Statistics and Its Interface Volume 11 (2018) 441–453

# Discovering stock chart patterns by statistical estimation and inference

Hoang Tran and Yiyuan She*

Statistical modeling of stock price data is challenging due to heteroskedasticity, heavy-tails and outliers. These issues can be particularly relevant to the technical analysis practitioner who extracts trading signals from geometric patterns in prices. In this work, we propose a new method called **N**on-Parametric **O**utlier **I**dentification and **S**moothing (NOIS), which robustly smooths stock prices, automatically detects outliers and constructs pointwise confidence bands around the resulting curves. In real-world examples of high-frequency data, NOIS successfully detects erroneous prices as outliers and uncovers borderline cases for further study. NOIS can also highlight notable features and reveal new insights in inter-day chart patterns.

Keywords and phrases: Outlier detection, Confidence bands, Technical analysis, High-frequency data.

## 1. INTRODUCTION

Technical analysis or "charting" is a financial discipline that attempts to extract information from the stock market through the use of heuristic rules and geometric patterns. Practitioners have traditionally performed technical analysis by examining stock charts, identifying patterns in the price trend that have historically yielded positive returns, then placing a market order [12]. Figure 1 is an example of a continuation pattern, which the trader uses as a signal for a continuing upward price trend.

A crucial aspect of identifying a pattern such as the cup and handle is that the trader uses the shape of a stock price series as the basis for his or her decisions. The inherent noise and non-linearity in stock price data can make it difficult to determine the salient features of a series. Thus, it could be preferable for the practitioner to instead use a curve with the noise "smoothed out". [15] described a statistical approach to this problem in which the non-parametric kernel smoothing method was used to estimate the underlying curve. If outliers are present in a stock chart, a curve fitted by the usual non-robust kernel smoother would be pulled away from the majority of the data in the direction of the anomalous points. The decisions of a technical analysis trader could then become unduly influenced. In addition, the outliers themselves can be informative and it can be time consuming to manually identify them. Therefore, we wish to systematically construct a robust function estimate while automatically detecting outliers.

*Corresponding author.

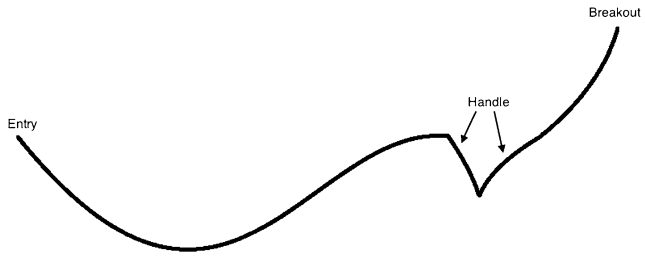


*Figure 1. Cup and handle chart pattern.*

One generally accepted definition of outliers is that they are data points appearing to deviate markedly from other members of the sample in which they occur [1]. In applying this definition to stock price data, we characterize outliers as uncharacteristic deviations in the price level that are separate from the majority of the data. These outliers are a pervasive problem, as financial data is often littered with individual observations that have seemingly been affected by unknown or external events [20]. This can be especially problematic in high-frequency or intra-day price data, where it is estimated that as many as $2\% - 3\%$ of the observations are outliers [4].

Figure 2 is an example of what might be considered a cup and handle pattern with one outlier in high-frequency trading prices of McDonald's (MCD). The solid line is a non-robust kernel smoothing estimate. There is a peak in the fitted curve due to the outlier which occurs about \$0.25 off the prevailing price level. Although it seems trivial to manually identify the outlier in this figure, due to the sheer volume of high-frequency data we require a method that can parse big data and *automatically* identify anomalies.

The contributions of this work are three-fold. First, our method robustly estimates the curve underlying stock price data and automatically detects outliers. We refer to our method as **N**on-Parametric **O**utlier **I**dentification and **S**moothing (NOIS). In contrast to some previous techniques that require multiple tuning parameters dependent on the market type and instrument [20], NOIS uses only two parameters with straightforward interpretations that can be intuitively set by a practitioner. Additionally, NOIS not only identifies outliers but also produces a smoothed curve

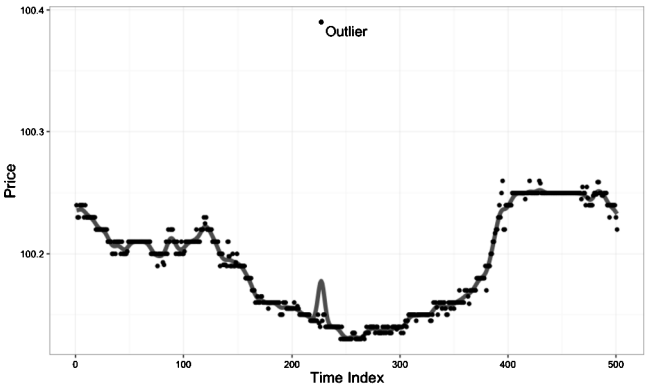


*Figure 2. McDonald's trading prices on Jan. 3rd, 2012 from 10:18:18 to 10:20:00 am.*

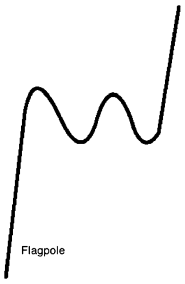


*Figure 3. Flag chart pattern.*

to be used for further applications such as stock chart pattern detection. Second, we use non-parametric techniques to construct pointwise confidence bands around the curve estimated by NOIS. These bands address the heteroskedasticity of financial data by providing a visual measure of the uncertainty in the smoothed curve. Taken together, we refer to a NOIS curve and its confidence bands as a **NOIS bundle**. Finally, we use real-world examples and simulation studies to demonstrate the efficacy of the outlier detection component of NOIS. We also reveal how NOIS bundles can assist the technical analysis practitioner in extracting additional information from previously identified patterns. In this work we focus on applications to stock price data, but NOIS can be used to analyze time series data of nearly any financial instrument, including derivatives and volatility.

It is perhaps best to illustrate the proposed method on a real-world example. The flag (cf. Figure 3) is a continuation pattern that can be used to confirm a prior upward or downward trend. It is typically characterized by accelerating prices (sometimes referred to as the flagpoles) on either side of price movement within two downward sloping trendlines. [26] identified a flag pattern on daily prices of Hewlett-Packard (HPQ) between April and August of 1999. Figure 4 presents a NOIS bundle fit to this data; the solid line is the NOIS function estimate, confidence intervals are shaded and detected outliers are marked by a triangle. The NOIS bundle allows us to clearly identify the prominent

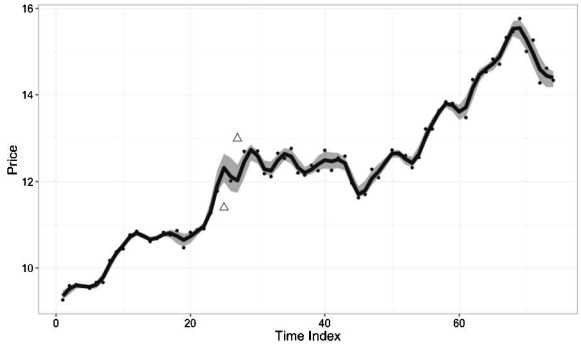


*Figure 4. NOIS bundle for HPQ from 04/12/1999 to 07/26/1999.*

features of a bullish flag pattern: upward acceleration, price movement between downward sloping trendlines and then upward acceleration out of the flag. An interesting trait is that the bundle seems to become thicker when the stock price is within the flag. This is perhaps unsurprising because in these regions, there is a greater amount of movement and consecutive prices are farther apart. NOIS also detects two outliers near the left flagpole that warrant further investigation, as it seems that they cause the bundle width to increase in this region.

In Section 2 we elucidate the various components of our method and describe how it can tackle the non-linearity, heavy-tails, outliers and heteroskedasticity of stock price data. Then, in Section 3 we conduct experiments on synthetic and real-world data to demonstrate the potency of NOIS. Finally, we conclude with a discussion in Section 4 and provide additional computational details in Section 5. The methods described in this paper are implemented in an R package [21] at https://github.com/hoangtt1989/NOIS.

## 2. CHALLENGES AND METHODOLOGY

### 2.1 Non-linearity of stock prices

It is clear from Figures 2 and 4 that a non-linear trend underlies this price series, which renders conventional parametric approaches such as linear regression inappropriate. Therefore, we need a non-parametric method that can capture the non-linear trend in price. Two common non-parametric methods for tackling non-linearity are basis approaches such as splines and kernel smoothing. Representing the underlying function with a basis expansion is often successful at capturing global effects, but it is perhaps not as applicable to the detection of local features in stock chart patterns. On the other hand, because it is a locally weighted average, kernel smoothing can extract these features with more granularity and thus adapts well to the rapid changes in stock prices.

[15] previously applied kernel smoothing to various stock price series to develop a pattern recognition algorithm for

automating technical analysis. Following their work, we use the Nadaraya-Watson kernel smoother. In summary, kernel smoothing or kernel regression reduces observational noise by taking a weighted average of observations in a neighborhood. The observations are weighted by a kernel function and the size of this neighborhood is determined by an accompanying bandwidth $h$. Some popular kernel functions include the Gaussian, Epanechnikov and Tri-cube kernels. Because only neighboring observations are included in the estimation of some function value $f(x)$, kernel smoothing is successful at capturing local behavior around a target point $x$ and it naturally handles the non-linearity of stock prices.

