Equilibrium Resolution for Epoch Partitioning

Wojciech Wisniewski, Yuri Kalnishkan, David Lindsay, Siân Lindsay

Equilibrium Resolution for Epoch Partitioning

Wojciech Wisniewski[1][0000−0002−4296−6490], Yuri Kalnishkan[1][0000−0003−1134−8937], David Lindsay[2][0000−0002−5535−4880], and Siân Lindsay[2][0000−0002−6985−953X]

[1] Department of Computer Science, Royal Holloway, University of London, Egham, United Kingdom
wojciech.wisniewski.2019@live.rhul.ac.uk, yuri.kalnishkan@rhul.ac.uk
[2] Algorithmic Laboratories Ltd, Ocean House, Bracknell, Berkshire, RG12 1AX, United Kingdom {david, sian}@algolabs.com

Abstract. This paper proposes a method for determining the resolution for the processing of irregularly-sampled time series data to provide a balanced perspective of agents' behaviour. The behaviour is described as a collection of prolonged events, which are characterised by start/open and end/close times in addition to other useful attributes. We propose the definition of an equilibrium resolution and carry out its analysis based on probabilistic assumptions. The resulting methods of determining the equilibrium resolution are tested on real-life time series data sets from the Financial and Travel problem domains.

Keywords: Time Series Resolution · Time Series Partitioning · Database Management · Big Data

1 Introduction

Dealing with a number of time series that have been monitored by an infrastructure recording only the beginning and the end of every one of them can be problematic if insight into what happened in between those times is required. A detailed data flow derived from a collection of agents acting within some interval of interest could be storage intensive, yet there remains the need to gain insight into agent behaviour. In this study we focus on two distinct data flows pertaining to the behaviours of 1) clients trading with financial market makers, and 2) taxis making journeys in New York City. The agents in both case studies present complex challenges which relate to their choice of action (i.e. buy or sell a particular financial instrument and geographic locations of a taxi pick-up and drop off), as well as the frequency and duration of their actions. Many hundreds of clients may trade with a financial market maker, with each client trading many different instruments and holding positions for different periods of time - seconds, days or months. Likewise with taxi journeys, customers may take trips throughout New York that can take minutes or hours. Studies such as [1] have introduced methods to help manage this complexity by aggregating related time series together, thereby gleaning further information about agent

behaviour, epoch-by-epoch and in a way that is practical to store and access. The choice of partition-size, or resolution, at the aggregation step is important for inferring best-possible insights into agent behaviour: too large and information is missed; too small and aggregation is no longer meaningful.

In this study we propose a method for determining a so called "equilibrium resolution" for agents' data flow using time series from the aforementioned case studies. First, we present a definition solely dependent on the duration of an agents action behaviour which is based on probabilistic assumptions. We then give an empirical extension of the this, introducing the concept of a "monitoring function" which binds the duration of agents behaviour with a feature that one wishes to infer insight into. For example, in the financial trading dataset we may want to track the profit and loss (PnL) as an attribute of client trading activity.

This paper is organised as follows: in Section 2 we provide a literature review, in Section 3 we describe a framework for efficient monitoring/information retrieval and in Section 4 we propose a definition of the equilibrium resolution based on the equilibrium between different types of events. In Section 5 we carry out the theoretical study based on probabilistic assumptions. Corollary 1 shows that under natural assumptions the equilibrium is achieved when the resolution equals to the average duration of the event. This provides an important intuition and gives us a method for calculating the equilibrium resolution. In Section 6 we apply the method to real-life datasets and analyse the performance. Section 7 discusses an extension of our approach based on monitoring functions.

2 Literature review

In the literature we can find many studies which deal with similar topics relating to the study of time series resolution. In [13], the optimal resolution of time series is proposed based on wavelet transform and structural similarity measure. Studies such as [8] look at methods of using variable width intervals that increase sampling during more active regions in the time series. Elsewhere, work by [2], [3] [4], and [12] chooses a resolution which results in the lowest forecast error.

Since agents' behaviour can be modelled via complex networks, several papers have addressed the issue of selecting the best time intervals for meaningful analysis and insights. The structural features of networks emerging from aggregating empirical data over different time intervals is studied in [6], focusing on networks derived from time-stamped, anonymized mobile telephone call records. In [9], the authors focus on identifying meaningful resolution levels that best reveal critical changes in the network structure, by balancing the reduction of noise with the loss of information. Whilst these studies tackled time series resolution, their methods were not able to completely satisfy the needs of our particular problem. For example, if we consider the approach by [9] - which involves the construction of a dynamic network at time intervals of the form $[t, t+w)$ (where w is the resolution) to produce a time series of labeled graphs. The optimal window of aggregation is determined to balance between the loss of temporal structural information and fluctuations that obscure what is relevant in a struc-

tural change. We applied Statistically Validated Networks (see in [5], [10]) to build dynamic networks from the financial trading dataset. However in order to obtain reasonably large and stable networks the time window needed to be set to a time resolution of roughly a month, which is far too large to track meaningful trading behaviour such as profit and loss.

