Computations of the Risk-Averse Strategies for Optimal Trade Execution

Qihang Lin
Tepper School of Business
Carnegie Mellon University
(Joint work with Javier Pena)

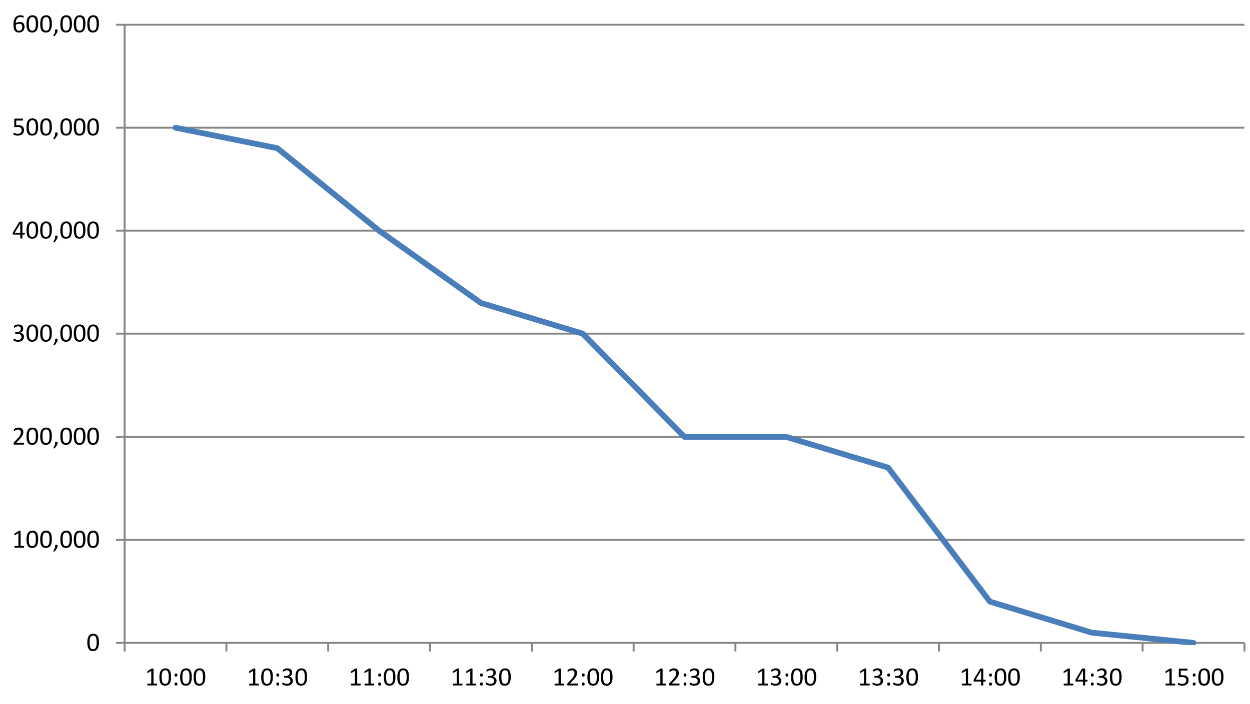
Industrial-Academic Workshop on Optimization in Finance and Risk Management
Fields Institute, Toronto

Introduction

- Model of optimal trade execution
- Dynamic programming with dynamic risk measure
- Second order cone programming
- Smoothing first order method
- Relation to regularized optimization

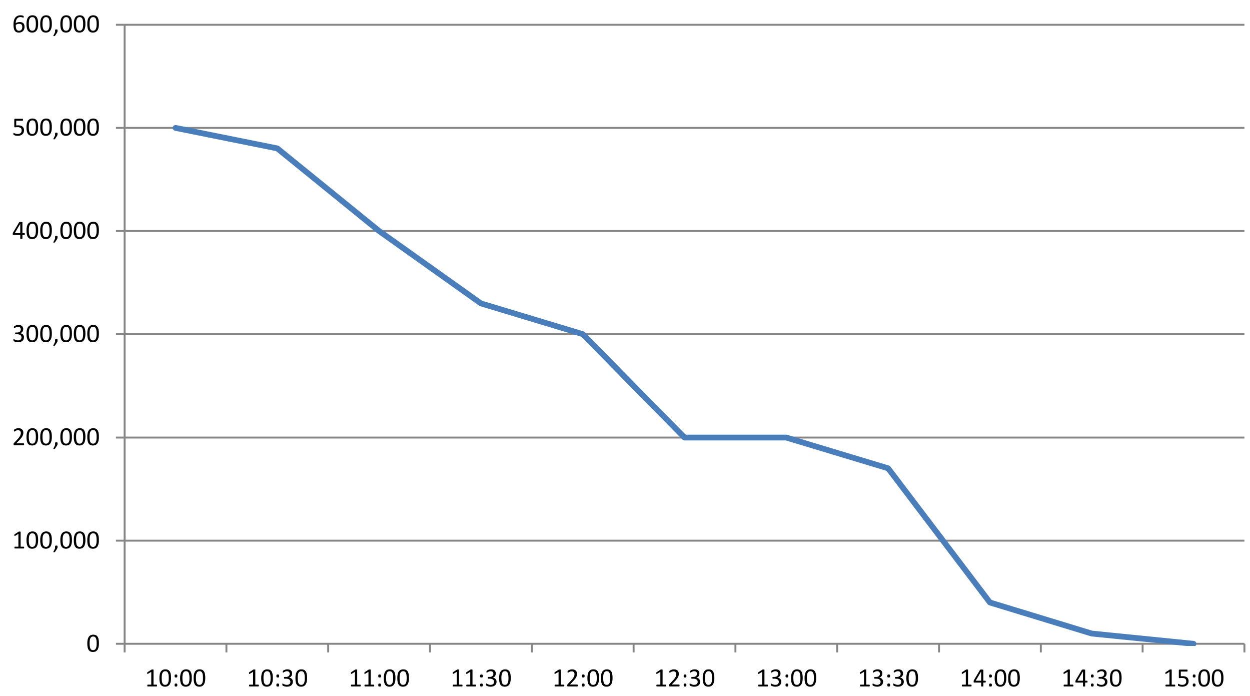
Optimal Trade Execution/ Optimal Liquidation

- We sell or buy a huge volume of a single asset or a portfolio of multi-assets using market orders.
- The market is illiquid and the prices are impacted by orders.
- The whole transaction has to be done within a finite time horizon.



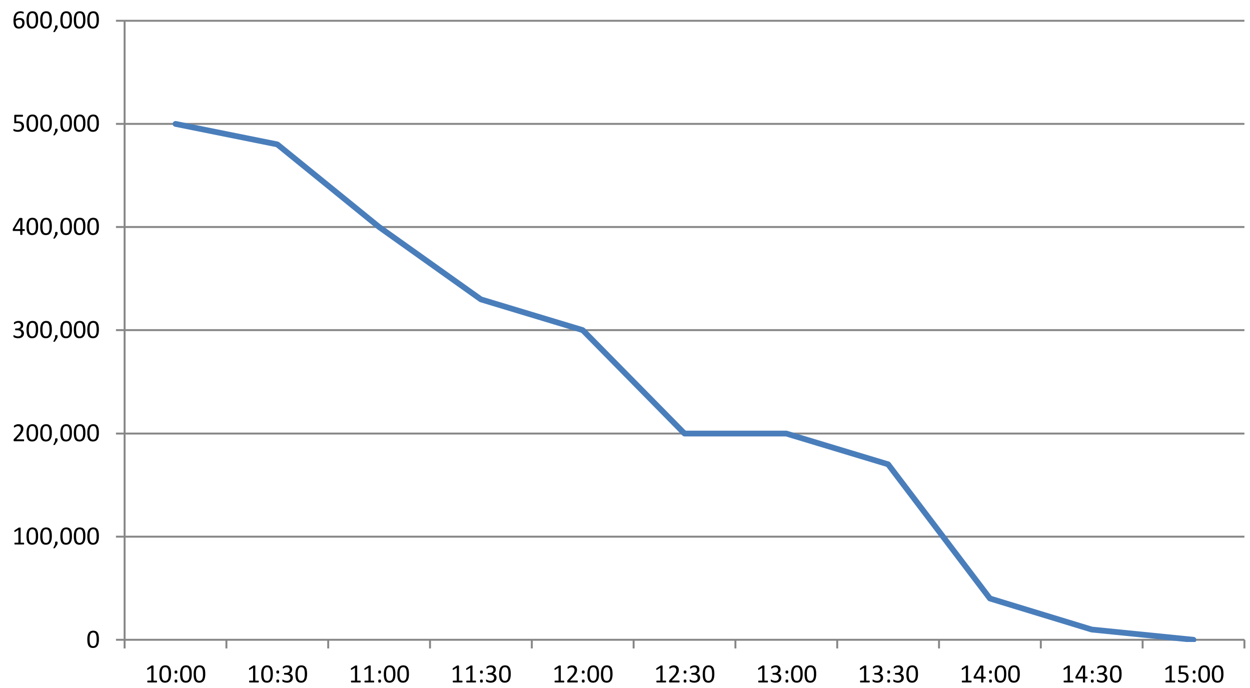
Optimal Trade Execution/ Optimal Liquidation

- Execute a large selling or buying program by a sequence of small market orders
- Focus on selling program in this talk



