Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 Keller Hall
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 14-025

Discovering the Longest Set of Distinct Maximal Correlated Intervals in Time Series Data

Gowtham Atluri, Michael Steinbach, Kelvin Lim, Angus MacDonald, Vipin Kumar

October 1, 2014
Discovering the Longest Set of Distinct Maximal Correlated Intervals in Time Series Data

Gowtham Atluri*, Michael Steinbach*, Kelvin O. Lim†, Angus MacDonald III‡ and Vipin Kumar*

*Dept. of Computer Science
University of Minnesota
†Dept. of Psychiatry
University of Minnesota
‡Dept. of Psychology
University of Minnesota

Abstract—In this paper we focus on finding all maximal correlated intervals where a given pair of time series have correlation above a user provided threshold for all its subintervals and for none of its immediate subsuming intervals. Our objective then is to find a longest set of such maximal correlated intervals. We propose a two step solution to achieve this objective. In the first step an efficient bottom-up approach is proposed to discover maximal correlated intervals. In the second step we use a dynamic programming approach to select the longest non-overlapping set. We evaluate the efficiency of our approach on synthetic datasets and compare it with that of a brute-force approach. Using neuroimaging data that contains activity time series from brain regions, we show the utility of our approach in studying transient nature of relationships between different brain regions.

I. INTRODUCTION

In this paper we focus on the problem of discovering similarities between time series of equal length. We are interested in capturing transient relationships that may exist in multiple non-overlapping time intervals. Traditional similarity measures that are computed on the entire time series are not appropriate, because they do not capture the transient similarity. For example, the pair of time series in Figure 1 have a correlation of 0.42, while there are smaller intervals [2 5] and [9 14] with a correlation of 1. Such transient relationships are of interest in several domains. In particular for fMRI (functional magnetic resonance imaging) data, time intervals where time series from two different brain regions are highly correlated indicate the periods where the brain regions are working synergistically to achieve a brain function. Different regions in the brain can be correlated for a longer spurts or for smaller spurts and discovering this information can be helpful in characterizing the synergistic behavior of the brain regions.

We are interested in finding all intervals where a given pair of time series have correlation above a user provided threshold for all its subintervals and for none of its immediate subsuming intervals. We refer to these as maximal correlated intervals. More formally, a maximal correlated interval is a time interval where a given pair of time series are highly correlated in the entire interval as well as all the sub-intervals, but no immediate subsuming interval is highly correlated. Note that high correlation in an interval does not automatically guarantee high correlation in all of its subintervals. For example, the time series in Figure 2 have a correlation of 0.95, while the correlation in the interval [4 9] is only 0.06. As there can be many maximal intervals of different sizes with potential overlaps (see Figure 3), our objective is to find the Longest set of non-overlapping Maximal correlated INtervals (LAMINA) (i.e., the set of intervals {2 7}, {11 15}) in Figure 3. As we will show later, LAMINA is useful in studying transient relationships between different brain regions.

Fig. 1. An example illustrating the presence of similarities in multiple small intervals. Intervals of interest are indicated with double sided arrows. (All figures in this paper are best seen in color).

Fig. 2. An example illustrating the need for maximal correlated intervals

Transient relationships between two time series is often studied using their relationship in small intervals of fixed size (often referred to as windows). Such relationships based on small intervals have been shown to be useful for several problems such as time series matching [1–7], classification [5], [6], clustering [7] and anomaly detection [8], [9]. As the name suggests these approaches use time windows of fixed size (say w) that are shifted by 1 time step and quantify relationships between two different windows.

These sliding window based approaches can also be used...
to assess similarity between time series of equal length. To achieve this, all sliding windows of length \( w \) are determined and the similarity between the two time series for each window is computed. The windows that satisfy a similarity constraint are determined as the windows where the two time series are similar. This is illustrated with an example in Figure 4. The two time series are shown in blue and green. Sliding windows of size 4 are used here and the correlation in each window is shown in the figure. The windows where correlation satisfies a threshold of 0.9 are indicated in red color.

![Figure 3. An example illustrating overlapping maximal correlated intervals. Note that the intervals [14] and [27] are maximal correlated intervals but they largely overlap.](image)

![Figure 4. An example illustrating a sliding window based approach for assessing similarity between two time series](image)

While sliding window based approaches can be used to determine the windows in which two time series are highly correlated (and the where they are not), they are not suited to discover maximal correlated intervals. It may appear that one could compute maximal correlated intervals by merging adjacent correlated windows that meet the correlation threshold. However, the correlation of the merged interval or any of its sub-intervals is not guaranteed to meet the threshold. For example, merging the windows [14], [25], [36], and [47] (each of which has a correlation of greater than 0.9), one can construct an interval [17]. However, in this interval the time series has a correlation of 0.84 that is smaller than the threshold of 0.9. Therefore, these merged windows do not form a maximal correlated interval.

Another approach to discovering the set of maximal correlated intervals is to exhaustively consider all possible intervals of different sizes, check whether the criteria for maximal correlated intervals is met, and filter out the intervals that are not maximal. This approach (referred to as a brute-force approach) is computationally intensive and can be infeasible when the time series are too long. In this paper we present a efficient bottom-up approach to enumerate all maximal correlated intervals for a pair of time series. Our approach starts with a smallest interval possible and constructs bigger intervals from those that meet the correlation threshold.

Once the maximal correlated intervals are discovered, we then present a dynamic programming approach to select the longest set of non-overlapping maximal correlated intervals from the ones enumerated by our bottom-up approach. Overall, our proposed approach has two parts, i) bottom-up approach to enumerate maximal correlated intervals, ii) a dynamic programming approach to discover the longest set of non-overlapping maximal correlated intervals from the set of maximal correlated intervals.