In [13] only a general idea is presented without a precise description. In studies by [2], [3], [4] and [12] the authors determine the best resolution for forecasting purposes. In our financial trading case study, predicting PnL is a hard task as it is based on the underlying dynamics of noisy trade data in addition to the randomness of price movements, making a meaningful comparison with such techniques infeasible. To best of the authors' knowledge no other study provided a suitable candidate method for tackling our problem. We could argue that our proposed method should be considered as an ad hoc approach for monitoring a desired feature in a complex system.

3 Preliminaries

3.1 Prolonged Events

Consider an environment where prolonged events occur in continuous time. We identify an *event* with a triple $\langle OT_j, CT_j, a_j \rangle$, where $OT_j \in \mathbb{R}_+$ is the *opening time*, $CT_j \in \mathbb{R}_+$ is the *closing time*, and a_j is the array of event attributes (such as id, the list of agents associated with it, certain discrete or continuous characteristics) which will be referred as *exogenous* variables that are deemed useful in helping to explain agent behaviour. We assume that $CT_j > OT_j$.

In this paper we consider two main examples (discussed in more detail in Section 6.). The first is *trading*. Here an event consists of opening, holding, and closing a position. The starting time describes when the position was opened, the closing time shows when it was closed, and the attributes include the id of the trader who opened the position, the assets involved, the size of the position, and whether the position is long or short. We study a publicly available dataset by [7] as exemplifying this environment.

In the second example the events are *taxi rides*. The opening time describes when the ride started, the closing time shows when it ended, and the attributes include the ids of the driver and passenger and the coordinates of the starting and finishing points. A dataset of taxi rides by [11] is based on data from the NYC Taxi and Limousine Commission (TLC).

3.2 Epoch Partitioning

We would like to apply the following transformation called *epoch partitioning* to the original set of events. It may render the set more suitable for further processing, reduce its size and so forth. Take a resolution $\delta > 0$ and consider the partitioning of the time line $\mathbb{R}$ into a union of disjoint *epochs* $\mathbb{R}_+ = \bigcup_{i=0}^{+\infty} [i\delta, (i+1)\delta)$. We will refer to the interval $[i\delta, (i+1)\delta)$ as *epoch* i. A natural choice of

resolution, which is problem specific, includes a second, a minute, an hour and so on. For an epoch i consider all events $\langle OT_j, CT_j, a_j \rangle$ intersecting with this epoch, i.e., such that $[OT_j, CT_j) \cap [i\delta, (i+1)\delta) \neq \varnothing$. Intersecting events can be classified into the following disjoint groups:

- *float*: the event has started in a previous epoch and is still open at the end of the current epoch, i.e. $OT_j < i\delta < (i+1)\delta \leq CT_j$
- *open*: the event opened but has not closed in the current epoch, i.e. $i\delta \leq OT_j < (i+1)\delta \leq CT_j$
- *closed*: the event closed in the current epoch, but opened before it started, i.e. $OT_j < i\delta \leq CT_j < (i+1)\delta$
- *locked*: the event opened and closed in the current epoch i.e., $i\delta \leq OT_j < CT_j < (i+1)\delta$

We will refer to the *epoch type* $\tau_{i,j} \in \mathcal{T} = \{\text{float}, \text{open}, \text{close}, \text{locked}\}$ of an epoch $[OT_j, CT_j)$ w.r.t. an overlapping event $\langle OT_j, CT_j, a_j \rangle$. This concept describes agents behaviour w.r.t. the epoch partitioning (see Figure 1).

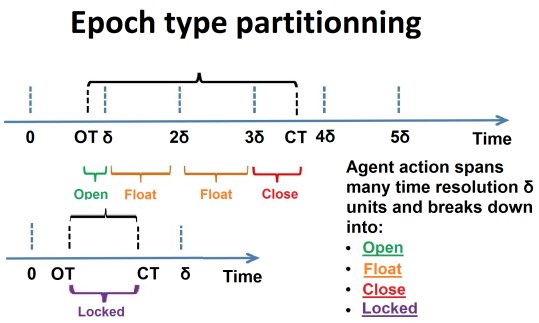

Fig. 1. Visual explanation of epoch types

The epoch partitioning transforms the original set of events $\mathcal{D}$ to a set of tuples $E_i = \langle f_i, o_i, c_i, l_i \rangle$, where i is the epoch number, f_i, o_i, c_i, and l_i are the numbers of floating, open, closed, and locked events, respectively, overlapping with this epoch. In this paper we are only considering partitioning which is time based. In section 7 we present an extension to the time based approach.

The values f_i, o_i, c_i, and l_i can be represented in the following manner:

$$f_i := \sum_j \mathbb{1}_{\{\tau_{i,j}=\text{float}\}} \ , \qquad o_i := \sum_j \mathbb{1}_{\{\tau_{i,j}=\text{open}\}} \ ,$$

$$c_i := \sum_j \mathbb{1}_{\{\tau_{i,j}=\text{close}\}} \ , \qquad l_i := \sum_j \mathbb{1}_{\{\tau_{i,j}=\text{locked}\}} \ .$$

The following variables will be needed below:

$$T_\delta^{(j)} := \frac{CT_j - OT_j}{\delta} \ , \quad S_\delta^{(j)} := \frac{OT_j}{\delta} - \left\lfloor \frac{OT_j}{\delta} \right\rfloor \ .$$

They represent the duration of an event i and its starting time within its epoch in the units of δ.

