УДК 004.021, 004.052
DOI: 10.20535/SRIT.2308-8893.2017.3.12

METRIC AND ALGORITHM FOR SIMILARITY BETWEEN TWO TEMPORAL EVENT SEQUENCES CALCULATION

S. NIKOLAIEV

Abstract. In several data analysis applications on temporal events flows, the problem of measuring "similarity" of these sequences arises. There are many different definitions of event sequences but in this paper under the term "sequence of events" the ordered array of the event occurrence times will be understood. In this paper the metric and procedure for similarity calculation between two ordered event sequences is presented. The procedure as the output returns measure of two event flows similarity and set of corresponding indices pairs which represent the mapping of the events between the input sequences.

Keywords: time stamped quasi periodic events; temporal event sequences similarity metric; computing distance between two real arrays of different length; accuracy, recall, precision for temporal event sequences; algorithm for distance of two event flows estimation.

INTRODUCTION

Temporal event data is at the forefront of today's Big Data boom. Across the world, sensors from IoT, telecommunication networks, medical equipment, gadgets, video cameras are capturing and storing every aspect of our existence as sequences of time stamped events. And all these amounts of information with temporal events series are analyzed and mined for different kind of knowledge by a broad variety of specialists. Nowadays temporal sequence analysis is being used across multiple domains starting with sociology [1] where individual human lives are considered as sequences of events to be compared to pool of all known life-trajectories of society, and finishing with healthcare applications where events are extracted from quasi periodic bio-signals.

In all these domains the task of measuring similarity between two temporal event sequences of different length arises. This problem often is named finding similar situations in events flows.

More formally, the problem of finding similar situations can be defined as follows. Given a time t and a window width w, find another time s such that the windows of the sequence occurring in intervals $(t-w, t)$ and $(s-w, s)$ are similar. The similarity between two slices can be defined using an edit distance notion [2], i.e., the distance is defined as the cost of the cheapest possible sequence of operations that transforms one slice to another. Usually the operations are insertion and deletion of an event and moving an event in time; each operation has an associated cost. The edit distance can be computed using a dynamic programming algorithm, but the computation is slow. Furthermore, assigning costs to the edit operations is quite problematic [3].

This works well for applications with multiple types of events where subsequences can be distinguished by type of the events. But if we have single-event-type domain then approach of finding longest common subsequence becomes not viable because event flows are represented as real-valued times series. So in general case we find zero or a few "common" elements between two sequences using direct comparison despite there are a lot of intuitively "close" elements. Single-type quasi periodic event temporal sequences comparison also begets problem of quasi periodic error function which complicates finding global error minimum and as the result computing the time shift between the sequences.

In domains like healthcare the original sequence of bio-signals events sometimes cannot be observed or measured directly because of different factors so often approximation models are used instead. Proposed similarity metric is needed, e.g., for building and tuning parametric models that produce as output an approximation of some temporal process in the form of event flow. It is supposed that the etalon sequence of events, generated by the process, can be measured only during experimental phase and therefore the problem of the model tuning can be formulated in terms of supervised learning. The specific of this point of view is that the target sequence and the output of the model are two finite real series of not equal length. To run any optimization procedures on two arrays of different length, metric is needed to calculate similarity between these arrays. Also taking into the account the fact that approximation procedures may work for millions of iterations on some domains and model simulation can take a lot of time, proposed similarity metric should be calculated in linear time from the length of the input arrays.

Further in this paper the metric and the algorithm for computing similarity of two single-type quasi periodic event temporal sequences is proposed. This algorithm and the similarity metric were developed for optimization and supervised learning tasks where target variable is a temporal array of not fixed length.

PROBLEM FORMULATION

Suppose we have a process P that generates one type of event with non-constant interval. All event occurrences in some predefined time diapason can be put into ordered sequence of time stamps.

