ISSN: 0828-3494
ISBN: 0-7731-0516-6

An Extension to the TIMERS Method

Kamran Karimi and Howard J. Hamilton
Technical Report CS-2005-02
March, 2005

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

An Extension to the TIMERS Method

Kamran Karimi and Howard J. Hamilton
Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada S4S 0A2
{karimi, hamilton}@cs.uregina.ca

Abstract. In this paper we present TIMERS II (Temporal Investigation Method for Enregistered Record Sequences II). Assuming that the effects take time to manifest, TIMERS II merges the input records and brings the causes and effects together. The output is in the form of a set of decision rules. The condition attributes' values could have been observed in the past or the future relative to the decision attribute's value. In TIMERS II the past values can influence the present, thus establishing causality. But we consider it possible to reference observations that appear after the decision attribute, which forms the basis for acausality, or temporal co-occurrence. Three tests are performed using three different assumptions on the nature of the relationship. Each test results in a number of classification rules. The quality of the output rulesets determine if the decision attribute's value is best described by a non-temporal (instantaneous) relationship, or a temporal (causal or acausal) one. In previous work, TIMERS would allow referencing the values of condition attributes that were all observed either after or before the decision attribute, and so the rules followed a unique direction in time. In this paper, we consider it possible to reference both the past and the future values in the same rule to predict an attribute's value. We also present a new algorithm that uses the accuracy of the generated rulesets to determine whether the relationship between the decision attribute and the condition attributes is instantaneous, causal, or acasual.

1. Introduction

Discovering the existence of causal relations among attributes has been an active research field. Given a number of attributes, the input usually consists of records, each containing the values of these attributes observed together. An example would be the record <sunny, 25, yes> which contains one set of values of the attributes {outlook, temperature, play}. The prevalent approach is to consider the problem to be that of creating a graph, where the parent nodes denote causes, while the children denote effects. Conditional independence plays a great role in the construction of these causal graphs. The main techniques for discovering causal relations include learning Bayesian networks, which use conditional probability distributions in each node [1]. This probability-based approach is presented in [8], and TETRAD is a famous example of a causal discoverer that is based on this method [9]. Another programme for discovering causal relations uses the Minimum Message Length (MML) method to generate a Bayesian network. CaMML is an example of a causal discoverer based on this principle [6]. It measures the goodness-of-fit of a causal model to the data [12].

There are some common characteristics for the methods mentioned above. One is that they consider all the available attributes in the process of causal discovery. This means that they try to find causal relationships among all the variables. We have shown that this can result in very long execution times as the number of attributes increases [4]. The other common consideration is that the input records are considered independent of each other, and no assumptions are made as to when or where they may have been obtained. The records could have come from different sources and at different times. Assuming no temporal relationship among the records allows these approaches to work on many datasets.

The problem we are considering is to classify the relation between a distinguished decision attribute and a number of condition attributes as one of instantaneous, causal, or acausal. To solve this problem we present another framework, the Temporal Investigation Method for Enregistered Record Sequences II (TIMERS II), for causal discovery. TIMERS II assumes the passage of time between the input records, and differs from the other methods because: first, it does not try to create a graph of causal relations. Instead it focuses on the relationship between a decision attribute and the rest of the attributes. It is possible to run TIMERS II several times with a different target (decision) attribute each time, but the results are not to be combined into a graph. Second it assumes that the input records are temporally sorted and come from the same source. This temporal characteristic of the data is the basis for the justification of causal discovery in the presented method. While TIMERS II is fast and can handle many more attributes in the record than other methods [4], proper input is less widely available. However, when applicable, the results are meaningful, because with temporal decision rules the user can answer "what" is related to what, as well as "how." This makes the results useful in a data mining context, where the user can employ this method to gather insight about the data.

Causality's definition and characteristics are subject to considerable debate, with some believing it is an undiscoverable notion. Instead of entering mostly philosophical debates as to who is "more right," we consider all the approaches to the automatic discovery of causality, including our own, to be based on arbitrary definitions and methods.

This paper's contribution is in introducing the differences between TIMERS II and its predecessor. In addition to allowing both the past and the present values to be referenced in the same rule, TIMERS II incorporates a statistical method for recommending the best relation type between the decision attribute and the condition attribute.

Suppose we have gathered some data about the weather outlook, the temperature, and whether it was possible to play that day. The data for five consecutive days are in Table 1.