We can define the aggregate quantities $L_\delta, O_\delta, C_\delta$, and F_δ as follows:

$$L_\delta := \sum_{i=0}^{+\infty} l_i = \sum_j \mathbb{1}_{\{T_\delta^{(j)}+S_\delta^{(j)}<1\}}, \;\; O_\delta := \sum_{i=0}^{+\infty} o_i = \sum_j \mathbb{1}_{\{1\le T_\delta^{(j)}+S_\delta^{(j)}\}}, \;\; C_\delta := \sum_{i=0}^{+\infty} c_i$$

$$F_\delta := \sum_{i=0}^{+\infty} f_i = \sum_j \sum_{i\ge 1} i\mathbb{1}_{\{i+1\le T_\delta^{(j)}+S_\delta^{(j)}<i+2\}} = \sum_j \sum_{i\ge 1} \mathbb{1}_{\{i+1\le T_\delta^{(j)}+S_\delta^{(j)}\}}$$

The choice of resolution δ should provide a trade-off making the epoch representation concise but still meaningful and retaining as much useful information as possible about the original set of events.
In what follows, we define an equilibrium resolution δ^* based on these numbers and study its properties.

4 Equilibrium resolution: a Numerical Example

In this section, we apply epoch partitioning to the example datasets. Table 1 shows the total numbers of floating and locked events at various resolutions.

One can observe that as resolution increases, the number of locked events L_δ increases, while the number of floating events F_δ drops. At some point they become approximately equal (see the rows in bold). This motivates the following definition.

Definition 1. *A resolution δ is called an* empirically equilibrium resolution *for a dataset $\mathcal{D}$ if*

$$L_\delta = F_\delta \;\; .$$

It will be denoted as $\delta^*(\mathcal{D})$.

The equilibrium resolution provides an equilibrium between Locked and Float epoch types. If there is a significant imbalance of either epoch type, the choice of δ must be very small (a lot of floats) or very big (a lot of locked).

Table 1. Number of epoch types at various resolutions

Case Study	Resolution	Float	Locked	Locked/Float
	1 day	$3.2 \cdot 10^6$	$3.7 \cdot 10^6$	1.16
Case Study 1:	**0.91 day**	$\mathbf{3.61 \cdot 10^6}$	$\mathbf{3.61 \cdot 10^6}$	**1.00**
Trade Data	0.5 day	$7.3 \cdot 10^6$	$3.2 \cdot 10^6$	0.44
	15 min	$1.7 \cdot 10^6$	$5.0 \cdot 10^6$	2.9
Case Study 2:	**11.5 min**	$\mathbf{3.7 \cdot 10^6}$	$\mathbf{3.7 \cdot 10^6}$	**1.00**
Taxi Data	10 min	$5 \cdot 10^6$	$3.0 \cdot 10^6$	0.6

Remark 1. An equilibrium resolution for a given sample dataset $\mathcal{D}$ can be expressed as

$$\delta^*(\mathcal{D}) = \arg_\delta \{N = \sum_j \sum_{i\ge 1} \mathbb{1}_{\{T_\delta^{(j)}+S_\delta^{(j)}\ge i\}}\}$$

5 Probabilistic Modelling

5.1 The Probabilistic Approach to equilibrium Resolution

We will develop an analytic approach to finding the equilibrium resolution. It is based on probabilistic modelling.

Let the opening and closing times OT_j and CT_j be random variables, $i = 1, 2, \ldots, N$, where N is the total number of events. Then L_δ and F_δ become random variables. In this framework, one can use the following working definition.

Definition 2. *Let $T_\delta^{(j)} \in \mathbb{L}^1$ for all $j = 1, \ldots, N$. Then a resolution δ^* is* average equilibrium *if $\mathbb{E}L_{\delta^*} = \mathbb{E}F_{\delta^*}$.*

For further analysis we will assume that all $T_\delta^{(j)}$ and $S_\delta^{(j)}$ are independent and identically distributed, $j = 1, 2, \ldots, N$. One can think of the values of $T_\delta^{(j)}$ and $S_\delta^{(j)}$ as of independent realisations of the same variables T_δ and S_δ.

Theorem 1. *A resolution δ is average equilibrium if and only if $\mathbb{E}\lfloor T_\delta + S_\delta \rfloor = 1$.*

Proof.

$$\delta^* = \arg_\delta \left\{ \mathbb{E}\left(\sum_j \mathbb{1}_{\{T_\delta^{(j)} + S_\delta^{(j)} < 1\}} \right) = \mathbb{E}\left(\sum_j \sum_{i \geq 1} \mathbb{1}_{\{i+1 \leq T_\delta^{(j)} + S_\delta^{(j)}\}} \right) \right\}$$

$$= \arg_\delta \left\{ N = N \cdot \mathbb{E}\left(\sum_{i \geq 1} \mathbb{1}_{\{i \leq T_\delta + S_\delta\}} \right) \right\} = \arg_\delta \left\{ 1 = \mathbb{E}\lfloor T_\delta + S_\delta \rfloor \right\}$$

We will now make an assumption on the behaviour of S_δ. It is natural to assume that the starting time of an event is not coordinated with the epochs and therefore S_δ is uniformly distributed on $[0, 1]$, i.e., $S_\delta \sim U(0, 1)$.

