Applied Quantitative Investment Management

Lecture 4: Resampling and Generative Machine Learning

Anton Vorobets

Agenda

- Market states, structural breaks, and time conditioning (Section 1.2)
- Entropy Pooling introduction (pages 70-73 in Section 5.1)
- Projection of stationary transformations (Section 3.2)
  - Fully Flexible Resampling method (an instance of the Time- and Stat-Dependent Resampling class)
  - Time series variational autoencoders (VAEs) and generative adversarial networks (GANs)
  - Perspectives on no arbitrage and stochastic differential equations
- Better backtesting of CVaR versus variance optimization (Section 3.5)

Market states

- Imagine a coin with heads probability p and tails probability $1-p$
- "Fair" coin corresponds to $p = 0.5$
- Biased coin can, for example, be $p = 0.1$ or $p = 0.9$
- Imagine that we have some discrete time period $t = 1,2,\ldots,T$ where the heads probability changes between the three possible outcomes
- We call the heads probability the "market state"

Market states transition probabilities

Markov chain transition probability matrix:

$$\mathcal{T} = \begin{pmatrix} 0.9 & 0.1 & 0.0 \\ 0.1 & 0.8 & 0.1 \\ 0.0 & 0.1 & 0.9 \end{pmatrix}$$

- **Important realization:** stochasticity in both the state and the outcome
- **Coin example:** state is the heads probability, outcome is heads or tails
- **In reality:** complex market states, for example, combination of VIX and interest rates as well as complex risk factor and return distributions
- **Conclusion:** same concepts but higher complexity in investment markets

Structural breaks

Definition: Any change to the state transition probabilities our outcome distributions

Time conditioning

Definition: A method for capturing residual market state by assigning less importance to older data than newer data

Investment modeling summary

We must be good at estimating both the market state and the corresponding joint outcome distribution to be successful

Entropy Pooling intro

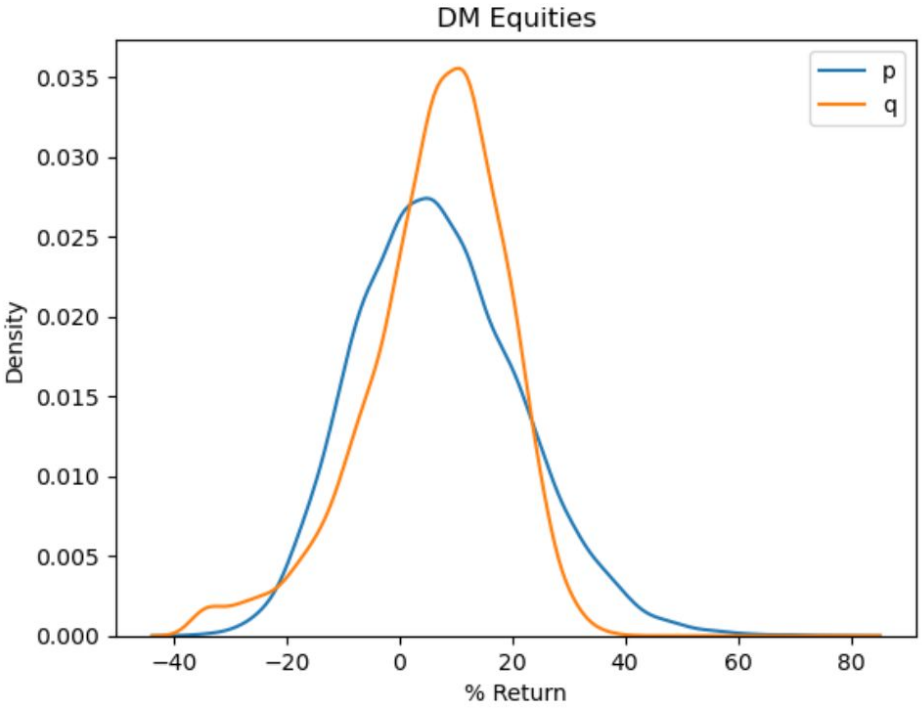
$$R = \begin{pmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,I} \\ R_{2,1} & R_{2,2} & \cdots & R_{2,I} \\ \vdots & \vdots & \ddots & \vdots \\ R_{S,1} & R_{S,2} & \cdots & R_{S,I} \end{pmatrix} \quad p = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_S \end{pmatrix} \quad q = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_S \end{pmatrix}$$

