Sluggish news reactions

A combinatorial approach for synchronizing stock jumps

Nabil Bouamara

KU Leuven, VU Brussel

Joint work with:

Kris Boudt, Sébastien Laurent and Christopher J. Neely

CFE 2021

Motivation

High-frequency stock prices are noisy. Do stock prices react instantly to news? Do stock prices **jump** at the same time news is released?

If they do not, can we rearrange the stock **jumps** in time in a sensible way?

- We can, using combinatorics.

Is there any information in the **rearranged** observations? Are there any gains from doing so?

- It seems hard to believe, but we show that indeed there is.

Background

A difficult-to-interpret FOMC statement released at 14:15.

Board of Governors of the Federal Reserve System

The Federal Reserve, the central bank of the United States, provides the nation with a safe, flexible, and stable monetary and financial system.

September 18, 2007

FOMC statement

For immediate release

Share

The Federal Open Market Committee decided today to lower its target for the federal funds rate 50 basis points to 4-3/4 percent.

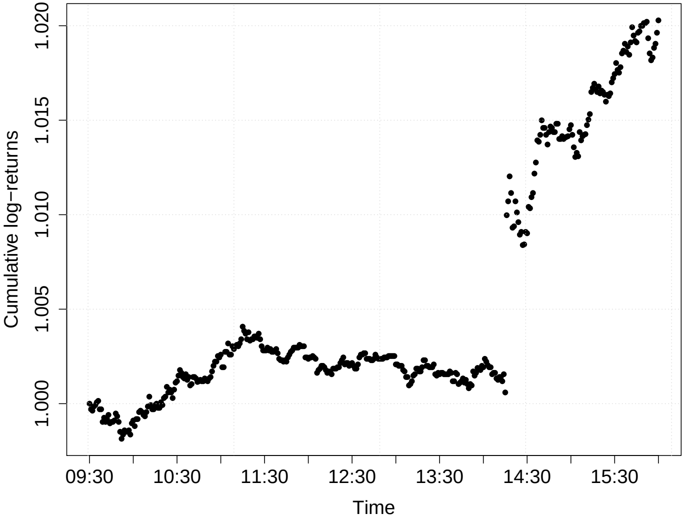
Economic growth was moderate during the first half of the year, but the tightening of credit conditions has the potential to intensify the housing correction and to restrain economic growth more generally. Today's action is intended to help forestall some of the adverse effects on the broader economy that might otherwise arise from the disruptions in financial markets and to promote moderate growth over time.

Readings on core inflation have improved modestly this year. However, the Committee judges that some inflation risks remain, and it will continue to monitor inflation developments carefully.

Developments in financial markets since the Committee's last regular meeting have increased the uncertainty surrounding the economic outlook. The Committee will continue to assess the effects of these and other developments on economic prospects and will act as needed to foster price stability and sustainable economic growth.

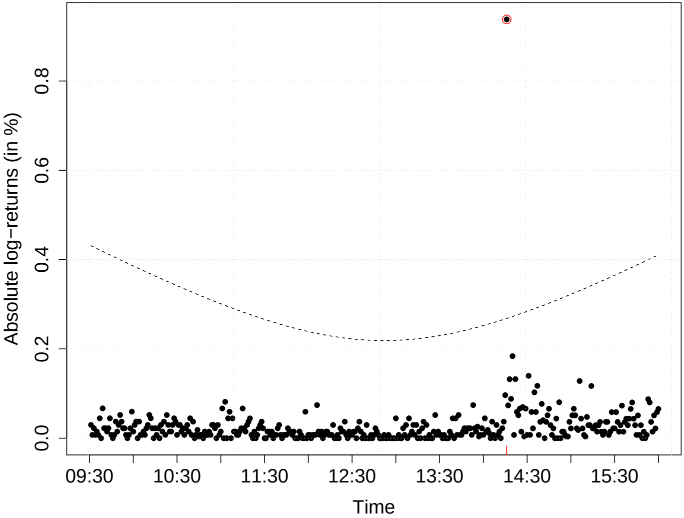
Background

One-minute prices of the DIA ETF on the same day.