We evaluate the efficiency of our approach on multiple synthetic datasets and compared it with a brute-force approach. Our results suggest that the proposed approach is at least an order of magnitude faster than the alternative approach, even on time series of moderate sizes. We also evaluate our approach on a real world neuroimaging dataset to show the utility of LAMINA in identifying the relationship between different pairs of brain regions under two scenarios: i) while the subject is resting and ii) while the subject is watching cartoons.

Although our objective of LAMINA is defined using a correlation measure, the proposed approach can in general work with any measure that can be used to assess time series similarity. Our approach can also be used to discover maximal unrelated intervals where correlation for the interval and all sub-intervals is smaller than a threshold. This is particularly useful for discovering intervals where the time series are negatively correlated. As we will show in our evaluation, this can be used to discover maximal intervals where the time series are poorly correlated.

The key contributions of this paper are as follows:

- A problem formulation to capture distinct correlated intervals in a pair of time series.
- An efficient algorithm to discover the longest set of non-overlapping maximal correlated intervals.
- Experimental evaluation on synthetic data to demonstrate the efficiency of the proposed approach.
- Experimental evaluation on neuroimaging data to illustrate practical utility of the new formulation.

The rest of this paper is organized as follows. Section 2 presents our problem formulation. Section 3 discusses related work. The proposed method is presented in Section 4. Section 5 presents our evaluation and results. We conclude with Section 6.

### II. Problem Formulation

Let \( X \) and \( Y \) be two time series of equal length \( n \). We indicate the value of \( X \) at time \( i \) as \( X_i \).
Definition 1: Interval. The set of time points \((i \ldots j)\), where \(i < j\), \(i \geq 1\) and \(j \leq n\), is referred to as an interval \(I_{(i,j)}\).

The length of an interval \(I_{(i,j)}\), denoted using \(l_{(i,j)}\), is the number of time points covered by the interval and is computed as \(l_{(i,j)} = j - i + 1\). The set of values in a time series \(X\) in the interval \(I_{(i,j)}\) is represented using \(X_{(i,j)}\).

Definition 2: Non-overlapping Intervals. Two intervals \(I_{(a,b)}\) and \(I_{(c,d)}\) are said to be non-overlapping when \((a < c \text{ and } b < c)\) or \((c < a \text{ and } d < a)\), i.e., when the two intervals do not share any time points.

Definition 3: Correlated Interval. Given two time series \(X\) and \(Y\) an interval \(I_{(a,b)}\) is referred to as a correlated interval if (i) length \(l_{(a,b)} \geq \alpha\), (ii) Pearson’s correlation between \(X_{(a...b)}\) and \(Y_{(a...b)}\), exceeds a user-provided threshold \(\beta\), and (iii) for all subintervals \(I_{(a',b')}\) of length \(l_{(a',b')} \geq \alpha\), \(a' \geq a\), \(b' \leq b\), \(r(X_{(a'...b')}, Y_{(a'...b')}) \geq \beta\).

The first constraint that ensures that every interval is of \(\alpha\) or greater is useful to avoid very small intervals that may exhibit high correlations due to random noise in the data. The smallest interval for which a correlation can be computed is 3 (correlation for any interval of size 2 has a trivial correlation of 1). When there no noise in the data \(\alpha = 3\) can be used. The third constraint is useful to ensure that an interval that satisfies the correlation threshold \(\beta\) does not contain an interval with a correlation lower than the threshold. This will address the problem illustrated in Figure 2.

Definition 4: Maximal Correlated Interval. A correlated interval that is not subsumed by an immediately larger correlated interval is treated as a maximal correlated interval. Formally, \(l_{(a,b)} \geq \alpha\), \(r(X_{(a,b)}, Y_{(a,b)}) \geq \beta\) is a correlated interval and is maximal when \(r(X_{(a-1,b)}, Y_{(a-1,b)}) < \beta\) and \(r(X_{(a,b+1)}, Y_{(a,b+1)}) < \beta\).

Note that two adjacent intervals \(I_{(a,b)}\) and \(I_{(a+1,b+1)}\) can be maximal correlated when there is no immediately subsuming correlated interval \(I_{(a,b+1)}\). These adjacent intervals overlap largely. To be able to capture distinct correlated intervals in such scenarios, we define the notion of non-overlapping maximal correlated intervals.

Definition 5: Non-overlapping Maximal Correlated Intervals. Given two time series \(X\) and \(Y\) and the set of all maximal correlated intervals \(I = \{I_{(a,b)}, I_{(c,d)}, \ldots I_{(k,l)}\}\), non-overlapping intervals are those subset of intervals in which each interval is a maximal correlated interval and every pair of intervals are non-overlapping.

Definition 6: Longest Set of Non-overlapping Maximal Correlated Intervals. The set of non-overlapping maximal correlated intervals whose sum of intervals is the largest is referred to as the ‘longest set of non-overlapping maximal correlated intervals’, also referred to as LAMINA.

In Section 4 we provide an efficient approach to discover a LAMINA from a pair of time series.

III. RELATED WORK
To the best of our knowledge, no existing work in time series data mining captures the goal of LAMINA that is defined in this paper. In this section we discuss the approaches that are related to our objective.

One related problem is that of time series subsequence matching [2]–[4], [10]–[14] that has been widely studied in the last two decades. Given a dataset of time series and a query time series, the problem of subsequence matching is to discover all time series from the dataset such that at least one window of fixed size in a time series is similar to a window in the query time series. This problem can be addressed in many angles: (i) They consider matches that are not necessarily between the same intervals in two time series, (ii) Their approaches typically work with fixed size intervals while we look for arbitrarily long correlated intervals, (iii) They look for matching at least one window while we are interested in discovering all non-overlapping maximal correlated intervals.

