# The TIMERS II Algorithm for the Discovery of Causality

Howard J. Hamilton and Kamran Karimi

Department of Computer Science University of Regina Regina, Saskatchewan Canada S4S 0A2

{hamilton, karimi}@cs.uregina.ca

**Abstract.** We present the Temporal Investigation Method for Enregistered Record Sequences II (TIMERS II), which can be used to classify the relationship between a decision attribute and a number of condition attributes as instantaneous, causal, or acausal. In this paper we consider it possible to refer to both previous and next values of attributes in temporal rules, and thus enhance the definition of acausality. We also present a new algorithm for distinguishing between causality and acausality.

#### 1. Introduction

In this paper we present the Temporal Investigation Method for Enregistered Record Sequences II (TIMERS II), which can be used to classify the relationship between a decision attribute and a number of condition attributes as instantaneous, causal, or acausal.

Instantaneous rules are normal decision rules. An example rule is: if  $\{(\text{Outlook}_t = \text{sunny}) \text{ AND (Temperature}_t > 20)\}$  then  $(\text{Play}_t = \text{yes})$ , where t indicates the time step of observing the attribute's value. For causality and acausality, the results are temporal decision rules. For the causal case, the decision attribute's value is causally determined by the condition attributes, whose values all appear in the past relative to the decision attribute. An example is: If  $\{(\text{outlook}_{t-1} = \text{sunny}) \text{ then (outlook}_t = \text{sunny}) \text{.}$  The index t-1 indicates that the attribute's value is seen in the previous time step.

For an acausal relationship, values at time steps bigger than t are used in the process of predicting the decision attribute at time t. In TIMERS II it is also possible for some condition attributes to have happened in the past. An example acausal rule is: if {(outlook<sub>t-1</sub> = overcast) AND (outlook<sub>t+1</sub> = rainy) then (outlook<sub>t</sub> = rainy). In an acausal relation, the decision attribute's value is not caused by the condition attributes, but just happens to be seen together over time. In this case there may have been hidden common causes that affected all the attributes in the same rule. The same method can be used for linear spatial data, where "back" and "forward" can be used to indicate the relative position of an attribute's observation.

The formal definitions of instantaneous and causal sets of rules are given in [2]. In TIMERS II a set of rules is acasual if the current value of the decision attribute relies on the future value of at least one condition attribute [3].

The rest of this paper is organised as follows. Section 2 introduces the TIMERS II algorithm. Section 3 presents a number of experimental results obtained from TIMERS II. Section 4 concludes the paper.

### 2. The TIMERS II Algorithm

We consider there to be an *order of conceptual simplicity* among the three types of the relations, with instantaneous being the simplest type of relationship, followed by acausality, and then causality being the most complex. Hence, instantaneous <<sub>simplicity</sub> acausal <<sub>simplicity</sub> causal. The intuition behind this ordering is that as we move from instantaneous to acausal and then to causal, more claims are being made about the relationship. As a principle, we try to explain a relationship with the simplest possible type. As we will see in Section 3, this ordering is used to choose a winning relations type when the results of the three tests are close.

Temporalisation was introduced in [2]. TIMERS II introduces the *sliding position* temporalisation as explained in [3]. The temporalisation operator  $Temporalise(w, pos, \mathbf{D}, d)$  takes as input a window size w, the position of the decision attribute within the window pos, the input records  $\mathbf{D}$ , and the decision attribute d, and outputs temporalised records. The TIMERS II algorithm is shown in Figure 1.

**Input:** A sequence of sequentially ordered data records D, minimum and maximum temporalisation window sizes  $\alpha$  and  $\beta$ , where  $0 < \alpha \le \beta$ , a minimum accuracy threshold  $ac_{th}$ , a decision attribute d, and a confidence level cl. The attribute d can be set to any of the observable attributes in the system, or the algorithm can be tried on all attributes in turn. *Preference* determines whether the user prefers higher accuracy or a simpler method.

**Output:** A set of accuracy values and a verdict as to the nature of the relationship among the decision attribute and the condition attributes. It could be spontaneous, causal, or acausal.