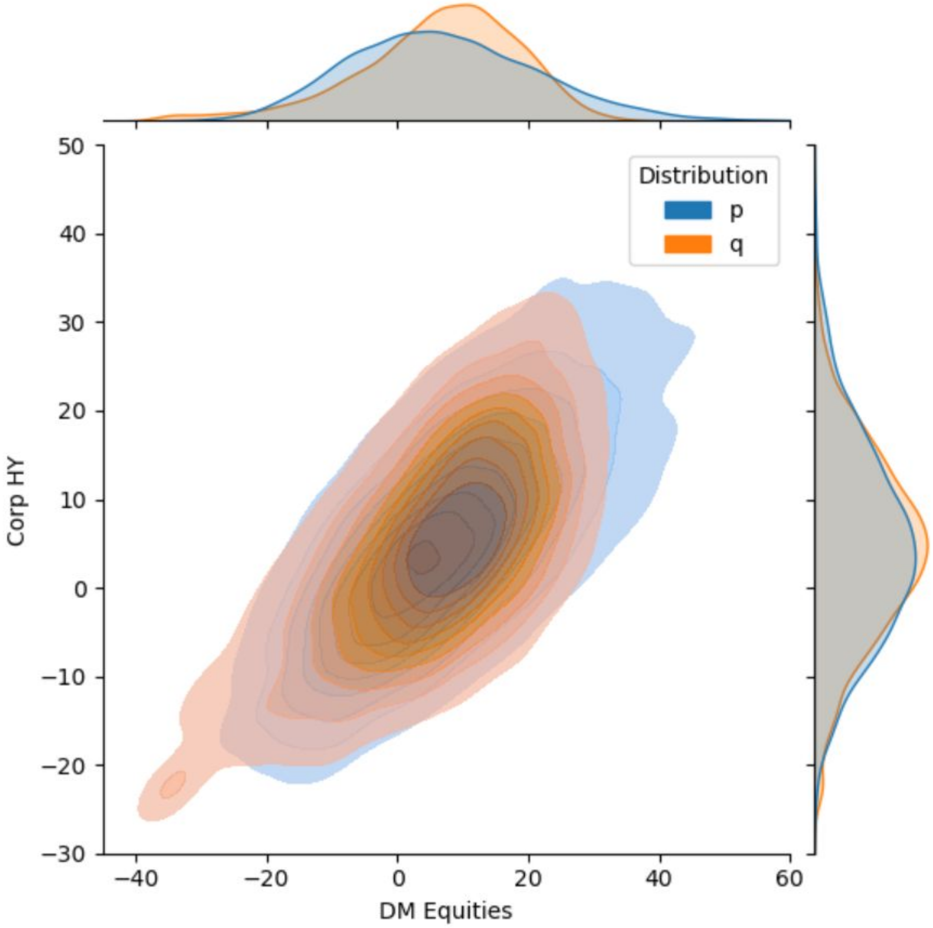
$$q = \underset{x}{\operatorname{argmin}} \left\{ x^T \left(\ln x - \ln p \right) \right\}$$

