# Communication Avoiding Linear Model Inference for Irregularly Sampled Time Series with Long Range Dependencies

Francois Belletti

CS294/MATH270, UC Berkeley

# Outline



- Rewinding on time domain analytics
- Stochastic processes in the frequency domain
- Time series for “wild” data
- Cross-correlogram estimation via the frequency domain
- From scalable analytics to predictions

# Time domain analytics

## Rewinding on time domain analytics

- Synchronously observed process $(X_t)_{t \in \mathbb{Z}} \in \mathbb{R}^d$
  - Tolerate a few missing observations
- Second order stationarity:
  - $E(X_t) = \mu^X \in \mathbb{R}^d$ (constant)
  - $\gamma^X(t, h) = \text{Cov}(X_t, X_{t+h})$ is only a function of $h$
  - $h \to \gamma^X(h) \in \mathbb{R}^{dxd}$ is the **autocovariance** function.
- Want to estimate a linear time dependency model:
  - $X_t = A_1 X_{t-1} + A_2 X_{t-2} + \ldots + A_p X_{t-p} + \varepsilon_t$

# Copy based communication avoidance

## Rewinding on time domain analytics

- Cross-correlation or locally dependent likelihood based analysis can rely on simple padding strategies
  - Parallelism with respect to time axis
    - Only copy a necessary look ahead and look back region
  - Parallelism with respect to space
    - Parallel model calibration with predictor surrounded by "helper data" region

