Estimating Population Range by Recurring Online Chunk Bootstrap with Non-cumulative Data in Streaming Data Environment

***Abstract*—**

***Index Terms*—Article submission, IEEE, IEEEtran, journal, LaTeX, paper, template, typesetting.**

I. Introduction

The bootstrap method is one of the powerful and widely statistical methods for estimating the uncertainty from finite samples of any parameter of interest, for example, standard error, confidence interval, accuracy, etc. Efron [2] introduced bootstrap methods, which generalize the jackknife method. In this work, the jackknife was mathematically expressed as the linear approximation for the bootstrap. The empirical results demonstrated the proposed methods capable of estimating the standard error of complex estimators. Then, Efron and Tibshirani [3] overview the basic concepts and applications of the bootstrap methods for estimating standard errors, confidence intervals, and other measures of statistical accuracy. For standard error estimation, the original data for one population will be resampled with replacement to create many bootstrap samples, and the statistics with their standard deviations for each bootstrap sample will be calculated. The results showed that several examples provided reasonably accurate and efficient between the bootstrap estimations and theoretical density curves. For confident intervals, several methods were empirically investigated for constructing bootstrap confidence intervals, which provide more accurate intervals than standard methods in cases where the statistic distribution is non-normal. A sufficient number of bootstrap replications were given to obtain accurate results. Carpenter and Bithell [4] presented a practical guide for bootstrap confidence intervals in healthcare data, addressing three key questions: when/ which/ and how to apply or implement bootstrap methods. Various bootstrap methods were evaluated for confidence intervals from three families: pivotal, non-pivotal, and test-inversion. The experimental results concluded that when the assumptions of the underlying distribution do not hold (like asymptotic normality), the bootstrap confidence intervals are the alternative approach. , especially with small sample sizes or complex data structures.

Range approximation in one-dimensional data plays an important role in statistics, data analysis, and various computational sciences. The objective of range approximation in 1-D data is to estimate the interval between a dataset's minimum and maximum values.

II. Studied Problem and Objectives

Let a chunk of integer data, a set of bins, and a set of integer data be defined as follows:

1) A set of integer data, $\mathbf{D} = \{a \leq d_j \leq b \mid 1 \leq j \leq p\}$, for integer constants p, a, and b.
2) An integer data chunk $\mathbf{C}$ divided into n smaller chunks, $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_i, \ldots, \mathbf{c}_n \mid 1 \leq n < \infty\}$. Each chunk has integers randomly taken from $\mathbf{D}$.
3) A bin $\mathbf{B}$ for containing incoming all integers in each chunk. There are two attributes, $v_i^{(min)}$ and $v_i^{(max)}$, defining the interval of integer values such that any integer $v_i^{(min)} \leq d_j \leq v_i^{(max)}$ can be assigned to this bin.

Each chunk $\mathbf{c}_i$ sequentially flows into the bin. If some integers whose values are less than $v_1^{(min)}$ or larger than $v_m^{(max)}$, then the width of $\mathbf{B}$ must be expanded.

In the beginning, the size of bin $\mathbf{B}$ is made large enough to contain all integers in the first incoming chunk $\mathbf{c}_1$. Then size of $\mathbf{B}$ is occasionally expanded so that all integers in the other next incoming chunks can be assigned to the intervals of $\mathbf{B}$. Bin $\mathbf{B}$ is expanded if the values of some integers in some incoming chunks are either less than $v^{(min)}$ or larger than $v^{(max)}$. Therefore, the studied problem is defined as follows.

Let $min(\mathbf{D})$ and $max(\mathbf{D})$ be the minimum and maximum values of $\mathbf{D}$. After capturing of integers in the first chunk, how to achieve the minimum number of expansions of $\mathbf{B}$ so that

1) All incoming integers in the next other chunks can be assigned to $\mathbf{B}$.
2) $(min(\mathbf{D}) - v_1^{(min)}) \geq 0$ is minimum.
3) $(v_i^{(max)} - max(\mathbf{D})) \geq 0$ is minimum.