$$\text{s.t.} \quad Gx \leq h \quad Ax = b$$

$$x > 0 \quad \sum_{s=1}^{S} x_s = 1$$

Quick Entropy Pooling example

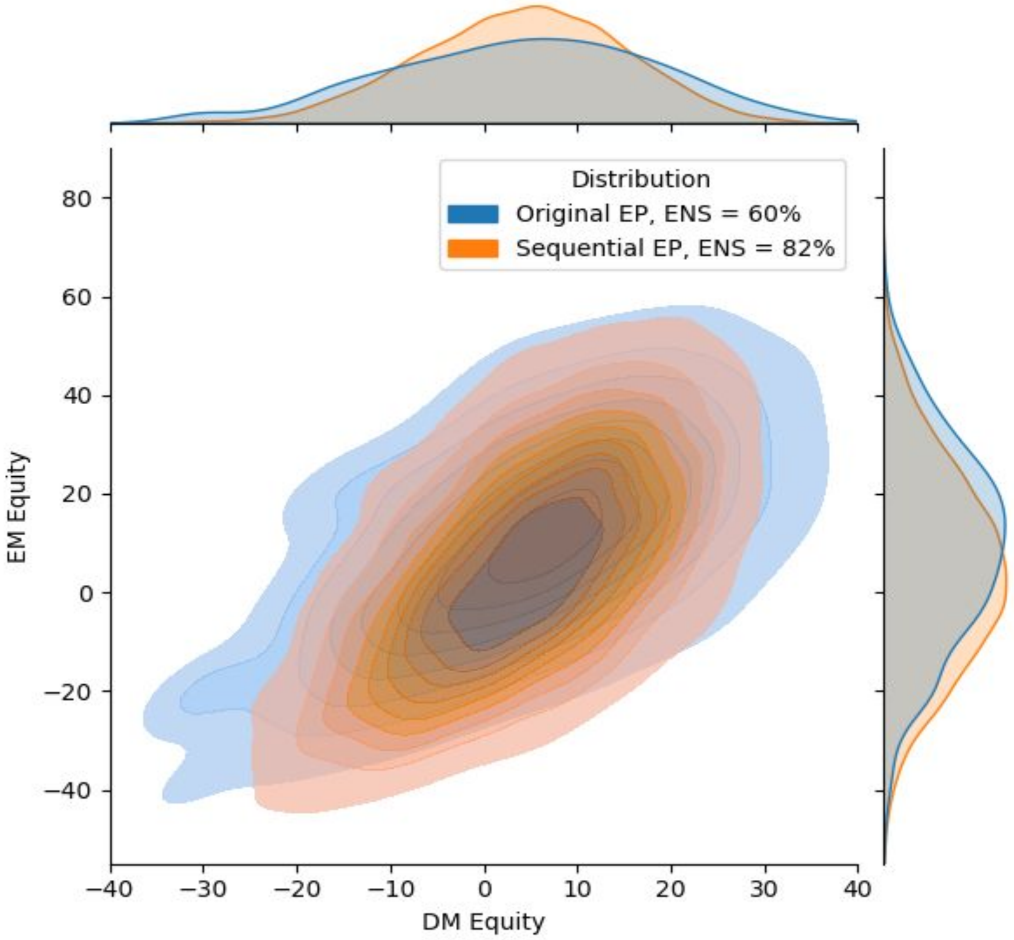




Effective number of scenarios (ENS)

A measure of scenario probability concentration

$$\hat{S} = \exp\left\{-\sum_{s=1}^{S} q_s \ln q_s\right\}$$



Time- and state-conditioning fundamentals

Time conditioning

$$p_t^{exp} \propto e^{-\frac{\ln 2}{\tau}(\tilde{T}-t)}$$

State conditioning

$$p_t^{crisp} \propto \begin{cases} 1 & \text{if } z_t \in R(z^\star) \\ 0 & \text{otherwise.} \end{cases}$$

Symmetric range definition

$$\sum_{\{t|z_t \in [\underline{z}, z^\star]\}} p_t = \frac{\alpha}{2} = \sum_{\{t|z_t \in [z^\star, \bar{z}]\}} p_t$$

Fully Flexible Resampling

$$R\left(z_j^{\star}\right) = \begin{cases} z_t \leq v_j & \text{for } j = 1, \\ v_{j-1} < z_t \leq v_j & \text{for } j = 2, \ldots, J-1 \\ v_{j-1} < z_t & \text{for } j = J. \end{cases}$$

Entropy Pooling views:

$$\sum_{t=1}^{\tilde{T}} x_t z_t = \mu_j,$$

$$\sum_{t=1}^{\tilde{T}} x_t z_t^2 \leq \mu_j^2 + \sigma_j^2$$

Right-hand side (RHS) values:

$$\mu_j = \sum_{t \in \left\{t \mid z_t \in R\left(z_j^{\star}\right)\right\}} p_t^{crisp} z_t,$$

$$\sigma_j^2 = \sum_{t \in \left\{t \mid z_t \in R\left(z_j^{\star}\right)\right\}} p_t^{crisp} z_t^2 - \mu_j^2$$

Fully Flexible Resampling procedure

With an initial state $j = j_0$ and the probability vectors q_j at hand, we can generate S paths using the Fully Flexible Resampling method with following procedure for each $s \in \{1, 2, \ldots, S\}$:

1. Sample a historical scenario $t \in \left\{1, 2, \ldots, \tilde{T}\right\}$ according to the scenario probabilities q_j.

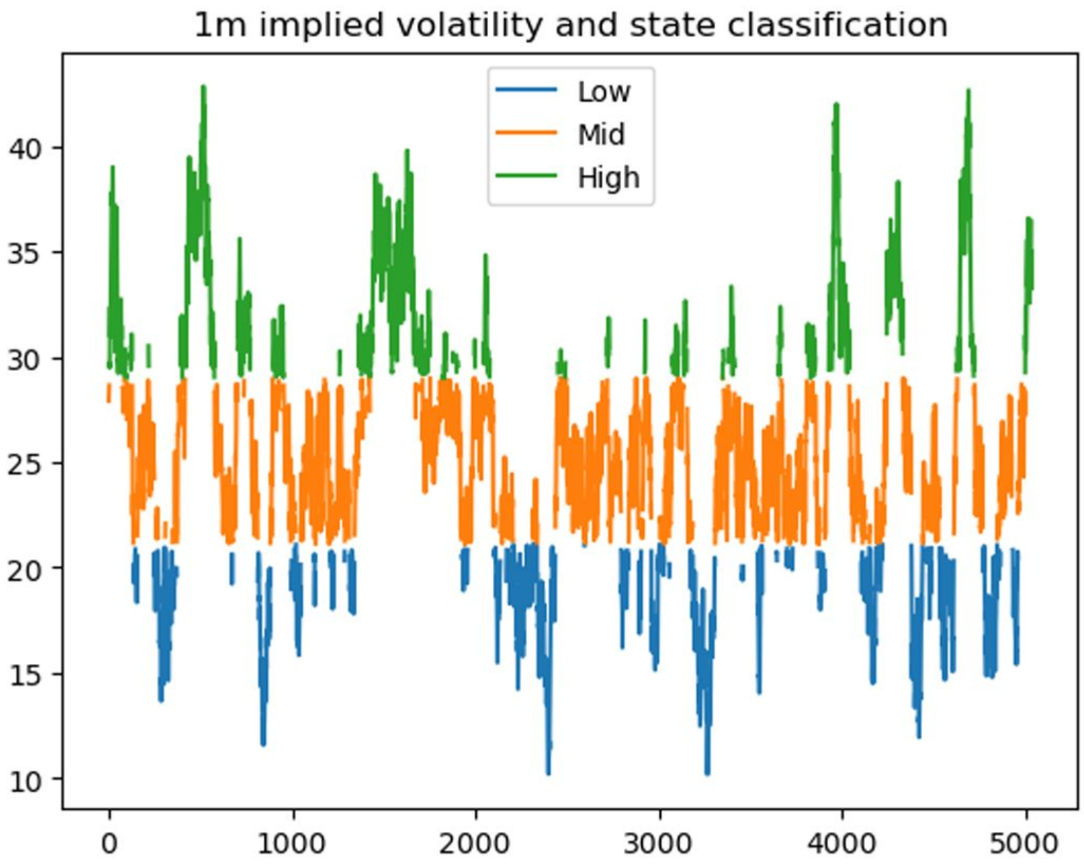
2. Update the state j, so it corresponds to the state of the historical scenario t from 1.

