1. Introduction

This is an adaptive filter that learns from the data while sequentially cleaning the feed. It continuously updates its information base in real time. That means all rejection decisions are made solely base on the time series. This filter will protect us from outliers provided by reliable providers.

The filter will be applied on output of black/white list filter of our Full feed (multiple ticks each second).

2. Intuition

The main idea of the filter is get a price prediction from historical ticks, use this prediction as a centre point, treat any new ticks within certain amount of standard deviation to the centre point as acceptable and reject any tick that is out of the standard deviation boundary.

We set a cap to the percentage of ticks the filter can reject in the previous look back window. When the rejection amount hit the cap, we force the next tick to pass. This feature help the filter to react very fast when a jump or drop happens in the feed.

3. Rejection decision

To get better distribution features, we convert ticks into their nature log. S in this document means nature log of spot.

Every time a new tick comes in, there should be these data already in the filter:
a look back window size to control how many seconds to look back, all the nature logs of ticks in look back window, trust weights of all nature logs of ticks in the look back window and current volatility.

These terms will be explained in the following sections.

3.1 Look back window

After testing and analysing, we set look back window equals to 4 seconds for all the underlyings.
Here is a example of how to calculate this window:
First, t is time in second and we could have multiple ticks within one second. If the current tick came between t_n and $t_{n+1,}$ this 4 seconds' window should contain ticks from time point t_{n-4} to the tick previous to the current tick, inclusive. That means the window contains ticks comes in previous 4 seconds plus ticks come in this second but before the current tick.

To guarantee we will have significant number of ticks in our look back window but not too many ticks to drag our efficiency, we have a cap and a floor to tick number in the window: The look back window should not contain more than 20 or less than 6 ticks. If the tick number in the look back window is less than 6, look back further until we have 6 ticks.

3.2 S values in look back window

Set the array $[S_{1,} S_{2,} S_{3,} S_4 \ldots S_{n-1}]$, in which S_1 is the oldest one.

3.3 Trust weights in look back window

The definition will be presented in Reject test part.
Set the array $[T_{1,} T_{2,} T_{3,} T_4 \ldots T_{n-1}]$, in which T_1 is trust weight of the oldest tick.

3.4 Get historical average

Historical average: price prediction calculated from look back window.

Since we want the historical average to represent the current market price, we give higher weight to the newer S. The newest S value from look back window, S_n, will have weight ½ * $T_{n,}$ the second newest S value, $S_{n-1,}$ will have weight ¼ * T_{n-1} and the next one S_{n-2} will have weight 1/8 * T_{n-2} and so on. In this and combination, factors, like ½, means we want the price prediction to be driven mainly by the S values that come closer to current time point; trust weight, such as $T_{n,}$ means we assign weight to each S base on how much we trust it. Historical average is a weighted average of all S values from look back window. The formula is as follow:

$$HA = \frac{\sum_{k=1}^{n-1} 2^{-(n-k)} * T_k * S_k}{\sum_{k=1}^{n-1} 2^{-(n-k)} * T_k}$$

where HA means historical average. The denominator is to guarantee that the sum of all weight always equals to 1.

3.5 Reject test

The next step will be calculating reject test value using new S value, historical average and volatility.

Reject test measures how many volatilities is new S value from the historical average. Formula is

$$R = \frac{|S_n - HA|}{volatility}$$

where R means reject test value.

After having R, we compare it with reject criteria (equals to 4 currently), the maximum amount of volatilities the new S value can diverse from the historical average. Every tick that has a reject test value greater than reject criteria will be rejected.

Base on the reject test and reject criteria, every S will have a trust weight (represented by T in section 3.3), the weight imply how much we trust an S value. Trust weight is decided by formula

$$T_n = \frac{1}{1+\left({}^{R}/{}_{C}\right)^8}$$

where T is trust weight, R is reject test value and C is reject criteria.

The method of calculate volatility is complicated and will be presented in the next section.

