IE501 Project Report
Conditional Downturn based Portfolio Optimisation

Group 6
IIT Bombay

November 25, 2024

Teammate Name	Roll Number
Praveen	22b3931
Shaheem Basheer	22b0446
Harsh Raj	210100068
Poras Sanjay Kolgane	17D170014

Contents

1 Problem Statement

Optimising Portfolio return using **Dips** in Return function to measure the risk, especially over the long term. We contrast this with **Conditional Value at Risk (CVaR)** [1]. , which accounts only for net loss

2 Problem Description

	Using net loss for Risk	**Using Dips in return**
Time Horizon	Short-term	Long-term
Focus	Potential losses beyond a given percentile in a short period	Maximum potential loss during a downturn over a longer period
Risk Interpretation	Assesses the potential for extreme losses in a short term	Provides insights into the drawdown and potential recovery over a longer term

Table 1: Comparison of Basic and Dips based modelling approach

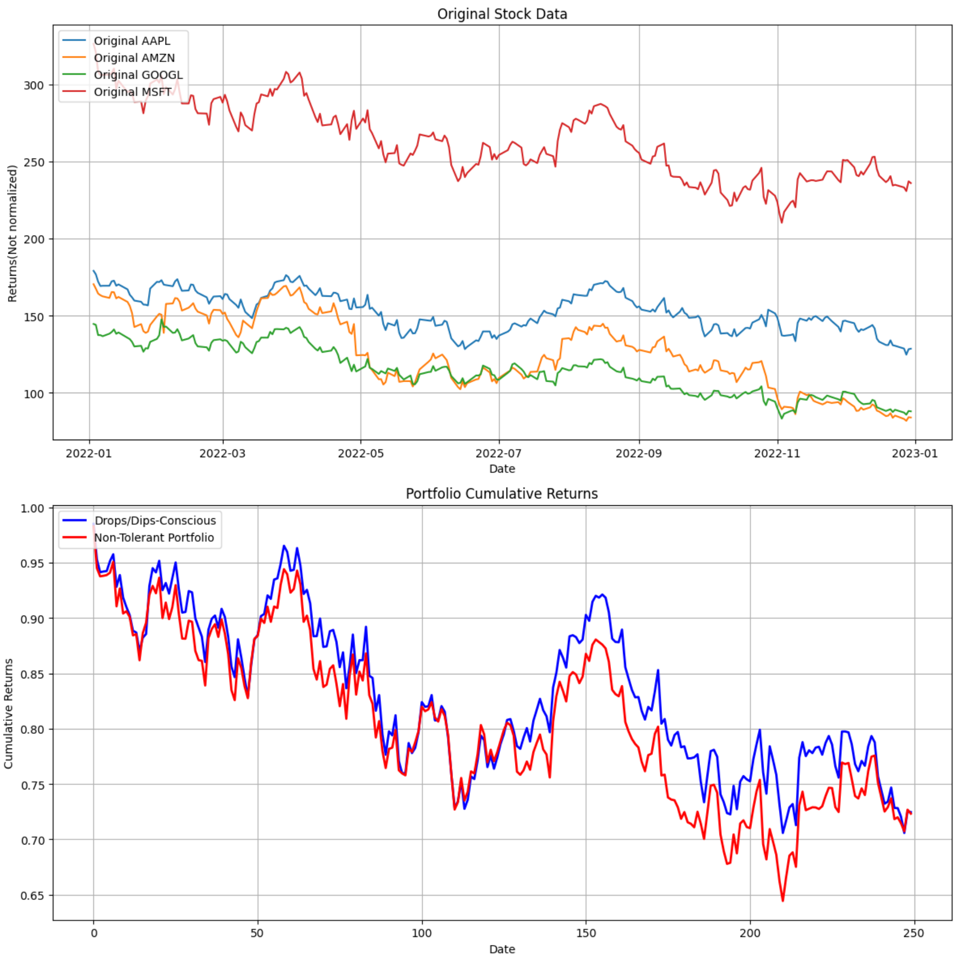


Figure 1: Normal vs Dips based modelling

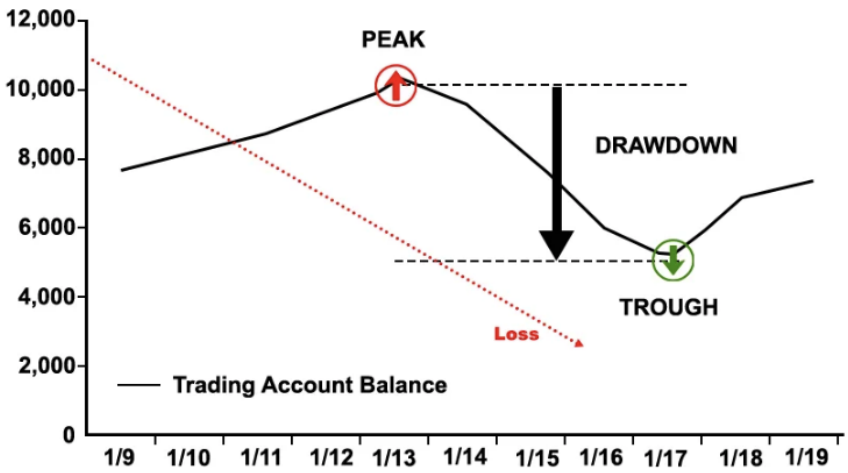
Stock	AAPL	AMZN	GOOGL	MSFT
Non Dip Tolerant Model	0.015	5.55e-17	0.0000	0.9
Drops/Dips-Conscious Model	0.498363	0.0000	0.0000	0.501637

Table 2: Comparison of Weights for Drops/Dips-Conscious and Non-Tolerant Models

Example Consider a cryptocurrency portfolio that peaks at $10,000, then falls to $5,000, and eventually recovers to $7,000. In this case, the loss and drawdown can be calculated as follows:

$$\text{Loss} = 10,000 - 7,000 = 3,000$$

$$\text{Drawdown} = 10,000 - 5,000 = 5,000$$