3. Repeat 1. and 2. H times.

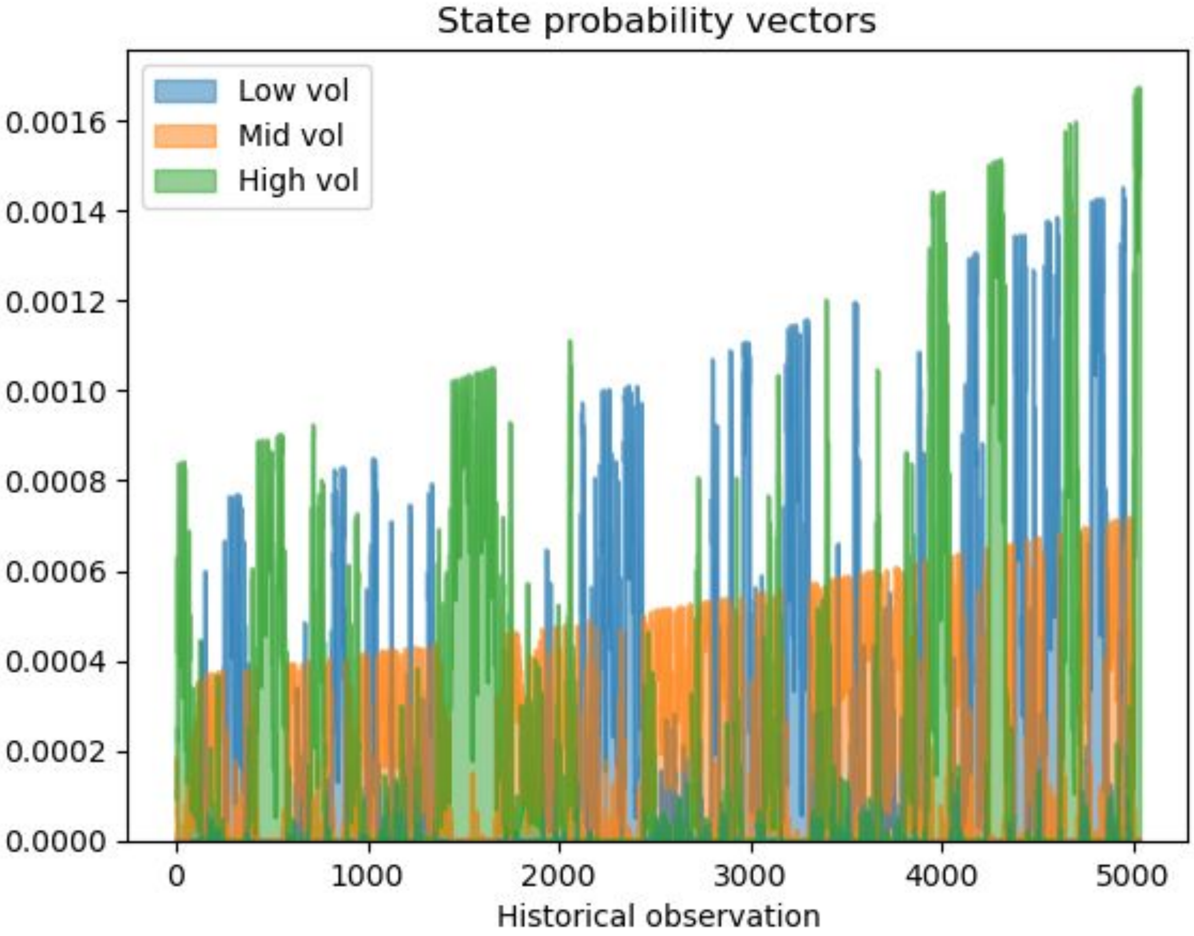
NOTE: The Markov chain might not be immediately obvious, because it is implicit.