Day number	Outlook	Temperature	Play
1	Sunny	25	Yes
2	Rainy	13	No
3	Overcast	20	Yes
4	Rainy	10	No
5	Rainy	12	No

Table 1. Consecutive records observed once a day

The problem is to discover decision rules that predict when we can play. We can consider any row in Table 1 to be the "current" row and thus signifying the current day. Other records are then considered to have been observed in the past if they happen before the current row, or to have been observed in the future if they appear after the current row. The cornerstone of TIMERS II method is that we can use not only the past, but also the future observations to predict the present. Depending on whether there is a time difference between the decision attribute and the condition attributes, there are two broad categories for the relations. These categories are: atemporal, where there is no time delay between the decision attribute and the condition attribute (classical classification), and temporal, where the decision attribute's value appears at another time relative to the condition attributes.

There are three possible verdicts for a relationship in TIMERS II: instantaneous (which is atemporal), causal, and acausal (which are both temporal). In the instantaneous case by definition there are no temporal relationships at work, and the value of the target attribute is best determined by the values of the condition attributes as observed at the same time. The resulting rules are normal decision rules. An example such rule would be: if{(Outlook = sunny) AND (Temperature > 20)} then (Play = yes).

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, which all appear in the past relative to the target attribute. In other words, in a causal relationship the past predicts the future [10], which is the normal direction of time. An example would look like this: If {(outlook$_{current-1}$ = sunny) then (outlook$_{current}$ = sunny). We have added indices so that we can distinguish between the same attribute's value happening at different times. "Current-1" indicates that the attribute was seen in the previous time step, or in this case, yesterday.

For an acausal relationship, future values are used in the process of predicting the decision attribute at the present time [7]. For a relationship to be acausal at least one condition attribute should have been observed after the decision attribute. However, in TIMERS II it is also possible for some condition attributes to have happened in the past. An example acausal rule would be: if {(outlook$_{current-1}$ = overcast) AND (outlook$_{current+1}$ = rainy) then (outlook$_{current}$ = rainy). Here "current+1" means the same thing as "tomorrow." In an acausal relation the decision attribute's value is not caused by the condition attributes, but just happen 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. This *test* gives us a measure for judging the acausality of a relationship, though usually we do not know the value of future observations and hence cannot execute an acausal rule in real-time. However, referring to them in a rule when dealing with saved data is meaningful, as is shown in the experiments in this paper.

The same method can be used for linear spatial data, where the “past” and “future” can be substituted by “back” and “forward” in a spatial neighbourhood sense [5]. An acausal relationship is a temporal co-occurrence of the attributes. In other words, it is similar to an instantaneous relationship, but spread over time. If there is a need to execute the rules in real-time, one could employ the rules generated using the causal method, which do not reference the future values. But the relationship will not be labelled as causal.

The effect happening before the cause is called *backward causation*, which is considered a mainly philosophical curiosity [11]. The classical problem of time travel is an example of backward causation: a person can return to his past and change the causes of his travel in time. Some philosophers do not consider this impossible or paradoxical. It is possible that in the absence of free will, we go back to our past and not change anything that interferes with our time travel. An implication of this assumption is that we may not be able to change anything, then travel to the past again and again, and thus be stuck in a never-ending time loop. In this paper we call backward causation as acausality, and propose a computable method for testing its applicability. We consider acausality to imply the presence of hidden common causes. The effects of these common causes are spread over time, and so form a temporal association. This interpretation acknowledges the temporal characteristics of the relationship, and avoids any philosophical paradoxes.

As will be explained, TIMERS II usually does not provide definite verdicts with 100% certainty. Whether TIMERS II judges the system as causal or acausal should be considered a *hint* about the real relationship at work. Another point is that unlike in physics and philosophy, where we either move forward, or move backward in time, in TIMERS II we can reference both the past and the future values in the same rule.

We define any reference to the future to be a sign of acausality, even if there are references to the past observations. In this regard, TIMERS II's treatment of time, as shown in the sliding position temporalisation method of Section 2, is more general than the usual treatment in the literature.

The formal definitions of instantaneous and causal sets of rules are as in [3], and are repeated here for completeness, but the definition of acausality has changed to account for the possibility of time moving in both forward and backward directions.

Instantaneous. In this kind of rule the current value of the decision attribute is determined solely by the current values of the condition attributes.