Theorem 2. *Let $S_\delta \sim U(0, 1)$. Then*

$$\frac{\mathbb{E}L_\delta}{\mathbb{E}F_\delta} = \frac{\mathbb{E}(1 - T_\delta)_+}{\mathbb{E}(1 - T_\delta)_+ + \mathbb{E}T_\delta - 1}$$

We use the notation $x_+ = \max(x, 0)$.

Proof. For a positive random variable X, let $\lfloor X \rfloor$ be the integer part of X. The expected number of Float epochs can be expressed in the following manner:

$$\mathbb{E}(F_\delta) = \mathbb{E}\left(\sum_j \sum_{i \geq 1} \mathbb{1}_{\{i+1 \leq T_\delta^{(j)} + S_\delta^{(j)})\}} \right) = \sum_j \sum_{i \geq 1} \mathbb{P}(i + 1 \leq T_\delta^{(j)} + S_\delta^{(j)})$$

$$= N \cdot \sum_{i \geq 1} \mathbb{P}(S_\delta + T_\delta \geq i + 1) = N \cdot \sum_{i \geq 1} \mathbb{E}[\mathbb{1}_{S_\delta \geq (i+1) - T_\delta} \cdot \mathbb{1}_{T_\delta \in [i, i+1)} + \mathbb{1}_{T_\delta \geq i+1}]$$

$$= N \cdot \sum_{i \geq 1} \mathbb{E}[(T_\delta - i) \cdot \mathbb{1}_{T_\delta \in [i, i+1)} + \mathbb{1}_{T_\delta \geq i+1}] = N \cdot \Big(\mathbb{E}T_\delta + \mathbb{E}(1 - T_\delta)_+ - 1 \Big)$$

The expected number of Locked epochs can be expressed in the following manner:

$$\mathbb{E}(L_\delta) = \mathbb{E}\left(\sum_j \mathbb{1}_{\{T_\delta^{(j)}+S_\delta^{(j)}<1\}}\right) = N \cdot \mathbb{P}(S_\delta + T_\delta < 1) = N \cdot \mathbb{E}[\mathbb{1}_{S_\delta<1-T_\delta}\mathbb{1}_{T_\delta<1}]$$
$$= N \cdot \mathbb{E}\left[(1-T_\delta)\mathbb{1}_{T_\delta<1}\right] = N \cdot \mathbb{E}(1-T_\delta)_+$$

Corollary 1. *If* $S_\delta \sim U(0,1)$ *then*

$$\delta^* = \mathbb{E}T_1 \ .$$

Proof. From the definition of δ^*:

$$1 = \frac{\mathbb{E}L_{\delta^*}}{\mathbb{E}F_{\delta^*}} = \frac{\mathbb{E}(1-T_{\delta^*})_+}{\mathbb{E}(1-T_{\delta^*})_+ + \mathbb{E}T_{\delta^*} - 1} \ .$$

This is equivalent to: $0 = \mathbb{E}T_{\delta^*} - 1$. Since $T_{\delta^*} = \frac{T_1}{\delta^*}$ therefore $\delta^* = \mathbb{E}T_1$.

Recall that $T_1^{(j)} = CT_j - OT_j$ and thus $\mathbb{E}T_1$ is the expected duration of an event. The corollary implies that our definition of the equilibrium resolution has a natural intuitive interpretation. Corollary 1 provides an easy way for calculating the equilibrium resolution. While calculating $\delta^*(D)$ from the definition would require a solver, calculating the average is simple and straightforward. In the case studies we fit distributions to better understand the nature of the processes, but this is clearly not necessary to find the equilibrium resolution.

Instead of 1, one could be interested in a ratio $c > 0$, hence the following generalisations of the empirical and average equilibrium resolution.

Definition 3.

$${}^c\delta^*(\mathcal{D}) := \arg_\delta\left\{\frac{L_\delta}{F_\delta} = c\right\} \ , \ {}^c\delta^* := \arg_\delta\left\{\frac{\mathbb{E}L_\delta}{\mathbb{E}F_\delta} = c\right\}$$

Using this notation one can write $\delta^*(\mathcal{D}) = {}^1\delta^*(\mathcal{D})$ and $\delta^* = {}^1\delta^*$.

Corollary 1 does not generalise to an arbitrary ${}^c\delta^*$ straightforwardly, but one can find the equilibrium value numerically using the ratio from Theorem 2.

5.2 Approximating equilibrium resolution

For practical reasons one may be interested in the the resolution to be equilibrium up to some error ϵ. This motivates the following definition.

Definition 4. *A resolution* δ *is* approximately c-equilibrium up to ε, *if*

$$\left|\frac{\mathbb{E}L_\delta}{\mathbb{E}F_\delta} - c\right| \leq \varepsilon \ .$$

Hence two handy approximations ${}_\epsilon\delta^+_*$ *and* ${}_\epsilon\delta^-_*$ *are such that*

$$ {}^c_\epsilon\delta^*_- = \arg_\delta \left\{ \frac{\mathbb{E}L_\delta}{\mathbb{E}F_\delta} = c(1-\epsilon) \right\} \ , \ {}^c_\epsilon\delta^*_+ = \arg_\delta \left\{ \frac{\mathbb{E}L_\delta}{\mathbb{E}F_\delta} = c(1+\epsilon) \right\} $$

and thus the resolutions ${}_\epsilon\delta^*_+$ *and* ${}_\epsilon\delta^*_-$ *are approximately equilibrium up to* ϵ *since*

$$ {}^c_\epsilon\delta^*_- \leq {}^c\delta^* \leq {}^c_\epsilon\delta^*_+ $$

5.3 Equilibrium resolution for lognormaly distributed agents action duration

The analysis of case studies (see section 6 and [1]) suggests the duration of agent's action is often lognomal. We derive formulas for this distribution and its truncated counterpart.