Another related problem is dealt by Li et al. [1]. Given a query time series, a set of time series and a correlation threshold, they addressed the problem of discovering a time series that share a longest correlated interval with a given query time series. Our objective differs from this in many aspects: (i) They look for longest correlated interval, while we look for maximal correlated interval, (ii) They look for ‘one longest correlated interval while we are interested in discovering all maximal correlated intervals, (iii) In their case the intervals of similarity need not be at the exact same location in both the time series, while we study similarity between the same intervals in both the time series.

IV. PROPOSED METHODS
The problem of finding the longest set of non-overlapping maximal correlated intervals can be dealt with in two parts. First, the set of maximal correlated intervals can be enumerated. Second, the longest set of non-overlapping maximal correlated intervals can be discovered from the set of maximal correlated intervals. In this section we first show a brute force method to enumerate all the maximal correlated intervals and we propose an efficient method for this problem. We present a dynamic programming based solution for discovering the longest set of non-overlapping maximal correlated intervals for the latter part of the problem.

A. Discovering Maximal Correlated Intervals

1) Brute-force Approach: The problem of discovering maximal correlated intervals can be divided into three subproblems: (i) enumerating all intervals that satisfy the correlation threshold, (ii) pruning correlated intervals whose subintervals are not correlated, and (iii) discovering maximal correlated intervals among them.

a) Enumerating All Intervals: First, let us consider the problem of finding all intervals that satisfy the correlation threshold for two time series \(X\) and \(Y\). Note that the definition of correlated intervals entails three constraints: (i) minimum length of interval \((\alpha)\) (ii) minimum correlation \((\beta)\) (iii) condition to avoid intervals whose subintervals are not correlated. In this part we will only address the first two constraints and the third constraint will be addressed in the next part.

A brute force approach enumerates each interval of valid length \((\geq \alpha)\) and tests if the correlation threshold \(\beta\) is satisfied. From the first constraint, the candidate intervals are all
intervals of length $\alpha$ that potentially start at every time point. For a chosen interval length $l$, the number of valid intervals to consider in a time series of length $n$ are $n-l+1$. Therefore the total number of valid intervals of all valid lengths are $\sum_{l=\alpha}^{n} n - \alpha + 1$. i.e., $O(n^2)$ and computing correlation for an interval of length $l$ takes $O(l)$ time. Note that $l$ can potentially approach $n$ when there are longer correlated intervals. Overall, the computational complexity of brute force enumeration is $O(n^3)$.

Algorithm 1 BruteForceIntervalEnumeration

Input:
i. $X$ and $Y$, two real valued time series of length $n$
ii. $\alpha$, interval length threshold
iii. $\beta$, correlation threshold

Output:
$Intvl_{Mat}$ where a 1 in element $(i,j)$ indicates that interval $I_{(i,j)}$ is of length at least $\alpha$ and has correlation at least $\beta$

1. $Intvl_{Mat} =$ zero matrix of size $n \times n$
2. $Corr_{Ints} =$ $\phi$
3. for $win_{len} =$ $\alpha$ to $(n-1)$
4. for $i =$ 1 to $(n - win_{len} + 1)$
5. $r =$ $corr(X(i,i+win_{len}-1), Y(i,i+win_{len}-1))$
6. if $(r \geq \beta)$
7. $Intvl_{Mat}[i,i+win_{len}-1] =$ 1
8. end if
9. end for
10. Exit if no intervals with correlations $\geq \beta$ are found for this $win_{len}$
11. end for
12. Return $Intvl_{Mat}$

The brute force approach for enumerating all correlated intervals is shown in Algorithm 1. $Intvl_{Mat}$ is a two-dimensional array where a value 1 will be placed in element $(i,j)$ when the interval $I_{(i,j)}$ of length at least $\alpha$ and has correlation at least $\beta$ and 0 otherwise. The algorithm iterates over all valid interval lengths in steps 3-11 and over all possible intervals of a chosen length $win_{len}$ in steps 4-9. The correlation is tested in step 6 and $Intvl_{Mat}$ is updated in step 7. Note that the algorithm exits in step 10 if no intervals are found to be correlated for a given interval length.

There can be two types of irrelevant correlated intervals that are discovered by Algorithm 1: i) correlated intervals whose subintervals of length at least $\alpha$ are not correlated intervals. ii) correlated intervals that are not maximal. In the following we address each of these issues separately.

b) Pruning Spurious Intervals: Once all the intervals that satisfy the correlation threshold are enumerated, in order to address the third constraint in the definition of correlated intervals we prune those intervals whose subintervals are not correlated. The pruning step is performed for each interval that is considered in step 5 with correlation $< \beta$, by listing all the enclosing intervals and filtering out all the enclosing intervals with correlation $\geq \beta$. This approach is shown in Algorithm 2. Steps 1-10 iterate over all valid interval lengths and steps 2-9 iterate over all possible intervals of a given length ($win_{len}$). For a chosen interval $(i, i+win_{len}-1)$ all enclosing intervals are determined in step 4 and their corresponding values in $Intvl_{Mat}$ are set to 0 to suggest that they are not valid correlated intervals. The computational complexity of the $PruningSpuriousIntervals$ is $O(n^2)$.

