GS Warsaw Quant Challenge

Goldman Sachs

1 Background

For this challenge imagine you are a risk manager in a financial institution who is responsible for calculating a risk metric described in Section 2. The risk metric will be calculated on losses from a portfolio of loans. The metric is a quantile of the aggregate loan loss distribution of a portfolio. Such a quantile is called Value-at-Risk (VaR) and is used to set up risk tolerances within the company.

The financial performance of each company is modelled by a factor model with an aggregate factor, sector factor and an idiosyncratic factor. Hence, the financial performance of all companies is determined by their exposure to the overall economy/market , the sector the company operates in and there is some idiosyncratic variation in performance independent between companies. Loans only go bad under rather adverse circumstances, therefore we will only incur a loss for a company if its financial performance is below some threshold. Since positions in your portfolio will generally be different in size, variability and have fat tails [1], the loss event indicator will be shifted by company specific constants and amplified by a company specific Student t-distributed random variable with 3 degrees of freedom. Mind that because the Student t-distribution is defined on the entire real number line losses can also be negative (= gains/profits).

Due to this complexity, deriving the individual loss distribution and hence the aggregate portfolio loss distribution and calculating the quantile from that is not possible. Therefore, you will resort to Monte Carlo simulations. In simple terms this comes down to using random number generators to draw values from the underlying Normal and Student-t distributions to generate simulated losses. The general idea is that if you use enough simulations you should have a good approximation of the loss distrbution to compute the quantile (Value-at-Risk). However, since Monte Carlo is stochastic you will always be faced with some estimation uncertainty. This challenge will revolve around managing, quantifying and reducing this estimation uncertainty while also making sure the improvements you make are still computationally feasible. To be more precise, your task is to modify the existing estimator and demonstrate a quantifiable increase of confidence, per unit of computational resource.

While in this challenge the portfolio is fixed keep in mind that in the real world the composition of your portfolio and the dependence between companies change. This will lead to changing loss distributions and hence to a changing Value-at-Risk. Therefore, your method should not be tailored to the composition of a specific portfolio. Take into account that you would need to to calculate the Value-at-Risk of the incurred losses on a weekly basis.

[1] Generally, a distribution is called heavy tailed if its tails decrease slower than the tails of a Normal Distribution. Hence, extreme events are much more likely under these distributions as there is more probability mass in the tails. Fat-tailedness is a special kind of heavy-tailedness where the tails decay like a power law.

2 Problem description

Your portfolio is composed of positions that convey a certain type of risk (possibility of financial loss) due to rare market events considering other companies. The risk metric of interest is a quantile (VaR) at order 0.999 of an aggregated loss. The underlying probabilistic model is specified as follows.

The exposure can be organized into m distinct groups, each associated with a different company g. Companies/groups are further organized into k sectors. For each group g there are two possibilities, each occurring with probabilities p_g, $1 - p_g$, respectively:

1. Rare market event occurs and your company experiences a stochastic financial gain/loss (of a specified distribution depending on the group).
2. Nothing happens.

Market events are not assumed independent across groups and instead the loss indicators (random variables assuming value of 1 if the event occurs and value 0 otherwise) are assumed to feature a particular Gaussian copula with the covariance parameter $\Sigma \in M_{m\times m}$,

$$\Sigma_{ij} = \begin{cases} 1 & \text{if } i = j \\ r_0 & \text{if } i \neq j \text{ and i'th and j'th groups/companies belong to different sectors} \\ r_s & \text{if } i \neq j \text{ and i'th and j'th groups/companies belong to the sector } s,\ s \in \{1, 2, \ldots, k\}, \end{cases}$$

where $r_0, r_1, \ldots, r_k$ are known model parameters.