We could consider the following as a prototype model (assuming that no observations are outlying): $y_t = f(x_t) + \epsilon_t$, $t = 1, \ldots, n$ where $\epsilon_t \sim N(0, \sigma^2(x_t))$ and $f(x_t)$ is an unknown function. Here, $y_t$ represents the stock price and $x_t$ is an index of time. Thus, this prototype assumes that a stock price at some time $t$ is the sum of an unknown, possibly non-linear function and a Gaussian noise term. At a target point $x_0$, the Nadaraya-Watson estimating equation for $f(x_0)$ in the above model is:

$$\sum_{t=1}^{n} K_h(x_0 - x_t)[y_t - f(x_0)] = 0 \tag{1}$$

where $K_h(\cdot)$ is a kernel function with bandwidth $h$; for convenience we use the Gaussian kernel in this work, but the method is easily generalizable to other choices of the kernel function.

As seen in Figures 2 and 4, the trend of a stock price can be quite "wiggly" in certain regions. This presents a further challenge because the kernel smoothing estimator is biased in areas of high curvature and at the boundaries [9]. Additionally, if one wishes to forecast or extrapolate the model, the boundary bias can result in spurious predictions.

Some standard techniques for reducing the effect of the bias include undersmoothing with a small bandwidth [2] and estimating the second derivative, both of which are non-trivial in practice [9]. We address the bias correction issue by the method of [14], which uses a different estimating equation than (1):

$$\sum_{t=1}^{n} K_h(x_0 - x_t)(y_t - \hat{f}(x_t) + \hat{f}(x_0) - r(x_0)) = 0 \tag{2}$$

where $\hat{f}(x_0)$ is found by solving (1) for the Nadaraya-Watson function estimate. By solving for $r(x_0)$ in (2), we can obtain a *bias-corrected* kernel smoothing estimate. The idea is that $K_h(x_0 - x_t)(y_t - \hat{f}(x_t) + \hat{f}(x_0) - r(x_0))$ will be close to $K_h(x_0 - x_t)(y_t - f(x_t) + f(x_0) - r(x_0)) = K_h(x_0 - x_t)(\epsilon_t + f(x_0) - r(x_0))$ which is an unbiased estimating equation for $f(x_0)$. Our method for outlier resistant kernel smoothing first uses (1) for robust function estimates, then corrects for the bias by solving for $r(x_0)$ in (2). Further details for bias correction are given in Section 5.2.

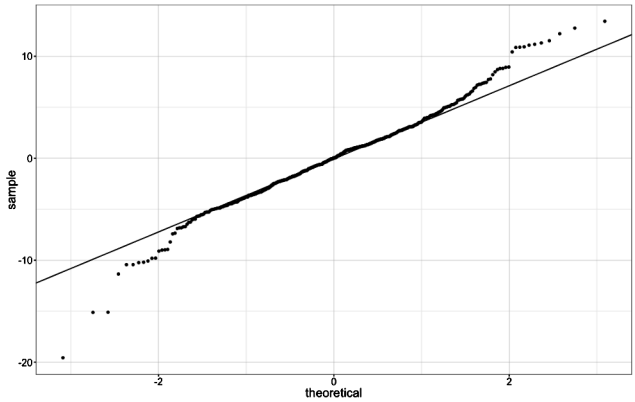


*Figure 5. Normal Q-Q plot of S&P 500 residuals from 08/16/1989 to 08/07/1991.*

Another critical topic in smoothing stock price data is bandwidth selection. If the chosen bandwidth $h$ is too small, the fitted curve will be too jagged and offer little improvement over the raw stock price series for identifying chart patterns. On the other hand, with a bandwidth that is too large for the given data, the curve will be unnecessarily smooth and we may miss features of technical analysis chart patterns. Therefore, we require a **data-dependent** method to determine a suitable value for $h$. The conventional approach to tuning $h$ is leave-one-out cross-validation (LOOCV), but this can produce inappropriate bandwidths for dependent data [3]. In our experience, the method from [28] (referred to as MCV for modified cross-validation), can produce more reasonable results than LOOCV for detecting technical analysis chart patterns. Computational details of cross-validation are left to Section 5.1.

## 2.2 Outlier-resistant kernel smoothing

Another statistical challenge of stock price data is its heavy-tails and outliers. Figure 5 is a Q-Q plot of the residuals from a kernel smoothing fit of 500 days of S&P 500 closing prices from August 16th, 1989 to August 7th, 1991. The plot shows that the noise distribution of the data is heavy-tailed. In other words, large residuals and extreme price movements occur more often than would be expected under a Gaussian distribution. The large deviations from normality near the bottom of the plot are particularly problematic and could be outliers.

Anomalous prices in this data introduce an additional challenge because the functional data techniques from Section 2.1 are sensitive to outliers. Specifically, (1) results from a non-robust squared error loss function

$$\min_{f(\cdot) \in \mathcal{F}} \sum_{t=1}^{n} K_h(x_0 - x_t)[y_t - f(x_0)]^2 \tag{3}$$

where $\mathcal{F}$ is a functional family, e.g., $\mathcal{F} = \{f(x) = \alpha + \beta x : a \in \mathbb{R}, b \in \mathbb{R}\}$. Unless otherwise specified, in this work we

limit $\mathcal{F}$ to the constant term $\alpha$. The squared error loss function penalizes large deviations too heavily and can become dominated by only a small number of outliers. This can be exacerbated in the presence of gross outliers, in which a single severe anomaly can have a large effect on the curve estimate, as seen in Figure 2. To rectify this problem, some previous approaches substitute an $M$-estimate of the regression mean, where the estimate $\hat{f}(x_0)$ solves an estimating equation with an appropriately chosen $\Psi$ function (e.g. [6], [10] and [11]). Other approaches substitute robust loss functions such as the absolute value, leading to a least absolute deviations type of estimator [5]. However, a loss function must be non-convex and non-smooth to be robust to both *gross* and moderate outliers, which introduces difficulties in computation [8]. Our proposed method is resistant to both moderate and gross outliers because it explicitly identifies them by introducing an outlier term and it requires no modification of the squared error loss function.

In the spirit of [25], we propose a new functional form with a mean-shift parameter $\gamma_t$ to achieve our goals of outlier detection and robust estimation:

$$y_t = f(x_t) + \gamma_t + \epsilon_t \quad t = 1, \ldots, n \tag{4}$$

where $\epsilon_t \sim N(0, \sigma^2(x_t))$. When an observation $t$ is an outlier, $\gamma_t \neq 0$. For a "clean" or non-outlying observation $k$, $\gamma_k = 0$. Since outliers should not be the norm, we also impose sparsity on $\boldsymbol{\gamma} = [\gamma_1 \ldots \gamma_t]$. In the context of stock prices, a sparsity assumption is reasonable considering the $2\% - 3\%$ estimate of outliers in high-frequency data.

The sparse mean-shift parameter $\boldsymbol{\gamma}$ in (4) leads to a penalized kernel smoothing objective function

$$\min_{f(x_0),\boldsymbol{\gamma}} \sum_{t=1}^{n} K_h(x_0 - x_t)[y_t - \gamma_t - f(x_0)]^2 + \lambda \sum_{t=1}^{n} P(\gamma_t). \tag{5}$$

where $P(\cdot)$ is a sparsity-promoting penalty function with tuning parameter $\lambda$ and $K_h(\cdot)$ is a kernel with bandwidth $h$. For example, $P(\cdot)$ can be a convex function such as the $L_1$ penalty. An interesting fact is that with the $L_1$ penalty, the above regression problem amounts to using Huber's loss [24], which is not robust enough to multiple gross outliers.

In this work, we instead use the constrained form with an $L_0$ constraint on $\boldsymbol{\gamma}$:

$$\begin{aligned} &\min_{f(x_0),\boldsymbol{\gamma}} \sum_{t=1}^{n} K_{h,t}(x_0)[y_t - \gamma_t - f(x_0)]^2 \\ &\text{s.t.} \sum_{t \in \mathcal{J}(x_0)} \mathbb{1}_{\gamma_t \neq 0} \leq q(x_0) \end{aligned} \tag{6}$$

where $K_{h,t}(x_0) \equiv K_h(x_0 - x_t)$, $\mathcal{J}(x_0) = \{\, t : K_h(x_0 - x_t) \neq 0 \,\}$ and $J(x_0) = |\mathcal{J}(x_0)|$. It is easy to argue that when optimizing (6) the constraint is equivalent to $\|\boldsymbol{\gamma}\|_0 \leq q(x_0)$. The discreteness, non-differentiability and non-convexity of the $L_0$ constraint introduce difficulties in jointly optimizing over $f(x_0)$ and $\boldsymbol{\gamma}$. On the other hand, suppose that estimation of $f(x_0)$ by kernel smoothing involves $d$ free parameters. By including $\boldsymbol{\gamma}$ in the objective function, this actually becomes a higher dimensional problem with $n + d$ degrees of freedom, but the $L_0$ constraint effectively reduces the number of estimable parameters to $q(x_0) + d$. Another advantage of the constrained form is that we set $q(x_0)$ which is simply the number of detected outliers, rather than tuning $\lambda$ for the penalized form, which can be difficult in practice. For example, if $J(x_0) = 100$ at some target point $x_0$, we could set $q(x_0) = 5$ to identify 5% of the points with non-zero kernel weights as outliers.

