Extreme Value Scenario Tree Generation Approaches

Li YANG [1, a], Weize Wang[1, b]

[1] School of Economics & Management, Guangxi Normal University, Guangxi, Guilin 541001, China

[a]email: yangli65608@163.com, [b]email:weizew@gmail.com

Keywords: Scenario; Scenario tree; Extreme value k-means clustering

Abstract. In actual investment decision activity, investors are usually risk averse and always pay more attention to the tail characteristic of the loss distribution. However, the tail matching effect is not considered in existing scenario tree generation approaches. In light of this phenomenon, a new extreme value k-means clustering method is first presented to generate the scenario tree, and also, by simultaneously utilizing simulation, extreme value k-means clustering, we devise one new multistage scenario tree generation approach. Empirical results show that the better actual performance effect of the new multistage scenario tree generation approach.

Introduction

Scenario generation is a key step to transform the random decision programming model into its deterministic problem. The number of scenarios and the branching structure of the scenario tree directly influence the complexity.Šutienė et al. (2010) [1] designed a multi-period k-means clustering method to generate the scenario tree. Although clustering techniques can reduce the scale of scenario, However, due to the randomness of the initial clustering center selection in k-means clustering method, it is possible to lead to the selection of the representative node distribution is relatively concentrated, can not reflect the distribution of data in extreme cases, however extreme scenarios are often concerned by investors. Considering the above reasons, by adding the extremum samples into the initial clustering centroids of the k-means clustering algorithm, we will devise an extremum k-means clustering method for scenario tree generation.

Compared with the k-means clustering algorithm, the extremum k-means clustering algorithm can fit the tail of the data distribution; however, based on this, it may be very difficult to fit the statistical properties of the original data. In view of this, there are a lot of literature, such as Gülpınar et al. [2] designed a scenario tree generation method from the matching of the statistical properties of the data process angle, which compared to the classical moment matching method see Høyland and Wallace (2001) [3]. Based on above discussion, we design a multi-stage scenario tree generation method by combining the new designed extremum k-mean clustering algorithm and the linear moment matching method.

Extreme Scenario Tree Generation Approaches

The generation of data process is the precondition of the financial management activities. Due to the precision requirement, the size of the scenario derived by the simulation method VAR (r)-DCC-MVGARCH (p,q) [5] is often large, the k-means clustering method (Algorithm 1 for short) are usually used to reduce the number of scenarios by choosing the representative scenario. However, due to the randomness of the initial clustering center selection in Algorithm 1, the ultimately selected representative scenarios not well reflect the tail situation of the data distribution. But extreme results of random variables (extreme value) are often concerned in the risk management. To add extreme scenarios to the k-means clustering algorithm, we design a new extremum k-means clustering algorithm, as shown in algorithm 2.

It may be assumed that the path number derived by VAR (r)-DCC-MVGARCH (p,q) is S, denoted respectively $\xi^s = (\xi_1^s, \xi_2^s, \cdots, \xi_T^s), s = 1, \cdots, S$, where ξ_1^s is the root node. For convenience, it is assumed that the generated scenario tree has a symmetrical branching structure $(b_1, b_2, \cdots, b_T)$, that is,

all scenarios passing through any non leaf node are divided into b_t classes $C^1,\cdots,C^{b_t}$.

Algorithm 2: Extremum k-means clustering algorithm

S1 Initialization. Define the branching structure of the scenario tree, i.e., the number of branches b_k from each non-leaf node k. Make $t=1, k_t=1$.

S2 Clustering centroid selection. Two steps. First, from $\{\xi_{t+1}^s\}_{s\in\{1,\cdots,s_{k_t}\}}$ connected with the node k_t, search the scenarios containing the worst and the best results of the asset $i(i=1,\cdots,n)$. Remember all those scenarios (select only one if there are several vectors with the same extremal component) as for ξ_{t+1}^i, $i=1,\cdots,m$, If there is no duplication, then m=2n. The second step: from $\{\xi_{t+1}^s\}_{s\in\{1,\cdots,s_{k_t}\}}$, we randomly choose $b_{k_t}-m$ scenarios different from $\{\xi_{t+1}^i\}_{i\in\{1,\cdots,m\}}$, and together with the ξ_{t+1}^i, $i=1,\cdots,m$, are denoted as the initial cluster centroids.