Two sequences of events A and B are given:

$$A = \{t_{a,i}\}_{i=1..K}, \quad B = \{t_{b,i}\}_{i=1..M},$$

where $t_{a,i}$ is time of i^{th} event occurrence in A and $t_{b,i}$ is time of i^{th} event occurrence in B. K and M correspond to lengths of A and B.

It is known that the finite length sequence A is the target sequence of events obtained by the direct event detection from the process during experiment and times $t_{a,i}$ in the sequence have zero lag.

A method exists that gets as input some derivative signal generated by the process P and makes approximations of $t_{a,i}$ The sequence B is obtained as an output of this parametric method. The lengths of A and B usually do not match

and for the same event its occurrence times $t_{a,i}$ and $t_{b,i}$ in A and B are shifted for varying value.

The goal is to create metric and algorithm for A and B similarity evaluation.

METRIC REQUIREMENTS

The metric equals to one for the zero-error method. In this case B sequence should coincide with the sequence A:

1) length of A equals to length of B;

2) $t_{a,i}$ $t_{b,i}$ for all *i* in I.

If the metric is less than one but greater than zero then either lengths of the sequences do not match or some of times $t_{b,i}$ differ from $t_{a,i}$. The lower the similarity between A and B, the smaller metric value.

The metric value cannot be lower than zero or be greater than one.

DEFINITIONS

Let's call the event $z \in B$ "normal" if it "matches" to the event $z \in A$. We denote the set of normal events through Z.

The event $z \in A$ is called "missed" in B, if this event is not present in B.

The event $z \in B$ is called "odd" (erroneous or false alarm), if this event is not present in A.

In general $K \neq M$. This means that there are either odd events or missed ones. It is also known that the times of the same event occurrence $t_{a,i}$ and $t_{b,i}$ in the sequences A and B may be shifted by a random value δ_i elative to each other (Fig. 1 below)

$$|\delta_i| < |t_{a,i} - t_{b,i}|; \quad t_{a,i} = t_{b,i} - \delta_i .$$

Example of the sequences is shown in the fig. 1 below:

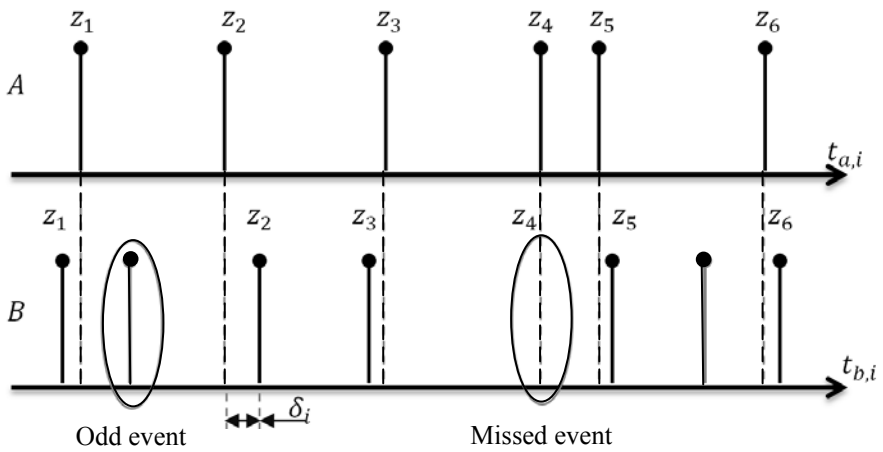

Fig 1. Example of event sequences A and B

In the sequence B the second and penultimate events are odd ones. z_4 event from A is missing in B.

DEFINITION OF "MATCHING EVENT" IN A AND B

It is necessary to introduce the concept of matching events $t_{a,i} \sim t_{b,j} = z$ in two given sequences of real numbers A and B where A is the target sequence of events.