Algorithm 2 PruneSpuriousIntervals

Input:
$Intvl_{Mat}$ that encodes all correlated intervals

Output:
$Intvl_{Mat}$ where a 1 in element $(i,j)$ indicates that interval $I_{(i,j)}$ is of length at least $\alpha$ and has correlation at least $\beta$ and all smaller intervals of length $\alpha$ or more are also correlated intervals

1. for $win_{len} =$ $\alpha$ to $(n-1)$
2. for $i =$ 1 to $(n - win_{len} + 1)$
3. if $(Intvl_{Mat}[i,i+win_{len}-1] = 0)$
4. enclosing_ints = list all enclosing intervals
5. for all enclosing intervals $(a,b)$
6. $Intvl_{Mat}[a,b] =$ 0
7. end for
8. end if
9. end for
10. end for
11. Return $Intvl_{Mat}$

c) Enumerating Maximal Intervals: The goal of this part is to enumerate the maximal correlated intervals. This can be achieved by testing for every correlated interval if an immediately subsuming interval is also correlated. If no subsuming interval is correlated, a correlated interval can be enumerated as a maximal interval. This approach is shown in Algorithm 3. Steps 1-9 iterate over all valid interval lengths and steps 2-8 iterate over all possible intervals of a given length $win_{len}$. For a chosen interval $(i, i+win_{len}-1)$ if the correlation threshold is satisfied $(Intvl_{Mat}[i,i+win_{len}-1] = 1)$ and if its immediately enclosing intervals did not satisfy the correlation threshold (step 4) then interval $I_{(i,i+win_{len}-1)}$ is added to the list of maximal correlated intervals. The computational complexity of the $ListMaximalIntervals$ is $O(n^2)$.

Algorithm 3 ListMaximalIntervals

Input:
$Intvl_{Mat}$ that encodes all correlated intervals

Output:
$Corr_{Ints}$ List of all maximal correlated intervals

1. for $win_{len} =$ $\alpha$ to $(n-1)$
2. for $i =$ 1 to $(n - win_{len} + 1)$
3. if $(Intvl_{Mat}[i,i+win_{len}-1] = 1)$
4. if $(Intvl_{Mat}[a,b] = 0)$ for all enclosing intervals $(a,b)$
5. $Corr_{Ints} =$ $Corr_{Ints}$,$I_{(j,i+j+win_{len}}$
6. end if
7. end if
8. end if
9. end for
10. Return $Corr_{Ints}$

2) A Bottom-up Approach: The brute force approach first enumerates all intervals of length $\alpha$ and longer that satisfy the correlation threshold $\beta$. It then prunes out spurious as well as non-maximal correlated intervals. Here we propose a relatively efficient approach (as we show in our evaluation section) that does not compute correlations for all intervals to determine correlated intervals.

a) Enumerating Correlated Intervals: The third constraint in the definition of a correlated interval is that all
subintervals of length \( \alpha \) or more are also required to have a correlation \( \geq \beta \). One way to address this constraint would be to start with all intervals of length \( \alpha \) and then build bigger intervals only when immediately smaller intervals satisfy the correlation threshold. Using this observation, we propose a bottom-up enumeration scheme where we compute correlation of a longer interval only when both the immediate sub-intervals are correlated. Formally, if intervals \( I_{(a,b-1)} \) and \( I_{(a+1,b)} \) satisfy the correlation threshold, only then the correlation of the interval \( I_{(a,b)} \) is evaluated. Using this procedure has two advantages: i) we do not have to evaluate correlations for all the candidate intervals, and ii) we can avoid the pruning spurious intervals step in the case of the brute force approach (shown in Algorithm: PruneSpuriousIntervals). The computational complexity of this approach is also \( O(n^3) \). However, the number of correlated intervals found using the brute-force approach are lower bounded by the number of correlated intervals discovered using this bottom-up approach. When most of the candidate intervals are correlated, the number of correlations evaluated by the bottom-up approach will approach the number of correlations evaluated by the brute-force approach.

**Algorithm 4** BottomUpIntervalEnumeration

**Input:**
- \( i. \) \( X \) and \( Y \), two real valued time series of length \( n \)
- \( ii. \) \( \alpha \), interval length threshold
- \( iii. \) \( \beta \), correlation threshold

**Output:**
A matrix \( \text{Intvl} \_ \text{Mat} \) indicating all correlated intervals with 1’s

1. \( \text{Intvl} \_ \text{Mat}[1 : \text{num} \_ \text{win}, 1 : \text{num} \_ \text{win}] = 0 \)
2. for \( \text{win} \_ \text{len} = \alpha \) to \( n - 1 \)
3. \hspace{1em} for \( i = 1 \) to \( (n - \text{win} \_ \text{len} + 1) \)
4. \hspace{2em} if \( (\text{win} \_ \text{len} = \alpha) \)
5. \hspace{3em} if \( (r \geq \beta) \)
6. \hspace{4em} \text{Intvl} \_ \text{Mat}[i, i + \text{win} \_ \text{len} - 1] = 1
7. \hspace{1em} end if
8. \hspace{1em} end if
9. \hspace{1em} if \( (\text{win} \_ \text{len} > \alpha) \)
10. \hspace{2em} and \( \text{Intvl} \_ \text{Mat}[i, i + \text{win} \_ \text{len} - 2] = 1 \)
11. \hspace{2em} and \( \text{Intvl} \_ \text{Mat}[i + 1, i + \text{win} \_ \text{len}] = 1 \)
12. \hspace{2em} if \( (r \geq \beta) \)
13. \hspace{3em} \text{Intvl} \_ \text{Mat}[i, i + \text{win} \_ \text{len} - 1] = 1
14. \hspace{2em} end if
15. \hspace{1em} end if
16. exit if no correlated intervals are found for this \( \text{win} \_ \text{len} \)
17. end for
18. return \( \text{Intvl} \_ \text{Mat} \)

The bottom-up interval enumeration is shown in Algorithm 4. Similar to brute force enumeration scheme it iterates over all valid interval lengths and over all possible intervals for a chosen interval length in steps 2-19 and 3-18, respectively. When the interval length is the least possible (\( \alpha \)), the correlations are computed for all the intervals. For other interval lengths \( l \), correlation is estimated only if the two smaller subintervals of length \( l - 1 \) were found to have a correlation of at least \( \beta \) (steps 11-15). Note that the algorithm exits when no intervals are found to be correlated for a given interval length.