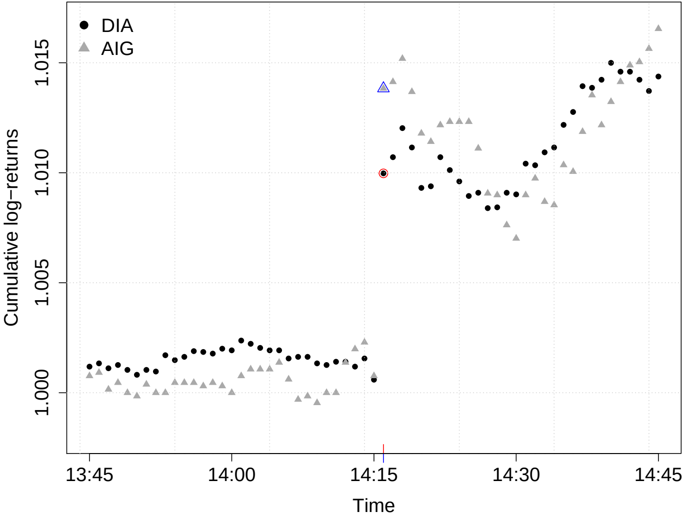
Background

Flagged jump at 14:16 using Mancini (2001)'s jump test.



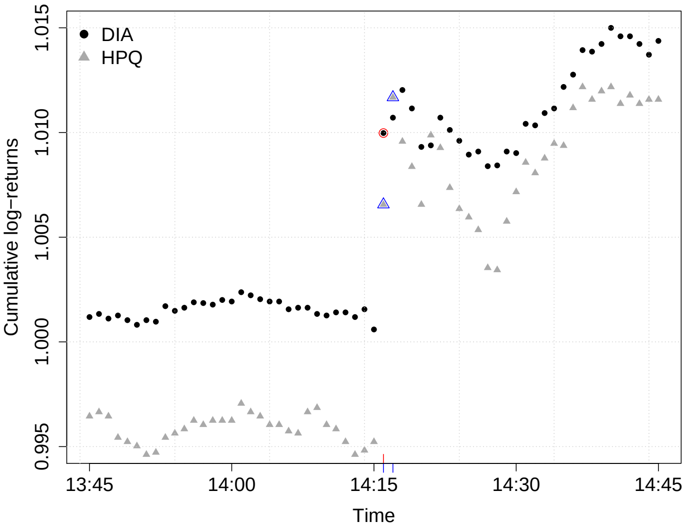
Background

Stocks jump at the same time as the ETF.



The Problem

We sometimes observe jumps at close but distinct points in time.



The main idea: Synchronizing stock jumps

We use the relationship between the jumps in the ETF and the corresponding cojumps in the underlying stocks.

We **synchronize** the stock jumps.

- We optimally **rearrange** the scattered jumps in time such that they hit some **target**.
- Let J_n be a jump-event matrix which reflects the relationship between the ETF and the stocks:

$$J_n = \left[\begin{array}{cccc} 0.000 & 0.000 & 0.000 & -0.002 \\ 0.000 & 0.000 & 0.000 & 0.003 \\ 0.000 & 0.000 & 0.000 & -0.807 \\ \underline{0.210} & 0.000 & \underline{0.400} & 0.004 \\ \underbrace{0.000 \quad \underline{0.217} \quad 0.000}_{\substack{\text{Weighted} \\ \text{stock jumps}}} & & & \underbrace{-0.028}_{\text{Target}} \end{array} \right] \text{ with row-sums } J_n^+ = \begin{bmatrix} -0.002 \\ 0.003 \\ -0.807 \\ 0.614 \\ 0.189 \end{bmatrix}.$$

The main idea: Synchronizing stock jumps

We use the relationship between the jumps in the ETF and the corresponding cojumps in the underlying stocks.

We **synchronize** the stock jumps.

- We optimally **rearrange** the scattered jumps in time such that they hit some **target**.
- Let J_n be a jump-event matrix which reflects the relationship between the ETF and the stocks:

$$J_n^{\pi} = \left[\underbrace{\begin{matrix} 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ \underline{0.210} & 0.000 & 0.000 \\ 0.000 & 0.000 & \underline{0.400} \\ 0.000 & \underline{0.217} & 0.000 \end{matrix}}_{\substack{\text{Rearr. weighted} \\ \text{stock jumps}}} \quad \underbrace{\begin{matrix} -0.002 \\ 0.003 \\ -0.807 \\ 0.004 \\ -0.028 \end{matrix}}_{\text{Target}} \right] \text{ with row-sums } J_n^{\pi,+} = \begin{bmatrix} -0.002 \\ 0.003 \\ -0.597 \\ 0.404 \\ 0.189 \end{bmatrix}.$$

The main idea: Synchronizing stock jumps

We use the relationship between the jumps in the ETF and the corresponding cojumps in the underlying stocks.