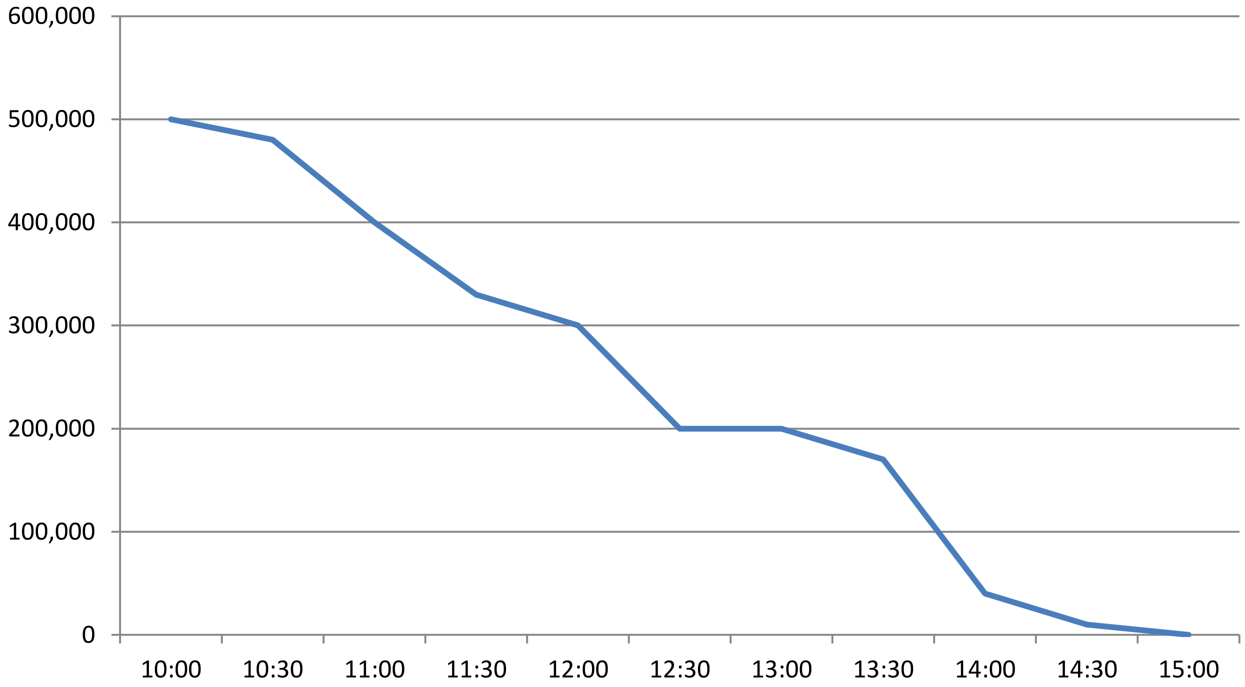
Continuous Time

- Obizhaeva and Wang (2006)
- Schied and Schoneborn (2009)
- Schied, Schoneborn, and Tehranchi (2010)
- Forsyth et al (2009, 2010, 2011)
- Gatheral and Schied (2011)
- Alfonsi, Fruth, and Schied (2010)



Discrete Time

- Bertsimas and Lo (1998)
- Almgren and Chriss (2000)
- Krokhmal and Uryasev (2004)
- Almgren and Lorenze (2007, 2011)
- Moazeni, Coleman and Li (2008, 2011)



Notations and Models

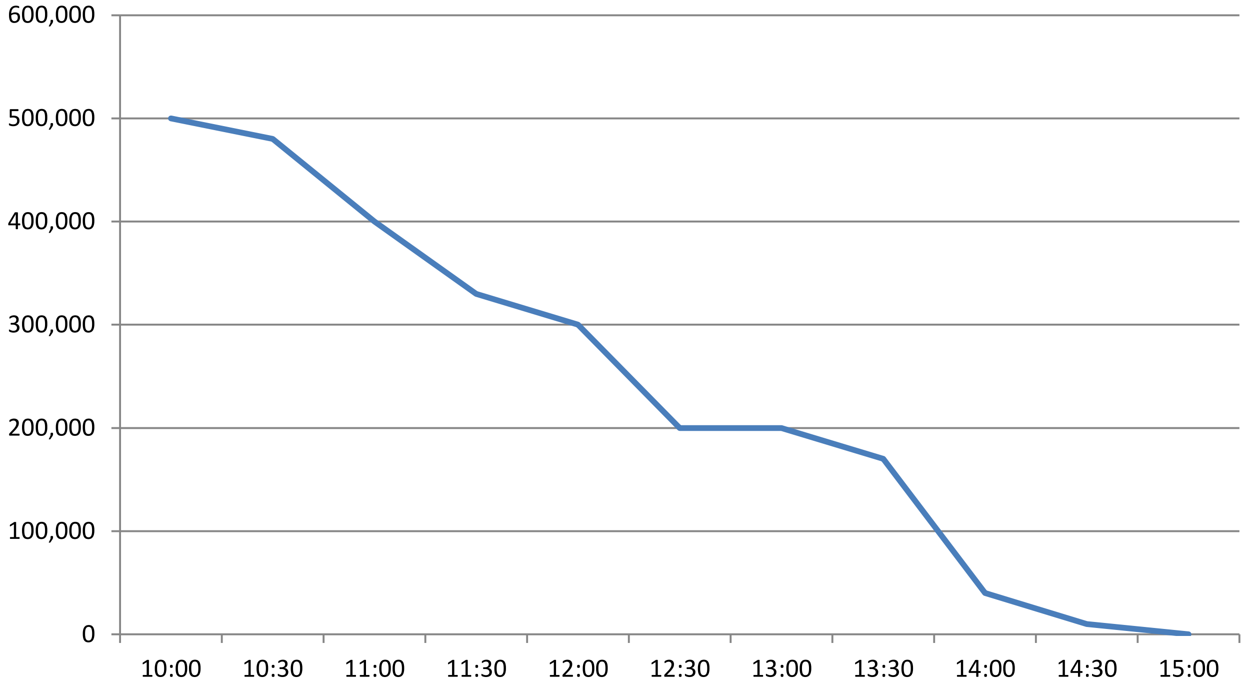
m: The number of assets in the portfolio

$x_0 \in \mathbb{R}^m_+$: The initial portfolio

$[0, T]$: Time period

N: The number of orders

$t_k = \tau k$: The time we submit the kth order, where $\tau = \frac{T}{N}$

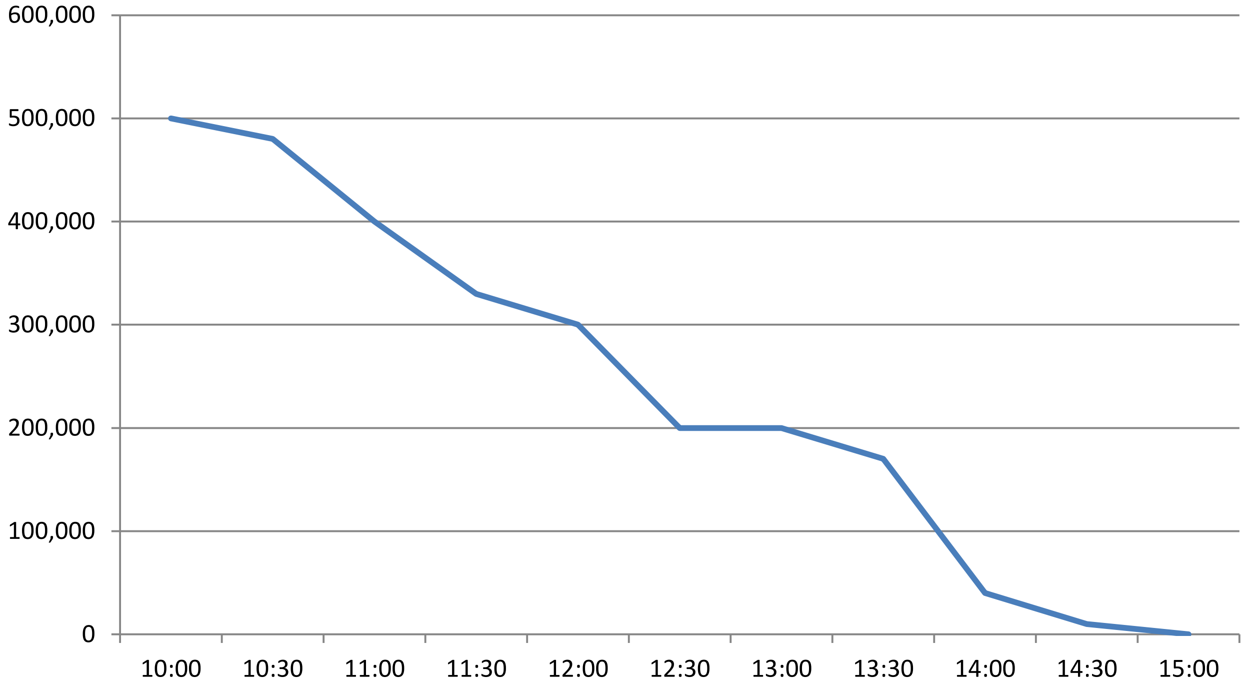


Notations and Models

$x_k \in \mathbb{R}^m_+$: The remaining portfolio at time t_k, $k = 1, \ldots, N$

$n_k \in \mathbb{R}^m_+$: The part of portfolio sold at time t_k, $k = 1, \ldots, N$

