Biclustering Approaches for High-Frequency Financial Time Series

Nalini Ravishanker
Dept. of Statistics, U. Connecticut, Storrs

nalini.ravishanker@uconn.edu
www.stat.uconn.edu/~nalini

U Bahrain 2020

Joint work with: Haitao Liu and Jian Zou, Worcester Polytechnic Institute, USA

Outline

- Introduction and Motivation
- A Biclustering Algorithm using a Linear Metric
- Biclustering based on Mutual Information (a nonlinear metric)

Introduction and Motivation

We consider analysis of high-frequency transaction-by-transaction financial returns data. Suppose we look at time series of one-minute averaged returns within each trading day.

AAPL Prices for Jan. 2, 2013 before One-minute Averaging

DATE	TIME_M	SYM_ROOT	SIZE	PRICE
20130102	9:30:00.012	AAPL	100	553.820
20130102	9:30:00.015	AAPL	100	553.630
20130102	9:30:00.030	AAPL	100	553.630
20130102	9:30:00.048	AAPL	100	553.820
20130102	9:30:00.089	AAPL	100	553.620
20130102	9:30:00.097	AAPL	100	553.560

AAPL Log Returns from Jan. 2, 2013, after Preprocessing

Time / Stock	9:31	9:32	9:33	9:34		15:57	15:58	15:59
AAPL	-0.00323	-0.00118	0.00074	0.00213	...	-0.00033	0.00017	-0.00077
ABBV	0.00323	-0.00676	0.00000	0.00092	...	0.00207	0.00117	0.00148
ABT	-0.00030	0.00187	-0.00058	-0.00389	...	0.00059	-0.00034	-0.00129
ACN	0.00212	0.00202	0.00095	-0.00005	...	0.00045	0.00085	0.00048
....								
GOOG	0.00554	-0.00126	-0.00009	0.00074	...	0.00024	0.00015	0.00046
WAG	0.00165	0.00111	0.00443	-0.00221	...	-0.00018	0.00026	0.00077

Suppose a set of stocks form the rows of a matrix, while the times form columns of a matrix. For example: Consider returns from 7 stocks at 9 consecutive time points, T1-T9.

Time / Stock	T1	T2	T3	T4	T5	T6	T7	T8	T9
Stock1	r_{11}	r_{12}	r_{13}	r_{14}	r_{15}	r_{16}	r_{17}	r_{18}	r_{19}
Stock2	r_{21}	r_{22}	r_{23}	r_{24}	r_{25}	r_{26}	r_{27}	r_{28}	r_{29}
Stock3	r_{31}	r_{32}	r_{33}	r_{34}	r_{35}	r_{36}	r_{37}	r_{38}	r_{39}
Stock4	r_{41}	r_{42}	r_{43}	r_{44}	r_{45}	r_{46}	r_{47}	r_{48}	r_{49}
Stock5	r_{51}	r_{52}	r_{53}	r_{54}	r_{55}	r_{56}	r_{57}	r_{58}	r_{59}
Stock6	r_{61}	r_{62}	r_{63}	r_{64}	r_{65}	r_{66}	r_{67}	r_{68}	r_{69}
Stock7	r_{71}	r_{72}	r_{73}	r_{74}	r_{75}	r_{76}	r_{77}	r_{78}	r_{79}

Clustering can help us group stocks that exhibit similar stochastic behavior over time. Stocks 3, 4, 5 are clustered together by some criterion of similarity.

Biclustering can give us more information. By clustering stocks (rows) and columns of a data matrix simultaneously, we can now say that stocks 3, 4 and 5 co-move from times T3 to T6, and not at other times (light gray shaded submatrix).

Clustering Time Series

Groups stocks with similar behavior over all times.

Aghabozorgi et al. 2015 Information Systems 53: review of a decade of clustering. whole series clustering, sub-sequence clustering, time point clustering. Approaches are distance based, feature based, or model based clustering.

My work on clustering time series
Ravishanker and co-authors: MCAP 2010, 2017; CSDA 2013; WIREs 2018; ACM International Conference on Information and Knowledge Management 2019; ASMBI 2020: distance based clustering using frequency domain statistics.
Ravishanker and co-authors: Journal of Big Data Analytics in Transportation 2019: distance based clustering via Topological Data Analysis.

Challenges: memory demands, high dimensionality, choice of similarity measure.

