An Online Approach to Determine Correlation between Data Streams

Devesh Kumar Lal[†]
School of Computer Science & Information Technology
Devi Ahilya University, Indore, MP, India
devesh2222@gmail.com

Ugrasen Suman
School of Computer Science & Information Technology
Devi Ahilya University, Indore, MP, India
ugrasen123@yahoo.com

ABSTRACT

Real time stream processing demands processed outcomes in minimal latency. Massive streams are generated in real time where linear relationship is determined using correlation. Existing approaches are used for correlating static data sets such as, Kendall, Pearson, and Spearman etc. These approaches are insufficient to solve noise free online correlation. In this paper, we propose an online ordinal correlation approach having functionalities such as single pass, avoiding recalculation from scratch, removing outliers, and low memory requirements. In this approach, Compare Reduce Aggregate (CRA) algorithm is used for determining association between two feature vectors in real time using single scanning technique. Time and space complexities in CRA algorithm are measured as $O(n)$ and $O(1)$, respectively. This algorithm is used for reducing noise or error in a stream and used as a replacement of rank based correlation. It is recommended to have distinct elements and less variability in the streams for gaining maximum performance of this algorithm.

CCS CONCEPTS

• **General and reference** → Cross-computing tools and techniques → Design • **Mathematics of computing Probability and statistics** → Statistical paradigm → Time series analysis

KEYWORDS

Correlation approach; data analytics; online algorithm; real time big data; stream processing.

ACM Reference format:

Devesh Kumar Lal, and Ugrasen Suman. 2021. An Online Approach to Determine Correlation between Data Streams. *In the Companion of the 2021 ACM/SPEC International Conference on Performance Engineering, (ICPE'21 Companion), April 19-23, 2021, Virtual Event, France.* ACM, New York, NY, USA. *7* pages. https://doi.org/10.1145/3447545-3451900

1 INTRODUCTION

A linear relationship between two data streams is determined by using a straight line [1]. Here, data streams are used as a series of variables arrive from two different sources. An association between variables can be computed by applying correlation techniques [2]. In correlation techniques, an online approach may provide better understanding about linear relationship of streams in real time. This online correlation algorithm is helpful for various applications such as, determining odd streams, calculating fluctuation in streams, event based response over streams etc. [3].

Online correlation approach benefits can be understood by real time tweets from Twitter Api [4]. A real time application of cricket matches (as an event) is illustrated in Figure 1 along with its associated tweets count in time based manner. In Figure 1, two matches such as, RCB vs. RR and MI vs. DC, occur one after another. Therefore, the discussion on Twitter platform shows variable aggregated hash tag in time series manner. Here, hash-tag is aggregated for a fixed number of time slots. Figure 1 clearly indicates that a frequency of particular event is increased with the occurrence of event. This graph can be used to find correlation between two real time events, whereas correlation is discussed in subsequent sections.

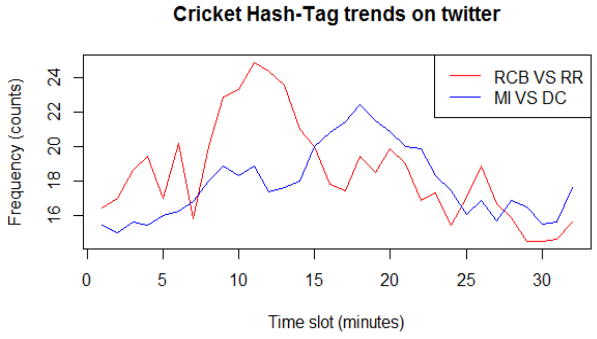
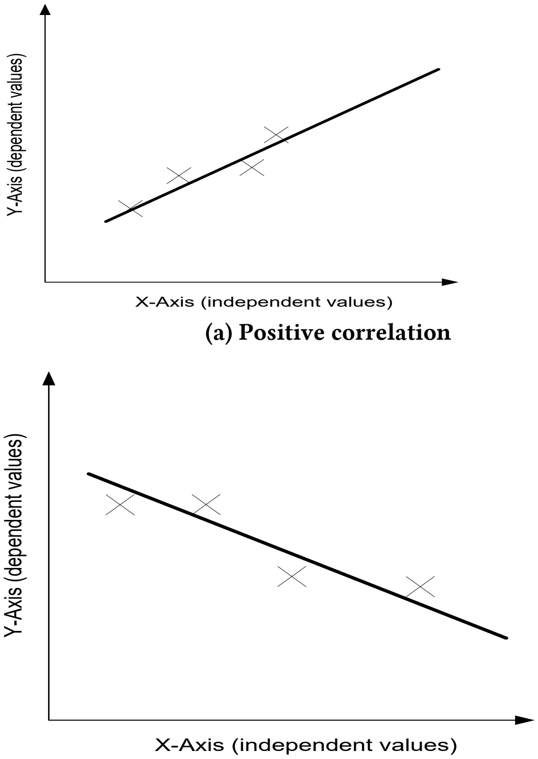