Let $T_1 \sim \text{LogNormal}(\mu, \sigma)$. Then we have $T_\delta = \frac{T_1}{\delta} \sim \text{LogNormal}(\mu - \ln\delta, \sigma) = \text{LogNormal}(\mu_\delta, \sigma)$, where $\mu_\sigma = \mu - \ln\sigma$. With use of properties of this law the equilibrium resolution estimators are

$$ \delta^* = e^{\mu + \frac{\sigma^2}{2}} \ , \ {}_\epsilon\delta^*_\pm = \arg_\delta \left\{ \frac{A^\delta}{A^\delta + B^\delta - 1} = 1 \pm \epsilon \right\} \ . $$

where

$$ A^\delta = \Phi\left[\beta_{1,\delta}\right] - e^{\mu_\delta + \frac{\sigma^2}{2}} \cdot \Phi\left[-\alpha_{1,\delta,1}\right] \ , \ B^\delta = e^{\mu_\delta + \frac{\sigma^2}{2}} \ . $$

Φ is the normal cumulative distribution function and

$$ \alpha_{n,\delta,k} = \frac{\mu_\delta + k\sigma^2 - \ln n}{\sigma} \ , \ \beta_{n,\delta} = \frac{\ln n - \mu_\delta}{\sigma} \ . $$

5.4 Equilibrium resolution for truncated lognormal distributed agents action duration

If $T_\delta = \frac{T_1}{\delta} \sim \text{TruncatedLogNormal}(\mu_\delta, \sigma, \frac{l}{\delta}, \frac{u}{\delta})$. Using the properties of this distribution, we can write the equilibrium resolution estimators as

$$ \delta^* = \arg_\delta \left\{ B^\delta - C^\delta = 0 \right\} \ , \ {}_\epsilon\delta^*_\pm = \arg_\delta \left\{ \frac{A^\delta}{A^\delta + B^\delta - C^\delta} = 1 \pm \epsilon \right\} $$

where

$$ A^\delta = \Phi\left[\beta_{1,\delta}\right] - \Phi\left[\beta_{\frac{l}{\delta},\delta}\right] - e^{\mu_\delta + \frac{\sigma^2}{2}} \left\{ \Phi\left[-\alpha_{1,\delta,1}\right] - \Phi\left[-\alpha_{\frac{l}{\delta},\delta,1}\right] \right\} $$

$$ B^\delta = e^{\mu_\delta + \frac{\sigma^2}{2}} \cdot \left\{ \Phi\left[-\alpha_{\frac{u}{\delta},\delta,1}\right] - \Phi\left[-\alpha_{\frac{l}{\delta},\delta,1}\right] \right\} \ , \ C^\delta = \Phi\left[\beta_{\frac{u}{\delta},\delta}\right] - \Phi\left[\beta_{\frac{l}{\delta},\delta}\right] $$

6 Application on case study data

6.1 Case Study: Trading Data

The first case study considers financial data gathered from the client trades of a retail foreign exchange broker. The data comprises irregular time series data pertaining to opening and closing trade times of client trades. The exogenous data stream in this case describes the price of the underlying currency pair which is then converted in USD dollars.

The FX broker is the "middleman" – it links to the best liquidity providers (or LPs), such as investment banks, and the LP's stream "tradeable" currency prices to the broker. The FX broker then passes these prices on to its thousands of clients worldwide all of whom can trade from their mobile phones or personal computers at the click of a button. A typical retail broker will provide their clients with online trading platform software such as MetaTrader 4 where clients may place trades, monitor positions, track both historic and live movements in prices, and access the latest world economic news.

Table 2. Example of client order data

OpenTime	Client	Amount	Sign	Symbol	OpenPrice	CloseTime	ClosePrice
03/01/2017 03:24	82	8232	1	EUR/CAD	1.40524	03/01/2017 03:48	1.40548
03/01/2017 03:24	82	11000	-1	EUR/CHF	1.07079	03/01/2017 05:16	1.07096

The publicly available dataset [7] consists of order data and quote data (exogenous) for the foreign exchange broker data.

Table 3. Example of price quote data

Datetime	Symbol	OpenUsdMult	ContraUsdMult	Price
01/03/2017 03:30	AUD/JPY	0.72239	0.00852	84.11
01/03/2017 03:30	EUR/CHF	0.04713	0.97813	1.0705

In order to find and analyse the equilibrium resolutions for different values of c, a modelling stage has to be done. For that purpose we tried to fit 85 distributions from `scipy.stats` library. By ranking the fit by the smallest Akaike Information Criterion (AIC) we identified that the truncated lognormal distribution seems to fit reasonably well. A comparison between complementary cumulative distribution functions of lognormal fit and empirical data justified the choice of the truncation. Standard maximum likelihood estimators were calculated on the truncated data returning $\mu = 4.583$, $\sigma = 2.55$, $l = 0$, and $u \approx 6 \cdot 10^4$.

