SRSARA: A SARA-INSPIRED MODIFICATION OF PETTITT'S TEST FOR NON-PARAMETRIC CHANGE-POINT DETECTION

by

Elliot Owen Kennedy

A Master's Thesis Submitted to the Faculty of the

DEPARTMENT OF MATHEMATICS

In Partial Fulfillment of the Requirements

For the Degree of

MASTER'S OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

2024

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Master's Committee, we certify that we have read the thesis prepared by: Elliot Kennedy
titled: SRSaRa: A SaRa-Inspired Modification of Pettitt's Test for Non-Parametric Change-Point Detection

and recommend that it be accepted as fulfilling the thesis requirement for the Master's Degree.

Ning Hao
_______________________________ Date: Jul 29, 2024
Ning Hao

Xueying Tang (Jul 30, 2024 10:14 PDT)
_______________________________ Date: Jul 30, 2024
Xueying Tang

Yue Niu (Jul 29, 2024 15:01 PDT)
_______________________________ Date: Jul 29, 2024
Yue Niu

Final approval and acceptance of this thesis is contingent upon the candidate's submission of the final copies of the thesis to the Graduate College.

I hereby certify that I have read this thesis prepared under my direction and recommend that it be accepted as fulfilling the Master's requirement.

Ning Hao
_______________________________ Date: Jul 29, 2024
Ning Hao
Thesis Committee Chair
Department of Mathematics and GIDP in Statistics and Data Science

ACKNOWLEDGMENTS

Thank you to Dr. Ning Hao for all his help and support on this project. Additionally, thank you to my committee members, Dr. Tang and Dr. Niu, for their feedback. Finally, thank you to my family for believing in me and encouraging me every step of the way.

LAND ACKNOWLEDGMENT

We respectfully acknowledge the University of Arizona is on the land and territories of Indigenous peoples. Today, Arizona is home to 22 federally recognized tribes, with Tucson being home to the O'odham and the Yaqui. Committed to diversity and inclusion, the University strives to build sustainable relationships with sovereign Native Nations and Indigenous communities through education offerings, partnerships, and community service.

DEDICATION

Dedicated to anyone experiencing points of change in their lives.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The Signed-Rank Screening and Ranking Algorithm or SRSaRa is a non-parametric change-point detection technique that is based on a SaRa-like process with a diagnostic function inspired by Pettitt's test. Possessing two modes, 'LM' and 'MAX' for single and multiple change-point detection respectively, the SRSaRa is flexible and robust to outliers through its diagnostic function. The SRSaRa's 'MAX' mode for single change-point detection outperforms Pettitt's test in several scenarios while maintaining Type-I error control, while the SRSaRa's 'LM' mode is capable of controlling FDR at the desired level and shows promise as a non-parametric multiple change-point detection technique.

Chapter 1

Introduction

In the 70 years since Page's seminal *Continuous Inspection Schemes* was published (Page, 1954), the number of change-point detection techniques available has continued to grow in an attempt to match the ever increasing variety of data sets that may contain change-points. Originally used in quality control applications, change-point detection has been applied in disciplines ranging from linguistics to genomics. Since there is no one-size-fits-all change-point test, new detection techniques are developed to suite the unique problems posed by new data sets (Sen and Srivastava, 1975). This innovation has led to techniques such as Circular Binary Segmentation (Olshen et al., 2004), PELT (Killick et al., 2012), and the Screening and Ranking Algorithm (SaRa) (Niu and Zhang, 2012), among others. For more on detecting single change-points, we refer to Bhattacharya (1994) and for an excellent overview of multiple change-point detection, we refer to the work of Niu et al. (2016).

Non-parametric change-point tests are a point of intrigue for this paper. Soon after Page's 'CUMSUM' based change-point detection techniques became known, non-parametric change-point techniques began to surface such as in the work of Bhattacharya and Johnson (1968), Pettitt (1979) and later Aly and BuHamra (1996). More recently, the works of Matteson and James (2013) and Wang et al. (2020) contributed to the existing battery of non-parametric change-point tests. In this paper, we propose a new non-parametric change-point detection technique that is an amalgam of the Screening and Ranking Algorithm (Niu and Zhang, 2012) and Pettitt's test (Pettitt, 1979). Before we describe this new test, which we call the Signed-Rank Screening and Ranking Algorithm (SRSaRa), we must first describe Pettitt's test and the SaRa in more detail.

1.1 Pettitt's Test

The eponymous Pettitt's Test was first described by A. N. Pettitt in his 1979 paper *A Non-Parametric Approach to the Change-Point Problem* (Pettitt, 1979). Pettitt proposed using a version of the two-sample Mann-Whitney statistic to test the null hypothesis of "no change" against the alternative of "one change exists" in a non-parametric manner. Consider the sequence of random variables $X_1, \ldots, X_n$, to test H_0 v.s. H_1, Pettitt proposes the following. Let

$$D_{ij} = \operatorname{sgn}\left(X_i - X_j\right)$$

where $\operatorname{sgn}(x) = 1$ if $x > 0$, 0 if $x = 0$ and -1 if $x < 0$, then

$$U_{t,T} = \sum_{i=1}^{t} \sum_{j=t+1}^{T} D_{ij}.$$

This gives the test statistic,

$$K_T = \max_{1 \le t < T} |U_{t,T}|$$

Pettitt notes that under H_0, $\mathbb{E}(D_{i,j}) = 0$ and the distribution of $U_{t,T}$ is symmetric about zero for each t. Additionally, large values of K_T indicate a change-point (a favorable property for use in a SaRa like structure). Pettitt also provides a p-value approximation for the $\alpha = 0.05$ level test:

$$p \simeq 2 \exp\left\{-6K_T^2 / \left(T^3 + T^2\right)\right\}$$

Pettitt's test has become a mainstay of non-parametric change-point detection due to its simplicity and robustness to outliers. Computationally, it is incredibly quick to calculate - requiring only a single iteration over the data array. Additionally, when there is a change-point near the middle of a large sequence of data, Pettitt's test has a very high detection probability. However, as we will demonstrate in the Comparison Section 2.5, due to the global-information approach of Pettitt's test, it has difficulty detecting change-points near the start or end of the data array, and in the case where there are actually two change-points, it can fail to detect them entirely. Clearly, there is room for improvement beyond Pettitt's test.

1.2 SaRa: The Screening and Ranking Algorithm

Proposed in 2012 by Yue S. Niu and Heping Zhang in their paper *The Screening and Ranking Algorithm to Detect DNA Copy Number Variations*, the Screening and Ranking algorithm (SaRa) adopts a local information approach for multiple change-point detection (Niu and Zhang, 2012). As the title of the paper suggests, the SaRa was developed for detecting copy number variations in genetic data, however it is also applicable to a wide variety of change-point detection problems. The SaRa is a two step process comprised of a screening step and a ranking step, in that order. The SaRa differs notably from Pettitt's test in two manners: 1. it employs a local information approach and 2. it is usually a parametric test. This local information approach is achieved through the use of a sliding window and a local diagnostic function to assess the likelihood of a change-point occurring within the aforementioned window. An example diagnostic function is,

$$D(x) = \sum_{k=1}^{h} \frac{Y_{x+1-k}}{h} - \sum_{k=1}^{h} \frac{Y_{x+k}}{h}$$

the difference between averages of h data points on the left and right of x. Similar to Pettitt's K_T statistic, large values of $D(x)$ are also indicative of a change-point. Loosely, the Screening and Ranking algorithm is comprised of the following steps,

1. Select a fixed integer window size h
2. Screening: Calculate $D(x)$ for each x in the array
3. Ranking: Rank the local maximizers of $|D(x)|$
4. Use a thresholding rule to determine which $|D(x)|$ are significant

The thresholding rule in question is typically the Benjamini–Hochberg procedure, allowing for false discovery rate (FDR) control at a desired level. The FDR control guarantee of the SaRa enables it to be used for multiple change-point detection.

Similar to Pettitt's test, the SaRa is capable of change-point detection in linear time. The SaRa has an advantage over Pettitt's test in its ability to detect change-points near the start and end of the data sequence via the local information approach. Pettitt's test has the advantage of being a non-parametric test and is robust to outliers in the data sequence, however it is not capable of multiple change-point detection. What if we could combine the favorable properties of the SaRa with the favorable properties of Pettitt's test - constructing a non-parametric change-point detection method capable of detecting location changes near the start and end of the data array, while also being able to detect multiple location changes in the middle of the array?