For any rule r in rule set R, if the decision attribute d appears at time T, then all condition attributes should also appear at time T, i.e., R is instantaneous iff ($\forall\ r \in R$, if d_T = DECISION(r), then $\forall\ a_t \in$ CONDITION(r), $t = T$). [3]

Causal. In a *causal* set of rules, the current value of the decision attribute relies only on the previous values of the condition attributes.

For any rule r in the rule set R, if the decision attribute d appears at time T, then all condition attributes should appear at time $t < T$, i.e., R is causal iff ($\forall\ r \in R$, if d_T = DECISION(r), then $\forall\ a_t \in$ CONDITION(r), $t < T$). [3]

Acausal. In an *acausal* set of rules, the current value of the decision attribute relies on the future value of at least one condition attribute.

For any rule r in the rule set R, if the decision attribute d appears at time T, then at least one condition attribute should appear at time $t > T$. i.e., R is acausal iff ($\forall\ r \in R$, if d_T = DECISION(r), then $\forall\ a_t \in$ CONDITION(r), $t \neq T \wedge \exists\ a_u \in$ CONDITION(r), $u > T$).

TIMERS II performs three tests: One for the instantaneous case, one for the causal case, and one for the acausal case. The resulting rulesets are then evaluated, and the one with highest quality is used to declare the nature of the relationship. In other words, the ability to predict the value of a decision attribute is used for judging the relationship as instantaneous, casual, or acausal. In this paper we use the training and predictive accuracy values of the rules as the quality measure.

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 $<_{\text{simplicity}}$ acausal $<_{\text{simplicity}}$ causal. The intuition behind this ordering is that as we move from instantaneous to acausal and 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.

The rest of the paper is organised as follows. Section 2 introduces temporalisation, which is the process we use to merge consecutive records together in different ways. This pre-processing technique allows us to bring together the causes and the effects into the same record. Section 3 introduces the TIMERS II algorithm. Section 4 presents a number of experimental results obtained from TIMERS II. Section 5 concludes the paper.

2. Temporalisation

Normally, to determine the value of a decision attribute, we use the condition attributes from the same record. An example such record sequence would be <3, Left>, <2, Left>, <1, Right>, <2, Right>, etc. Each record indicates the current position along a line, and the direction of movement at that position, considered to have been generated randomly. Any rule derived from such atemporal data involves attribute values that are seen at the same time. In this case, we may use the current movement direction (the condition attribute) to determine the current position (the decision attribute). The results would be instantaneous rules, and we can tell that they probably will not have good accuracy values, because there is no inherent relationship between a position and the random direction of movement at that position. To explore causality, we use the intuitive notion that the condition attributes' effects take time to appear, and thus are seen in the future records. Given a temporally sorted sequence of records, we merge consequent records into one record, bringing the possible causes and the effect together. We call this operation *temporalising*, and the number of records merged is determined by the *window size*. Temporalisation enables us to use normal tools and applications (that do not consider the passage of time), for the purpose of temporal analysis of data.

An example record sequence with a window size of two in the forward direction of time, going from past to the current time, would be <3, Left, 2, Left>, <2, Left, 1, Right>, <1, Right, 2, Right>, etc., where each record includes data from two time steps. Here we have the previous position and movement direction, as well as the current values. Obviously, given the previous position and the previous direction of movement, it is easy to determine the current movement. All two consecutive records are thus merged, and each record will contain the causes (the first record) and the effects (the second record) in turn. If the rules derived from these temporalised records result in a better accuracy value than the original records, then we declare the relationship between the current position and other attributes at causal. In this example we can expect very good results because when we know the past position and the past movement, we can say with certainty where we end up (assuming a perfect world where actions do not fail). However, we may be dealing with a temporal relation that is not causal. For this reason, we should also consider the possibility that the future position and the future direction of movement are creating the current position. A temporalised record would now look like: <2, Left, 3, Left>. In this particular example this hypothesis is not as good as the causal one, because knowing where we are in the future is not sufficient to know where we are now (there are two possibilities: being to the left or to the right of the future position).

The temporalisation technique prepares the data for rule extraction, and the final judgment about the type of the relationship is based on the quality of the rules. The quality can be measured using the rules' accuracy value. For the instantaneous test, no temporalisation is performed. Alternatively, one could say we temporalise with a window size of 1. For the causal test, temporalisation involves merging every w consecutive records together, and setting the decision attribute to be in the last record (past predicting the future with a window size of w). For acausality, the direction of time is considered to point from future to the past, and the decision attribute can be anywhere from the first record to w-1, so the temporalised record has some elements from the future.

