading
save(mmr.data, file = "mmr_clean_returns.RData")


## Write a function to spit back the eligible PERMNOs

mmr.p <- function(YEAR) {

  sub <- subset(mmr.data, year == YEAR - 1)
  sub <- na.omit(sub)

  for(i in 1:length(constituents$PERMNO)) {

    x <- constituents$PERMNO[i]
    w <- which(sub$year <= constituents$start[i])
    y <- which(sub$year > constituents$ending[i])
    z <- which(sub$PERMNO == x)

    remove <- c(w, y)
    remove <- unique(remove)
```

```
    remove <- intersect(remove, z)


    if(length(remove) != 0) {

      sub <- sub[-remove, ]
    }
  }

  medians <- ddply(sub, "PERMNO", summarise, x = median(returns))
  medians <- medians[order(medians$x, decreasing = TRUE), ]

  perms <- medians$PERMNO[1:20]

  return(perms)

}

## Now we experiment with harmonic mean
hm.p <- function(YEAR) {

  sub <- subset(mmr.data, year == YEAR - 1)
  sub <- na.omit(sub)

  for(i in 1:length(constituents$PERMNO)) {

    x <- constituents$PERMNO[i]
    w <- which(sub$year <= constituents$start[i])
    y <- which(sub$year > constituents$ending[i])
    z <- which(sub$PERMNO == x)

    remove <- c(w, y)
    remove <- unique(remove)

    remove <- intersect(remove, z)


    if(length(remove) != 0) {

      sub <- sub[-remove, ]
    }
  }

  hm <- ddply(sub, "PERMNO", summarise, x = 1 / mean(1 / returns))
  hm <- medians[order(hm$x, decreasing = TRUE), ]

  perms <- hm$PERMNO[1:20]

  return(perms)

}
```

```r
## Make a function to backtest for MMR return for any given year
## using previous year's data
mmr <- function(YEAR) {

  sub <- subset(mmr.data, year == YEAR)
  sub <- na.omit(sub)

  perms <- mmr.p(YEAR)
  sub <- subset(sub, PERMNO %in% perms)

  rets <- ddply(sub, "PERMNO", summarise, x = prod(returns))

  ret <- mean(rets$x)

  return(ret)

}

## Harmonic mean backtester
hm <- function(YEAR) {

  sub <- subset(mmr.data, year == YEAR)
  sub <- na.omit(sub)

  perms <- hm.p(YEAR)
  sub <- subset(sub, PERMNO %in% perms)

  rets <- ddply(sub, "PERMNO", summarise, x = prod(returns))

  ret <- mean(rets$x)

  return(ret)

}



## Create our tables
years <- 1971:2012
mmr.returns <- c()
hm.returns <- c()

for(i in 1971:2012) {
  mmr.returns[i - 1970] <- mmr(i)
}

for(i in 1971:2012) {
  hm.returns[i - 1970] <- hm(i)
}
```

```
benchmark.table <- read.csv("benchmark.csv")
benchmark.table$MMR <- mmr.returns
benchmark.table$HMR <- hm.returns


### Calculate total return over 32 years and CAGR
total <- prod(benchmark.table$MMR)
cagr <- total^(1/42)
total.hm <- prod(benchmark.table$HMR)
cagr.hm <- total.hm^(1/42)
total.spd <- prod(benchmark.table$SPDividends)
cagr.spd <- total.spd^(1/42)
total.sp <- prod(benchmark.table$SPNoDividends)
cagr.sp <- total.sp^(1/42)
total.dow <- prod(benchmark.table$DowNoDividends)
cagr.dow <- total.dow^(1/42)

colnames(benchmark.table) <- c("Year", "SPDiv", "SP", "DowNoDiv", "
                               Max-Med", "Max HM")

xtable(benchmark.table)

tm <- matrix(nrow = 2, ncol = 6)
tm[1, 1] <- "Total Return"
tm[1, 2] <- round(total, 3)
tm[2, 1] <- "CAGR"
tm[2, 2] <- round(cagr, 3)
tm[1, 3] <- round(total.spd, 3)
tm[1, 4] <- round(total.sp, 3)
tm[2, 3] <- round(cagr.spd, 3)
tm[2, 4] <- round(cagr.sp, 3)
tm[1, 5] <- round(total.dow, 3)
tm[2, 5] <- round(cagr.dow, 3)
tm[1, 6] <- round(total.hm, 3)
tm[2, 6] <- round(cagr.hm, 3)

tm <- as.data.frame(tm)
colnames(tm) <- c("Type", "Max-Med", "SPDiv", "SP", "DowNoDiv",
                  "Max HM")

xtable(tm)
```