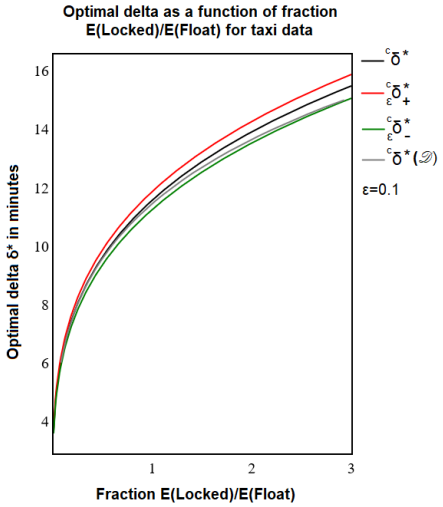

Fig. 2. The relationship between $\frac{\mathbb{E}L_\delta}{\mathbb{E}F_\delta}$ and ${}^c\delta^*$ for the taxi data.

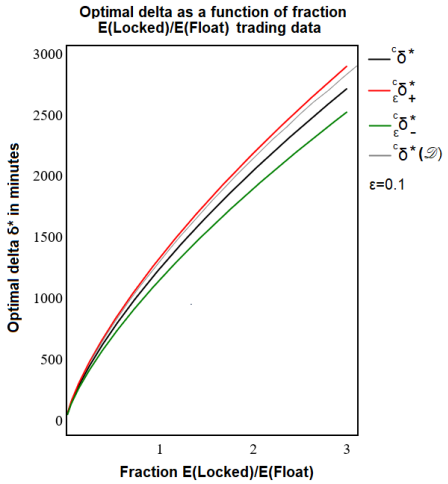

Fig. 3. The relationship between $\frac{\mathbb{E}L_\delta}{\mathbb{E}F_\delta}$ and ${}^c\delta^*$ for the trading data.

In Figure 3 the equilibrium resolution is calculated, which is around 0.86 days. This is close to the sample-wise equilibrium resolution (0.91 days). Again we note that the empirical equilibrium resolution $\delta^*(\mathcal{D})$ is within the approximation bounds, ${}^c_\epsilon\delta_-$ and ${}^c_\epsilon\delta_+$, of ϵ equal 0.1.

Once applied with equilibrium resolution to the source data, the resulting target data was used in the analysis of [1].

6.2 Case Study: NYC Taxi Journey Data

For the second case study, which is about taxi journeys around NYC, we obtained similar results. The dataset [11] comprises irregular time series data pertaining to the pick-up and drop-off time stamps of taxi journeys undertaken by individual taxis. For the taxi ride duration the lognormal fit appeared to be satisfactory and MLE estimators are $\hat{\mu} = 6.31$ and $\hat{\sigma} = 0.69$. In Figure 2 one can see that for different values of parameter c, the resolution ${}^c\delta^*$ stays close to the empirical equilibrium resolution ${}^c\delta^*(\mathcal{D})$.

7 Partitioning based on monitoring functions

In this section we extend the notion of epoch partitioning. So far we have only considered the time-based approach. It is natural to partition the dataset with respect to the duration of events as well as the attributes. For instance a market maker may want to efficiently partition a dataset in order to gain insight into the evolution of their profit and loss through time.

If we calculated the PnL over the whole order period we would have no insight into what happened during the order. Indeed the longer the lifetime of a trade, the more likely that its PnL will fluctuate due to other exogenous factors such as price movement, economic news releases and related temporal events. Figure 4 illustrates this point, showing quite a stark contrast in client trading PnL profile when all price data is used (full resolution - pink line), to the profile shown when PnL is calculated using only the price data available at the open and close of a trade (no resolution - purple line). Neither profile is ideal; a simplistic approach would be to use resolution that derives from periods such as hours or minutes, but one would need a rule of thumb.

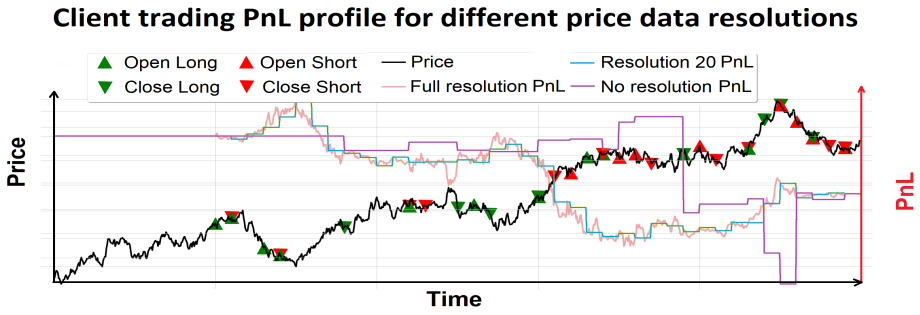

Fig. 4. Client trading PnL profiles when all price data used (full resolution - pink line); when price data only at the open and close of a trade is used (no resolution - purple line); and when a resolution of 20 units of time according to the method proposed in this study is used (blue line). The example is based on 30 randomly selected trades with random positions (Long/Short) over 1000 units of time.