# Overlapping blocks for time axis parallelism

Rewinding on time domain analytics

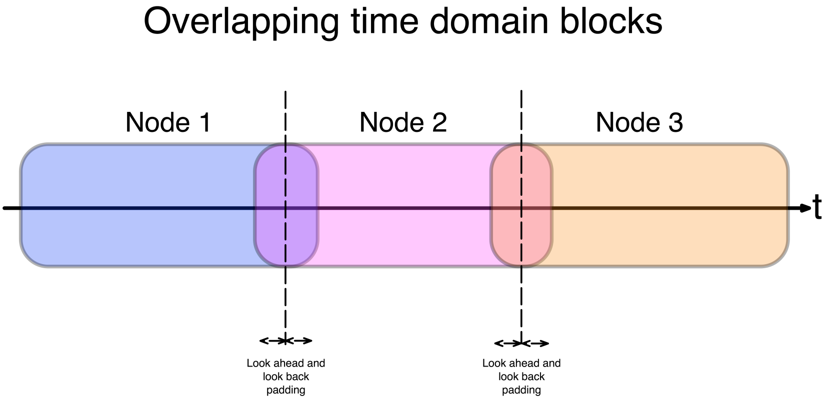



Figure: Overlapping blocks for short memory models

# Overlapping blocks for spatial domain parallelism

Rewinding on time domain analytics

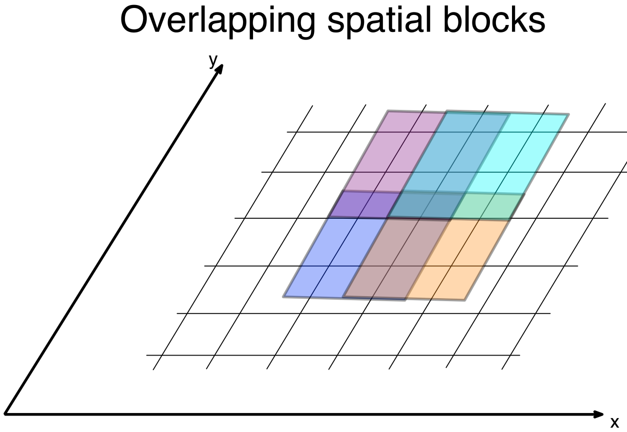



Figure: Overlapping blocks for local spatial dependencies

# Analyzing autoregressive models

## Rewinding on time domain analytics

- In discrete time:
  - Model is $X_t = A_1 X_{t-1} + A_2 X_{t-2} + \ldots + A_p X_{t-p} + \varepsilon_t$.
- If the process is second order stationary then
  - Autocovariance function is well defined: $\Gamma(h) = E\left(X_t X_{t-h}^T\right)$
  - $\widehat{\Gamma(h)} = \frac{1}{N-h-1} \sum_{n=1}^{N-h} X_t X_{t-h}^T$ is a consistent estimator of $\Gamma(h)$
- $LLR_{i \Rightarrow j}(X) = \frac{\sum_{h \geq 0} \Gamma_{i,j}(h)}{\sum_{h \leq 0} \Gamma_{i,j}(h)}$ reveals which of the $i$ and $j$ components of $X$ is a linear causator of the other
  - Causation here is understood in terms of ability to predict the future of a component based on the observation of the past of another

# Estimating autoregressive models

## Rewinding on time domain analytics

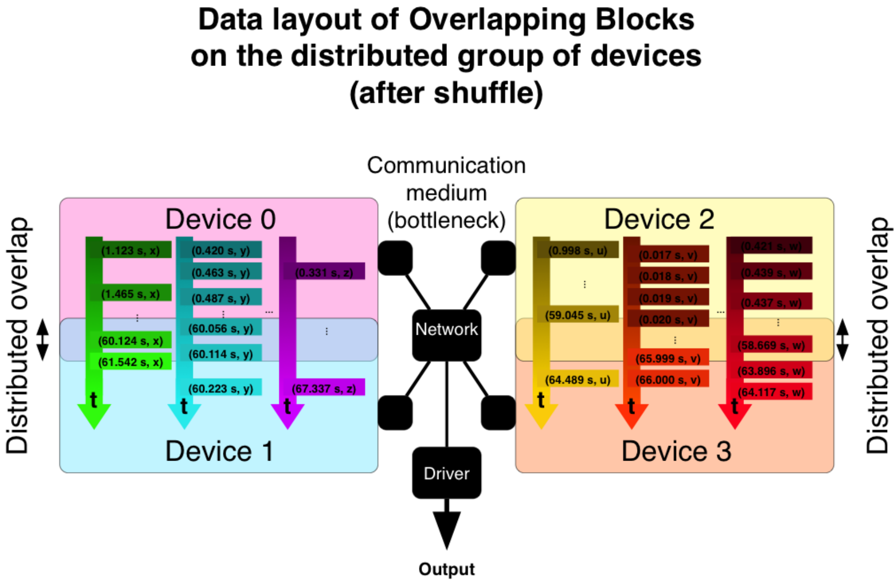
- Want to estimate the matrices $A_1$ ... $A_p$ in
  - $X_t = A_1 X_{t-1} + A_2 X_{t-2} + \ldots + A_p X_{t-p} + \varepsilon_t$.
- Yule-Walker equations:
  - $$\begin{bmatrix} \widehat{\Gamma(0)} & \widehat{\Gamma(1)} & \cdots & \widehat{\Gamma(p-1)} \\ \widehat{\Gamma(-1)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \widehat{\Gamma(1)} \\ \widehat{\Gamma(-(p-1))} & \cdots & \widehat{\Gamma(-1)} & \widehat{\Gamma(0)} \end{bmatrix} \begin{bmatrix} A_1^T \\ A_2^T \\ \vdots \\ A_p^T \end{bmatrix} = \begin{bmatrix} \widehat{\Gamma(1)} \\ \widehat{\Gamma(2)} \\ \vdots \\ \widehat{\Gamma(p)} \end{bmatrix}$$
- Solving this block Toeplitz system yields consistent estimators of the parameters of the model

# Analyzing regularly observed time series at scale

## Rewinding on time domain analytics

- How to we compute the estimator $\widehat{\Gamma(h)} = \frac{1}{N-h-1} \sum_{n=1}^{N-h} X_t X_{t-h}^T$ without shared memory?
  - By creating a padding of $h$ between computation node
- Distributed overlapping data structures:
  - Overlapping blocks