$n_k = x_{k-1} - x_k$

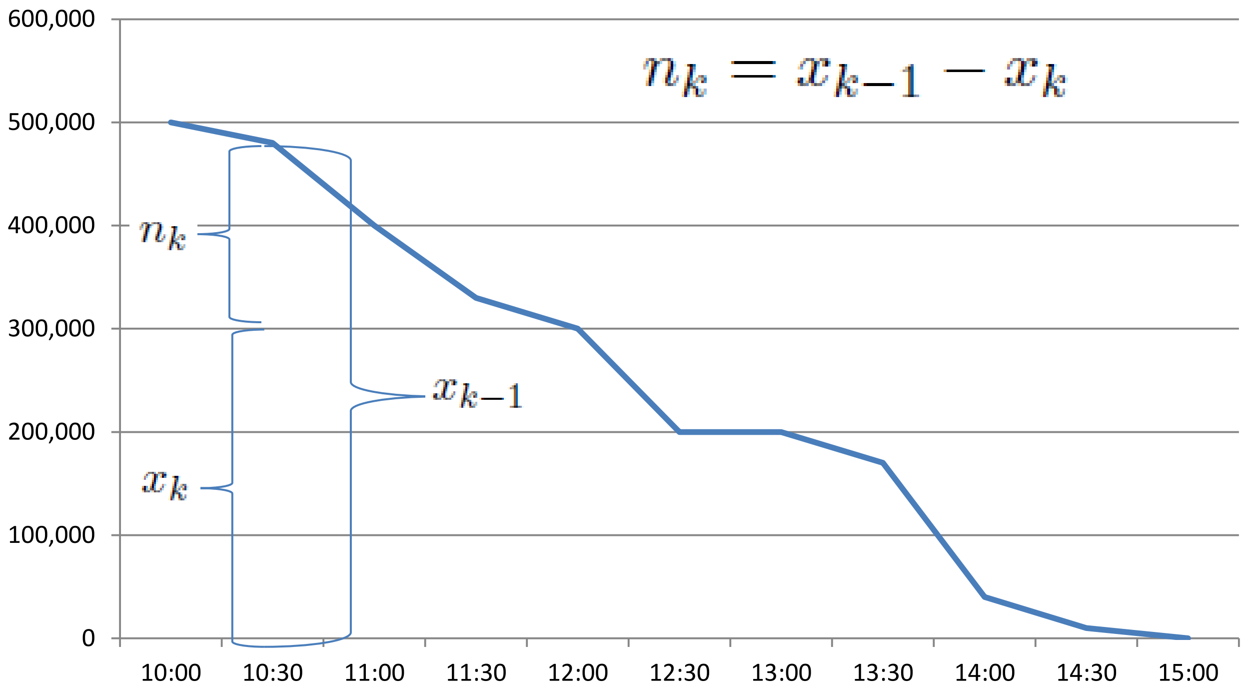


Notations and Models

$x_k \in \mathbb{R}^m_+$: The remaining portfolio at time t_k, $k = 1, \ldots, N$

$n_k \in \mathbb{R}^m_+$: The part of portfolio sold at time t_k, $k = 1, \ldots, N$

$n_k = x_{k-1} - x_k$



Almgren and Chriss (2000)
Almgren and Lorenz(2007, 2011)

$S_k \in \mathbb{R}^m$: The market prices of the assets at time t_k

$S_k = S_{k-1} + \tau^{1/2}\Sigma\xi_k - \tau g(\frac{n_k}{\tau})$, for $k = 1, \ldots, N$

$\xi_k \sim N(0, I_m)$ Stagewise independent

$\tilde{S}_k \in \mathbb{R}^m$: The transaction prices of the assets at time t_k

$\tilde{S}_k = S_{k-1} - h(\frac{n_k}{\tau})$, for $k = 1, \ldots, N$

$g(\frac{n_k}{\tau})$: Permanent market impact

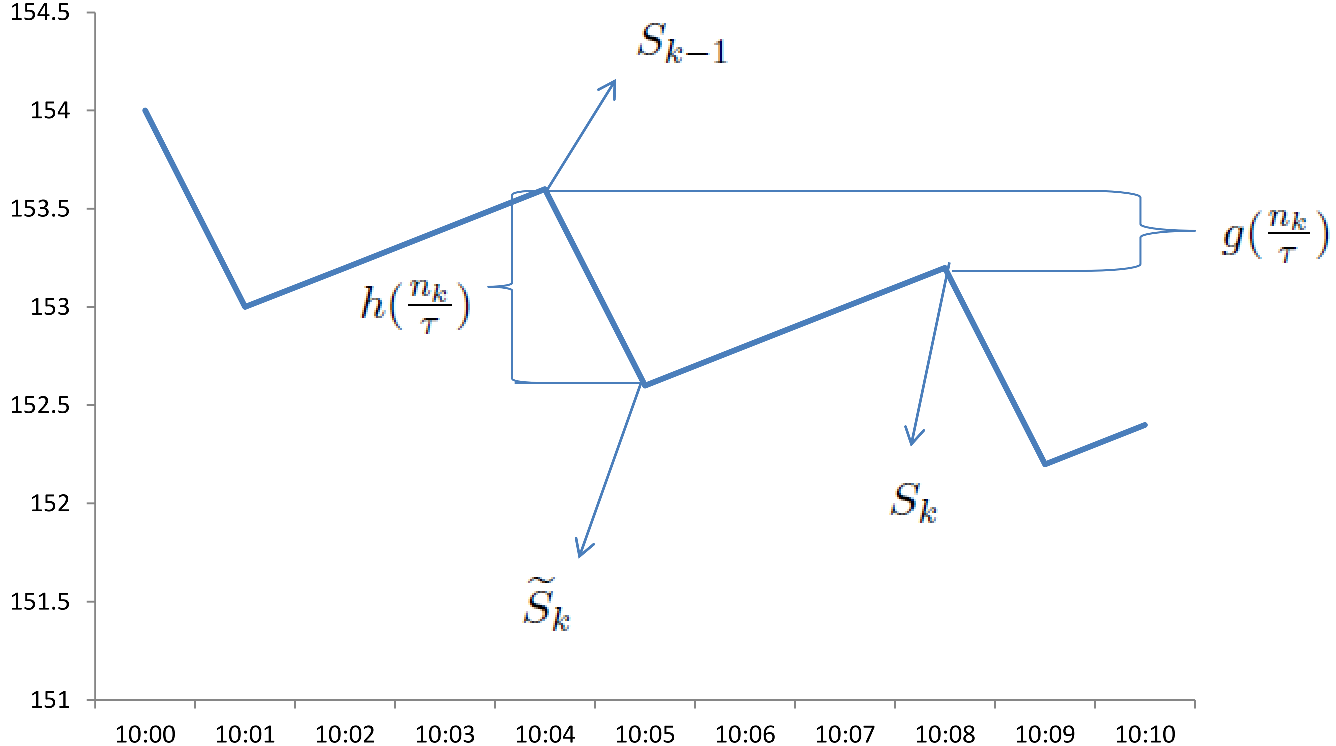
$h(\frac{n_k}{\tau})$: Temporary market impact

$\{F_k\}_k$: The filtration generated by $\{\xi_1, \ldots, \xi_N\}$

Notations and Models

$g(\frac{n_k}{\tau})$: Permanent market impact

$h(\frac{n_k}{\tau})$: Temporary market impact



Notations and Models

Assume linear impact functions which prevents the opportunity of quasi-arbitrage (Huberman and Stanzl, 2004)

$$g\left(\frac{n_k}{\tau}\right) = G\frac{n_k}{\tau}$$
$$h\left(\frac{n_k}{\tau}\right) = H\frac{n_k}{\tau}$$

Positive definite (Almgren and Chriss, 2004)

$$v^T H v > 0 \text{ and } v^T G v > 0 \text{ for all } v \neq 0$$

Dynamic risk measure

Risk measure on sequential cost:

$$Z_1, Z_2, \ldots, Z_N, \quad Z_k = -\tilde{S}_k^T n_k$$

Dynamic risk measure

$$\begin{aligned} \min \quad & \rho(-\tilde{S}_1^T n_1, \ldots, -\tilde{S}_N^T n_N) \\ \text{s.t} \quad & \Delta = (n_1, \ldots, n_N) : \Omega \longmapsto \mathbb{R}_+^{m \times N} \\ & \Delta \mathbf{1}_N = x_0 \\ & n_k \text{ is measurable with respect to } F_k \end{aligned}$$

Time-consistent

Optimal strategy is deterministic

Solved by Second Order Cone Programming (SOCP)

Dynamic risk measure

Risk measure on sequential cost:

$$Z_1, Z_2, \ldots, Z_N, \quad Z_k = -\tilde{S}_k^T n_k$$

Definition

Let $L_k = L_\infty(\Omega, F_k, P)$. A **one-step conditional risk measure** is a mapping $\rho_k : L_{k+1} \longmapsto L_k$, which satisfies the coherent axioms by Artzner et al [1999]:

A1: $\rho_k(\lambda Z + (1-\lambda)W) \leq \lambda \rho_k(Z) + (1-\lambda)\rho_k(W)$;

A2: If $Z \leq W$ then $\rho_k(Z) \leq \rho_k(W), \forall Z, W \in L_{k+1}$;

A3: $\rho_k(Z + W) = Z + \rho_k(W), \forall Z \in L_k, W \in L_{k+1}$;

A4: $\rho_k(\beta Z) = \beta \rho_k(Z), \forall Z \in L_{k+1}, \beta \geq 0$.

Ruszczynski and Shapiro (2004), Ruszczynski (2009), Shapiro (2011)

Example

Mean-semideviation:

$$\rho_k(Z_{k+1}) = \mathbb{E}[Z_{k+1}|F_k] + \beta\mathbb{E}[(Z_{k+1} - \mathbb{E}[Z_{k+1}|F_k])_+^r|F_k]^{\frac{1}{r}}$$

Example

Conditional Value at Risk (CVaR):

$$\rho_k(Z_{k+1}) = \inf_{U\in L_k}\left\{U + \frac{1}{\alpha}\mathbb{E}[(Z_{k+1} - U)_+|F_k]\right\}$$

Dynamic risk measure

Property

Suppose $[Z_{k+1}|F_k]$ *follows a normal distribution* $N(\mu, \sigma^2)$ *and* ρ_k *is chosen to be Mean-semideviation with* $r = 2$ *or CVaR, we have*

$$\rho_k(Z_{k+1}) = \mathbb{E}[Z_{k+1}|F_k] + \beta_k \sqrt{Var[Z_{k+1}|F_k]} = \mu + \beta_k \sigma$$

for a $\beta_k \geq 0$.