Let the epoch partitioning reduce the original set of events $\mathcal{D}$ to a set of tuples $E_i = \langle f_i, o_i, c_i, l_i, d_i^{(1)}, , ..., d_i^{(K)} \rangle$, where i is the epoch number, f_i, o_i, c_i, and l_i are the numbers of floating, open, closed, and locked events, respectively, overlapping with this epoch, and $d_i^{(k)}$ are derived fields, i.e., functions of attributes calculated on the basis of the events intersecting with the epoch (see Figure 5). Every $d_i^{(k)}$ will be an output of what we will refer to from now on as a *monitoring function*.

For $k = 1, 2, \ldots, K$ let

$$d_i^{(k)} = F_k(\phi_{k,i}(\langle OT_{j_1}, CT_{j_1}, a_{j_1} \rangle, ..., \phi_{k,i}(\langle OT_{j_I}, CT_{j_I}, a_{j_I} \rangle)) \ .$$

where $j_1, j_2, \ldots, j_I$ are the events overlapping with $[\delta i, \delta(i+1))$, F_k is an aggregation function and $\phi_{k,i}$ is given by

$$\phi_{k,i}(\langle OT_j, CT_j, a_j \rangle) := \begin{cases} \phi_k(a_j, i\delta, (i+1)\delta) & \text{if } \tau_{i,j} = \text{float} \\ \phi_k(a_j, i\delta, CT_j) & \text{if } \tau_{i,j} = \text{close} \\ \phi_k(a_j, OT_j, (i+1)\delta) & \text{if } \tau_{i,j} = \text{open} \\ \phi_k(a_j, OT_j, CT_j) & \text{if } \tau_{i,j} = \text{locked} \end{cases}$$

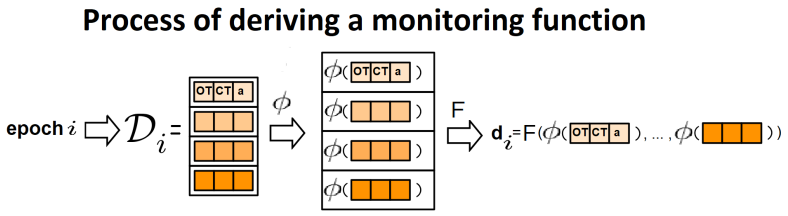

Fig. 5. The process of deriving a monitoring function in epoch partitioning. A function ϕ is applied to every prolonged events in the considered dataset and then an aggregation function F transforms the latter data into an element of E_i.

The derived field monitoring functions are defined by the user and f_i, o_i, c_i, and l_i are defined as before.

Table 4. Illustration of epoch Epoch type partitioning procedure on raw data.

ID	OT	CT	OP	CP
1	2.5	6.5	10	20
2	6.2	6.7	4	8

ϕ

Epoch	ID	Open	Float	Locked	Close	OpenPnL	FloatPnL	LockedPnL	ClosePnL
2	1	1	0	0	0	1.25	0	0	0
3	1	0	1	0	0	0	2.5	0	0
4	1	0	1	0	0	0	2.5	0	0
5	1	0	1	0	0	0	2.5	0	0
6	1	0	0	0	1	0	0	0	1.25
6	2	0	0	1	0	0	0	4	0

F

Epoch	Open	Float	Locked	Close	OpenPnL	FloatPnL	LockedPnL	ClosePnL
2	1	0	0	0	1.25	0	0	0
3	0	1	0	0	0	2.5	0	0
4	0	1	0	0	0	2.5	0	0
5	0	1	0	0	0	2.5	0	0
6	0	0	1	1	1	0	4	1.25

To be precise, a USD normalised profit and loss for each order is required. Therefore, the PnL monitoring function is calculated over interval $[\delta i, \delta(i+1))$ for a currency s as follows:

$$\mathrm{PnL}_i^{(s)} := \sum_{l=1}^{I} \phi_{s,i}(\langle OT_{j_l}, CT_{j_l}, a_{j_l} \rangle)$$

where F_s is a simple sum, $\phi_{s,i}(\langle OT_j, CT_j, a_j \rangle)$ depends on the $\tau_{i,j}$ and ϕ_s is given by

$$\phi_s(a_j, t_1, t_2) := (P_{t_1}(s) - P_{t_2}(s)) \cdot CM_{t_2}(s) \cdot SD_j \cdot AM_j \ ,$$

where notation $P_t(s)$ represents the price of the symbol s at time t, AM_j refers to the number of units for the underlying the order was for and SD_j refers to whether the order was either long or short, +1 or -1 respectively.

However, all PnL made as a result of trading currencies will be paid in the contra currency of the symbol pair traded. This is often not the most useful figure when trying to conduct financial analysis of a collection of trades. It is standard practice to normalise the PnL's by calculating the value in some common currency which is typically USD. This is achieved by multiplying the PnL with the exchange rate between the contra currency and USD at the close time of the order. We will refer to this as the contra multiplier using the notation, $CM_t(s)$ to be the contra multiplier at time t of symbol s.

An example of the epoch partitioning process is visible in Table 4, where the monitoring function is the Profit and Loss (PnL). The raw data consists of only two rows of data with resolution equal to 1 unit where ID is the identification number, OP and CP are opening and closing prices. For the sake of simplicity the price is supposed here to evolve linearly.