3 Setup for Formulation

3.1 Definitions and Key Concepts

Term	**Definition**
$w(x,t)$	Uncompounded portfolio return at time t
$x = (x_1, x_2, \ldots, x_m)$	Portfolio vector: Weights of m instruments in the portfolio
$D(x,t) = \max\{w(x,t)\} - w(x,t)$	Dip function
N	Number of time sub-periods in the time interval $[0, T]$
$\alpha \in [0,1]$	Confidence level

Table 3: Definitions of terms

3.2 Three ways to estimate risk

We define the key risk measures used for portfolio optimization as follows:

3.2.1 Average Dip

$$A(x) = \frac{1}{N}\sum_{t=1}^{N} D(x,t).$$

The average drawdown is the mean of all drawdowns over the time interval.

3.2.2 Maximum Dip

$$M(x) = \max\{D(x,t) : t \in [0,T]\}.$$

The maximum drawdown is the largest observed drawdown in the portfolio over the given time horizon.

3.2.3 Conditioned Dip function

$$\Delta_\alpha(x) = \frac{1}{(1-\alpha)N}\sum_{t\in Q} D(x,t)$$

where:

$$Q = \{t \in [0,T] : D(x,t) > L\}.$$

Here, L is the threshold for drawdowns at the confidence level α. When $\alpha \to 1$, This function approximates the maximum drawdown: $\Delta_1(x) = M(x)$. When $\alpha = 0$, it coincides with the average drawdown: $\Delta_0(x) = A(x)$.

3.3 Portfolio Returns and Historical Data

Portfolio returns and their relationship to historical data are defined as:

- $y_j(t) = \sum_{k=1}^{t} r_j(k)$: Cumulative return of the j-th instrument up to time t
- $y(t) = (y_1(t), y_2(t), \ldots, y_m(t))$: Return vector at time t
- $w(x,t) = \sum_{i=1}^{m} y_i(t)x_i$: Cumulative portfolio return at time t.