Figure 1: Real time Cricket matches hash-tag trends on twitter.

Correlation is a technique in data analytics used for various purposes such as, feature selection, finding linear relationship,

ACM ISBN 978-1-4503-8331-8/21/04...$15.00.
DOI: https://doi.org/10.1145/3447545.3451900

determining trends etc. Correlation can also be used for reliability, validation, prediction, and verification process. Correlation is achieved by comparing two different feature vectors. A comparison between data streams of two feature vectors may provide results as positive, negative and neutral followed by a factor of correlated strength [5, 6]. A positive and negative correlation is depicted in Figure 2a and Figure 2b, respectively. These figures of different correlation illustrate dependency of ‘y’ streams over ‘x’ streams.



(a) Positive correlation

(b) Negative correlation

Figure 2: Types of correlation.

An association between ‘x’ and ‘y’ streams is determined using correlation approach such as, Pearson product moment coefficient and various Rank based correlation approaches. Among them, Pearsons’s coefficient of correlation is the most popularly used correlation approach. It can be used either in historical data sets or in online approach. Pearsons’s coefficient of correlation equation 1 and equation 2 are used for computing coefficient of correlation that can also be applied over historical data sets. Online correlation is performed using equation 3, where it may not require sample mean to perform correlation. Here, N and n represent population size and sample size, repectively.

$$population\ covariance\ \sigma_{xy} = \frac{\sum_{i=0}^{N}(x_i - \mu_x)(y_i - \mu_y)}{N}$$

$$Sample\ covariance\ S_{xy} = \frac{\sum_{i=0}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

$$coefficient\ of\ correlation\ \rho = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

$$Sample\ coefficient\ of\ correlation\ S = \frac{S_{xy}}{S_x S_y} \qquad (1)$$

By placing equation of covariance upon standard deviation will provide equation (2).

$$Sample\ coefficient\ of\ correlation = \frac{\left[\frac{\sum_{i=0}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}\right]}{\left[\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{n-1}\right] \times \left[\frac{\sum_{i=1}^{N}(y_i - \bar{y})^2}{n-1}\right]} \qquad (2)$$

Equation 2 is modified for real time processing where sample mean is removed and equation 3 is obtain. This equation is beneficial for determining real time correlation.

$$Sample\ coefficient\ of\ correlation = \frac{\left[\sum_{i=1}^{n} x_i y_{i-} \frac{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n}\right]}{\left[\sum_{i=1}^{n} {x_i}^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}\right] \times \left[\sum_{i=1}^{n} {y_i}^2 - \frac{(\sum_{i=1}^{n} y_i)^2}{n}\right]} \qquad (3)$$

An online correlation may be performed by using various approaches such as, two-pass approach, naïve one-pass, incremental one pass etc. In equation 3, Pearsons’s coefficient of correlation results in computation for each streams generation. There are certain drawbacks associated with using Pearsons’s coefficient of correlation equations. It is strongly sensitive towards outlier, noises present in data streams. Second, its performance is limited at the time of heteroscedasticity in static data [7]. Heteroscedasticity is a situation where independent variables possess similar variance but increase in error, as it is unbiased towards square of mean difference. It can be overcome by effective use of windowing mechanism.

