A discussion of "Discovering latent network structure in point process data"

Marcos Costa Santos Carreira

CMAP - École Polytechnique

Machine Learning Journal Club - 2017

Contents

Why go back to 2014?

- Machine learning can sometimes focus too much on correlation and forget causation
- Network analysis is important
- Time series have a structure
- Point processes are useful models of reality

And what is this paper about?

- Cutthroat competitors, alternating between:
  - Standing their ground with watchfulness
  - Swift and merciless aggression
- And gang violence in Chicago

And what is this paper about?

- Cutthroat competitors, alternating between:
  - Standing their ground with watchfulness
  - Swift and merciless aggression
- And gang violence in Chicago

Why write a paper about it?

- Network structures
  - Vertices ✓
  - Edges ✓
- Implicit networks
  - Vertices ✗
  - Edges ✗
  - Noisy observations ✓
- Need to infer structure

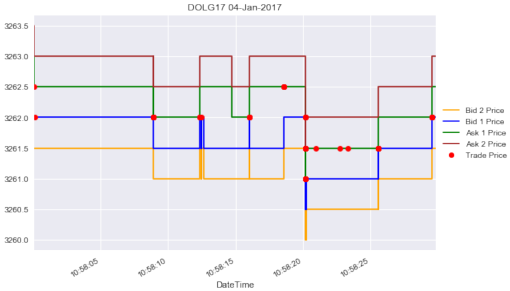
Data

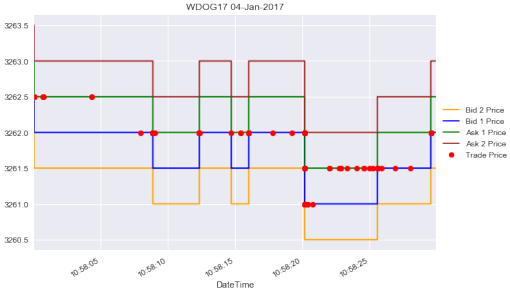
- Microstructure
  - Tick data
  - Order book data
  - Correlated instruments
- Research applications
  - Trader: trade
  - Exchange: Design products and markets
  - Regulators: Design markets avoiding externalities, analyze risks

Model is realistic

- Many choices of models come from tractability:
  - "The advantage of the Hawkes process is that its linear form allows for elegant fully-Bayesian inference" (Linderman and Adams 2015)
- But in this case the tractable model makes sense on the real world
  - Observable (timestamped) characteristics of two similar contracts
    - Volumes and prices traded
    - Aggregated order book behavior (arrival and cancelation of orders can be inferred)
- Directed cross dependencies on orders and prices
- Endogenous trading modeled by self-excitation

Moving pictures





Python package

- Linderman and Adams 2015 improve results with:
  - Discrete-time formulation
  - Stochastic Variational Inference (SVI) algorithm
- PyHawkes package:
  - `https://pypi.python.org/pypi/pyhawkes/0.3.1`

1. "Discovering latent network structure in point process data"
   - Motivation
   - More motivation

2. **Learning to learn machine learning**
   - **Why can we learn?**
   - What can we learn?
   - How can we learn?

3. The uncertainty zones model
   - Spreads
   - Price changes
   - The uncertainty zones model
   - Interpreting the model

4. Hawkes processes and model calibration
   - Keep it simple
   - Measure
   - Extend

Exchangeable sequences

- Infinite sequence is exchangeable if all permutations of finite subsets of sequence have the same distribution
  - If sequence is iid => exchangeable
  - But not the opposite (e.g. Polya's Urn)
- Example:
  - Gaussian multivariate, constant and positive correlation

De Finetti's theorem

- Sequence X is exchangeable if and only if
  - There exists a random probability measure Θ on X such that
  - X conditional on Θ is independent and Θ-distributed
- Θ represents the common structure
  - P[X given Θ] independent remaining randomness
- All exchangeable binary sequences are mixtures of Bernoulli sequences

Extensions

- Hewitt-Savage
  - Limiting proportion (frequency)
  - Prior
- Diaconis and Freedman
  - Finite versions
  - Approximations
- Arrays
- Random Graphs

Parametrization

- Hawkes impulse response as product of:
  - Binary adjacency matrix
  - Weight matrix
  - Temporal aspect (probability density)
- Separate beliefs about:
  - Sparsity structure
  - Strength of interactions

