Conquering Big Data in Volatility Inference and Risk Management

Jian (Frank) Zou

Worcester Polytechnic Institute

Volatility Modeling and Estimation

- Volatility is the conditional variance of the asset price.
- Volatility modeling is concerned with studying the evolution of the volatility over time.
- Critical role in finance.

Examples

- Portfolio allocation;
- Derivative pricing and hedging;
- Risk management using measures like VaR.

Low-Frequency Model Features

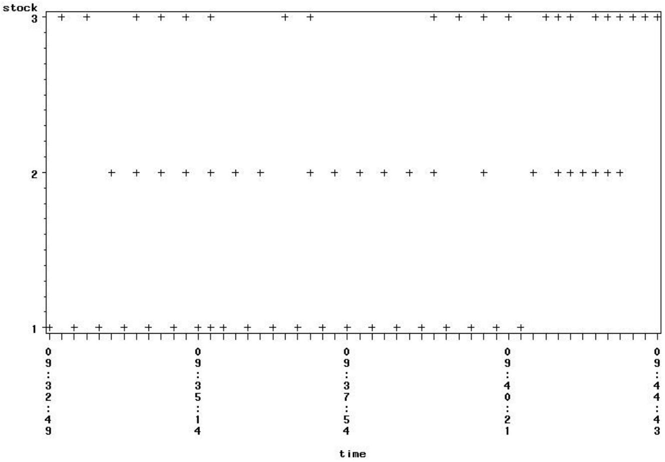
- Black-Scholes mathematically attractive
- GARCH and SV work well for low-frequency data
- Stationary returns
- Do not fit high-frequency data.

High-Frequency Financial Data

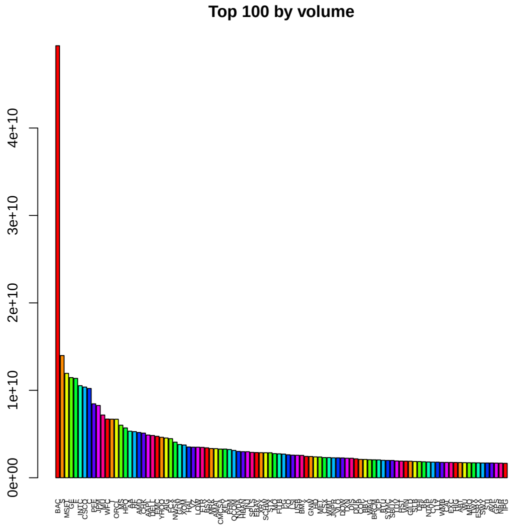
High-frequency financial data possess unique features absent in data measured at lower frequencies:

- Microstructure noise
- Nonstationary with jumps
- Irregularly spaced and random numbers of observations
- Nonsynchronous trading

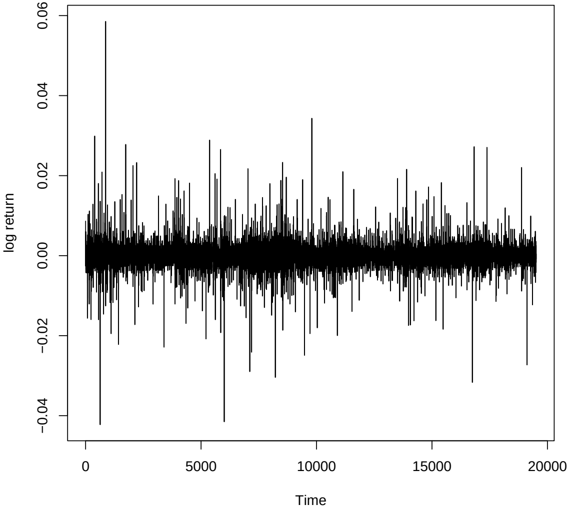
High-Frequency Financial Data



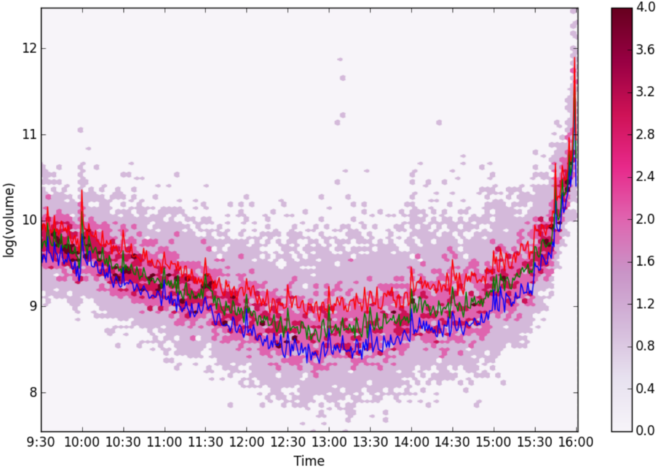
High-Frequency Financial Data



High-Frequency Financial Data



High-Frequency Financial Data



Problems and Challenges

There are major difficulties facing the portfolio allocation and volatility matrix estimation in high frequency financial data:

- Both number of observations (n) and number of assets (p) are large;
- Existing estimators (similar to MLE for covariance estimation) perform poorly;
- Existing dimension reduction methods fail due to non-synchronous data structure.