We **synchronize** the stock jumps.

- We optimally **rearrange** the scattered jumps in time such that they hit some **target**.
- Let J_n be a jump-event matrix which reflects the relationship between the ETF and the stocks:

$$J_n^{\pi} = \left[\underbrace{\begin{matrix} 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ \underline{0.210} & \underline{0.217} & \underline{0.400} \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{matrix}}_{\substack{\text{Rearr. weighted} \\ \text{stock jumps}}} \quad \underbrace{\begin{matrix} -0.002 \\ 0.003 \\ -0.807 \\ 0.004 \\ -0.028 \end{matrix}}_{\text{Target}} \right] \text{ with row-sums } J_n^{\pi,+} = \begin{bmatrix} -0.002 \\ 0.003 \\ 0.020 \\ 0.004 \\ -0.028 \end{bmatrix}.$$

Related work

- **Jumps and cojumps:** Lee and Mykland (RFS 2008, JE 2012); Bollerslev, Law and Tauchen (JE 2008); Bibinger and Winkelmann (JE 2015); Christensen, Oomen and Podolskij (JFE 2014); Li, Todorov, and Tauchen (ECMA 2017); and Li, Todorov, Tauchen, and Lin (JBES 2019).
- **Realized (semi)covariances and (semi)betas:** Bollerslev, Patton and Quaedvlieg (ECMA 2020, JFE 2021)
- **Sampling of asynchronous observations for covariances:** Hayashi and Yoshida (BERN, 2005); Aït-Sahalia, Fan, and Xiu (JASA 2010); Barndorff-Nielsen, Hansen, Lunde, and Shephard (JE 2011); Boudt *et al.* (JE 2017).
- **The rearrangement algorithm:** Puccetti and Rüschendorf (JCAM 2012); Embrechts, Puccetti, and Rüschendorf (JBF 2013); Bernard, Rüschendorf, and Vanduffel (JRI 2017); Bernard, Bondarenko, and Vanduffel (AOR 2018); Boudt, Jakobsons, and Vanduffel (4OR 2018).

A DGP for sluggish news reactions

The efficient stock price.

Econometricians (see *e.g.*, Aït-Sahalia and Jacod, Book 2014) model (efficient) stock prices $\boldsymbol{X}_t = (X_{1,t}, \ldots, X_{p,t})^\top$ as a **jump-diffusion**, which includes a continuous **Brownian** component and a **jump** component:

$$\boldsymbol{X}_t = \boldsymbol{X}_t^c + \boldsymbol{X}_t^d, \text{ with,}$$
$$\boldsymbol{X}_t^c \equiv \boldsymbol{X}_0 + \int_0^t \boldsymbol{b}_s ds + \int_0^t \boldsymbol{\sigma}_s d\boldsymbol{W}_s,$$
$$\boldsymbol{X}_t^d \equiv \textstyle\sum_{s \leq t} \Delta \boldsymbol{X}_s,$$

- $\boldsymbol{b}$ is the drift process,
- $\boldsymbol{\sigma}$ is the stochastic (co)volatility process,
- $\boldsymbol{W}$ is a multivariate Brownian motion
- $\Delta \boldsymbol{X}_t \equiv \boldsymbol{X}_t - \boldsymbol{X}_{t-}$, with $\boldsymbol{X}_{t-}$ the left limit at time t, denotes the jumps of $\boldsymbol{X}$ at time t.

A DGP for sluggish news reactions

The observed (contaminated) stock price.