Dynamic risk measure

Definition

(Ruszczynski 2009) A **dynamic risk measure** is a mapping $\rho_{k,N} : L_{k+1} \times \cdots \times L_N \longmapsto L_k$ given by:

$$\rho_{k,N}(Z_k, \ldots, Z_N) = Z_k + \rho_k \left(Z_{k+1} + \rho_{k+1}(Z_{k+2} + \cdots + \rho_{N-1}(Z_N))\right)$$

for $k = 1, \ldots, N$

Dynamic risk measure

$$\rho_{1,N}(Z_1, \ldots, Z_N) = Z_1 + \rho_1(Z_2 + \rho_2(Z_3 + \cdots + \rho_{N-1}(Z_N)))$$

$$Z_k = -\tilde{S}_k^T n_k$$

We want to solve

$$\begin{aligned} V_0(x_0, S_0) \equiv \quad & \min \quad \rho_{1,N}(-\tilde{S}_1^T n_1, \ldots, -\tilde{S}_N^T n_N) \\ & \text{s.t} \quad \Delta = (n_1, \ldots, n_N) : \Omega \longmapsto \mathbb{R}_+^{m \times N} \\ & \qquad \Delta \mathbf{1}_N = x_0 \\ & \qquad n_k \in L_k. \end{aligned}$$

Solving the optimal liquidation strategy

State variables:

$$x \in \mathbb{R}_+^m : \text{remaining portfolio}$$

$$s \in \mathbb{R}_+^m : \text{current prices}$$

Cost to go function:

$$\begin{aligned} V_{k-1}(x,s) \equiv \quad \min \quad & \rho_{k,N}(-\tilde{S}_k^T n_k, \ldots, -\tilde{S}_N^T n_N) \\ \text{s.t} \quad & \Delta = (n_k, \ldots, n_N) : \Omega \longmapsto \mathbb{R}_+^{m \times N-k+1} \\ & \Delta \mathbf{1}_{N-k+1} = x \\ & n_k \in L_k \\ & S_{k-1} = s \end{aligned}$$

$V_k(x,s)$ is the minimal dynamic risk, given that we are at time t_k with the observed prices s and the remaining portfolio x.

Solving the optimal liquidation strategy

Theorem

The minimal dynamic risk of selling a portfolio x_0 with initial prices S_0 is $V_0(x_0, S_0)$, which can be solve recursively using the following dynamic programming equations

$$\begin{aligned} V_{N-1}(x,s) &= -\left(s - \frac{1}{\tau}Hx\right)^T x \\ V_k(x,s) &= \min_{0 \le n \le x} -\left(s - \frac{1}{\tau}Hn\right)^T n \\ &\quad + \rho_{k+1}\left(V_{k+1}(x-n, s+\sqrt{\tau}\Sigma\xi - Gn)\right) \end{aligned}$$

This is a particular case of Ruszczynski (2009)

Solving the optimal liquidation strategy

Corollary

The optimal solution is a Markovian policy $\Delta = (n_1, \ldots, n_N)$ *given by*

$$
\begin{aligned}
n_N(x,s) &= x \\
n_k(x,s) &= \arg\min_{0\le n\le x} \left(s - \frac{1}{\tau} H n \right)^T n \\
&\quad + \rho_{k+1} \left(V_{k+1}(x - n, s + \sqrt{\tau}\Sigma\xi - Gn) \right)
\end{aligned}
$$

Solving the optimal liquidation strategy

Corollary

The optimal solution is a Markovian policy $\Delta = (n_1, \ldots, n_N)$ *given by*

$$
\begin{aligned}
n_N(x,s) &= x \\
n_k(x,s) &= \underset{0 \le n \le x}{\arg\min} \left(s - \frac{1}{\tau} H n \right)^T n \\
&\quad + \rho_{k+1} \left(V_{k+1}(x - n, s + \sqrt{\tau}\Sigma\xi - Gn) \right)
\end{aligned}
$$

Solving these equations is still challenging

Solving the optimal liquidation strategy

Theorem

Suppose the risk measure ρ_k is either Mean-semideviation with $r = 2$ or CVaR, we have $V_k(x, s) = f_k(x) - s^T x$ for $k = N-1, \dots, 0$, where the function $f_k(\cdot)$ is defined recursively as follows.

$$
\begin{aligned}
f_{N-1}(x) &= \frac{1}{\tau} x^T H^T x \\
f_k(x) &= \min_{0 \le n \le x} \frac{1}{\tau} n^T H^T n + n^T G^T (x - n) + f_{k+1}(x - n) \\
&\quad + \beta_{k+1} \sqrt{\tau} \|\Sigma^T (x - n)\|_2
\end{aligned}
$$

for $k = N-2, \dots, 0$.

Solving the optimal liquidation strategy

Theorem

Suppose the risk measure ρ_k is either Mean-semideviation with $r = 2$ or CVaR, we have $V_k(x, s) = f_k(x) - s^T x$ for $k = N - 1, \ldots, 0$, where the function $f_k(\cdot)$ is defined recursively as follows.

$$
\begin{aligned}
f_{N-1}(x) &= \frac{1}{\tau} x^T H^T x \\
f_k(x) &= \min_{0 \le n \le x} \frac{1}{\tau} n^T H^T n + n^T G^T (x - n) + f_{k+1}(x - n) \\
&\quad + \beta_{k+1} \sqrt{\tau} \|\Sigma^T (x - n)\|_2
\end{aligned}
$$

for $k = N - 2, \ldots, 0$.

Recall that β_k is from the property of Mean-semideviation and CVaR:

$$\rho_k(Z) = \mathbb{E}[Z] + \beta_k \sqrt{\mathrm{Var}[Z]}$$

when Z is normal distributed.

Solving the optimal liquidation strategy

Corollary

The optimal solution is a deterministic policy $\Delta = (n_1, \ldots, n_N)$ *given by*

$$\begin{aligned} n_N(x) &= x \\ n_k(x) &= \underset{0 \le n \le x}{\arg\min} \frac{1}{\tau} n^T H^T n + n^T G^T (x - n) + f_{k+1}(x - n) \\ &\quad + \beta_{k+1} \sqrt{\tau} \|\Sigma^T (x - n)\|_2 \end{aligned}$$

which does not depend on prices s.

Solving the optimal liquidation strategy

Corollary

In the case of liquidating a single asset ($m = 1$) with $H = h$ and $G = g$, the optimal strategy has a closed form

$$\begin{aligned} n_N(x) &= x \\ n_k(x) &= \min\left\{ \frac{x}{N-k+1} + \frac{\sigma\sqrt{\tau}\sum_{i=1}^{N-k}(i\beta_{N-i})}{(N-k+1)(2h/\tau - g)}, x \right\} \end{aligned}$$

$k = 1, \ldots, N-1$

Solving the optimal liquidation strategy

Corollary

Suppose $\beta_k = 0$ (risk neutral).In the case of liquidating a single asset $(m = 1)$ with $H = h$ and $G = g$, the optimal strategy has a closed form

$$
\begin{aligned}
n_N(x) &= x \\
n_k(x) &= \frac{x}{N-k+1}
\end{aligned}
$$

$k = 1, \ldots, N-1$

Bertsimas and Lo (1998)

Convert to a second order cone programming

Expand the nested optimization

$$
\begin{aligned}
f_{N-1}(x) &= \frac{1}{\tau} x^T H^T x \\
f_k(x) &= \min_{0 \leq n \leq x} \frac{1}{\tau} n^T H^T n + n^T G^T (x-n) + f_{k+1}(x-n) \\
&\quad + \beta_{k+1} \sqrt{\tau} \|\Sigma^T (x-n)\|_2
\end{aligned}
$$

Convert to a second order cone programming

Expand the nested optimization

$$\begin{aligned} f_{N-1}(x) &= \frac{1}{\tau} x^T H^T x \\ f_k(x) &= \min_{0 \le n \le x} \frac{1}{\tau} n^T H^T n + n^T G^T (x - n) + f_{k+1}(x - n) \\ &\quad + \beta_{k+1} \sqrt{\tau} \|\Sigma^T (x - n)\|_2 \end{aligned}$$

$$\Downarrow$$

$$V_0(x_0, S_0) + S_0^T x_0 = f_0(x_0) =$$

$$\min \quad \sum_{k=1}^{N} \left\{ \frac{1}{\tau} n_k^T H^T n_k + n_k^T G^T (x_0 - \sum_{i=1}^{k} n_i) + \beta_k \sqrt{\tau} \|\Sigma^T (x_0 - \sum_{i=1}^{k} n_i)\|_2 \right\}$$

$$\text{s.t} \quad \sum_{k=1}^{N} n_k = x_0$$

$$n_k \ge 0$$

Convert to a second order cone programming

Matrix Version:

$$
\begin{aligned}
\min \quad & \frac{1}{\tau}\,\mathrm{Tr}(\Delta^T H^T \Delta) + \mathrm{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J)) \\
& + \sqrt{\tau} \sum_{k=1}^{N-1} \beta_k \|\Sigma^T (x_0 - \sum_{l=1}^{k} n_l)\|_2 \\
\text{s.t} \quad & \Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{m \times N} \\
& \Delta \mathbf{1}_N = x_0.
\end{aligned}
$$

$$
J = \begin{pmatrix}
1 & 1 & 1 & \cdots & 1 & 1 \\
0 & 1 & 1 & \cdots & 1 & 1 \\
0 & 0 & 1 & \cdots & 1 & 1 \\
\vdots & \vdots & \ddots & \cdots & \vdots & \\
0 & \cdots & 0 & \cdots & 0 & 1
\end{pmatrix}_{N \times N}
$$

Convert to a second order cone programming

minimize the dynamic risk measure--------(SOCP)

$$
\begin{aligned}
\min \quad & \frac{1}{\tau}\mathrm{Tr}(\Delta^T H^T \Delta) + \mathrm{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J)) \\
& + \sqrt{\tau}\sum_{k=1}^{N-1} \beta_k \|\Sigma^T (x_0 - \sum_{l=1}^{k} n_l)\|_2 \\
\text{s.t} \quad & \Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{m\times N} \\
& \Delta \mathbf{1}_N = x_0.
\end{aligned}
$$

minimize the mean-variance risk measure-------(QP)

Almgren and Chriss (2000): Static strategy

Moazeni, Coleman and Li (2011): Model the impacts from other large traders

Moazeni, Coleman and Li (2008): Sensitivity analysis to price impact matrix H and G

$$
\begin{aligned}
\min \quad & \frac{1}{\tau}\mathrm{Tr}(\Delta^T H^T \Delta) + \mathrm{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J)) \\
& + \tau \mathrm{Tr}((x_0 \mathbf{1}_N^T - \Delta J)^T \Sigma \Sigma^T (x_0 \mathbf{1}_N^T - \Delta J)) \\
\text{s.t} \quad & \Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{m\times N} \\
& \Delta \mathbf{1}_N = x_0.
\end{aligned}
$$