Computation is a very challenging due to large data sets and vast number of iterations in simulations and optimizations.

Portfolio Allocation and Risk Management

- Portfolio allocation is one of the most fundamental problems in finance.
- The process of determining the optimal mix of assets to hold in the portfolio is a critical issue in risk management.
- Dividing an investment portfolio among different assets based on the volatilities of the asset returns
- Ideal scenario: portfolio with maximum return and minimum risk

Modern Portfolio Theory

Markowitz (1952) was the original milestone paper for modern portfolio theory on the mean-variance analysis by solving an unconstrained quadratic optimization problem. It was later expanded in the book Markowitz (1959).

- Tradeoff between risk and expected return
- Aim to select a collection of investment assets that has lower risk than any individual asset
- Provide ways to find the best possible diversification strategy
- Sharpe (1966) introduced the Sharpe ratio for the performance of mutual funds, which is a direct measure of reward-to-risk.

Modern Portfolio Theory - Cont'd

Limitations

- very sensitive to errors in the estimates of the expected return and the conditional covariance of daily returns (which is often called volatility matrix)
- works well only if the portfolio size is small
- unstable performance when the portfolio size is large

Methodology

The proposed methodology consists of three steps:

1. Estimate integrated volatility matrix for each day by average realized volatility matrix (ARVM) estimators.
2. Regularize the inverse ARVM estimator using smoothly clipped absolute deviation (SCAD) penalty to obtain the ARVM-SCAD volatility estimator.
3. Make portfolio allocation based on the ARVM-SCAD volatility estimator.

High Performance Computing

We exploit a variety of HPC techniques, including

- parallel **R**
- Intel Math Kernel Library (MKL)
- automatic offloading to Intel Xeon Phi SE10P Co-processor

to speed up the simulation and optimization procedures in our statistical investigations.

Price Model

Suppose that there are p assets and their log price process $\mathbf{X}(t) = \{X_1(t), \cdots, X_p(t)\}^T$ obeys an Itô process governed by

$$d\mathbf{X}(t) = \boldsymbol{\mu}_t \, dt + \boldsymbol{\sigma}_t^T \, d\mathbf{B}_t, \qquad t \in [0, L], \tag{1}$$

Our goal is to estimate the integrated volatility matrix for the ℓ-th day, which is defined as

$$\Sigma_x(\ell) = \int_{\ell-1}^{\ell} \sigma_s \sigma_s^T ds, \qquad \ell = 1, \cdots, L. \tag{2}$$

Portfolio Allocation Problem

- For the portfolio with allocation vector $\boldsymbol{w}$ and a holding period T, the variance (risk) of the portfolio return is given by $R(\boldsymbol{w}, \boldsymbol{\Sigma}) = \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}$.
- However, it is well known that the estimation error in the mean vector $\boldsymbol{\mu}_t$ could severely affect the portfolio weights and produce suboptimal portfolios.
- This motivates us to adopt another popular portfolio strategy: the *global minimum variance portfolio*, which is the minimum risk portfolio with weights that sum to one. These weights are usually estimated proportional to the inverse covariance matrix, i.e., $\boldsymbol{w} \propto f(\Sigma^{-1})$.

Global Minimum Variance Portfolio

Following Jagannathan and Ma (2003) and Fan, Zhang and Yu (2012), we consider the following risk optimization with two different constraints:

$$\min \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}, \qquad s.t. \|\boldsymbol{w}\|_1 \leq c \text{ and } \boldsymbol{w}^T \mathbf{1} = 1 \tag{3}$$

where c is the gross exposure parameter which specifies the total exposure allowed in the portfolio. Here we consider two cases:

- $c = 1$ corresponds to the no short sale restriction.
- $c = \infty$ is the global minimum risk portfolio without any short sale constraint.

Other cases with varying c can be easily generalized in our methodology.