We model the observed log-price process $\boldsymbol{Y}_t = (Y_{1,t}, \ldots, Y_{p,t})^\top$ of the p stocks as the sum of a contaminated **Brownian** component and a contaminated **jump** component:

$$\begin{aligned}
\boldsymbol{Y}_{i\Delta_n} &= \boldsymbol{Y}^c_{i\Delta_n} + \boldsymbol{Y}^d_{i\Delta_n}, \text{ with,} \\
\boldsymbol{Y}^c_{i\Delta_n} &\equiv \boldsymbol{X}^c_{i\Delta_n} + \boldsymbol{u}_{i\Delta_n} \\
\boldsymbol{Y}^d_{i\Delta_n} &\equiv \textstyle\sum_{h\Delta_n \leq i\Delta_n} \Delta \boldsymbol{Y}_{h\Delta_n}.
\end{aligned}$$

There are two kinds of noise.

- Microstructure noise $\boldsymbol{u}$ contaminates the efficient price process $\boldsymbol{X}$ (see *e.g.* Christensen, Oomen, and Podolskij, JFE 2014; Lee and Mykland, JE 2012), but is typically **too small** to contaminate the discontinuous part $\boldsymbol{X}^d$. It cannot generate gradual jumps, nor can it generate jump delays.
- A separate noisy jump component $\boldsymbol{Y}^d$ captures the mistimed or mismeasured jumps. It emulates a **sluggish news reaction** by spreading the jump across several time intervals.

A DGP for sluggish news reactions

We simulate prices to clarify the mechanics behind our rearrangement.

We draw the efficient jump process from a **compound Poisson process**:

$$X^d_{i\Delta_n} \equiv \sum_{j=1}^{N^J} I_{\{U_j \cdot T \leq i\Delta_n\}} \Delta X_j,$$

- $I(\cdot)$ is an indicator function
- N^J governs the total number of jumps that occur across a day,
- U_j are the random arrival times of the jumps
- the sequence of normally distributed random variables, $\Delta X_1, ..., \Delta X_{N^J}$, are the corresponding jump sizes.

A DGP for sluggish news reactions

We simulate prices to clarify the mechanics behind our rearrangement.

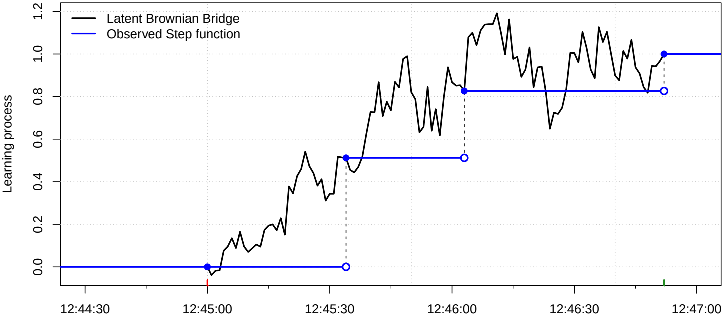
We contaminate the efficient jump process to emulate a **sluggish news reaction**:

$$Y^{d}_{i\Delta_n} \equiv \sum_{j=1}^{N^J} I_{\{U_j \cdot T \leq i\Delta_n\}} \sum_{d=0}^{N_j^D} I_{\{W_{j,d} \cdot T \leq i\Delta_n\}} \Delta L_{j,d} \, \Delta X_j,$$

- $I(\cdot)$ is an indicator function, N^J governs the total number of jumps that occur across a day, U_j are the random arrival times of the jumps, the sequence of normally distributed random variables, $\Delta X_1, ..., \Delta X_{N^J}$, are the corresponding jump sizes,
- for each jump j, **a step function spreads each jump size across multiple time intervals**;
  - N_j^D governs the total number of steps in the step function
  - $W_{j,0}, ..., W_{j,N_j^D}$ are the step arrival times
  - $\Delta L_{j,0}, ..., \Delta L_{j,N_j^D}$ are increments in the step sizes

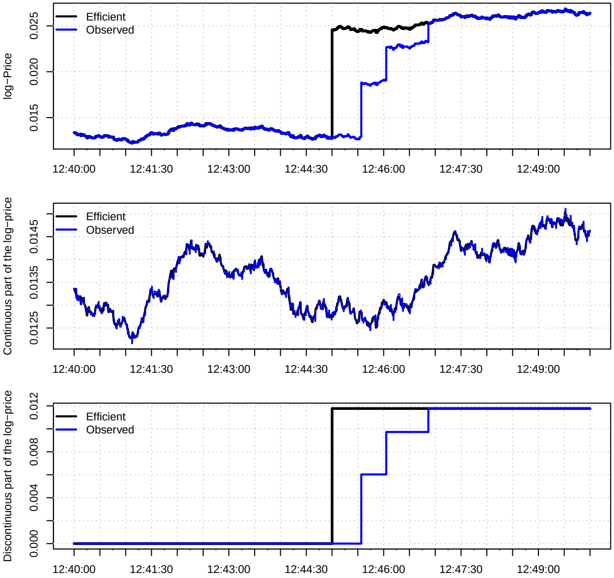
A DGP for sluggish news reactions

A learning step process captures the speed with which observed prices incorporate new information.

