

# Market-Neutral Volatility Harvesting (MN VH) と Ledger-SPT

— 離散時間・自己資金制約のもとで分散を収益化する実装可能フレーム  
ワーク

Daiki Sasaki, MD, PhD

September 23, 2025

## 概要

本稿は、Market-Neutral Volatility Harvesting (MN VH) を、自己資金 (self-financing) と会計厳密性を満たす実装層 Ledger-SPT と共に、離散時間で定式化する。MN VH は各資産のステップ収益  $r_{i,t}$  とポートフォリオ収益  $R_t$  の乖離  $R_t - r_{i,t}$  を駆動信号として数量を調整し、市場中立を維持しつつ横断的分散 (cross-sectional variance) と平均回帰から期待収益を得る。取引は初期ポジションゼロから開始 ( $q_{i,0} = 0$ ) し、以後は Ledger-SPT により外部資金注入なしで回す。

## 1 導入

Shannon's Demon は再均衡の成長効果を示し、SPT (Stochastic Portfolio Theory) は関数生成ポートフォリオの相対成長率に「分散 × 回転」の項が現れることを与える。一方で多くの理論は連続時間・無摩擦を仮定し、実運用に不可欠な台帳・手数料・ロット等の制約を省略する。本稿の貢献は、MN VH の最小実装を離散時間で与え、自己資金制約・手数料・ロット調整を Ledger-SPT で厳密化し、初期条件  $q_{i,0} = 0$  を明示する点にある。

## 2 記法と前提（離散時間・初期ゼロ）

- 資産  $i = 1, \dots, n$ 、価格  $p_{i,t} > 0$ 、単純収益  $r_{i,t} = \frac{p_{i,t+1} - p_{i,t}}{p_{i,t}}$ 。
- 保有数量  $q_{i,t}$ 、ポートフォリオ時価  $V_t = \sum_i p_{i,t} q_{i,t}$ 。
- 目標ノーション重み  $w_i$  ( $\sum_i w_i = 1$ 、負値可)。
- 初期条件:  $q_{i,0} = 0$ ,  $V_0 = C_0$  (現金のみ)。
- 自己資金制約: 各ステップの取引現金収支 + 手数料 = 0。外部キャッシュフロー  $CF_t = 0$ 。

### 3 MNVH の更新則（数量制御）

提案する基本更新則は数量ベースで次式とする：

$$q_{i,t+1} = \lambda q_{i,t} + \beta \frac{w_i}{p_{i,t}} (R_t - r_{i,t}), \quad R_t = \sum_j w_j r_{j,t}. \quad (1)$$

ここで  $\lambda \in [0, 1]$  は在庫の自己回帰、 $\beta > 0$  は取引強度である。直近ステップで過小パフォーマンスの銘柄を買い、過大パフォーマンスを売ることにより、次期の平均回帰を収穫する。数量で制御するため価格水準の影響を相対的に回避しやすい。

### 4 Ledger-SPT：会計厳密・自己資金実行

Ledger-SPT は、台帳（ledger）に銘柄ごとのロット（数量・価格・手数料）と現金口座  $C_t$  を保持し、自己資金性を強制する。

- 取引案:  $\Delta q_{i,t} = q_{i,t+1} - q_{i,t}$  を集計し、想定手数料  $\phi_{i,t}$  を含めて検証する。
- 自己資金チェック:

$$\sum_i p_{i,t} \Delta q_{i,t} + \sum_i \phi_{i,t} = 0. \quad (2)$$

満たさない場合はスカラー  $\kappa_t \in (0, 1]$  を求め、 $\Delta \tilde{q}_{i,t} = \kappa_t \Delta q_{i,t}$  に一括スケールする。

- PnL 分解（ミッド評価）：

$$\Delta V_t = \underbrace{\sum_i q_{i,t} p_{i,t} r_{i,t}}_{\text{MTM}} + \underbrace{\sum_i p_{i,t} \Delta \tilde{q}_{i,t}}_{\text{取引キャッシュフロー}} - \sum_i \phi_{i,t}. \quad (3)$$

### 5 期待収益の形（平均回帰仮定）

一步先の平均回帰モデル  $(r_{i,t+1} - R_{t+1}) \approx -\rho(r_{i,t} - R_t) + \varepsilon$  を仮定すると、一次近似で

$$\mathbb{E}[\Delta V_{t \rightarrow t+1}^{\text{trade}}] \approx \beta V_t \rho \mathbb{E}\left[\sum_i w_i (r_{i,t} - R_t)^2\right] - \mathbb{E}[\text{コスト}]. \quad (4)$$