S3 Clustering assignment. for each sub node ξ_{t+1}^s connected with k_t, if ξ_{t+1}^i is nearest to ξ_{t+1}^s, that is, $l=\arg\min_{i\in\{1,\cdots,b_{k_t}\}} d(\xi_{t+1}^s,\xi_{t+1}^i)$, then ξ_{t+1}^s is classified to the cluster C^l, where $d(\xi_{t+1}^s,\bar{\xi}_{t+1}^i)=\|\xi_{t+1}^s-\bar{\xi}_{t+1}^i\|_2$.

S4 Clustering update. Compute the mean of all branches of the class C^i, denoted as $\bar{\xi}_{t+1}^i$, $\bar{\xi}_{t+1}^i=E\{\xi_{t+1}^s\}_{\xi_{t+1}^s\in C^i}$, $i=1,\cdots,b_{k_t}$.

S5 Probability calculation. Select $\bar{\xi}_{t+1}^i, i=1,\cdots,b_{k_t}$ as child nodes of node k_t, and the probability assigned to $\bar{\xi}_{t+1}^i$ equals the sum of probabilities of all those scenario branches belonging to the cluster C^i.

S6 Termination test. If $k_t<N_t$ make $k_t=k_t+1$, and return to **S2**. Otherwise, make $t=t+1$, if $t=T$, stop; otherwise, let $k_t=1$, return to S2.

Whether the algorithm 1 or algorithm 2, they cannot guarantee that statistical properties of the derived data process may match that of the simulation path. Considering this point, we will design a new multi-stage scenario tree generation method by combining the linear moment matching model (LP1) in the paper [4].

We first assume that N indicates the number of nodes required for the current node, $\bar{r}=(\bar{r}_1,\cdots,\bar{r}_n)^T$, $\Sigma=(\mathrm{Cov}_{ij})_{n\times n}$, $M_3=(M_{31},\cdots,M_{3n})^T$ and $M_4=(M_{41},\cdots,M_{4n})^T$ stand for expected return, variance-covariance matrix, third order and fourth order central moments of the n risky assets, respectively. $r^s=(r_1^s,\cdots,r_n^s)^T$ respects the s scenario of return vector of n risky assets. $r=(r^1,\cdots,r^N)\in\mathbb{R}^{n\times N}$, p^s is the probability of the r^s, and $p=(p^1,\cdots,p^N)^T$.

Algorithm 3: Multi-stage scenario tree generation method

S1 Initialization. Choose the parameters r, p, q for the VAR(r)-DCC-MVGARCH (p,q) model, and define the branching structure of the scenario tree, i.e., the number of branches b_k from each non-leaf node k. Make $t=1, k_t=1$.

S2 Simulation. According to the VAR (r)-DCC-MVGARCH (p,q) model, generate a scenario fan by simulating a large number of data paths through node k_t, and estimated by the fan tree conditional statistical properties of $\bar{r},\Sigma,M_3,M_4$ in the model (LP1) need to match the statistics.

S3 Clustering. Using algorithm 2, aggregate the scenario fan into b_{k_t} class $C_1,\cdots,C_{b_{k_t}}$. Select the mean ξ_{t+1}^i of all simulation paths contained in class C_i as the representative scenario of random data, $i=1,\cdots,b_{k_t}$, and these means are the elements of matrix r in model (LP1).

S4 Probability calculation. Substitute the $\bar{r},\Sigma,M_3,M_4$, and matrix r into the model (LP1), and solve the resulting linear programming problem to determine the probabilities of the scenarios.

