MATH584 - Math for Algo Trading Homework 2

Niti Wattanasirichaigoon
A20406934

File Descriptions

File Name	Description
a_PnL.csv	1a) PnL process over 25 windows (2500 days)
a_strat.csv	1a) Annualized mean returns, variance, and Sharpe ratio
b_PnL.csv	1b) PnL process over 3 windows (300 days)
b_strat.csv	1b) Annualized mean returns, variance, and Sharpe ratio
c_PnL.csv	1c) PnL process over 15 days
c_strat.csv	1c) Annualized mean returns, variance, and Sharpe ratio

1. Construct a dynamic utility-optimizing trading strategy (1 risky, 1 riskless).

The exercise has been performed only on one asset '*Apple*' as the DPP process is time consuming and does not affect the general analysis.

a. Here, I solved Merton's problem explicitly for α in terms of μ and σ.

The Value function takes the form of

$$V(t,x) = x^{\gamma} c(t),$$

Which we can solve for the solution that α that maximizes

$$c(t) = \max_{\alpha} EV[X_0 \max_{\alpha} E[2(1+\alpha(\mu+\epsilon)+(1-\alpha)r)^{1/2}]]$$

Where α is the weight of the risky return, and r is the rate of the riskless return

Which is equivalent to solving the term

$$\max_{\alpha} E[2(1+\alpha(\mu+\epsilon)+(1-\alpha)r)^{\frac{1}{2}}]$$

$$= (1+\alpha(\mu+\sigma)+(1-\alpha)r)^{\frac{1}{2}} + (1+\alpha(\mu+\sigma)+(1-\alpha)r)^{\frac{1}{2}}$$

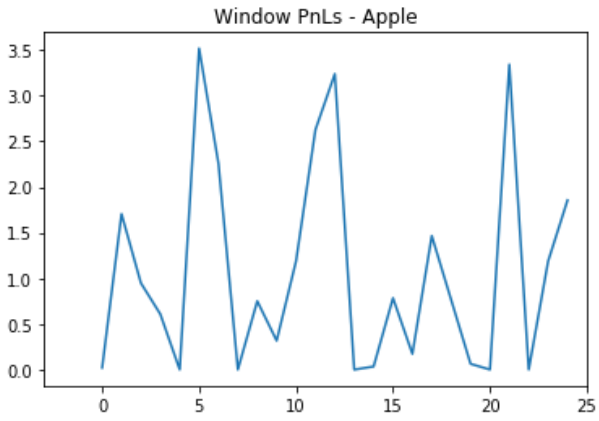
$$= (1+r+\alpha(\mu-r+\sigma))^{\frac{1}{2}} + (1+r+\alpha(\mu-r-\sigma))^{\frac{1}{2}}$$

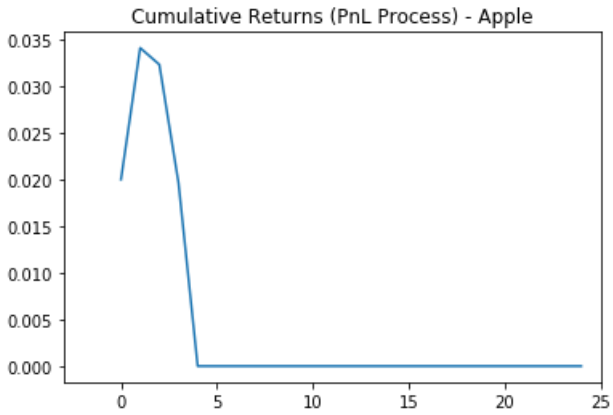
Then I took the derivative with respect to α, and set it to 0 to solve for the α that maximizes.