The event $\boldsymbol{t_{b,j}}$ is called **matching (~) to the event** $\boldsymbol{t_{a,i} = z}$ if:

a) $t_{b,j}$ has a minimum distance δ_i to the event $t_{a,i}$ by comparison with the distances to $t_{a,i-1}$ and to $t_{a,i+1}$, and

b) there are no other events $t_{b,k}$ for which the distance to $t_{a,i}$ would be less than δ_i.

$$\text{if } i = \arg\min_{i\in[1..\text{Card}(A)]} \left| t_{a,i} - t_{b,j} \right|, \text{ and}$$

$$\delta_i = \left| t_{a,i} - t_{b,j} \right| = \min_{k\in[1..\text{Card}(B)]} \left| t_{a,i} - t_{b,k} \right|.$$

$$\text{then } t_{b,j} \sim t_{a,i} = z_i$$

If there are two events $t_{b,j}$ and $t_{b,j+1}$ with equal distances to the event $t_{a,i}$, then the event $t_{b,j}$ will be considered as matching to event $t_{a,i}$.

The fig. 2 shows how target events from sequence A match to approximated events in sequence B. White and shaded areas indicate membership time intervals that correspond to target events z_i from A. All events from B that are present in the time interval corresponding to event z_i are considered to be candidates for normal event.

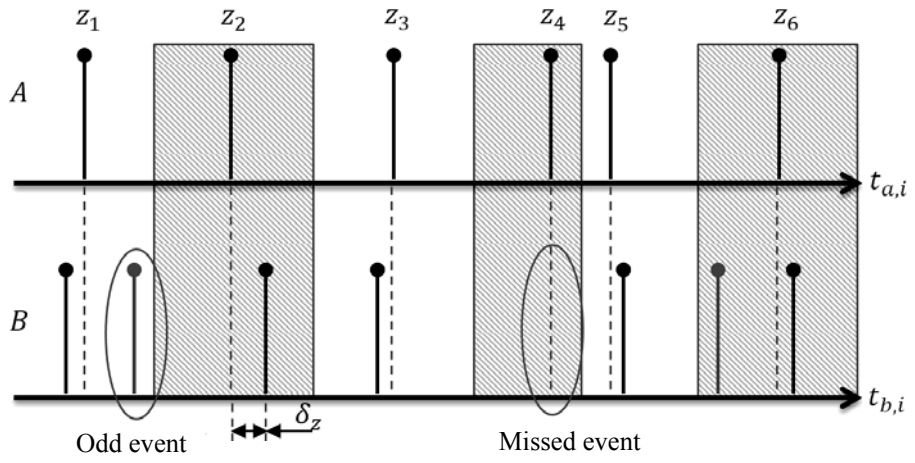

Fig 2. Correspondence of events in *A* to events in sequence *B*

DERIVATION OF SIMILARITY METRIC FOR EVENT SEQUENCES

Let $K = M$ and assume that B has no pairs of false events identifications of different types (for example, odd and missed event at the same time) in the entire interval under consideration so that A and B contain only normal events.

In this case, A and B can be treated as two finite real vectors with the same dimensions, where the i^{th} element in each of the vectors corresponds to the i^{th} event. For two real valued vectors with equal lengths a lot of distance and (dis)similarity metrics are available. As an example generalized distance metric can be treated as error or dissimilarity metric:

$$d = \sqrt[m]{\frac{1}{M}\sum_{i=1}^{M}\left|t_{a,i} - t_{b,i}\right|^{k}} .$$

Depending on m and k we obtain the formulas for calculating arithmetic mean, average and standard deviation.

For normal events from B we define the root mean square error in the form of:

$$RMSE_Z = \sqrt{\frac{\sum_{z \epsilon Z} \left|\delta_z\right|^2}{\mathrm{Card}(Z)}},$$

where $\mathrm{Card}(Z)$ is the cardinality of the set Z and it shows the number of normal events in Z. The smaller the RMSE metric value, the more accurate the method.

If $K \neq M$ then at least one missed or one odd event is present in B.