**Algorithm 1** Kernel smoothing with joint outlier detection.

**Input:** $\vartheta \in (0,1)$, $\boldsymbol{x}$, $\boldsymbol{y}$, $h$ (by cross-validation)
  **for** $k \leq n$ **do**
    $j \leftarrow 0$, $\boldsymbol{\gamma}^{(j)} \leftarrow \mathbf{0}$
    **while** not converged **do**
      $\boldsymbol{y}^{\text{adj}} \leftarrow \boldsymbol{y} - \boldsymbol{\gamma}^{(j)}$
      $\hat{f}(x_k) \leftarrow \sum_{t=1}^{n} K_{h,t}(x_k) y_t^{\text{adj}} / \sum_{t=1}^{n} K_{h,t}(x_k)$
      $\mathcal{J}(x_k) \leftarrow \{\, t : K_{h,t}(x_k) \neq 0 \,\}$, $J(x_k) \leftarrow |\mathcal{J}(x_k)|$, $q(x_k) \leftarrow \text{floor}(\vartheta \times J(x_k))$
      $\boldsymbol{w}(x_k) \leftarrow K_{h,t \in \mathcal{J}^k}(x_k) / \sum_{t=1}^{n} K_{h,t}(x_k)$
      $\boldsymbol{r}^{(j)} \leftarrow \text{diag}\{\sqrt{\boldsymbol{w}(x_k)}\}(\boldsymbol{y} - \hat{f}(x_k))$
      $\boldsymbol{\gamma}^{(j+1)} \leftarrow (\text{diag}\{\sqrt{\boldsymbol{w}(x_k)}\})^{-1} \Theta^{\#}(\boldsymbol{r}^{(j)}; q(x_k))$
      $p^{(j+1)} \leftarrow \|\boldsymbol{\gamma}^{(j+1)} - \boldsymbol{\gamma}^{(j)}\|_\infty$
      converged $\leftarrow p^{(j+1)} < \epsilon$
      $j \leftarrow j + 1$
    **end while**
    $\boldsymbol{\Omega}[:, k] \leftarrow \boldsymbol{\gamma}^{(j)}$, $\boldsymbol{Y}^{\text{adj}}[:, k] \leftarrow \boldsymbol{y}^{\text{adj}}$
  **end for**
**Output:** $\hat{\boldsymbol{f}}$, $\hat{\boldsymbol{\Omega}}$, $\boldsymbol{Y}^{\text{adj}}$

Due to the nature of the $L_0$ constraint, Algorithm 1 will classify the $q(x_0)$ largest estimates of $\gamma_0$ as outliers, regardless of their magnitude. It is arguable that information about the magnitude of $\gamma_0$ should be taken into account when detecting outliers, but by using the $L_0$ constraint our method takes a conservative perspective. As long as $q(x)$ is reasonably large, the method removes "dangerous" points lying far away from the majority to ensure robustness and guard against outliers, which is one of the goals of this work.

For $\boldsymbol{\gamma}$ given, the solution to (6) is the Nadayara-Watson estimator with a shifted $y_t$: $\hat{f}(x_0) = \sum_{t=1}^{n} K_{h,t}(x_0)(y_t - \gamma_t) / \sum_{t=1}^{n} K_{h,t}(x_0)$. Of course, this estimator can only be computed if $\boldsymbol{\gamma}$ is known. Therefore, we devise a scheme that alternates between estimating $f(x)$ through kernel smoothing then detecting outliers by thresholding the resulting weighted residuals. We first introduce some notation before outlining the procedure in Algorithm 1. The parameter $\vartheta \in (0,1)$ controls the percentage of points that will be detected as outliers around some target point $x_0$. The operator

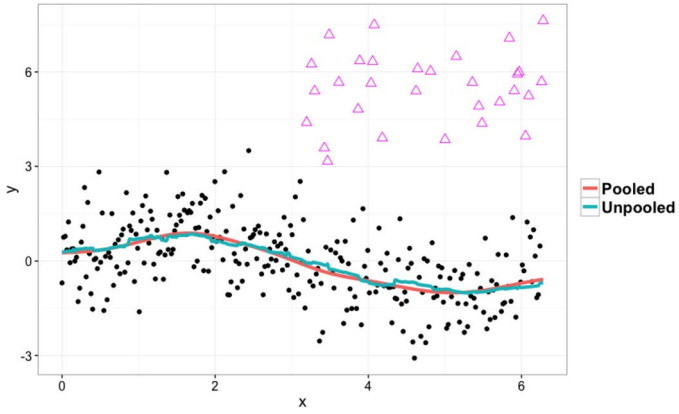


*Figure 6. Pooled and unpooled NOIS estimates. Triangles represent outliers detected by NOIS, which are also the true outliers.*

$\Theta^{\#}$ is quantile thresholding [23]:

$$\Theta^{\#}(\boldsymbol{x}; q) = \begin{cases} x_{(j)} & \text{if } j \le q \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

Here, $x_{(1)}, \ldots, x_{(n)}$ are the order statistics of $\boldsymbol{x} = [x_1 \ldots x_n]$: $|x_{(1)}| \ge \cdots \ge |x_{(n)}|$. We use floor$(x)$ to denote the largest integer less than or equal to $x$ and diag$\{\boldsymbol{x}\}$ to denote a diagonal matrix with the vector $\boldsymbol{x}$ as its diagonal entries. For a matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, the notation $\boldsymbol{X}[:, k]$ is an abbreviation for $\boldsymbol{X}[1\!:\!m, k]$ (the $k$th column of $\boldsymbol{X}$).

After running Algorithm 1 for a given dataset, we return two matrices $\hat{\boldsymbol{\Omega}}$ and $\boldsymbol{Y}^{\text{adj}}$. Each column in $\hat{\boldsymbol{\Omega}}$ is a vector $\hat{\gamma}$ estimated at a target point $x_0$ and each column in $\boldsymbol{Y}^{\text{adj}}$ is a vector of *clean* prices associated with a target point $x_0$. As shown in Figure 6, the resulting function estimate is "bumpy" since a different set of prices is used to compute each $\hat{f}(x_t)$. In addition, it is difficult to interpret the raw outlier detection results because Algorithm 1 returns $n$ different $\hat{\gamma}$'s. By accumulating the information in $\hat{\boldsymbol{\Omega}}$, we can construct a single "pooled" vector denoted by $\hat{\gamma}^{\text{P}}$ that contains the estimated values of the identified outliers across the whole curve. Then, we generate a single clean data set for which outlying prices have been removed and non-outlying prices are undisturbed, as well as a pooled function estimate. The entire pooled procedure is described below:

- Form the $n \times n$ matrix: $\hat{\boldsymbol{\Omega}} = [\hat{\gamma}_1 \ \ldots \ \hat{\gamma}_n]$ where a row $i$ contains each $\hat{\gamma}$ at time $i$.
- Let $\hat{\boldsymbol{\omega}}_i$ be the $i$th row in $\hat{\boldsymbol{\Omega}}$. Apply the function $m(\boldsymbol{x}) = \text{sign}(\boldsymbol{x}) \odot \max(|\boldsymbol{x}|)$ to each $\hat{\boldsymbol{\omega}}_i$, where $\odot$ is element-wise multiplication and sign$(\boldsymbol{x})$ and $|\boldsymbol{x}|$ are applied element-wise. Combine these values into an $n \times 1$ vector $\boldsymbol{\nu} = [m(\hat{\boldsymbol{\omega}}_1) \ \ldots \ m(\hat{\boldsymbol{\omega}}_n)]^T$. Then, compute the pooled outlier estimate $\hat{\underline{\gamma}} = \Theta^{\#}(\boldsymbol{\nu}; \text{floor}(\varrho \times n))$ where $\varrho$ is the percentage of observations that will be detected as outliers.
- The pooled clean data set is $\{(x_t, y_t^{\text{adj}} = y_t - \hat{\underline{\gamma}}_t), 1 \le t \le n\}$. Perform cross-validation on the pooled clean data set to find $h$. The pooled function estimates $\hat{\underline{f}}(x_t)$ are found by computing the Nadaraya-Watson estimate for each $(x_t, y_t^{\text{adj}})$. The **bias corrected** function estimate is then:

$$\hat{\underline{f}}_b(x_0) = \frac{\sum_{t=1}^{n} K_{h,t}(x_0)(y_t^{\text{adj}} - \hat{\underline{f}}(x_t) + \hat{\underline{f}}(x_0))}{\sum_{t=1}^{n} K_{h,t}(x_0)}. \tag{8}$$

Our method consists of a pipeline from Algorithm 1 which solves $n$ objective functions, to the pooling technique which accumulates the unpooled outlier and function estimates into single vectors $\hat{\underline{\gamma}}$ and $\hat{\underline{\boldsymbol{f}}}_b = \left[\hat{\underline{f}}_b(x_1) \ldots \hat{\underline{f}}_b(x_n)\right]$. We call this entire procedure NOIS for **N**on-Parametric **O**utlier **I**dentification and **S**moothing.

Figure 6 compares the unpooled and pooled estimated curves for simulated data with outliers. We generated 300 points from $y_t = \sin(x_t) + e_t$ where $x_t$ is evenly spaced on the interval $[0, 2\pi]$ and $e_t \sim N(0, 1)$. Outliers were introduced by shifting 30 of the $y_t$ points corresponding to $x_t \in [\pi, 2\pi]$ up by a $U(10, 12)$ random number; estimated outliers are marked by a magenta triangle. In comparison to the unpooled function estimate, the pooled function estimate from NOIS is smoother.

