Multi-period Portfolio Optimization Model with Investors' Sentiments

Objectives

- To introduce investors' sentiments into the standard multi-period Mean-Variance portfolio selection model with the help of prospect theory.
- To analyze the impacts of investors' sentiments on portfolio selection model performance.

In multi-period portfolio model, investors can update capital allocations on securities for seeking benefits or risk-aversion at each period. Asymmetric investor sentiments play an important role in updating portfolio strategy and must be incorporated into the multi-period portfolio optimization model. In this project, asymmetric investors sentiments are modeled by prospect theory to iterate investors' expected return level at each period, and a Dynamic Sentiment Adjusted Model (DSAM) is implemented in which the objective is to minimize the risk of the portfolio at each period.
In this model, at most k securities can be bought out of N securities, where k is small compared to N to reduce transaction cost. This is represented by the l_0 constraint.

Definitions

Investors' sentiments: Investors' sentiments are defined as the market participants' beliefs towards future cash flows.

Asymmetric investors' attitude: The investors are more sensitive towards negative loss than positive return, i.e their attitude is asymmetric.

Asymmetric investors' sentiments: It is the combination of investors' sentiments and asymmetric investors' attitudes that represents that investors change their expected return level for the next period based on the investment results of the current period as well as their sentiments.

Methodology

Multi-period MV portfolio optimization model

The objective of the standard MV model is to minimize the risk i.e. variance of the portfolio, given the expected return. The MV optimization problem for a period t can be expressed as :

$$\min_{w} \; w_t^T Q_t w_t$$

$$X_t^T w_t \geq E_{r(t)}$$

$$E_{r(t+1)} = E_{r(t)}$$

$$\sum_i w_t^i = 1$$

$$0 \leq w_t^i \leq 1 \qquad i = 1,2,\ldots,N$$
$$||w_t||_0 \leq k \qquad t = 1,2,\ldots,T \qquad \ldots(1)$$

Where at period t, w_t is the vector of securities weights, Q_t is the covariance matrix of securities, X_t is the vector of return rate of securities, $E_{r(t)}$ is the expected return level of investor.
In the standard multi-period MV model, the expected return level of investor for all periods are equal.

Asymmetric investors' sentiments and Prospect Theory

Asymmetric investors' sentiments is formulated by the value function of the Prospect Theory.

We assume the investors' expected risk level as the reference point in the prospect function. Let R_r be the investors' expected risk level. Δ_r represents the gain or loss relative to the reference point and is given by:

$$\Delta_r = w_t^T Q_t w_t - R_r \qquad \ldots(2)$$

where, $w_t^T Q_t w_t$ is the risk in current period

The value function of the prospect theory is given by:

$$f(\Delta_r) = \begin{cases} \lambda(-\Delta_r)^\alpha & \Delta_r < 0 \\ -\theta(\Delta_r)^\beta & \Delta_r \geq 0 \end{cases} \qquad \ldots(3)$$

Where, λ and θ are adjustment coefficients of gain and loss for investment risk, and 0 < α, β < 1 indicates that the sensitivity of investors to returns and losses is decreasing.

The dynamic iteration rule of investors' expected return level formulated with the asymmetric investor sentiment is given by (Wei, Yang, Jiang & Liu, 2021) :

$$E_{r(t+1)} = E_{r(t)} (1 + f(\Delta_r)) \quad ...(4)$$

The DSAM Model

In the DSAM (Wei, Yang, Jiang & Liu, 2021) model, the asymmetric investors' sentiments are incorporated into the standard multi-period MV model by iterating the expected return level of each period based on the value function of prospect theory. For the portfolio selection problem, k securities are to be selected from N securities and capital is allocated to the selected securities.

The stochastic neural networks algorithm with re-parametrization trick is used to select k securities out of N securities. Due to the presence of l_0 constraint, it is difficult to train the stochastic neural network with discrete neurons because the back propagation algorithm can not be applied to non-differentiable functions. By Gumbel-Softmax sampling method, re-parametrization trick is used to sample the non-differentiable sample with a differentiable sample. In the SNNrP algorithm, a combination of k or fewer securities is selected in turn and by continuous loop iteration optimal results are provided. In this process, k or fewer securities indices are selected.