The specification above can be conveniently expressed in the following way:

1. Random variables $X_0, X_1, \ldots X_k, \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ are independent, identically, standard normal distributed.
2. For company/group g, belonging to the i'th sector, with a known input value t_g $\left(= F^{-1}_{N(0,1)}(p_g)\right)$, we have
$$\mathbb{1}_g := \begin{cases} 1 & \text{if } \sqrt{r_0} \cdot X_0 + \sqrt{r_i - r_0} \cdot X_i + \sqrt{1 - r_i} \cdot \varepsilon_g \leq t_g \\ 0 & \text{otherwise.} \end{cases}$$
3. For company/group g, with a known input values m_g, d_g, the associated loss is
$$L_g := \mathbb{1}_g \cdot (m_g + d_g \cdot \nu_g),$$
where ν_g, identically distributed as Student's t with 3 degrees of freedom, are (jointly with random variables from point 1.) independent.
4. The risk metric $C := F^{-1}_L(0.999)$, where $L := \sum_{g=1}^{m} L_g$.

The standard Monte Carlo approach involves sampling L_g 100,000 times (obtaining realizations $L^1, \ldots, L^{100.000}$), based on provided inputs m_g, d_g, p_g, $g \in \{1, 2, \ldots, m\}$, assignment of sectors to companies/groups and model parameters $r_0, r_1, \ldots, r_k$ and sets $\hat{C} = L^{99.900:100.000}$ (99,900'th largest sampled cumulative loss). Consider only the positive tail (99.9% level) as this corresponds to extreme losses which we would like to avoid and manage.

The approach explained above makes use of two separate methods:

1. The aggregate loss distribution L is simulated through Monte Carlo simulation of the individual L_g. This comes down to taking (in our case) 100.000 random draws from all the random variables which leads to 100.000 simulated losses per L_g. Then, within each simulation the L_g's are summed to obtain one simulated aggregate loss L. Therefore, we end up with 100.000 simulated aggregate losses L. The main idea behind Monte Carlo simulations is that by taking a sufficient amount of random draws one gets close enough to the theoretical loss distribution without having to derive it analytically. This method can be used more broadly to estimate any statistical variable (distribution, expected value, quantiles etc.).

2. The quantile of interest $F_L^{-1}(0.999)$ is estimated by taking the empirical quantile of the simulated distribution. Therefore we pick the 99.900 th largest simulated loss value.

Because for a simulation of sample size n we are using random draws from the underlying distributions it makes sense that between different simulation runs the resulting loss distribution and the corresponding estimated quantile will be different. How much they differ between runs is important because ideally one would like to use as few as possible simulations to obtain an accurate estimate.

This approach has the property that it is asymptotically unbiased [2]. Therefore, if the sample size of the simulations increases eventually the expected value of the estimates will be equal to the true quantile $F_L^{-1}(0.999)$. To put it simply: if we use a large enough sample size then on average we will get the correct estimate. Therefore, this property is important as any bias would imply that on average our estimates would either be too low or high. In mathematical notation asymptotic unbiasedness boils down to: $\lim_{n\to\infty} \mathbb{E}[\hat{\theta}_n] = \theta$ where θ is the true value of the statistical quantity of interest, $\hat{\theta}_n$ is the estimator with sample size n.

A drawback of this method is the variance of the estimates i.e. how much they fluctuate between different simulation runs. Controlling the variance is important for efficient estimation. After all, if the estimates do not fluctuate a lot between runs we do not need as many runs. The variance of this method is rather large. Mathematically, this follows from the fact that the asymptotic variance of the empirical quantile estimator is $\frac{p(1-p)}{f_L^2(F_L^{-1}(p))}$ at a level p [3] . Therefore, as p gets close to zero or one the asymptotic variance blows up if at these extreme p $f_L(F_L^{-1}(p))$ gets close to zero at a faster rate than $p(1-p)$. Intuitively, because we are interested in an extreme quantile (i.e. an extremely unlikely loss) far into the right tail of the loss distribution due to the low probability of losses occuring at that level or beyond we require a very large sample size to obtain estimates that do not fluctuate too much between different simulations.

Hence, the goal of the quant challenge is to design and implement an estimator which improves on the current one. The improvement can be at the cost of some bias as there could be a trade off. In order to help you with quantifying the trade off between bias and variance we provide you with an estimation metric: the mean squared error (MSE). Mathematically, the MSE of an estimator is: $\text{MSE}(\hat{\theta}_n) = \mathbb{E}[(\hat{\theta}_n - \theta)]$. One can show that $\text{MSE}(\hat{\theta}_n) = \text{Bias}^2(\hat{\theta}_n) + \text{Variance}(\hat{\theta}_n)$. Therefore, even if some estimator is (asymptotically) unbiased its variance could be so large that in terms of MSE it can be outperformed by an estimator with a much smaller variance even at the cost of a little bias.