**RuleGenerator()** is a function that receives input records, generates decision trees, rules, or any other representation for predicting the decision attribute, and returns the training or predictive accuracy, as well as the size of the generated rules.

```
TIMERS II(D, \alpha, \beta, Ac_{th}, d, cl, preference) {
 ac_i = RuleGenerator(D, d); // instantaneous accuracy; window size = 1
  for (win = \alpha \text{ to } \beta)
        for (pos = 1 \text{ to } win)
           (ac_{w,pos}, ruleSize_{w,pos}) = RuleGenerator(Temporalise(win, pos, D, d), d)
  ac_c = \max(ac_{\alpha,\alpha}, ..., ac_{\beta,\beta}) // best causal result
  ac_a = \max(ac_{\alpha,pos1}, \ldots, ac_{\beta,pos2}), \ \forall \ ac_{x,pos}, \ 1 \le pos \le x \ // \ best acausal result
  if (\max(ac_i, ac_c, ac_a) < ac_{th}) then stop. // Maybe there is not enough related information?
  Verdict = "for attribute" + d + ", "
  Relation = RelationType(cl, (ac_i, ruleSize_i), (ac_a, ruleSize_a), (ac_c, ruleSize_c), preference)
  Case relation of
       INSTANTANEOUS: verdict += "the relation is instantaneous"
       ACAUSAL: verdict += "the relation is acausal" // an element from the future is present
       CAUSAL: verdict += "the relation is causal" // all condition attributes are from the past
  end case
  return verdict.
```

Fig. 1. TIMERS II algorithm for discovering the nature of a relationship.

TIMERS II has been implemented in an application programme called TimeSleuth [1]. TIMERS II first performs the instantaneous test. Since it may not be obvious which window size is most appropriate for a particular dataset, TIMERS II tries a range of

window sizes. The resulting temporalised data are fed to a rule generator which comes up with decision rules, and returns the accuracy and also the complexity of the rules. These measures are used to decide on a relation type. For an analysis of the time and space requirements of TIMERS II, see [3].

We use the accuracy and complexity of the rules obtained from each method to choose the best relation type that applies to the data. Normally the method with the highest accuracy value would be selected. However, it may happen that the accuracy values are close to each other. In such cases we choose the simpler relationship because the gains from choosing another relationship type may not be worth the extra complexity. Users can employ their discretion in making this decision. However, TIMERS II includes a statistical method. The RelationType() routine uses accuracy intervals to make a judgment about the type of the relationship. Using the confidence level provided by the user in the *cl* parameter, and assuming normal distribution, it constructs a confidence interval for the accuracy [4]. Then it checks to see if the corresponding intervals overlap. If they do, the method with the simpler type of relationship will be chosen provided it has simpler rules. The intuition is that even if the simpler method has resulted in less accuracy, it could have *potentially* produced better or the same results. After selecting a winner between the first two methods, the winning relation type is tested against the third relation type using similar comparison of intervals, to determine the final winner.

As an example, suppose with a confidence level of 90%, we have: the instantaneous accuracy  $ac_i = 32.5\%$ ,  $interval_{aci} = [31\%, 34\%]$ , the acausal accuracy  $ac_a = 35\%$ ,  $interval_{aca} = [33\%, 37\%]$ , and the causal accuracy  $ac_c = 37\%$ ,  $interval_{acc} = [35\%, 39\%]$ . For simplicity of the example we assume all methods have the same size of rules. Because the confidence intervals of the instantaneous method and the acausal methods intersect, instantaneous is chosen because it is considered simpler. Since the intervals of the instantaneous and causal methods do not overlap, the causal method is chosen as the final verdict because of its higher accuracy value. This example also shows the special case when the every two intervals are overlapping. In this case, starting with the first two or the last two methods give different results. In the first case, as shown above, we choose the method with the highest accuracy. But when starting from right to left (higher accuracy value to lower values) we choose the simplest method. We leave the decision about which direction to follow to the user. In the TimeSleuth programme the user can choose between "Prefer simpler method" (right to left) and "Prefer higher accuracy" (left to right) options. See [3] for more details.