# Overlapping blocks for time axis parallelism



# Power spectrum of a time series

## Stochastic processes in the frequency domain

- Consider the Fourier transform of $(X_t)$: $\left(\widehat{X}_\lambda = \sum_{t=1}^{T} X_t e^{-2i\pi t\lambda}\right)$
- The power spectrum of a time series is the Fourier transform of its autocovariance function:
  - $I(\lambda) = \sum_{|h|<n} \gamma(h) e^{-2i\pi h\lambda}$ (it is a $d \times d$ matrix where $d$ is the dimension of the system).
- It is also the covariance of the Fourier transform of the signal
  - $I(\lambda) = \widehat{X}_\lambda \widehat{X}_\lambda^*$
- One can consider it is a signature of a multivariate time series

# Distribution of the power spectrum



- Consider a set of frequencies $\lambda_1, \ldots, \lambda_m$
- As the number of samples $T$ increases, $\widehat{I(\lambda_1)}, \ldots, \widehat{I(\lambda_m)}$ jointly converge in distribution toward independent random matrices
  - $\widehat{I(\lambda)} \sim W_k W_k^*$ where $W_k$ is a complex Gaussian variable with distribution $N_c(0, I(\lambda))$
- If $\lambda_1 \neq \lambda_2$, $Cov\left(\widehat{I_{pq}(\lambda_1)}, \widehat{I_{rs}(\lambda_2)}\right) = O\left(\frac{1}{n}\right)$

# Study in frequency domain as a form of compressing projection

## Stochastic processes in the frequency domain

- Cross correlation in time domain:

$$\gamma(h) = E\left(X_t X_{t+h}^T\right), h \in \{-H \dots H\}$$

- Power spectrum in frequency domain:

$$I(\lambda) = E\left(\widehat{X_\lambda}\widehat{X_\lambda}^T\right), \lambda \in \left[-\frac{1}{2}, \frac{1}{2}\right]$$

- The Fourier transform $\widehat{X_\lambda}$ is the result of the projection of $(X_t)$ onto the $\lambda^{th}$ element of the discrete Fourier basis, $P_\lambda(X_t)$.

$$I(\lambda) = E\left(P_\lambda(X_t) P_\lambda(X_t)^T\right), \lambda \in \left[-\frac{1}{2}, \frac{1}{2}\right]$$

- Sufficient information is contained in the series of projections $(P_\lambda(X_t))_{\lambda \in \left[-\frac{1}{2}, \frac{1}{2}\right]}$.
  - Compressed representation of process (on the driver)

# Exploratory data analysis in frequency domain

## Stochastic processes in the frequency domain

- Let $(x_t)$ and $(y_t)$ two univariate processes
  - We now study the series of correlations

$$I(\lambda) = E\left(P_\lambda\left(\begin{bmatrix} x_t \\ y_t \end{bmatrix}\right) P_\lambda\left(\begin{bmatrix} x_t \\ y_t \end{bmatrix}\right)^T\right) = \begin{pmatrix} I_{xx}(\lambda) & I_{xy}(\lambda) \\ I_{yx}(\lambda) & I_{yy}(\lambda) \end{pmatrix}$$

  - Two quantities of interest:
    - Coherency:

$$\frac{\left|I_{xy}(\lambda)\right|}{\sqrt{I_{xx}(\lambda) I_{yy}(\lambda)}}$$

    - Phase:

$$\text{angle}(I_{xy}(\lambda))$$

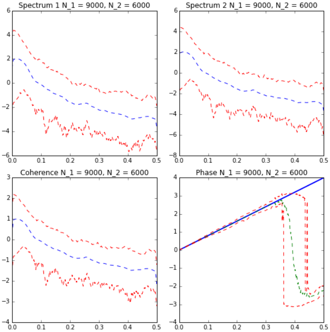
  - Detection of seasonality: clear spikes in the spectrum
  - Detection of lag: high coherency at all frequencies and phase increases linearly. Not really handy for a human...