3.4 Constraints

The optimization problem is subject to the following constraints:

$$\frac{1}{(1-\alpha)N}\sum_{k=1}^{N}\left[\max_{w(x,j)}\{y_j\cdot x\} - w(x,k) - L\right]^{+} + L \le \nu C \tag{1}$$

where:

- L: Variable defining the threshold for Dips to consider
- ν: Allowable Fraction of capital for loss
- C: Total capital invested

- $[g]^+ = \max\{g, 0\}$: ReLU funtion

Additionally, the allocation weights x are bounded:

$$x_{\min} \le x_i \le x_{\max}, \quad \sum_{i=1}^{m} x_i = 1.$$

3.5 Optimization Formulation

The optimization problem is formulated as:

$$\max_x w(x, N) = \max_x \sum_{i=1}^{m} y_i(N) x_i$$

Subject to the constraint (1) and the allocation bounds, this is reformulated as a linear program (LP):

$$\max_{x, L, u_k, z_k} \sum_{i=1}^{m} y_i(N) x_i$$

subject to:

$$\begin{aligned}
&\frac{1}{(1-\alpha)N} \sum_{k=1}^{N} z_k + L \le \nu C, \\
&z_k \ge u_k - y_k \cdot x - L, \quad z_k \ge 0, \quad 1 \le k \le N, \\
&u_k \ge y_k \cdot x, \quad u_k \ge u_{k-1}, \quad u_0 = 0, \quad 1 \le k \le N, \\
&x_{\min} \le x_i \le x_{\max}, \quad 1 \le i \le m.
\end{aligned}$$

4 Code Implementation

```
# Decision variables
x = cp.Variable(m)
z = cp.Variable(N, nonneg=True)
u = cp.Variable(N)
L = cp.Variable()

# Constraints
constraints = []

# Constraint: L + (1 / (1-alpha)N) * sum(z) <= v * C
constraints.append(L + (1 / ((1 - alpha) * N)) * cp.sum(z) <= v * C)

# Constraints for z and u
for k in range(N):
    constraints.append(z[k] >= u[k] - cp.sum(cp.multiply(y[k], x)) - L)
    constraints.append(z[k] >= 0)
    constraints.append(u[k] >= cp.sum(cp.multiply(y[k], x)))
    if k > 0:
        constraints.append(u[k] >= u[k - 1])

constraints.append(u[0] == 0)
```

```python

# New constraint: Sum of elements of x must equal 1
constraints.append(cp.sum(x) == 1)

# Constraints for bounds on portfolio weights
for i in range(m):
    constraints.append(x[i] >= x_min[i])
    constraints.append(x[i] <= x_max[i])

# Objective function: Maximize portfolio return
objective = cp.Maximize(cp.sum(cp.multiply(y[-1], x)))

# Problem definition and solving
problem = cp.Problem(objective, constraints)
problem.solve()

# Results
optimal_weights = x.value
```

4.1 Link to the Notebook

https://drive.google.com/file/d/1xOc8Pw0Y4fb62EW77hjxpEZobpbWjhaT/view?usp=sharing

5 Results on 10 Stocks data for Last 10 years

Instrument	Weight
AAPL	0.0468
MSFT	0.0006
GOOGL	0.0006
AMZN	0.0005
TSLA	0.0005
META	0.0013
NVDA	0.4990
JPM	0.4490
UNH	0.0008
V	0.0008