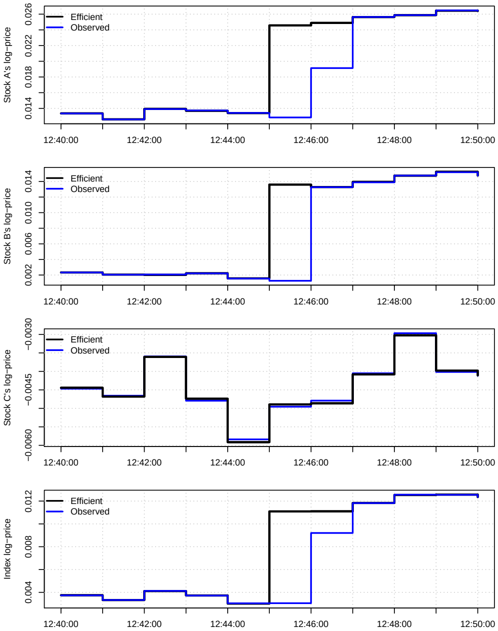


A DGP for sluggish news reactions

A stock's price reacts sluggishly to news



Stock prices vary in how quickly they impound new information



Price and return spreads

Relationship between jumps in the ETF and the corresponding cojumps in the underlying stocks.

- The **spread in prices** is the difference between the observed prices on a synthetic index of stocks $S_{i\Delta_n}$ and the prices of an observable ETF trading the index $Z_{i\Delta_n}$:

$$\delta^p_{i\Delta_n} := S_{i\Delta_n} - Z_{i\Delta_n}.$$

- The spread measures the **collective misalignment** of the observed stock prices with their effcient levels.
- Under our model assumptions, **sluggish news reactions drive the spread** in prices:

$$S_{i\Delta_n} - Z_{i\Delta_n} = \ldots = \sum_{k=1}^{p} w_{k,i\Delta_n} \underbrace{u_{k,i\Delta_n}}_{\substack{\text{Micro –}\\ \text{structure}\\ \text{noise}}} + \sum_{k=1}^{p} w_{k,i\Delta_n} \underbrace{\left(Y^d_{k,i\Delta_n} - X^d_{k,i\Delta_n}\right)}_{\substack{\text{The stock jump}\\ \text{which is not yet included}\\ \text{in the observed price}}}$$

The jump-event matrix

- **Isolate the jumps** in the return spread:

$$\delta^r_{i\Delta_n} := \Delta_i^n S - \Delta_i^n Z = \sum_{k=1}^{p} \underbrace{w_{k,i\Delta_n} \Delta_i^n J_k}_{\substack{\text{Weighted} \\ \text{stock jumps}}} + \underbrace{\sum_{k=1}^{p} \underbrace{w_{k,i\Delta_n} \Delta_i^n C_k}_{\substack{\text{Weighted} \\ \text{continuous} \\ \text{stock returns}}} - \underbrace{\Delta_i^n Z}_{\text{ETF returns}}}_{\text{Target}} .$$

The jump-event matrix

- **Isolate the jumps** in the return spread:

$$\delta^r_{i\Delta_n} := \Delta_i^n S - \Delta_i^n Z = \sum_{k=1}^{p} \underbrace{w_{k,i\Delta_n}\Delta_i^n J_k}_{\substack{\text{Weighted}\\ \text{stock jumps}}} + \underbrace{\sum_{k=1}^{p} \underbrace{w_{k,i\Delta_n}\Delta_i^n C_k}_{\substack{\text{Weighted}\\ \text{continuous}\\ \text{stock returns}}} - \underbrace{\Delta_i^n Z}_{\text{ETF returns}}}_{\text{Target}} .$$