A reasonable question is how one should set the outlier detection parameters $\vartheta$ and $\varrho$. For the previous example, $\vartheta = 0.075$ and thus $q(x) = \text{floor}(0.075 \times J(x))$. In general, if severe outliers are believed to be a problem, a more conservative or larger choice of $\vartheta$ is better so that more potential outliers are available to the pooled component of NOIS. As long as it is reasonably large, empirical studies suggest that $\vartheta$ does not seem to have a significant effect on the detection performance. Due to the ad-hoc nature of the pooling procedure, $\varrho$ is generally set by prior experience with the type of data. For example, if we are working with high-frequency data and want to detect 5% of the points as outliers, we can set $\varrho = 0.05$.

Another strategy for tuning $\varrho$ would be to vary $\underline{q} := \text{floor}(\varrho \times n)$ on a grid from, say 1 to $n/2$ then pick the value that minimizes an information criterion such as BIC$^*(\underline{q})$ [25]. There is theoretical support (details not reported in this paper) for applying BIC$^*$ to each unpooled $\hat{\gamma}$; however, because we use the pooling procedure to produce the final outlier estimate $\hat{\underline{\gamma}}$, further theoretical work is needed. Still, some preliminary empirical studies suggest that BIC$^*$ performs well in selecting the correct $\underline{q}$. Our recommendation is to use BIC$^*$ as a starting point when there is little practical guidance for selecting $\underline{q}$.

## 2.3 NOIS confidence bands

Even after removing outliers by NOIS, the amount of noise in a given stock price series can still be large, which introduces additional uncertainty in a point estimate. Additionally, previous empirical work has suggested that finan-

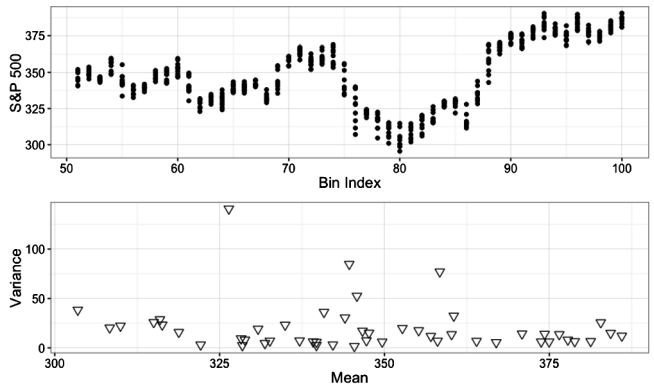


*Figure 7. Binned S&P 500 closing prices and mean-variance plot.*

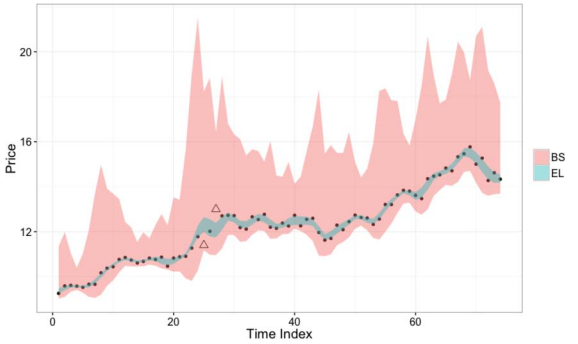


*Figure 8. Comparison of $\alpha = 0.01$ BS and EL confidence bands for HPQ.*

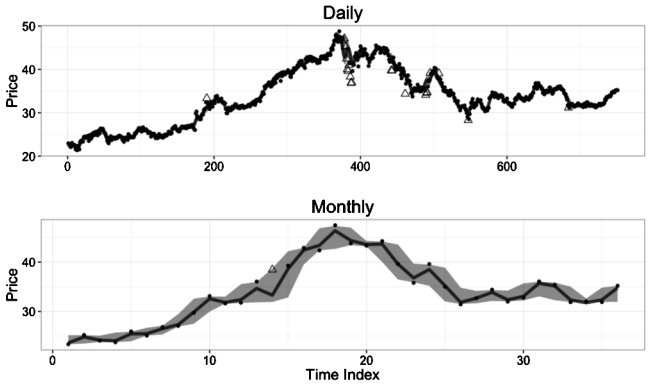


*Figure 9. Daily and monthly stock prices for LMT with empirical likelihood NOIS bundles.*

cial data exhibits heteroskedasticity (e.g. [13] and [22]). Figure 7 illustrates the issue on the S&P 500 data from Figure 5, with the data binned into groups of 10 days to emphasize the distributional characteristics. It is clear that this data is heteroskedastic and the relationship between the mean and the variance would be difficult to model using conventional parametric procedures such as the Box-Cox method.

We use pointwise confidence bands to account for stock price data's heteroskedasticity and the stochastic nature of NOIS curve estimates. By examining their widths, we can use these bands to identify pertinent features in the smoothed curve. We will refer to a NOIS function estimate along with its confidence bands as a **NOIS bundle**.

The standard approach to constructing confidence intervals would be to use a normal approximation to the distribution of the bias corrected function estimate $\hat{\underline{f}}_b(x_t)$. However, this is based on asymptotics and may not be a satisfactory approximation for finite samples, especially if the data exhibits heavy-tails and skewness. Additionally, the normal approximation intervals are symmetric, which means they may not respect the shape of the data. Therefore, we require a non-parametric method for confidence interval construction that can overcome these challenges.

We use empirical likelihood (EL) to construct pointwise confidence bands around the $\hat{\underline{f}}_b(x_t)$'s obtained by NOIS; computational details of EL are left to Section 5.2. Suppose that we wish to test the following hypothesis at some target point $x_0$:

$$H_0 \ : f(x_0) = \theta_0 \quad \text{vs} \quad H_1 \ : f(x_0) \neq \theta_0.$$

EL forms a non-parametric likelihood ratio test for the above hypothesis and does not require stringent distributional assumptions about the point estimate. Its data-adaptive nature means that an EL confidence interval around a kernel smoothing point estimate will not necessarily be symmetric and will automatically reflect emphasis in the observed data [7]. Another advantage of EL is that studentization is carried out internally via the optimization procedure [2] so it requires no explicit estimate of the standard deviation and is particularly applicable to heteroskedastic data.

The residual bootstrap is another popular non-parametric method which resamples the residuals from $\hat{\underline{f}}_b(x_t)$ to construct pointwise confidence bands around the function estimates. This technique can also handle heteroskedasticity and non-Gaussinity, but it requires a standard deviation estimate, which can sometimes result in unstable confidence bands for stock price data. Figure 8 illustrates the issue on daily prices of Hewlett-Packard (HPQ) during the period of April 12th, 1999 to July 26th, 1999. The two outliers identified by NOIS are marked by a triangle and the bootstrap (BS) and empirical likelihood (EL) confidence bands (without bias correction) are the shaded areas. Due to the instability of bootstrap confidence bands for stock prices, we will use EL. However, in a situation where computational speed is of the utmost importance, a practitioner may prefer to use the bootstrap because in our experience it requires less computation time for NOIS bundles than EL. Further details of the bootstrap are in Section 5.3.

Figure 9 plots roughly three years of Lockheed Martin's (LMT) daily and monthly stock prices. Estimated outliers are marked by a triangle, the bias corrected NOIS function estimate $\hat{\underline{f}}_b$ is a solid line and the pointwise confidence

*Table 1. Outlier detection simulation results as $O$ (the true number of outliers) varies*

| | $O = 10$ | $O = 20$ | $O = 35$ |
|---|---|---|---|
| **MSE** | 0.13 | 0.14 | 1.2 |
| **M** (%) | 0 | 0.07 | 19 |
| **S** (%) | 11 | 5.6 | 1.1 |
| **JD** (%) | 100 | 99 | 0 |

bands are shaded. Because the daily data is so dense, the confidence bands are nearly indistinguishable from the function estimates and overall offer little additional information. In comparison, the monthly NOIS bundle is computed on a sparser data set which has more variation from point to point, resulting in confidence bands with varying levels of thickness. Thus, the pointwise confidence band component of a NOIS bundle is most beneficial in the data-insufficient case where the resolution is relatively low.

## 3. EXPERIMENTS

### 3.1 Synthetic data

We conducted two experiments to test the outlier detection capabilities of NOIS and the coverage probabilities of NOIS bundles. We used the following simulation setup for each experiment:

- 200 points were generated according to $y_t = \sin(x_t) + e_t$ with the $x_t$'s evenly spaced on $[0, 2\pi]$ and $e_t \sim N(0, 1)$.
- Outliers were introduced by randomly sampling $O \in \{10, 20, 35\}$ of the $y_t$'s where $x_t \in [\pi, 2\pi]$, then shifting them up by a $U(10, 12)$ random number.
- For the outlier detection experiment we fixed the outlier detection parameters at $\vartheta = 0.15$ and $\varrho = 0.15$. In practice, a more conservative or larger choice of $\varrho$ can be made. In the coverage probability experiment $\vartheta = 0.15$ and $\varrho = O/200$.
- The data was simulated 300 times.

We report four metrics for outlier detection:

- **MSE**: the mean squared error computed as MSE $= n^{-1}\sum(\sin(x_t) - \underline{\hat{f}}_b(x_t))^2$ averaged across all runs.
- **M**: the mean masking probability (percentage of undetected true outliers).
- **S**: the mean swamping probability (percentage of good points labeled as outliers).
- **JD**: the joint outlier detection rate (percentage of simulations with 0 masking).

Ideally, both M and S will be low and JD $\approx 100\%$, but masking is more problematic than swamping because undetected outliers can cause gross distortions in the estimated curve. The results for outlier detection are presented in Table 1.