Another form of correlation is ordinal association or rank based correlation. Some of the rank correlation techniques are Spearman Rank Correlation, Goodman Kruskal’s gamma, Kendall’s Tau, Somer’s D, etc. Among them, two most popularly used Rank correlation approaches are Kendall’s Tau and Spearman Rank Correlation Coefficient; the corresponding equations are mentioned in this paper as equations (4) and (5) respectively. In Spearman Rank Correlation, rank has been assigned to attributes either increasing or decreasing order. In equation 4, Spearman Rank coefficient of Correlation ρ (Rho) is derived.

$$\rho = 1 - \frac{6(\sum d^2 + f)}{n(n-1)} \qquad (4)$$

Here, d indicates difference between two ranks with n number of observations, whereas f is calculated for repeated items. In case of similar items, a set of additional computation is required. A combined rank is allotted which can be determined as sum of rank for total similar items divided by total number of items. Here, m indicates repeated counts in a data sets whereas ‘f’ is determined using following equation.

$$f = \frac{m(m^2 - 1)}{12}$$

Whereas relative rank is assigned in Kendall's Tau approach as shown in equation (5). Where 'S' indicates overall sum and 'n' is number of items.

$$\tau = \frac{2S}{[n(n-1)]} \qquad (5)$$

The major drawback of using rank based correlation is as follows. Rank correlation may not be used as online algorithm. Here, the entire data stream has to be available for respective rank allotment. Allotment of ranks itself requires multiple scanning because this approach is similar to sorting a data in a set. Rank based correlation is based on approximation as it may behave similar for multiple streams sample size. Rank correlation is depicted in Table 1, where a window size of 5 elements is used to assign a respective rank. In first pass, element 55 assigned to rank 1 whereas other elements are marked with ND (not defined). Next pass 37, 25, 15, and 12 is assigned as rank 2, 3, 4, and 5, respectively. A total of four passes occurs for complete assignment of rank, or in other words rank assignment require four complete scanning for five data elements.

Compare Reduce Aggregate (CRA) algorithm uses single scanning to provide correlation results. Here, six different parameters are used to determine correlation. These parameters are updated with every passing data streams. A correlation may be determined in any time interval by using this parameters.

The paper is organized as follows. Section 2 explains about related work on the field of real time stream processing by correlation. Section 3 describe about our proposed algorithms. Section 4 describes the experimental setup, implementation, and result. Section 5 concludes the paper with future work

Table 1: Number of passes for Rank correlation.

Window of size 5	1st Pass	2nd Pass	3rd Pass	4th Pass
25	ND	ND	3 Rank	3 Rank
15	ND	ND	ND	4 Rank
37	ND	2 Rank	2 Rank	2 Rank
55	1 Rank	1 Rank	1 Rank	1 Rank
12	ND	ND	ND	5 Rank

2 RELATED WORK

Online algorithms differ from traditional algorithms, where data sets are available from the beginning. The most popular online algorithms are Probabilistic counting, count-min sketch, top-k element problems etc. [8, 9, 10, 11]. Few research has been performed to determine linear relationship between data streams. A brief literature survey is performed as discussed below. A Boolean representation of time series streams is performed, where a streams is compared with a mean value of the stream [12]. The combination of Boolean streams is used for determining correlation between two sets of data. Distributed Correlation is applied over streaming time series data in sliding window manner [13]. Here, Pearson's correlation coefficient is applied over two consecutive sliding windows. Correlation is used in pre-processing phase of data mining for feature selection [14]. In multidimensional data analysis, Pearson's correlation coefficient is applied on static data [15]. Linear relationship between two variables of stock market is achieved through real time correlation [16].