- The **jump-event matrix** highlights how stock jumps influence the return spreads:

$$J_n = \left[\underbrace{\begin{matrix} 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ \underline{0.210} & 0.000 & \underline{0.400} \\ 0.000 & \underline{0.217} & 0.000 \end{matrix}}_{\substack{\text{Weighted}\\ \text{stock jumps}}} \quad \underbrace{\begin{matrix} -0.002 \\ 0.003 \\ -0.807 \\ 0.004 \\ -0.028 \end{matrix}}_{\text{Target}} \right] \text{ with row-sums } J_n^+ = \begin{bmatrix} -0.002 \\ 0.003 \\ -0.807 \\ 0.614 \\ 0.189 \end{bmatrix} .$$

Rearranging the elements within the jump-event matrix

- **Isolate the jumps** in the return spread:

$$\delta^r_{i\Delta_n} := \Delta_i^n S - \Delta_i^n Z = \sum_{k=1}^{p} \underbrace{w_{k,i\Delta_n} \Delta_i^n J_k}_{\substack{\text{Weighted}\\ \text{stock jumps}}} + \underbrace{\sum_{k=1}^{p} \underbrace{w_{k,i\Delta_n} \Delta_i^n C_k}_{\substack{\text{Weighted}\\ \text{continuous}\\ \text{stock returns}}} - \underbrace{\Delta_i^n Z}_{\text{ETF returns}}}_{\text{Target}} .$$

- The **jump-event matrix** highlights how stock jumps influence the return spreads:

$$J_n^{\pi} = \begin{bmatrix} \underbrace{\begin{matrix} 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ \underline{0.210} & 0.000 & 0.000 \\ 0.000 & 0.000 & \underline{0.400} \\ 0.000 & \underline{0.217} & 0.000 \end{matrix}}_{\substack{\text{Rearr. weighted}\\ \text{stock jumps}}} & \underbrace{\begin{matrix} -0.002 \\ 0.003 \\ -0.807 \\ 0.004 \\ -0.028 \end{matrix}}_{\text{Target}} \end{bmatrix} \text{ with row-sums } J_n^{\pi,+} = \begin{bmatrix} -0.002 \\ 0.003 \\ -0.597 \\ 0.404 \\ 0.189 \end{bmatrix} .$$

Rearranging the elements within the jump-event matrix

- **Isolate the jumps** in the return spread:

$$\delta^r_{i\Delta_n} := \Delta_i^n S - \Delta_i^n Z = \sum_{k=1}^{p} \underbrace{w_{k,i\Delta_n} \Delta_i^n J_k}_{\substack{\text{Weighted} \\ \text{stock jumps}}} + \underbrace{\sum_{k=1}^{p} \underbrace{w_{k,i\Delta_n} \Delta_i^n C_k}_{\substack{\text{Weighted} \\ \text{continuous} \\ \text{stock returns}}} - \underbrace{\Delta_i^n Z}_{\text{ETF returns}}}_{\text{Target}} \,.$$

- The **jump-event matrix** highlights how stock jumps influence the return spreads:

$$J_n^{\pi} = \begin{bmatrix} \underbrace{\begin{matrix} 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ \underline{0.210} & \underline{0.217} & \underline{0.400} \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{matrix}}_{\substack{\text{Rearr. weighted} \\ \text{stock jumps}}} & \underbrace{\begin{matrix} -0.002 \\ 0.003 \\ -0.807 \\ 0.004 \\ -0.028 \end{matrix}}_{\text{Target}} \end{bmatrix} \text{ with row-sums } J_n^{\pi,+} = \begin{bmatrix} -0.002 \\ 0.003 \\ 0.020 \\ 0.004 \\ -0.028 \end{bmatrix} .$$

Rearranging the elements within the jump-event matrix

The best rearrangement flattens the return spreads

The **combinatorial optimization problem** is as follows: Rearrange the elements in the jump-event matrix to minimize the variability of the row-sums $J_n^{\pi,+}$ of the rearranged matrix:

$$\min_{\pi} V\left(J_n^{\pi,+}\right)$$

- the row-sums $J_n^{\pi,+}$ are return spreads expressed as a function of the timing of stock jumps
- $V(\cdot)$ measures the variability of the vector of row-sums (*e.g.* variance, range)

How to find the best rearrangement?