After selecting k' (≤ k) number of securities, the optimal weights of the securities are determined by quadratic programming. After securities selection, the optimization problem for period t for the sentiment adjusted model can be expressed as:

$$\min_{w} \; w'^{T}_{t} Q'_{t} w'_{t}$$

$$X'^{T}_{t} w'_{t} \geq E_{r(t)}$$

$$E_{r(t+1)} = E_{r(t)} (1 + f(\Delta_r))$$

$$\sum_i w'^{i}_{t} = 1$$

$$0 \leq w'^{i}_{t} \leq 1 \qquad i = 1,2,\ldots,k' \quad ,t = 1,2,\ldots,T \qquad \ldots(5)$$

Where, $X' \in \mathbf{R}^{k'}$ is the return rate of the k' securities, $Q' \in \mathbf{R}^{k' \times k'}$ is the covariance matrix of the k' securities. w' is the vector of weights of the k' selected securities.

The DSAM model has the following steps:

i) Select k securities from N securities using Stochastic Neural Network algorithm with re-parametrization trick.

ii) Allocate capital to selected securities. The optimal weights for securities is determined using quadratic programming and the return and risk of each period is obtained.

The Stochastic Neural Networks with re-parametrization trick Algorithm

The profit and cardinality constrained risk minimization problem is NP-hard because of the existence of the l_0 constraint. Due to the presence of non-differentiable samples, back propagation algorithm cannot be applied. By Gumbel-Softmax sampling method, re-parametrization trick is used to sample the non-differentiable sample with a differentiable sample and then stochastic gradient descent can be applied to update the model parameters to minimize the loss function.

The input layer of the algorithm is the mean log returns of securities $X \in \mathbf{R}^{\mathbf{N}}$. The hidden layer is the probability that a stock is selected $\pi \in \mathbf{R}^{\mathbf{k \times N}}$. The output layer is the portfolio return and risk predicted by the model.

Securities Selection Process

To understand the selection process, we can imagine that we have k bags, each bag having N balls corresponding to stocks. The selection of assets can be thought of as drawing a ball from a bag where the probability of drawing a ball indexed j from bag i is π_{ij}. Learning these probabilities to optimize the loss function is the key to solving the selection problem.

We use a one-hot encoding vector to record the outcome of the selection process.

$$z_i = \text{one\_hot}(\text{argmax}([\pi_{i1}, \pi_{i2}, \ldots\ldots, \pi_{iN}]))$$

For example, for the first turn if 5^{th} asset is picked then z_1 = [0,0,0,0,1,0,….,0]. At the end of the process, we have k one-hot vectors $\{z_1, z_2, \ldots\ldots, z_k\}$.

Taking $z = \sum_i z_i$, we get $z \in \mathbf{R}^{\mathbf{N}}$ which will be a mask vector for asset and only those assets will be selected for which the corresponding entry in z is non-zero. It is easy to see that z has at most k non-zero entries.

Re-parametrization

The probability π_{ij} that a stock is selected is bounded in [0,1]. These can be generated in terms of unconstrained parameters $S \in \mathbf{R}^{k \times N}$ by applying a softmax function.

$$\pi_{ij} = \frac{\exp(S_{ij})}{\Sigma_j \exp(S_{ij})} \qquad \ldots(6)$$

Capital Allocation Weights

The allocation weights can be generated from unconstrained parameters $\hat{w} \in \mathbf{R}^{N}$ in 3 steps:

i) Element-wise exponentiation: $w' = [\exp(\hat{w}_1), \exp(\hat{w}_2), \ldots\ldots, \exp(\hat{w}_N)]$.

ii) Element-wise product with z: $w'' = w' \odot z$.

iii) Normalization: $w = \frac{w''}{\Sigma_i w''_j}$.

It is ensured that w satisfies the constrains: $w_i \geq 0$, $\Sigma_i w_i = 1$, $||w||_0 \leq k$.