Figure 1 illustrates an example scenario of capturing chunks. There are 3 sequentially incoming chunks containing these integers: $\mathbf{c}_1 = \{7, 9, 23, 10\}$, $\mathbf{c}_2 = \{2, 11, 1, 8\}$, $\mathbf{c}_3 = \{25, 14, 6, 13\}$. After capturing $\mathbf{c}_1$, the values of left and right ends of bin $\mathbf{B}$ are set to $v^{(min)} = 7$ and $v^{(max)} = 23$ and all data are discarded. Then, both ends are expanded in advance to $v^{(min)} = 4$ and $v^{(max)} = 25$, preparing for $\mathbf{c}_2$. When $\mathbf{c}_2$ enters, all integers except 2 can be captured because the left end value $v^{(min)} = 4$ is larger than 2. Thus, the left end $v^{(min)}$ is expanded to $v^{(min)} = 2$. No need to expand the right end. All data in $\mathbf{c}_2$ are discarded. Then, get $\mathbf{c}_3$. The interval of $\mathbf{B}$

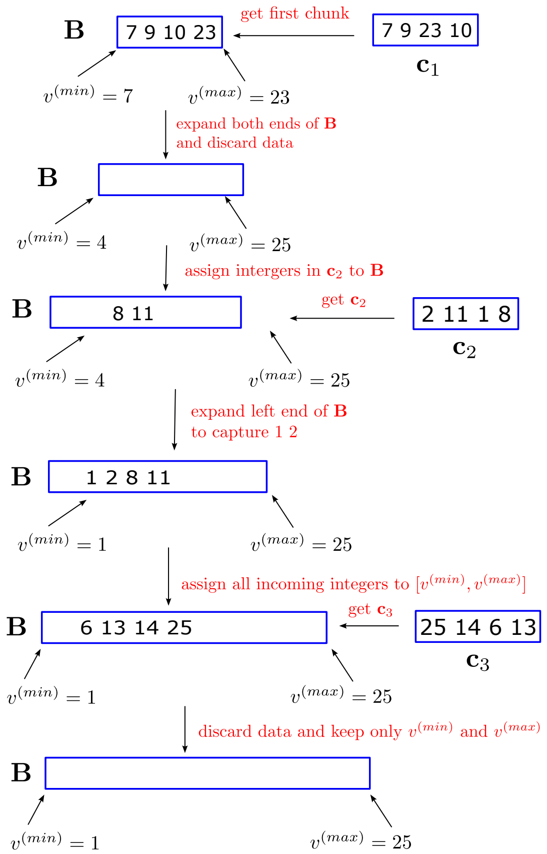

Fig. 1. An example of studied problem.

is large enough to capture all integers of $\mathbf{c}_3$. After capturing $\mathbf{c}_3$, all integers are discarded. In this example, the number of expansions is 2, after chunks 1 and 2. Generally, how to achieve the minimum number of expansions of $\mathbf{B}$ in advance so that both expanded values are also minimum.

A. Constraints

The amount of integers in each $\mathbf{c}_i$ is denoted by $|\mathbf{c}_i|$. Bin B can be considered as an interval having left end denoted by L and right end denoted by R. The following constraints are imposed.

1) The probability of distribution of each integer d_j in $\mathbf{c}_i$ is unknown.
2) $|\mathbf{c}_i| \neq |\mathbf{c}_{i+1}|$ or $|\mathbf{c}_i| = |\mathbf{c}_{i+1}|$.
3) After assigning all integers of $\mathbf{c}_i$ inside the bin, chunk $\mathbf{c}_i$ is completely discarded and never reentered the bin assignment.

III. Concept of Recurring Online Bootstrap in Streaming Environment with Unrecorded Data

A. Algorithm

explain standard histogram first

The algorithm has two phases. The first phase

Algorithm 1: Capturing data chunks and expanding $\mathbf{B}$ when it is necessary.

Input: Stream of data chunks, $\mathbf{c}_i$ for $1 \leq i < \infty$.
Output:

Phase 1: Capturing $\mathbf{c}_1$ and get initial expansion of $\mathbf{B}$.