**b) Enumerating Maximal Intervals:** Once the correlated intervals are determined in a bottom up fashion the maximal intervals can then be enumerated by listing only those intervals whose immediate enclosing intervals are not correlated. Formally, if \( I_{(a,b)} \) is correlated and neither \( I_{(a-1,b)} \) and \( I_{(a,b+1)} \) are correlated, \( I_{(a,b)} \) is listed as a maximal correlated interval. Algorithm 3 (ListMaximalIntervals) does this.

**B. Discovering the Longest Set of Non-overlapping Maximal Correlated Intervals**

Given a set of potentially overlapping intervals, the goal is to find the longest set of non-overlapping intervals. This problem can be treated as the classical dynamic programming problem of weighted interval scheduling [15] where the objective is to determine a schedule such that no two scheduled jobs overlap in time and the entire schedule maximizes the sum of weights of scheduled jobs. Therefore, by treating the each maximal correlated interval as a job and its length as the weight of the job, we can use the standard dynamic programming algorithm to find the desired longest set.

The dynamic programming algorithm FindLAMINA for discovering longest non-overlapping sets from maximal correlated intervals is shown in Algorithm 5. This approach first sorts the intervals in ascending order of their start times. It then starts at the end of the list of intervals and traces back to the first interval by determining at each step the optimum weight from the current interval to the end of the list of intervals by considering two choices: i) include the current interval ii) ignore the current interval. Once the optimal weight for the entire list of intervals is determined, one can trace from the first interval to the last to determine if an interval was included in the optimal list. The computational complexity is \( O(n \log n) \).

In summary, the brute force approach uses algorithms BruteForceIntervalEnumeration, PruneSpuriousIntervals, and FindLAMINA to determine LAMINA, given a pair of time series and parameters (\( \alpha \) and \( \beta \)). The bottom-up approach uses algorithms BottomUpIntervalEnumeration, ListMaximalIntervals, and FindLAMINA to determine LAMINA. Note that both these approaches share the last two algorithms. The key difference is in how the intervals are enumerated. Due to the bottom-up style of enumeration the latter approach reduces the search space and it does not have to filter out spurious correlated intervals. Although our approach has a worst complexity of \( O(n^3) \), as we will show in our evaluation, the bottom-up scheme is practically much faster.

**C. Proof of correctness**

We now prove that the proposed bottom-up approach discovers the longest set of non-overlapping maximal correlated intervals. The proposed approach relies on three components: i) Bottom-up enumeration approach ii) List Maximal correlated intervals iii) Discovering the longest set using a dynamic programming approach. The second part is a filtering step where all non-maximal intervals are filtered out and the dynamic programming approach in the third part is proven to be correct [15]. So, we are left with proving that the bottom-up enumeration approach lists all intervals and their sub-intervals of length at least \( \alpha \) are correlated.

**Theorem 1:** Bottom-up interval enumeration algorithm lists all intervals and their sub-intervals of length at least \( \alpha \) when they pass the correlation threshold \( \beta \).
Algorithm 5 FindLAMINA

Input:
Set of \( k \) correlated intervals \( CI \)

Output:
A set of non-overlapping correlated intervals whose intervals are collectively the longest

1. \( \text{sorted} \_CI = \text{sort}_\text{by}_\text{starting}_\text{time}(CI) \)
2. \( \text{next}_\text{interval}[1 : k] = 0 \)
3. for \( i = 1 \) to \( k \)
   4. \( \text{foll}_\text{intvls} = \) an immediate interval starting after \( i \) ends
5. end for
6. \( \text{cum} \_\text{sum} \_\text{sel}_\text{intvls}[1 : k] = 0 \)
7. \( \text{sel}_\text{intvls}[1 : k] = 0 \)
8. for \( i = n \) to \( 1 \)
   9. \( \text{cum} \_\text{sum} \_\text{sel}_\text{intvls}[i] = \max(l_i + \text{cum} \_\text{sum} \_\text{sel}_\text{intvls}[	ext{foll}_\text{intvls}[i]], \text{cum} \_\text{sum} \_\text{sel}_\text{intvls}[i + 1]) \)
10. \( \text{sel}_\text{intvls}[i] = 1 \), if interval \( i \) was part of solution
11. end for
12. \( \text{Result} \_\text{Corr}_\text{Intvls} = \emptyset \)
13. \( \text{current} = \text{smallest} \_i \) with \( \text{sel}_\text{intvls}[i] = 1 \)
14. while true
15. 16. if \( \text{sel}_\text{intvls}[\text{current}] == 1 \)
17. 18. \( \text{Result} \_\text{Corr}_\text{Intvls} = \text{Result} \_\text{Corr}_\text{Intvls} \cup i \)
19. 20. \( \text{current} = \text{sel}_\text{intvls}[\text{current}] \)
21. end if
22. if \( \text{sel}_\text{intvls}[\text{current}] == 0 \)
23. \( \text{current} = \text{current} + 1 \)
24. end if
25. Exit if no more intervals can be selected
26. end while
27. return \( \text{Result} \_\text{Corr}_\text{Intvls} \)

Proof: This can be proved in two parts. i) When an interval is enumerated all subintervals of length at least \( \alpha \) are correlated. ii) All such intervals are enumerated.