The outlier detection results demonstrate that NOIS is successful at minimizing M when $\varrho > O$. For both $O = 10$ and $O = 20$, the ideal result is M = 0. When $O = 30$, the lowest possible M is $5/35 \approx 14\%$ so the masking probability of 18% is not optimal but still acceptable. The low S rates show that even when the tuning parameters are misspecified, the loss in efficiency is not too problematic.

*Table 2. Confidence band simulation results at $1 - \alpha = 0.95$ nominal coverage level. Numbers presented are the average coverage probabilities and average interval lengths (in parentheses) across all simulations*

| | With bias correction | | | |
|---|---|---|---|---|
| | NOIS+EL | NOIS+BS | KS+EL | KS+BS |
| $O = 10$ | 0.85 (0.6) | 0.9 (0.75) | 0.42 (1) | 0.55 (1.3) |
| $O = 20$ | 0.82 (0.6) | 0.88 (0.79) | 0.37 (1.3) | 0.49 (1.6) |
| $O = 35$ | 0.75 (0.83) | 0.84 (1.2) | 0.36 (1.9) | 0.48 (2.5) |
| | Without bias correction | | | |
| | NOIS+EL | NOIS+BS | KS+EL | KS+BS |
| $O = 10$ | 0.85 (0.61) | 0.87 (0.64) | 0.4 (1) | 0.5 (1.2) |
| $O = 20$ | 0.82 (0.62) | 0.85 (0.68) | 0.35 (1.3) | 0.45 (1.4) |
| $O = 35$ | 0.76 (0.86) | 0.81 (1.1) | 0.35 (1.9) | 0.44 (2.2) |

We compared four pointwise confidence bands for the coverage probability experiment: NOIS with EL (NOIS+EL) and the bootstrap (NOIS+BS) as well as non-robust kernel smoothing with EL (KS+EL) and the bootstrap (KS+BS). The average coverage probabilities along with the average interval lengths (in parentheses) across all simulations are reported in Table 2. For NOIS bundles, bias corrected confidence bands are constructed by using EL or bootstrap on $\hat{\boldsymbol{f}}_b$. Non-bias corrected NOIS bundles are formed by using EL or bootstrap on $\hat{\underline{\boldsymbol{f}}}$. Confidence bands with and without bias correction are constructed around the non-robust kernel smoothing estimates in a similar manner. The conclusions are summarized below:

- Clearly, the non-robust kernel smoothing bundles are inferior in terms of coverage probability and interval widths.
- The coverage probabilities of the bias corrected NOIS bundles are often superior to their non-bias corrected counterparts, but the improvement is not very large. The bias correction seems to offer no discernible benefit for the NOIS+EL bundles and even seems to perform worse for $O = 35$.
- The bootstrapped confidence bands are closer to the nominal coverage level than empirical likelihood confidence bands, but this is accompanied by larger interval widths, especially in the bias corrected cases.

Overall, NOIS bundles for these simulated data do not closely match the nominal coverage level. Figure 10 plots the coverage probabilities at each $x_t$ along with one of the simulated datasets. Clearly, most of the undercoverage issues occur in regions with outliers and at the boundaries. It is possible that the pooling component of NOIS introduces

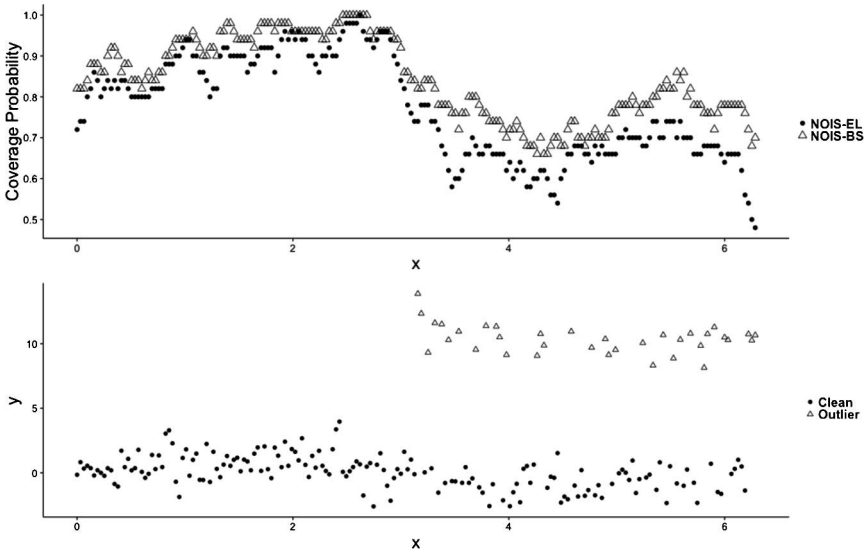


*Figure 10. Coverage probabilities for bias corrected NOIS-EL and NOIS-BS ($O = 35$).*

an additional source of bias in outlier detection and the resulting function estimate. We still advocate the use of NOIS bundles for stock price data because the undercoverage is not so sizable that they become uninformative.

## 3.2 Real-world high-frequency data

In this section we demonstrate the efficacy of NOIS on high-frequency (or tick) stock data retrieved from the New York Stock Exchange (NYSE) Trades and Quotes (TAQ) database. When fitting a curve to high-frequency data, outliers can create difficulties for chart pattern detection, as seen in the distorted kernel smoothing estimate from Figure 2. Thus, a robust function estimate should be free from the effects of these anomalous prices. Outliers in tick data often occur due to errors in the price discovery and trading mechanism processes, such as price differences between exchanges trading the same stock. There is no steadfast or universal definition for "unclean" tick data, but in general "you know it when you see it" [4]. In the following examples, we will consider three types of outliers: bad ticks which are so far from the prevailing price level that it is highly likely they are erroneous, borderline cases that would be considered outliers by some but not all traders and misidentified ticks which are identified as outliers by NOIS but are unlikely to be of practical concern.

We first consider roughly eleven minutes of McDonald's high-frequency data. Here, we set $\vartheta = 0.05$ and $\varrho = 0.03$. Figure 11 plots bias corrected non-robust and NOIS function estimates, with detected outliers marked by a magenta triangle. We classify the three most extreme outliers as bad ticks because they all occur at the same price level which is far away from the rest of the data. Interestingly, there are a number of detected outliers that are likely misidentified ticks. Recall that when running the NOIS algorithm,

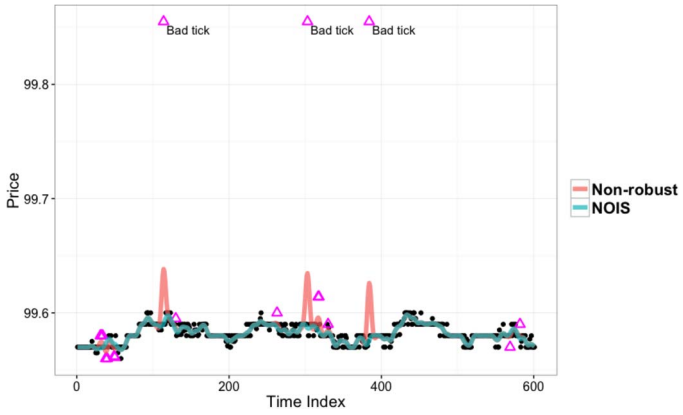


*Figure 11. Function estimates for MCD on Jan. 3rd, 2012 from 3:00:37 to 3:08:55 pm.*

outliers are identified within the neighborhood around each target point $x_0$ and then pooled to yield the final detection result. It is possible that $\vartheta$ and $\varrho$ are too large for this data, which causes NOIS to remove too many prices. However, even after these more moderate outliers are removed, the robust fitted curve does not seem to be altered too much.

Figure 12 is taken from three minutes of the MCD high-frequency data; once again, the magenta triangles are identified as outliers by NOIS. There is a price that is about \$0.20 away from the curve and is likely a bad tick. The plot highlights two detected outliers that could be borderline cases. Because NOIS explicitly identifies these points, they are marked for further review by a trader.

In the above examples, we found that NOIS was able to detect gross outliers which were likely bad ticks occurring due to errors in trading mechanisms. Although some

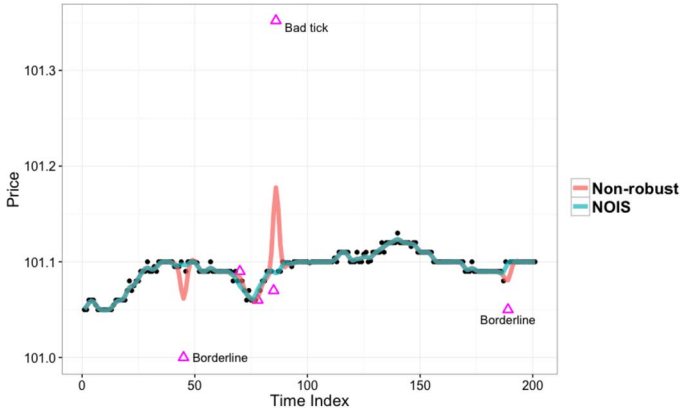


*Figure 12. Function estimates for MCD on Jan. 3rd, 2012 from 9:40:01 to 9:41:17 am.*

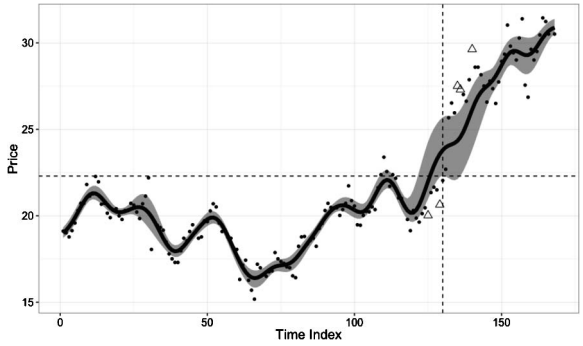


*Figure 13. NOIS bundle for JBL cup and handle pattern.*

prices were possibly misidentified as outliers for conservatively large values of $\vartheta$ and $\varrho$, it is better to remove a small number of non-outlying points than *underscrub* the data and miss the bad ticks which can distort a fitted curve. In practice, one can adjust $\vartheta$ and $\varrho$ to achieve a desired level of outlier detection and removal. As long as these parameters are not too prohibitive, the gross outliers or bad ticks will still be detected. From a statistical viewpoint, the misidentified ticks are not too alarming as the loss in efficiency is usually small when $\varrho \ll 1$.