1. Get $\mathbf{c}_1$ and set $total\_data = |\mathbf{c}_1|$.
2. Let $v^{(min)} = \min(\mathbf{c}_1)$ and $v^{(max)} = \max(\mathbf{c}_1)$.
3. Divide $\mathbf{B}$ into 8 equal sub-intervals $\mathbf{b}_i$ for $1 \leq i \leq 8$, each of size $(v^{(max)} - v^{(min)})/8$.
4. Put the integers in $\mathbf{c}_1$ whose values are within sub-intervals $\mathbf{b}_1$ and $\mathbf{b}_8$ into these two sub-intervals.
5. Count the number of integers in $\mathbf{b}_1$ and $\mathbf{b}_8$ and let $|\mathbf{b}_1|$ and $|\mathbf{b}_8|$ denote these numbers.
6. Let $avg = (v^{(max)} + v^{(min)})/2$ be the middle value of $\mathbf{B}$.
7. Find the types of probability distribution in list P best fitting the data in $\mathbf{b}_1$, and $\mathbf{b}_8$ by using Algorithm 2.1 with $total\_data$, $\mathbf{b}_1$, and $\mathbf{b}_8$.
8. Compute the standard number of integers in $\mathbf{b}_1$ denoted by $lstd$, and in $\mathbf{b}_8$ denoted by $rstd$. from the best fitted probability distribution by using Algorithm 2.2.
9. **While** $|\mathbf{b}_1| > lstd$ or $|\mathbf{b}_8| > rstd$ **do**
10. **If** $|\mathbf{b}_1| > lstd$ **then**
11.   Expand $v^{(min)}$ by using Algorithm 3 with all integers in $\mathbf{b}_1$.
12. **EndIf**
13. **If** $|\mathbf{b}_8| > rstd$ **then**
14.   Expand $v^{(max)}$ by using Algorithm 4 with all integers in $\mathbf{b}_8$.
15. **EndIf**
16. Adjust the width of $\mathbf{b}_1$ and $\mathbf{b}_8$ by dividing $\mathbf{B}$ into 8 equal sub-intervals $\mathbf{b}_i$ for $1 \leq i \leq 8$, each of size $(v^{(max)} - v^{(min)})/8$.
17. **EndWhile**
18. Discard $\mathbf{c}_1$ and all integers in $\mathbf{b}_1$ and $\mathbf{b}_8$.

Phase 2: Capturing other $\mathbf{c}_i$ and determining the necessity of expanding $\mathbf{B}$.

1. **while** there exists a new incoming chunk $\mathbf{c}_i$ **do**
2. $total\_data = total\_data + |\mathbf{c}_i|$.
x. **If** $|\mathbf{b}_1| \geq min_B$ **then**
x. $\mathbf{B}_1 = \{\min(\mathbf{c}_i)\} \cup \mathbf{B}_1$.
x.   Apply **Alg.** 3 with $\mathbf{B}_1$ to get $v_B^{(min)}$.
x.   **If** $v^{(min)} > v_B^{(min)}$ **then**
x. $v^{(min)} = v_B^{(min)}$.
x.   **EndIf**
x. **Else**
x. $v^{(min)} = \{\min(\mathbf{c}_i)\}$.
x. **EndIf**
xx. **If** $|\mathbf{b}_8| \geq min_B$ **then**
x. $\mathbf{B}_8 = \{\max(\mathbf{c}_i)\} \cup \mathbf{B}_8$.
x.   Apply **Alg.** 4 with $\mathbf{B}_8$ to get $v_B^{(max)}$.
x.   **If** $v^{(max)} < v_B^{(max)}$ **then**
x. $v^{(max)} = v_B^{(max)}$.
x.   **EndIf**