Brief Literature Review on Biclustering

- Hartigan, JASA 1972: original biclustering idea for election data.
- Cheng & Church, Proc. Int. Syst. Mol. Biol. 2000: sequenced biological gene expressions, not preserving contiguity.
- Zhang et al., Infor. Tech. 2005 (CC-TSB algorithm): eliminate only start and end points in a sequence.
- Huang, App. Math & Comp. 2011: explored comovement in a small set of currencies across nonconsecutive time periods.
- Xue *et al.*, Math Prob. Eng. 2015: biclustering on daily stock prices for learning trading rules.

High-Frequency Financial Data Description

Biclustering consists of grouping the stocks and times simultaneously using data based techniques, no model or distributional assumptions made.

Transaction-by-transaction financial data: stock prices are seen at irregular spacing at a very fine time scale within each day.

We use 2013 data on 94 stocks from the Trade and Quotes (TAQ) database at Wharton Research Data Services (WRDS), U. Penn.

About 1 billion records, over 60 GB. Preprocessed and cleaned using python.

$S = 94$: number of stocks in our universe
$T = 249$: number of trading days in 2013
$N = 389$: number of one-minute time intervals within each day.

Within each trading day, for $i = 1, \ldots, S$ and $j = 1, \ldots, N$, let
r_{ij}: average 1-min log return of stock i at time j:

$$r_{ij} = \frac{P_{ij} - P_{i,j-1}}{P_{i,j-1}} = (1 - B) \log P_{ij}$$

1-minute averaged Log Returns, Jan. 2, 2013

Time / Stock	9:31	9:32	9:33	9:34		15:57	15:58	15:59
AAPL	-0.00323	-0.00118	0.00074	0.00213	...	-0.00033	0.00017	-0.00077
ABBV	0.00323	-0.00676	0.00000	0.00092	...	0.00207	0.00117	0.00148
ABT	-0.00030	0.00187	-0.00058	-0.00389	...	0.00059	-0.00034	-0.00129
ACN	0.00212	0.00202	0.00095	-0.00005	...	0.00045	0.00085	0.00048
....								
GOOG	0.00554	-0.00126	-0.00009	0.00074	...	0.00024	0.00015	0.00046
WAG	0.00165	0.00111	0.00443	-0.00221	...	-0.00018	0.00026	0.00077

A Biclustering Algorithm using a Linear Metric

This work appeared in STAT, 2018, Liu, Zou, Ravishanker.

We describe a multiple day time series biclustering algorithm, which extends the algorithm by Zhang et al. 2005, which extend the Cheng and Church 2000 greedy algorithm.

Our CC-MDTSB algorithm uses the Mean residue score metric and maintains contiguity in times while identifying a bicluster.

For each of the $T = 249$ trading days in 2013, let A_1 denote the $S \times N$ matrix of one-minute averaged log returns, where $S = 94$ and $N = 389$.

Main Steps in our CC-MDTSB Algorithm

Within a single trading day, the algorithm consists of:

Step 1: Initial H Score. Set $I = S = 94$ and $J = N = 389$. Use (1) to compute the H score for the $S \times N$ matrix of log returns A_1:

$$H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2. \tag{1}$$

Call this H_1.

Step 2: Row and Column Deletion. Start with input matrix A_1. The pair (I, J) will change dynamically when a row (stock) or a column (time point) is deleted.

- **Step 2a**. For each row $i \in I$, compute

$$R_i = \frac{1}{H(I,J)\,|J|} \sum_{j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2. \qquad (2)$$

If $R_i \geq \alpha$, delete row (stock) i from the row list (of stocks), I. Let A_{2a} be the resulting submatrix of A_1.

For example, if $5/94$ stocks are deleted, A_{2a} will be an 89×389 matrix. Recall A_1 was 94×389.

Compute the Mean Residue Score of A_{2a} as H_{2a} using (1).

- **Step 2b***. For each column $j \in \{\min(J), \max(J)\}$, (only start and end times), compute

$$C_j = \frac{1}{H(I,J)\,|I|} \sum_{i \in I} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2. \tag{3}$$

If $C_j \geq \alpha$, delete the corresponding time point from the column list (of times points), J, resulting in A_{2b}. For example, if one time point is deleted, then A_{2b} will have dimension 89×388. Using (1), compute its H score as H_{2b}.

- **Step 2c**. If $|H_1 - H_{2b}| < \theta$, stop the deletion process and go to Step 3. Otherwise, update the H_1 score as H_{2b} and repeat Steps 2(a) - 2(c) with A_{2b} as input.

- Suppose the final output of the deletion process is A_3.

Step 3: Insertion Process. Start with input A_3.