TIMERS I temporalised records by considering only the past records (forward temporalisation: the normal direction of time) or only the future records (backward temporalisation). TIMERS II introduces the *sliding position* temporalisation method which includes forward and backward temporalisation as special cases. The principle behind the sliding position method is that both previous and next records can be influential in determining the current value of the decision attribute. With any fixed window size w, the new temporalisation algorithm first places the current decision attribute at position one, and uses the next w-1 records to predict its value. This corresponds to a backward temporalisation. Then the current attribute is set at position 2, and the previous record (position one) and the next *w*-2 records are used for prediction. This case has no correspondence in our previous algorithm in [3]. This movement of the current position continues and at the end it is set to *w*, and the previous *w*-1 records are used for prediction. This corresponds to forward temporalisation.

As an example consider four temporally consecutive records, each with four fields: R1: <1, 2, 4, true>, R2:<2, 3, 5, false>, R3:<6, 7, 8, true>, R4:<5, 2, 3, true>. Suppose we are interested in predicting the value of the last (Boolean) variable. Using a window of size 3, we can merge them as in Table 2. The decision attribute is indicated in **bold** characters. When it comes to the record involving the decision attribute, we do not consider any condition attributes in the same record as the decision [3]. The *Record.value* notation means that we are only including the decision attribute. For example, <R_1, R_2, R_3.**false**> would contain <1, 2, 4, true, 2, 3, 5, true, **false**>, where *false* is the decision attribute in R_3. This omission makes sure that minimum amount of data is shared between the original (instantaneous) record and the temporalised record.

Instantaneous. $w = 1$ (original data)	Forward (Causality). $w = 3$	Backward (Acausality). $w = 3$	Sliding position. $w = 3$
R_1 = <1, 2, 4, **true**>	<R_1, R_2, R_3.**false**>	<R_3, R_2, R_1.**true**>	<R_2, R_3, R_1.**true**>
R_2 = <2, 3, 5, **true**>	<R_2, R_3, R_4.**true**>	<R_4, R_3, R_2.**true**>	<R_1, R_3, R_2.**true**>
R_3 = <6, 7, 8, **false**>			<R_1, R_2, R_3.**false**>
R_4 = <5, 2, 3, **true**>			<R_3, R_4, R_2.**true**>
			<R_2, R_4, R_3.**false**>
			<R_2, R_3, R_4.**true**>

Table 2. Results of temporalisation using the forward, backward, and sliding position methods

For the acausal test, we can have a mix of past and future attributes. Given a window size *w*, *p* previous records and *f* future records can be involved, with the decision attribute happening in between. So we have $p+1+f = w$. The "1" in this equation indicates the location of the decision attribute at the current time. The requirement is that *f* be at least 1 (at least one record from the future for the acausality test to be valid), so we have $1 \le f \le w-1$, and $0 \le p \le w-2$. The decision attribute's position slides in the merged records. It moves from being in the first position (no past records) to being in record number *w*-1 (w-2 previous records, 1 future record). The sliding position temporalisation is presented in Figure 1.

```
for (i = 0; i ≤ |D| - w; i++) {
    temporalisedRecord = <>
    for (j = 1; j < pos, j++)          // previous records
       temporalisedRecord += D_i+j
    for (j = pos + 1; j ≤ w, j++)      // next records
       temporalisedRecord += D_I+j
     temporalisedRecord += Field(d, D_I+pos)  // decision attribute
     output(temporalisedRecord)
}
```

Figure 1. The Sliding position temporalisation method

The temporalisation operator *Temporalise*(*w*, *pos*, ***D***, *d*) takes as input a window size *w*, The position of the decision attribute within the window *pos*, the input records ***D***, and the decision attribute *d*, and outputs temporalised records. D_i returns the *i*th record in the input **D**. *Field*() returns a single field in a record, as specified by its first variable. The += operator stands for concatenating the left hand side with the right hand side, with the results going to the left hand side variable. <> denotes an empty record.

This algorithm covers all three temporalisation methods: 1) For the instantaneous test, we provide it with a window size of 1 and a position of 1. Alternatively we could refrain from using the algorithm and simply employ the original input data. 2) For the causality test, window size w would be any desired value bigger than 1, and the position of the decision attribute would be w (last record). 3) For the acausality test, the window size could be set to any value bigger than 1, and the position of the decision attribute would change between 1 and w-1.