Until now the equilibrium resolution exclusively took into account the time dependency of agents behaviour, however it may be of interest to define a binding function merging the latter with a target monitoring function (or functions). Taking the example of the broker we propose an analogical definition of equilibrium resolution which makes the mean of the PnL to be equal for Locked an Float epoch events.

Definition 5. *The equilibrium resolution for a PnL monitoring function is:*

$$\delta^*_{PnL} = \arg_\delta \left\{ \mathbb{E}\left[PnL \mathbb{1}_{\{T_\delta + S_\delta < 1\}} \right] = \mathbb{E}\left[\sum_{i \geq 1} PnL \mathbb{1}_{\{i+1 \leq T_\delta + S_\delta < i+2\}} \right] \right\}$$

Deriving the explicit formula is not that simple but we will use estimated values and we assume trading duration is identically distributed. The empirical results are shown in Figure 6. We obtain a set of possible equilibrium resolutions and observe that a good candidate for equilibrium resolution is around 10 minutes (since the expected PnL is less noisy in this time-neighbourhood) which is more insightful for a broker than equilibrium time resolution obtained with respect to agent event duration (1200 minutes).

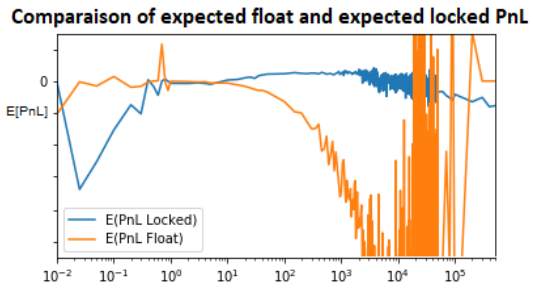

Fig. 6. $\mathbb{E}[PnL_\delta^{\text{Locked}}]$ vs. $\mathbb{E}[PnL_\delta^{\text{Float}}]$

8 Conclusion

We proposed definitions of equilibrium resolution for epoch partitioning of a dataset of prolonged events based on the equilibrium of locked and float events and carried out theoretical analysis using probabilistic assumptions. The study of two real-life datasets showed that the theoretical results are in agreement with empirical observations. We have thus developed a novel method for determining the equilibrium resolution based on events duration and on a monitoring function, which can be considered as an ad hoc approach for finding a meaningful partitioning of irregularly-sampled time series data.

References

1. Al-baghdadi, N., Wisniewski, W., Lindsay, D., Lindsay, S., Kalnishkan, Y., Watkins, C.: Structuring time series data to gain insight into agent behaviour. In: 2019 IEEE International Conference on Big Data. pp. 5480–5490 (2019)
2. Al-Hmouz, R., Pedrycz, W., Balamash, A., Morfeq, A.: Granular representation schemes of time series: A study in an optimal allocation of information granularity. In: 2013 IEEE Symposium on Foundations of Computational Intelligence (FOCI). pp. 44–51 (2013)
3. Arandia, E., Eck, B., McKenna, S.: The effect of temporal resolution on the accuracy of forecasting models for total system demand. Procedia Engineering **89**, 916 – 925 (2014), 16th Water Distribution System Analysis Conference, WDSA2014
4. Athanasopoulos, G., Hyndman, R.J., Kourentzes, N., Petropoulos, F.: Forecasting with temporal hierarchies. European Journal of Operational Research **262**(1), 60 – 74 (2017)
5. Challet, D., Chicheportiche, R., Lallouache, M., Kassibrakis, S.: Statistically validated leadlag networks and inventory prediction in the foreign exchange market. Advances in Complex Systems (Dec 2018), 22 pages, 15 figures
6. Krings, G., Karsai, M., Bernhardsson, S., Blondel, V.D., Saramäki, J.: Effects of time window size and placement on the structure of an aggregated communication network. EPJ Data Science **1**(1) (May 2012)
7. Lindsay, D.: Fxclienttrades. Available at https://www.kaggle.com/davidlindsay1979/toptradingclientdata/kernels
8. Nason, G.P., Powell, B., Elliott, D., Smith, P.A.: Should we sample a time series more frequently?: decision support via multirate spectrum estimation. Journal of the Royal Statistical Society: Series A (Statistics in Society) **180**(2), 353–407 (2017)
9. Sulo, R., Berger-Wolf, T., Grossman, R.: Meaningful selection of temporal resolution for dynamic networks. In: Proceedings of the Eighth Workshop on Mining and Learning with Graphs. p. 127–136. MLG '10, Association for Computing Machinery, New York, NY, USA (2010)
10. Tumminello, M., Miccichè, S., Lillo, F., Piilo, J., Mantegna, R.N.: Statistically validated networks in bipartite complex systems. PLoS ONE **6**(3), e17994 (2011)
11. Wong, C.: Nyc taxi trips. Available at http://www.andresmh.com/nyctaxitrips/
12. Wu, X., Shi, B., Dong, Y., Huang, C., Faust, L., Chawla, N.: Restful: Resolution-aware forecasting of behavioral time series data. pp. 1073–1082 (10 2018)
13. Xue-dong, G., Chen, H.: The method of time granularity determination on time series based on structural similarity measure algorithm. In: The International Symposium on the Analytic Hierarchy Process (ISAHP) (2016)