State transition probability matrix can be computed based on the posterior vectors if desired.

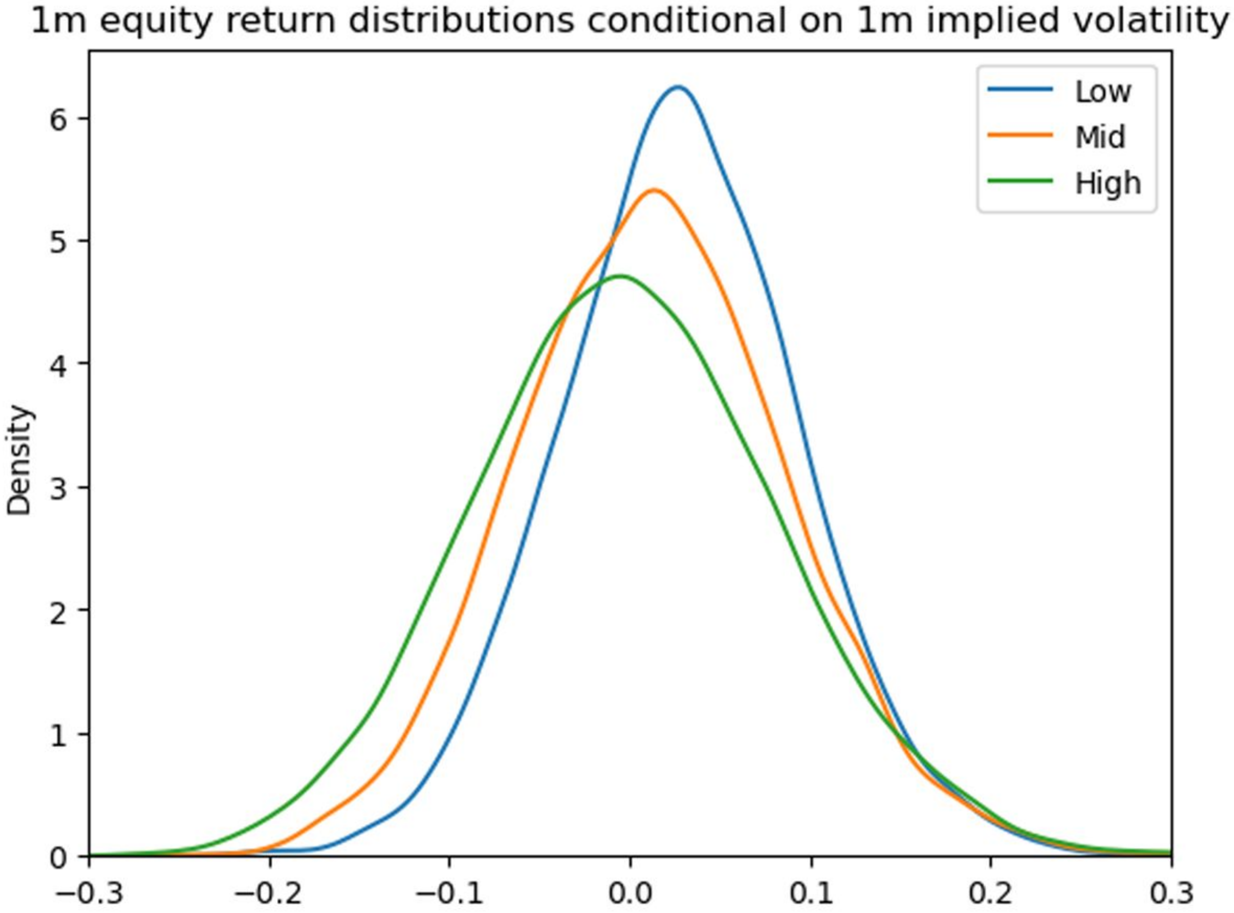
Fully Flexible Resampling



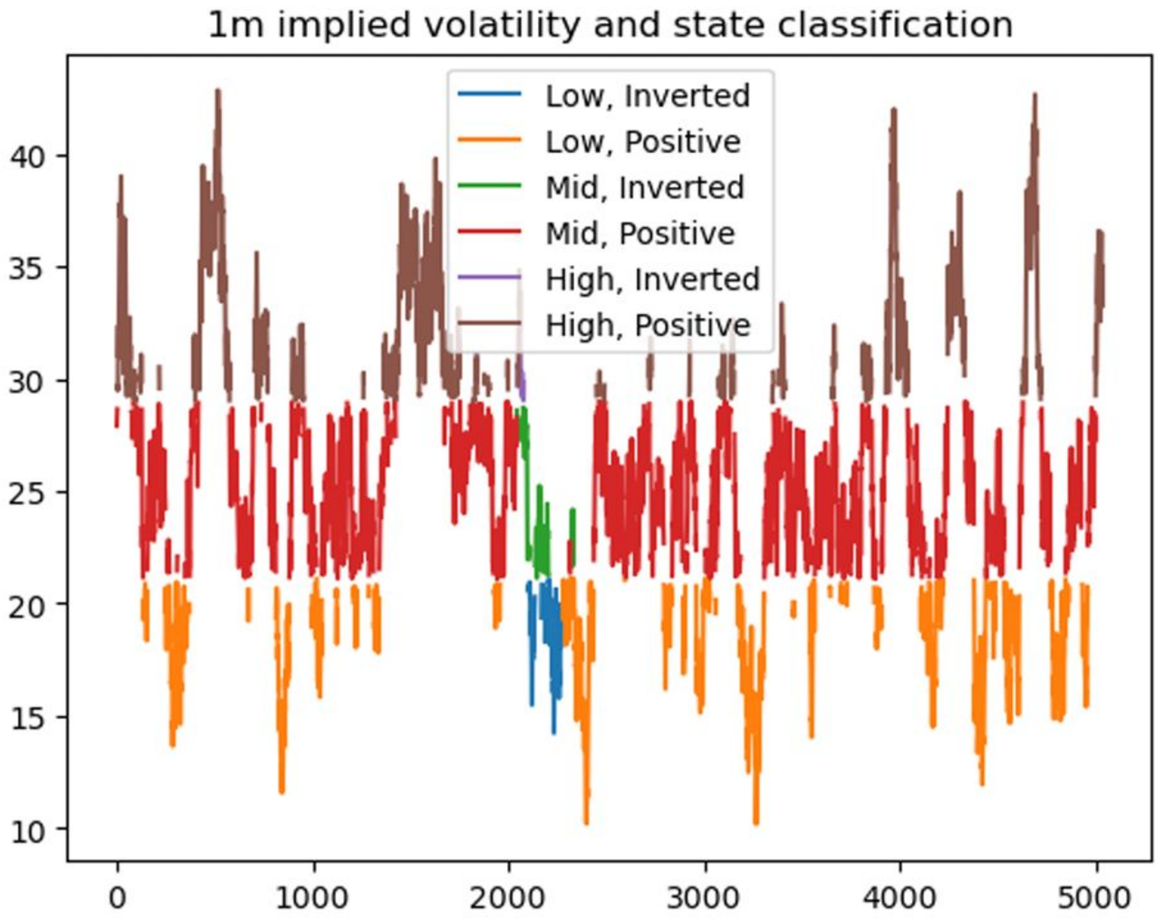
Fully Flexible Resampling