Here is how the algorithm to choose a method works. To determine which method/relation type to choose, we sort the accuracy values in either ascending order (preferring higher accuracy), or descending order (preferring simpler method). This different ordering simplifies the algorithm, because we do not need to worry about the direction after this point. Starting with the two methods with the lowest (or highest) accuracy values, we test to see if there is an overlap among their confidence intervals. If so, then we choose the simpler method. The choice of the simpler method depends on both the conceptual complexity of the relation as defined above, and also the size of the rules that are needed to express the relationship. In our method the more space needed for

the rules, the more complex that relationship. We use the number of conjuncts in the rules to measure their size, as in the Minimum Description Length (MDL) principle. We make the decision as to which method to choose the following way: If a conceptually simpler method overlaps with a conceptually more complex method, but at the same time requires more space to represent the rules, then priority is given to the more complex rule. In other words, for a simpler method to over-ride a more complex method, not only should there be an overlap between their accuracy intervals, but the simpler method should result in fewer or shorter rules. While our assumed order of complexity is subjective, including the size of rules adds an objective element to the complexity measure. If there is no overlap in the accuracy intervals, we choose the method with the better accuracy value. A winner is thus selected among the first two methods. This winning relation type is then compared with the third method to determine the final method. Figure 2 shows how the best method is selected.

```
Input: A confidence level cl, three accuracy values corresponding to the instantaneous, acausal, and causal
methods: ac_i, ac_a, ac_c, and their corresponding size of rules: ruleSize_i, ruleSize_a, ruleSize_c, a preference p for
higher accuracy vs. a simpler method.
Output: A verdict as to the best relationship type.
//info[].method contains one of INSTANTANEOUS, CAUSAL, or ACSUAL. info[]. Accuracy is the best
//accuracy value. info[].interval contains the interval of the accuracy value, computed using a confidence value
Function RelationType(cl, (ac_i, ruleSize_i), (ac_a, ruleSize_a), (ac_c, ruleSize_c), p) {
  // initialise the info[] structure
  forEach (method = INSTANTANEOUS, ACAUSAL, CAUSAL)
     info[method] = (method, accuracy_{method}, ruleSize_{method}, Interval_{method} = (method)
                                                                 ComputeAccuracyInterval(accuracymethod))
  // if preference is given to higher accuracy, then start the search from lower accuracy values
  if (p = HIGHER ACCURACY)
    sort Ascending(info[]); // sort in ascending order of accuracy.
  else // SIMPLER_METHOD
    sort Descending(info[])
    winner = 1
    for (count = 2 \text{ to } 3)
      if (overlap(info[winner].interval, info[count].interval)) {// if overlap, then choose the simpler method
       if (info[count].method <<sub>simplicity</sub> info[winner].method and
          info[count].ruleSize \le info[winner].ruleSize) then
         winner = count
     else { // if no overlap, choose the method with higher accuracy
        if (info[count].accuracy > info[winner].accuracy) then
          winner = count
   return info[winner].method //one of INSTANTANEOUS, ACAUSAL, or CAUSAL
```

Fig. 2. Selecting the best relationship.

If needed, this algorithm can also select the best window size based on a number of accuracy values obtained in either the acausal or casual case. The order of simplicity is then determined by the window size, with bigger window sizes considered less simple. In

the TIMERS II algorithm in Figure 1, we use the use the window size that gives the maximum accuracy.

# 3. Experimental Results

We report on experiments using two temporal datasets. The first one is generated by an artificial life programme called URAL, and involves an artificial robot moving left, right, up and down on an  $8 \times 8$  board. The goal is for us to discover the effects of moving the robot. The position is expressed by a x and y pair. We used 2500 records for training, and 500 for testing the rules (to compute the predictive accuracy). This data comes from a controlled environment with no exceptions.

Each record in the robot dataset contains x and y position values at any given time and the direction of movement at that time. We set the decision attribute to be the current value of x, and the other three attributes are set as the condition attributes. There is no relationship between the current value of x on one hand, and the current values of y, direction of the movement, or the presence of food on the other hand. So we predict that an instantaneous test (window size of 1) will give poor results. From our understanding of the domain we know that the current value of x depends on the previous value of x, and the previous direction of movement. We expect the method to classify the relationship as a causal one. The acausal hypothesis says that you can tell where you were before if you know where you are now, and which direction you are will be going next. This hypothesis is clearly wrong, as we could have ended at the current position from a different number of previous positions. Hence we do not expect to get good results with our acausality test. The results are shown in Table 1. Even though an acausal method may have been used, the output rules may not have any references to attributes that appear after the decision attribute. In this case the rules are considered to be causal, seen in under "Actual rules."