# Example with two lagged signals with different sampling rates

## Stochastic processes in the frequency domain

Power spectrum
(covariances of Fourier transforms)

Cross-covariance
(Inverse Fourier transform)



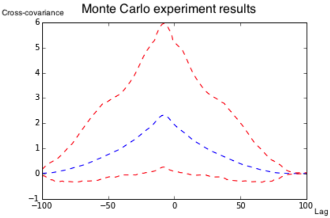




Figure: Frequency domain and time domain detection of lag between two correlated Brownian motions (lag $= -12$) with random time stamps. The first signal has 9000 samples. The second signal has 6000 samples.

- "Wild data"
  - Unsorted
  - Sampled at random (random timestamps)

Computing a Fourier transform is still trivial in that context with a map reduce operation:

$$P_\lambda(x_t) = \sum_t x_t e^{-2i\pi t\lambda}$$

$$P_\lambda(y_t) = \sum_t y_t e^{-2i\pi t\lambda}$$

$$\text{Cross Covariance}(x, y)_h = \frac{1}{K} \sum_{k=1}^{K} P_{\lambda_k}(x) P_{\lambda_k}(y)^* e^{-2\pi i h \lambda_k}$$

# Modern age for Time Series: big data

## Time series for “wild” data

- Data is scattered across a data center or collected by a distributed sensor network

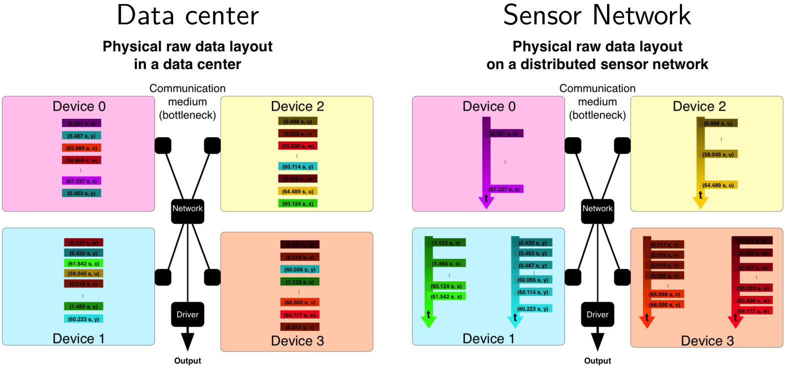



Figure: Distributed time series analysis

# Another issue with actual data

## Time series for “wild” data

- Let $(x_t)$ and $(y_t)$ two univariate independent Brownian motions.
  - Let us compute their cross-correlation naively
  - No bias, empirical average is 0
  - But variance is very high.
- Two methods to address the issue:
  - Differentiation $(\Delta x_t = x_t - x_{t-1})$, $(\Delta y_t = y_t - y_{t-1})$
  - Then compute cross-correlation
  - Needs sorted data.
  - Compute differentiation in frequency domain

$$\widehat{\Delta x}_f = \widehat{x_f} \times if$$

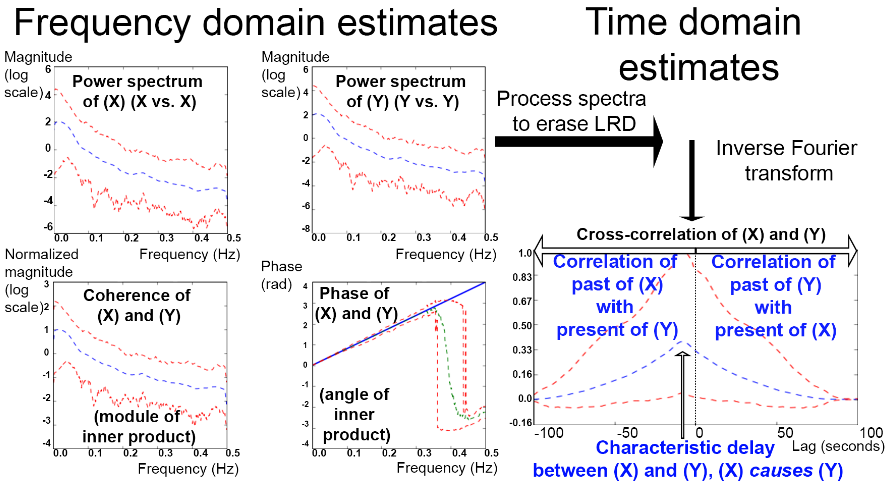
- Also valid for long range dependencies
  - Fractional differentiation would require complete shuffling of the data
  - $\Delta_\alpha x_t = F(x_t, x_{t-1}, \ldots, x_{-\infty})$