S5 Termination test. If $k_t<N_t$, make $k_t=k_t+1$, and return to S2. Otherwise, make $t=t+1$, if $t=T$, stop; otherwise let $k_t=1$, and return to S2.

Numerical Results

These sections mainly evaluate the performance of the newly designed scenario tree generation approaches using the generated scenario tree. Also, we illustrate the rationality and advantages of

algorithm 3. For this purpose, four indices in the Shenzhen Stock Market, China, are selected as risky assets in the following experiments. They are the pharmaceutical index (PHA), the financial index (FIN), the petrochemical index (PET), and the metal and non-metal index (MET). The 487 weekly returns with dividend re-invested for each index from July 6, 2001 to March 11, 2011 are used to determine the values of parameters under the model VAR(r)-DCC-MVGARCH (p, q). According to the numerical experiments in the article [5] for VAR(r)-DCC-MVGARCH (p, q) model, here set $r = p = q = 1$.

First of all, based on the VAR (1)-DCC-MVGARCH (1, 1) model, 30,000 paths with three stages are simulated. In this subsection, CVaR_αs with different tail probability levels α are used to describe the tail of the return process based on a scenario fan or scenario tree generated by Algorithm 1 or Algorithm 2. To test the effect of the tail matching of the scenario tree generated by the different algorithms with the original scenario fan, under every chosen branching structure, we perform 10 experiments independently based on Algorithm 1 or Algorithm 2 and record the CVaR_αvalue of return distribution for each risky asset at each stage. Table 1 and Table 2 show that, compared to the original CVaR_αvalues, the average closeness number and the average deviation of the different CVaR_αvalues derived by different algorithms under 10 experiments.

Table 1. Compared to the original CVaR_α values, the average closeness number and the average deviation of the different CVaR_α values derived by different algorithms under 10 experiments.

Branching structure	Algorithm	$\alpha = 0.01$				$\alpha = 0.05$			
		Stage			CVaR_α deviation	Stage			CVaR_α deviation
		1	2	3		1	2	3	
(10,9,8)	Algorithm1	1.9	1.5	0.1	3.5017	2	0.7	0.6	3.5210
	Algorithm 2	2.1	2.5	3.9	3.4991	2	3.3	3.4	3.5197
(15,12,8)	Algorithm 1	1.3	1.1	0	3.4808	1.6	2.7	0.3	3.5107
	Algorithm 2	2.7	2.9	4	3.4743	2.3	1.3	3.7	3.5085
(18,12,10)	Algorithm 1	2.1	1.3	0	3.4708	1.1	1.7	0.3	3.5049
	Algorithm 2	1.9	2.7	4	3.4649	2.9	2.3	3.7	3.5035

From table 1 and table 2, compared with algorithm 1, when the α is 0.01, 0.05 or 0.1, at any given branch structure, the CVaR_α value of the return distribution for each risky asset at each stage on the scenario tree generated by Algorithm 2 is much closer to the corresponding original CVaR_α value compared to the corresponding value on the scenario tree generated by Algorithm 1. In addition, the deviation caused by Algorithm 2 is much smaller than that caused by Algorithm 1. And, more to the leaf node, this advantage is more obvious; Similarly, the smaller the α, the more obvious the advantage. In combination with a large number of numerical experiments, when α equals 0.2 or a much larger value, the above advantages disappear, and the performance of above two methods are equivalent. This is mainly because, the original intention of Algorithm 2 is primarily to ensure that the generated scenario tree as far as possible match the tail scenarios of simulation path, which is also the problem that the risk managers care. Based on the above facts, it is reasonable to conclude that: if a portfolio selection problem is constructed with some tail risk measure as the objective function, such as CVaR_α, WES_α, TNT_α, PCVaR_α as well as two newly introduced two sided risk measures (the generalized Rachev ratio and the Farinelli-Tibiletti ratio), then the optimal portfolios based on the scenario tree generated by Algorithm 2 has stronger competitiveness and more guiding significance for investors than that based on the scenario tree generated by Algorithm 1. Please refer to papers [6-8] for a detailed definition and computation of the above tail risk measures.

