Pandemic-proofing Out-of-sample Portfolio Evaluations

H. D. Vinod [*]

August 15, 2023

Abstract

Evaluation of the performance of portfolios and of various methods of ranking them has to be out-of-sample. Otherwise, selection methods that fit the past data best would always win. Suppose the time series chosen for out-of-sample evaluation happens to have any (upward, downward, zigzag) trend. In that case, portfolio selection methods for that trend will work best but fail in general. We describe algorithms for the removal of such bias by using randomization. The R package ‘generalCorr’ has them. We use 169-month Dow Jones stock data to illustrate `outOFsamp()`, `outOFsell()`.

1 Introduction

Our illustration uses six stock-picking algorithms. We leave detailed descriptions of the algorithms outside the scope of this short paper. Our data matrix of returns denoted by ‘mtx’ $\{x_{j.t}\}$ has $t = 1, 2, \ldots N_j$ rows and $j = 1, \ldots, p$ columns for p stocks. The individual returns are earned at times t. Our example in Section 2 has $p = 28$ stocks from among those in the Dow Jones Industrial Average (DJIA) where $N_j = 169$ months, making the dimensions of ‘mtx’ (169×28). Given a capital of K dollars, the portfolio selection amounts to allocating a certain proportion $(w_j, \in \{[-1, 1]\})$ of K to the j-th stock, where negative allocation means short-selling that stock.

2 Performance of Dow Jones Stocks

Our data (using the R package ‘quantmod’ function ‘monthlyReturn’) start in January 2007 and end in January 2021, comprising 169 monthly returns. We choose only 28 out of 30 components of the Dow Jones Industrial Average (DJIA) because two of the 30 ticker symbols, ‘DOW’ and ‘V,’ have too many missing data months.

[*] address: H. D. Vinod, Professor of Economics, Fordham University, Bronx, New York, USA 10458. E-mail: vinod@fordham.edu. JEL codes C30, C51. Keywords: portfolio choice. No funding was received.

Various stock-picking methods yield their own ranking of p stocks. We have $p = 28$. We begin with commands to clean up the memory of R and get the DJIA data called dj28 from a website. We also call the generalCorr package in the following R code.

```
rm(list=ls())
options(width=60)
lin="https://faculty.fordham.edu/vinod/dj28.txt"
dj28=read.table(file=lin,header=T)
library(generalCorr)
options(np.messages=FALSE)
```

Now we call a function to compute the deciles of the implicit probability distribution $f(x_j)$ of returns from investing in the j-th stock. We can define the dominance of a stock by the size of return earned at each of its nine deciles.

The R function to rank stocks by an average of nine deciles is called as follows.

```
dv1=decileVote(dj28)
round(dv1$out[20,],0)
```

The output of the above code gives the rank of each stock labeled by its ticker symbol, except that here we have denoted all symbols using lowercase.

```
unh   gs   hd amgn msft   ba  cat  mcd  crm  hon  dis  mmm
2   17    5   14    6   11    8   10    4    7    9   12
jnj  trv  jpm  axp  nke  wmt   pg  ibm aapl  cvx  mrk intc
19   18   15   20    3   22   24   25    1   28   21   12
vz   ko  wba csco
26   16   27   22
```

A similar function computes the rank based on moments where a larger mean and skewness are desirable while lower variance and kurtosis are desirable. Using signed weights, one can rank the DJIA stocks by the moment method. The code is next. There are some default weights on the first four moments. The overall rank is a weighted average. The reader is invited to adjust the default weights to his liking.

```
mv1=momentVote(dj28)
round(mv1[12,],0)
```

The twelfth row of the output matrix has the ranks so that rank 1 (crm) is the best stock to buy, and rank 28 (pg) is the best to sell. The output of the code follows.

```
h         gs         hd        amgn       msft         ba
4         18         12           7          3         10
cat        mcd        crm        hon        dis         mmm
8         14         1         11          4         23
jnj        trv        jpm        axp        nke         wmt
22         16         14          6          9          17
```

```
pg        ibm       aapl        cvx        mrk       intc
28         26          2         20         24         15
vz         ko        wba       csco  smpReliWt
27         25         20         19         NA
```

We have described the decile and moment methods for picking stocks. A third method of interest is stochastic dominance (SD), which comes in four flavors, for four orders of dominance denoted as SD1 to SD4, and their summary is denoted as SDAll4 in this paper. Details available in the textbook by Vinod (2022) are omitted for brevity.