In this case false positives and event omissions should be separated, and then metrics from the theory of classification can be applied. The difference between known metrics and proposed metric is that the theory of classification usually requires the presence of at least two classes of objects and the task is to determine to which of these classes object belongs. But in our case, there is only one class of the event and the method decides in which moments the event occurs so there may be missed events or multiple detections of false alarms. This mainly happens because the method uses indirect evidences to determine sequence B without knowing anything about sequence A.

Suppose we know the true separation of events for B (i.e., we know which of the events are normal, false positives and missing). Then the following metrics can be applied to the sequences of events B and A:

K — total number of events in A;

TP (true positives) $\in [0..K)$ — the number of normal (coincident) events in B with respect to A. ($TP = \mathrm{Card}(Z)$)

FP (false positives) $\in [0..+\infty)$ — the number of odd events in B (false positives detected by the method). Each event from A can have a lot of false positives in B. (Type I error);

FN (false negatives) $\in [0..K)$ — the number of missed events. (type II error);

Let Acc be the accuracy of finding the set of normal events:

$$Acc = \frac{TP}{TP + FP + FN} = \frac{TP}{K + FN} \in [0, 1] .$$

The closer Acc to one, the less erroneous detections of events are present in B.

Let R (Recall) be the percentage of normal events among all target events.

$$R = \frac{TP}{TP + FP} = \frac{TP}{K} \in [0, 1].$$

The closer this metric to one, the more precise method.

$$Acc \le R.$$

Summarizing the cases described above, we obtain the metric for evaluating the quality of the method on the given two sequences A and B:

$$\mathrm{Q} = \frac{Acc}{1 + \alpha\, RMSE_Z} \in [0, 1] \to \max.$$

where $\alpha \in [0\,..+\infty)$ is weight, which determines the prevalence of normal events local shifts error $RMSE_Z$ over the accuracy of finding the set of normal events Z. The higher α the more important becomes precise localization of normal events from B in time. The concrete value of α is calculated for a particular pair of sequences A and B, depending on objectives of the study.

The higher Q the better sequence B approximates the sequence A. This criterion reaches its maximum value of one for two identical sequences of A and B.

ALGORITHM FOR THE Q METRIC CALCULATION

In previous chapter the formula for similarity metric Q between two sequences of events A and B was introduces. The cornerstone of this metric is the algorithm for defining the sets of normal, missed and odd events.

The basic idea of the algorithm is to find the right ($b_{i,\text{right}}$) and left ($b_{i,\text{left}}$) borders or margins for each element $t_{a,i}$ from A and in the resulting interval between these boundaries determine the nearest event $t_{b,j}$ from B. Each of the borders is calculated as the middle of the intervals between adjacent with $t_{a,i}$ events $t_{a,i-1}$ and $t_{a,i+1}$

$$b_{i,\text{ left}} = \frac{t_{a,i} + t_{a,i-1}}{2},\ 0 < i < K - 1.$$

$$b_{i,\text{right}} = \frac{t_{a,i} + t_{a,i+1}}{2}, 0 < i < K - 1$$

In case $i = 0$ instead of using not existent $t_{a,-1}$, $b_{0,\text{left}}$ is passed to algorithm as input parameter. The same applies to $b_{K,\text{right}}$.

This algorithm after performing calculation on given input sequences A and B, returns three arrays: two arrays with indices of missed and odd events in B and one array with pairs of indices of corresponding normal events. From the third array δ_i are easily calculated. Further the metric Q is calculated according to the formula.