ARVM Estimator

Let $\boldsymbol{\tau} = \{\tau_r, r = 1, \cdots, m\}$ be the pre-determined sampling frequency. For asset i, define previous-tick times

$$\tau_{ir} = max\{t_{i\ell} \leq \tau_r, \ell = 1, \cdots, n_i\}, \qquad r = 1, \cdots, m.$$

Based on $\boldsymbol{\tau}$ we define realized co-volatility between assets i_1 and i_2 by

$$\tilde{\Sigma}_y(1, \tau)[i_1, i_2](\boldsymbol{\tau}) = \sum_{r=1}^{m} \left[Y_{i_1}(\tau_{i_1,r}) - Y_{i_1}(\tau_{i_1,r-1})\right] \left[Y_{i_2}(\tau_{i_2,r}) - Y_{i_2}(\tau_{i_2,r-1})\right], \tag{4}$$

and realized volatility matrix by

$$\tilde{\boldsymbol{\Sigma}}(1, \boldsymbol{\tau}) = \left(\tilde{\Sigma}_y(1, \tau)[i_1, i_2]\right). \tag{5}$$

ARVM-SCAD Estimator

With the estimated volatility matrix ARVM $\tilde{\Sigma}$, we define the ARVM-SCAD estimator as follows:

- consider penalized estimation of the covariance matrix Σ and its inverse matrix, precision matrix $\Omega = \Sigma^{-1}$. Denote their $(i,\ j)$-element by σ_{ij} and ω_{ij}, respectively.
- proceed to apply the SCAD penalty $p_\lambda(\cdot)$ to achieve a penalized estimator by solving the following optimization problem.

$$\min_{\Omega} - \log|\Omega| + \text{tr}(\tilde{\Sigma}\Omega) + \sum_{i \neq j} p_\lambda(\omega_{ij}). \tag{6}$$

- Note that (6) is not a convex programming. We will use the local linear approximation algorithm. At the end of tth step, denote the solution by $\hat{\Omega}^{(t)} = (\hat{\omega}_{ij}^{(t)})$. By using the local linear approximation, at the next step we solve the following optimization problem

$$\min_{\Omega} - \log|\Omega| + \text{tr}(\tilde{\Sigma}\Omega) + \sum_{i \neq j} p'_\lambda(|\omega_{ij}^{(t)}|)|\omega_{ij}| \tag{7}$$

Asymptotic Theory

Theorem 1

Under some regularity conditions, if $\max\{|p'_{\lambda_n}(\theta_{j0})| : \theta_{j0} \neq 0\} \to 0$, *then there exists a local maximizer* $\hat{\boldsymbol{\theta}}$ *of* $Q(\boldsymbol{\theta})$ *such that* $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\| = \mathcal{O}_P\left(e_n + b_n\right)$ *where* $a_n = \max\{|p'_{\lambda_n}(\theta_{j0})| : \theta_{j0} \neq 0\}$, $b_n = da_n$, *suppose* $b_n \to 0$ *as* $\lambda_n \to 0$. $e_n \sim n^{-1/6}$ *for the case with microstructure noise and* $e_n \sim n^{-1/3}$ *for the noiseless case.*

Theorem 2

Under some regularity conditions, if $\lim_{n\to\infty} n^{-1}/(e_n\lambda_n) \to 0$, *and* $\liminf_{n\to\infty} \liminf_{\theta\to 0+} p'_{\lambda_n}(\theta)/\lambda_n > 0$, *then our estimator in Theorem 1 satisfies*

$$P\left(\hat{\boldsymbol{\theta}}_2 = 0\right) \to 1, \quad \text{as } n \to \infty$$

where e_n *follows the same rate as in Theorem 1.*

Simulation Model

Assume the true log price $X(t)$ of p assets follow the diffusion model

$$dX(t) = \sigma_t^T \, dW_t \qquad t \in [0,\ 1]$$

where we take $\boldsymbol{\sigma}$ as a Cholesky decomposition of $\boldsymbol{\gamma}(t) = \boldsymbol{\sigma}_t \boldsymbol{\sigma}_t^T = (\gamma_{ij}(t))_{1 \le i,j \le p}$.
The diagonal elements of $\gamma(t)$ are generated from four common stochastic volatility models with leverage effect.

- Geometric Ornstein-Uhlenbeck process
- Sum of two CIR processes
- The volatility process in Nelson's GARCH diffusion limit model
- Two-factor log-linear stochastic volatility process.

Parallel R

While most features in **R** are implemented as single thread processes, efforts have been made in enabling parallelism with **R** over the past decade. Parallel package development coincides with the technology advances in parallel system development. For computing clusters.