The temporalisation function is called by the TIMERS II algorithm which is provided in Figure 2. Given $|D|$ input records, For each run, *Temporalise*() generates $|D|$ - (w - 1) temporalised records. Since it may not be obvious which window size is more appropriate for a particular dataset, we consider trying a range of values, and the one that results in best accuracy value will be considered for decision making.

Input: A sequence of sequentially ordered data records D, minimum and maximum temporalisation window sizes α and β, where $0 < \alpha \leq \beta$, 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, α , β, Ac_th, d, cl, preference)
{
  ac_i = RuleGenerator(D, d);  // instantaneous accuracy. window size = 1
  for (win = α  to β)
        for(pos = 1 to win)
            (ac_w,pos , ruleSize_w,pos) = RuleGenerator(Temporalise(win, pos, D, d), d)
        end for
  end for

  ac_c = max(ac_α,α, …, ac_β,β)  // best causal test
  ac_a = max(ac_α,pos1, …, ac_β,pos2),  ∀ ac_x,pos, 1 ≤ pos < x  // best acausal result

  // Maybe there is not enough related information?
  if (ac_th > max(ac_i, ac_c, ac_a)) then stop.

  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.
}
```

Figure 2. TIMERS II algorithm for discovering the nature of the relationship for a decision attribute

3. The TIMERS II Algorithm

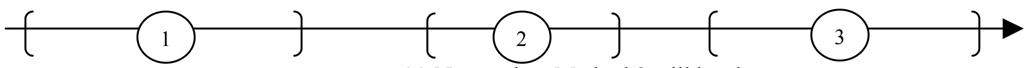
The TIMERS II algorithm is presented in Figure 2. It has been implemented in an application programme called TimeSleuth [2]. TIMERS II first performs the instantaneous test. After that a range of window sizes are tried for the sliding position temporalisation. The resulting temporalised data are fed 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.

The memory space needed by TIMERS II is computed as follows. For every run of the *Temporalise*() operator, we get a dataset of $|D|$-(w-1) records, hence the total number of the output records created by the TIMERS algorithm is $\sum_{w=\alpha}^{\beta} |D| - (w-1)$, where α and β are the starting and ending window sizes. For a window size of 1, the dataset already exists (the original dataset). There is no need to save each temporalised dataset after it has been used for rule generation. So there would be a maximum of $|D|-(\beta - 1)$ temporalised records at any iteration. Considering that the number of attributes in each record is multiplied by the window size, the maximum number of fields in the temporalised dataset will be $\beta \times (|D|-(\beta - 1)) \times$ <number of fields in each input data record>. Computation wise, the number of times that RuleGenerator() runs is equal to $1 + \sum_{w=\alpha}^{\beta} w = 1 + [\beta \times (\beta + 1) - (\alpha - 1) \times \alpha] / 2$. Hence the time complexity of TIMERS II is polynomially related to the time complexity of the RuleGenerator().

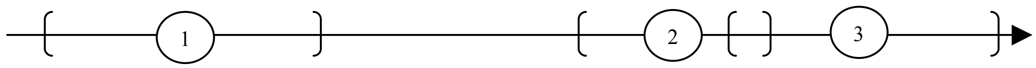
We use the accuracy and complexity of the rules obtained from each method to choose the best relation type that applies to the data. Normally the method with the highest accuracy value would be selected. However, it may happen that the accuracy values are close to each other. In such cases we may choose the simpler relation type because the gains of choosing another type may not be worth the extra complexity. Users can employ their discretion in making this decision. However, TIMERS II proposes a statistical method for this purpose. 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 [13]. 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 similar results. After selecting a winner between the first two methods, the winning relation type is tested against the third relation type using a similar comparison of intervals, and the results determine the final winner.

As an example, suppose with a confidence level of 90%, we have the following accuracy and interval values: 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. Then we consider the causal case, and 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 where every two neighbouring 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.

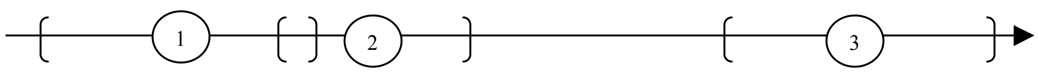
Figures 3(a) to 3(e) show the different possibilities for the accuracy intervals and also the winning method. The circles 1 to 3 represent the accuracy value of a method. The accuracy values are sorted in an ascending order. The brackets show the accuracy intervals. In this Figure we assume that in all three cases the generated rules have the same size.



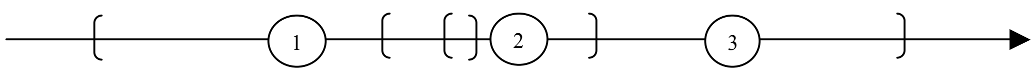
(a) No overlap. Method 3 will be chosen.



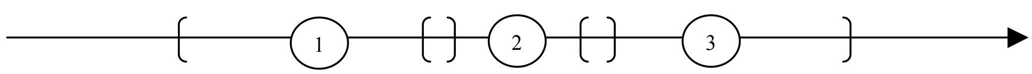
(b) Methods 2 and 3 overlap. The simpler of methods 2 and 3 is chosen.



(c) Methods 1 and 2 overlap. Method 3 is chosen.



(d) All three methods overlap. The simplest method is chosen.



(e) Neighbours overlap. If prefer higher accuracy, 3 is chosen. Otherwise 1 is chosen.

Figure 3. Different possibilities for the accuracy intervals

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 an 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 we assume that 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 override a more complex method, not only should there be an overlap between their accuracy intervals, but the simpler method should result in less rules and/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 between the first two methods. This winning relation type is then compared with the third method to determine the final method. Figure 4 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 = ComputeAccuracyInterval(accuracy_method))

  // if preference is given to higher accuracy, then start the search from lower accuracy values
  if(p == HIGHER_ACCURACY) then
    sort_Ascending(info[]); // sort in ascending order of accuracy.
  else  // SIMPLER_METHOD
    sort_Descending(info[])

  winner = 1
  for(count = 2 to 3)
    if (overlap(info[winner].interval, info[count].interval)
    { // if there is an overlap, then choose the simpler method
      if(info[count].method <_simplicity info[winner].method and
         info[count].ruleSize ≤ 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
    }
  end for
  return info[winner].method  //one of INSTANTANEOUS, ACAUSAL, or CAUSAL
}
```