In short, research works heavily depend on Pearson's correlation coefficient for determining linear relationship of data stream. Whereas, rank based correlation reduces noise, outlier in a data set but requires whole frame of data. Rank correlation is limited to correlation for fixed sized window because of multiple scanning of data as seen in Table 1. These multiple scanning or passes are not suitable for use cases, which requires a real time rank based correlation. Therefore, CRA correlation algorithm is used for an online version solution of rank based correlation. This approach for streaming data with least computational cost would be beneficial, where result is to be calculated in real time.

3 CRA ALGORITHM

We have proposed CRA algorithm, which is helpful for determining linear relationship in real time between two data streams. This algorithm can be applied for more than one feature vectors in a given time stamp. It receives data streams as an input and produces output as ranges from -1 to 1. CRA algorithm notations and its working are discussed in subsequent section.

CRA algorithm is executed by performing three different operations sequentially such as, compare, reduce and aggregate. In comparing stage, two different data streams are compared with one another. These compared results will assign as '1' or '-1' in reducer stage, the '1' indicates a compared stream is greater whereas, '-1' indicates vice-versa. In this algorithm, '1' and '-1' is termed as high and low values. The last stage aggregates an overall high and low value.

In CRA algorithm, two streams of s_i and s_j are compared in one pass. A single scan may produce six different attributes namely, $h1, l1, h2, l2$, s and d that are shown in Table 2.

Table 2: Notation.

Symbols	Definition
$h1$	Positive similarity between first stream
$l1$	Negative similarity between first stream
$h2$	Positive similarity between second stream
$l2$	Negative similarity between second stream
s	Positive similarity between first and second streams
d	Dissimilarity between first and second streams
W	Weight Associativity i.e. positive, negative or neutral
f_i	Feature vectors of stream, where $i \rightarrow 0 \ to \ n$

s_i stream is reactive to $h1$ and $l1$, where these variables are adaptive for each record. In any instance, these variables may produces a recent processed values. Whereas, s_j stream produces variables such as, $h2$ and $l2$. A combined similarity comparison between s_i and s_j streams are stored in variables s and d. These parameters may help to provide linear relationship between s_i and s_j streams by equation 6. Maximum of variables s and d provides a direction of positive or negative growth. A $max(s, d)$ is subtracted by mode of difference between $|h1 - l1|$ and $|h2 - l2|$. The total terms divided by the sum of s and d. A number of records ingested for computation in a particular time slice is always $(s + d + 1)$.

$$Correlation = (W) * \frac{max(s, d) - |(|h1 - l1| - |h2 - l2|)|}{s + d} \tag{6}$$

Weight (W) is associated with the direction of relationship, which will be computed by using three different cases as shown below. Positive and negative association is determined in any instance by placing respective variables in following cases.

Case I $|h1 - l1| = |h2 - l2| = |s - d|$

$$W = -1, \forall \ \frac{|h1 - l1| + |h2 - l2|}{2} > s - d$$

$$W = 1, \forall \ \frac{|h1 - l1| + |h2 - l2|}{2} \leq s - d$$

Case I works for equal mode differences of $|h1 - l1|$, $|h2 - l2|$ and $|s - d|$. W is assigned with -1 when the average sum of mode difference is greater than differences occurred by two streams and 1 is assigned in other condition. This special case occurs majorly when streams possess a strong positive or negative relationship.

Case II $|h1 - l1| = |h2 - l2| \neq |s - d|$

$$W = -1, \forall \ \frac{|h1 - l1| + |h2 - l2|}{2} > |s - d|$$

$$W = 1, \forall \ \frac{|h1 - l1| + |h2 - l2|}{2} \leq |s - d|$$

Here, case II is similar to case I with difference in on comparison with mode value of $(s - d)$. This case majorly falls in the bracket of deciding weak to strong linear association.

