An Algorithm for Determining Price Channels

Nicholas Hausch

Independent Research

July 2024

1 Introduction

Price channels are an important tool in technical analysis. A price channel refers to a set of two parallel lines between which the price of an asset oscillates. The upper and lower lines are formed as the asset's price is buffeted by supply and demand, and the direction of the lines indicates trend. In a trading context, one could buy at the lower line and sell at the upper line, or use deviations from the price channel to identify breakouts. Identifying price channels manually with a pencil and paper is fairly straightforward. Doing so programmatically on the other hand requires a robust and flexible algorithm. The benefit of using an algorithm is that price channels can be generated *en masse*, eliminating the need for human labor and creating the foundation for an automated trading strategy. Additionally, a mathematically calculated price channel is more precise than one drawn by a human, and can be reproduced. This article describes a method for determining price channels using regression lines and integrals.

2 Definitions

Consider a price signal $y_t = f(t)$ where $t \in (0, N)$ is the discrete index of the price data $f(t)$. Define the upper and lower boundary lines of a price channel as $u_t = mt + b_u$ and $l_T = mt + b_l$ respectively. A price channel can start at any point t in time. It could also end at any t after the start point, however for this algorithm we assume that every price channel includes t_N in order to generate results that are actionable. With that assumption we define PC_{n,m,b_u,b_l} as the price channel composed of an upper and a lower boundary line where $t \in (n, N)$. The boundary lines should satisfy the conditions $u_t \geq y_t$ and $l_t \leq y_t$ for $t \in (n, N)$.

3 Method

The steps of the algorithm are as follows:

- Determine the least squares regression lines for y_t where $t \in (n, N)$ and $n = 0, 1, \ldots, N-1$.
- Select a subset of the regression lines where the derivative of the squared error sum is close to 0.
- Use the subset of regression lines to generate upper and lower boundary lines.
- Compute the areas between the price signal and each boundary line then select price channels where the areas closely match.
- Eliminate near duplicates.

We now explore each step in more detail. First we find the slope m of the least squares regression line for each start point $t \in (0, N-1)$, fixing the end point at N. This gives a sequence

$$[E] = [E_0, E_1, \ldots, E_{N-1}]$$

where E_n is the total squared error for the corresponding regression line. Next we compute the first derivative sequence

$$[\Delta_E] = [\Delta_{E_1}, \Delta_{E_1}, \ldots, \Delta_{E_{N-1}}]$$

where $\Delta_{E_n} = E_n - E_{n-1}$. By choosing lines where $\Delta_{E_n} \approx 0$ we obtain a list of regression lines that represent the data well. These lines generally start around the center of the price signal, rather than closer to one of the local minima or maxima.

Figure 1: Least squares regression lines where the first derivative of the squared error is close to 0

Next we use the selected regression lines seen in Figure 1 to generate possible price channels. For each regression line $r_t = mt + b_0$ we begin by making two copies for the initial upper and lower boundary lines. Thus we have

$$u_t = mt + b_u$$

$$l_t = mt + b_l$$

where $b_u = b_l = b_0$. Now we increase b_u and decrease b_l by some step value until we satisfy the conditions $u_t \geq y_t$ and $l_t \leq y_t$. The step value should be scaled based on the stock price, so we use $b_0\lambda$ where $\lambda \leq 1/1000$.

Finally we evaluate each of the generated price channels. We use the intuition that within a price channel, upward and downward price movement is roughly equal. Therefore we can compute the discrete sum of the area between the price signal and each of the boundary lines, then select price channels where the areas are roughly equal. Thus we compute for each PC_n:

$$\Sigma_{u_t} = \sum_{t=n}^{N} u_t - y_t$$

$$\Sigma_{l_t} = \sum_{t=n}^{N} y_t - l_t$$

then select price channels where $abs(\Sigma_{u_t} - \Sigma_{l_t}) < \epsilon$ where ϵ is some small multiple of b_0. Figure 2 shows the set of three price channels that satisfy that condition. We now have essentially the final set of price channels for the price signal. Note that it is possible that no price channels are found.

Figure 2: Generated price channels after evaluation

The final step is to eliminate near duplicates. This can be accomplished by iterating over the price channels in order and eliminating an entry if $m, b_l, and b_l$ all match an existing entry within some small threshold. In Figure 3 we see that one of the price channels is removed since it's slope and intercepts of the upper and lower boundary lines closely match another channel.

Figure 3: Generated price channels after near duplicate elimination

4 Conclusion

In this article we have outlined an algorithm for determining price channels for a given price signal. We have limited our search to the time frames that include the latest data point. One possible adjustment to this process could be allowing for some degree of outliers outside the price channel, which could increase the frequency of good results. Further analysis on the fit of each price channel can also be performed. One could examine the number of times the price signal comes close to touching the upper and lower boundary lines. Thoughtful analysis of the price channels generated by this process can lead to a lucrative trading strategy.

5 Conflicts of Interest

None declared.

6 Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

7 Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.