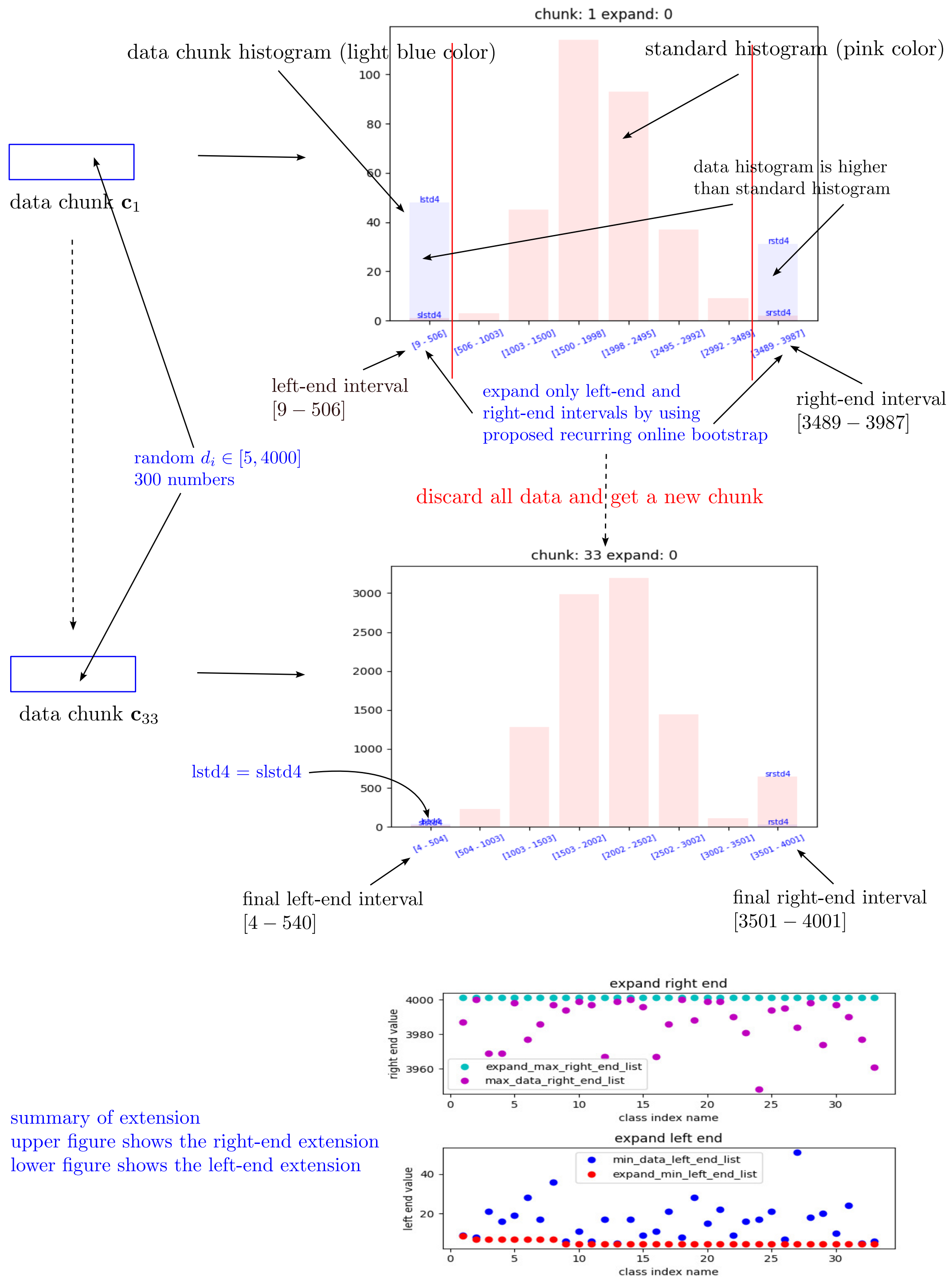

Fig. 2. Framework.

x. **Else**
x. $v^{(max)} = \{\max(\mathbf{c}_i)\}$.
x. **EndIf**
9. Divide $\mathbf{B}$ into 8 equal sub-intervals $\mathbf{b}_i$ for $1 \le i \le 8$, each of size $(v^{(max)} - v^{(min)})/8$.
10. Put the integers in $\mathbf{c}_i$, whose values are within sub-intervals $\mathbf{b}_1$ and $\mathbf{b}_8$, into these two sub-intervals.
11. Count the number of elements in $\mathbf{b}_1$ and $\mathbf{b}_8$ and let $|\mathbf{b}_1|$ and $|\mathbf{b}_8|$ denote these numbers.
12. Find the types of probability distribution in list P best fitting the data in $\mathbf{b}_1$, and $\mathbf{b}_8$ by using Algorithm 2.1 with $total\_data$, $\mathbf{b}_1$, and $\mathbf{b}_8$.
13. Compute the standard number of elements in $\mathbf{b}_1$ denoted by $lstd$, and in $\mathbf{b}_8$ denoted by $rstd$ from the best fitted probability distribution by