Convert to a second order cone programming

minimize the dynamic risk measure--------(SOCP)

$$\min \quad \frac{1}{\tau}\mathrm{Tr}(\Delta^T H^T \Delta) + \mathrm{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J))$$

$$\boxed{+\sqrt{\tau}\sum_{k=1}^{N-1} \beta_k \|\Sigma^T (x_0 - \sum_{l=1}^{k} n_l)\|_2}$$

$$\text{s.t} \quad \Delta = (n_1, \dots, n_N) \in \mathbb{R}_+^{m \times N}$$

$$\Delta \mathbf{1}_N = x_0.$$

minimize the mean-variance risk measure-------(QP)

Almgren and Chriss (2000): Static strategy

Moazeni, Coleman and Li (2011): Model the impacts from other large traders

Moazeni, Coleman and Li (2008): Sensitivity analysis to price impact matrix H and G

$$\min \quad \frac{1}{\tau}\mathrm{Tr}(\Delta^T H^T \Delta) + \mathrm{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J))$$

$$\boxed{+\tau \mathrm{Tr}((x_0 \mathbf{1}_N^T - \Delta J)^T \Sigma \Sigma^T (x_0 \mathbf{1}_N^T - \Delta J))}$$

$$\text{s.t} \quad \Delta = (n_1, \dots, n_N) \in \mathbb{R}_+^{m \times N}$$

$$\Delta \mathbf{1}_N = x_0.$$

First-order algorithm

$$\begin{aligned}
\min \quad & \frac{1}{\tau}\mathrm{Tr}(\Delta^T H^T \Delta) + \mathrm{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J)) \\
& + \sqrt{\tau} \sum_{k=1}^{N-1} \beta_k \|\Sigma^T (x_0 - \sum_{l=1}^{k} n_l)\|_2 \\
\text{s.t} \quad & \Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{m \times N} \\
& \Delta \mathbf{1}_N = x_0.
\end{aligned}$$

- This SOCP is too large for interior-point method, especially when m and N are both large
- A quick response to the financial market is needed
- Interior-point method is not fast enough
- First-order method is considered

First-order algorithm

$$\begin{array}{ll} \min & f(\Delta) + \phi(\Delta) \\ \text{s.t} & \Delta \in Q_1 \end{array}$$

First-order algorithm

$$\min \quad f(\Delta) + \phi(\Delta)$$

$$\text{s.t} \quad \Delta \in Q_1$$

$$f(\Delta) \equiv \frac{1}{\tau}\operatorname{Tr}(\Delta^T H^T \Delta) + \operatorname{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J))$$

Smooth component

First-order algorithm

$$\min \quad f(\Delta) + \phi(\Delta)$$
$$\text{s.t} \quad \Delta \in Q_1$$

$$f(\Delta) \equiv \tfrac{1}{\tau}\,\mathrm{Tr}(\Delta^T H^T \Delta) + \mathrm{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J))$$ Smooth component

$$\phi(\Delta) \equiv \sqrt{\tau} \textstyle\sum_{k=1}^{N-1} \beta_k \|\Sigma^T (x_0 - \sum_{l=1}^{k} n_l)\|_2$$ Non-smooth component

First-order algorithm

$$\min \quad f(\Delta) + \phi(\Delta)$$

$$\text{s.t} \quad \Delta \in Q_1$$

$$f(\Delta) \equiv \frac{1}{\tau} \mathrm{Tr}(\Delta^T H^T \Delta) + \mathrm{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J))$$

Smooth component

$$\phi(\Delta) \equiv \sqrt{\tau} \sum_{k=1}^{N-1} \beta_k \|\Sigma^T (x_0 - \sum_{l=1}^{k} n_l)\|_2$$

Non-smooth component

$$Q_1 \equiv \{\Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{m \times N} | \Delta \mathbf{1}_N = x_0\}$$

Cartesian product of scaled simplexes

First-order algorithm

$$\begin{aligned} \min \quad & f(\Delta) + \phi(\Delta) \\ \text{s.t} \quad & \Delta \in Q_1 \end{aligned}$$

$$f(\Delta) \equiv \frac{1}{\tau}\operatorname{Tr}(\Delta^T H^T \Delta) + \operatorname{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J))$$ Smooth component

$$\phi(\Delta) \equiv \sqrt{\tau} \sum_{k=1}^{N-1} \beta_k \|\Sigma^T (x_0 - \sum_{l=1}^{k} n_l)\|_2$$ Non-smooth component

$$Q_1 \equiv \{\Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{m \times N} | \Delta \mathbf{1}_N = x_0\}$$ Cartesian product of scaled simplexes

Accelerated first-order algorithm
(Nesterov 1988, Tseng 2008)

For $t = 0, 1, 2, \ldots T$:

1. $V_t = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_t$
2. $U_{t+1} = \arg\min_{U \in Q_1} \{\langle U, \nabla f(V_t)\rangle + \phi(U) + \frac{2L}{2+t} D(U, U_t)\}$
3. $\Delta_{t+1} = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_{t+1}$

Output: Δ_{T+1}

First-order algorithm

For $t = 0, 1, 2, \ldots T$:

1. $V_t = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_t$
2. $U_{t+1} = \arg\min_{U \in Q_1}\{\langle U, \nabla f(V_t)\rangle + \phi(U) + \frac{2L}{2+t}D(U, U_t)\}$
3. $\Delta_{t+1} = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_{t+1}$

Output: Δ_{T+1}

User specifies:

$D(X, Y)$: Bregman divergence (1967)

First-order algorithm

For $t = 0, 1, 2, \ldots T$:

1. $V_t = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_t$
2. $U_{t+1} = \boxed{\arg\min_{U \in Q_1}\{\langle U, \nabla f(V_t)\rangle + \phi(U) + \frac{2L}{2+t}D(U, U_t)\}}$
3. $\Delta_{t+1} = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_{t+1}$

Output: Δ_{T+1}

This sub-problem is as hard as the original problem!

User specifies:

$D(X, Y)$: Bregman divergence (1967)

First-order algorithm

For $t = 0, 1, 2, \ldots T$:

1. $V_t = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_t$
2. $U_{t+1} = \arg\min_{U \in Q_1} \{\langle U, \nabla f(V_t)\rangle + \boxed{\phi(U)} + \frac{2L}{2+t} D(U, U_t)\}$
3. $\Delta_{t+1} = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_{t+1}$

Output: Δ_{T+1}

Approximate $\phi(\Delta)$ with a smooth function $\phi_\mu(\Delta)$

First-order algorithm

For $t = 0, 1, 2, \ldots T$:

1. $V_t = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_t$
2. $U_{t+1} = \arg\min_{U \in Q_1}\{\langle U, \nabla f(V_t) + \nabla\phi_\mu(V_t)\rangle + \frac{2L_\mu}{2+t}D(U, U_t)\}$
3. $\Delta_{t+1} = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_{t+1}$

Output: Δ_{T+1}

Approximate $\phi(\Delta)$ with a smooth function $\phi_\mu(\Delta)$

First-order algorithm

For $t = 0, 1, 2, \ldots T$:

1. $V_t = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_t$
2. $U_{t+1} = \arg\min_{U \in Q_1} \{\boxed{\langle U, \nabla f(V_t) + \nabla \phi_\mu(V_t)\rangle} + \frac{2L_\mu}{2+t} D(U, U_t)\}$
3. $\Delta_{t+1} = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_{t+1}$

Output: Δ_{T+1}

Approximate $\phi(\Delta)$ with a smooth function $\phi_\mu(\Delta)$

First-order algorithm

For $t = 0, 1, 2, \ldots T$:

1. $V_t = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_t$
2. $U_{t+1} = \arg\min_{U \in Q_1} \{\boxed{\langle U, \nabla f(V_t) + \nabla \phi_\mu(V_t)\rangle} + \frac{2L_\mu}{2+t} D(U, U_t)\}$
3. $\Delta_{t+1} = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_{t+1}$

Output: Δ_{T+1}

Approximate $\phi(\Delta)$ with a smooth function $\phi_\mu(\Delta)$

First-order algorithm

For $t = 0, 1, 2, \ldots T$:

1. $V_t = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_t$
2. $U_{t+1} = \arg\min_{U \in Q_1}\{\langle U, \nabla f(V_t) + \nabla\phi_\mu(V_t)\rangle + \frac{2L_\mu}{2+t}D(U, U_t)\}$
3. $\Delta_{t+1} = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_{t+1}$

Output: Δ_{T+1}

Approximate $\phi(\Delta)$ with a smooth function $\phi_\mu(\Delta)$

Choose appropriate $D(\cdot, \cdot)$

Smoothing first-order algorithm

For $t = 0, 1, 2, \ldots T$:

1. $V_t = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_t$
2. $U_{t+1} = \arg\min_{U \in Q_1}\{\langle U, \nabla f(V_t) + \nabla\phi_\mu(V_t)\rangle + \frac{2L_\mu}{2+t}D(U, U_t)\}$
3. $\Delta_{t+1} = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_{t+1}$

Output: Δ_{T+1}

Approximate $\phi(\Delta)$ with a smooth function $\phi_\mu(\Delta)$

Choose appropriate $D(\cdot,\cdot)$

Weighted entropy divergence for the Cartesian product of scaled simplexes

$$D(X, Y) = \sum_{i=1}^{m}\sum_{k=1}^{N}\frac{x_{ik}}{x_{0i}}\ln\left(\frac{x_{ik}}{y_{ik}}\right) \quad \text{for all } X, Y \in Q_1$$

Smoothing first-order algorithm

For $t = 0, 1, 2, \ldots T$:

1. $V_t = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_t$
2. $\boxed{U_{t+1} = \arg\min_{U \in Q_1} \{\langle U, \nabla f(V_t) + \nabla \phi_\mu(V_t)\rangle + \frac{2L_\mu}{2+t} D(U, U_t)\}}$
3. $\Delta_{t+1} = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_{t+1}$