## 3.3 NOIS bundles for technical analysis

In this section, we demonstrate applications of NOIS bundles to daily stock price data for analyzing technical analysis chart patterns previously identified by practitioners. The outlier detection parameters are $\vartheta = 0.15$ and $\varrho = 0.03$, the confidence level is $\alpha = 0.01$ and empirical likelihood pointwise confidence bands are formed around the bias corrected function estimate. In each plot outliers detected by NOIS are marked by a triangle.

### 3.3.1 Cup and handle

As previously shown in Figure 1, the cup and handle is a continuation pattern with three distinguishing features: a rounded "cup" shape and a "handle" that leads to a breakout of upward price action. Sometimes after the breakout, the price will return towards the handle in a throwback before continuing on its upward movement. Traders will often look for a throwback as further confirmation of a buy signal.

[26] identified a cup and handle pattern in daily stock price data of Jabil Circuit, Inc. (JBL) between May 3rd, 1999 to December 12, 1999. Figure 13 is a NOIS bundle applied to JBL's daily stock prices, with the curves and outliers marked as in the previous section. We use dashed lines to highlight two features: the horizontal line shows the top of the "cup" and the intersection of the lines show the end of the "handle" into the breakout. The breakout exhibits upward accelerating price movement with a gap between prices. This causes the confidence bands to become quite thick, which could perhaps be used as a confirmatory signal for the breakout. Another interesting attribute is that the confidence bands seem to exhibit a throwback towards the dashed horizontal line. If a trader were only looking at the raw stock prices or the fitted curve, it would be easy to miss this feature.

### 3.3.2 Rectangle

A rectangle is a continuation pattern in which prices alternate between touching resistance and support lines that provide boundaries for movement. After exiting the rectangle (the breakout), prices can sometimes return to the rectangle through a throwback before continuing on the previous trend.

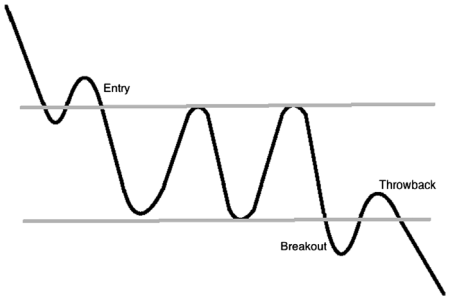


*Figure 14. Rectangle chart pattern.*

Figure 15 plots a NOIS bundle applied to daily stock prices of Lockheed Martin (LMT) between May 1st, 1999 and October 29, 1999, during which a rectangle pattern was identified by [26]. The NOIS bundle "bounces" between the two horizontal dashed lines, forming the rectangle pattern. The price takes a large jump into the rectangle pattern, which causes the bundle to widen in this region. After the throwback, the price accelerates downward and the bundle once again becomes thicker. In comparison to the flag in Figure 4, the confidence bands within the pattern are relatively thin due to "tighter" price action. A trader could

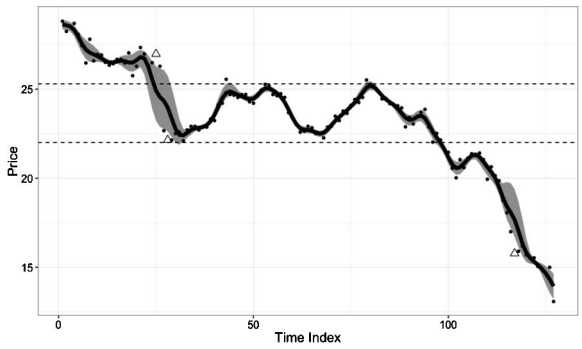


*Figure 15. NOIS bundle for LMT rectangle pattern.*

perhaps interpret this as a stronger signal that a rectangle is forming.

## 4. DISCUSSION

The analysis of financial data can be challenging due to its non-linearity, heteroskedasticity, and heavy tails. In combination with the prevalence of gross outliers in high-frequency data, these issues can be problematic for the technical analysis practitioner who wishes to identify geometric patterns in stock charts. We proposed a method that can overcome these challenges by robustly estimating the function underlying a stock's prices while automatically detecting outliers (NOIS). Additionally, we demonstrated that pointwise confidence bands around these robust estimates form a curve bundle that can assist the technical analyst in identifying notable features when the data resolution is relatively low.

The examples in this work suggested that gross outliers are not as prevalent in daily inter-day data as they are in high-frequency intra-day data. Indeed, by the time a daily closing price is reported, ticks that appear as irregularities or errors to the intra-day trader are unlikely to be visible on the inter-day time scale [4]. Therefore, we advocate more aggressive or smaller values for the outlier detection parameters of NOIS when analyzing inter-day data. On the other hand, in empirical studies the prevalence of outliers in tick data provided a compelling reason to use NOIS, but the high resolution of this data makes the confidence bands less revealing. Despite these characteristics, the proposed approach can adapt to these different situations without much human intervention.

High-frequency prices introduce additional challenges to practical data analysis such as asynchronous ticks, the discrete sampling frequency and microstructure noise [27]. While our method does not directly address these issues, as shown in our real data examples NOIS still has practical utility in detecting high-frequency outliers. In a future work we intend to investigate how NOIS can be modified to deal with these challenges.

We applied our method to multiple examples of technical analysis chart patterns that had previously been identified by practitioners. The NOIS bundles were able to highlight notable features in the patterns and the smoothed curve assisted in identification. A further direction of research would combine the robust estimation and outlier detection of NOIS with techniques from machine learning and computer vision, allowing for automated identification of technical analysis chart patterns.

In simulation studies the outlier detection capabilities of NOIS were sound but the pointwise empirical likelihood confidence bands sometimes showed undercoverage. We postulated that the pooled procedure of NOIS introduced some bias into the robust function estimate. In a future work we will investigate simultaneous confidence bands, which typically have greater lengths and could therefore alleviate the issue. Notwithstanding the undercoverage on these synthetic data, our real-world examples showed that NOIS bundles can still be useful for technical analysis practitioners.

In a future work we hope to explore volatility function estimation for a single stock price using NOIS. [29] utilized a thresholded kernel smoother on squared logarithmic returns to detect volatility anomalies. NOIS could be similarly applied to this task by using its robust smoothing component on squared logarithmic returns. Of course, due to its robustness and non-parametric nature, NOIS can handle other prospective transformations.

In principle, NOIS can be extended to the multivariate setting by taking into account the correlations among multiple time series. Specifically, suppose the goal is to robustly estimate the underlying curves for $m$ different stock prices measured at the same $n$ time indices. Let $\boldsymbol{W} := \boldsymbol{\Sigma}^{-1}$ (the $m \times m$ inverse covariance matrix of the stocks); $\boldsymbol{y}_t \in \mathbb{R}^m$ be a vector of stock prices at time $t$; $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1 \dots \boldsymbol{\gamma}_n]$ be an $m \times n$ matrix in which the $i$th column $\boldsymbol{\gamma}_i \in \mathbb{R}^m$ is a vector of mean-shift parameters at time $i$; and $\boldsymbol{f}(x_0) \in \mathbb{R}^m$ be a vector of unknown functions evaluated at a target point $x_0$. We intend to investigate the following optimization criterion in a future work: $\min_{\boldsymbol{f}(x_0),\boldsymbol{\Gamma},\boldsymbol{W}} \sum_{t=1}^n K_{h,t}(x_0)\{(\boldsymbol{y}_t - \boldsymbol{\gamma}_t - \boldsymbol{f}(x_0))^T \boldsymbol{W}(\boldsymbol{y}_t - \boldsymbol{\gamma}_t - \boldsymbol{f}(x_0)) - \log(\det(\boldsymbol{W})) + P(\boldsymbol{W})\}$ s.t. $\|\boldsymbol{\Gamma}[k,:]\|_0 \leq q_k(x_0)\, \forall k \in 1,\dots,m$, where $P(\cdot)$ is, say, a sparsity-promoting penalty.