Table 4: Optimal Weights for Each Instrument

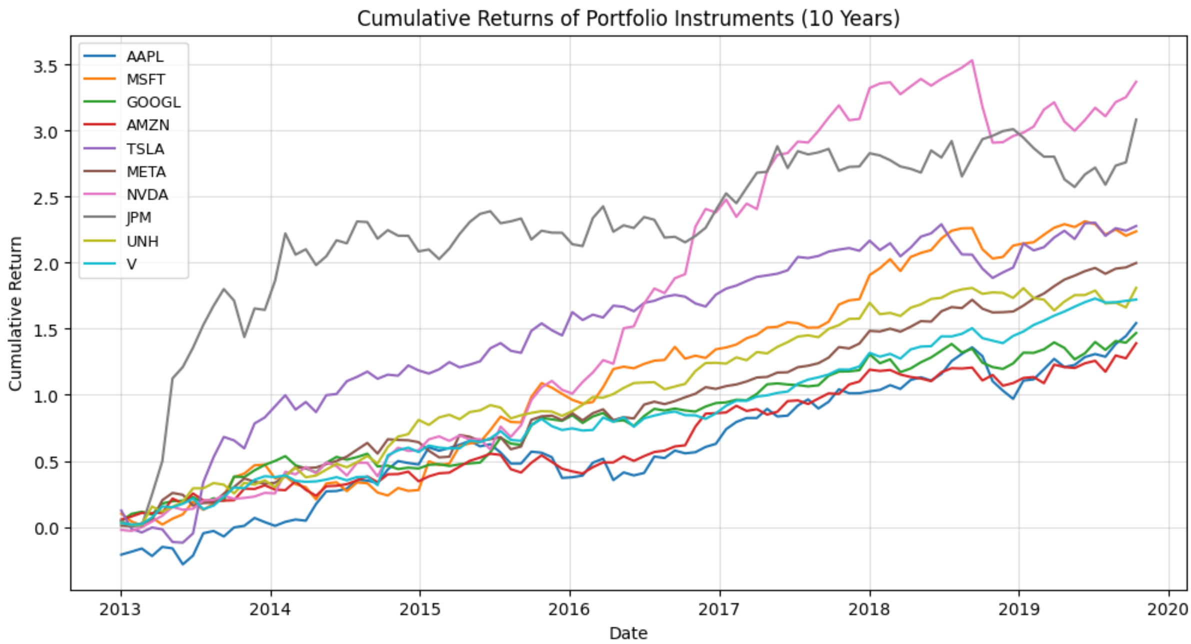


Figure 2: Returns of 10 Instruments

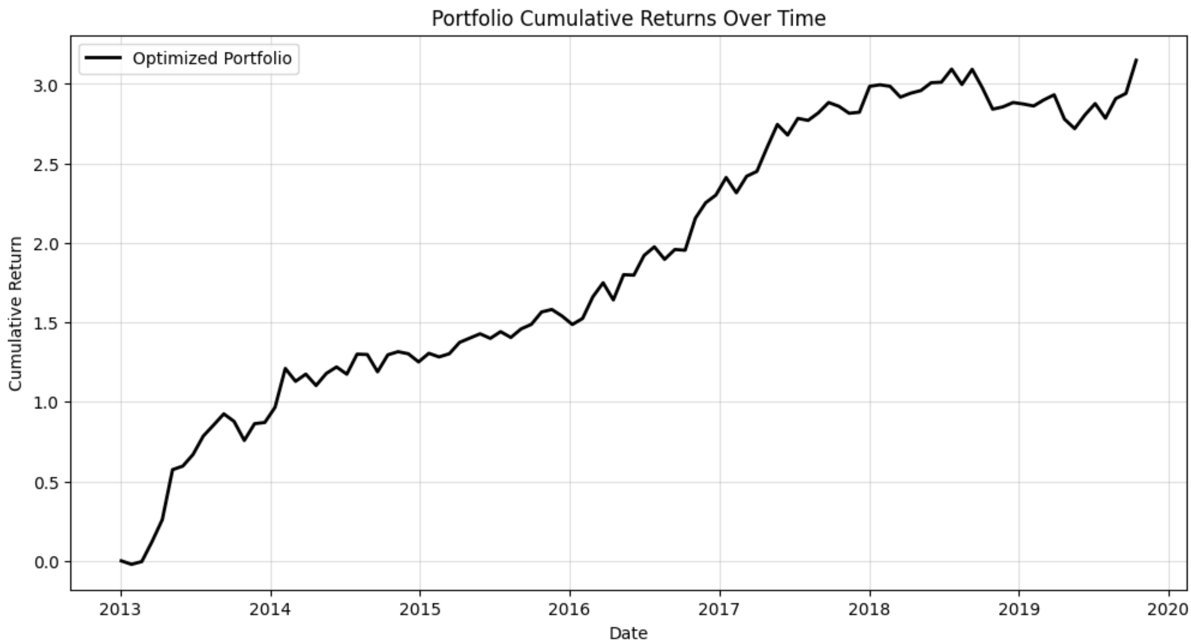


Figure 3: Cumulative return using Dip based formulation

References

[1] Jakob Kisiala. Conditional value-at-risk: Theory and applications, 2015.