1.3 SRSaRa: The Signed-Rank Screening and Ranking Algorithm

We propose using a SaRa-like procedure with a diagnostic function based on the sum of signed differences akin to Pettitt's test for non-parametric change-point detection. We call this procedure the Signed-Rank Screening and Ranking algorithm or SRSaRa. In order for a diagnostic function to be appropriate within a SaRa-like process, extreme values of the diagnostic function should be indicative of change-points. Extreme values of Pettitt's test statistic are indicative of change-points, but we must modify the way the statistic is calculated to make it work within a SaRa-like structure. First, instead of calculating Pettitt's test statistic as the maximum of all sums of signed differences across each possible break in the array, we will calculate our diagnostic function within a sliding window of even integer width. Second, we will opt to calculate the sum of signed differences across only the middle of the window. Structuring the calculation of the diagnostic function in this way allows us to leverage the SaRa's local information approach. As the window moves along the array, we ask whether there is a change-point at the middle of the window. If there is a change-point, we will receive a extreme diagnostic function value for that window, but how do we know if an extreme value is truly extreme enough to indicate a change-point? Through the use of two different thresholding methods, the SRSaRa is capable of both single and multiple change-point detection. The first, for single change-point detection, uses repeated permutations of the data array to calculate the empirical distribution of absolute global maximum diagnostic function values for a given window size, from which we may draw the threshold value that controls Type-I error at the desired level. For multiple change-point detection we make use of a similar thresholding process, this time relying on the distribution of absolute local maximum diagnostic function values, from which we may draw the threshold value that controls False Discovery Rate (FDR) at the desired level. In the following chapters, we detail the intricacies of the Signed-Rank Screening and Ranking Algorithm. In Chapter 2, we outline the process for single change-point detection with the SRSaRa, demonstrate the global maximum thresholding procedure's ability to control Type-I error, illustrate several scenarios where the SRSaRa outperforms Pettitt's test, and perform single change-point detection with the SRSaRa on two classic change-point detection data sets. In Chapter 3, we detail the SRSaRa's multiple change-point detection process and its corresponding thresholding procedure, demonstrate the SRSaRa's ability to control FDR, and compare the SRSaRa to its predecessor the SaRa. Through this paper, we hope to highlight the SRSaRa's excellent single change-point detection abilities and call attention to the SRSaRa's promising multiple change-point detection capabilities.

Chapter 2

Single Change-Point Detection

Consider a sequence of random variables $\mathbf{X} = X_1, \dots, X_n$. The sequence is said to have a change-point at τ if for $t = 1, \dots, \tau$, X_t follow a common distribution function $F_1(x)$ and for $t = \tau + 1, \dots, n$, X_t follow another common distribution function $F_2(x)$, where $F_1(x) \neq F_2(x)$. We are faced with the problem of testing the null-hypothesis of "no change", $H_0 : \tau = n$ against the alternative of "one change exists", $H_1 : 1 \leq \tau < n$ in a non-parametric manner. We make no assumptions regarding the form of F_1 and F_2 other than that they are continuous. In this chapter, we propose using the 'MAX' mode of SRSaRa (Signed-Rank Screening and Ranking Algorithm) to test the hypothesis of "no change" against the hypothesis of "one change exists".

The procedure for single change-point detection using global maximum thresholding is outlined below,

1. For $\mathbf{X} = X_1, \dots, X_n$, choose an even fixed window size $w < n$ and let $h = \frac{w}{2}$
2. Calculate U_k for $k = h, h+1, \dots, n-h$ where

$$U_k = D(\mathbf{X}, k) = \sum_{i=k-h+1}^{k} \sum_{j=k+1}^{k+h} \operatorname{sgn}(X_i - X_j).$$

3. Permute $\mathbf{X}$ many times, calling one instance of the permuted array $\mathbf{X}'$.
4. Calculate $U'_k = D(\mathbf{X}', k)$ for each permutation and store $U'_{k,\text{MAX}} = \max |U'_k|$.
5. The $(1-\alpha)$th quantile of all $U'_{k,\text{MAX}}$ is the rejection threshold U_{Thresh} to control Type-I error at level α.

6. If $\max |U_k| > U_{\text{Thresh}}$, reject H_0

2.1 U Statistic Calculation

Since we must calculate both the U_k statistics for the original ordered array and the U_k' statistics for each permutation used for thresholding, it is essential that the code for calculating these U_k statistics is fast. Initially, a naive triple-loop implementation was used, however, we quickly found a faster way to calculate these U_k statistics in base R. The second base R implementation, based on rank statistics is shown as pseudo-code in Algorithm 1. This algorithm calculates the U_k statistics and returns them in a two-dimensional list that also includes a list of τ values which correspond to the index in the original array around which each U_k statistic was calculated.

Algorithm 1 Base R Rank Based U Statistic Calculation Code

```
procedure U_CALC(x, w)
    assert no missing values in x
    n ← length of x
    U ← array of size (n − w + 1)
    k ← sequence from w/2 to n − (w/2)
    if n < w then
        stop with error message "Requires n > w"
    end if
    for i ← 1 to length of k do
        if k[i] + (w/2) > n then
            break
        end if
        r ← rank of subarray x[(k[i] − (w/2) + 1) : (k[i] + (w/2))]
        U_k ← −(2 · sum of first half of r − (w/2) · (w + 1))
        U[i] ← U_k
    end for
    return {U, k}
end procedure
```

Now that we have the ability to calculate these U_k statistics, we shall examine the graph of these U_k statistics in order and their corresponding histograms at a variety of window sizes. For a given window size w, the most extreme possible diagnostic function value is $\pm(w/2)^2$. In Figure 2.1, we can see that at lower window size w, the plot of U_k statistics in the original array order appears to be more jagged and haphazard. With window size $w = 32$, when the

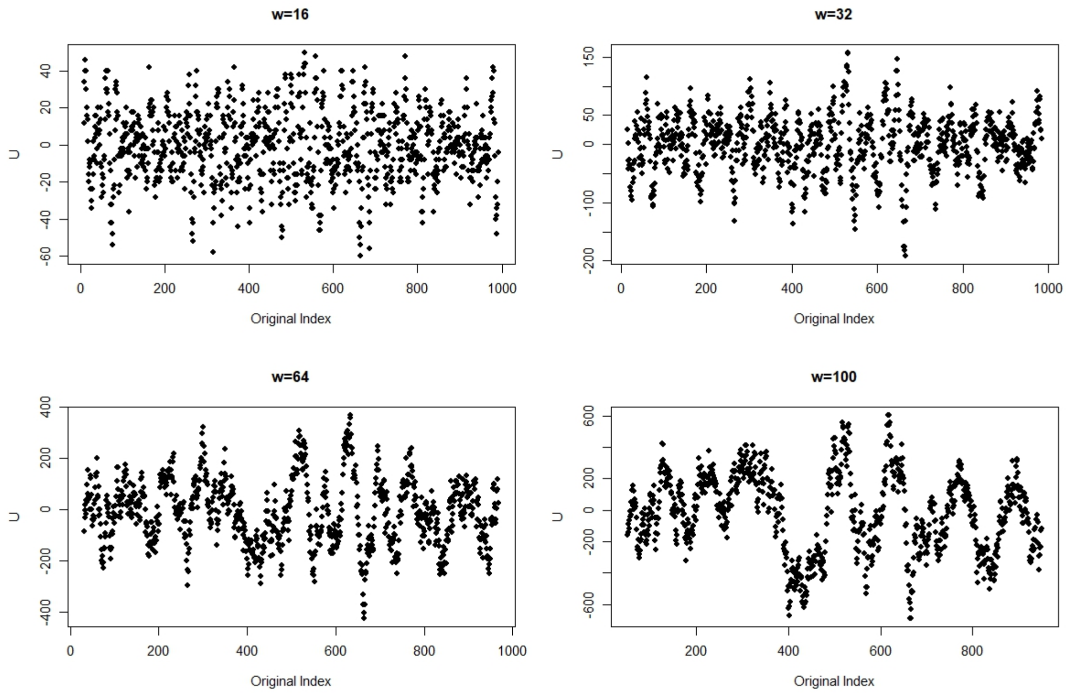

Figure 2.1: U_k Statistics in Array Order for $w = 16, 32, 64, 100$

window shifts over by one, the inclusion of a new data point and the exclusion of an old data point can change the value of U_k by far more relative to the most extreme possible value when compared to larger window sizes. When the window size increases, the inclusion of a new data point and the exclusion of an old data point changes the value of U_k less relative to the most extreme possible value. This can be seen visually in the more smooth plots at higher window sizes. The histograms for each of the plots in Figure 2.1 may be seen in Figure 2.2.

2.2 Global Max Thresholding

The global maximum permutation thresholding procedure for single change-point detection relies heavily on the calculation of U_k statistics described above and as such does not require as much attention to detail. To calculate the thresholds for absolute global maximum U_k values, we employ a permutation thresholding process. For a given window size w, one replicate of the thresholding process is detailed below:

1. Permute the data array $\mathbf{X}$, call the permuted array $\mathbf{X}'$

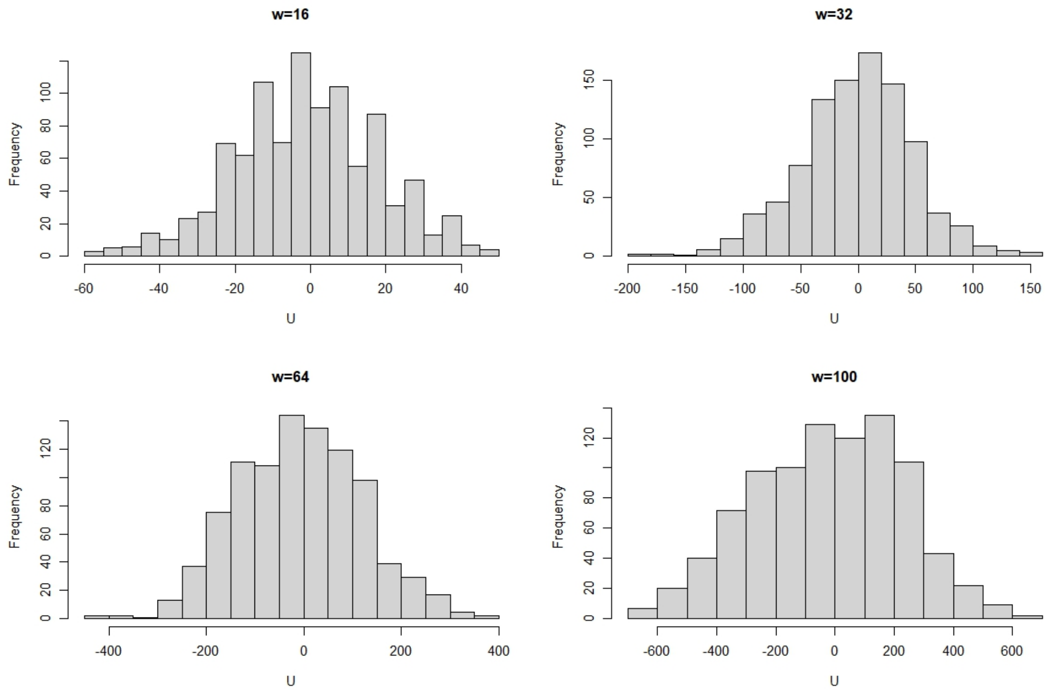

Figure 2.2: U_k Statistics Histograms for $w = 16, 32, 64, 100$

2. Calculate U'_k where

$$U'_k = D(\mathbf{X}', k) = \sum_{i=k-h+1}^{k} \sum_{j=k+1}^{k+h} \operatorname{sgn}(X'_i - X'_j)$$

for $k = h, h+1, \ldots, n-h$ and $h = w/2$

3. Store $\max |U'_k|$

Once we have performed the desired number of replicates for the thresholding process, we obtain a empirical distribution of absolute global maximum U_k values for the chosen window size w. For a desired Type-I error control rate α, we may calculate the $1-\alpha$ quantile of this empirical distribution of absolute global maximum U_k values to obtain the critical U_{Thresh} value.

When using the SRSaRa for single change-point detection, it is important to note that both the window size w and the array length n effect the threshold value U_{Thresh}. Please refer to Figure 2.3 for an illustration of the effect of array length on the calculated threshold value for three different window sizes. As we may see from the graph, the threshold values tend to

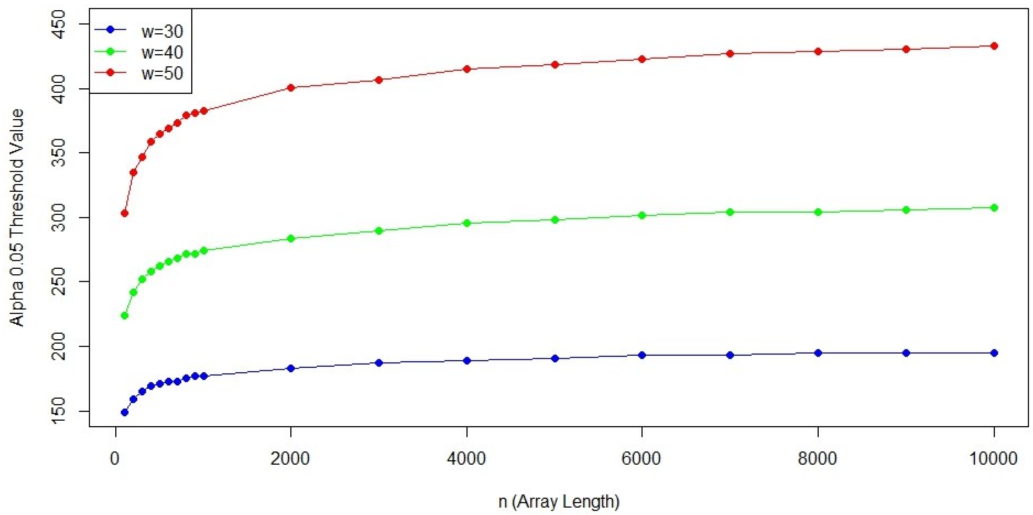

Figure 2.3: Plot of $\alpha = 0.05$ Threshold Value vs Array Length

increase rapidly as array length increases until around $n = 10000$ where the threshold values continue to increase but at a much slower rate. For small data arrays, it is more difficult to observe extreme values of $\max |U'_k|$ under H_0, leading to smaller threshold values. For large arrays, we have many more chances to observe large $\max |U'_k|$ values - leading to higher thresholds. For example, with window size $w = 30$ on data arrays of length $n = 10000$, the threshold for controlling Type-I error at the $\alpha = 0.05$ level is $U_{\text{Thresh}} = 195$. In contrast, when we perform the thresholding process > 20000 times on an array of length $n = 100$ with $w = 30$, the threshold for controlling Type-I error at the $\alpha = 0.05$ level is ≈ 149. Holding all else constant, with an array of length $n = 200$, the corresponding threshold value is ≈ 159. As we can see, threshold values for small arrays can increase dramatically with array length, while threshold values for arrays of length $n > 10000$ will still increase, just at a slower rate.

2.3 Type-I Error Control for SRSaRa

We will now demonstrate that using the procedure described above, the 'MAX' mode of the SRSaRa is capable of controlling Type-I error at or below the desired $\alpha = 0.05$ level on both $\mathcal{N}(0, 1)$ and highly tailed t_3 data. For both distributions, we tested the SRSaRa's Type-I error control with window sizes $w = 30, 60, 100$ and array length $n = 10000$, performing 2000 trials using global maximum thresholds calculated over 20000 replicates. A false-positive was counted when the SRSaRa returned a non-empty list. The Type-I error rates of the SRSaRa under these conditions are shown in Table 2.1.

Distribution	Window Size	Threshold	Type-I Error
N(0,1)	30	195	0.0470
N(0,1)	60	570	0.0495
N(0,1)	100	1224	0.0490
t_3	30	195	0.0490
t_3	60	570	0.0480
t_3	100	1224	0.0470

Table 2.1: Table of Type-I Error Rates for $\mathcal{N}(0, 1)$ and t_3 Data

From the results in Table 2.1, we can see that SRSaRa is successful in controlling Type-I error for two distributions and for a variety of window sizes.

To show what a typical "fail to reject" test looks like for $\mathcal{N}(0, 1)$ and t_3 data with window sizes $w = 64, 100$, refer to Fig. 2.4 and Fig. 2.5. In these plots, we can see the $|U_k|$ statistics in black and the threshold in blue. The threshold value is labeled and no $|U_k|$ exceeds the threshold, thus we fail to reject H_0 for these two data sets and conclude that there is no change-point. This is the correct conclusion. Notice how the threshold value is identical for the same window size on both types of data. This indicates that the threshold value, for arrays of the same length, is only dependent on the choice of window size w regardless of the form of the data.

2.4 SRSaRa: Detection Visualization

We will now show what a typical detection of a change-point looks like graphically when using the SRSaRa. For this visualization, we generate a sequence of $X_1, \ldots, X_{1000}$ where $X_1, \ldots, X_{500} \sim \mathcal{N}(0, 1)$ and $X_{501}, \ldots, X_{1000} \sim \mathcal{N}(1, 1)$ thus the sequence has a mean change at $\tau = 500$. We will now go through the process of the SRSaRa for single change-point detection step-by-step, demonstrating how a change-point is detected on this dataset with window sizes $w = 64$ and 100.

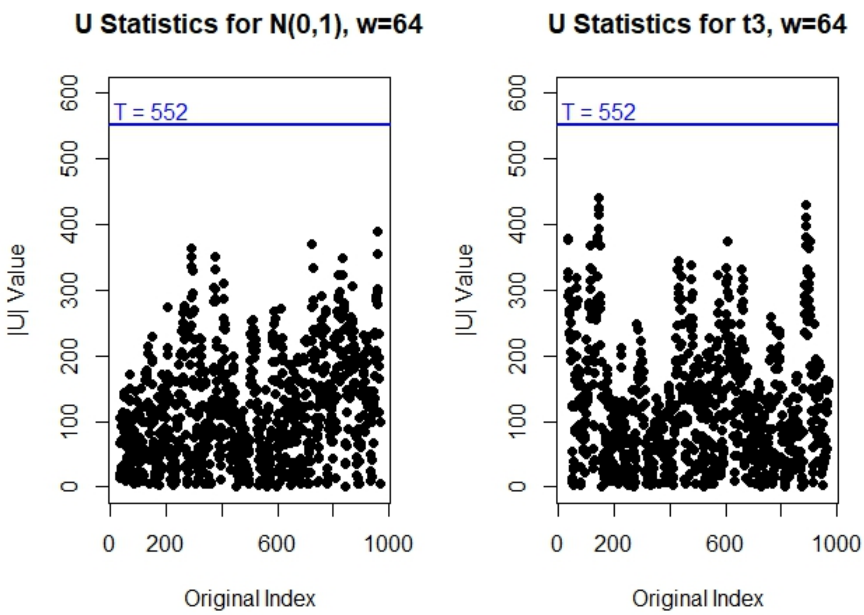

Figure 2.4: U_k Statistics for $w = 64$ on $\mathcal{N}(0, 1)$ and t_3 with Threshold in Blue