Lastly, keep also in mind that your improved approach should also be computationally efficient. Computational resources are constrained in any real life company setting so the less compute time we need to use on a given task the better. Therefore, just using a simulation with a larger sample size will not be considered an improvement. The yardstick to keep in mind is to make it at most as computationally intensive in terms of computation time as the approach outlined in the previous section.

[2] See: Asymptotic Statistics by A.W. Van der Vaart, Chapter 21 for this result.
[3] See: Asymptotic Statistics by A.W. Van der Vaart, Chapter 21 for this result.

3 Starting point

As a starting point and source of inspiration we provide some topics (in no particular order) for you to look into:

- Variance reduction methods for Monte Carlo simulations: the utility of this is self-explanatory[4].
- A helpful variance reduction method is importance sampling: if you can change Monte Carlo method to sample more from the relevant tail region you could reduce the variance.
- Estimation of (extreme) quantiles. If you can estimate the quantile from our simulations in a better way you could improve performance. This can either be done by using some parameteric distribution model for L (parametric methods) or more flexible functions like kernels and splines to estimate the distribution of L from the simulations (nonparametric methods). The empirical method is fully-data driven but rather crude [5]. So by using a quantile estimation method that imposes more restrictions you could obtain better performance. However, this only holds when said restrictions hold. Otherwise the resulting misspecification error can easily outweigh any previously mentioned benefits.

[4] Useful starting reference: Variance Reduction Techniques in Monte Carlo Methods by Kleijnen, Ridder and Rubinstein. Free download on SSRN.

[5] To get a better idea of this check the paper Fat tails, VaR and subadditivity by Danielsson, Jorgensen, Samorodnitsky, Sarma and de Vries in Journal of Econometrics 2013. Free download on SSRN.

4 Evaluation

You will be evaluated based on the following criteria:

1. Demonstrable improvement of estimator variance over the baseline Monte Carlo approach (eg. estimating and comparing variances of baseline and alternative approaches), which takes into account computation cost (eg. a simple increase of sample size is not considered an improvement).

2. Degree to which the observed variance reduction brings negative impact in terms of increase of bias. Baseline approach is asymptotically unbiased, but this property is not strictly required if the resulting bias of the solution is appropriately quantified.

3. An assessment, made by a panel of judges, which takes into account (among others):

   - Ingenuity of the solution.
   - Demonstrable explainability of the selected approach.
   - Demonstrable understanding of model performance, recognition of the possible pitfalls, resilience of the solution to the portfolio composition changes, etc.

A few teams ranking highest according to the above criteria will be asked to present their solutions live during the finals, followed by a Q&A session.

5 Submission

Submissions should be uploaded to Lockbox under a directory with the team's name and must contain:

1. A solution implementation (+code) used to produce the variance comparison.

2. A max-5-minutes video explaining and showcasing the selected approach.

3. Any additional materials that document the proposed solution and its properties such as slides, a notebook with markdown etc. will be appreciated.

6 Example implementation of the baseline Monte Carlo estimator

```
import numpy as np, pandas as pd, scipy.stats as st

data = pd.read_csv('data/data.csv')
r = np.array([.295, .49, .41, .415, .338, .64, .403, .476])
data['sec_loading'], data['t'] = r[data['sector'].values], st.norm.ppf(data.p)
factors, sample = np.random.normal(0,1, (100_000, len(r)+len(data))), []

for obs in factors:
       m_factor, sec_factor, res_factor = obs[0], obs[:len(r)][data.sector.values],
            obs[len(r):]
       ind = r[0]**.5 * m_factor + (data.sec_loading-r[0])**.5 * sec_factor + (1-
           data.sec_loading)**.5 * res_factor < data.t
       loss = np.zeros((len(data),))
       loss[ind] = data[ind].m + data[ind].d * np.random.standard_t(3, size=sum(ind
           ))
       sample.append(sum(loss))
VaR = sorted(-sample)[100]
```