Figure 4. Selecting the best Relationship

If needed, the algorithm in Figure 4 can also be used to select the best window size among a number of accuracy values obtained in either the acausal or casual case. In that case the order of simplicity is determined by the window size, with bigger window sizes being less simple.

4. Experimental Results

In this section we will use two temporal datasets. The first one is from an artificial life program called URAL [14], and involves an artificial robot randomly moving left, right, up and down through an 8 × 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. We consider the results of this test as a form of "sanity check" and have been using them as such in our papers. The second dataset is from a weather station in Louisiana. It includes 343 training records, each with 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 generating predictive accuracy values.

4.1 The Artificial Robot

Each record in this dataset contains *x* and *y* position values at any given time, the direction of movement at that time, and also a binary variable indicating the presence or absence of food. 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 3. T Accuracy means training accuracy, where the same data are used for both deriving and testing the rules. P Accuracy means Predictive accuracy, where the rules are tested on data other than those used for deriving them. In the "Actual rules" column, the actual type of the rules is displayed. Even though an acausal temporalisation method could be used, the output rules may not have any references to attributes that appear after the decision attribute. In this case the results are considered to be causal.

Window	Position	T Accuracy	P Accuracy	Type of test	Actual rules
1	1	19.7%	20.4%	Instantaneous	Instantaneous
2	1	56.2	55.7%	Acausal	Acausal
2	2	**100%**	**100%**	Causal	Causal
3	1	57.6%	55.6%	Acausal	Acausal
3	2	100%	100%	Acausal	Causal
3	3	100%	100%	Causal	Causal
4	1	58.4%	58.1%	Acausal	Acausal
4	2	100%	100%	Acausal	Causal
4	3	100%	100%	Acausal	Causal
4	4	100%	100%	Causal	Causal
5	1	58.4%	57.1%	Acausal	Acausal
5	2	100%	100%	Acausal	Causal
5	3	100%	100%	Acausal	Causal
5	4	100%	100%	Acausal	Causal
5	5	100%	100%	Causal	Causal

Table 3. TIMERS II's accuracy result with the robot data.

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.

4.2 The weather data

The subject of experiments in this subsection is a real-world dataset from weather observations in Louisiana [15], and hence interpreting the dependencies and relationships is harder. We have set the soil temperature to be the decision attribute. The results obtained are shown in Table 4.