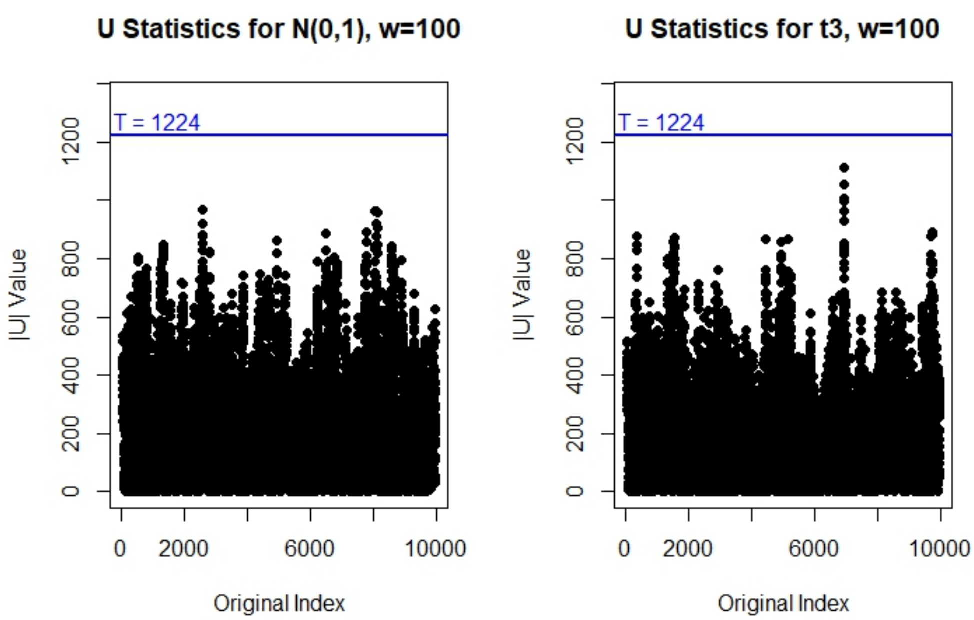

Figure 2.5: U_k Statistics for $w = 100$ on $\mathcal{N}(0, 1)$ and t_3 with Threshold in Blue

First, we calculate the U_k statistics for the array. The plots of U_k statistics, in order, for $w = 64, 100$ may be seen in Figure 2.6 with the location of the change-point indicated by the red line. Notice the "spike" in U_k values around this change-point, with the global maximum for U_k occurring at the red line in both graphs.

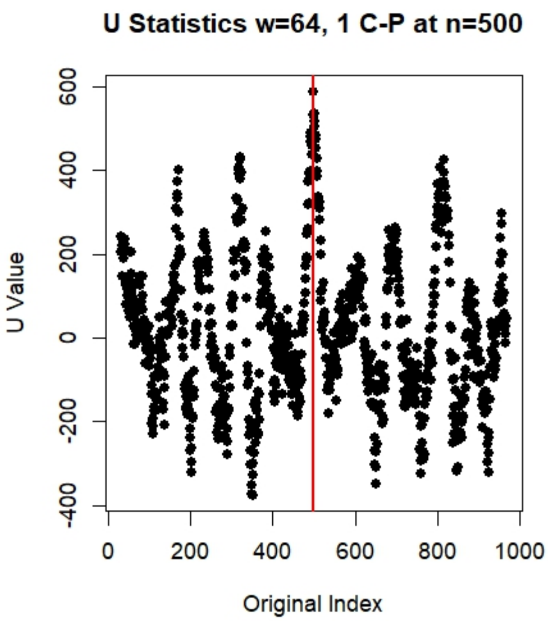

Figure 2.6: U_k Statistics for $w = 64, 100$ Data with 1 Change-Point at the Red Line

Next, we calculate the absolute U_k statistics for $w = 64, 100$, $|U_k|$. Large values of $|U_k|$ indicate a change-point. In Figure 2.7, notice how the global maximum for $|U_k|$ occurs exactly at the red line which indicates the position of the change-point in the array of data. So far, so good.

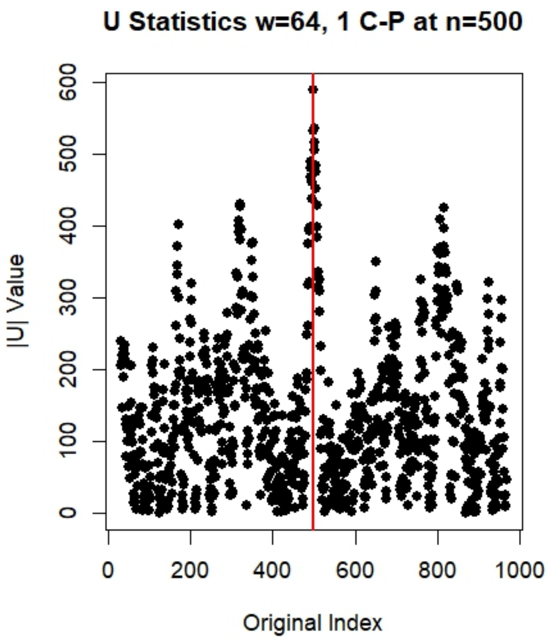

Figure 2.7: $|U_k|$ Statistics for $w = 64, 100$ Data with 1 Change-Point at the Red Line

Now we calculate the level $\alpha = 0.05$ threshold $U_{\text{Thresh}} = |U_{k',\text{MAX}}|$ for both $w = 64$ and 100 by permuting the original data array many times. This calculated threshold is indicated by the blue line in Figure 2.8. Notice how in both the graphs for window size 64 and 100, the global maximum of $|U_k|$ exceeds the threshold - indicating rejection of H_0 and detection of a change-point.

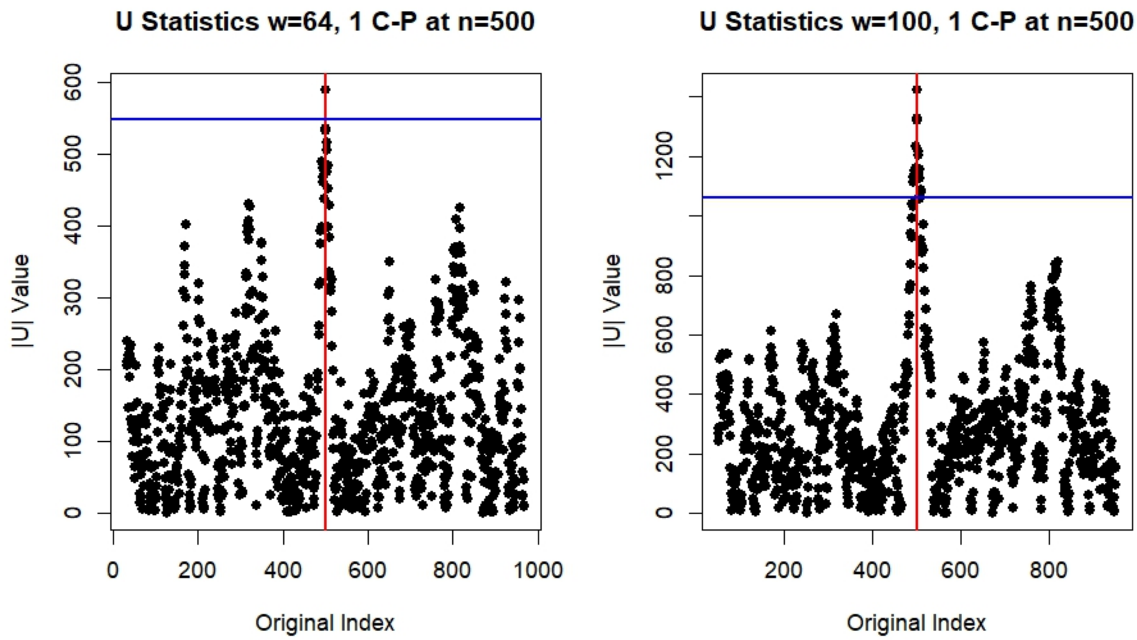

Figure 2.8: $|U_k|$ Statistics for $w = 64, 100$ Data with Change-Point at Red Line, Threshold in Blue

With this, we demonstrate that the SRSaRa is capable of change-point detection in a visually intuitive manner. Now that we know that the SRSaRa is capable of detection and Type-I error control, we may compare its detection ability with that of Pettitt's test.