using Algorithm 2.2.
14. **While** $|\mathbf{b}_1| > lstd$ or $|\mathbf{b}_8| > rstd$ **do**
x. $v^{(min)}_{old} = v^{(min)}$ and $v^{(max)}_{old} = v^{(max)}$
15. **If** $|\mathbf{b}_1| > lstd$ **then**
16. Expand $v^{(min)}$ by using Algorithm 3 with all integers in $\mathbf{b}_1$.
17. **EndIf**
18. **If** $|\mathbf{b}_8| > rstd$ **then**
19. Expand $v^{(max)}$ by using Algorithm 4 with all integers in $\mathbf{b}_8$.
20. **EndIf**
x. **If** v^{min} and v^{max} do not change.
x. Go to Line XX.
e. **EndIf**
21. Adjust the width of $\mathbf{b}_1$ and $\mathbf{b}_8$ by dividing $\mathbf{B}$ into 8 equal sub-intervals $\mathbf{b}_i$ for $1 \le i \le 8$, each of size $(v^{(max)} - v^{(min)})/8$.
22. **EndWhile**.
23. Discard $\mathbf{c}_i$ and all integers in $\mathbf{b}_1$ and $\mathbf{b}_8$.
24. **EndWhile**.

Algorithm 2.1: Finding the types of probability distribution in list P best fitting the data in $\mathbf{b}_1$, and $\mathbf{b}_8$.
Input: (1) A list of standard probability distribution P. (2) $total\_data$. (3) $\mathbf{b}_1$. (4) $\mathbf{b}_8$.
Output: $lname$ and $rname$.

1. **For** each type probability distribution $p \in P$ **do**
2. Divide the area under standard probability distribution p into 8 stripes of equal width.
3. Let $l_4^{(p)}$ be the percentage of data in the 4^{th} stripe to the left of mean.
4. Let $r_4^{(p)}$ be the percentage of data in the 4^{th} stripe to the right of mean.
5. Compute the difference between standard number of integers in the 4^{th} left stripe and $|\mathbf{b}_1|$: $ld^{(p)} = abs(l_4^{(p)} * total\_data - |\mathbf{b}_1|)$.
6. Compute the difference between standard number of integers in the 4^{th} right stripe and $|\mathbf{b}_8|$: $rd^{(p)} = abs(r_4^{(p)} * total\_data - |\mathbf{b}_8|)$.
7. **EndFor**
8. Find $lname = \arg\min_{p \in P}(ld^{(p)})$.
9. Find $rname = \arg\max_{p \in P}(rd^{(p)})$.
10. **Return** $lname$ and $rname$.

Algorithm 2.2: Computing $lstd$ and $rstd$.

Input: (1) A list of standard probability distribution P. (2) $total\_data$. (3) $lname$. (4) $rname$. (5) $l_4^{(p)}$ and $r_4^{(p)}$; $\forall p \in P$ from Algorithm 2.1.
Output: $lstd$ and $rstd$.

1. **If** $lname$ is the same as $rname$ **then**
2. Set $lstd = l_4^{(lname)} * total\_data$.
3. Set $rstd = r_4^{(rname)} * total\_data$.
4. **EndIf**
5. **If** $lname$ is different from $rname$ **then**
6. Set $lstd = \max_{p \in P}(l_4^{(p)}) * total\_data$.
7. Set $rstd = \max_{p \in P}(r_4^{(p)}) * total\_data$.
8. **EndIf**
9. **Return** $lstd$ and $rstd$.

Algorithm 3: Recurring online chunk bootstrap for $\mathbf{B}_1$.

Input: (1) Present set of incoming integers in $\mathbf{B}_1$; (2) Number of bootstrap iterations N; (3) $mean(\mathbf{a})$ is a function computing the mean of set $\mathbf{a}$; (4) $std(\mathbf{a})$ is a function computing the standard deviation of set $\mathbf{a}$; (5) $abs(x)$ is the absolute value of constant x.
Output: $v^{(min)}$.