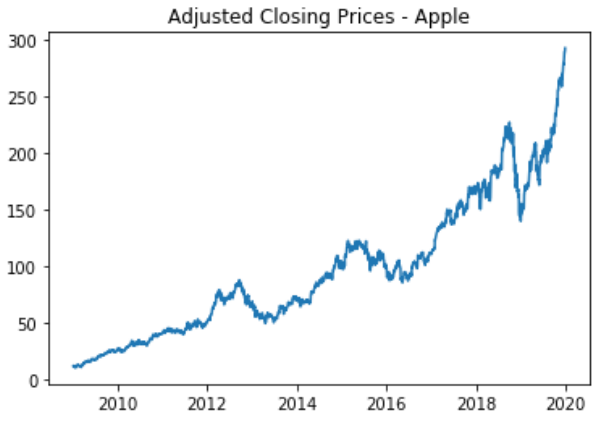
$$0 = \frac{1}{2}(\mu-r+\sigma)(1+r+\alpha(\mu-r+\sigma))^{-\frac{1}{2}} + \frac{1}{2}(\mu-r+\sigma)(1+r+\alpha(\mu-r-\sigma))^{-\frac{1}{2}}$$

$$\alpha = \frac{2(r+1)(r-\mu)}{(\mu-r)^2-\sigma^2}$$

Merton's problem yields us an optimal strategy over a given period of time. I retrieved my parameters (μ, σ) from each calibration window to compute the best α from the formula above for the succeeding 100 days. Using the alpha stars to calculate the cumulative returns for each window, we get the following results.







The PnL for each window is the product between all the days within that window. I plotted the PnL process by windows because Merton's problem solves for an optimal alpha for a given interval (if done by DPP, I would have alphas for each day). Nonetheless we can see that the window returns spike around the periods corresponding to where the Apple stock reaches its peaks and valleys, suggesting how the modified returns will change during these periods. The cumulative returns (entire PnL process), on the other hand, only increased for the first couple of windows before crashing to a number near zero. Any following increase in returns were not enough to bring the returns back to an observable value.

Applying the alpha stars on the original 2500 returns, I achieved a new set of returns that represents our strategy. The (annualized) mean and variances of the new returns are **2.0897** and **13.8075** respectively. The corresponding Sharpe Ratio is **0.5597**, which appears to be low but arguably reasonable since we can see from the PnL process that this strategy is not a good one.

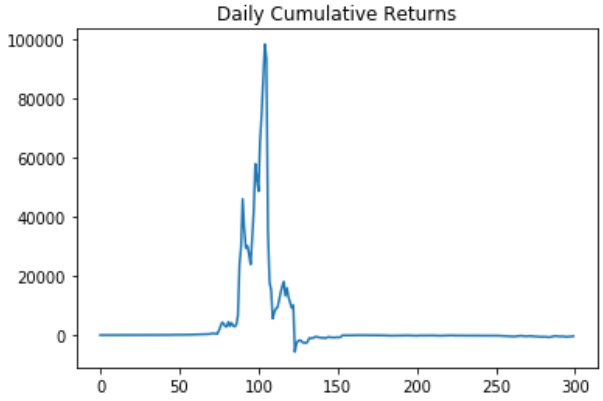
b. In this section, we are required to use DPP because Merton's problem cannot solve for α explicitly. I gathered parameters from the calibration window to setup the DPP process. Of which the expanded value function is:

$$V(t,f) = \frac{1}{4}(1+\alpha(a+cf+\sigma)+(1-\alpha)r)^{\gamma}$$
$$(V\left(t+1, fe^{-1}+(\hat{\mu}+\hat{\sigma})(a+cf+\sigma)\right)+V\left(t+1, fe^{-1}+(\hat{\mu}-\hat{\sigma})(a+cf+\sigma)\right)$$
$$+\frac{1}{4}(1+\alpha(a+cf-\sigma)+(1-\alpha)r)^{\gamma}$$
$$(V\left(t+1, fe^{-1}+(\hat{\mu}+\hat{\sigma})(a+cf-\sigma)\right)+V\left(t+1, fe^{-1}+(\hat{\mu}-\hat{\sigma})(a+cf-\sigma)\right)$$

Where the function $V(t+1,f)$ will be interpolated after obtaining optimal values for each f on the grid. I then interpolate alpha as a function of f to use for predicting the best alpha for each realized value of f (factor value).