2.5 SRSaRa v.s. Pettitt's Test

In this section, we will compare the ability of the SRSaRa and Pettitt's test to detect change-points in a variety of different situations. Before we may make these comparisons, however; it is important to ensure that Pettitt's test is also capable of controlling Type-I error at or below the $\alpha = 0.05$ level. To do this, we used the `pettitt.test()` command from the 'trend' package on arbitrary arrays of $\mathcal{N}(0, 1)$ and t_3 data with no change-points and length $n = 1000$. Performing 500 replicates on each data set, Pettitt's test had Type-I error rates of 0.042 and 0.05 on $\mathcal{N}(0, 1)$ and t_3 data respectively. Now that we are sure Pettitt's test is capable of adequate Type-I error control, we may proceed with our comparisons. Comparisons will be performed on an array with length $n = 1000$ of $\mathcal{N}(0, 1)$ data with a location change to $\mathcal{N}(1, 1)$ at select points along the array. We first compare each tests

Change Location	Pettitt Detection	SRSaRa Detection
500	1.00	0.93
50	0.80	0.90
25	0.15	0.37
500, 550	0.21	0.98
500, 525	0.11	0.55

Table 2.2: Detection Comparison: SRSaRa v.s. Pettitt's Test

ability to detect a change-point exactly in the middle of the data array, then at 50 entries from the start of the array, followed by only 25 entries from the start of the array. Finally, we compare the ability of both tests to detect two change-points, in the middle of the array, first with length 50, then with length 25. The goal of these comparisons is to emphasize the strength of the SRSaRa's local information approach and ability to adapt to a variety of data structures. For each comparison, due to limited time, the SRSaRa was only allowed 1000 replicates for thresholding. In practice, > 10000 replicates should be allowed for threshold calculation. For each situation, the SRSaRa and Pettitt's test were performed 500 times to assess their detection abilities. All tests were performed with window size $w = 50$ for the SRSaRa. The results of these comparisons may be found in Table 2.2.

From Table 2.2, it is apparent that the SRSaRa performs better than Pettitt's test in nearly every circumstance. We must, however, give credit where it is due. Pettitt's test has an astounding ability to detect change-points that occur exactly in the middle of the data array - detecting the location change in all 500 trials. When the change-point is not exactly in the middle of the array, the SRSaRa clearly outperforms Pettitt's test. Not only does the SRSaRa detect short changes occurring near the start (and end) of the data array with higher frequency than Pettitt's test, it also detects two change-points in the middle of the array at a higher rate than Pettitt's test. This latter part is especially intriguing. Often, in practical situations, we do not know if an array has one change-point or multiple. If there is more than one change-point, Pettitt's test has a very poor ability to detect even one of them. The SRSaRa on the other hand, due to its local information approach, has a high detection rate for relatively short change-points in the middle of the array. From this, we can conclude that in most situations, the SRSaRa is a more robust test than Pettitt's test.

2.6 Application: Nile River & UK Driver Deaths Data Sets

We will now use the 'MAX' mode of the SRSaRa to perform single change-point detection on two data sets available in R's 'datasets' package: the Nile river discharge data set and the UK driver deaths data set. Both data sets have one change-point caused by an event that occurs at a known time. This allows us to verify if our detection of a change-point is correct. We aim to demonstrate that the SRSaRa is capable of highly accurate change-point detection on real world data.

2.6.1 Nile River Data

The Nile river discharge data set measures the annual flow of the river Nile at the Aswan dam from 1871 to 1970. The data set consists of 100 measurements in this time period. Prior to the Aswan dam's construction in 1898, the Nile flowed freely. After the dam's construction, the flow was now controlled by the dam causing a change in measured annual flow. We aim to detect this change using the SRSaRa. Since the data set contains 100 observations, we must select a window size small enough to fit within the 100 measurements, but large enough to still detect changes. Window size $w = 30$ should be sufficient. To perform single change-point detection with SRSaRa on this data set, we run the following command: `SRSaRa(datasets::Nile, mode = "MAX", w = 30, nreps = 10000, q = c(0.95))`. This command runs the SRSaRa in 'MAX' mode with a window size of 30, allowing 10000 replicates for thresholding and controlling Type-I error at the $\alpha = 0.05$ level.

```
$thresh_148
$thresh_148$U
[1] -186

$thresh_148$tau
[1] 28
```

Figure 2.9: Nile River Detection Result

The results of this test are shown in Fig. 2.9. In the first line of the results we see the threshold value calculated via absolute global maximum permutation thresholding by the SRSaRa, $U_{\text{Thresh}} = 148$. Note that this threshold value for the length $n = 100$ Nile River data set is nearly identical to the one calculated on an arbitrary data array in Section 2.2 and shown in Figure 2.3. In the second line we see the maximum $|U_k|$ value calculated by the SRSaRa prior to taking the absolute value, -186. While the test for significance is done by comparing the global maximum absolute U_k value to the threshold value, the global maximum U_k value (if significant) is returned in its pre-absolute value form since this can convey additional information about the change-point. Negative values indicate a negative location shift for the change-point, while positive values indicate a positive location shift for the change-point. In this case, the value of -186 indicates a negative location shift in the data or a decrease in measured annual flow rate at the time of the change-point. The final entry in the results is the location of the change-point, returned as the index in the original data array where the change-point was detected: 28. Since the data has one observation per year, starting in 1871, we simply add $1871 + 28 = 1899$. The SRSaRa indicates that the change-point occurred in 1899, which corresponds to the first year after the Aswan dam was built. With this, we have shown that the SRSaRa is capable of detecting a single change-point in a real-world data set with high accuracy.

2.6.2 UK Driver Deaths Data

The UK driver deaths data set is time series data measuring the monthly total of car drivers killed or seriously injured in the UK between Jan 1969 and Dec 1984. The change-point in this data occurs when seat belts were made mandatory in the UK on Jan 31, 1983. We aim to use the SRSaRa to detect this change. This data set contains 192 observations, so it is possible to use a slightly larger window than used for the Nile river data, however, for the sake of consistency, we will use $w = 30$ once more. Additionally, since $n = 192 < 2000$, we will again perform the thresholding step manually. We run the test using the following command: `SRSaRa(datasets::UKDriverDeaths, mode = "MAX", w = 30, nreps = 10000, q = c(0.95))`. Once again, we allow 10000 replicates for thresholding and control Type-I error at the $\alpha = 0.05$ level.

```
$thresh_159
$thresh_159$U
[1] -192

$thresh_159$tau
[1] 169
```

Figure 2.10: UK Driver Deaths Detection Result

The results for this test are shown in Fig. 2.10. For this data set, we have $U_{\text{Thresh}} = 159$ with a global maximum absolute U_k value of 192. This value is greater that the threshold, so we have detected a change-point. Again, the threshold value calculated for the length $n = 192$ UK Driver Death data set is nearly identical to the threshold calculated on an arbitrary length $n = 200$ array in Section 2.2. The negative U_k value returned indicates a negative location shift in the data, i.e. a decrease in the monthly driver deaths recorded after the change-point. Finally, the SRSaRa indicates that the change-point occurs at index 169 in the original array. This corresponds to the change-point occurring $(169/12 = 14 + 1/12)$ 14 years and 1 month after the start of the data set. The SRSaRa finds that the change-point occurs in February of 1983, which is the first month of measurement after seat belts became mandatory in the UK. Once again, we have shown that the SRSaRa can detect change-points in real-world data with high accuracy.

2.7 Computation Time & Time Complexity

We now briefly discuss the computational efficiency of 4 different U_k statistic calculation functions as well as the time complexity of the SRSaRa as a whole. As mentioned in Sec. 2.1, we originally implemented two functions to calculate U_k statistics in base R. One utilized three "for" loops and the other made use of one "for" loop (as seen in Algorithm 1). As expected, the rank based single "for" loop implementation was quite a bit faster than the naive triple "for" loop implementation. We then implemented the original naive triple "for" loop code using the 'Rcpp' package in R, allowing the code to be compiled in `C++`. `C++` is

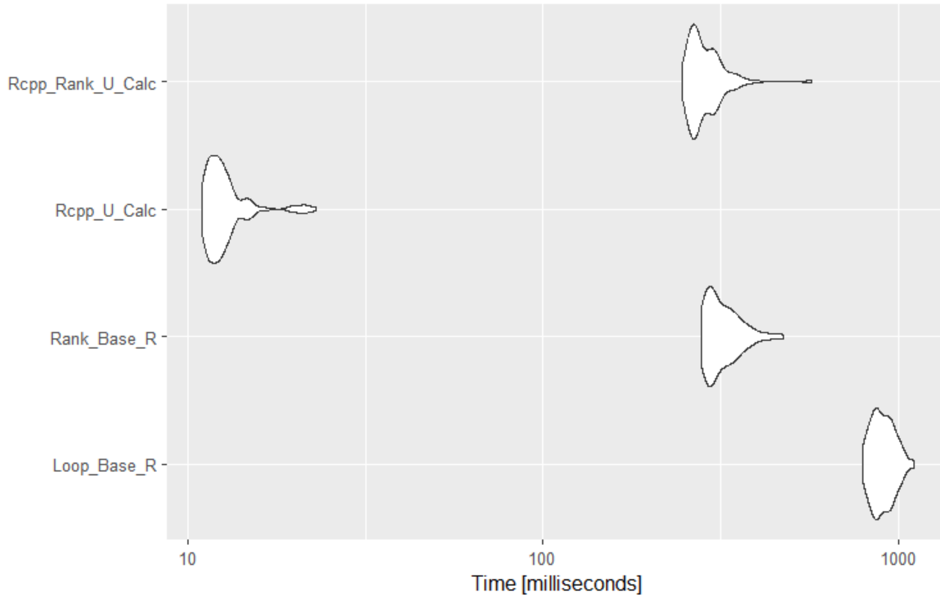

Figure 2.11: Microbenchmark Time Plot for U Statistic Implementations

usually much faster than base R code. We also implemented the single "for" loop rank based implementation using the 'Rcpp' package. Since `C++` does not have a native `rank()` function like that available in base R, we called base R's `rank()` function within the 'Rcpp' code block, allowing us to only use a single "for" loop. After confirming that the four aforementioned methods gave the same results for both U_k values and τ values, we use the 'microbenchmark' package to compare the efficiency of these implementations. The plot of calculation times for each implementation on an array of size $n = 10000$ over 100 replicates is shown in Fig. 2.11.