The **combinatorial optimization problem** is as follows: Rearrange the elements in the jump-event matrix to minimize the variability of the row-sums $J_n^{\pi,+}$ of the rearranged matrix:

$$\min_{\pi} V\left(J_n^{\pi,+}\right)$$

We formulate a **linear program** to achieve the best rearrangement. Our program imposes penalites to prevent rearrangements that are economically implausible. It is rooted in the work of Puccetti and Rüschendorf (JCAM 2012) and Embrechts, Puccetti, and Rüschendorf (JBF 2013) on the rearrangement **algorithm**.

$$\begin{aligned} \text{Find a vector} \quad & \boldsymbol{x} \\ \text{that maximizes} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{x} \geq 0, \ldots \end{aligned}$$

Rearrangement Linear Program

Decision variable

We formulate a linear program to achieve the best rearrangement.

- Permutations as a decision variable using a **permutation matrix** for each column:

$$\pi_1 = \begin{pmatrix} 1 & 2 & \underline{4} & \underline{3} & 5 \end{pmatrix} \text{ is equivalent to } P_{\pi_1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \underline{1} & 0 \\ 0 & 0 & \underline{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Rearrangement Linear Program

Decision variable

We formulate a linear program to achieve the best rearrangement.

- Permutations as a decision variable using a **permutation matrix** for each column:

$$\pi_1 = \begin{pmatrix} 1 & 2 & \underline{4} & \underline{3} & 5 \end{pmatrix} \text{ is equivalent to } P_{\pi_1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \underline{1} & 0 \\ 0 & 0 & \underline{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- We rearrange multiple columns: Concatenate the permutation matrices in a **co-permutation** matrix:

$$\Pi = \left[\begin{array}{ccccc|ccccc|ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \underline{1} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \underline{1} & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Permutation matrices and the return spreads

$$\left[\begin{array}{ccccc|ccccc|ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \times \left[\begin{array}{r} 0.000 \\ 0.000 \\ 0.000 \\ 0.210 \\ 0.000 \\ \hline 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.217 \\ \hline 0.000 \\ 0.000 \\ 0.000 \\ 0.400 \\ 0.000 \\ \hline -0.002 \\ 0.003 \\ -0.807 \\ 0.004 \\ -0.028 \end{array}\right] = \left[\begin{array}{r} -0.002 \\ 0.003 \\ -0.807 \\ 0.614 \\ 0.189 \end{array}\right].$$

Permutation matrices and the return spreads

$$\left[\begin{array}{ccccc|ccccc|ccccc|ccccc} 1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0\\ 0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0\\ 0&0&0&\underline{1}&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0\\ 0&0&\underline{1}&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0\\ 0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1 \end{array}\right] \times \left[\begin{array}{r} 0.000\\ 0.000\\ 0.000\\ 0.210\\ 0.000\\ \hline 0.000\\ 0.000\\ 0.000\\ 0.000\\ 0.217\\ \hline 0.000\\ 0.000\\ 0.000\\ 0.400\\ 0.000\\ \hline -0.002\\ 0.003\\ -0.807\\ 0.004\\ -0.028 \end{array}\right] = \left[\begin{array}{r} -0.002\\ 0.003\\ -0.597\\ 0.404\\ 0.189 \end{array}\right].$$

Rearrangement Linear Program

Objective function

We formulate a **linear program** to achieve the best rearrangement.

- The sample **range** is the difference between the maximum and the minimum row-sum:

$$R(J_{n,1}^{\pi,+}, J_{n,2}^{\pi,+}, \dots, J_{n,h}^{\pi,+}) = J_{n,(h)}^{\pi,+} - J_{n,(1)}^{\pi,+}$$

Rearrangement Linear Program

Objective function

We formulate a **linear program** to achieve the best rearrangement.

