Loomis Project

Xinhui Gu

1 Target & Data

The credit rating transition matrix follows Moody's rating standard so I have assumed the transition matrix would follow Moody's methodologies. A quick check on the dataset:

- **TRANSITION MATRIX** Given the probability for each issuer, I added one more row to identify the defaulted bonds. "Default" is an absorbing state and hence it has only diagonal probability of 1.

- **STATE PROBABILITY** For each issuer grade, the transition probability should sum up to 1. But from the dataset provided, they are not summing up to 1. From the Moody's credit transition model document, there is an additional status "WR"(Withdrawn). I calculated the difference and it follows similar pattern as listed in Moody's methodology document (Probability of "WR" is increasing as the credit rating goes down). I have ignored the possibility goes to "WR" since it could be also considered as an absorbing state similar to "Default".

From the credit rating scale, investment grade includes Baa3 and above. Our target is to build a portfolio which replicates the investment grade return and minimize the risk(standard deviation). This methodology applies to other ratings.

2 Assumption

- **HOMOGENEOUS ISSUER WITH SAME RATING** We are assuming all issuers that falls within the same credit rating class follows same probability density. However in practice there is a momentum effect. A firm that has been recently downgraded are more likely to be downgraded again than other firms in the same rating category. So we should further separate the issuers within the same ratings further to capture the probability differences.

- **INVESTMENT HORIZON** Estimation is only for next 3 years horizon and we are assuming the transition matrix holds constant for the next 3 years.

3 Methodology

We are trying to find the efficient frontier given the credit transition matrix and yields for each rating.
We will need to calculate the expected return for each issuer using formula as below.

$$\mathbf{E}_R = \mathbf{P_i} \cdot \mathbf{y}$$

Where $\mathbf{P_i}$ is probability distribution for issuer with $\mathbf{i}$-th rating and $\mathbf{y}$ is the given yields.
Next step is to calculate the var-covariance matrix for each issuer. The var-covariance matrix is symmetric and for issuer $\mathbf{i}$ and $\mathbf{j}$, applying the covariance formula for $\mathbf{r_i}$ and $\mathbf{r_j}$

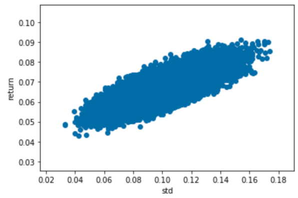
$$\mathbf{Cov}(r_i, r_j) = \mathbf{E}(r_i \cdot r_j) - \mathbf{E}(r_i) \cdot \mathbf{E}(r_j)$$

This applies for all $\mathbf{i}$ and $\mathbf{j}$ in the covariance matrix.
Assuming the portfolio weight is $\mathbf{w}$, expected return $\mathbf{R_w}$ and portfolio standard deviation are calculated as follows:

$$\mathbf{E}_w = \mathbf{w} \cdot \mathbf{E}_R$$

$$\sigma_w = \sqrt{\mathbf{w} \cdot \mathbf{Cov} \cdot \mathbf{w}^T}$$

We are setting the expected return $\mathbf{E}_w$ as BAA3 and out target portfolio is the one with minimum risk. Depending on the investment mandate, we could add constraints to the portfolio optimization logic. Using $\mathbf{w}$ as input, the function iterates to calculate $\mathbf{E}_w$ and σ_w using the formula above and searching for the optimized portfolio $\mathbf{E}_{opt}$ that maximize the Sharp Ratio: $SR = \frac{\mathbf{E}_{opt}}{\sigma_{opt}} = \frac{\mathbf{w}_{opt} \cdot \mathbf{E}_R}{\sqrt{\mathbf{w}_{opt} \cdot \mathbf{Cov} \cdot \mathbf{w}_{opt}^T}}$. I ran a simulation of 1000 random portfolios and plot the investment universe.



4 Results

Apart from the target return $\mathbf{E}_w = 5.98\%$, the only constrain we have so far is on portfolio weights: $\sum_i \mathbf{w}_i = 1$. This will serve is the benchmark portfolio. I will show 2 results below, one without constraint and one with constraint.

No Constrains Imposed :

In this case, there are no bounds for each weights and long/short positions are both available. Results as below:

```
message: 'Optimization terminated successfully.'
   nfev: 479
    nit: 20
 status: 0
success: True
      x: array([0.05404336, 0.07846041, 0.00347803, 0.055031   , 0.02406774,
0.07916171, 0.03280903, 0.04816922, 0.03879453, 0.0747573 ,
0.0313994 , 0.06267437, 0.03168558, 0.04099527, 0.07410369,
0.00318682, 0.02186923, 0.06860522, 0.01109054, 0.01663804,
0.06790558, 0.08107394])
```

With Constrains :

Suppose a hypothetical investment mandate from a client: for each issuer, the maximum weight could not exceed 8% of the target portfolio and no short position is allowed. We will impose this constrain to the optimization function and have the following results

```
message: 'Optimization terminated successfully.'
   nfev: 25
    nit: 1
   njev: 1
 status: 0
success: True
      x: array([0.0540945 , 0.07851155, 0.00352917, 0.05508214, 0.02411888,
       0.07921285, 0.03286017, 0.04822036, 0.03884567, 0.07480844,
       0.03145054, 0.06272551, 0.03173672, 0.04104641, 0.07415483,
       0.00323796, 0.02192037, 0.06865636, 0.01114168, 0.01668918,
       0.06795672, 0.08      ])
```

The constrains could be customized and also multiple constraints are also supported by the optimization function. Code are attached.