Case III $|h1 - l1| \neq |h2 - l2| \neq |s - d|$

$$W = -1, \forall \ max\left|\left((h1 - l1), (h2 - l2)\right)\right| + remain((h1 - l1), (h2 - l2)) > |s - d|$$

$$W = 1, \forall \ max\left|\left((h1 - l1), (h2 - l2)\right)\right| + remain((h1 - l1), (h2 - l2)) \leq |s - d|$$

It is the most general case used for determining weak to strong linear relationship. Here, difference of $(h1 - l1)$ and $(h2 - l2)$ are not absolute. The maximum value is added with remaining elements of the set. This approach can be explained through an example of two sets of feature vectors. Let us suppose that real time streams are generated in any time stamp such as, $f_1 = \{25, 35, 40, 55, 45, 60, 70, 85, 95, 98\}$ and $f_2 = \{32, 44, 51, 49, 44, 55, 61, 68, 72, 79\}$. These elements are compared and reduced simultaneously. The reduced values are aggregated in the form of $h1, l1, h2, l2, s1$ and d. The calculation performed by CRA algorithm and final results of variables $h1, l1, h2, l2, s1$ and $d1$ are depicted in Figure 3. These variable are adaptive to change for every arrival instance. In this figure, the variables satisfy case III of CRA algorithm. Strength is calculated by placing the values in equation 6.

$$correlation = \frac{(W) * (8) - abs(5 - 7)}{8 + 1} = \frac{(W) * 6}{9}$$
$$= (W) * (0.66)$$

Here, (W) is calculated by using case III of CRA equation, by placing the values we got $(W) = -1$. Therefore, the final calculated value is -0.666, which lies under negatively correlation.

CRA Algorithm 1 consists of three conditional statements executed after every entries of stream. The n streams provide 3n comparisons, whereas three comparisons are required for determining sign and one constant equation used for finding correlation. Therefore, the polynomial equation is 3n+3n+1 for this algorithm. Upper bound of this polynomial equation is O(n), which is the time complexity of this algorithm. This algorithm uses six variables for storing n numbers of stream. A constant number of storage provides constant space complexity. Therefore, space complexity for this approach is constant O(1). Pseudo code for CRA algorithm is shown below as Algorithm 1. Here, input parameter receives data streams of length varies from 1 to n. In this algorithm, stream length of one is considered. This algorithm produces correlated result in output parameters. In line 2 to 8 and 31 to 36 initialised with temporary variable and six essential variables. Here, three different cases as mention above are placed on line number 49, whereas line 47 is used correlation calculation.

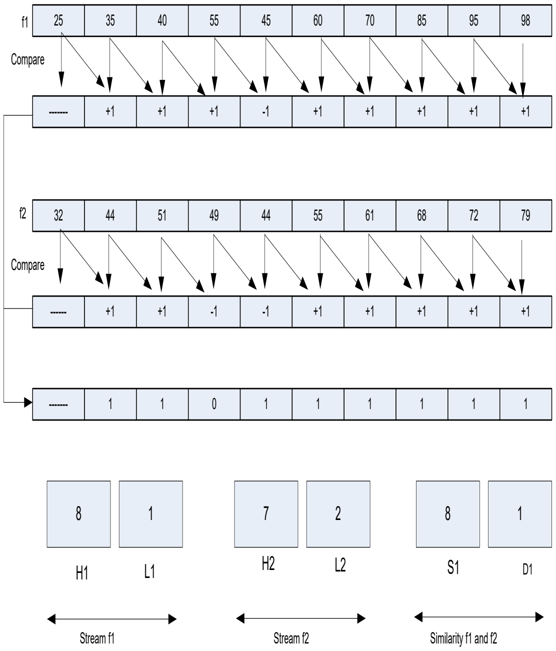


Figure 3: CRA algorithm example.

Algorithm 1 : CRA algorithm

In this algorithm, temporary variables $size\_x$, $size\_y$, i and j are used. Here, two different procedure is defined for generating six essential parameters for CRA. The *procedure binarized_1 (arg1, arg2)* generates four parameters, i.e., *x1, y1, x2,* and *y2.* The *procedure binarized_2 (arg1, arg2)* calculates correlation *(corr)* and check for cases I, II, and III.

Input parameter: data stream 's'.