ここで  $\rho \in (0, 1)$  は平均回帰係数、 $\sum_i w_i (r_{i,t} - R_t)^2$  は横断的分散（XS-Var<sub>t</sub>）である。十分条件は  $\beta V_t \rho \mathbb{E}[\text{XS-Var}_t] > (\text{手数料} + \text{スリッページ})$ 。

### 6 SPT との関係

SPT の連続時間理論における相対成長の「超過成長率」項（分散由来のドリフト）に対応して、MNVH は離散時間で  $(R_t - r_{i,t})$  に比例する経験的勾配一步を踏むことで、分散

× 回転を設計的に回収する。Ledger-SPT がその会計回収を保証する。

## 7 SPT 定義式からの $(R - r)$ 信号の導出と市場中立系への落とし込み

### 7.1 SPT の基本式と離散時間近似

SPT の連続時間において、価格プロセス  $S_i$  とウェイト  $\pi_i$  をもつポートフォリオの相対富  $V^\pi/V^{\text{mkt}}$  の対数微分は、生成関数型の定理により

$$d \log V_t^\pi = \sum_i \pi_{i,t} d \log S_{i,t} + \frac{1}{2} \Gamma_t(\pi) dt \quad (\text{簡略形}), \quad (5)$$

ここで第2項は超過成長率であり、概念的には「構成銘柄の分散 – ポートフォリオ分散」型の量で正に寄与する（詳細は連続時間の定式化に依存）。これを1ステップの離散時間に写像して、単純収益  $r_{i,t}$  とポートフォリオ収益  $R_t = \sum_i w_i r_{i,t}$  を用いると、次の関数

$$\mathcal{G}_t(w) := -\frac{1}{2} \sum_i w_i (r_{i,t} - R_t)^2 \quad (\text{横断的分散を負符号でとった「生成関数」}) \quad (6)$$

は、「横断的分散が大きいほど（小さくする方向に）勾配が大きい」という性質を持ち、SPT の超過成長率項に対応する離散時間の代理目的になりうる。

### 7.2 勾配と数量更新： $(R - r)$ の自然発生

数量  $q_{i,t}$  による微小更新を考える。 $V_t = \sum_j p_{j,t} q_{j,t}$ 、 $w_i = \frac{p_{i,t} q_{i,t}}{V_t}$  とおき、 $\mathcal{G}_t$  の  $q_{i,t}$  に関する一階変分を（ $V_t$  の1ステップ安定近似の下で）とると

$$\frac{\partial \mathcal{G}_t}{\partial q_{i,t}} \propto \frac{\partial \mathcal{G}_t}{\partial w_i} \frac{\partial w_i}{\partial q_{i,t}} \propto -(r_{i,t} - R_t) \frac{w_i}{p_{i,t}}. \quad (7)$$

よって勾配上昇方向（ $\mathcal{G}_t$  を増やす方向）への最小ステップは

$$\Delta q_{i,t} \propto + \frac{w_i}{p_{i,t}} (R_t - r_{i,t}), \quad (8)$$

すなわち MNVH の核となる信号  $(R_t - r_{i,t})$  が、横断分散型の生成関数から機械的に導かれる。直観的には、「直近で平均（ポートフォリオ）から外れた分だけ平均へ引き戻す」のが最短の勾配一步になっている。

### 7.3 自己資金・市場中立系への落とし込み

上式の無拘束ステップはそのままでは現金制約を満たさない。Ledger-SPT は以下で市場中立な自己資金系に埋め込む：

1. 自己資金制約の強制：提案  $\Delta q_{i,t}$  から発生する現金需要  $S = \sum_i p_{i,t} \Delta q_{i,t} + \sum_i \phi_{i,t}$  を評価し、スカラー  $\kappa_t \in (0, 1]$  を用いて  $\Delta \tilde{q}_{i,t} = \kappa_t \Delta q_{i,t}$  として  $S = 0$  を満たす。
2. 名目（ベータ）中立化：必要に応じ、ベンチマークに対する回帰  $\beta$  を 0 に近づけるため、(a) ドライバに中立化項を入れる、(b) ヘッジ資産を追加して  $\sum_i \hat{\beta}_i p_{i,t} \Delta \tilde{q}_{i,t} = 0$  を近似的に課す、のいずれかを探る。
3. 初期ゼロ条件： $q_{i,0} = 0$ （現金のみ）から開始し、以降は常に 1. の自己資金制約で運用するため、外部資金の注入・引出しなしに市場中立の在庫が生成・消滅する。