We conclude the section by recognizing that different criteria will generally yield different choices to buy. These rankings are based on the entire data and are in-sample. We have only one realization and no randomization so far.

3 Pandemic-proof out-of-sample evaluations

One of the problems with defining out-of-sample time periods used in evaluating portfolio selection algorithms is that the peculiar characteristics of the particular time periods can bias the evaluations. Covid-19 was a rare, once in a hundred years, pandemic that caused unexpected economic disruptions around the world.

If a portfolio manager wants to evaluate the performance of three (SDAll4, decile, moment) (say) algorithms using stock return data till 2019 for estimation while leaving out the stock returns data from pandemic months during 2020 to 2021 as out-of-sample for comparing the three algorithms. Clearly, this will be a biased sample of out-of-sample months with abysmal returns in many sectors of the economy. The portfolio choice algorithms with a pessimistic bias will obviously perform better if the three algorithms are judged on the basis of their performance during the pandemic months.

A fair comparison among the three algorithms requires identical allocation of K, once the ranking is known. We suggest choosing 'maxChosen' stocks for non-zero weights and a simple linearly declining allocation. In our illustration, maxChosen=2, and the capital K is allocated as (2/3, 1/2) for the top two stocks.

The total return starting with K capital is known for each of the three competing portfolio choice algorithms, for each random choice of 5% data left for out-of-sample evaluation. Now, the average performance of each of the three algorithms over 100 (say) randomly chosen out-of-sample data is claimed to be unbiased or pandemic-proof.

An R algorithm `outOFsamp(mtx, maxChosen=2, reps=100)` implements such pandemic-proof evaluation of three portfolio selection algorithms out-of-sample for the data on returns from 28 stocks of DJIA studied in the previous section.

It is interesting to check if zero-cost profitable arbitrage exists out-of-sample for investing in Dow Jones stocks. Let us review the R commands to check it. First, we illustrate the use of `outOFsamp(mtx, maxChosen=2, reps=100)` for out-of-sample evaluations.

```
obuy=outOFsamp(mtx,verbo=FALSE,maxChosen=2, reps=100)
round(head(obuy,1),4)
```

The left side of the above code means the output is saved to the R object named 'obuy.' The choice `maxChosen=2` means we choose the top two stocks for buying. The algorithm uses linear weighting by allocating 2/3 of capital K to the top stock and 1/3 of K to the second-ranked stock. Note that stock rankings will change for each method and for each subset (say 95%) of data chosen for 'in-sample' analysis. The R object 'obuy' has reps=100 rows for each out-of-sample evaluation of seven stock-picking algorithms compared by the outOFsamp() function reported in seven columns and identified by column headings. The output of the above code is next.

```
> round(head(obuy,1),4)
   SD1      SD2      SD3    SD4 SDAll4   decile   moment
 12732.24 12.8393 0.0296 0.0296 3.6141 12728.65 6371.886
```

We find that the ranking of stocks by SD1 to SD4 is identical, even though their magnitudes are different. Hence, aggregating SD1 to SD4 into one method called SDAll4 is appropriate here. That is, we might work with the last three columns of the about output.

Now we summarize the object 'obuy' by computing column-wise means and standard deviations.

```
a1=apply(obuy,2,mean)
a1sd=apply(obuy,2,sd)
```

Now we turn to illustrate the selling of the poorest performing stocks. We summarize the R object for selling called 'osell' by computing column-wise means and standard deviations.

```
osell=outOFsell(mtx,verbo=FALSE,maxChosen=2, reps=100)
a2=apply(osell,2,mean)
a2sd=apply(osell,2,sd)
```

We shall compress the seven-column output into only three columns entitled SDAll4, decile, and moment, respectively.

```
a12=rbind(a1,a2)
a123=a12[,5:7]
a12sd=round(rbind(a1,a1sd,a2,a2sd),3)
rownames(a12sd)=c("buy best","buy sd", "sell worst","sell sd")
a123sd=a12sd[,5:7]
print(a123sd)
```