- The sample **range** is the difference between the maximum and the minimum row-sum:

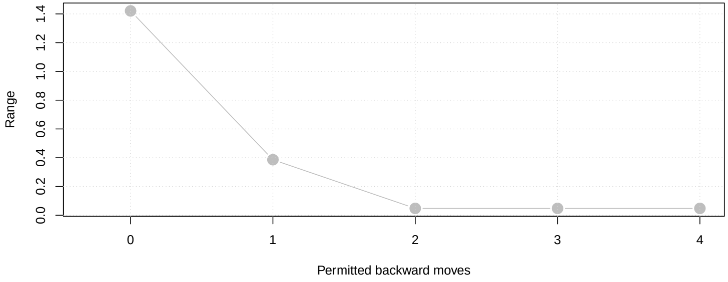
$$R(J_{n,1}^{\pi,+}, J_{n,2}^{\pi,+}, ..., J_{n,h}^{\pi,+}) = J_{n,(h)}^{\pi,+} - J_{n,(1)}^{\pi,+}$$

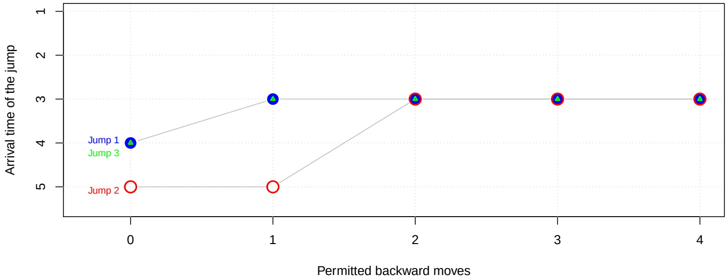
- **Minimize the range** of the return spreads:

$$\begin{aligned} \text{Find a vector} \quad & \left[\text{vecr}\left(\Pi\right), J_{n,(1)}^{\pi,+}, J_{n,(h)}^{\pi,+}\right] \\ \text{that minimizes} \quad & [0, 0, 0, ..., 0, -1, 1]^\top \times \left[\text{vecr}\left(\Pi\right), J_{n,(1)}^{\pi,+}, J_{n,(h)}^{\pi,+}\right] \\ \text{subject to} \quad & \text{some linear constraints on: row-sums, permutations,} \\ & \text{target, } \textbf{moves}, \textit{ etc.}) \end{aligned}$$

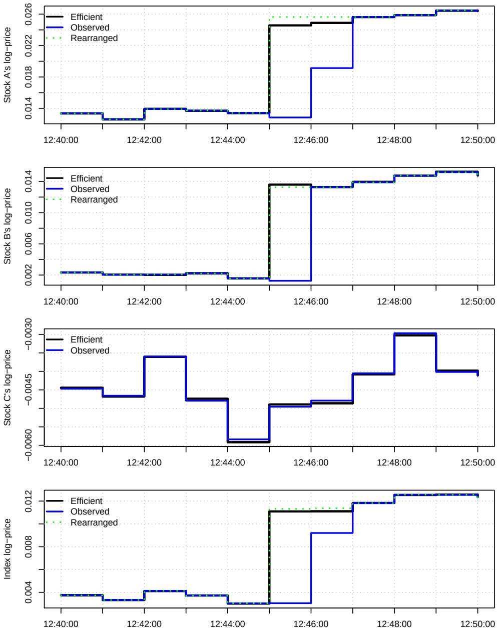
Rearrangement Linear Program

Range as a function of the permitted moves





The best rearrangement recovers the efficient stock jumps.



Et alors?

It may be of interest to look at **realized measures**.

- Realized variances and covariances shoot up after rearrangement:

$$\mathbf{\Sigma} = \begin{bmatrix} 2.386 & 0.901 & 0.108 \\ - & 3.035 & 0.045 \\ - & - & 1.751 \end{bmatrix} \xrightarrow{\text{Rearrangement}} \begin{bmatrix} 3.058 & 1.536 & 0.163 \\ - & 2.965 & 0.131 \\ - & - & 1.751 \end{bmatrix} = \mathbf{\Sigma}^{\pi}$$

- Realized betas also shoot up after rearrangement:

$$\boldsymbol{\beta} = \begin{bmatrix} 0.462 & 0.444 & 0.549 \end{bmatrix} \xrightarrow{\text{Rearrangement}} \begin{bmatrix} 1.258 & 1.225 & 0.549 \end{bmatrix} = \boldsymbol{\beta}^{\pi}$$

Main findings

The Treachery of Images

Main findings

The Treachery of Noisy Data

Sluggish news reactions manifest as gradual jumps and jump delays. These noisy jumps show up as a **sluggish cojump** in a panel of high-frequency intraday stock returns.

- We introduce the DGP to emulate sluggish news reactions.

We introduce the **tools** to synchronize stock jumps and recover the efficient common jump.

- Our rearrangement linear program protects against profligacy.