## 5. FURTHER COMPUTATIONAL DETAILS

### 5.1 Kernel smoothing bandwidth selection

As stated before, the bandwidth $h$ is typically chosen by leave-one-out cross-validation (LOOCV). That is, we select $h$ to minimize the cross validation function:

$$CV(h) = \frac{1}{n}\sum_{t=1}^{n}\left[y_t - \hat{f}_{h,t}(x_t)\right]^2 w(x_t) \tag{9}$$

where

$$\hat{f}_{h,t}(x_t) := \frac{1}{n}\frac{\sum_{j\neq t}^{n} K_h(x_t - x_j)y_j}{\sum_{j\neq t}^{n} K_h(x_t - x_j)} \tag{10}$$

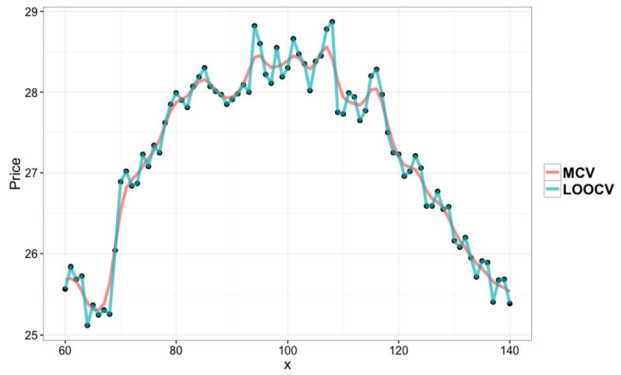


*Figure 16. Comparison of cross-validation methods for daily prices of MSFT.*

and $w(\cdot)$ is a weight function. For this paper, $w(\cdot)$ is an indicator function corresponding to the inner estimated 95% sample space of the $x_t$'s. LOOCV requires $n(n-1)$ kernel evaluations but these computations can be easily parallelized if multiple cores are available. For independent data such as those generated for Figure 6, LOOCV performs well. However, because the purpose of this work is to extract patterns from stock prices and analyze their shape characteristics, we require a cross-validation method that can produce bandwidths that are reasonably smooth for dependent data. For example, Figure 16 plots a kernel smoothing estimate on eighty days of Microsoft's stock price. The bandwidth found by LOOCV was 0.125, which results in a curve too sensitive to the noise. This leads us to consider a procedure suitable for dependent data from [28], which will be referred to as modified cross-validation (MCV) and is summarized below:

- Split the data into two pieces $\{(x_t, y_t), 1 \le t \le s\}$ and $\{(x_t, y_t), s < t \le n\}$.
- Choose the bandwidth $h$ such that $\hat{f}_{h,s}(x)$ gives the best prediction for $y_t$ for $s < t \le n$ in the sense that $h = \hat{h}_s$ minimizes

$$\mathrm{ECV}_s(h) = \frac{1}{n-s} \sum_{t=s+1}^{n} \left[ y_t - \hat{f}_{h,t}(x_t) \right]^2 w(x_t)$$

  over $H_s = \left[ a s^{-1/5-\epsilon_0}, b s^{-1/5+\epsilon_0} \right]$ where $0 < a < b < \infty$ and $\epsilon_0 \in (0, 1/150)$ are constants. Thus, $\hat{f}_{h,t}(x)$ is the estimator constructed using data in $\{(x_t, y_t), 1 \le t \le s\}$ and the cross validation function is then evaluated on the remaining data with $h$ defined in a reasonably small interval.
- The bandwidth for the whole sample is $\hat{h}_n = \hat{h}_s (s/n)^{1/5}$.

The adjustment of $h_s$ in the last step is motivated by the theoretical optimal bandwidth of $h \propto n^{-1/5}$ that minimizes the mean squared error of the kernel smoothing estimator. Under certain conditions MCV is asymptotically optimal. This method results in more reasonable bandwidths for stock prices and is computationally faster than LOOCV. For example, with $s = \mathrm{floor}(0.66n)$ the bandwidth of $\hat{h}_n = 1.26$ chosen by MCV in Figure 16 seems more reasonable than LOOCV. For the real data applications in this paper, a choice of $s$ between $\mathrm{floor}(0.60n)$ and $\mathrm{floor}(0.70n)$ tends to work well.

## 5.2 Empirical likelihood and bias correction

In this section we describe the steps to compute empirical likelihood (EL) for NOIS function estimates. The empirical likelihood for a candidate value $\mu(x_0)$ for $f(x_0)$ is

(11)
$$L_n(\mu(x_0)) = \max\left\{ \prod_{t=1}^{n} p_t \,\middle|\, \sum_{t=1}^{n} p_t = 1,\ p_t \ge 0,\ \sum_{t=1}^{n} p_t K_{h,t}(x_0)(y_t^{\mathrm{adj}} - \mu(x_0)) = 0 \right\}.$$

This is a multinomial likelihood with probability weights placed on each observation $y_t^{\mathrm{adj}}$. The first two constraints in (11) specify that the $p_t$'s are probability weights while the third is known as the structural constraint, which is specific to the parameter of interest. The above expression is maximized at $p_t = n^{-1}$, which corresponds to $\mu(x_0) = \hat{f}(x_0)$ in the structural constraint. Therefore, the **log empirical likelihood ratio** is $\mathrm{LR}_n(\mu(x_0)) = \log(L_n(\mu(x_0))/L_n(\hat{f}(x_0)))$ or

(12)
$$\mathrm{LR}_n(\mu(x_0)) = \max\left\{ \sum_{t=1}^{n} \log(np_t) \,\middle|\, \sum_{t=1}^{n} p_t = 1,\ p_t \ge 0,\ \sum_{t=1}^{n} p_t K_{h,t}(x_0)(y_t^{\mathrm{adj}} - \mu(x_0)) = 0 \right\}.$$

Since the objective function in (12) is strictly concave, a unique global maximum exists. Using Lagrange multipliers, we can find an expression for the optimal weights

(13)
$$p_t = \frac{1}{n} \frac{1}{1 + \lambda K_{h,t}(x_0)(y_t^{\mathrm{adj}} - \mu(x_0))},$$

and the log EL ratio evaluated at $\mu(x_0)$ becomes

(14)
$$\mathrm{LR}_n(\mu(x_0)) = -\sum_{t=1}^{n} \log\left(1 + \lambda K_{h,t}(x_0)(y_t^{\mathrm{adj}} - \mu(x_0))\right)$$

where $\lambda$ is a Lagrange multiplier satisfying

(15)
$$G(\lambda) = \sum_{t=1}^{n} \frac{(y_t^{\mathrm{adj}} - \mu(x_0))K_{h,t}(x_0)}{1 + \lambda(y_t^{\mathrm{adj}} - \mu(x_0))K_{h,t}(x_0)} = 0.$$

A safe-guarded root finding algorithm such as Brent's method can be used to find a $\lambda$ satisfying (15) [16]. An

alternative approach is to use the dual form, which involves minimizing

$$Q(\lambda) = -\sum_{t=1}^{n} \log\left(1 + \lambda K_{h,t}(x_0)(y_t^{\text{adj}} - \mu(x_0))\right) \tag{16}$$

over the 1-dimensional Lagrange multiplier $\lambda$, subject to

$$1 + \lambda K_{h,t}(x_0)(y_t^{\text{adj}} - \mu(x_0)) \geq \frac{1}{n} \quad \forall t.$$

The previous constraint stems from $0 \leq p_t \leq 1$ for all $t$. Noting that $-G$ is the derivative of $Q$ with respect to $\lambda$, we can see that a stationary point of $Q$ satisfies (15). [17] showed that with a modification of the $\log(\cdot)$ function, the dual form can be minimized without the constraint and EL then becomes an unconstrained convex problem.

With an undersmoothed bandwidth, $-2\text{LR}_n(f(x_0)) \to \chi_1^2$ in distribution [2]. Thus, a $1-\alpha$ confidence interval is $\left\{\mu(x_0) : -2\text{LR}_n(\mu(x_0)) \leq \chi^2_{1,1-\alpha}\right\}$. To compute an EL confidence interval at some point $x_t$, one must conduct a search for $\mu(x_t)$ such that $-2\text{LR}_n(\mu(x_t))$ equals the $1-\alpha$ quantile of the $\chi^2$ distribution.

Recall that because kernel smoothing is biased, without an undersmoothed bandwidth the confidence interval is actually for $E[\underline{\hat{f}}(x_0)]$ instead of $f(x_0)$. In practice it may be undesirable to use a small bandwidth, so [14] proposed a bias corrected empirical likelihood which uses the following estimating equation in the structural constraint:

$$\sum_{t=1}^{n} K_{h,t}(x_0)(y_t^{\text{adj}} - \underline{\hat{f}}(x_t) + \underline{\hat{f}}(x_0) - r(x_0)) = 0 \tag{17}$$

where $\underline{\hat{f}}(\cdot)$ is the pooled NOIS function estimate. With the above estimating equation, the log EL ratio at a candidate value $\psi(x_0)$ for $r(x_0)$ is:

$$\begin{aligned} \text{LR}_n^*(\psi(x_0)) = \max\Bigg\{ &\sum_{t=1}^{n} \log(np_t) \,\Bigg|\, \sum_{t=1}^{n} p_t = 1,\ p_t \geq 0, \\ &\sum_{t=1}^{n} p_t K_{h,t}(x_0)(y_t^{\text{adj}} - \underline{\hat{f}}(x_t) + \underline{\hat{f}}(x_0) - \psi(x_0)) = 0 \Bigg\}. \end{aligned} \tag{18}$$

The only difference between (18) and the log EL ratio in (12) is the structural constraint. Therefore, the same convex optimization procedure applies and under certain conditions, $-2\text{LR}_n^*(f(x_0)) \to \chi_1^2$ [14].