Output parameter: correlation result.

```
1.  $\textbf{procedure}\ binarised\_1(arg1, arg2)$
2.      $x1 = 0$
3.      $y1 = 0$
4.      $x2 = 0$
5.      $y2 = 0$
6.      $i = 1\ ,\ j = 1$
7.      $size\_x = length(arg1)$
8.      $size\_y = length(arg2)$
9.      $\textbf{repeat}$
10.         $\textbf{if}\ (arg1[i] < arg1[i+1])$
11.             $x1 = x1 + 1$
12.         $\textbf{else}$
13.             $y1 = y1 + 1$
14.         $i = i + 1$
15.         $\textbf{if}\ (i == size\_x)$
16.             $\textbf{break}$
17.     $\textbf{repeat}$
18.         $\textbf{if}\ (arg2[i] < arg2[i+1])$
19.             $x2 = x2 + 1$
20.         $\textbf{else}$
21.             $y2 = y2 + 1$
22.         $j = j + 1$
23.         $\textbf{if}\ (j == size\_y)$
24.             $\textbf{break}$
25. $\textbf{end procedure}$
26. $\textbf{procedure}\ Binarized\_2(arg3,\ arg4)$
27.     $diff\_1 = 0$
28.     $oth\_1 = 0$
29.     $i1 = 1$
30.     $j1 = 1$
31.     $size1 = length(arg3)$
32.     $size2 = length(arg4)$
33.     $\textbf{repeat}$
34.         $\textbf{if}((arg3[i1] <\ arg3[i1 +$
    $1]) \&\& (arg4[j1] <\ arg4[j1+1])\ ||(arg3[i1] >$
    $arg3[i1+1]) \&\& (arg4[j1]\ >\ arg4[j1+1]))$
35.             $diff\_1 = diff\_1 + 1$
36.         $\textbf{else}$
37.             $oth\_1 = oth\_1 + 1$
38.         $i1 = i1 + 1$
39.         $\textbf{if}(i == size1)$
40.             $\textbf{break}$
41.     $oth\_2 = oth\_1\ + 1$
42. $total\_size = diff\_1 + oth\_2$
43. $t1 = |x1 - y1|$
44. $t2 = |x2 - y2|$
45. $t3 = |diff\_1 - oth\_2|$
46. $/*\ correlation\ equation\ */$
47. $Corr = \max(diff_1, oth_1) - |t1 - t2|/total\_size$
48. $/*\ calculate\ w\ equation\ */$
49. $check\ for\ three\ possible\ cases$
50. $\textbf{end procedure}$
51. $\textbf{procedure}\ call(void)$
52.     $binarised\_1(val1, val2)$
53.     $binarised\_2(val3, val4)$
54. $\textbf{end procedure}$
55. $\textbf{procedure}\ correlation()$
56.     $\textbf{for}\ (val1 = 0 \rightarrow n\ \textbf{do})$
57.         $\textbf{for}\ (val2 = 0 \rightarrow (n - val1)\ \textbf{do})$
58.             $call(void)$
59. $\textbf{end procedure}$
```

This algorithm can be applied over a series of feature vectors, as mentioned from line numbers 55 to 58. Here, *function call(void)* initiate the program and executed for two dimensional feature vectors. It is implementing with the help of two *for loops.* An implementation and testing of this approach is explained below.

4 EXPERIMENTAL RESULTS

Experiments are performed on intel core2Duo CPU E7500 @2.93GHzX2. The machine has 1.8 GB of DDR3 RAM with a disk size of 57.5GB, running ubuntu16.04 LTS x86_64. Dataset generated by python 2.7.12 with gcc version5.4.0, execution is performed on R programming language. We have used real and synthetic data sets. In real data sets, mtcars, iris, USArrests, etc. are used as real time streams in R programming for sample size less than 100 units. Whereas, for sample size more than 1000 units synthetic data streams are used to test our CRA algorithm. These data streams is generated using random generator with less number of variability and repeated values.

The results are generated by performing correlation on two feature vectors. The number of values associated with feature vectors are computed with different length. Results is compared with three different existing correlation approaches such as, Kendall, Pearson, and Spearman. Results are computed with different sets of inputs, which maps into window. A count based window is used with three different sizes. The first window consists values ranging from 1 to 100 units. Second window ranges from 100 to 1000 and last window contains values greater than 1000 units. These results will infer about CRA approach fall into the range of other three correlation approaches.

