EVOLUTIONARY COMPUTATION

FINAL PROJECT REPORT

Okan İhsan Bağrıaçık
okanihsanbagriacik@pota.mu.edu.tr

Thursday 24th June, 2021

1 Introduction

Portfolio optimization is one of the most interesting fields of study of financial mathematics. Since the birth of Modern Portfolio Theory (MPT) by Harry Markowitz, many scientists have studied a lot of analytical and numerical methods to build the best investment portfolio according to a defined set of assets. The power of genetic algorithms makes it possible to find the optimal portfolio.

In dealing with this kind of Optimization problem, Harry Markowitz 1959 developed a quantitative model, also called mean-variance model. The mean-variance model has been usually considered as either the minimization of an objective function representing the portfolio variance (risk) for a given level of return or the maximization of an objective function representing the portfolio return for a given level of risk.

I used Genetic Algorithm to find the weights such that we maximize the returns and at the same time minimize the risk. Sharpe Ratio will be used to evaluate the fitness. I used Python programming language to write my code.

2 SYSTEM PROPOSAL

I took 6 different .csv files about stock values. These Stocks information are belong to HDFC Bank, ITC Limited, Larsen Toubro, Mahindra Mahindra, Sun Pharmaceutical Industries Ltd., Tata Consultancy Services. I chose these because I found a website of these in same format. I used Colab Google to demonstrade my code. I chose Colab to give details about every part of the code.

3 IMPLEMENTATION

First I imlemented the data sets and combined them into one data frame.

```
files=['hdfc.csv','itc.csv','l&t.csv','m&m.csv','sunpha.csv','tcs.csv']
dfs=[]

for file in files:
    temp=pd.read_csv(file)
    temp.columns=['Date',file.replace('.csv','')]
    dfs.append(temp)

stocks = reduce(lambda left,right: pd.merge(left,right,on='Date'), dfs)
print(stocks.shape)
stocks.head()
```

Figure 1: Combining Data Sets Into One

I then calculated the historical returns of these data at 3,6,12,24 and 36 months.

```
def hist_return(months):
    ''' It calculates Stock returns for various months and returns a dataframe.
        Input: Months in the form of a list.
        Output: Historical returns in the form of a DataFrame. '''
    idx=[]
    df=pd.DataFrame()
    for mon in months:
        temp=(stocks.iloc[0,1:] - stocks.iloc[mon,1:])/(stocks.iloc[mon,1:])
        idx.append(str(mon)+'_mon_return')
        df=pd.concat([df, temp.to_frame().T], ignore_index=True)
    df.index=idx
    return df
```

Figure 2: Calculating Historical Returns

Then I identified the genes. I chose these as weights. After this stage, I went to the part of determining the chromosomes and made a study so that 6 genes represent 1 chromosome. To create my population, I created a 2D population of random chromosomes.

```
def chromosome(n):
    ''' Generates set of random numbers whose sum is equal to 1
        Input: Number of stocks.
        Output: Array of random numbers'''
    ch = np.random.rand(n)
    return ch/sum(ch)
```

Figure 3: Chromosome Creation

```
n=6 # Number of stocks = 6
pop_size=100 # initial population = 100

population = np.array([chromosome(n) for _ in range(pop_size)])
print(population.shape)
print(population)
```

Figure 4: Population Function

I then created a fitness function to determine fitness. I set this function as S = (μ r)/. I then identified an elite population. The reason I set this up was to retrieve high value data.

```
def Select_elite_population(population, frac=0.3):
    ''' Select elite population from the total population based on fitness function values.
        Input: Population and fraction of population to be considered as elite.
        Output: Elite population.'''
    population = sorted(population,key = lambda x: fitness_fuction(x),reverse=True)
    percentage_elite_idx = int(np.floor(len(population)* frac))
    return population[:percentage_elite_idx]
```

Figure 5: Elite Population Function

Then I did my mutation and crossover operations. I used 2 different methods in my crossover process. I've run it from worst parent to best. That way, I'll be able to get the best genomes. Then I did Iterate Process and finalized the process.

4 CONCLUSION

As a result of my iteration processes, I got the stock values and found the risks and expected treatments. In this way, I have created a nice portfolio for people who will actually trade on the stock market. Financial developments and stock market monitoring is one of the most important financial issues today. With this method, we have created our portfolio and seen the expected values and risks. Thanks to the genetic algorithm, we have been able to see what the stock markets can do to us without being dependent on foreign news.

```
Portfolio of stocks after all the iterations:

hdfc : 0.03338912374928946
itc : 0.25892793802773834
l&t : 0.08571133727686643
m&m : 0.06183100580231876
sunpha : 0.27661979411361814
tcs : 0.28352080103016897

Expected returns of 0.14453015567745017 with risk of 0.0002592181124772639
```

Figure 6: Outputs of Optimization Model