Optimal Rebalancing Strategy Using Dynamic Programming for Institutional Portfolios

Motivation & Model Description

Financial institutions generally employ fixed methods to rebalance portfolios, such as calendar basis or tolerance band triggers. In the proposed setup, the investor has a risk profile derived from the log wealth utility function:

$$f_1(x) = \log_{10}(1 + x)$$

,where x represents the portfolio return in each period. As per literature[1], expected utility is modeled as follows:

$$U(\mu, \sigma) = \log_{10}(1 + \mu) - \frac{\sigma^2}{2(1 + \mu)^2}$$

,where μ is the mean return and σ is the standard deviation of the portfolio up to the point in time. This utility function identifies the investor as a risk-averse individual; therefore the optimal portfolio can be calculated using Mean Variance Optimization:

$$min_w(w'\Lambda\, w) \quad s.t. \;\; w'\mu = \mu_P, \sum_i w_i = 1\,, w \geq 0$$

,where w are the possible portfolio weights, Λ is the covariance matrix of the available assets and μ is the vector of expected asset returns. Solving the above minimization problem for a Two-Asset-Model yields the weight that minimizes variance - w^* (Figure 1).

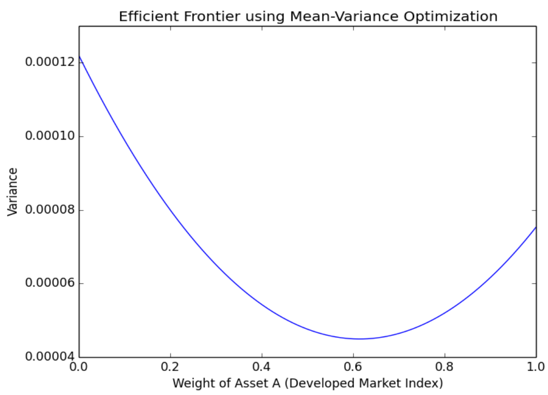

Figure 1 – Mean Variance Optimization

Assuming a portfolio employs this optimal weight in period 0, the subsequent-in-time weights will drift due to the different returns in the assets. Standard strategies only take into account specific bounds for weight movements; once this threshold is reached, the portfolio is rebalanced to the optimal portfolio (the weight that minimizes variance). The shortfall of these decisions is they do not account for costs related to the rebalancing. The strategy used in the literature[1], and replicated in the current application, makes rebalancing decisions dynamically, that is on the basis of transaction costs, sub-optimality costs and future costs in each period, for the whole sample.

Transaction costs of rebalancing can be modeled as:

$$TC = C_A|w_A - w_A^*| + C_B|w_B - w_B^*|$$

,where C_A and C_B are the costs (in *bps*) of the two assets in the portfolio. Furthermore, not rebalancing leads to the portfolio deviating from the optimal portfolio, resulting in a lower utility for the investor. This leads to an increase in sub-optimality costs (i.e. the utility shortfall in comparison to having a perfectly balanced optimal portfolio – measured in *bps*). These are modeled using Certainty Equivalent Costs:

$$\epsilon(w) = r_{CE}(w^*) - r_{CE}(w)$$

,where $r_{CE}(w) = exp[U_1(\mu^T w, w^T \Lambda w)] - 1$. Finally, the rebalancing decision takes into account the future costs of rebalancing $J_{t+1}(w_{t+1})$; this results in a dynamic model, characterized by the discrete-time Bellman Equation:

$$J_t^*(w_t) = \; min_{u_t} E\left[TC_{u_t} + \epsilon_{u_t} + J_{t+1}(w_{t+1})\right]$$

[1] Optimal Rebalancing Strategy Using Dynamic Programming for Institutional Portfolios - Walter Sun, Ayres Fan, Li-Wei Chen, Tom Schouwenaars, Marius A. Albota – MIT Working Paper

Procedure

For the implementation of this model the Python programming language was employed, and daily historical data on the Russell 3000 Index and the MSCI EAFE Canada Index were acquired through Bloomberg for the period 01/01/2013 – 14/04/2014. Employing Mean Variance Optimization the optimal portfolio was calculated to be 61.45% for the MSCI EAFE Canada Index and 38.55% for the Russel 3000 (Figure 1). This was conducted by calculating the weight that minimizes the variance:

```
A['Variance'][i] = wA * (wA * Covariance_Matrix[0,0] + \
             (1 - wA) * Covariance_Matrix[0,1]) + (1 - wA) * (wA * Covariance_Matrix[1,0] + (1 - wA) * Covariance_Matrix[1,1])
```