During implementations, I increased the initial state X_0 by 100 times so that the changes afterwards are large enough. Without doing this, the DPP process would still be able to run until t=0, but the change value function in too small for *scipy* to minimize. Also, due to time and processing constraints, the calibration window has been moved for 6 times (600 days).

Plotting the PnL process of cumulative returns across 300 days, we can see that the returns exploded towards a very large number around day 100. It can be speculated that the factor values in the calibration windows around this interval gave rise to subsequent large values of alphas, causing the returns to increase abruptly. Similar to part a), it later on converges to zero.

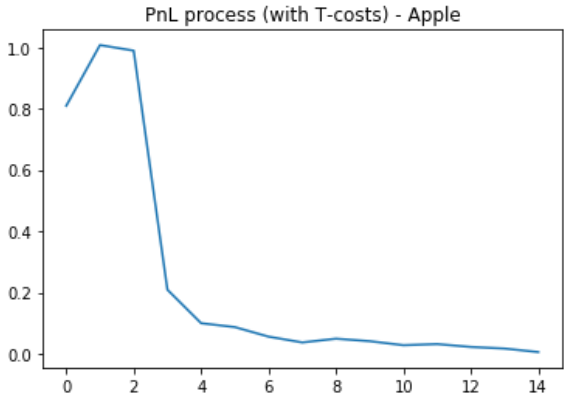


The (annualized) mean and variances of the new returns according to our strategy are **13.5037** and **26.5375** respectively. One would expect a higher mean than part a) with such a more sophisticated strategy. The Sharpe ratio for this strategy is **2.6194**, which better than the strategy in part a).

c. In this last section we carry a similar strategy to part a) except now there is also a transaction cost proportional to the change in weights. We solve this by applying DPP with the expanded value function with the riskless asset:

$$V(t,a) = \frac{1}{2}(1+\alpha(\mu+\sigma)+(1-\alpha)r-\lambda|\alpha-a|)^{\gamma}V\left(t+1,\frac{\alpha(1+\mu+\sigma)}{1+\alpha(\mu+\sigma)+(1-\alpha)r}\right)$$
$$+\frac{1}{2}(1+\alpha(\mu-\sigma)+(1-\alpha)r-\lambda|\alpha-a|)^{\gamma}V\left(t+1,\frac{\alpha(1+\mu-\sigma)}{1+\alpha(\mu-\sigma)+(1-\alpha)r}\right)$$

During implementation, I used the *scipy.optimize.minimize* function to optimize my set of weights$\{\alpha_1, \alpha_0\}$. After computing all the alpha stars for each a (prior weight), I interpolated α as a function of a and saved it into a list. During back-testing, the corresponding α function of a will be called to compute the next weight. It is important to note that optimization process takes a very long time to compute, and I ended up computing the DPP for 15 days.



Plotting the PnL process for 15-day DPP gives us the following graph. We can see that the returns increased after the first few days then dropped drastically and gradually decreases its way towards zero.

With only applying our strategy to such a small sample (15 days), it is not recommended to look at the mean and variance of returns. The (annualized) mean and variances are -55.808 and 24.720 respectively. The mean is negative obviously because the values in this interval are negative. Thus, the Sharpe ratio is -11.2267 (Based on these 15 days, this is a bad strategy). If we have more data (300+ days) with most returns being positive, then this could be a viable strategy.

Certainly, many improvements can be made to this last part. I learned that there is a function, *scipy.optimize.minimize_scalar* which can carry optimization for one variable which should be more efficient than my method of feeding 2 values into the objective function. Although I tried to use this *scalar* version of the function, I encountered problems where the function could not maximize, which I assume might have to deal with the concavity.