$$\widehat{\Delta_\alpha x_f} = \widehat{x_f} \times (if)^\alpha$$

# Going back to the time domain

## Cross-correlogram estimation via the frequency domain

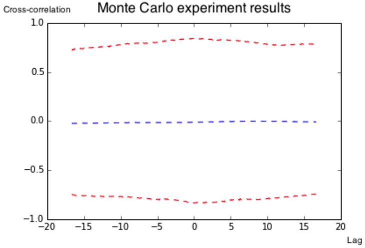
- For linear causality inference, estimating a cross-correlogram in time domain is most important



# Fractional differentiation in frequency domain, Monte Carlo experiments

## Cross-correlogram estimation via the frequency domain

No differentiation
Spurious Cross correlation




Fractional differentiation in freq. domain
No spurious cross correlation

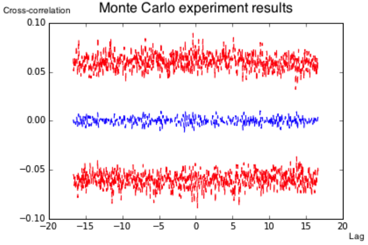



Figure: The red lines above indicate the 5% and 95% percentiles over the distribution of 1000 correlation computations with surrogate data. The blue lines indicate the average correlation. The first signal has 9998 samples and the second 6000. Independent signals with random irregular timestamps. Long range dependency with Hurst exponent 0.4. $\alpha = 0.9$.

# Number of projections and communication avoidance

## Cross-correlogram estimation via the frequency domain

- Compressing data with Fourier transforms is how we achieve scalability

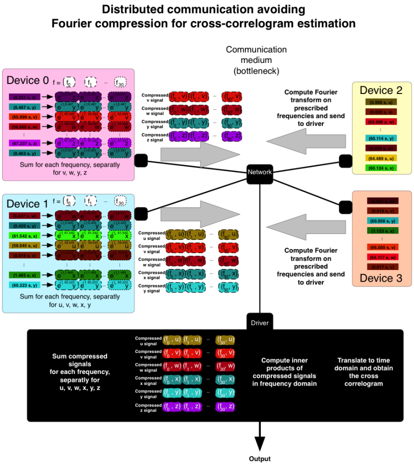



Figure: Communication needed $= O(\#\text{projections})$

# Number of projections and variance

## Cross-correlogram estimation via the frequency domain

- Compressing the data has a statistical cost we pay in terms of variance

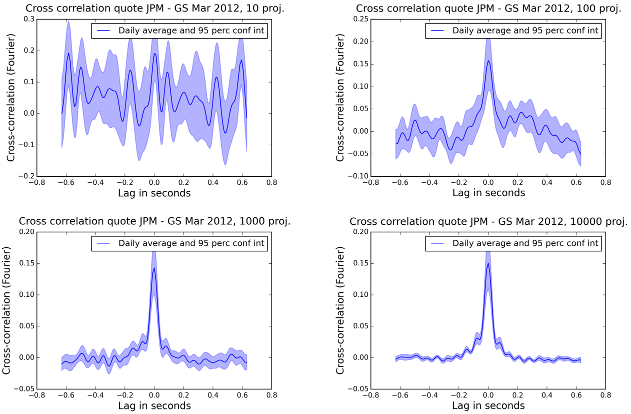
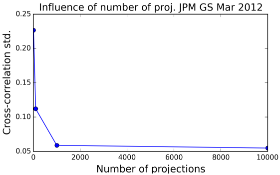



Figure: Empirical distributions of daily cross-correlogram of stock market price variations

# Achieving scalability...

## Cross-correlogram estimation via the frequency domain

- A few thousand projections are enough, small communication cost
- Communication time split up: 1 message of $10^3$ doubles (as compared to GB sized data set).