Output: Δ_{T+1}

This sub-problem has a closed form solution

Approximate $\phi(\Delta)$ with a smooth function $\phi_\mu(\Delta)$

Choose appropriate $D(\cdot, \cdot)$

Weighted entropy divergence for the Cartesian product of scaled simplexes

$$D(X, Y) = \sum_{i=1}^{m} \sum_{k=1}^{N} \frac{x_{ik}}{x_{0i}} \ln\left(\frac{x_{ik}}{y_{ik}}\right) \quad \text{for all } X, Y \in Q_1$$

SFOM:

For $t = 0, 1, 2, \ldots T$:

1. $V_t = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_t$
2. $U_{t+1} = \arg\min_{U \in Q_1}\{\langle U, \nabla f(V_t) + \nabla \phi_\mu(V_t)\rangle + \frac{2L_\mu}{2+t}D(U, U_t)\}$
3. $\Delta_{t+1} = \frac{t}{2+t}\Delta_t + \frac{2}{2+t}U_{t+1}$

Output: Δ_{T+1}

Smoothing first-order algorithm

$$
\begin{aligned}
\phi(\Delta) &= \sqrt{\tau}\sum_{k=1}^{N-1}\beta_k\|\Sigma^T(x_0-\sum_{l=1}^{k}n_l)\|_2 \\
&= \sum_{k=1}^{N-1}\max_{\|w_k\|_2\le\sqrt{\tau}\beta_k}\left\langle w_k,\Sigma^T(x_0-\sum_{l=1}^{k}n_l)\right\rangle \\
&= \max_{W\in Q_2}\left\langle W,\Sigma^T\left(x_0\mathbf{1}_N^T-\Delta J\right)\right\rangle
\end{aligned}
$$

Smoothing first-order algorithm

$$\begin{aligned}\phi(\Delta) &= \sqrt{\tau}\sum_{k=1}^{N-1}\beta_k\|\Sigma^T(x_0-\sum_{l=1}^{k}n_l)\|_2\\ &= \sum_{k=1}^{N-1}\max_{\|w_k\|_2\le\sqrt{\tau}\beta_k}\left\langle w_k,\Sigma^T(x_0-\sum_{l=1}^{k}n_l)\right\rangle\\ &= \max_{W\in Q_2}\left\langle W,\Sigma^T\left(x_0\mathbf{1}_N^T-\Delta J\right)\right\rangle\end{aligned}$$

$$Q_2=\{W=(w_1,\ldots,w_N)\in\mathbb{R}^{p\times N}\,|\,\|w_k\|_2\le\sqrt{\tau}\beta_k\}$$

Smoothing first-order algorithm

$$\begin{aligned}\phi(\Delta) &= \sqrt{\tau}\sum_{k=1}^{N-1}\beta_k\|\Sigma^T(x_0-\sum_{l=1}^{k}n_l)\|_2 \\ &= \sum_{k=1}^{N-1}\max_{\|w_k\|_2\le\sqrt{\tau}\beta_k}\left\langle w_k,\Sigma^T(x_0-\sum_{l=1}^{k}n_l)\right\rangle \\ &= \max_{W\in Q_2}\left\langle W,\Sigma^T\left(x_0\mathbf{1}_N^T-\Delta J\right)\right\rangle\end{aligned}$$

$$Q_2=\{W=(w_1,\ldots,w_N)\in\mathbb{R}^{p\times N}\,|\,\|w_k\|_2\le\sqrt{\tau}\beta_k\}$$

Nesterov's smoothing technique (2005):

$$\phi_\mu(\Delta) = \max_{W\in Q_2}\left\langle W,\Sigma^T\left(x_0\mathbf{1}_N^T-\Delta J\right)\right\rangle-\frac{\mu}{2}\|W\|_F^2$$

Smoothing first-order algorithm

$$\phi_\mu(\Delta) \quad = \quad \max_{W \in Q_2} \left\langle W, \Sigma^T \left(x_0 \mathbf{1}_N^T - \Delta J \right) \right\rangle - \frac{\mu}{2} \|W\|_F^2$$

$$Q_2 = \{ W = (w_1, \ldots, w_N) \in \mathbb{R}^{p \times N} \mid \|w_k\|_2 \le \sqrt{\tau} \beta_k \}$$

$\phi_\mu(\Delta)$ is convex and smooth

Smoothing first-order algorithm

$$\phi_\mu(\Delta) = \max_{W \in Q_2} \left\langle W, \Sigma^T \left(x_0 \mathbf{1}_N^T - \Delta J \right) \right\rangle - \frac{\mu}{2} \|W\|_F^2$$

$$Q_2 = \{ W = (w_1, \ldots, w_N) \in \mathbb{R}^{p \times N} \| \|w_k\|_2 \leq \sqrt{\tau} \beta_k \}$$

$\phi_\mu(\Delta)$ is convex and smooth

$$\phi_\mu(\Delta) \leq \phi(\Delta) \leq \phi_\mu(\Delta) + \tfrac{\mu\tau}{2} \textstyle\sum_{k=1}^N \beta_k^2$$

Smoothing first-order algorithm

$$\phi_\mu(\Delta) = \max_{W \in Q_2} \left\langle W, \Sigma^T \left(x_0 \mathbf{1}_N^T - \Delta J \right) \right\rangle - \frac{\mu}{2} \|W\|_F^2$$

$$Q_2 = \{ W = (w_1, \ldots, w_N) \in \mathbb{R}^{p \times N} | \|w_k\|_2 \leq \sqrt{\tau} \beta_k \}$$

$\phi_\mu(\Delta)$ is convex and smooth

$$\phi_\mu(\Delta) \leq \phi(\Delta) \leq \phi_\mu(\Delta) + \tfrac{\mu\tau}{2} \textstyle\sum_{k=1}^N \beta_k^2$$

$$\nabla \phi_\mu(\Delta) = -\Sigma W_\mu(\Delta) J^T$$

$$W_\mu(\Delta) = \arg\max_{W \in Q_2} \left\langle W, \Sigma^T \left(x_0 \mathbf{1}_N^T - \Delta J \right) \right\rangle - \frac{\mu}{2} \|W\|_F^2$$

Smoothing first-order algorithm

Theorem

Let Δ^* be the optimal solution. If we set $\mu = \frac{\epsilon}{\tau \sum_{k=1}^{N} \beta_k^2}$, the solution in the t-th iteration, Δ_t, is an ϵ-optimal solution when

$$t \geq 4\|x_0\|_1 \sqrt{\frac{2\|H\|_\infty + 2\tau\|G\|_\infty}{\epsilon\tau} + \frac{T(\sum_{k=1}^{N} \beta_k^2) \max_i \|\Sigma_i\|_2^2}{\epsilon^2}}$$

where Σ_i is the i-th row of Σ.

Smoothing first-order algorithm

Theorem

Let Δ^* be the optimal solution and Δ_t be the solution in the t-th iteration in SFOM. We have

$$f(\Delta_t) + \phi(\Delta_t) - f(\Delta^*) - \phi(\Delta^*) \leq O\left(\frac{1}{t}\right).$$

Numerical experiments

Compare to SDPT3 (Toh, Todd and Tutuncu, 1999) contained in CVX Software (Grant and Boyd)

Assets	Trans	SFOM			SDPT3		
m	N	iter	time(s)	obj	iter	time(s)	obj
10	10	1000	0.4	961	18	0.4	960
10	50	1000	0.5	1605	24	6.5	1603
10	100	1000	0.7	2683	25	51	2681
100	10	1000	0.8	6815	35	70	6815
200	10	1000	1.6	11339	37	689	11338
200	20	1000	2.7	13382	40	4418	13380
500	10	1000	9.5	23352	41	8839	23351
500	20	1000	11.5	30101	N/A	N/A	N/A
1000	10	1000	34.7	44969	N/A	N/A	N/A
1000	20	1000	41.9	47589	N/A	N/A	N/A

Numerical experiments

Compare to SDPT3 (Toh, Todd and Tutuncu, 1999) contained in CVX Software (Grant and Boyd)

Assets	Trans	SFOM			SDPT3		
m	N	iter	time(s)	obj	iter	time(s)	obj
10	10	1000	0.4	961	18	0.4	960
10	50	1000	0.5	1605	24	6.5	1603
10	100	1000	0.7	2683	25	51	2681
100	10	1000	0.8	6815	35	70	6815
200	10	1000	1.6	11339	37	689	11338
200	20	1000	2.7	13382	40	4418	13380
500	10	1000	9.5	23352	41	8839	23351
500	20	1000	11.5	30101	N/A	N/A	N/A
1000	10	1000	34.7	44969	N/A	N/A	N/A
1000	20	1000	41.9	47589	N/A	N/A	N/A

Connection to Regularized Optimization

Lasso (Tibshirani,1996)

$$\min \quad \frac{1}{2}\|Ax - b\|_2^2 + \lambda \sum_{k=1}^{N} |x_k|$$

Single asset liquidation under dynamic risk measure

$$\begin{aligned}
\min \quad & \frac{h}{\tau}\|\Delta\|_2^2 + g\,\langle \Delta, \Pi \rangle + \sqrt{\tau}\sigma \sum_{k=1}^{N} \beta_k |x_k| \\
\text{s.t} \quad & \Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{1\times N} \\
& \Pi = (x_1, \ldots, x_N) \in \mathbb{R}_+^{1\times N} \\
& \Pi = x_0 \mathbf{1}_N^T - \Delta J \\
& \Delta \mathbf{1}_N = x_0 .
\end{aligned}$$

Connection to Regularized Optimization

Lasso (Tibshirani,1996)

$$\min \quad \frac{1}{2}\|Ax - b\|_2^2 + \lambda \boxed{\sum_{k=1}^{N} |x_k|}$$

They contain a same term in the objective function

Single asset liquidation under dynamic risk measure

$$\begin{aligned}
\min \quad & \frac{h}{\tau}\|\Delta\|_2^2 + g\left\langle \Delta, \Pi \right\rangle + \sqrt{\tau}\sigma \boxed{\sum_{k=1}^{N} \beta_k |x_k|} \\
\text{s.t} \quad & \Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{1\times N} \\
& \Pi = (x_1, \ldots, x_N) \in \mathbb{R}_+^{1\times N} \\
& \Pi = x_0 \mathbf{1}_N^T - \Delta J \\
& \Delta \mathbf{1}_N = x_0 .
\end{aligned}$$