Below the algorithm in Python is presented:

```
def detect_tp(a, b, left_margin=1, right_margin=1):
   miss = [] # Declaration of an empty array
   odd = []
   ijs = []
   j = 0
   for i in range(len(a)):
     b_candidates = []
     l_marg, r_marg = get_margins(a, i, left_margin,
     right_margin)
     # collect odd js
     while len(b) > j and b[j] < l_marg:
       odd.append(j)
       j + = 1
     # collect candidate js
     while len(b) > j and l_marg <= b[j] < r_marg:
       b_candidates.append(j)
       j += 1
     # process collected js
     if len(b_candidates) == 0:
       miss.append(i)
     elif len(b_candidates) == 1
       ijs.append((i, b_candidates[0]))
     else:
       jj = find_best_j_match(b_candidates, a[i], b)
       for k in b_candidates:
           if k != jj:
             odd.append(k)
           else:
             ijs.append((i,   jj))
      # collect rest bs that are further than the right
        margin of the last a[-1]
      while len(b) > j:
        odd.append(j)
        j += 1
      return ijs, miss, odd
```

Function **detect_tp** takes two arrays A and B, as well as the left and right default borders as input parameters **left_margin** and **right_margin** for the first and last events in the sequences. After completion, this function returns three sequences: ijs, miss and odd, where ijs is a sequence of pairs (i, j) of indices events $t_{a,i} \in A$ and $t_{b,j} \in B$ where $t_{b,j} \sim t_{a,i} = z_i$.

This function also calls function **get_margins** that calculates margins for i^{th} event $t_{a,i} \in A$

```
def get_margins(a, i, left_margin, right_margin):
   # calculate left margin
   if i == 0 or len(a) <= 1:
     if len(a) == 0:
```

```
      l_marg = 0
    else:
      l_marg = a[i] - left_margin
  else: # i>0
    l_marg = (a[i] + a[i - 1]) / 2.0
  # calculate right margin
  if i == len(a) - 1 or len(a) <= 1:
    if len(a) == 0:
      r_marg = 0
    else:
      r_marg = a[i] + right_margin
  else:
    r_marg = (a[i] + a[i + 1]) / 2.0
  return l_marg, r_marg
```

The function named **find_best_j_match** is designed for finding the best candidate in array of proposed candidates **b_candidates** from B for given event $t_{a,i}$. It returns the index for the closest event among **b_candidates** to a given event **av** from A.

```
def find_best_j_match(b_candidates, av, b):
  minj   =   b_candidates[0]
  for   k   in   b_candidates:
    if abs(av - b[minj]) > abs(av  - b[k]):
      minj = k
  return minj
```

CONCLUSION

In this paper the metric and the algorithm for computing similarity of two single-type quasi periodic event temporal sequences was described. The procedure as output returns sequence similarity metric and set of corresponding indices pairs of given occurrences of the event showing its most probable location in each of input arrays. The algorithm complexity is $O(M + K)$ where M and K are lengths of input sequences.

Successful modification of existing approaches on dissimilarity metrics over vectors with equal dimensions and their combination with classification metrics for events can be concluded. This process as the result led to creation of single metric for similarity calculation of different length temporal sequences.

The experiments show that the method is performing quite good at separating noisy event occurrences from normal event flow. The described metric Q for calculating similarity between two event series of different length returns intuitively plausible results.

The algorithm for events classification into normal, odd and false positives types works best when the time shift between input sequences is in range of one period of event occurrence. The experiments conducted show that the shift can be automatically minimized by iteratively applying the method of gradient descent using developed metric. The developed metric can be used in single-event

multiple-sequence analysis (MSA) applications including but not limited to optimization and supervised-learning problems.

Further the author plans to research application of proposed metric and algorithm for extracting heart-beats intervals from video.

REFERENCES

1. *Pollock Gary*. Holistic trajectories: a study of combined employment, housing and family careers by using multiple-sequence analysis. *Journal of the Royal Statistical Society*, *Series A* (*Statistics in Society*), 170(1): P. 167–183, 2007.
2. *Mannila H., Moen P*. Similarity between event types in sequences, Proc. First Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK'99), Florence, Italy, 1999, P. 271–280.
3. *Moen P*., Attribute, Event Sequence, and Event Type Similarity Notions for Data Mining, Ph.D. thesis, Department of Computer Science, University of Helsinki, Finland, 2000.

Received 29.11.2016