- **Step 3a.** For each $j' \in \{\min(J)-1, \max(J)+1\}$, compute (3). If $C_{j'} < \alpha$,insert this time point into column list (J), getting matrix A_{3a}. Use (1) to compute its H score as H_{3a}.

- **Step 3b.** For each $i \notin I$, compute (2); if $R_i < \alpha$, insert row (stock) i into row list (I, giving matrix A_{3b}, after inserting some stocks. Compute its H score as H_{3b}.

- **Step 3c.** If no row (stock) or column (time) is inserted into the row or column list, stop the insertion process, and go to Step 4. Otherwise repeat Steps 3(a) and 3(b).

- Suppose the final output of the insertion process is A_4. This is the first identified bicluster.

Step 4. Identify the Next Bicluster.

- From A_1, eliminate the stocks in A_4. Repeat Steps 2-3.
- Let A_5 denote the next identified bicluster.

Step 5. Stop Search for Biclusters.

• Let $\ell = 1, \ldots, L$ denote the number of identified biclusters for a trading day.

• Let S_ℓ denote the number of stocks in the ℓth bicluster within the trading day.

• Let U denote the total number of stocks (out of the S stocks) that belong in these L biclusters.

• If $U \leq \beta$, continue to repeat Steps 1-4.

• Otherwise, stop identifying biclusters for that trading day.

• Repeat this process for each of the T trading days, by repeating Steps 1-5.

Forecasting Comovement Trading Days

Over what percentage of trading days do a selected set of m stocks (say $m = 2$ or $m = 3$) move together?

Let $w = 1, \ldots, W$ denote the calendar week. Let $T(w)$ denote trading days in week w. Let $\tau_{\mathrm{F}}(w)$ be number of trading days for which the m-tuple is in the same biclusters in week w.

Cumulative comovement probability *up to* a given week w_c is

$$P_{\mathrm{F}}(w_c) = \frac{\sum_{w=1}^{w_c} \tau_{\mathrm{F}}(w)}{\sum_{w=1}^{w_c} T(w)}.$$

We use $P_{\mathsf{F}}(w_c)$ for $w_c = 1, \ldots, G$ as our training sample and $P_{\mathsf{F}}(w_c)$ for $w_c = G+1, \ldots, W$ as our test sample.

Our goal is to predict $\tau_{\mathsf{F}}(w_c) = \sum_{w=1}^{w_c} \tau_{\mathsf{F}}(w)$ for the test period.

On the training data, we use Double Exponential Smoothing using the Holt-Winters{stats} function in R to forecast future values in the test sample by $\hat{P}_{\mathsf{F}}(w_c)$ for $w_c = G+1, \ldots, W$.

We use algebra to solve the following identity

$$\hat{P}_{\mathbf{F}}(w_c) = \frac{\sum_{w=1}^{G} \tau_{\mathsf{F}}(w) + \sum_{w=G+1}^{w_c} \hat{\tau}_{\mathsf{F}}(w)}{\sum_{w=1}^{G} T(w) + \sum_{w=G+1}^{w_c} T(w)}.$$

to obtain

$$\hat{\tau}_{\mathsf{F}}(w_c) = \hat{P}_{\mathsf{F}}(w_c)\left(\sum_{w=1}^{G} T(w) + \sum_{w=G+1}^{w_c} T(w)\right) - \left(\sum_{w=1}^{G} \tau_{\mathsf{F}}(w) + \sum_{w=G+1}^{w_c-1} \hat{\tau}_{\mathsf{F}}(w)\right).$$

This enables us to estimate number of trading days over which a selected m-tuple of stocks comoves for any given w_c.

Number of biclusters on each trading day ranges from 6 to 13, with most trading days having 8 or 9 biclusters.

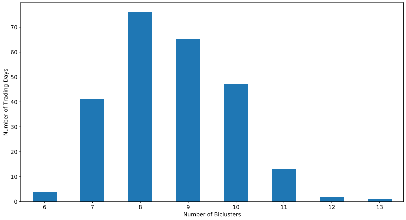

Figure: Distribution of Number of Biclusters.

- A few stocks - BAC, C, COF, GS, JPM, MET, USB and WFC, do not belong to any identified bicluster on any trading day!

Comovement Probability-Percent trading days when 2 stocks are in the same bicluster.