Parametrization

- Models for graphs
  - Empty graphs: independent background models
  - Complete graph: Standard Hawkes process
- Weight matrix
  - Units: expected number of events
- Shared background fluctuations:
  - Sparse log Gaussian Cox process

Methods

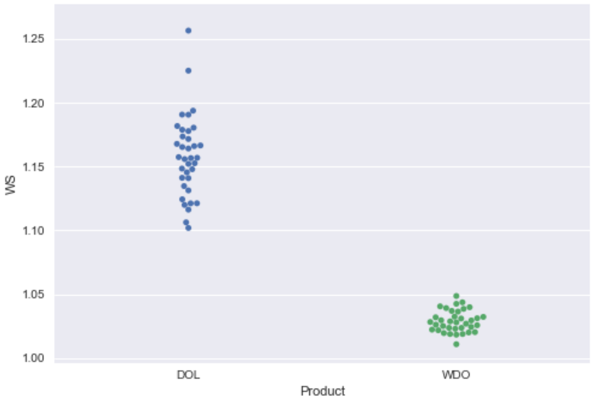
- Gibbs sampling
  - collapsing to improve algorithm (parallelism)
  - but still computationally intensive
  - Prior
- Improved later
  - Approximation to model
  - Efficient stochastic variational inference

Stability

- Constrain Hawkes
    - weak priors
    - tune sparsity parameter
- But strong weights need counterbalance
    - Strong sparsity limits
    - Less excitatory feedback loops

Average spreads

- Time-weighted spreads (in ticks):
  $WS = \left(\frac{1}{\alpha}\right)\left(\frac{1}{T}\right)\sum\left((Ask - Bid)_{t_j \to t_{j+1}} * (t_{j+1} - t_j)\right)$



Large ticks

- Large tick assets: Spread almost always equal to one tick (α)
- Small tick assets: Spread typically a few ticks
  - Importance of relative tick size: $\frac{\alpha}{S}$
  - A more volatile asset should make the relative tick smaller
- Too large a tick will generate too many bounces (alternations) and not many level changes (continuations)
  - And a large order might hijack the queue
- Too small a tick and the spreads might show flickering quotes and no trades
  - Externality of processing quotes with low informational content
- Goldilocks: What is the ideal tick size? Dayri and Rosenbaum (2012)

AC/DC

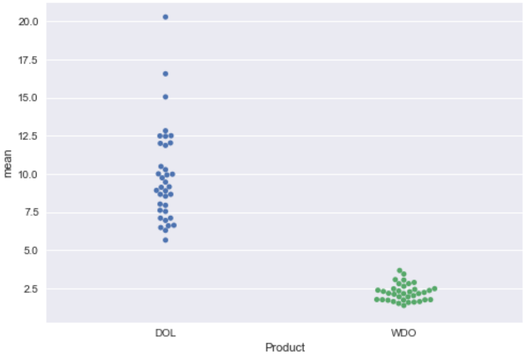
- Alternations:
  - Last price change has opposite sign of previous last price change (bid-ask bounce)
- Continuations:
  - Last price change has same sign of previous last price change (trends, trades-through)
- Durations:
  - Time between traded price changes
- Can we estimate duration?
  - Yes

$$D = ET = \left(\frac{\alpha}{2 * S * \sigma_s} \right)^2 \quad (1)$$

1. "Discovering latent network structure in point process data"
   - Motivation
   - More motivation
2. Learning to learn machine learning
   - Why can we learn?
   - What can we learn?
   - How can we learn?
3. The uncertainty zones model
   - Spreads
   - Price changes
   - The uncertainty zones model
   - Interpreting the model
4. Hawkes processes and model calibration
   - Keep it simple
   - Measure
   - Extend

Durations

- Time between price changes: $D = \left(\frac{1}{N}\right) \sum \left(\tau_{j+1} - \tau_j\right)$



42

- Interesting results:
  - Why are these durations (and other statistics) different if the relative tick size is the same?
- The answer lies is how the trading happens on each contract
  - Banks are risk averse when executing flows
  - Benchmark is arrival price
  - Take all displayed liquidity
    - In the DOL contract, not in the WDO