We now prove the first part.

This part has two scenarios: When an interval is enumerated it has; a) either no subintervals of length \( \alpha \) or b) all its subintervals of length at least \( \alpha \) are correlated. The algorithm estimates the correlation of intervals of length \( \alpha \) and enumerates it when the correlation threshold is satisfied. Hence the former scenario is addressed. Correlation for an interval \( I_{(a,b)} \) is computed only when its subintervals \( I_{(a,b-1)} \) and \( I_{(a+1,b)} \) (where \( b - a - 1 \geq \alpha \)) are found to be correlated. Therefore, all subintervals starting from length \( \alpha \) to \( b - a - 1 \) are bound to be correlated due to this bottom-up style of enumeration. This addresses the second scenario.

We now prove the second part.

This can be proved by contradiction. Let us assume that there is a correlated interval \( I_{(a,b)} \) and is not enumerated. The bottom-up approach estimates the correlation of all intervals of length \( \alpha \) and enumerates those that satisfy the correlation threshold. Therefore, \( I_{(a,b)} \) cannot be of length \( \alpha \). For any bigger intervals \( I_{(a,b)} \) (\( l_{(a,b)} > \alpha \)) that is not enumerated, it is impossible that a subinterval \( I_{(a',b')} \), \( l_{(a',b')} \geq \alpha \) does not satisfy the correlation threshold. Therefore, \( I_{(a,b)} \) is not a correlated interval as it defies the third constraint and its contricts the original assumption. Therefore, all correlated intervals are enumerated by the algorithm.

D. Generalizations

LAMINA is defined using correlation measure to assess similarity between two time series in a given interval. Note that no relationship exists between the correlation of an interval and its subinterval. Based on this observation, BottomUpIntervalEnumeration approach (Alg. 4) is designed to ensure that all subintervals of an interval are highly correlated to determine if an interval is a maximal correlated interval. As this approach does not rely on any special property that correlation holds, any other similarity measures such as a euclidean distance or a cosine measure can also be used here. Specifically lines 6 and 12 in Algorithm 4 can be replaced with any similarity measure of choice and the algorithm by design finds maximal intervals based on that similarity measure.

The notion of a maximal correlated interval is defined based on the constraint that the correlation of the time series in the interval, its subintervals and none of its immediate subsuming intervals has to exceed a given threshold. This allows us to find intervals where the time series are highly correlated. Similarly, one may be interested in finding maximal intervals where a time series is negatively correlated (say \( r \leq -0.8 \)) or poorly correlated (say \(-0.2 \leq r \leq 0.2 \)). By design the BottomUpIntervalEnumeration (Alg. 4) approach ensures that the subintervals meet a given correlation threshold. Therefore, one can choose to use any criteria in this approach and it will discover maximal intervals that meet the criteria. For example, maximal intervals that exhibit negative (or weak) correlation can be discovered by changing the nature of the constraint in the BottomUpIntervalEnumeration approach (Line 7 and 13 of Algorithm 4) to capture negative (or weak) correlations.

V. Evaluation and Results

In this section we present an evaluation of the proposed approach on synthetic and real datasets. We compared the efficiency of the proposed approach with a brute force approach. Using a real world neuroimaging dataset, we show the utility of our approach in discovering intervals of high similarity and the use of these intervals in characterizing transient relationships.

A. Efficiency comparison

In order to study the efficiency of the proposed bottom-up approach in comparison to the brute force approach we generated synthetic datasets with varying lengths of correlated intervals.

Synthetic data

We first created two vectors of length 1500 whose values are sampled from a uniform distribution with range [0 1]. Each vector is now smoothed by computing each value as the average of preceding 5 values and succeeding 5 values. Following this smoothing the two vectors are now treated as a time series with temporal auto-correlation, i.e., consecutive values in a time series often have highly similar values. We marked the duration of the time series with consecutive intervals of length that is sampled uniformly between 41 and 80 time points. Starting with the first interval every alternative interval is treated as a ‘synchronous’ interval and the remaining intervals are treated as ‘asynchronous’ intervals. For the ‘synchronous’ intervals the values in the first time series are copied to the second time series by adding a small amount of Gaussian noise.
\( (\mu = 0, \sigma = 0.01) \). We then created 100 such pairs of synthetic time series where synchronous intervals are of length between 41 and 80. We refer to this set as \( TS_{41-80} \). Using the same approach we created two sets with 100 pairs of time series with synchronous intervals of length from 101 to 140 and 161 to 200, respectively. We refer to these sets as \( TS_{101-140} \) and \( TS_{161-200} \), respectively. These datasets where each set has different lengths of synchronous intervals will be useful in studying the impact of correlated interval lengths on the performance of the proposed approach.

**Parameter choices**

We used three different parameter choices for interval length: \( \alpha = [20, 30, 40] \). We did not use an \( \alpha \) that is longer than 40 as the smallest imputed interval is of length 41. We used \( \beta = 0.8 \). We also varied the length of the time series used as input by starting from the first point and ending at several points including \( \{100, 300, 500, \ldots, 1500\} \) to study the impact of length of time series on the time taken by the algorithms.

Both the brute force and the bottom-up approaches were implemented in Matlab\textsuperscript{®} and were executed on a node with 15 Xeon 2.40GHz processors and 100GB of main memory. Nevertheless, our implementation does not use more than one processor at a time.