In Figure 4, inputs size of samples is less than or equals to hundred units. Here, variety of sample are passed through four different correlation techniques. A graph is plotted between different correlated results vs. different combinations of sample size (input). The following graph shows all four approaches inhibits similar correlation behaviour due to low sample size. The CRA approach as compared to other methods lies in the boundaries of Pearson's and rank correlations. Here, sample size is less than 100 units. Therefore, results from Kendall, Spearman and CRA approach shows deflection by comparing with Pearson's approach.

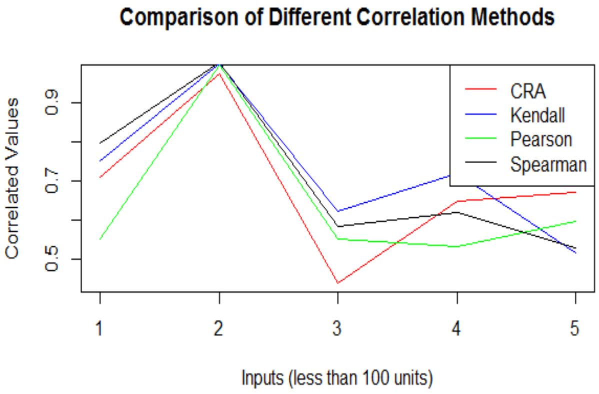


Figure 4: Correlation of input values $(input) \leq 100$.

Whereas in Figure 5, different input sizes ranges from 100 to 1000 are included. Here, higher variability of data sets is considered. The adequate sample size will enhance the correlation results. Therefore, all rank based and CRA approach show similar behaviour as compared to Pearson correlation.

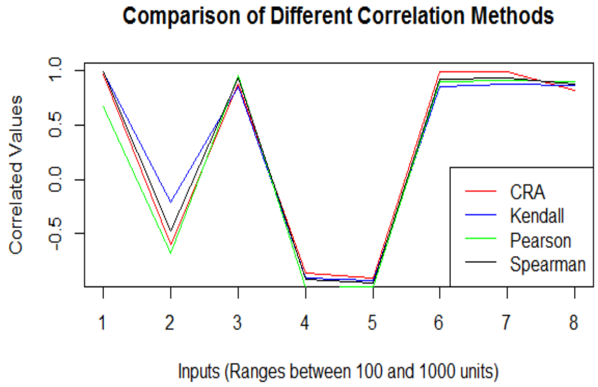


Figure 5 Correlation of input values $100 \leq (input) \leq 1000$.

The increased number of input sizes may saturate the correlation results by neglecting noises, errors, similar streams. Large number of input sizes may tends to show similar trajectory patterns by all correlation approaches. The graph for input more than 1000 counts is depicted in Figure 6. The higher counts of inputs may provide correlated results, which is overlapped by other four correlation approaches. Therefore, by considering the results on different sizes of input streams these approach may be suitable for various real time applications.

CRA approach helps in removing outliers to a certain extent, where data streams are treated as ordinals. The ordinal based comparison between data streams results incrementing by a unit in CRA parameters. These parameters remain neutral during occurrence of an outliers. There are some limitation of CRA approach which has to be considered before applying in real time application scenarios. CRA approach is an ordinal based association where equation $(a < b)$ is consider as $(0\ or\ 1)$ and its scalar values may not effect in correlation. If there is major repetition in a stream, in such scenario this approach may or may not provide better results.

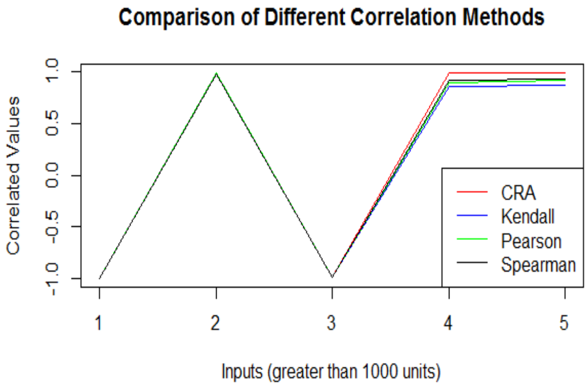


Figure 6: Correlation of input values $(input) \geq 1000$.