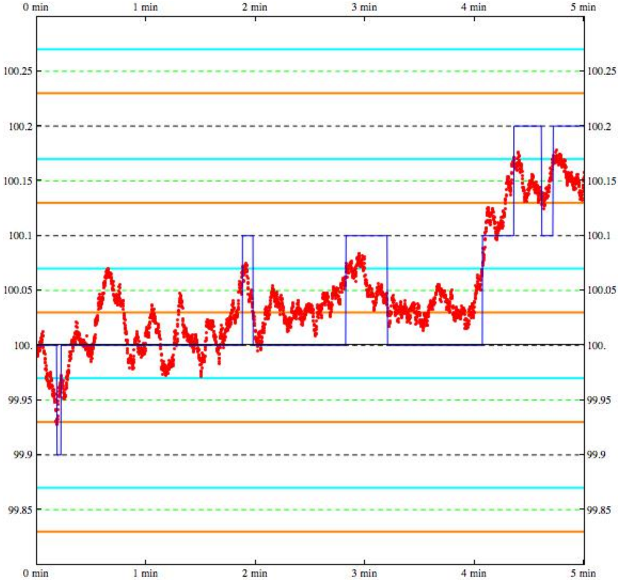
Some models are useful

- Start with: $\varepsilon_t = \log P_t - \log X_t$ where:
  - X_t is the efficient price (the model price)
  - P_t is the transaction price (in the grid defined by the tick value)
  - ε_t is the microstructure noise process
- The model allows:
  - Price discreteness
  - Price movements of one or several ticks
  - A behavior depending of several factors such as features of the order book
  - Delays caused by the reaction times of the market participants are not null
- If a transaction occurs at some value and leads to a price change, it means that the efficient price process has been close enough to this value shortly before the transaction time.

Necessary but ...

- How, when and where trades happen ?
  - Suppose the tick size α is 0.10, so mid-ticks would be: 100.05, 100.15, 100.25, ... and allowable transaction prices would be: 100.00, 100.10, 100.20, 100.30, ...
  - Uncertainty band: bands $\pm \eta \, \alpha$ around the mid-ticks, with $0 < \eta < 1$ and α equal to the tick size
  - In the example above, make $\eta = 0.20$
  - The bands will be: {100.03,100.07}, {100.13,100.17}, ...
  - A necessary condition for a trade to happen at 100.10 would be for the efficient price to have left the zone {100.03,100.07} on the way up (through 100.07) or to have left the zone {100.13,100.17} on the down (through 100.13)
  - Market conditions still need to allow the trade to happen though, which might lead to price movements of several ticks (efficient price moves through several zones before a trade happens)

I can see clearly now



1. "Discovering latent network structure in point process data"
   - Motivation
   - More motivation
2. Learning to learn machine learning
   - Why can we learn?
   - What can we learn?
   - How can we learn?
3. The uncertainty zones model
   - Spreads
   - Price changes
   - The uncertainty zones model
   - Interpreting the model
4. Hawkes processes and model calibration
   - Keep it simple
   - Measure
   - Extend

Small eta good, large eta ... ?

- What does η mean?
  - η quantifies the aversion to price changes
  - The larger the η, the farther from the last traded price the efficient price has to be so that a price change may occur
- By counting the alternations and continuations of the traded price, we can estimate η, and from that estimate the efficient price from the traded price. Then we'll estimate the realized volatility on the efficient price, rather than the traded price.

$$N^{(a)}_{\alpha,t,k} = \sum_{t_i \leq t} \mathbb{I}\{(P_{t_i} - P_{t_{i-1}})(P_{t_{i-1}} - P_{t_{i-2}}) < 0 \text{ } AND \text{ } |P_{t_i} - P_{t_{i-1}}| = k\alpha\} \tag{2}$$

$$N^{(c)}_{\alpha,t,k} = \sum_{t_i \leq t} \mathbb{I}\{(P_{t_i} - P_{t_{i-1}})(P_{t_{i-1}} - P_{t_{i-2}}) > 0 \text{ } AND \text{ } |P_{t_i} - P_{t_{i-1}}| = k\alpha\} \tag{3}$$

$$\frac{N^{(c)}_{\alpha,t,k}}{N^{(a)}_{\alpha,t,k}} \text{ is an estimator of } 2\eta \tag{4}$$

