Risk-constrained dynamic asset allocation via stochastic dual dynamic programming

Prof. Davi Valladão – Industrial Eng. – PUC-Rio
Thuener Silva – Industrial Eng. – PUC-Rio
Prof. Marcus Poggi– Department of Informatics – PUC-Rio
Objective and contributions

Develop a realistic and computationally tractable stochastic dynamic asset allocation model considering time consistent and intuitive risk constraints, time dependence and transaction cost.

Main Contributions:

• Time consistent risk-constrained stochastic dynamic asset allocation model
  – Realistic: transaction costs and time dependence
  – Risk aversion: intuitive user defined loss limit
  – Computationally tractable: SDDP with Markovian policy
No transactional cost and time independence

\[ Q_t(W_t) = \max_{a_t, c_t, d_t \geq 0} \phi \left[ Q_{t+1} \left( c_t + \sum_{i \in \mathcal{A}} (1 + r_{i,t+1}) a_{i,t} \right) \right] \]

s.t.

\[ c_t + \sum_{i \in \mathcal{A}} a_{i,t} = W_t \]

Where, \( Q_T(W_T) = W_T \)
Time consistent (recursive) model

• Usually the risk measure is the convex combination of the expect return and the CVaR

\[ \phi(W_t) = (1 - \lambda)E[W_t] + \lambda CVaR_\alpha(W_t) \]

• Economic interpretation: certain equivalent
  – Rudloff, Street, Valladão (2014)

• Problem: How should we define \( \lambda \)?
Risk constrained model

\[ V_t(W_t) = \max_{a_t, c_t \geq 0} \mathbb{E} \left[ V_{t+1} \left( c_t + \sum_{i \in \mathcal{A}} (1 + r_{i,t+1}) a_{i,t} \right) \right] \]

\[ s.t. \quad \rho_t \left[ \sum_{i \in \mathcal{A}} r_{i,t+1} a_{i,t} \right] \leq \gamma W_t \]

\[ c_t + \sum_{i \in \mathcal{A}} a_{i,t} = W_t \]

• Intuitive risk averse parameter \( \gamma \)
• Relative complete recourse for \( \gamma \geq 0 \).
  – One can always allocate in cash
• Positively homogeneous

If \( W_t \geq 0 \), \( V_t(W_t) = W_t \cdot V_t(1) \)
Myopic Solution

\[ V_t(W_t) = \max_{a_t,c_t \geq 0} \mathbb{E} \left[ V_{t+1} \left( c_t + \sum_{i \in \mathcal{A}} (1 + r_{i,t+1}) a_{i,t} \right) \right] \]

s.t.
\[ \rho_t \left[ \sum_{i \in \mathcal{A}} r_{i,t+1} a_{i,t} \right] \leq \gamma W_t \]
\[ c_t + \sum_{i \in \mathcal{A}} a_{i,t} = W_t \]
Myopic Solution

\[ V_t(W_t) = \max_{a_t, c_t \geq 0} \begin{cases} & c_t + \sum_{i \in \mathcal{A}} (1 + r_{i,t+1}) a_{i,t} \\ \text{s.t.} & \rho_t \left[ \sum_{i \in \mathcal{A}} r_{i,t+1} a_{i,t} \right] \leq \gamma W_t \\ & c_t + \sum_{i \in \mathcal{A}} a_{i,t} = W_t \end{cases} \]
No transaction costs and time dependence

- Assuming a factor $z_t$ affecting asset returns

$$z_{t+1} = f_z(z_t) \quad \text{e} \quad r_{t+1} = f_r(z_{t+1})$$

- One can solve the following problem with Stochastic Dynamic Programming (SDP)

$$Q_t(W_t, z_t) = \max_{a_t, c_t \geq 0} \mathbb{E}[Q_{t+1}(W_{t+1}, z_{t+1})|z_t]$$

s.t.  \quad \rho_t \left[ \sum_{i \in \mathcal{A}} r_{i,t+1} a_{i,t} \mid z_t \right] \leq \gamma W_t

$$c_t + \sum_{i \in \mathcal{A}} a_{i,t} = W_t$$

- Where $W_{t+1} = c_t + \sum_{i \in \mathcal{A}} (1 + r_{i,t+1})a_{i,t}$
Transactional cost and time independence

\[ V_t(x_t) = \max_{a_t, c_t, d_t \geq 0} \mathbb{E}[V_{t+1}(x_{t+1})] \]

\[ s.t. \quad \rho_t \left[ \sum_i \left( r_{i,t+1} a_{i,t} - \tau (d_{i,t}^+ + d_{i,t}^-) \right) \right] \leq \gamma \left( x_t^c + \sum_i x_{i,t}^a \right) \]