以上により、SPT の「分散が富の相対成長に寄与する」という定性的知見は、離散時間の数量制御則  $\Delta q_{i,t} \propto (R_t - r_{i,t}) w_i / p_{i,t}$  と、それを台帳レベルで自己資金化する工程に二分され、実装可能な形で結実する。

## 8 推定とパラメータ化

- 重み  $w_i$ : 等重み／逆ボラ ( $\propto \sigma_i^{-1}$ ) ／縮小共分散を用いた軽量 Markowitz。
- $\rho$ :  $(r_{i,t+1} - R_{t+1})$  を  $(r_{i,t} - R_t)$  に回帰しローリング推定、 $[0, 1]$  にクリップ。
- $\beta$ : 回転率・証拠金・参加率から逆算し、Fee/Gross を 20–40% 程度で安定化。
- $\lambda$ : 在庫の自然減衰 (1 ステップで 0.8–0.98 を目安)。
- コスト: 手数料 + 参加率 × スプレッド × ボラの経験式をログ化。

## 9 実験設計（初期ゼロを厳守）

- 初期条件:  $q_{i,0} = 0$ ,  $C_0 = V_0$ 。以降は自己資金のみ。
- ユニバース: 高流動の先物/パーペチュアル or 現物株 30–200 銘柄。
- タイムスケール: 1–60 分 or 日次。全銘柄同期ステップ。
- BT 設定: OOS ローリング、現実的フィー、約定順序/隊列スリッページ、資金調達・借株。
- 評価指標: CAGR、Vol、Sharpe、回転率、Fee/Gross、勝率、歪度・尖度、DD、混雑・ストレス耐性。
- アブレーション:  $\lambda, \beta$ 、重み設計、ヘッジ有無、ドライバ（本稿は  $R - r$  のみ）。

## 10 リスク管理と市場中立

- $\beta$  ニュートラル: ベンチマーク回帰で偏りを監視し、ヘッジ資産の追加やドライバに中立化項を付与。
- レバレッジ/証拠金: 予定注文の最悪ケース消費を前計算。
- レジーム変化: 横断分散がトレンド要因で支配される局面では  $\beta$  を抑制。

## 11 代表的な落とし穴

- タイムスタンプ非整合 ( $R_t$  と  $r_{i,t}$  のズレ)。
- 手数料過小見積もり (資金調達・借株・リベート)。
- 在庫ドリフト ( $\lambda$  が高すぎると  $\beta$  が滲む)。
- ストレス局面での「平均回帰の崩れ」。

## 12 結論

MNVH は「直近の横断的乖離に対して、数量で平均へ踏み込む」という最小ルールで、分散  $\times$  平均回帰－コストという単純な構造を、初期ゼロかつ自己資金で持続的に現実化する。Ledger-SPT は会計面の欠落を埋め、実運用での安全性と可監査性を担保する。

## 付録 A：初期ゼロ・自己資金を満たす擬似コード

```
INPUT: prices p[i], weights w[i], lambda, beta
STATE: cash C, positions q[i]=0 (for all i at t=0), ledger lots = {}

At each step t:
1) compute simple returns r[i] from p[i]->p_next[i]
2) R = sum_i w[i]*r[i]
3) propose q_star[i] = lambda*q[i] + beta*(w[i]/p[i])*(R - r[i])
   dq_star[i] = q_star[i] - q[i]
4) simulate fees phi[i](dq_star[i]); S = sum_i p[i]*dq_star[i] + sum_i phi[i]
5) if S ≠ 0:
   find kappa in (0,1] s.t. sum_i p[i]*(kappa*dq_star[i]) + sum_i phi_i(kappa)
   ) = 0
   dq[i] = kappa*dq_star[i]
else:
   dq[i] = dq_star[i]
```

```
6) place orders dq[i]; update ledger and cash; set q[i] <- q[i] + filled(dq[i])
7) risk checks (beta, leverage, drawdown); logs (MTM, trade cashflow, fees)
```