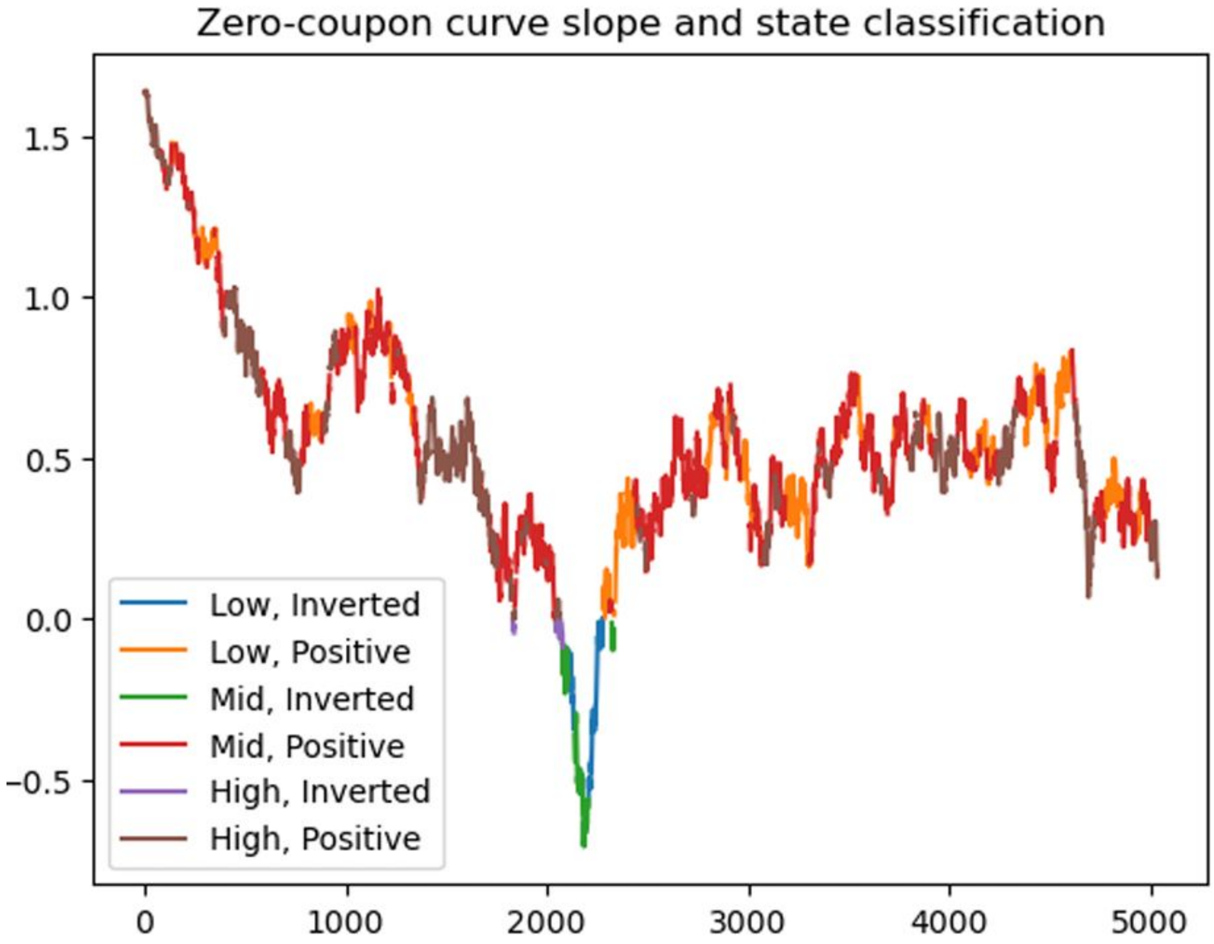
Fully Flexible Resampling



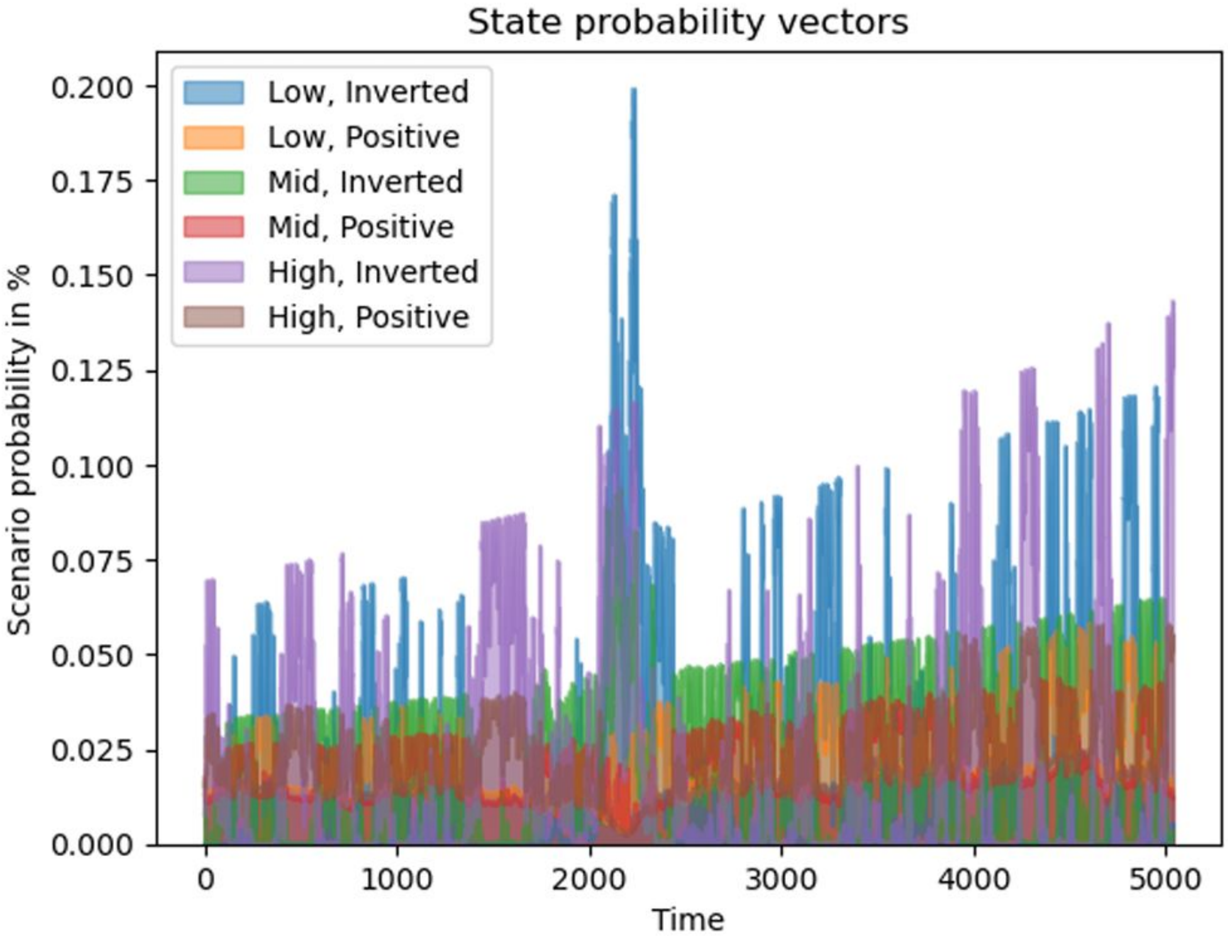
Multiple state variables



Multiple state variables



Multiple state variables



Multiple state variables formulas

$$R\left(z^{\star}_{i,m}\right) = \begin{cases} z_{t,m} \le v_{i,m} & \text{for } i = 1, \\ v_{i-1,m} < z_{t,m} \le v_{i,m} & \text{for } i = 2, \ldots, I_m - 1 \\ v_{i-1,m} < z_{t,m} & \text{for } i = I_m. \end{cases}$$