Loss Function

The loss function in the stochastic neural networks algorithm is taken as:

$$\min_{S, \hat{w}} \frac{E_r - X^T w}{w^T Q w \sqrt{(E_r - X^T w)^2 + \epsilon}} \qquad \ldots(7)$$

Where ϵ is small. This function is minimum when $E_r - X^T w \leq 0$ and $w^T Q w$ is minimum, ensuring that in optimal solution, $X^T w \geq E_r$ and risk is minimized.

Gumbel-Softmax estimation

Since, z are discrete hidden neurons, back propagation cannot be applied as z is a non-differentiable sample. To solve this, z is sampled from Straight-Through Gumbel Softmax estimator (Zheng, Chen, Hospedales & Yang, 2020) in which, the eq(8) is used in the forward pass,

$$z_i = \text{one\_hot}(\ \text{argmax}(\ g_{ij} + \log(\pi_{ij})\ )\ ) \qquad \ldots(8)$$

and the eq(9) is used in the backward pass.

$$z_i = \text{softmax}(\, g_{ij} + \log(\pi_{ij})\,) \qquad \ldots(9)$$

where, $[g_{i1}, g_{i2}, \ldots\ldots, g_{iN}]$ are i.i.d samples drawn from Gumbel(0,1) distribution.

Straight-Through Gumbel Softmax estimator gives a true one-hot encoding vector and the sample drawn from ST-Gumbel estimator is differentiable. Now, the gradient descent algorithm can be used, where the parameters S, $\hat{w}$ are updated for the optimization of the loss function.

The output of the SNNrP algorithm will give the mask vector z for security selection. After obtaining z, the optimization problem (5) can be solved by quadratic programming.

DSAM Algorithm

Algorithm for the sentiment adjusted model :

Input : The log returns of securities $X \in \mathbf{R}^N$ for each period, the covariance matrix $Q \in \mathbf{R}^{N\times N}$ of securities for each period, the initial expected return level E_r , risk tolerance level R_r, the upper number of securities k.

Output : The vector of securities weight w, investment return and risk at each period and the ratio of return and risk at each period.

The step-wise procedure for period t:

Step 1: Using the stochastic neural networks algorithm with re - parametrization trick, k securities are selected from N securities and the mask vector z is returned.

Step 2: The optimal weights of the selected securities is determined by solving the quadratic optimization problem and the risk and return of period t is obtained.

Step 3: The value function of the prospect theory is calculated using Eq.(3)

Step 4: The expected return level of next period is calculated:

$$E_{r(t+1)} = E_{r(t)}\,(1 + f\,(\Delta_r))$$

This procedure is repeated for every investment period.

Data

The effectiveness of the DSAM model is analyzed on numerical experiments done on 30 companies of S&P500 indexed by market capitalization [3]. The companies ranking from 1-30 are considered for the experiment. The data sets included historical data from Yahoo Finance. For the experiment 8 investment periods of 6 months each are considered from July 1, 2016 to July 1, 2020.

The algorithm is coded in python3.6 in the framework of PyTorch.

Conclusion

In this project, the asymmetric investors' sentiments are modeled by the prospect theory. The investor sentiments are incorporated into the DSAM model in which the expected return level of each period is iterated according to both investment results of the current period as well as their sentiments. Investors' portfolio selection and capital allocation is also effected by their own preferences for risk and returns.

The Stochastic Neural Networks algorithm with re-parametrization trick used gives an effective method to solve the cardinality and profit constrained portfolio optimization problems.

Overall, the DSAM model provides an effective way to control investment risk and solve the multi-period sparse portfolio selection problems.

References

1. Wei, J., Yang, Y., Jiang, M. and Liu, J., 2021. Dynamic multi-periodsparse portfolio selection model with asymmetric investors' sentiments. *Expert Systems with Applications*, *177*, p.114945.

2. Zheng, Y., Chen, B., Hospedales, T.M. and Yang, Y., 2020, April. Index tracking with cardinality constraints: A stochastic neural networks approach. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 01, pp. 1242-1249).

3. https://www.slickcharts.com/sp500