From the benchmark plot in Fig. 2.11, we notice that the triple "for" loop `C++` implementation via 'Rcpp' is more than $10\times$ faster than the best base R implementation. The single "for" loop 'Rcpp' function that uses base R's `rank()` function is similar in speed to the single "for" loop base R implementation. We will use the naive triple "for" loop 'Rcpp' implementation for the rest of the coding and testing of the SRSaRa.

Now that we have identified the fastest of our available methods to calculate U_k statistics, it is important to interrogate the effects of window size w and array length n on the computation time of these functions. In order to accomplish this, we again use 'microbenchmark' to evaluate the computation time of the U_k statistic calculation function across window sizes

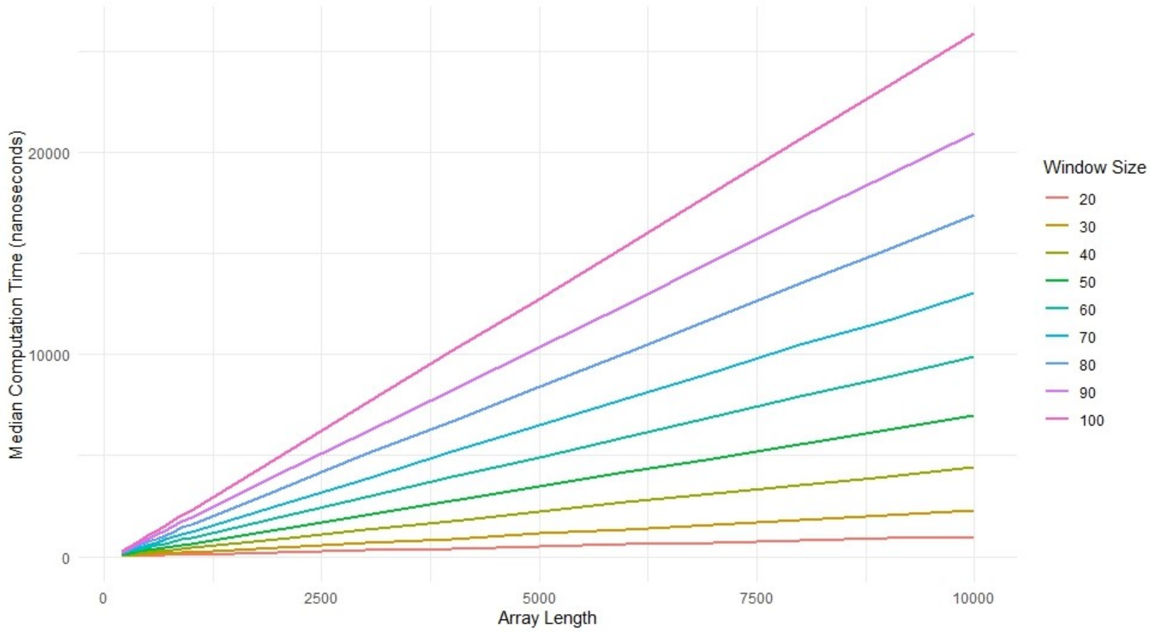

Figure 2.12: Graph of Computation Time against Array Length for Various Window Sizes

from $w = 20$ to $w = 100$ and across array lengths from $n = 200$ to $n = 10000$ over 100 replicates. The results of these tests are shown in Figure 2.12. From the graph, we see that for a given window size, the median computation time increases linearly with array length. As we would expect, there is also an increase in computation time for a given array length when the window size is increased.

2.8 Discussion

The SRSaRa's 'MAX' mode is a novel modification of Pettitt's test inspired by the SaRa that is capable of non-parametric single change-point detection. In this chapter, we have demonstrated the SRSaRa's ability to control Type-I error and visualized how it detects change-points. In Section 2.5, we demonstrated that the SRSaRa's detection ability outperforms that of Pettitt's test in every situation except for when the change-point occurs near the middle of the data array. Additionally, via its local information approach, the SRSaRa - using its "MAX" mode with global maximum based thresholding - is still capable of detection even when there are multiple change-points in the array. Through the graphs in this chapter, we aim to show that the SRSaRa for single change-point detection is visually intuitive - easily indicating the direction of change-points through the sign of the U_k statistic - and highly flexible through the choice of window size. Finally, we demonstrated the SRSaRa's ability to perform highly accurate single change-point detection on two real-world data sets in a non-parametric manner.

Chapter 3

Multiple Change-Point Detection

We now move on to the problem of multiple change-point detection. Once again, consider a sequence of random variables $\mathbf{X} = X_1, \ldots, X_n$. We are tasked with the problem of testing the hypotheses $H_0 :$ no change against $H_1 :$ at least one change exists in a non-parametric manner. To pose this problem in a more formally, we refer to Niu et al. (2016) who formulate the multiple change-point problem as follows. We wish to test $H_0(j) : j$ is not a change-point against $H_1(j) : j$ is a change-point where $j = 1, \ldots, n$. This tests a large number of hypotheses. Since the SRSaRa makes use of a sliding window approach akin to the SaRa, we may exploit this and reformulate the testing problem (as done in Hao et al. (2013)) into a series of sliding window hypotheses:

$$\begin{aligned} H_0(j) \quad &: \quad F_{j+1-h} = \cdots = F_{j+h} \text{ versus} \\ H_1(j) \quad &: \quad F_{j+1-h} = \cdots = F_j \neq F_{j+1} = \cdots = F_{j+h} \end{aligned}$$

where h is a fixed integer value under the assumption that two change-points are no less than h data points apart. In our case, h is half of the selected window size w. This hypothesis testing setup is well suited for the SRSaRa.

In this chapter we propose that the local maximum 'LM' mode of the SRSaRa is capable of testing these hypotheses while controlling FDR (False Discovery Rate) at the desired level. The procedure for the SRSaRa's 'LM' mode is outlined below:

1. For $\mathbf{X} = X_1, \ldots, X_n$, choose an even fixed window size $w < n$ and let $h = \frac{w}{2}$

2. Calculate U_k for $k = h, h+1, \ldots, n-h$ where

$$U_k = D(\mathbf{X}, k) = \sum_{i=k-h+1}^{k} \sum_{j=k+1}^{k+h} \operatorname{sgn}(X_i - X_j).$$

3. Using pre-calculated threshold ECDFs, determine the critical value U_{Thresh} which depends on
   - window size w
   - desired FDR control
4. Determine the local maxima of $|U_k|$, call these $U_{\text{LM},\tau}$
5. For each $U_{\text{LM},\tau}$, if $U_{\text{LM},\tau} > U_{\text{Thresh}}$, reject $H_0(\tau)$

With regards to the hypothesis testing problem posed above, since extreme U_k statistic values are more indicative of a possible change-point, we only explicitly test the locations in the array which have $|U_k|$ values that are local maxima. This allows us to test far fewer hypotheses than we would if we considered every possible $|U_k|$ calculated.

3.1 Local Max Thresholding

In section 2.2, we described the permutation thresholding process for single change-point detection. For detecting single change-points, we collect many absolute global maximum diagnostic function values and compute the threshold which properly controls Type-I error from there. When we wish to test for multiple change-points, we must modify this thresholding procedure. The local maximum thresholding procedure for determining the critical value U_{Thresh} necessary to control FDR while performing multiple change-point detection differs from the global maximum thresholding procedure by including a "local maximum" function. Instead of collecting the global maximum $|U'_k|$ values from the permuted data array, we first iterate along the permuted array with another sliding window which we call the neighborhood or 'span'. Due to the nature of our hypothesis testing setup, this 'span' is half the window size or $h = w/2$. Within this 'span', the "local maximum" function returns any $|U'_k|$

value greater than all other $|U'_k|$ values within the 'span'. If there is no diagnostic function value strictly greater than all others within this 'span', we do not return a local maximum value. This collection of strict local maximum $|U'_k|$ values gathered over many permutations of the original data array $\mathbf{X}$ is the ECDF of local maximum $|U'_k|$ values from which we may determine the threshold for testing if a given local maximum $|U_k|$ is a change-point. Ideally, this process will allow us to determine the threshold values which control FDR at the desired rate. In the following section, we explore the challenges encountered when using the process described above to calculate the multiple detection threshold values.

3.2 More on Local Maxima Thresholding & FDR Control

Determining the thresholds for a given window size that properly control FDR has been the crux of this project. After completing the single change-point detection procedure, we outlined a similar thresholding procedure for the multiple change-point detection problem that was based on the local maxima of $|U_k|$ statistics under H_0. Ideally, using thresholds computed from the ECDFs of local maximum $|U_k|$ statistics, we should be able to control FDR appropriately and detect multiple change-points in an array while doing so. After obtaining the thresholds and verifying there were no errors in the code, we used the thresholds in the SRSaRa and attempted to control FDR at levels 0.05, 0.10, and 0.15. Things did not go as well as hoped. The following paragraphs explore how things would have gone ideally, the problem encountered, and some ideas that may rectify the issue.