1. Let $\mathbf{S} = \emptyset$ be a set of bootstrapped samples.
2. Let $\mathbf{M} = \emptyset$ be a set of mean of each bootstrapped sample.
x. Let $\mathbf{Max} = \emptyset$ be a set of maximum values of each bootstrapped samples.
x. Let $\mathbf{Min} = \emptyset$ be a set of minimum values of each bootstrapped samples.
3. Let $\mathbf{P} = \emptyset$ be a set of standard deviation of each bootstrapped sample.
4. $prev\_mean = 0$.
5. **For** $1 \le i \le N$ **do**:
6. Let $\mathbf{s}_i$ be a set of randomly sampled integers of size $|\mathbf{B}_1|$ from $\mathbf{B}_1$ with replacement.
7. $\mathbf{S} = \mathbf{S} \cup \{\mathbf{s}_i\}$.
8. $present\_mean = (mean(\mathbf{s}_i) + prev\_mean)/2$.
9. $prev\_mean = present\_mean$.
10. $\mathbf{M} = \mathbf{M} \cup \{present\_mean\}$.
x. $\mathbf{Min} = \mathbf{Min} \cup \{min(\mathbf{s}_i)\}$.
11. **EndFor**.
12. $\mu^{(boot)} = mean(\mathbf{M})$.
13. **For** each $\mathbf{s}_i \in \mathbf{S}$ **do**
14. $\mathbf{P} = \mathbf{P} \cup \{std(\mathbf{s}_i)\}$.
15. **EndFor**
16. $\sigma^{(boot)} = mean(\mathbf{P})$.
17. $\mu^{(diff)} = abs(mean(\mathbf{B}_1) - \mu^{(boot)})$.
18. $\sigma^{(diff)} = abs(std(\mathbf{B}_1) - \sigma^{(boot)})$.
x. **If** $MinmaxBoost$
x. $min_{left} = meanProbBased(\mathbf{Min})$.
x. **Else**
x. $min_{left} = min(\mathbf{B}_1)$.
19. **If** $mu^{(boot)} < mean(\mathbf{B}_1)$ **do**
x. $v^{(min)} = min_{left} - \mu^{(diff)}$.
21. **If** $mean(\mathbf{B}_1) < mu^{(boot)}$ **do**
x. $v^{(min)} = min_{left} - \sigma^{(diff)}$.

Algorithm 4: Recurring online chunk bootstrap for $\mathbf{B}_8$.

Input: (1) Present set of incoming integers in $\mathbf{B}_1$; (2) Number of bootstrap iterations N; (3) $mean(\mathbf{a})$ is a function computing the mean of set $\mathbf{a}$; (4) $std(\mathbf{a})$ is a function computing the standard deviation of set $\mathbf{a}$; (5) $abs(x)$ is the absolute value of constant x.
Output: $v^{(max)}$.

1. Let $\mathbf{S} = \emptyset$ be a set of bootstrapped samples.
2. Let $\mathbf{M} = \emptyset$ be a set of mean of each bootstrapped sample.
3. Let $\mathbf{P} = \emptyset$ be a set of standard deviation of each bootstrapped sample.
4. $prev\_mean = 0$.
5. **For** $1 \leq i \leq N$ **do**:
6.   Let $\mathbf{s}_i$ be a set of randomly sampled integers of size $|\mathbf{B}_8|$ from $\mathbf{B}_8$ with replacement.
7. $\mathbf{S} = \mathbf{S} \cup \{\mathbf{s}_i\}$.
8. $present\_mean = (mean(\mathbf{s}_i) + prev\_mean)/2$.
9. $prev\_mean = present\_mean$.
10. $\mathbf{M} = \mathbf{M} \cup \{present\_mean\}$.

x. $\mathbf{Max} = \mathbf{Max} \cup \{max(\mathbf{s}_i)\}$.