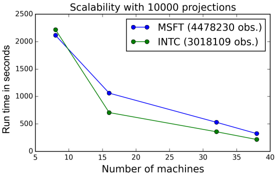




Figure: Fourier projections as a communication avoidance mechanism

# ....while achieving consistency

## Cross-correlogram estimation via the frequency domain

- A few thousand projections are enough, small communication cost

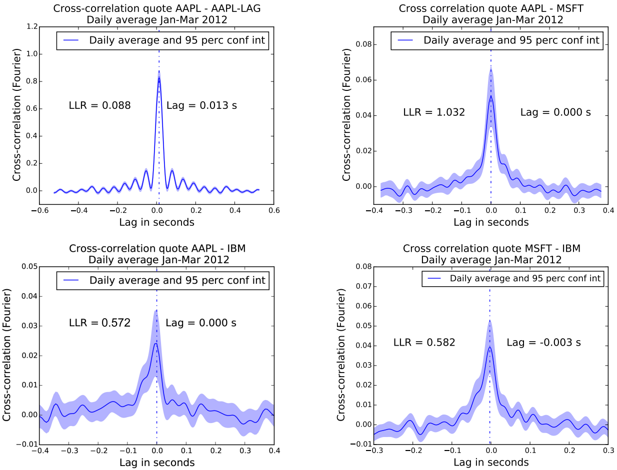



Figure: Empirical distributions of daily cross-correlogram of stock market price variations

# Inference of causality at scale

## Cross-correlogram estimation via the frequency domain

- A few thousand projections are enough, small communication cost

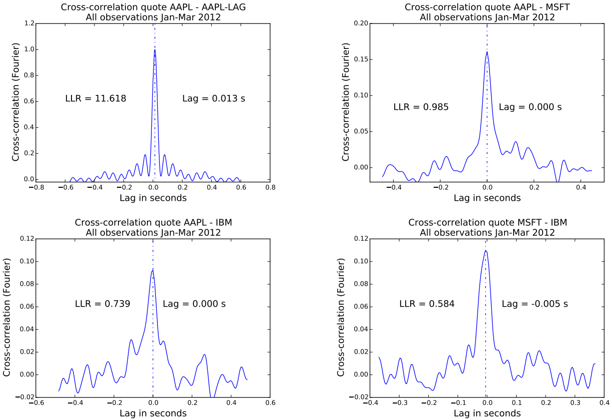



Figure: Empirical distributions of daily cross-correlogram of stock market price variations

# Back to the Yule-Walker equations

## Continuous model estimation

- We estimate
  - $\gamma_{xx}(0) = E(x_t x_t)$, $\gamma_{xx}(\Delta t) = E(x_t x_{t-\Delta t})$ , ... , $\gamma_{xx}(h\Delta t) = E(x_t x_{t-h\Delta t})$
  - $\gamma_{yy}(0) = E(y_t y_t)$, $\gamma_{yy}(\Delta t) = E(y_t y_{t-\Delta t})$ , ... , $\gamma_{yy}(h\Delta t) = E(y_t y_{t-h\Delta t})$
  - $\gamma_{xy}(0) = E(x_t y_t)$, $\gamma_{xy}(\Delta t) = E(x_t y_{t-\Delta t})$ , ... , $\gamma_{xy}(h\Delta t) = E(x_t y_{t-h\Delta t})$
  - $\Gamma(h) = \begin{bmatrix} \gamma_{xx}(h) & \gamma_{xy}(h) \\ \gamma_{xy}(-h) & \gamma_{yy}(h) \end{bmatrix}$
- We solve the corresponding Yule-Walker equations
  - $$\begin{bmatrix} \widehat{\Gamma(0)} & \widehat{\Gamma(1)} & \cdots & \widehat{\Gamma(p-1)} \\ \widehat{\Gamma(-1)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \widehat{\Gamma(1)} \\ \widehat{\Gamma(-(p-1))} & \cdots & \widehat{\Gamma(-1)} & \widehat{\Gamma(0)} \end{bmatrix} \begin{bmatrix} A_1^T \\ A_2^T \\ \vdots \\ A_p^T \end{bmatrix} = \begin{bmatrix} \widehat{\Gamma(1)} \\ \widehat{\Gamma(2)} \\ \vdots \\ \widehat{\Gamma(p)} \end{bmatrix}$$
- And what do we get? What model are we implicitly trying to infer?