5 CONCLUSION AND FUTURE WORK

In this paper, an online correlation algorithm termed as CRA is proposed. This correlation technique is used for determining linear relationship between different data streams. Space complexity and time complexity for CRA algorithm are, $O(1)$ and $O(n)$, respectively. This online CRA approach may be used as alternative to other static rank based correlation algorithms for real time stream processing use cases. In future work, CRA algorithm can modified in a manner, where attributes are able to store the frequency count of a stream. These frequencies are beneficial for computing correlation of identical streams.

ACKNOWLEDGMENTS

I offer my most sincere gratitude to the Council of Scientific and Industrial Research (CSIR), Government of India, for financial support in the form of Senior Research Fellowship.

REFERENCES

[1] Ting, W. A. N. G., and Z. H. A. N. G. Shiqiang. "Study on linear correlation coefficient and nonlinear correlation coefficient in mathematical statistics." Studies in Mathematical Sciences 3.1 (2011): 58-63.

[2] Beer, Colin, David Jones, and Damien Clark. "Analytics and complexity: Learning and leading for the future." Proceedings of the 29th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE 2012). Australasian Society for Computers in Learning in Tertiary Education (ASCILITE), 2012.

[3] Su, Shen, et al. "A correlation-change based feature selection method for IoT equipment anomaly detection." Applied Sciences 9.3 (2019): 437.

[4] Wu, Ye, and Fuji Ren. "Learning sentimental influence in twitter." 2011 International Conference on Future Computer Sciences and Application. IEEE, 2011.

[5] Ari, Ismail, Erdi Olmezogullari, and Ömer Faruk Çelebi. "Data stream analytics and mining in the cloud." 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings. IEEE, 2012.

[6] Rettig, Laura, et al. "Online anomaly detection over big data streams." Applied Data Science. Springer, Cham, 2019. 289-312.

[7] Tse, Yiu Kuen, and Albert K. C. Tsui. "A multivariate generalized autoregressive conditional heteroscedasticity model with time-varying correlations." Journal of Business & Economic Statistics 20.3 (2002): 351-362.

[8] Flajolet, Philippe, and G. Nigel Martin. "Probabilistic counting algorithms for data base applications." Journal of computer and system sciences 31.2 (1985): 182-209.

[9] Cormode, Graham, and Shan Muthukrishnan. "An improved data stream summary: the count-min sketch and its applications." Journal of Algorithms 55.1 (2005): 58-75.

[10] Lal, Devesh Kumar, and Ugrasen Suman. "A Survey of Real-Time Big Data Processing Algorithms." Reliability and Risk Assessment in Engineering. Springer, Singapore, 2020. 3-10.

[11] Metwally, Ahmed, Divyakant Agrawal, and Amr El Abbadi. "Efficient computation of frequent and top-k elements in data streams." International conference on database theory. Springer, Berlin, Heidelberg, 2005.

[12] Zhang, Tiancheng, et al. "Adaptive correlation analysis in stream time series with sliding windows." Computers & Mathematics with Applications 57.6 (2009): 937-948.

[13] Palma-Mendoza, et al. "Distributed correlation-based feature selection in spark." Information Sciences 496 (2019): 287-299.

[14] Guyon, Isabelle, and André Elisseeff. "An introduction to variable and feature selection." Journal of machine learning research 3.Mar (2003): 1157-1182.

[15] Zhang, Yi, et al. "Improved visual correlation analysis for multidimensional data." Journal of Visual Languages & Computing 41 (2017): 121-132.

[16] Attigeri, Girija V., et al. "Stock market prediction: A big data approach." TENCON 2015-2015 IEEE Region 10 Conference. IEEE, 2015.