**Observations**

The comparison of the time taken to discover the longest set of non-overlapping maximal correlated intervals is shown in Figure 5. X-axis in this figure shows the length of the time series used and Y-axis indicates time in seconds. Note that Y-axis in the figure is in logarithmic scale. As one would expect, the amount of time taken increases dramatically with increase in the length of the time series. From Figure 5 it can be seen that, in general, the brute force approach takes at least 10 fold more time to discover the LAMINA. This is due to the key difference in the two approaches that is bottom-up enumeration versus exploring all possible intervals. Additionally brute force approach has to prune intervals whose sub-intervals are not correlated.

The time taken for the brute force approach on different datasets \( (TS_{41-80}, TS_{101-140} \text{ and } TS_{161-200}) \) and for different choices of \( \alpha \) followed a very similar trend. On the other hand, the amount of time taken for the bottom-up approach on \( TS_{161-200} \) is more than that of \( TS_{101-140} \) and the time taken on \( TS_{101-140} \) is more than that of \( TS_{41-80} \). Therefore, the bottom-up approach is faster when the size of correlated intervals is shorter. This is because the bottom-up approach need not take into account the longer intervals.

Figure 5 also shows the impact of varying \( \alpha \). On \( TS_{41-80} \) dataset, as the length of the smallest interval is increased from 20 to 40, the overall time taken has reduced marginally. This change was relatively small for \( TS_{101-140} \) and was much smaller for \( TS_{161-200} \). This is due to the fact that the number possible intervals that need to be considered decreases with the increase of the valid interval length.

**B. Case study: Neuroimaging data**

We now show the utility of our proposed formulation on real world neuroimaging dataset.

**Data.**

Functional Magnetic Resonance Image (fMRI) data measures the amount of oxygen absorbed by gray matter tissue at every tiny cubic location in the brain (referred to as a voxel), at every time instant during the scan. The amount of oxygen absorbed at a given voxel and at a given time point is known to indicate the amount of activity occurring at the voxel. Data from an fMRI scan can be represented as a set of time series, one for every voxel. This can be represented in the form of a voxel\times time matrix, where every \( ij^{th} \) element in the matrix indicates the amount of neuronal activity occurring at a location represented by voxel \( i \) and at a time point \( j \). We used the dataset from [16] that contains 10 five minute resting state fMRI scans from one healthy subject obtained during one visit on a day. We append all these scans to get one 50 minute resting state scan. We refer to this dataset as rest dataset. The spatial resolution of each fMRI scan was \( 3\times3\times3 \) mm and the temporal resolution was 2 seconds. Several prepossessing steps have been performed on the data obtained from the scanner and they have been elaborately discussed in [16]. In addition, following the approach in [17], global mean time series is regressed from the data, as is done in most fMRI studies. The result voxel\times time matrix for each scan was of dimensions 47, 640 \times 1550. We further grouped voxels into 90 brain regions based on an anatomical atlas provided by [18]. We refer an interested reader to Table 2 in [19] for a list of these regions. The resultant matrix, \( D_{\text{rest}} \), was of size 90 \times 1550.

We also used the 10 five minute fMRI scans obtained from the same subject as above while the subject was watching cartoons [16]. This data allows us to study the differences that occur in the brain between rest and task. We refer to this data as cartoons data. The data was processed similarly to that of resting state data. The resultant matrix, \( D_{\text{cartoons}} \) was also of size 90 \times 1550.

**Approach**

We used our bottom-up approach to discover the longest set of non-overlapping maximal correlated intervals between every pair of brain regions in \( D_{\text{rest}} \). There were a total of 4005 \( \left(\frac{90}{2}\right) \) such pairs. Our approach was run to identify the set of
longest non-overlapping maximal correlated intervals for each of the 4005 pairs. We used parameters $\alpha = 25$ (minimum interval length) and $\beta = 0.7$ (correlation threshold). The reason for a relatively relaxed $\beta$ compared to the one used with synthetic datasets is that fMRI data is often noisy and a relaxed correlation threshold allows us to capture potentially relevant intervals.

We repeated this analysis on $D_{\text{cartoons}}$ using the same parameters. We then characterized the intervals captured on $D_{\text{rest}}$ and relate them with our findings on $D_{\text{cartoons}}$.

**Observations.**

- **Comparing LAMINA in rest and cartoons data** The correlated intervals in LAMINA that are discovered using our approach enables us to study the difference in overall duration of correlated intervals for a region pair in rest and cartoons data. For example, a pair of brain regions could be correlated for a small number of intervals in rest and they could be correlated for many more intervals in cartoons data. One such example where a dramatic difference is seen between rest and cartoons data is shown in Figure 6. This figure shows the correlated intervals found between brain regions 51 and 55 in rest and cartoons data with the help of double sided arrows where the left and right arrows indicate the start and end points, respectively. The total duration of these intervals in rest is 241 and in cartoons it is 1105. This large difference in the amount of time the two time series are correlated suggests that these two regions exhibit synergy more when the subject is watching cartoons than when the subject is resting. The time series from these two regions are shown in Figure 7 (only the first 500 time points are shown due to space limitation). The intervals discovered using our approach are indicated with double sided arrows as well as bold lines in time series. It is easy to see from this figure that when a region is not captured in a correlated interval, the time series from the two regions exhibit very different behavior.