- *Rmpi*
- *rparallel*
- *Snow*

Intel MKL

R enables linking to other shared mathematics libraries to speed up many basic computation tasks. One option for linear algebra computation is to use Intel Math Kernel Library (MKL). MKL includes a wealth of routines (e.g., the use of *BLAS* and *LAPACK* libraries) to accelerate application performance and reduce development time such as highly vectorized and threaded linear algebra, fast fourier transforms (FFT), vector math and statistics functions. Furthermore, the MKL has been optimized to utilize multiple processing cores, wider vector units and more varied architectures available in a high end system. Different from using parallel packages, MKL can provide parallelism transparently and speed up programs with supported math routines without changing code. It has been reported that the compiling **R** with MKL can provide three times improvements out of box.

Offloading to Phi Coprocessor

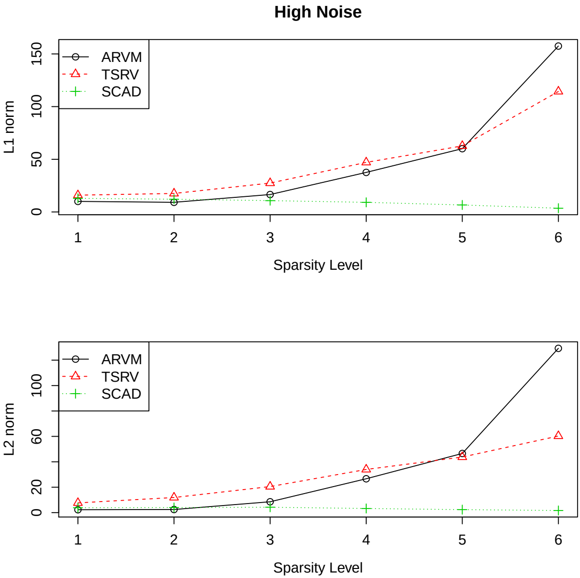
The basis of the Xeon Phi is a light-weight x86 core with in-order instruction processing, coupled with heavy-weight 512bit SIMD registers and instructions. With these two features the Xeon Phi die can support 60+ cores, and can execute 8 double precision (DP) vector instructions. The core count and vector lengths are basic extensions of an x86 processor, and allow the same programming paradigms (serial, threaded and vector) used on other Xeon (E5) processors. Unlike the GPGPU accelerator model, the same program code can be used efficiently on the host and the coprocessor. Also, the same Intel compilers, tools, libraries, etc. that you use on Intel and AMD systems are available for the Xeon Phi. **R** with MKL can utilize both CPU and Xeon Phi co-processor. In this model, **R** is compiled and built with MKL. Offloading to Xeon Phi can be enabled by setting environment variables as opposed to making modifications to existing **R** programs

Offloading to Phi Coprocessor

```
# enable mkl mic offloading
export MKL_MIC_ENABLE=0

# from 0.0 to 1.0 the work division
export MKL_HOST_WORKDIVISION=0.3
export MKL_MIC_WORKDIVISION=0.7

# Make the offload report big to be visible:
export OFFLOAD_REPORT=2

# now set the number of threads on host
export OMP_NUM_THREADS=16
export MKL_NUM_THREADS=16

# now set the number of threads on the MIC
export MIC_OMP_NUM_THREADS=240
export MIC_MKL_NUM_THREADS=240
```

Figure: Configuring environment variables to enable automatic offloading to Intel Xeon Phi Coprocessor. In this sample script, 70% of computation is offloading to Phi, while only 30% is done on host.

Simulation Results



Dow 30 Portfolio

We applied our methodology to a portfolio consisting 30 Dow Jones Industrial Average (DJIA) constituent stocks. The purpose of our empirical study is twofold: to demonstrate the applicability of our approach to a real high-frequency financial data set, as well as to provide some insights into regularization in the portfolio allocation using high-frequency data.

	Mean	Median	SD (%)
ARVM	0.084	0.094	4.659
ARVM(no short)	0.207	0.153	4.019
SCAD	0.101	0.133	4.603
SCAD(no short)	0.212	0.165	4.011

Table: Portfolio performance based on the Sharpe ratio

Summary

- Large portfolio allocation are very challenging due to the complexity of the problem.
- Volatility matrix modeling and estimation using high-frequency data pose additional difficulties.
- We proposed a new methodology to perform portfolio allocation that based on the regularized version of the estimated integrated volatility matrix.
- Theoretical and numerical studies indicate that the methodology works effectively.
- We exploit a variety of HPC techniques, including parallel **R**, Intel Math Kernel Library, and automatic offloading to Intel Xeon Phi coprocessor in particular to speed up the simulation and optimization procedures in our statistical investigations.

Thank you!