Measuring volatility

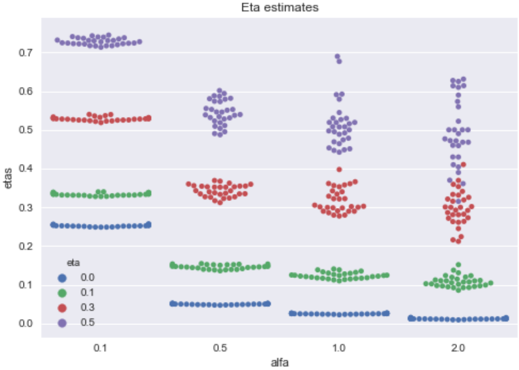
- Run Monte Carlo with fixed (known) volatility for efficient price
- Different values for α and η generate different traded prices
- From traded prices estimate η
- With traded prices and η estimate efficient prices
- Estimate volatility with traded prices and efficient prices and compare with original volatility
- 30 efficient price paths with a step of 0.1 seconds and a total time of 7 hours
- Parameters: Spot = 1000, σ = 2% (daily), t=1 day, steps=7*3600*(1/0.1)=252000

Results

α	η	$\hat{\eta}$	$\hat{\sigma}_{TP}$	$\hat{\sigma}_{EP}$	$D * 10^6$	D/DurT	ΔP
0.1	0.0	0.25	2.8%	2.0%	6	2.0	0.1
0.1	0.1	0.33	2.5%	2.0%	17	2.2	0.1
0.1	0.3	0.53	1.9%	2.0%	27	2.5	0.1
0.1	0.5	0.73	1.7%	2.0%	37	2.6	0.1
0.5	0.0	0.05	6.3%	2.0%	62	0.4	0.5
0.5	0.1	0.15	3.7%	2.0%	184	1.0	0.5
0.5	0.3	0.34	2.4%	2.0%	432	1.6	0.5
0.5	0.5	0.54	1.9%	2.0%	687	1.9	0.5
1.0	0.0	0.02	8.9%	2.0%	126	0.2	1.0
1.0	0.1	0.12	4.0%	2.0%	623	0.8	1.0
1.0	0.3	0.32	2.5%	2.0%	1640	1.6	1.0
1.0	0.5	0.52	1.9%	2.0%	2650	1.9	1.0
2.0	0.0	0.01	12.7%	2.0%	250	0.1	2.0
2.0	0.1	0.11	4.3%	2.0%	2222	0.7	2.0
2.0	0.3	0.30	2.5%	1.9%	6263	1.5	2.0
2.0	0.5	0.49	2.0%	1.9%	10416	1.9	2.0

Eta

- Eta estimates get better with a higher alfa



Realized volatility - traded price

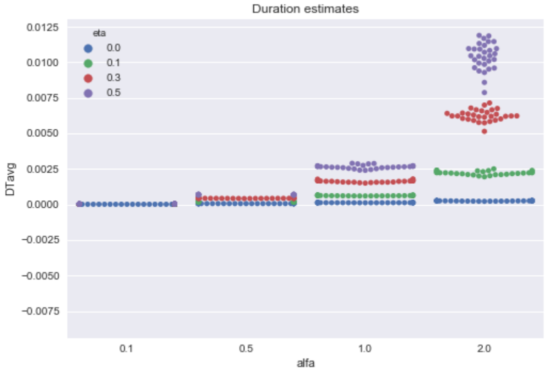
- Large ticks make it harder to estimate volatility directly

Realized volatility - efficient price

- Large ticks increase dispersion but no bias

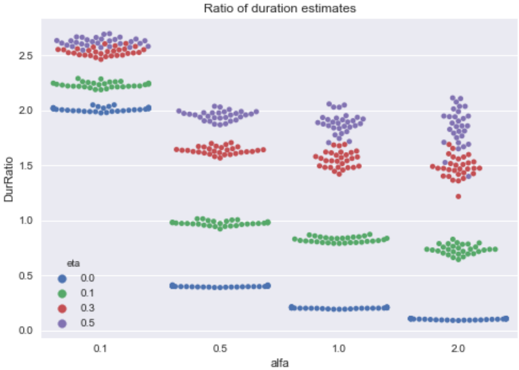
Durations

- Large ticks increase durations
- Larger etas increase durations



Expected durations

- Adjust formula $\left(\alpha \rightarrow \alpha\left(1+\eta\right)\right)$, calculate expected durations and divide realized durations by them



Shortcuts

- Adjust the expected duration with eta:

$$D_\eta = ET = \left(\frac{\alpha(1+\eta)}{2 \cdot S \cdot \sigma_s} \right)^2 \tag{5}$$