- **Different type of intervals in rest and cartoons data** In addition to the differences in the total duration of intervals between brain regions there could exist more subtle differences in how two brain regions work synergistically. For instance, two brain regions could behave similarly for longer intervals on average or could only behave similar for shorter intervals. These differences can be investigated by comparing the average length of intervals when their total duration of intervals is similar.
Fig. 8. Correlated intervals for regions 3 and 7, as well as other region pairs that have a similar total length of correlated intervals. Intervals indicated in black have a correlation $\geq 0.7$ and those in green have a correlation $\leq 0.5$.

Fig. 9. Correlated intervals for regions 3 and 7 while the subject is resting and while watching cartoons. Intervals indicated in black have a correlation $\geq 0.7$ and those in green have a correlation $\leq 0.5$.

An example of this type of scenario is shown in Figure 8. Here the black double sided arrows indicate the correlated intervals discovered using our approach. For the brain regions (3, 7), there are three long correlated intervals [79, 471], [1032, 1388], and [1389, 1550] of duration 393, 357, and 162, respectively. The total duration of these intervals is 912. We chose six other region pairs (58, 64), (69, 70), (46, 47), (7, 11), (25, 32), and (8, 62) whose total length of correlated intervals are similar to that of (3, 7) for comparison. Their total duration of intervals are 906, 908, 910, 913, 913, and 915, respectively. The longest correlated interval in these six different brain regions is 113 and the mean of all intervals in them is approximately 40.8, whereas the shortest correlated interval in (3, 7) is 162. Therefore the intervals in these six region pairs are much shorter than those of (3, 7).

The intervals indicated in black in Figure 8 are bound to have correlation $\geq 0.7$, however it is possible that between two successive intervals the correlation may be slightly smaller than 0.7 and they are split because of the choice of the threshold. Before concluding that there is a significant difference in average length of correlated intervals in LAMINA for the above pairs, it is necessary to ensure that these splits are due to a significant reduction in correlation and not due to the choice of $\beta$ threshold. One way to achieve this is to check if there are intervals of significantly lower correlations between two correlated intervals in a LAMINA. Our approach which can find maximal intervals with correlation above a threshold $\beta$ can also be used to find maximal intervals with correlation $< \beta'$ using the condition $r < \beta'$ in step 13 of Algorithm 4. We used this variant of our approach with $\beta' = 0.5$ to find maximal intervals with correlations $< 0.5$. The resultant intervals are shown in green in Figure 8. If every consecutive pair of correlated intervals shown in black are separated by a green interval it guarantees that a split was due to a significant change in correlation and was not due to the choice of $\beta$. Almost all of the consecutive pairs of black intervals in Figure 8 have green intervals between them. This provides support for our argument that indeed the intervals are correlated for longer duration in (3, 7) than the other pairs listed above.

The dramatic difference in the duration of intervals in (3, 7) compared to that of the other region pairs suggest a difference in operating principles for these two regions. These regions tend to work in a synergistic fashion for longer periods.

We also compared the duration of intervals for the pairs of regions (3, 7) between rest and cartoons datasets. This comparison is shown in Figure 9. The length of intervals in cartoons data varies from 25 to 207 with a mean of 77.7, approximately. While the length of intervals in rest data varies from 162 to 393 with a mean of 304. Interestingly, the longest interval (207) found in cartoons for the pair (3, 7) is smaller than the mean of the length of correlated intervals (304) in rest data. It is interesting to note that the total duration of intervals is 912 in rest and 1243 in cartoons data. Despite having correlated intervals for approximately 33% longer in cartoons data than in rest data, the average length of intervals in cartoons is smaller than the smallest interval length in rest data.

This indicates that these two brain regions (3 and 7) operate differently when the subject is resting and when the subject is watching cartoons. While the subject is resting the two regions work synergistically for longer periods than when cartoons are being watched. The time series from these two regions are shown in Figure 10 (only the first 500 time points are shown due to space limitation). The intervals discovered using our approach are indicated with double sided arrows as well as bold lines in time series. Based on these observations we claim that our approach enables one to characterize the
dynamic behavior between brain regions based on the duration of individual maximal correlated intervals.

VI. CONCLUSION AND FUTURE WORK

In this paper we presented a novel formulation to capture distinct maximal correlated intervals in a pair of time series. We proposed an efficient bottom-up approach to discover the longest set of such intervals. We evaluated our approach on synthetic datasets to demonstrate its efficiency and its effectiveness. We showed the utility of the proposed approach on a real-world neuroimaging data. We computed non-overlapping maximal correlated intervals for each pair of brain regions and studied the difference in behavior in resting state and while watching cartoons. We also studied the change in the longevity of the intervals between resting state and watching cartoons. We found pairs of brain regions that are involved in interpreting visual stimulus to be correlated for many more intervals while watching cartoons than in resting state. We also found novel pairs of brain regions that are synchronized for a longer duration than their equivalent pairs. Discovering such interesting insights would not have been possible without our definition of maximal correlated intervals and the proposed approaches to efficiently discover them.

A number of aspects of this problem need to be studied further. i) Real world time series datasets are often noisy and due to this an originally long correlated interval could be discovered as multiple disjoint correlated intervals. Error tolerant techniques for discovering correlated intervals need to be explored. ii) Often general trends in the data may result in intervals that do not reflect synergy between the two entities from which the time series are obtained. For example, the mortgage crisis in 2008 affected a majority of the stocks in a similar fashion and so correlated intervals that cover this period are not interesting in the context of any given pair of stocks. iii) The intervals that are discovered by our approach could be potentially used to cluster [7], classify [5] and discover patterns [21] in time series data. The suitability of LAMINA for these problems needs to be studied.

Acknowledgements We thank Dr. Jeffrey S. Anderson for providing us the fMRl data set. This work was supported by NSF Grant IIS-1355072.

REFERENCES