11. **EndFor**.
12. $\mu^{(boot)} = mean(\mathbf{M})$.
13. **For** each $\mathbf{s}_i \in \mathbf{S}$ **do**
14. $\mathbf{P} = \mathbf{P} \cup \{std(\mathbf{s}_i)\}$.
15. **EndFor**
16. $\sigma^{(boot)} = mean(\mathbf{P})$.
17. $\mu^{(diff)} = abs(mean(\mathbf{B}_8) - \mu^{(boot)})$.
18. $\sigma^{(diff)} = abs(std(\mathbf{B}_8) - \sigma^{(boot)})$.

x. **If** $MinmaxBoost$

x. $max_{right} = meanProbBased(\mathbf{Max})$.

x. **Else**

x. $max_{right} = max(\mathbf{B}_8)$.

19. **If** $mu^{(boot)} < mean(\mathbf{B}_8)$ **do**

x. $v^{(max)} = max_{right} + \sigma^{(diff)}$.

21. **If** $mean(\mathbf{B}_1) < mu^{(boot)}$ **do**

x. $v^{(max)} = max_{right} + \mu^{(diff)}$.

IV. Experimental Results

Two types of population data are considered for performance evaluation: simulated and real-world data. Five statistical distributions with different settings of relevant parameters were applied to simulate the population datasets. The descriptive statistics of the simulation were shown in

TABLE I
Range estimation of Wald distribution with $(1, 0.5)$ and $10,000$ units of population.

method	chunk size	e_l	e_r	e_s	n_l	n_r
Online BT	50	-0.0	0.09	0.1	152.33	0
Online BT	100	-0.0	0.09	0.1	301.67	0
Online BT	500	-0.0	0.09	0.1	645.0	0
Min-max online BT	50	-0.02	0.09	0.11	374.41	0
Min-max online BT	100	-0.01	0.09	0.11	426.0	0
Min-max online BT	500	-0.01	0.09	0.1	644.5	0
Online BT 1 ch	3000	-0.0	0.09	0.09	2600.0	0
Traditional BT	3000	-0.01	0.58	0.59	0.0	0

TABLE II
Range estimation of Wald distribution with $(1, 2)$ and $10,000$ units of population.

method	chunk size	e_l	e_r	e_s	n_l	n_r
Online BT	50	-0.0	0.09	0.1	152.33	0
Online BT	100	-0.0	0.09	0.1	301.67	0
Online BT	500	-0.0	0.09	0.1	645.0	0
Min-max online BT	50	-0.02	0.09	0.11	374.41	0
Min-max online BT	100	-0.01	0.09	0.11	426.0	0
Min-max online BT	500	-0.01	0.09	0.1	644.5	0
Online BT 1 ch	3000	-0.0	0.09	0.09	2600.0	0
Traditional BT	3000	-0.01	0.58	0.59	0.0	0

TABLE III
Range estimation of Wald distribution with $10,000$ units of population.

method	chunk size	e_l	e_r	e_s	n_l	n_r
Online BT	50	-0.02	6.36	6.38	118.31	1.57
Online BT	100	-0.02	6.36	6.38	231.36	1.4
Online BT	500	-0.02	6.36	6.38	609.0	4.0
Min-max online BT	50	-0.04	6.36	6.4	165.67	1.33
Min-max online BT	100	-0.04	6.96	7.0	248.58	1.0
Min-max online BT	500	-0.04	6.97	7.0	604.8	3.0
Online BT 1 ch	3000	-0.02	6.36	6.38	1761.0	0.0
Traditional BT	3000	-0.02	6.95	6.97	0.0	0.0

References

[1] Hong Yu, Zhanguo Liu, Guoyin Wang, “An automatic method to determine the number of clusters using decision-theoretic rough set”, International Journal of Approximate Reasoning 55 (2014) 101–115.

[2] B., Efron, “BOOTSTRAP METHODS: ANOTHER LOOK AT THE JACKKNIFE”, The Annals of Statistics, 7(1), pp. 101–115, 1976.

[3] B., Efron, R., Tibshirani, 1986, ”Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy”, Statistical Science, 1(1), 54–75. doi:10.1214/ss/1177013815 10.1214/ss/1177013815

[4] J., Carpenter, J., Bithell, 2000, ”Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. Statistics in Medicine”, 19(9), 1141–1164. doi:10.1002/(sici)1097-0258(20000515)19:9¡1141::aid-sim479¿3.0.co;2-f.