- How many trades (at least) in a day:

$$M \geq nPC = \frac{H}{D_H} \tag{6}$$

$$\sqrt{nPC} = \frac{\sqrt{H}}{\sqrt{D_H}} = \frac{2 \cdot \sqrt{H} \cdot S \cdot \sigma_H}{\alpha\,(1+\eta)} \tag{7}$$

- Daryi and Rosembaum (2012):

$$\eta \cdot \alpha \cdot \sqrt{M} = p_1 \cdot \sigma_H \tag{8}$$

- Could be explained by:

$$\eta \cdot \alpha \cdot \sqrt{M} = \left(2 \cdot \sqrt{H} \cdot \eta \cdot S\right) \sigma_H \tag{9}$$

Volatility of real data

- Covariance estimates with effective price different from Hayashi-Yoshida
04-Jan	$\hat{\sigma}_{TP}$	$\hat{\sigma}_{EP}$
DOL	0.99%	0.72%
WDO	1.89%	0.88%
Correl	0.23	0.55
- Epps effect

1. "Discovering latent network structure in point process data"
   - Motivation
   - More motivation
2. Learning to learn machine learning
   - Why can we learn?
   - What can we learn?
   - How can we learn?
3. The uncertainty zones model
   - Spreads
   - Price changes
   - The uncertainty zones model
   - Interpreting the model
4. Hawkes processes and model calibration
   - Keep it simple
   - Measure
   - Extend

Definitions

- Point process: $(t_i)_{i\in\mathbb{N}^*}$ $\forall i \in \mathbb{N}^*,\ t_i < t_{i+1}$
- Counting process: $N(t) = \sum_{i\in\mathbb{N}^*} \mathbb{1}_{t_i \le t}$
- Duration process: $\forall i \in \mathbb{N}^*,\ \delta t_i = t_i - t_{i-1}$
- Intensity process: $\lambda(t \mid \mathcal{F}_t) = \lim_{h\downarrow 0} \mathbb{E}\left[\frac{N(t+h)-N(t)}{h} \mid \mathcal{F}_t\right]$
- Simple Hawkes: $\lambda(t) = \lambda_0(t) + \sum_{t_i<t} \sum_{j=1}^{P} \left(\alpha_j e^{-\beta_j(t-t_i)}\right)$
- Even simpler Hawkes: $\lambda(t) = \lambda_0 + \sum_{t_i<t} \left(\alpha e^{-\beta(t-t_i)}\right)$

Simulation code

```
# %% Simulation - functions

def ru():
    return np.random.uniform()

def ls(L0,a,b,aT,s):
    return L0+sum([a*np.exp(-b*(s-t)) for t in aT if t<s])
# %% Simulation - general routine

def gr(a,b,L0,T,aN,aL,aT):
    aN=aN+[aN[-1]+1]
    s=aT[-1]
    L1=ls(L0,a,b,aT,aT[-1])+a
    grf=True
    while grf:
        U=ru()
        s=s-np.log(U)/L1
        if s<T:
            D=ru()
            Ls=ls(L0,a,b,aT,s)
            if D<=Ls/L1:
                grf=False
            else:
                L1=Ls
        else:
            grf=False
    aT=aT+[s]
    aL=aL+[L1]
    return [aN,aL,aT]
```

```
# %% Simulation - initial step and loop

def hwke(a,b,L0,T):
    aT=[0]
    aL=[L0]
    aN=[0]
    L1=aL[-1]
    aN=aN+[aN[-1]+1]
    U=ru()
    s=-np.log(U)/L1
    if s<=T:
        aT=aT+[s]
    while aT[-1]<T:
        [aN,aL,aT]=gr(a,b,L0,T,aN,aL,aT)
    sT=pd.Series(aT,index=aN[:len(aT)])
    sL=pd.Series(aL,index=aN[:len(aL)])
    dfN=pd.DataFrame({'t':sT,'L':sL})
    dfN.index.names='N'
    return dfN[dfN['t']<=T]
# %% Simulation - Initial values

a=0.6
b=0.8
L0=1.2
T=10
```