Connection to Regularized Optimization

Lasso (Tibshirani,1996)

$$\min \quad \frac{1}{2}\|Ax - b\|_2^2 + \lambda \boxed{\sum_{k=1}^{N} |x_k|}$$

To enforce a sparse solution

Single asset liquidation under dynamic risk measure

$$\begin{aligned}
\min \quad & \frac{h}{\tau}\|\Delta\|_2^2 + g\,\langle \Delta, \Pi \rangle + \sqrt{\tau}\sigma \sum_{k=1}^{N} \beta_k |x_k| \\
\text{s.t} \quad & \Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{1\times N} \\
& \Pi = (x_1, \ldots, x_N) \in \mathbb{R}_+^{1\times N} \\
& \Pi = x_0 \mathbf{1}_N^T - \Delta J \\
& \Delta \mathbf{1}_N = x_0.
\end{aligned}$$

Connection to Regularized Optimization

Lasso (Tibshirani,1996)

$$\min \quad \frac{1}{2}\|Ax-b\|_2^2+\lambda\sum_{k=1}^{N}|x_k|$$

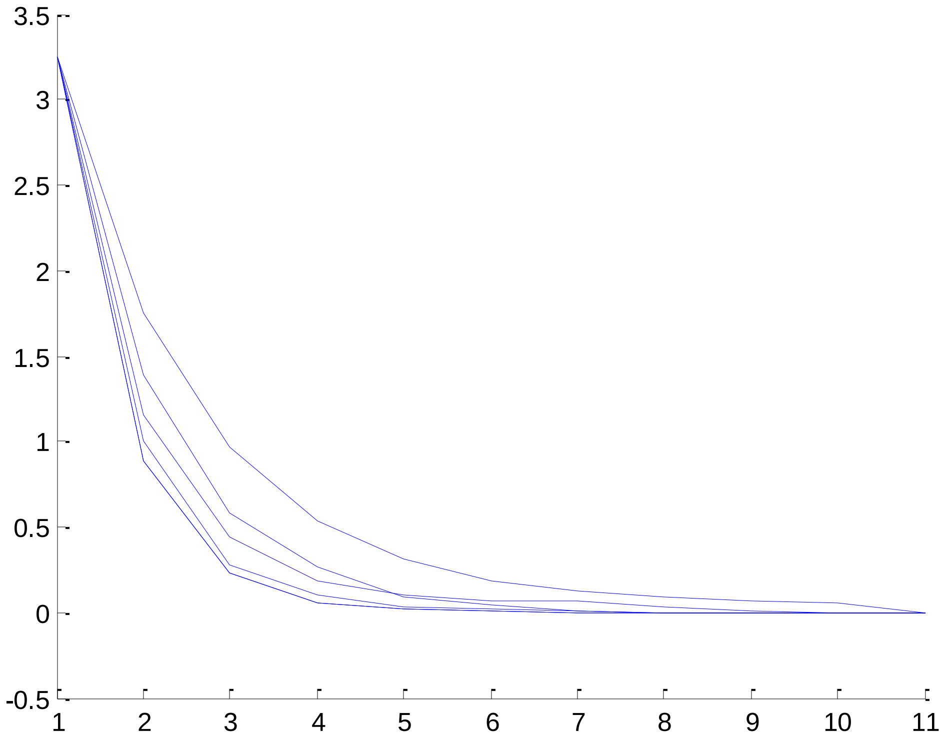
Single asset liquidation under dynamic risk measure

$$\begin{aligned}
\min \quad & \frac{h}{\tau}\|\Delta\|_2^2+g\,\langle\Delta,\Pi\rangle+\sqrt{\tau}\sigma\boxed{\sum_{k=1}^{N}\beta_k|x_k|}\\
\text{s.t} \quad & \Delta=(n_1,\ldots,n_N)\in\mathbb{R}_+^{1\times N}\\
& \Pi=(x_1,\ldots,x_N)\in\mathbb{R}_+^{1\times N}\\
& \Pi=x_0\mathbf{1}_N^T-\Delta J\\
& \Delta\mathbf{1}_N=x_0.
\end{aligned}$$

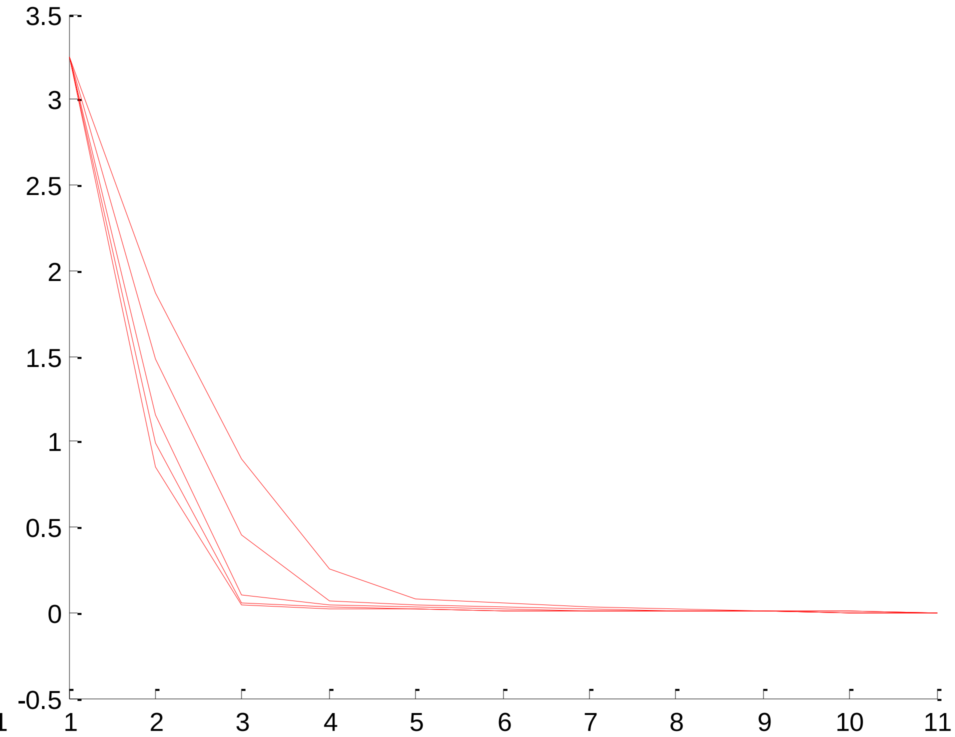
"Sparse" here means a Early-completed liquidation strategy

Connection to Regularized Optimization

Static Strategy which minimizes the Mean-Variance



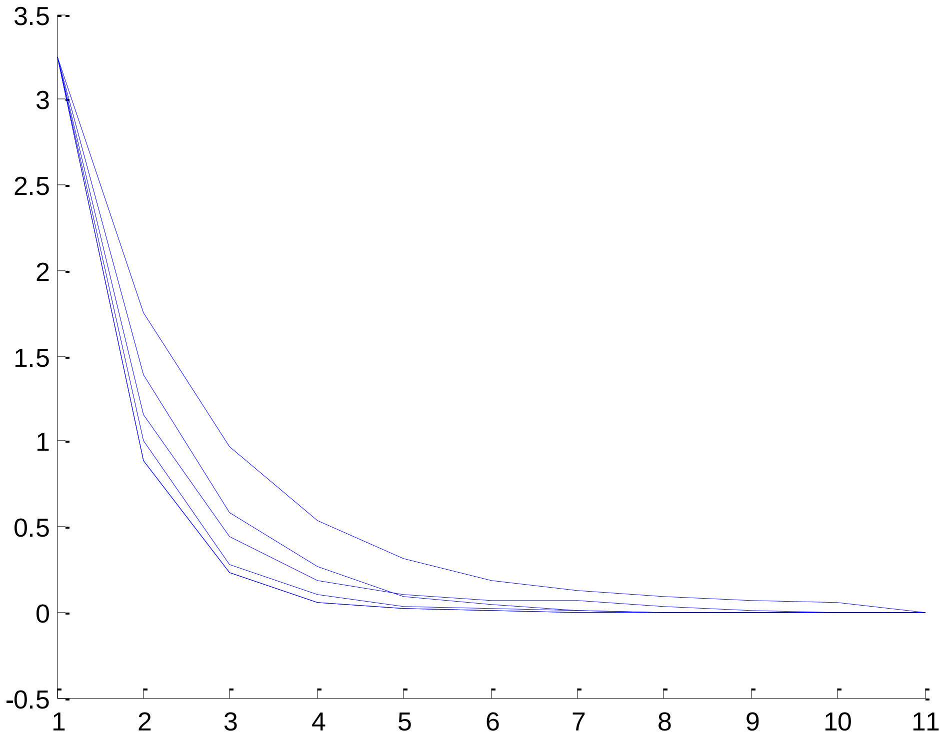
Strategy which minimizes the dynamic risk measure