# Continuous time autoregressive models

## Continuous model estimation

- Convolution type stochastic Volterra equations (cf. Anna Karczewska's monograph)
  - $X_t = X_0 + \int_{s=0}^{t} \phi(s) X_{t-s} ds + \int_{s=0}^{t} \sigma(s) dW_s,\ t > 0$
- We estimate the convolution kernel $\phi(s)$
- In cross-asset arbitrage,
  - $dy_t = (\phi \star dx)_t + \sigma(t) dW_t$
  - We estimate $\gamma_{xx}(h)$, $\gamma_{yy}(h)$ and $\gamma_{xy}(h)$ on a discrete grid
  - We solve the corresponding Yule-Walker equations
  - We get an estimate of $\phi(h)$ on the same discrete grid

# Inference of convolution kernel at scale

## Continuous model estimation

- 1000 projections are enough, small communication cost

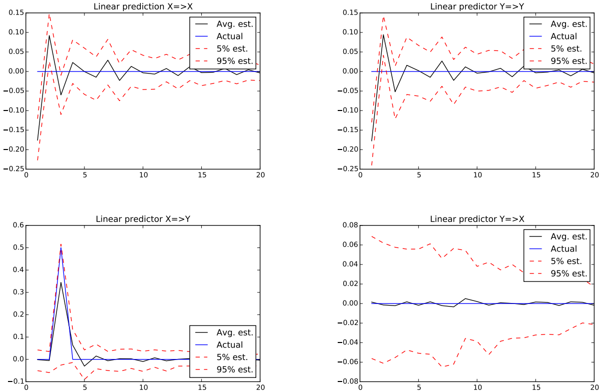



Figure: Empirical distributions of kernel estimates with correlated and lagged Brownian motions observed at random asynchronously, Fourier transforms

# Inference with Hayashi-Yoshida estimator

## Continuous model estimation

- Cross-correlation between randomly observed processes is estimates with another (less scalable) method

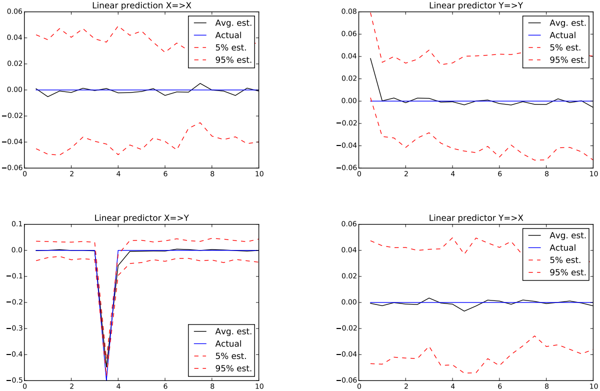



Figure: Empirical distributions of kernel estimates with correlated and lagged Brownian motions observed at random asynchronously, Hayashi-Yoshida

# Turning a convolution equation into a predictive tool

## Continuous model estimations

- Reinject observed values of $X$ into
  - $X_t = X_0 + \int_{s=0}^{t} \phi(s) X_{t-s} ds + \int_{s=0}^{t} \sigma(s) dW_s,\ t > 0$
- Interpolate the kernel, not the process

# Conclusion

## Wrapping up

- 3 main issues with actual time series data:
  - Distributed in a partitioned memory
  - Long memory
  - Irregularly spaced asynchronous timestamps
- We addressed them:
  - We estimated $\phi$ although consistent estimators are only available when considering the cross-correlogram of (unobserved) increments
  - We overcame the irregular sampling and the long memory issue in a single step thanks to frequency domain analytics
  - This method scales trivially.