Considering the result with a window size of 2, we declare the relation to be causal. With any position bigger than 1, the previous record which contains the relevant information for accurate prediction of current x value, is included in the temporalised data. The method discovers the correct temporal relation between the current value of x and the previous x and movement direction, with results having 100% accuracy with sliding positions of 2 or more. In other words, even with an acausal test, the rules are all causal, because they only contain attributes from the previous time step.

The second series of experiments concerns a real-world dataset from weather observations in Louisiana [5], and hence interpreting the dependencies and relationships is harder. It includes 343 training records, each containing the air temperature, the soil temperature, humidity, wind speed and direction, and solar radiation, gathered hourly. 38 other records were used for testing the rules and estimating predictive accuracy. We set the soil temperature to be the decision attribute. The results are shown in Table 1.

The relationship is not instantaneous, as observed by relatively poor results with a window size of 1 (instantaneous test). The accuracy goes up after temporalisation, implying that there is a temporal relationship at work. This relation is not causal, and the current value of the soil temperature just happens to change relative to its past values.

Since the accuracy values in causal and acausal tests are not much different, TimeSleuth declares the relationship between the soil temperature and other attributes to be acausal.

			Robot Data			Weather Data		
Win	Position	Type of test	Training Accuracy	Predictive Accuracy	Actual Rules	Training Accuracy	Predictive Accuracy	Actual Rules
1	1	Instant	19.7%	20.4%	Instant	27.7%	23.7%	Instant
2	1	Acausal	56.2	55.7%	Acausal	75.1%	59.5%	Acausal
2	2	Causal	100%	100%	Causal	82.7%	67.6%	Causal
3	1	Acausal	57.6%	55.6%	Acausal	85.3%	75.0%	Acausal
3	2	Acausal	100%	100%	Causal	82.4%	72.7%	Acausal
3	3	Causal	100%	100%	Causal	86.8%	77.8%	Causal
4	1	Acausal	58.4%	58.1%	Acausal	85.3%	74.3%	Acausal
4	2	Acausal	100%	100%	Causal	85.9%	74.3%	Acausal
4	3	Acausal	100%	100%	Causal	83.2%	74.3%	Acausal
4	4	Causal	100%	100%	Causal	84.4%	71.4%	Causal

Table 1. TIMERS II's accuracy result with the robot and weather data

## 4. Concluding Remarks

TIMERS II provides a method to discover and distinguish between instantaneous, causal, and acausal relationships between a decision attribute and a set of condition attributes. Our method is based on the passage of time between causes and effects. We generalised the ability to refer to attribute values from other time steps so that a rule can refer to condition attribute' values that appear before and after the decision attribute. This ability results in an enhancement to the definition of an acausal relationship. We also provided an algorithmic method of distinguished between instantaneous, causal, and acausal relations.

One can apply the same temporal considerations to associations, so the values of a number of attributes from different time steps can be associated together. However, in an association we do not have a distinguished decision attribute, observed at a reference time (the current time). So defining the future and the past may not be straightforward.

TimeSleuth is available from http://www.cs.uregina.ca/~karimi/downloads.html.

### References

- 1. Karimi, K., and Hamilton, H.J. TimeSleuth: A Tool for Discovering Causal and Temporal Rules, *The 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2002)*, Washington DC, November, 2002, pp. 375-380.
- Karimi, K., and Hamilton, H.J., Distinguishing Causal and Acausal Temporal Relations, *The Seventh Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'2003)*, Seoul, South Korea, April/May 2003, pp. 234-240.
- 3. Karimi, K., and Hamilton, H.J., An Extension to the TIMERS Method, *Technical Report CS-2005-02*, Department of Computer Science, University of Regina, Canada, March 2005.
- 4. Witten, I.A., and Frank, E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, 2000.
- 5. http://typhoon.bae.lsu.edu/datatabl/current/sugcurrh.html. Contents change with time.