Window	Position	T Accuracy	P Accuracy	Type of test	Actual rules
1	1	27.7%	23.7%	Instantaneous	Instantaneous
2	1	75.1%	59.5%	Acausal	Acausal
2	2	82.7%	67.6%	Causal	Causal
3	1	85.3%	75.0%	Acausal	Acausal
3	2	82.4%	72.7%	Acausal	Acausal
3	3	86.8%	**77.8%**	Causal	Causal
4	1	85.3%	74.3%	Acausal	Acausal
4	2	85.9%	74.3%	Acausal	Acausal
4	3	83.2%	74.3%	Acausal	Acausal
4	4	84.4%	71.4%	Causal	Causal
5	1	85.0%	73.5%	Acausal	Acausal
5	2	**87.0%**	76.5%	Acausal	Acausal
5	3	85.0%	76.5%	Acausal	Acausal
5	4	83.8%	76.5%	Acausal	Acausal
5	5	86.7%	73.5%	Causal	Causal

Table 4. Results of the sliding position temporalisation on the weather data.

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.

5. Concluding Remarks

We presented 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 as they appear in a sequence of records, and hence has a more restricted form of input than other techniques. TIMERS II tests to see whether a time difference between the attributes' values is needed to better predict the value of a decision attribute. If not, then the relationship is instantaneous. If time is required, then a distinction is made as to whether the relationship is causal (past determines the future) or acausal (the future determines the past). Each test is performed after an appropriate type of temporalisation. We used accuracy values of the rules as an indication of appropriateness of the temporalisation method, and hence the type of the relationship, but in general any other measure can be used.

The resulting rules indicate which attributes are important in predicting the value of the decision attribute. They also show how the relationship is formed. For example, in the Louisiana weather data, the soil temperature an hour before has most effect in determining the soil temperature at the current time [4].

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 attribute (the decision attribute) whose value is observed at a reference time (the current time). So defining the future and the past may not be straightforward.

The TimeSleuth package includes executables and source code in Java, as well as online help and example files. It can be downloaded freely from http://www.cs.uregina.ca/~karimi/downloads.html.

References

1. Heckerman, D., Geiger, D. and Chickering, D.M., Learning Bayesian Networks: The Combination of Knowledge and Statistical Data, *Machine Learning*, 20(3), 1995, pp. 197-243.
2. 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.
3. 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.
4. Karimi, K. and Hamilton H.J., Using TimeSleuth for Discovering Temporal/Causal Rules: A Comparison, *The Sixteenth Canadian Artificial Intelligence Conference (AI'2003)*, Halifax, Nova Scotia, Canada, June 2003, pp. 175-189.
5. Karimi, K., and Hamilton, H.J., From Temporal Rules to One Dimensional Rules, *Proceedings of the Workshop on Causality and Causal Discovery, Technical Report CS-2004-02*, Kamran Karimi (Ed.). Department of Computer Science, University of Regina, Canada, May 2004, pp. 30-44.
6. Kennett, R.J., Korb, K.B., and Nicholson, A.E., Seabreeze Prediction Using Bayesian Networks: A Case Study, *Proc. Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'2001).* Hong Kong, April 2001.
7. Krener, A. J. Acausal Realization Theory, Part I; Linear Deterministic Systems. *SIAM Journal on Control and Optimization*. 1987. Vol 25, No 3, pp. 499-525.
8. Pearl, J., *Causality: Models, Reasoning, and Inference*, Cambridge University Press. 2000
9. Scheines, R., Spirtes, P., Glymour, C. and Meek, C., *Tetrad II: Tools for Causal Modeling*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1994.
10. Schwarz, R. J. and Friedland B., *Linear Systems*. McGraw-Hill, New York. 1965.
11. Tooley, M. (Ed.), *Analytical Metaphysics: A Collection of Essays*, Garland Publishing, Inc., 1999.
12. Wallace, C., Korb, K., Dai, H., Causal Discovery Discovery via MML, *13th International Conference on Machine Learning (ICML'1996)*, pp. 516-524, 1996.
13. Witten, I.A., and Frank, E., *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, 2000.
14. http://www.cs.uregina.ca/~karimi/downloads.html/URAL.java
15. http://typhoon.bae.lsu.edu/datatabl/current/sugcurrh.html. Contents change with time.