Stock 1	**Stock 2**	$P_F(53)$ **(%)**
CVX	XOM	67.5
SO	AEP	55.8
PG	CL	54.6
JNJ	PEP	52.6
MMM	XOM	52.2
MMM	MCD	51.8
MMM	UPS	50.6
JNJ	MMM	50.2
NOV	SLB	49.4
CVX	MMM	48.6
JNJ	MCD	46.2
PEP	PG	45.8
APC	DVN	45.4
MMM	HON	45.0
JNJ	PG	44.2

Comovement Probability-Percent trading days when 3 stocks are in same bicluster.

Stock 1	**Stock 2**	**Stock 3**	$P_F(53)$ **(%)**
CVX	MMM	XOM	37.2
COP	CVX	XOM	30.6
PEP	PG	CL	28.3
JNJ	MMM	MCD	27.6
CVX	XOM	UPS	27.2
JNJ	PEP	PG	26.6
MMM	XOM	MCD	26.3
MMM	XOM	UPS	26.0
MMM	MCD	UPS	25.8
CVX	XOM	HON	25.6
JNJ	PG	CL	25.2
JNJ	MMM	XOM	24.6
MMM	UPS	HON	24.3
CVX	MMM	UPS	24.2
CVX	MMM	MCD	23.8

Refinement in the Time Interval Determination

Our CC-MDTSB algorithm results in very long intervals of times in the biclusters.

Our new refinement breaks the time interval resulting from only dropping end-points, thus leading to biclusters that do not stretch over very long time spans.

We also set a minimum contiguous length of 30 minutes in order to form meaningful biclusters.

Biclustering based on Mutual Information (a nonlinear metric)

Mutual Information (MI) is related to the joint entropy of two random variables and measures the dependence between them (Kraskov, 2004; Brillinger, 2004).

An MI based distance captures relationships between stock returns that are not detected by the usual linear distance metrics (Kim, 2017; Guo, 2018).

Entropy of X.

$$H(X) = \begin{cases} -\sum_x p(x) \log\left(p(x)\right) & \text{if } X \text{ is discrete,} \\ -\int f(x) \log\left(f(x)\right) dx & \text{if } X \text{ is continuous.} \end{cases}$$

Joint entropy of X and Y.

$$H(X,Y) = \begin{cases} -\sum_x \sum_y p(x,y) \log\left(p(x,y)\right) & \text{if } X \text{ and } Y \text{ are discrete,} \\ -\int\int f(x,y) \log\left(f(x,y)\right) dxdy & \text{if } X \text{ and } Y \text{ are continuous.} \end{cases}$$

MI between X and Y measures the information shared by X and Y:

$$\begin{aligned} MI(X,Y) &= H(X) + H(Y) - H(X,Y) \\ &= \begin{cases} \sum_x \sum_y p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right), & \text{discrete case} \\ \int\int f(x,y) \log\left(\frac{f(x,y)}{f(x)f(y)}\right) dxdy, & \text{continuous case.} \end{cases} \end{aligned} \tag{4}$$

A large value of $MI(X, Y)$ indicates that the two random variables are highly associated, while a low value indicates low association. If and only if $MI(X, Y) = 0$, the two variables are independent. While the MI is not a distance metric, it can be converted to a distance metric:

$$d(X, Y) = H(X, Y) - MI(X, Y), \tag{5}$$

which satisfies the non-negativity, symmetry and triangle inequality properties. The normalized distance is then defined as

$$D(X, Y) = 1 - \frac{MI(X, Y)}{H(X, Y)}, \tag{6}$$

so that $0 \leq D(X, Y) \leq 1$.

We use a reduced bias "jackknife estimate" of MI (Zeng, et al. 2018 to obtain the normalized distance.

We develop a biclustering algorithm using the normalized MI based distance as a metric.

As before, let $\mathcal{S} = \{1, \ldots, S\}$ denote S stocks observed over T times over N trading days.

For each trading day, our MI-MDTSB algorithm

- first clusters the rows (stocks) using a suitable clustering algorithm

1. Average Distance Gradient Change Algorithm
2. Average Silhouette Criterion Based Algorithm

- then, within each cluster, we employ an algorithm similar to the CC-MDTSB algorithm based on the MI based distance metric.

This yield the identified biclusters (again there are three user-defined threshold parameters, α, β and γ).

Details may be requested from the authors (nalini.ravishanker@uconn.edu).

A Few Comments

We can use raw returns and volatility adjusted returns and compare the resulting comovement patterns.

We can carry out a study on algorithm robustness to sampling rate (1-min averaging versus 5-min averaging, say).

We can carry out a post analysis on the comovements to identify patterns that may guide trading strategies.

Thank you!