Using the procedure described in Section 3.1, many replicates of the thresholding process were performed to compute the distribution of absolute local maximums for various window sizes. From this, we obtain the distribution of local maxima of $|U_k|$, thus to obtain the thresholds necessary to control FDR at the aforementioned levels, we must compute the 0.975, 0.95, and 0.925 quantiles respectively for each window size. Doing this, we obtain the table of local maximum thresholds found in Table A.1 of Appendix A.

Histograms of the local max values for $w = 50, 100, 150, 200$ are shown in Figure 3.1. In all four of the graphs, we notice a long right tail. This is expected as under H_0 it is rare to observe extreme U_k statistic values. For example, with a window size of $w = 100$, the maximum possible absolute U_k value is $(100/2)^2 = 2500$. With $w = 100$, observing a $|U_k|$

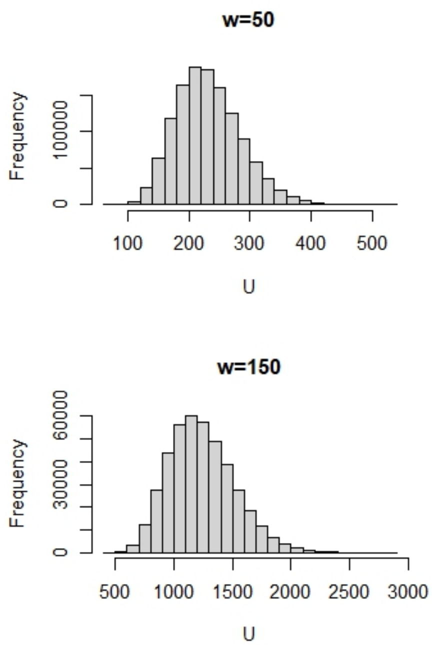

Figure 3.1: Local Max U ECDFs for Select Window Sizes

value of this magnitude would necessitate that all 50 values encompassed by one half of the window are strictly greater than or strictly less than the 50 values encompassed by the other half of the window. As one can imagine, under the null hypothesis, this is quite a rarity. In fact, over the 15000 replicates performed for thresholding on an array of length 10000, the largest local maximum absolute U_k statistic value observed was 1646, or $\approx 65\%$ of the maximum possible value.

Similar to how thresholds determined in this manner are able to control Type-I error for the SRSaRa's 'MAX' mode, we hoped that thresholds determined in this manner would also be able to control FDR at the desired levels listed above. Using the thresholds in Table A.1 and a FDR control experiment described later in 3.3, we found that the false discovery rates for these thresholds were a little less than double our target rates. Clearly the threshold values in Table A.1 are too small. Out of curiosity, we then repeated the FDR control experiment using thresholds calculated via the global maximum thresholding procedure. These global maximum threshold values resulted in FDR values well below our targets. If the values from our local maximum threshold procedure return FDRs that are too high, and the global maximum threshold values return FDRs that are too low, then

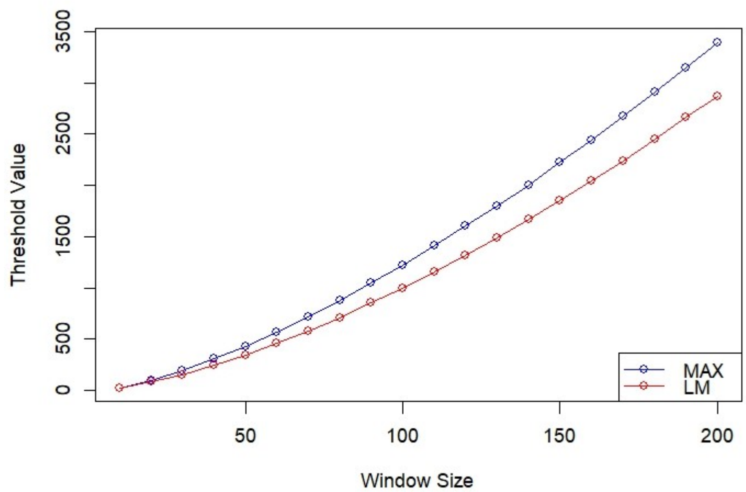

Figure 3.2: Plot of 'MAX' and 'LM' Threshold Values against Window Size

the values we desire must reside somewhere between the two. In Figure 3.2, we can see the global maximum threshold values plotted in blue and the local maximum threshold values plotted in red. The two threshold types follow a very similar pattern and the vertical distance between the two types of threshold values for a given window size widens as the window size increases. From this graph, it seems clear that the appropriate threshold values for multiple change-point detection that properly control the FDR should lie between the red and blue lines. As for why the local maximum threshold values we calculated return FDRs that are above our targets, we propose two possible explanations.

First, the local maximum function could be entirely too liberal in its determination of local maximums. This would lead to it returning a large number of small values, dragging down the threshold values. Second, it could be due to the rarity of extreme U_k statistic values detailed in a previous paragraph. The local maximum function could be operating perfectly fine. Perhaps extreme U_k statistic values are such a rarity that without performing an absurd number of replicates, we will not observe enough high values to determine an accurate threshold.

What can we do to fix this? For one, having a rough idea of the threshold values that yield desirable levels of FDR control for a few benchmark window sizes would allow us to know if we are moving in the right direction when trying to rectify our thresholding process. In terms of rectifying the thresholding process, we have two options: 1. create a more restrictive local

w	Threshold	Estimated FDR	Target FDR
50	380	0.0478	0.05
50	355	0.0952	0.10
50	340	0.144	0.15
100	1040	0.0484	0.05
100	950	0.0968	0.10
100	895	0.148	0.15
150	1875	0.0493	0.05
150	1685	0.0979	0.10
150	1575	0.147	0.15

Table 3.1: Thresholds and FDR for Three Window Sizes

maximum function that prioritizes higher U_k statistic values and 2. drastically increase the number of thesholding replicates for each window size.

3.3 False Discovery Rate Control

Let us now use the tried and true guess-and-check method to determine the threshold values that control FDR at levels 0.05, 0.10, and 0.15 for window sizes $w = 50, 100, 150$. Using the threshold values found in A.1 as a starting point, the threshold values were steadily increased until the false discovery rate experiment returned FDRs that were below the target values on average over 1000 replicates. The FDR experiment generates 50 location change-points at random along a data array of length 30000, then adds $\mathcal{N}(0, 1)$ noise to the array. The change-points all have a location change of magnitude 1.5. The experiment counts a true detection when the SRSaRa detects a change-point within 10 entries of the location of a true change-point. The estimated thresholds for $w = 50, 100, 150$ and their corresponding estimated FDRs are shown in Tab. 3.1. The FDR control test was run again with 50 change-points, each having a location change of magnitude 3, and produced lower FDR values than those in Tab. 3.1.

Since the SRSaRa is a non-parametric test, it is important to verify that we can control FDR when the noise distribution is not standard normal. We now perform the same FDR control experiment, but this time with a t_3 noise distribution. For the first trial, the change-point will have a location shift of 1.5 from the background noise. For the second trial, the change-points will have a location shift of 3 compared to the background. Since t_3 is a highly

w	Threshold	FDR Delta=1.5	FDR Delta=3	Target FDR
50	380	0.0762	0.0331	0.05
50	355	0.132	0.0761	0.10
50	340	0.186	0.12	0.15
100	1040	0.0923	0.0355	0.05
100	950	0.132	0.0755	0.10
100	895	0.185	0.13	0.15
150	1875	0.0899	0.0332	0.05
150	1685	0.14	0.0846	0.10
150	1575	0.176	0.121	0.15

Table 3.2: FDR for t_3 Noise Change-Point Experiment

tailed distribution, properly controlling FDR when the change-points have a location shift of just 1.5 is expected to be difficult. Using the same threshold values as shown in Tab. 3.1, the FDR for the first and second trials with t_3 noise are shown in Table 3.2.

As expected, controlling FDR when the change-points have a location shift of just 1.5 above the background t_3 noise is quite difficult. This is likely because a location shift of 1.5 is quite small compared to the normal variation in the highly-tailed t_3 distribution, making the exact location of the change-point difficult to pinpoint. In this simulation, any detection more than 11 data points away from a true change-point location is considered a false detection. We are likely detecting change-points close to their true location in this simulation, but because of how "blended" the data appears around the true location of change, the SRSaRa likely struggles to detect them within the 11 data point margin. When we increase the magnitude of the location change to 3 above the mean of the t_3 noise, this is no longer an issue. As seen in Tab. 3.2, the false discovery rates of the SRSaRa when change-points have a location shift of magnitude 3 are well below the target.

Overall, while the calculation of thresholds for multiple change-point detection with the SRSaRa and the local maxima detection function that the 'LM' mode relies upon need additional work, this section aims to show that controlling FDR with the SRSaRa is possible. We believe that the results of this section demonstrate that, with the right changes to the thresholding procedure and the local maxima function, the SRSaRa is a promising candidate for non-parametric multiple change-point detection.