As previously mentioned, the decision to rebalance or not today depends on the transaction costs and the sub-optimality costs of today, as well as the costs of rebalancing tomorrow. These were modeled as follows:

```
# Calculate the Certenty Equivalent Costs in the particular period (i.e. cost of not rebalancing to Optimal Portfolio)
Cost_Min['CEC'][line2] = (math.exp(Cost_Min['Expected_Utility_Optimal'][line2]) - math.exp(Cost_Min['Expected_Utility_Current'][line2]))*initial_amount_invested

# Calculate the Transaction Costs to be incurred if rebalancing is to take place
Cost_Min['TC'][line2] =  (CA*math.fabs(Optimal_WeightA-(Cost_Min['WeightA'][line2])) + CB*math.fabs((1-Optimal_WeightA)-(1-(Cost_Min['WeightA'][line2]))))

# Calculate Total Costs to be incurred if rebalancing is to take place
Cost_Min['Costs'][line2] = Cost_Min['CEC'][line2] + Cost_Min['TC'][line2]
```

The above costs were calculated for a range of weights (0.0 to 1.0 with step of 0.0005) for all the points in the time series (Figure 2), and the weight that minimizes costs was selected (Figure 3).

t	Returns_A	Returns_B	...
1	0	0	...
2	-0.0097	0.0119	...
3	0.0051	-0.0048	...
...	...	...	...
325	0.0052	-0.0029	...

Weight	CEC	TC	...
0			...
0.0005			...
0.001			...
0.0015			...
...	...	...	...
0.999			...
0.9995			...
1			...

Figure 3 – Calculation of Weight Minimizing Costs in Each Period

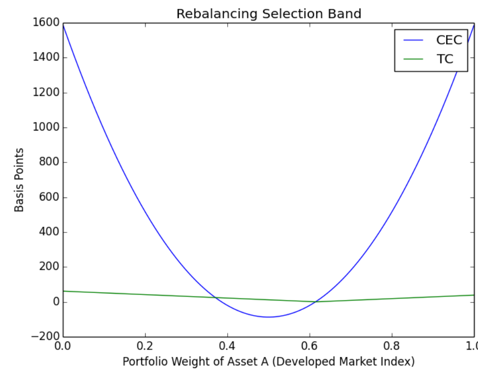

Figure 2 – Determine Weight Minimizing Costs & Rebalancing Thresholds

Furthermore, the thresholds for rebalancing were calculated; these are the points of intersection of sub-optimality (CEC) and transaction (TC) costs (Figure 3). The result of the iteration process yielded ideal, go-to weights that minimize costs in every period, if the decision to rebalance is made. The decision is based on whether the current, non-optimal weights cross any of the two respective thresholds; if this is the case, the portfolio of the current period is rebalanced to the weight minimizing costs. Since this change affects the weights in the subsequent periods, every time a rebalancing is conducted the future periods are adjusted accordingly, along with the future costs, $J_{t+1}(w_{t+1})$.

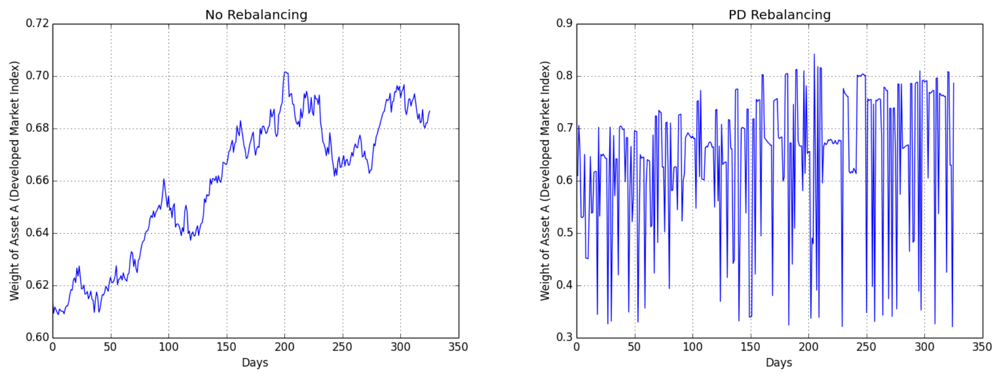

Figure 4 – Weight Variation with No Rebalancing and Dynamic Programming Rebalancing

The model developed was executed in three modes; No Rebalancing, Optimal Rebalancing (constantly rebalancing to mean variance optimization weight), and DP Rebalancing, the resulting costs incurred being \$ 33’242’104.3 (3.324%), \$ 23’589’137.1 (2.359%) and \$ 19’519’500.0 (1.952%). The results confirm that the Dynamic Programming solution will always outperform any fixed rebalancing strategy, since it ensures the investor incurs the minimum possible cost when rebalancing.