- Can be adapted for multivariate processes

Calibration code

```
# %% Log-likelihood auxiliary functions

def recR(b,ti,ti1,R1):
    return np.exp(-b*(ti-ti1))*(1+R1)

def fillR(b,aT,aR):
    aaR=aR
    for j in np.arange(2,len(aaR)):
        aaR[j]=recR(b,aT[j],aT[j-1],aaR[j-1])
    return aaR

def gfillR(df,L0,a,b,T):
    dfg=df.copy()
    dfg['R']=0.
    dfg['R']=fillR(b,dfg['t'],dfg['R'])
    dfg['eb']=(a/b)*(1-np.exp(-b*(T-dfg['t'])))
    dfg['eb'][0]=0
    dfg['lnR']=np.log(L0+a*dfg['R'])
    dfg['lnR'][0]=0
    return dfg
```

```
# %% Log-likelihood function

def lglk(dfh,L0,a,b,T):
    dfg=gfillA(dfh,L0,a,b,T)
    dfg=gfillR(dfg,L0,a,b,T)
    s1=T-T*L0-dfg['eb'].sum()
    s2=dfg['lnR'].sum()
    return -(s1+s2)

def glglk(x,df,T):
    return lglk(df,np.abs(x[0]),np.abs(x[1]),np.abs(x[2]),T)

# %% Run minimization

dff=dfh.copy()
tn=T

def flglk(x):
    return lglk(dff,x[0],x[1],x[2],tn)

opti.fmin(flglk,np.array([1,0.8,1.1]))
```

- Toke (2011) shows results of estimation

What should we count?

- Trading intensity
- Order arrival
- Price changes
- Trade-throughs (Toke and Pomponio (2012))
- Microstructure noise

DOL: Trading intensity - 2008

- Time series of timestamps where there was a trade
- Problem of discretization (timestamps in seconds)
- $\alpha = 0$ for both days below
  - Slow (03-Jul): $\lambda = 0.092$
  - Fast (23-Oct): $\lambda = 0.186$

2017

- Effects of resolution in timestamps
  - Intensity becomes constant once downsampled too much

04-Jan	dt (s)	λ	α	β	$\frac{\alpha}{\beta}$
DOL	0.001	0.20	91.4	268.4	0.34
WDO	0.001	0.90	57.6	160.2	0.36
DOL	0.01	0.18	4.0	19.7	0.20
WDO	0.01	0.31	0.75	1.05	0.71
DOL	0.1	0.08	0.07	0.12	0.58
WDO	0.1	0.14	0.16	0.20	0.83
DOL	1.0	0.16	0.00	2.52	0.00
WDO	1.0	0.47	0.00	1.88	0.00

- With dt=1s there's a trade at every 6 seconds for the DOL and a trade every 2 seconds for the WDO
- But the ratio of changes in the first two levels to trades is close to 10
  - Need to make sure there's enough timestamp resolution to treat orders

1. "Discovering latent network structure in point process data"
   - Motivation
   - More motivation
2. Learning to learn machine learning
   - Why can we learn?
   - What can we learn?
   - How can we learn?
3. The uncertainty zones model
   - Spreads
   - Price changes
   - The uncertainty zones model
   - Interpreting the model
4. Hawkes processes and model calibration
   - Keep it simple
   - Measure
   - Extend

A boil of hawks

- Calibrate multivariate Hawkes model
- Correlation
  - Careful with Epps effect
  - Use Uncertainty Zones model to estimate co-volatility

A network of hawks

- Linderman and Adams (2014) use random graphs models and Hawkes processes
    - Look for “hidden” structure in correlated data
- Will apply to DOL + WDO:
    - External factors affect both contracts
    - Flow in DOL affects WDO
- Quantifying endogeneity

Books and papers

- Bacry, Mastromatteo and Muzy (2015): "Hawkes processes in finance"
- Toke (2011): "An Introduction to Hawkes Processes with Applications in Finance"
- Toke and Pomponio (2012): "Modelling Trades-Through in a Limit Order Book Using Hawkes Processes"
- Linderman and Adams (2014): "Discovering latent network structure in point process data"
- Linderman and Adams (2015): "Scalable bayesian inference for excitatory point process networks"
- Robert and Rosenbaum (2011): "A new approach for the dynamics of ultra-high-frequency data: the model with uncertainty zones"
- Dairy and Rosenbaum (2012): "Large tick assets: implicit spread and optimal tick size"
- Carreira and Brostowicz (2016): "Brazilian Derivatives and Securities"