$$q_j = \sum_{m=1}^{M} w_{i_m,m} q_{i_m,m} \quad \text{for } j \in \{1, 2, \ldots, J\}$$

$$b_{m,\tilde{m}} = \sum_t^{\tilde{T}} \left(q_{t,i_m,m} q_{t,i_{\tilde{m}},\tilde{m}}\right)^{1/2}$$

$$w_{i_m,m} = \frac{ENS_{i_m,m} D_{i_m,m}}{\sum_{m=1}^{M} ENS_{i_m,m} D_{i_m,m}}$$

$$d_{m,\tilde{m}} = \sqrt{1 - b_{m,\tilde{m}}}$$

$$D_{i_m,m} = \frac{1}{M-1} \sum_{\tilde{m} \neq m} d_{m,\tilde{m}}$$

Time- and State-Dependent Resampling

- The Fully Flexible Resampling method belongs to the **Time- and State-Dependent Resampling class**

- Under certain (mild and natural) conditions, the resampling procedure produces strictly stationary simulations

- All proofs given by Kristensen and Vorobets (2025): https://ssrn.com/abstract=5117589

Generative machine learning

- Resampling methods are very capable of capturing cross-sectional dependencies, no matter how complex they are
- However, time series dependencies are more challenging for resampling methods, which is why we compute stationary transformations, perform filtering, and use time- and state-dependent methods
- **Alternative:** Generative machine learning methods like variational autoencoders (VAEs) and generative adversarial networks (GANs)

PCA, AEs and VAEs

PCA:

$$F = \bar{D}W \in \mathbb{R}^{T \times N} \qquad FW^{-1} = \bar{D} \in \mathbb{R}^{T \times N}$$

Autoencoders (AEs):

$$f(D) = F \in \mathbb{R}^{T \times \tilde{N}} \qquad g(F) = \tilde{D} \in \mathbb{R}^{T \times N}$$

Variational autoencoders (VAEs):

$$f(D) = F \sim \mathcal{N}\left(\mu, \mathrm{diag}\left(\sigma^2\right)\right)$$

Generative adversarial networks (GANs)

- Generator

$$\mathcal{G}(z) \qquad\qquad z \sim \mathcal{N}(0, \mathrm{diag}(1))$$

- Discriminator

$$\mathcal{D}(\mathcal{G}(z), D)$$

- **Objective:** the generator becomes so good at generating synthetic data that the discriminator cannot distinguish between synthetic and real data
- **Caveat:** can be hard to train, e.g., requiring minibatch discrimination

From IID Gaussian noise to time dependence

- Consider the AR(1) process:

$$X_t = \varphi_0 + \varphi_1 X_{t-1} + \varepsilon_t$$

$$\varepsilon_t \overset{iid}{\sim} \mathcal{N}\left(0, \sigma^2\right)$$

- VAEs and GANs are very capable of generating time dependent data, but current deep learning frameworks make it difficult to do so for tabular time series

Perspectives on no arbitrage and SDEs

- Consider a stochastic volatility model:

$$dX_t = \mu X_t dt + \sqrt{v_t} X_t dW_t$$

$$dv_t = \alpha_t dt + \beta_t dB_t$$

- Excellent for guaranteeing no arbitrage, but very limited when it comes to capturing the dynamics of high-dimensional markets
- Suitable for market makers that want to ensure that the prices they quote do not allow for arbitrage but probably not for investors

Market simulation summary

- We use methods that work well for (approximately) stationary data and directly simulate the stationary transformations
- Using simulated stationary transformations, we compute simulated risk factors that we can use for instrument and strategy pricing (Chapter 4)
- Resampling methods are very capable of capturing cross-sectional dependencies but require more work to capture the time series dependencies
- Generative machine learning methods are very capable of capturing time series dependencies but suffer from curse of dimensionality

Better backtesting

- Historical backtesting suffers from having only one path
- Equivalent to making distributional inference based on one observation
- We can use the synthetic paths to validate our strategies on more paths that have similar characteristics to the historical and gain more confidence
- See also:

  https://antonvorobets.substack.com/p/naive-backtesting

  https://antonvorobets.substack.com/p/better-backtesting

References

- Meucci, A (2012). Effective Number of Scenarios in Fully Flexible Probabilities
- Meucci, A. (2013). Estimation and Stress-Testing via Time- and Market-Conditional Flexible Probabilities
- Kristensen and Vorobets (2025). Time- and State-Dependent Resampling: https://ssrn.com/abstract=5117589