\[ a_{i,t} = x_{i,t}^a + d_{i,t}^+ - d_{i,t}^- \quad \forall i \in \mathcal{A} \]

\[ c_t = x_t^c - (1 + \tau) \sum_i d_{i,t}^+ + (1 - \tau) \sum_i d_{i,t}^- \]

\[
\begin{bmatrix}
a_{t-1} \\
c_{t-1}
\end{bmatrix}
\begin{bmatrix}
a_t \\
c_t
\end{bmatrix}
\begin{bmatrix}
r_t \\
x_t \\
r_{t+1}
\end{bmatrix}
\begin{bmatrix}
x_{t+1}^a \\
x_{t+1}^c \\
c_t(1 + r_f)
\end{bmatrix}
\]
Simplifying notation

\[ V_T(x_T) = x^c_t + \sum_i x^a_{i,t} \]
\[ V_t(x_t) = \max_{u \in U(x_t)} \mathbb{E}[V_{t+1}(x_{t+1}(u))], \quad \forall t \in \{0, \ldots, T - 1\} \]

\[ U(x) = U((x^c, x^a)) \]
\[ = \left\{ (c, a, d^+, d^-) \in \mathbb{R}^{3N+1}_+ \mid \rho_t \left[ \sum_i \left( r_{i,t+1} a_{i,t} - \tau (d^+_{i,t} + d^-_{i,t}) \right) \right] \leq \gamma (x^c_t + \sum_i x^a_{i,t}) \right\} \]

\[ c = x^c - (1 + \tau) \sum_{i \in A} d^+_{i} + (1 - \tau) \sum_{i \in A} d^-_{i} \]
\[ a_i = x^a_{i,t} + d^+_{i} - d^-_{i}, \quad \forall i \in A \]

\[ u = (c, a_1, \ldots a_N, d^+_1, \ldots, d^+_N, d^-_1, \ldots, d^-_N)' \]
\[ W_T(u) = W_T(c, a) = c + \sum_{i \in A} (1 + r_{i,T}) a_i \]
\[ x_{t+1}(u) = x_{t+1}(c, a) = (c_t, (1 + r_{1,t+1}) a_{1,t}, \ldots, (1 + r_{N,t+1}) a_{N,t})' \]
Transactional cost and time independence

- Solution algorithm: SDDP

\[ V_t(x_t) = \max_{u \in U(x_t)} \mathbb{E}[V_{t+1}(x_{t+1}(u))], \quad \forall t \in \{0, ..., T - 1\} \]
Transactional cost and time dependence

\[ V_t(x_t, r_{[t]}) = \max_{u \in U(x_t)} \mathbb{E}[V_{t+1}(x_{t+1}(u)) | r_{[t]}], \quad \forall t \in \{0, \ldots, T - 1\} \]
Transactional cost and time *dependence*

\[ V_t(x_t, r_{[t]}) = \max_{u \in U(x_t)} \mathbb{E}[V_{t+1}(x_{t+1}(u))|r_{[t]}], \quad \forall t \in \{0, \ldots, T - 1\} \]
Transactional cost and Markov dependence

\[ V_t^k(x_t) = \max_{u \in U(x_t)} \sum_{j=1}^K \mathbb{E}\left[ V^k_{t+1}(x_{t+1}(u)) | S_{t+1} = j \right] \mathbb{P}(S_{t+1} = j | S_t = k) \]
Transactional cost and Markov dependence

\[ \mathbb{P}(S_t = 1|S_{t-1} = k) + \mathbb{P}(S_t = 2|S_{t-1} = k) \]
Transactional cost and Markov dependence

\[ \mathbb{P}(S_t = 1|S_{t-1} = k) + \mathbb{P}(S_t = 2|S_{t-1} = k) \]
Transactional cost and Markov dependence

\[ \mathbb{P}(S_t = 1 | S_{t-1} = k) + \mathbb{P}(S_t = 2 | S_{t-1} = k) \]
Adapting SDDP

• Hazard-decision

\[ V_{T-1}(u_{T-1}) \]

\[ u_{T-1} : \text{allocation after buying and selling decisions} \]

• Decision-hazard

\[ V_{T-1}(x_{T-1}) \]

\[ x_{T-1} : \text{allocation before buying and selling decisions} \]
Adapting SDDP: Decision-Hazard information structure

\[ \mathcal{D}_t^j(x_t) = \max_{u \in U(x_t)} \mathbb{E}[\mathcal{D}_{t+1}(x_{t+1})] \]

\[ s.t. \quad x_{t+1} = x_t + A u_t \quad : \pi_t \]

\[ \rho_t [r(u_t)] \leq \gamma \mathbb{1}^\top x_t \quad : \eta_t \]