Table 2. Compared to the original $CVaR_\alpha$ values, the average closeness number and the average deviation of the different $CVaR_\alpha$ values derived by different algorithms under 10 experiments.

Branching structure	Algorithm	$\alpha = 0.1$				$\alpha = 0.2$			
		Stage			$CVaR_\alpha$ deviation	Stage			$CVaR_\alpha$ deviation
		1	2	3		1	2	3	
(10,9,8)	Algorithm1	1.8	1.3	1	3.5417	1.3	2.6	2.2	3.5729
	Algorithm2	2.2	2.7	3	3.5411	2.7	1.4	1.8	3.5731
(15,12,8)	Algorithm 1	1.8	1.3	1.2	3.5349	1.7	1.5	2.5	3.5709
	Algorithm 2	2.2	2.7	2.8	3.5348	2.3	2.5	1.5	3.5712
(18,12,10)	Algorithm 1	1.8	1.4	1.4	3.5328	1.9	2.1	3.6	3.5690
	Algorithm 2	2.2	2.6	2.6	3.5327	2.1	1.9	0.4	3.5694

Through numerical experiments, we further find the following: Algorithm 3, in addition to being stable, may lead the generated tree to have an appropriate scale and match the first four moments of the original scenario fan well.

Conclusions

In this paper, we first propose a new extremum k-means clustering method to increase accuracy in fitting the tail outcomes of the scenario fan. To match the higher order statistical properties of the random data process, we further design the algorithm 3. The test results indicate the following: the scenario tree generated by algorithm 2 has a much smaller tail deviation than that generated by Algorithm 1; and the scenario tree generated by algorithm 3 has an appropriate scale and match the first four moments of the original scenario fan well.

Acknowledgements

In this paper, the research was sponsored by the National Natural Science Foundation of China (Grant Nos. 71561006, 11461008); Natural Science Foundation of Guangxi Province (Grant No. 2014jjAA10065), Scientific Research Foundation of Higher Education of Guangxi Province (Grant No. KY2015YB050); Research Project of the Humanities and Social Sciences of Guangxi Education Department (Grant No. LX2014046); The Humanities and Social Sciences Research Foundation under Ministry of Education of China (Grant No.13YJA910003); The Key Project of Guangxi Normal University (Grant No. 2014ZD008); Research Project of Guangxi Normal University in 2013 (Scientific Research Foundation for Doctor).

References

[1] Šutienė K, Makackas D, Pranevičius H. Multistage K-Means clustering for scenario tree construction[J]. Informatica, 2010, 21: 123-138.

[2] Gülpınar N, Rustem B, Settergren R. Simulation and optimization approaches to scenario tree generation[J]. J. Econom. Dynam. Control, 2004, 28: 1291-1315.

[3] Høyland K, Wallace S W. Generating scenario trees for multistage decision problems[J]. Manage. Sci. 2001, 47: 295-307.

[4] Ji X D, Zhu S S, Wang S Y, Zhang S Z. A stochastic linear goal programming approach to multistage portfolio management based on scenario generation via linear programming[J]. IIE Transactions, 2005, 37: 957-969.

[5] Xu D B, Chen Z P, Yang L. Scenario tree generation approaches using K-means and LP moment matching methods[J]. Journal of Computational and Applied mathematics, 2012, 236: 4561-4579.

[6] Chen Z P, Yang L, Xu D B, Hu Q H. Tail nonlinearly transformed risk measure and its application[J]. OR Spectrum, 2012, 34: 817-860.

[7] Chen Z P, Yang L. Nonlinearly weighted convex risk measure and its application[J]. Journal of Banking & Finance, 2011, 35: 1777-1793.

[8] Yang L, Chen Z P, Zhang F. Time consistency and time consistent generalized convex multistage risk measures[J].IMA Journal of Management Mathematics,2015. doi: 10.1093/imaman/dpv005.