The output with three columns for three methods follows. It has averages and standard deviations of returns computed over 100 repetitions of the randomized out-of-sample dataset.

```
> print(a123sd)
name        SDAll4 decile moment
buy best    0.026  0.020  0.027
buy sd      0.028  0.024  0.030
sell worst  0.017  0.006  0.016
sell sd     0.018  0.021  0.015
```

We want the average out-of-sample return along the first row named 'buy best' (stock) in the above output table to significantly exceed the average out-of-sample return along the third row named 'sell worst.' Otherwise, the method in the corresponding column is failing to give good rankings of stocks. Let us use the usual tools for testing the difference between two means by using the standard deviations.

3.1 Statistical Significance of Difference of Means

Each method ranks the 28 stocks, and we know which two are the best and worst by that method. When we compare the three methods in the above table, we randomize our 169 months of data to determine the out-of-sample set. We have 100 such sets with 100 distinct returns from the best two stocks and the worst two stocks for each method.

We seek to compute a zero cost arbitrage where the idea is one short sells a dollar worth of the worst stock in DJIA and buy the appropriate fraction of the best stock as determined by each method identified in columns.

Let $\bar{x}_b$ denote the average return over 100 different best stocks for each random choice of the out-of-sample set. Let the standard deviation of these 100 numbers be denoted by σ_b. The subscript 'b' is for the best two stocks. The averages are along the first row and standard deviations are along the second row in the output table above.

Similarly, let $\bar{x}_w$ denote for each method (each column) the worst-performing stock in the particular out-of-sample set. Let σ_w denote the analogous standard deviation. The subscript 'w' identifies the worst stocks.

Under standard assumptions, we are interested in the existence of a statistically significant zero-cost arbitrage strategy out-of-sample. Let N (=100) denote the number of replications. The test statistic is

$$t_z = (\bar{x}_b - \bar{x}_w)/SE, \tag{1}$$

where the standard error SE is the square root of the variance. We have,

$$SE = \surd(\sigma_b^2/N + \sigma_w^2/N). \tag{2}$$

The following loop computes the t_z statistics for the three methods compared here.

```
out=rep(NA,3)
for(j in 1:3){
a4num=(a123sd[1,j]-a123sd[3,j])
a4v=(a123sd[2,j]^2/100-a123sd[4,j]^2/100)
out[j]=a4num/sqrt(a4v)}
print(out)
qt(0.975,df=99)
```

The estimated t-statistics for the three columns are (SDAll4 =4.196, decile = 12.049, and moment= 4.234). The t-table in R using code `qt(0.975,df=99)` gives 0.975 quantile at 1.984217 for 99 degrees of freedom. All three t_z values exceed this critical value for the usual 95% confidence level. They show that all methods yield statistically significant zero-cost arbitrage returns out-of-sample.

The decile statistic $t_z = 12.049$, being the largest of the three, suggests that ranking nine deciles of the 28 stock returns and averaging those ranks appears to be good here. More research is needed before reaching definitive conclusions. We have merely illustrated some free tools for comparing stock-picking strategies pandemic-proof and out-of-sample.

4 Final Remarks

The definition of out-of-sample data often refers to a particular time period toward the end of available time-series data. Unfortunately, such trailing time series can be subject to unknown biased evaluations, especially when the out-of-sample period happens to be rather unusual. We use recent stock market data to demonstrate how to remove such bias, that is, make the evaluations "pandemic proof." The algorithms for pandemic-proofing out-of-sample evaluations are `outOFsamp(mtx, maxChosen=2)` for buyers and `outOFsell(mtx, maxChosen=2)` for stock sellers.

References

Vinod, H. D., 2022. Hands-on Intermediate Econometrics Using R: Templates for Learning Quantitative Methods and R software, Second edition. World Scientific, Hackensack, NJ, ISBN 978-981-125-617-2, https://doi.org/10.1142/12831.
URL `https://www.worldscientific.com/worldscibooks/10.1142/12831`