Linear approximation:

\[ l_t^j(x_t) := \mathcal{D}_t^j(x_t) + (\pi_t^j + \gamma \mathbb{1}^\top \eta_t^j)^\top (x_t - \hat{x}_t) \]

Value function approximation update:

\[ \mathcal{D}_t^j(x_t) \leftarrow \min \{l_t^j(x_t), \mathcal{D}_t^j(x_t)\} \]
Case of study – Brown, Smith 2011

• 12 month horizon
• 3 risky assets and 1 risk-free asset
• Price dynamics given by a factor model

\[
\begin{bmatrix}
\rho_{t+1} \\
z_{t+1}
\end{bmatrix} = \begin{bmatrix}
ar + br z_t \\
a_z + b_z z_t
\end{bmatrix} + \begin{bmatrix}
e_{t+1}
\end{bmatrix}
\]

– Where

\[
\rho_t = \ln(1 + r_t), \quad r_t = (r_{1,t}, r_{2,t}, r_{3,t})^\top \text{ and } \mathbb{P}(r_{0,t} = 0) = 1
\]

• Consider:
  – No-transaction cost, time dependent model via SDP
  – M simulated paths scenarios \((r_{[1,T]}(s_f), s_f = 1, ..., M)\)
Markov chain approximation (MCA)

• How many states of a discrete Markov chain do we need to approximate the dynamics of the factor model?

\[ z_{t+1} = a_z + b_z \, z_t + \nu_{t+1} \]
Markov chain approximation (MCA)

• Compare the SDP and K-state MarkovSDDP
  – 1000 out-of-sample paths and perform a pairwise t-test with zero difference as the null hypothesis

<table>
<thead>
<tr>
<th>Markov SDDP policies</th>
<th>γ</th>
<th>K=1</th>
<th>K=2</th>
<th>K=3</th>
<th>K=4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.02</td>
<td>0.0685</td>
<td>0.0463</td>
<td>0.0481</td>
<td>0.6709</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.3806</td>
<td>0.0447</td>
<td>0.3015</td>
<td>0.6479</td>
</tr>
<tr>
<td></td>
<td>0.08</td>
<td>0.0653</td>
<td>0.0423</td>
<td>0.5115</td>
<td>0.4520</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.1906</td>
<td>0.2112</td>
<td>0.3641</td>
<td>0.4628</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>0.5927</td>
<td>0.5927</td>
<td>0.5927</td>
<td>0.5927</td>
</tr>
</tbody>
</table>

p-values for the difference of expected return
Empirical results

Benchmarks

• One step strategy
  – True price dynamics
  – Current transaction costs
  – Disregards future value

• One step modified strategy
  – True price dynamics
  – Current transaction costs
  – Disregards future transaction costs
  – Value function with simplified state space ($W_t$ instead of $x_t$)

Proposed methodology

• Markov SDDP
  – Approximate price dynamics
  – Current transaction cost
  – Future transaction cost
  – Value function with complete state space ($x_t$)
Case of study (3-asset) - Results
Case of study: 100 assets - 5 factors

Convergence of one instance of the SAA problem

- 12 months
- 3 states and 750 samples
- $\gamma = 0.05$ and $c = 0.01$
Case of study: 100 assets - 5 factors

Solution quality for the *true* problem:

Gap smaller than 0.5% with 99% certainty

• Solve I = 10 randomly generated instances of the SAA of size N = 750
  – Obtain $\overline{UB}_N = \frac{1}{I} \sum_{i=1}^{I} D_0(x_0)(i)$

• Simulate M = 2000 out-of-sample portfolio returns $R_N$
  – Obtain $\overline{LB}_N = \frac{1}{M} \sum_{m=1}^{M} R_N(s)$

• Compute probabilistic GAP

$$\text{GAP} = \overline{UB}_N - \overline{LB}_N + z_\alpha \sqrt{\frac{S_{UB}^2}{I} + \frac{S_{LB}^2}{M}}$$
Case of study: 100 assets - 5 factors

Out-of-sample risk-return curves
Conclusions and future developments

• Time consistent risk-constrained stochastic dynamic asset allocation model
  – Realistic: transaction costs and time dependence
  – Risk aversion: intuitive user defined loss limit
  – Computationally tractable: SDDP with Markovian policy

• Future developments
  – Use the hazard-decision structure with feasibility cuts
  – Formulate a distributionally robust extension of the model
  – How to model risk and ambiguity aversion in the same multistage allocation strategy.
References


Thank you!!!

Prof. Davi Valladão
Industrial Eng. – PUC-Rio

davimv@puc-rio.br