4. Volatility

The volatility we use in this filter is calculated from MAD (mean absolute deviation) and a convert factor.

One proved theory is, for normal distribution, there is a constant ratio between MAD (mean absolute deviation) and standard deviation. The formula is: *MAD/SD* $=\sqrt{2/\pi}$. We didn't prove our feeds are normal distributed, but we test across all the underlyings we currently offering and this ratio is fairly stable.

In this filter, we calculate MAD first and divided by the constant factor to get volatility.

$$volatility_n = MAD_n \,/\, \sqrt{2/\pi}$$

The reason of doing this is to keep volatility more stable. By definition, standard deviation puts more weight on the outliers than MAD and lead to a less smooth result.

4.1 MAD calculation
The MAD in this filter is an EMA of previous absolute differences between S values. In iterative formula, it is a weighted average between previous MAD and the new absolute difference.

4.2 Absolute difference
Currently, the absolute difference step size is 5. Take this step size as an example: For a new S, say S_n, check whether S_{n-5} is accepted, if it is, the absolute difference will be calculate as

$$AD_n = \frac{|S_n - S_{n-5}|}{\sqrt{5}}$$

where AD means absolute difference; if S_{n-5} is not valid, keep looking back till find the first valid S, say S_{n-i}, and the absolute difference equals to

$$AD_n = \frac{|S_n - S_{n-i}|}{\sqrt{i}}$$

4.3 Weight for MAD
The weight is calculated from two parts: tick density and trust weight.

Tick density is average tick number per second. We calculate this from 60 previous seconds started 3 second before the current time point. This means if current time is t_n, we count the total tick number from t_{n-3} to t_{n-63} and divided by 60.

$$\text{density} = \frac{N}{60}$$

where N is the total tick number in the counting period.

Density contains the information of how long it will take for a piece of new information (a new absolute difference) to come. This means it also decide how fast the old information should decay. The decay rate of old MAD is $exp(-DS\ /\ density)$, which makes the weight for new absolute difference *1 -* $exp(-DS\ /\ density)$. In the formula, DS represents decay speed. Different decay speeds affect how fast the the MAD reacts to the market shock.

Trust weight was introduced in 3.5.

To calculate the final weight, we first multiply the decay and trust weight of absolute difference

$$\mu_n = T_{n-1} * (1 - e^{-DS/density})$$

where μ_n is the weight for new absolute difference and DS is decay speed. T is trust weight as we mentioned in previous sections.

Weight for old MAD is

$$1 - \mu_n = (1 - T_{n-1}) + T_{n-1} * e^{-DS/density}$$

Using different decay speed, we created three MAD which have different reacting speed to the market movement. In the current example, Fast decay = 0.03. This decay rate make the MAD react fast to the market shock. Mid decay = 0.01; Slow decay = 0.003. Current setting can be changed. We always take the largest of three to be our final MAD and to convert to volatility.

4.4 Weighted average

To get the MAD, simply insert values into formula

$$MAD_n = MAD_{n-1} * (1 - \mu_n) + AD_{n-1} * \mu_n$$

where AD is absolute difference defined in section4.2.

5. Test for exceptions

If one of following two situations was fulfilled, the new tick will be forced to pass:

1) No valid tick in the look back window
2) The amount of ticks that were rejected in the look back window is greater or equal to filter amount cap of the totally amount in the look back window. Current setting of filter amount cap is 20%

When the tick was forced to pass, trust weight will be set to 1.

6. Build up time

The filter need a build-up time which is configurable.

In the build-up time, the filter will get S series to calculate the absolute differences. The first one will be AD_5 assuming the data start from index 0. To calculate MAD in the build-up time, we take average of absolute differences within 20% and 80% quantiles. We then convert this MAD into volatility to use in the first reject test. In the whole build up period, the filter accepts every tick, and that means all trust weight equal to one.

7. Configurable parameters

reject criteria
absolute difference step size
look back window
build up time
filter amount cap
fast decay, mid decay, slow decay