3.4 Comparison with SaRa

Since the SRSaRa is a combination of a SaRa-like procedure with a diagnostic function based on Pettitt's test and we have already compared the SRSaRa's 'MAX' mode with Pettitt's test, it seems only fair to now compare the SRSaRa and the original SaRa. Whereas the SaRa is a parametric multiple change-point test that works particularly well with Gaussian data, the SRSaRa is a non-parametric multiple change-point test, so the comparison here may not be as direct as the comparison between the SRSaRa's 'MAX' mode and Pettitt's test. Regardless, this section aims to highlight the comparative advantages and disadvantages of the SaRa and the SRSaRa.

For our first point of comparison, we return to the FDR control experiment from the previous section. The code for the FDR experiment was originally written for the SaRa, so as one may expect, the SaRa performs extremely in terms of both FDR control and total true positives detected. On average, with window size 100, SRSaRa achieves 43 of 50 possible true positives per trial while maintaining FDR control. In contrast, the SaRa achieves 47 of 50 possible true positives per trial while maintaining lower false discovery rates than the SRSaRa. The SaRa is tailor made for detecting change-points on Gaussian data.

As their names suggest, the SRSaRa and the SaRa involve similar processes - both making use of a local information approach - with the main differences between the two arising from the choice in diagnostic function. As mentioned, the SaRa uses a diagnostic function based on the difference in means within a small neighborhood on the left and right of the point of interest x. The SRSaRa on the other hand makes use of a non-parametric diagnostic function based on the sum of signed differences. From these diagnostic functions, we may draw out the main strengths and weaknesses of the SRSaRa and the SaRa. Since the SaRa relies on the difference between means to detect change-points, the SaRa excels on data where the mean is a useful statistic. Because the mean is a sufficient statistic, the SaRa makes use of all the information within the neighborhood on either side of the point of interest. This allows the SaRa to use relatively smaller window sizes than the SRSaRa, enabling the SaRa to detect shorter change-points than the SRSaRa. When faced with data containing large outliers, the use of mean difference as a diagnostic can become a weakness. Since the window sizes for the SaRa are relatively small, usually between 5-15 data points on either side of the point of interest x, outliers can cause the mean on one side of x to become inflated. In contrast, the sum of sign differences used by the SRSaRa performs a high degree of data reduction and as a consequence, the SRSaRa must use a larger window size to detect change-points. This high degree of data reduction and larger window size can also be a strength. Since the SRSaRa relies on the sum of sign differences across the point of interest, an outlier (even if it is greater than all other points in the window) will at most raise the value of the diagnostic statistic by half the window size. This makes the SRSaRa robust to outliers in the data.

Lastly, credit must be given to the theoretical guarantees of the SaRa. Research from Niu and Zhang (2012) and Hao et al. (2013), among others, has proved numerous favorable properties of the SaRa. More recently, in the work of Xiao et al. (2019), the SaRa has been modified to make use of information specific to Copy Number Variation (CNV) data, to improve its CNV detection ability. While the simulation results shown in this paper are promising for the SRSaRa, when theoretical guarantees are a must or when analyzing CNV data, the SaRa is far superior.

In summary, the main differences between the SRSaRa and the SaRa may be found in their respective diagnostic functions. In situations where the mean is a meaningful statistic or change-points are expected to be relatively short, the SaRa excels. In situations where the data is from an unknown (possibly non-Gaussian) distribution or contains many large outliers, the SRSaRa's robust diagnostic function finds its strength.

Chapter 4

Conclusion

The Signed-Rank Screening and Ranking Algorithm (SRSaRa) is a novel non-parametric change-point detection technique utilizing a SaRa-like process (Niu and Zhang, 2012) with a diagnostic function based on Pettitt's test (Pettitt, 1979). The SRSaRa has a 'MAX' and 'LM' mode used for single and multiple change-point detection respectively. In Chapter 2, we demonstrate the ability of the SRSaRa's 'MAX' mode to control Type-I error at the desired level (Section 2.3) while detecting change-points in a variety of distributions and in real-world data (Section 2.6). Additionally, we illustrate several situations where the SRSaRa's 'MAX' mode is superior to Pettitt's test (Section 2.5).

In Chapter 3, we outline the multiple change-point detection problem, describe the process and difficulties in calculating thresholds for the local maxima mode of the SRSaRa (Section 3.2), present ideas that may fix the thresholding process, estimate the values of the thresholds which give the desired FDR control (Section 3.3), and compare the SRSaRa to its predecessor the SaRa (Section 3.4). Overall, while the threshold calculation process and local maxima detection algorithm need additional effort to make them fully functional, we believe this chapter presents promising numerical results that demonstrate the SRSaRa's potential for non-parametric multiple change-point detection. Using the thresholds found via the guess-and-check method, we demonstrate that the SRSaRa is capable of controlling FDR at the desired level in its current state. With additional research and changes to the local maxima detection function, we believe the SRSaRa's 'LM' mode can become fully function and on par with its 'MAX' mode.

Overall, we believe the results in this paper demonstrate the SRSaRa's promising ability to perform both multiple and single change-point detection across a variety of distributions and data sets. The performance and quality of results from the SRSaRa's 'MAX' mode underscore its potential as a non-parametric change-point detection technique. However, further refinement of the thresholding process and an local maxima detection is necessary for the SRSaRa's 'LM' mode to fully realize its capabilities. Since this paper mainly presents numerical results and simulation studies, future research should focus on solidifying the theory behind the techniques used in the 'MAX' and 'LM' modes. With further research, it may be possible to determine the threshold values for single detection knowing only the array length and window size, likewise it may be possible to determine the threshold for multiple detection knowing only the window size. Findings based in theory may also aid in fixing the issues currently present in the SRSaRa's 'LM' mode.

Appendix A

Tables

w	$\alpha = 0.05$	$\alpha = 0.10$	$\alpha = 0.15$
10	25	25	25
20	84	78	76
30	157	149	143
40	246	232	222
50	347	325	313
60	458	430	412
70	581	545	523
80	712	670	642
90	853	801	767
100	1002	940	900
110	1159	1087	1043
120	1322	1240	1190
130	1491	1399	1341
140	1670	1568	1504
150	1857	1739	1667
160	2048	1922	1842
170	2243	2105	2019
180	2452	2298	2200
190	2661	2495	2389
200	2870	2692	2578

Table A.1: Initial Local Max Threshold Values for $w = 10, 20, \ldots, 200$

References

E.-E. A. Aly and S. S. BuHamra. Rank tests for two change points. *Computational Statistics & Data Analysis*, 22(4):363–372, 1996. ISSN 0167-9473. doi: https://doi.org/10.1016/0167-9473(95)00060-7. URL https://www.sciencedirect.com/science/article/pii/0167947395000607.

G. K. Bhattacharya and R. A. Johnson. Nonparametric tests for shift at an unknown time point. *The Annals of Mathematical Statistics*, 39(5):1731–1743, 1968. ISSN 00034851, 21688990. URL http://www.jstor.org/stable/2239432.

P. K. Bhattacharya. Some aspects of change-point analysis. *Lecture Notes-Monograph Series*, 23:28–56, 1994. ISSN 07492170. URL http://www.jstor.org/stable/4355761.

N. Hao, Y. S. Niu, and H. Zhang. Multiple change-point detection via a screening and ranking algorithm. *Statistica Sinica*, 23:1553–1572, 2013.

R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, Oct. 2012. ISSN 1537-274X. doi: 10.1080/01621459.2012.737745. URL http://dx.doi.org/10.1080/01621459.2012.737745.

D. S. Matteson and N. A. James. A nonparametric approach for multiple change point analysis of multivariate data. 2013. URL https://arxiv.org/abs/1306.4933.

Y. S. Niu and H. Zhang. The screening and ranking algorithm to detect dna copy number variations. *The Annals of Applied Statistics*, 6(3):1306–1326, 2012. doi: 10.1214/12-AOAS539SUPP. URL https://doi.org/10.1214/12-AOAS539SUPP.

Y. S. Niu, N. Hao, and H. Zhang. Multiple change-point detection: a selective overview. 2016. URL https://arxiv.org/abs/1512.04093.

A. B. Olshen, E. S. Venkatraman, R. Lucito, and M. Wigler. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4):557–572, 10 2004. ISSN 1465-4644. doi: 10.1093/biostatistics/kxh008. URL https://doi.org/10.1093/biostatistics/kxh008.

E. S. Page. CONTINUOUS INSPECTION SCHEMES. *Biometrika*, 41(1-2):100–115, 06 1954. ISSN 0006-3444. doi: 10.1093/biomet/41.1-2.100. URL https://doi.org/10.1093/biomet/41.1-2.100.

A. N. Pettitt. A non-parametric approach to the change-point problem. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(2):126–135, 1979. URL https://www.jstor.org/stable/2346729.

A. Sen and M. S. Srivastava. On Tests for Detecting Change in Mean. *The Annals of Statistics*, 3(1):98 – 108, 1975. doi: 10.1214/aos/1176343001. URL https://doi.org/10.1214/aos/1176343001.

Y. Wang, Z. Wang, and X. Zi. Rank-based multiple change-point detection. *Communications in Statistics-Theory and Methods*, 49(14):3438–3454, 2020.

F. Xiao, X. Luo, N. Hao, Y. Niu, X. Xiao, G. Cai, C. I. Amos, and H. Zhang. An accurate and powerful method for copy number variation detection. *Bioinformatics (Oxford, England)*, 35(17):2891–2898, 2019. doi: 10.1093/bioinformatics/bty1041.