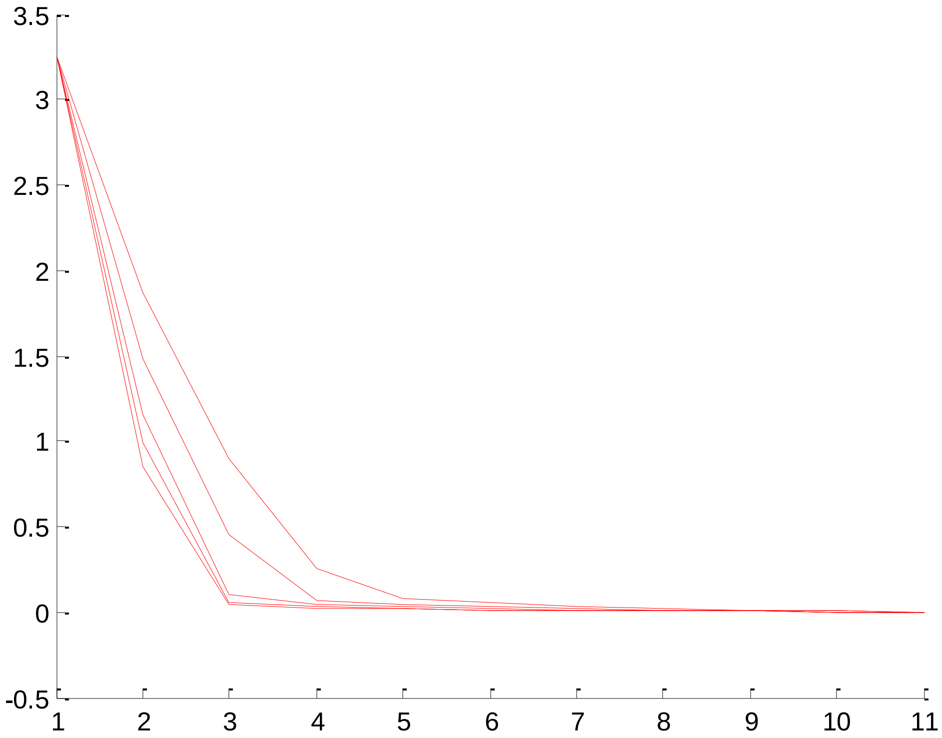


Connection to Regularized Optimization

Static Strategy which minimizes the Mean-Variance

Strategy which minimizes the dynamic risk measure





Allow a long tail
Finish with a sequence of tiny orders
Not practical since a fixed transaction fee could be charge for each order

No long tail
All shares would be sold at an early stage
Due to the regularization property

Connection to Regularized Optimization

Group Lasso (Yuan and Lin, 2004)

$$\min \quad \frac{1}{2}\|AX - b\|_2^2 + \lambda \sum_{k=1}^{N} \|x_k\|_2$$

Portfolio liquidation under dynamic risk measure

$$\begin{aligned}
\min \quad & \frac{1}{\tau}\mathrm{Tr}(\Delta^T H^T \Delta) + \mathrm{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J)) \\
& + \sqrt{\tau} \sum_{k=1}^{N-1} \beta_k \|\Sigma^T x_k\|_2 \\
\text{s.t} \quad & \Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{m \times N} \\
& \Pi = (x_1, \ldots, x_N) \in \mathbb{R}_+^{1 \times N} \\
& \Pi = x_0 \mathbf{1}_N^T - \Delta J \\
& \Delta \mathbf{1}_N = x_0.
\end{aligned}$$

Connection to Regularized Optimization

Group Lasso (Yuan and Lin, 2004)

$$\min \quad \frac{1}{2}\|AX - b\|_2^2 + \boxed{\lambda \sum_{k=1}^{N} \|x_k\|_2}$$

Portfolio liquidation under dynamic risk measure

$$\begin{aligned}
\min \quad & \frac{1}{\tau}\operatorname{Tr}(\Delta^T H^T \Delta) + \operatorname{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J)) \\
& + \sqrt{\tau}\boxed{\sum_{k=1}^{N-1} \beta_k \|\Sigma^T x_k\|_2} \\
\text{s.t} \quad & \Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{m \times N} \\
& \Pi = (x_1, \ldots, x_N) \in \mathbb{R}_+^{1 \times N} \\
& \Pi = x_0 \mathbf{1}_N^T - \Delta J \\
& \Delta \mathbf{1}_N = x_0.
\end{aligned}$$

Connection to Regularized Optimization

Group Lasso (Yuan and Lin, 2004)

$$\min \quad \frac{1}{2}\|AX - b\|_2^2 + \boxed{\lambda \sum_{k=1}^{N} \|x_k\|_2}$$

Enforce a group sparsity in the solution

Portfolio liquidation under dynamic risk measure

$$\begin{aligned}
\min \quad & \frac{1}{\tau}\mathrm{Tr}(\Delta^T H^T \Delta) + \mathrm{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J)) \\
& + \sqrt{\tau} \sum_{k=1}^{N-1} \beta_k \|\Sigma^T x_k\|_2 \\
\text{s.t} \quad & \Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{m \times N} \\
& \Pi = (x_1, \ldots, x_N) \in \mathbb{R}_+^{1 \times N} \\
& \Pi = x_0 \mathbf{1}_N^T - \Delta J \\
& \Delta \mathbf{1}_N = x_0.
\end{aligned}$$

Connection to Regularized Optimization

Group Lasso (Yuan and Lin, 2004)

$$\min \quad \frac{1}{2}\|AX - b\|_2^2 + \lambda \sum_{k=1}^{N} \|x_k\|_2$$

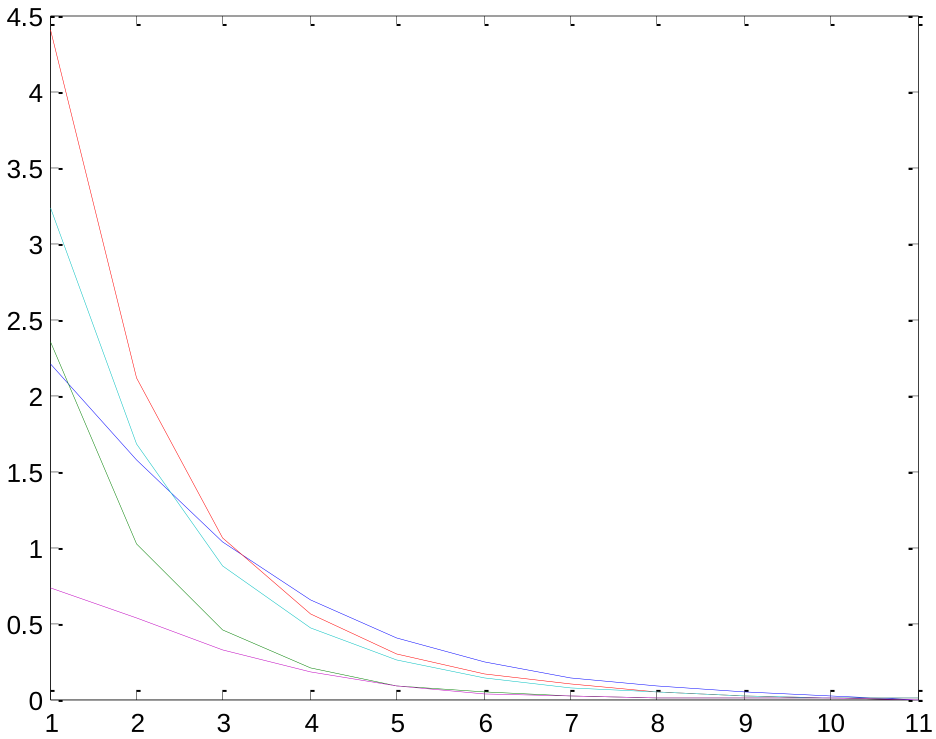
Portfolio liquidation under dynamic risk measure

$$\begin{aligned}
\min \quad & \frac{1}{\tau}\operatorname{Tr}(\Delta^T H^T \Delta) + \operatorname{Tr}(\Delta^T G^T (x_0 \mathbf{1}_N^T - \Delta J)) \\
& + \sqrt{\tau} \boxed{\sum_{k=1}^{N-1} \beta_k \|\Sigma^T x_k\|_2} \longrightarrow \\
\text{s.t} \quad & \Delta = (n_1, \ldots, n_N) \in \mathbb{R}_+^{m \times N} \\
& \Pi = (x_1, \ldots, x_N) \in \mathbb{R}_+^{1 \times N} \\
& \Pi = x_0 \mathbf{1}_N^T - \Delta J \\
& \Delta \mathbf{1}_N = x_0.
\end{aligned}$$

"Sparse" here means a Early-completed liquidation strategy

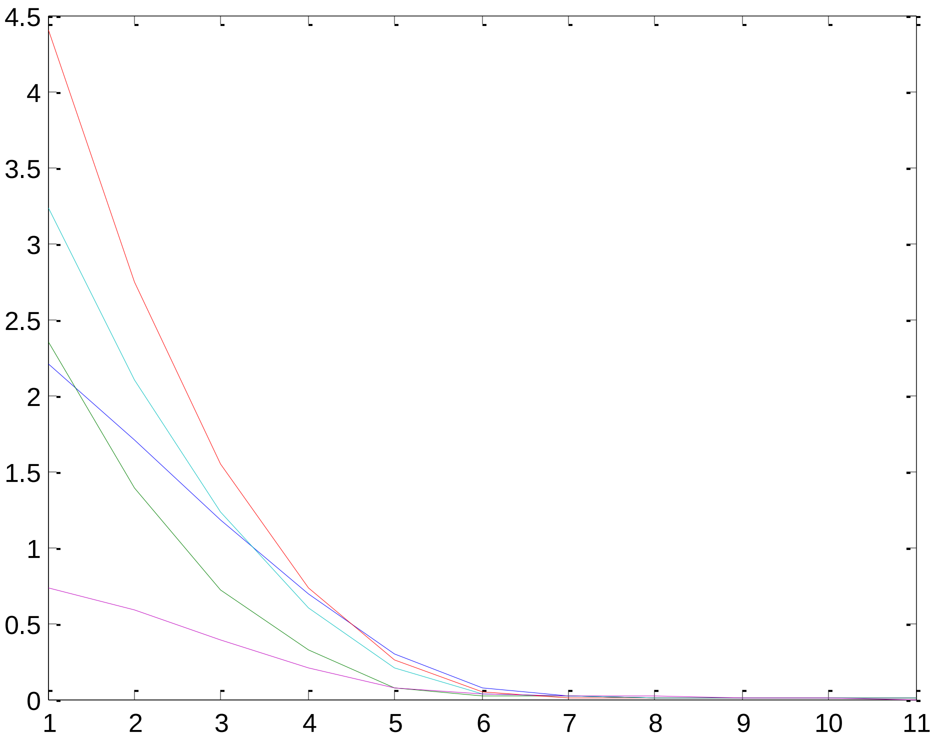
Connection to Regularized Optimization

Static Strategy which minimizes the Mean-Variance



Allow a long tail
Finish with a sequence of tiny orders
Not practical since a fixed transaction fee could be charge for each order

Strategy which minimizes the dynamic risk measure



No long tail
All shares would be sold at an early stage
Due to the regularization property

Thank you!
Questions and Comments