EL confidence intervals can be computed by a one-dimensional root finding algorithm such as bisection or Brent's method. Each function evaluation in the root finding algorithm solves the convex dual problem in (16), which can be minimized using an optimization routine such as Newton's method with backtracking line search. Computation of EL was performed according to the methods and code from [18]. A drawback is that this process must be repeated $n$ times to construct pointwise confidence bands across the whole estimated curve, which causes EL to be relatively slow compared to the bootstrap. However, these intervals can be computed in parallel which can significantly decrease the computation time.

## 5.3 Bootstrap

In this section we present the predictive residual bootstrap [19] as an alternative to empirical likelihood. The predictive residuals can reduce the under-coverage effect of bootstrap confidence bands in finite samples and we outline the procedure below. First define the leave-one-out estimates

$$\hat{f}_{h,t}(x_t) := \frac{\sum_{j\neq t}^{n} K_h(x_t - x_j) y_j^{\text{adj}}}{\sum_{j\neq t}^{n} K_h(x_t - x_j)} \tag{19}$$

and

$$\hat{M}_{h,t}(x_t) := \frac{\sum_{j\neq t}^{n} K_h(x_t - x_j)(y_j^{\text{adj}})^2}{\sum_{j\neq t}^{n} K_h(x_t - x_j)}. \tag{20}$$

The steps to compute a predictive residual bootstrap confidence interval for a NOIS function estimate are:

1. Using the clean data set $\{(x_t, y_t^{\text{adj}}), 1 \leq t \leq n\}$, construct the leave-one-out function estimates $\hat{f}_{h,t}(x_t)$.
2. Estimate the leave one out standard deviations: $s_x = \sqrt{\hat{M}_{h,t}(x) - (\hat{f}_{h,t}(x))^2}$.
3. Compute the residuals: $e_t = (y_t^{\text{adj}} - \hat{f}_{h,t}(x_t))/s_{x_t}$.
4. Compute the centered residuals: $r_t = e_t - n^{-1}\sum e_t$. For $B$ bootstrap replicates, repeat the following procedure:
   - Sample the residuals $r_1, \ldots, r_n$ with replacement to create the bootstrap residuals $r_1^*, \ldots, r_n^*$.
   - Create the bootstrap pseudo-data by setting $y_t^* = \underline{\hat{f}}(x_t) + s_{x_t} r_t^*$.
   - Use the pseudo-data set $\{(x_t, y_t^*), 1 \leq t \leq n\}$ to compute new function estimates $\underline{\hat{f}}(x_t)^*$.
   - Calculate the bootstrap confidence "roots": $\underline{\hat{f}}(x_t) - \underline{\hat{f}}(x_t)^*$.
5. After computing $B$ replicates, for a desired confidence level $1-\alpha$, calculate the quantiles of the bootstrap confidence roots, given by $q(\alpha)$.
6. A $(1-\alpha)100\%$ confidence interval for a point $f(x_t)$ is

$$\left[\underline{\hat{f}}(x_t) + q(\alpha/2), \underline{\hat{f}}(x_t) + q(1-\alpha/2)\right].$$

Thus, the bootstrap attempts to approximate the distribution of $f(x_t) - \underline{\hat{f}}(x_t)$ by $\underline{\hat{f}}(x_t) - \underline{\hat{f}}(x_t)^*$. Bias corrected confidence intervals can be constructed by computing the residual bootstrap for $\underline{\hat{f}}_b(x)$ instead of $\underline{\hat{f}}(x)$ and replacing the leave-one-out estimates with

$$\hat{f}^b_{h,t}(x_t) := \frac{\sum_{j\neq t}^{n} K_h(x_t - x_j)(y_j^{\text{adj}} - \hat{f}_{h,j}(x_j) + \underline{\hat{f}}(x_t))}{\sum_{j\neq t}^{n} K_h(x_t - x_j)} \tag{21}$$

and

(22)
$$\hat{M}^b_{h,t}(x_t) := \frac{\sum_{j\neq t}^n K_h(x_t - x_j)(y_j^{\text{adj}} - \hat{f}_{h,j}(x_j) + \hat{\underline{f}}(x_t))^2}{\sum_{j\neq t}^n K_h(x_t - x_j)}.$$

As previously illustrated in Figure 8, the standard deviation estimation of the residual bootsrap can sometimes result in unstable confidence bands for stock price data. In the example, $s_{x_{19}}$ is up to 10 times smaller than many of the other $s_{x_t}$'s, causing $r_{19}$ to be quite substantial. Thus, if the residual at $x_{19}$ is resampled and paired with a relatively large $s_{x_t}$, the resulting $y_t^*$ and bootstrap confidence "root" can be extreme. In these situations, a more reasonable confidence band can be found by resampling the residuals without scaling by $s_x$, with the caveat that heteroskedasticity is no longer accounted for.

## ACKNOWLEDGMENTS

We would like to acknowledge and thank an associate editor and a referee for their helpful comments and suggestions. We also thank Dr. Minjing Tao for providing us with the high-frequency data set. This research was supported in part by NSF grants DMS-1352259 and CCF-1617801.

*Received 20 April 2017*

## REFERENCES

[1] Barnett, V., Lewis, T., et al. (1994). *Outliers in statistical data*, volume 3. Wiley New York. MR1272911
[2] Chen, S. X. (1996). Empirical likelihood confidence intervals for nonparametric density estimation. *Biometrika*, 83(2):329–341. MR1439787
[3] Chu, C.-K. and Marron, J. S. (1991). Comparison of two bandwidth selectors with dependent errors. *The Annals of Statistics*, 19(4):1906–1918. MR1135155
[4] Falkenberry, T. N. (2002). High-frequency data filtering. Technical report, Tick Data.
[5] Fan, J., Hu, T.-C., and Truong, Y. K. (1994). Robust nonparametric function estimation. *Scandinavian journal of statistics*, 21(4):433–446. MR1310087
[6] Hall, P. and Jones, M. (1990). Adaptive m-estimation in nonparametric regression. *The Annals of Statistics*, 18(4):1712–1728. MR1074431
[7] Hall, P. and La Scala, B. (1990). Methodology and algorithms of empirical likelihood. *International Statistical Review/Revue Internationale de Statistique*, 58(2):109–127.
[8] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (2011). *Robust statistics: the approach based on influence functions*, volume 114. John Wiley & Sons. MR0829458
[9] Härdle, W. and Bowman, A. W. (1988). Bootstrapping in nonparametric regression: local adaptive smoothing and confidence bands. *Journal of the American Statistical Association*, 83(401):102–110. MR0941002
[10] Hardle, W. and Gasser, T. (1984). Robust non-parametric function fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(1):42–51. MR0745214
[11] Huber, P. J. (2011). *Robust statistics*. Springer. MR1491392
[12] Kirkpatrick II, C. D. and Dahlquist, J. (2010). *Technical analysis: the complete resource for financial market technicians*. FT press.
[13] Lamoureux, C. G. and Lastrapes, W. D. (1990). Heteroskedasticity in stock return data: volume versus garch effects. *The Journal of Finance*, 45(1):221–229.
[14] Lian, H. (2012). Empirical likelihood confidence intervals for nonparametric functional data analysis. *Journal of Statistical Planning and Inference*, 142(7):1669–1677. MR2903379
[15] Lo, A. W., Mamaysky, H., and Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The Journal of Finance*, 55(4):1705–1770.
[16] Owen, A. B. (1988). Empirical likelihood ratio confidence intervals for a single functional. *Biometrika*, 75(2):237–249. MR0946049
[17] Owen, A. B. (1990). Empirical likelihood ratio confidence regions. *The Annals of Statistics*, 18(1):90–120. MR1041387
[18] Owen, A. B. (2013). Self-concordance for empirical likelihood. *Canadian Journal of Statistics*, 41(3):387–397. MR3101590
[19] Politis, D. N. (2014). Bootstrap confidence intervals in nonparametric regression without an additive model. In *Topics in Nonparametric Statistics*, pages 271–282. Springer. MR3333354
[20] Puigvert, J. and Fortiana, J. (2008). Clustering techniques applied to outlier detection of financial market series using a moving window filtering algorithm. Technical Report 948, ECB.
[21] R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
[22] Schwert, G. W. and Seguin, P. J. (1990). Heteroskedasticity in stock returns. *The Journal of Finance*, 45(4):1129–1155.
[23] She, Y. (2017). Selective factor extraction in high dimensions. *Biometrika*, 104(1):97–110. MR3626471
[24] She, Y. and Chen, K. (2017). Robust reduced rank regression. *Biometrika*, 104(3):633–647. MR3694587
[25] She, Y. and Owen, A. B. (2011). Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association*, 106(494):626–639. MR2847975
[26] StockCharts.com (2016). Chart patterns.
[27] Wang, Y. and Zou, J. (2010). Vast volatility matrix estimation for high-frequency financial data. *The Annals of Statistics*, 38(2):943–978. MR2604708
[28] Yao, Q. and Tong, H. (1998). Cross-validatory bandwidth selections for regression estimation based on dependent data. *Journal of Statistical Planning and Inference*, 68(2):387–415. MR1629607
[29] Yu, C., Fang, Y., Li, Z., Zhang, B., and Zhao, X. (2014). Nonparametric estimation of high-frequency spot volatility for brownian semimartingale with jumps. *Journal of Time Series Analysis*, 35(6):572–591. MR3300688

Hoang Tran
Department of Statistics
Florida State University
Tallahassee, Florida
USA
E-mail address: hoang.t.tran@icloud.com

Yiyuan She
Department of Statistics
Florida State University
Tallahassee, Florida
USA
E-mail address: yshe@stat.fsu.edu